

Hochflexibles Workforce Management: Herausforderungen und Lösungsverfahren

Dissertation
zur Erlangung des akademischen Grades
eines Doktors der Wirtschaftswissenschaften
(„Doctor rerum politicarum“)
an der Fakultät für Wirtschaftswissenschaften
der Technischen Universität Ilmenau

vorgelegt von
Dipl.-Wirtschaftsinformatiker Maik Günther

Erstgutachter: Univ.-Prof. Dr. Volker Nissen
Zweitgutachter: Univ.-Prof. Dr. Udo Bankhofer

Eingereicht am: 26. Juni 2010
Tag der Verteidigung: 09. Dezember 2010

URN: [urn:nbn:de:gbv:ilm1-2010000468](https://nbn-resolving.org/urn:nbn:de:gbv:ilm1-2010000468)

Kurzfassung

Zunehmend ist bei Unternehmen ein Trend weg von der starren Schicht- oder Dienstplanung hin zu einer auf den Personalbedarf ausgerichteten Planung festzustellen. In diesem Zusammenhang wird der Begriff hochflexibles Workforce Management (WFM) geprägt. Mit Instrumenten wie der Planung untertägiger Arbeitsplatzwechsel, der Kombination aus Arbeitszeitmodellerstellung und Einsatzplanung sowie der kombinierten Personaleinsatz- und Tourenplanung kann der Personaleinsatz sehr gut an den Personalbedarf angepasst werden. Derartige Problemstellungen sind neuartig und fanden in der Forschung bisher wenig Beachtung.

In dieser Arbeit wird anhand von Problemstellungen aus der Logistik, aus dem Handel und von der British Telecom untersucht, ob sich eher klassische OR-Verfahren, Metaheuristiken oder Multiagentensysteme (MAS) für Probleme des hochflexiblen WFM eignen. Es zeigt sich, dass klassische OR-Verfahren keine praktikablen Rechenzeiten aufweisen und somit nicht geeignet sind. Demgegenüber erweisen sich ausgewählte Metaheuristiken als überaus gut geeignet. MAS sind nicht zu bevorzugen, da ausgewählte Metaheuristiken immer bessere Ergebnisse lieferten. Hervorzuheben ist, dass ein hybrider Ansatz aus Metaheuristik und problemspezifischer Reparatur erst für den Erfolg einiger Metaheuristiken verantwortlich ist. MAS fehlt eine zentrale Planungskomponente, die tiefgreifende Änderungen vornimmt, zu denen Agenten nicht in der Lage sind. Zahlreiche Lösungsverfahren, die in dieser Arbeit untersucht werden, sind ursprünglich für kontinuierliche Problemstellungen entwickelt worden. In der Arbeit wird die Anpassung dieser Verfahren an kombinatorische Problemstellung detailliert beschrieben. Dabei wird auch auf die geeignete Parametrierung eingegangen.

Abstract

It can be observed that companies tend to use a very demand driven personnel scheduling instead of using fixed shifts. In this context the term highly flexible workforce management (WFM) is used. With instruments such as the planning of subdaily workplace rotations, the combination of working time model generation and personnel scheduling or the combination of personnel scheduling and vehicle routing the demand for personnel can be covered very well. Such problems are novel and found little attention by researchers up to now.

In this work classical OR-algorithms, metaheuristics and multi-agent systems (MAS) are evaluated on real world problems from logistics, retail and British Telecom. It can be shown, that classical OR-algorithms are not appropriate for these problems of highly flexible WFM, because of impractical CPU-times. On the other hand selected metaheuristics are very suitable. MAS should not be favoured, because selected metaheuristics performed always better. It must point out that a hybrid algorithm (a metaheuristic with a problem-specific repair) is responsible for the success of metaheuristics. MAS lack of a central planning instance that makes major changes for which agents are not able to do. Numerous algorithms of this work where originally developed for continuous problems. The adaption to combinatorial problems is described too. The appropriate adaption of parameters is also addressed.

Vorwort

Gerade in Branchen, in denen die Personalkosten einen Großteil der Gesamtkosten ausmachen, ist der bedarfsorientierte Personaleinsatz für den Unternehmenserfolg maßgeblich mit entscheidend. Dies gilt besonders für Hochlohnländer. Doch die Planung des Personaleinsatzes ist nicht trivial – gerade wenn es sich um Problemstellungen des hochflexiblen Workforce Managements handelt. Bei ihnen sind z.B. untertägige Arbeitsplatzwechsel, sehr kleine Zeitintervalle und lange Planungszeiträume sowie flexible Arbeitszeitmodelle oder sogar Touren zu berücksichtigen. Eine Vielzahl an Zeitintervallen und komplexe Regelungen kennzeichnen diese Probleme. Die automatische Erstellung von Einsatzplänen ist daher sehr fordernd. In dieser Arbeit werden zahlreiche Optimierungsverfahren auf ihre Eignung für derartige Problemstellungen überprüft. So lassen sich Handlungsempfehlungen für die Wahl des besten Verfahrens je nach Planungsszenario ableiten. Gerade die Frage, ob sich eher klassische OR-Verfahren, Metaheuristiken oder Multiagentensysteme für derartige Probleme eignen, steht in dieser Arbeit im Vordergrund.

Ein besonderer Dank gilt meinem Doktorvater Univ.-Prof. Dr. Volker Nissen für die Betreuung und die professionelle Zusammenarbeit, die ich immer sehr geschätzt habe. Er hat meine Arbeit mit Anregungen, Denkanstößen und konstruktiver Kritik in die richtigen Bahnen gelenkt. Zudem danke ich Univ.-Prof. Dr. Udo Bankhofer für die Übernahme des Zweitgutachtens.

Ebenso möchte ich mich bei den zahlreichen Personen bedanken, die mich mit wertvollen Anregungen und spannenden Diskussionen begleitet haben. Allen voran die Mitarbeiterinnen und Mitarbeiter des Fachgebiets Wirtschaftsinformatik für Dienstleistungen der TU Ilmenau. Mein Dank gilt zudem Dr. Ole Brodersen, Univ.-Prof. Dr. Andreas Fink, Dr. Jens Czogalla, Prof. Dr. Raik Stolletz, Prof. Dr. Edward Tsang, Tim Curtois, René Schumann, René Birnstiel sowie Thomas Helbig.

Ohne das Verständnis meiner Freundin, Nadine Haugg, wäre diese Arbeit nicht möglich gewesen. Wenig gemeinsame Zeit und straff organisierte Wochenenden waren in den letzten Jahren die Regel. In diese Arbeit sind mehrere tausend Stunden an CPU-Zeit eingeflossen – auch das ewig monotone dröhnen des Rechners hat sie mit engelsgleicher Geduld ertragen. Ebenso danke ich meinen Eltern, Arnold und Carmen Günther, die mich immer unterstützt und gefördert haben.

Diese Arbeit habe ich als externer Doktorand erstellt. Ohne die ATOSS Software AG, die mir die Möglichkeit gegeben hat, meine Arbeitszeit sehr flexibel zu gestalten, hätte sich die Fertigstellung der Dissertation wohl verlängert. Mein Dank gilt daher den Mitarbeiterinnen und Mitarbeitern der ATOSS Software AG.

Während der Arbeit an meiner Dissertation sind mehr als 20 Publikationen entstanden. Diese zielen entweder auf Algorithmen und deren Eignung für hochflexibles Workforce Management ab [146], [287], [159], [160], [161], [164], [162], [165], [163], [288], [289] oder auf branchenspezifische Aspekte, Nutzenpotenziale und Prozesse [147], [157], [149], [150], [148], [158], [166], [156], [151], [152], [154], [155]. Bei der Arbeit an meiner Dissertation konnte ich maßgeblich von diesen Publikationen profitieren und schon früh hilfreiches Feedback von Gutachtern und Herausgebern erhalten.

Maik Günther

Inhaltsverzeichnis

Abbildungsverzeichnis	xii
Tabellenverzeichnis	xiv
Abkürzungsverzeichnis	xvii
1 Einleitung	1
1.1 Motivation	1
1.2 Methodik der Forschung und Forschungsfragen	3
1.3 Aufbau der Arbeit	5
2 Begrifflichkeiten und Zusammenhänge	8
2.1 Begrifflichkeiten	8
2.1.1 Personalplanung	8
2.1.2 Personaleinsatzplanung	9
2.1.3 Hochflexibles Workforce Management	10
2.2 Nutzen durch Workforce Management	11
2.3 Planungsprozess im Workforce Management	14
2.3.1 Phase 1: Personalbedarfsermittlung	14
2.3.2 Phase 2: Arbeitszeitmodellerstellung	15
2.3.3 Phase 3: Personaleinsatzplanung	15
2.3.4 Phase 4: Arbeitszeitmanagement	16
2.3.5 Zusammenlegung von Phasen	16
2.4 Rahmenbedingungen des Workforce Managements	17
2.5 Typische Planungstools in der Praxis	21
3 Problemstellungen für hochflexibles Workforce Management	25
3.1 Klassifikation von Problemstellungen	26
3.2 Untertägige Personaleinsatzplanung in der Logistik	29
3.2.1 Untertägige Personaleinsatzplanung	29
3.2.2 Beschreibung der Problemstellung	31
3.2.3 Varianten der Problemstellung	35
3.2.4 Einordnung der Problemstellung	38
3.2.5 Vergleichbare Problemstellungen in der Literatur	38
3.3 Automatische Arbeitszeitmodellerstellung im Handel	40
3.3.1 Automatische Arbeitszeitmodellerstellung	40

3.3.2	Beschreibung der Problemstellung	43
3.3.3	Varianten der Problemstellung	47
3.3.4	Einordnung der Problemstellung	48
3.3.5	Vergleichbare Problemstellungen in der Literatur	48
3.4	Kombinierte Personaleinsatzplanung und Tourenplanung bei Technikern	50
3.4.1	Kombination von Touren- und Personaleinsatzplanung	50
3.4.2	Beschreibung der Problemstellung	52
3.4.3	Varianten der Problemstellung	58
3.4.4	Einordnung der Problemstellung	59
3.4.5	Vergleichbare Problemstellungen in der Literatur	60
4	Lösungsverfahren für hochflexibles Workforce Management	65
4.1	Abbildung der Problemstellungen	65
4.1.1	Vorüberlegungen	66
4.1.2	Untertägige Personaleinsatzplanung in der Logistik	68
4.1.3	Automatische Arbeitszeitmodellerstellung im Handel	70
4.1.4	Kombination von Touren- und Personaleinsatzplanung	71
4.2	Komplexität	72
4.3	Abbruchkriterium	75
4.4	Allgemeiner Überblick über Lösungsverfahren	77
4.5	Ganzzahlige Optimierung	81
4.5.1	Grundlegender Ablauf	83
4.5.1.1	MIP-Verfahren	83
4.5.1.2	MINLP-Verfahren	88
4.5.1.3	GAMS-Solver	89
4.5.2	Anwendungen in der Literatur	92
4.5.3	Anpassung an hochflexibles Workforce Management	95
4.5.3.1	Untertägige Personaleinsatzplanung in der Logistik	95
4.6	Konstruierende Verfahren	103
4.6.1	Grundlegender Ablauf	103
4.6.2	Anwendungen in der Literatur	103
4.6.3	Anpassung an hochflexibles Workforce Management	104
4.6.3.1	Automatische Arbeitszeitmodellerstellung im Handel	105
4.6.3.2	Kombination von Touren- und Personaleinsatzplanung	106
4.7	Constraint Logic Programming	107
4.7.1	Grundlegender Ablauf	108
4.7.2	Anwendungen in der Literatur	110
4.7.3	Anpassung an hochflexibles Workforce Management	110
4.7.3.1	Kombination von Touren- und Personaleinsatzplanung	110
4.8	Hill Climbing	114
4.8.1	Grundlegender Ablauf	115
4.8.2	Anwendungen in der Literatur	116
4.8.3	Anpassung an hochflexibles Workforce Management	117
4.8.3.1	Untertägige Personaleinsatzplanung in der Logistik	117

4.8.3.2	Automatische Arbeitszeitmodellerstellung im Handel . . .	118
4.8.3.3	Kombination von Touren- und Personaleinsatzplanung . . .	119
4.9	Fast Local Search und Guided Local Search	119
4.9.1	Grundlegender Ablauf	120
4.9.1.1	Fast Local Search	120
4.9.1.2	Guided Local Search	120
4.9.2	Anwendungen in der Literatur	122
4.9.3	Anpassung an hochflexibles Workforce Management	123
4.9.3.1	Kombination von Touren- und Personaleinsatzplanung . . .	123
4.10	Tabu Search	125
4.10.1	Grundlegender Ablauf	125
4.10.2	Anwendungen in der Literatur	126
4.10.3	Anpassung an hochflexibles Workforce Management	127
4.10.3.1	Untertägige Personaleinsatzplanung in der Logistik . . .	127
4.11	Simulated Annealing	133
4.11.1	Grundlegender Ablauf	133
4.11.2	Anwendungen in der Literatur	134
4.11.3	Anpassung an hochflexibles Workforce Management	136
4.11.3.1	Kombination von Touren- und Personaleinsatzplanung . . .	136
4.12	Evolutionsstrategien	137
4.12.1	Grundlegende Zusammenhänge	137
4.12.2	Grundlegender Ablauf	140
4.12.2.1	Rekombination	141
4.12.2.2	Mutation	142
4.12.2.3	Selektion	145
4.12.2.4	ES-Varianten	147
4.12.3	Anwendungen in der Literatur	148
4.12.4	Anpassung an hochflexibles Workforce Management	149
4.12.4.1	Untertägige Personaleinsatzplanung in der Logistik . . .	150
4.12.4.2	Automatische Arbeitszeitmodellerstellung im Handel . . .	155
4.12.4.3	Kombination von Touren- und Personaleinsatzplanung . . .	157
4.13	Genetische Algorithmen	159
4.13.1	Grundlegender Ablauf	160
4.13.2	Anwendungen in der Literatur	162
4.13.3	Anpassung an hochflexibles Workforce Management	163
4.13.3.1	Kombination von Touren- und Personaleinsatzplanung . . .	163
4.14	Particle Swarm Optimization	163
4.14.1	Grundlegende Zusammenhänge	164
4.14.2	Grundlegender Ablauf	166
4.14.2.1	Nachbarschaftstopologien	170
4.14.2.2	Anpassung an binäre und diskrete Lösungsräume	173
4.14.2.3	Anpassung an kombinatorische Problemstellungen	174
4.14.3	Anpassung an hochflexibles Workforce Management	178
4.14.3.1	Untertägige Personaleinsatzplanung in der Logistik . . .	178

4.14.3.2	Automatische Arbeitszeitmodellerstellung im Handel . . .	181
4.14.3.3	Kombination von Touren- und Personaleinsatzplanung . . .	182
4.15	Multiagentensysteme	184
4.15.1	Grundlegender Ablauf	185
4.15.2	Anwendungen in der Literatur	186
4.15.3	Anpassung an hochflexibles Workforce Management	189
4.15.3.1	Vorüberlegungen	189
4.15.3.2	Untertägige Personaleinsatzplanung in der Logistik	191
4.15.3.3	Automatische Arbeitszeitmodellerstellung im Handel	195
4.15.3.4	Kombination von Touren- und Personaleinsatzplanung	198
4.16	Hybride Lösungsverfahren	205
4.16.1	Anwendungen in der Literatur	206
4.16.2	Anpassung an hochflexibles Workforce Management	206
4.16.2.1	Untertägige Personaleinsatzplanung in der Logistik	206
4.16.2.2	Automatische Arbeitszeitmodellerstellung im Handel	210
4.16.2.3	Kombination von Touren- und Personaleinsatzplanung	211
5	Rahmenbedingungen für Implementierung und Experimente	214
5.1	Die Softwarelösung ARROW	214
5.2	XML-Format für Workforce Management-Probleme	216
5.2.1	Existierende Datenformate	217
5.2.1.1	Allgemeiner Überblick	217
5.2.1.2	TTML: Timetabling Markup Language	218
5.2.1.3	Format der ASAP	218
5.2.2	Anforderungen an ein Datenformat	219
5.2.3	Erweiterung des Datenformates der ASAP	220
5.3	Statistische Grundlagen	224
5.3.1	Hypothesen und Irrtumswahrscheinlichkeiten	224
5.3.2	Levene-Test	226
5.3.3	t-Test für unabhängige Stichproben	227
5.3.4	Poweranalyse	228
5.4	Beurteilungskriterien	230
6	Experimente und Ergebnisse	233
6.1	Untertägige Personaleinsatzplanung in der Logistik	234
6.1.1	Lower Bound	234
6.1.2	Ganzzahlige Optimierung	237
6.1.3	Hill Climbing	240
6.1.4	Guided Local Search	242
6.1.5	Tabu Search	243
6.1.6	Evolutionsstrategien	245
6.1.7	Particle Swarm Optimization	249
6.1.8	Multiagentensysteme	253
6.1.9	Hybride Lösungsverfahren	254

6.1.10	Vergleich der Lösungsverfahren	257
6.2	Automatische Arbeitszeitmodellerstellung im Handel	262
6.2.1	Lower Bound	263
6.2.2	Ganzzahlige Optimierung	264
6.2.3	Konstruierende Verfahren	265
6.2.4	Hill Climbing	267
6.2.5	Guided Local Search	268
6.2.6	Evolutionsstrategien	269
6.2.7	Particle Swarm Optimization	271
6.2.8	Multiagentensysteme	273
6.2.9	Vergleich der Lösungsverfahren	274
6.3	Kombinierte Personaleinsatzplanung und Tourenplanung bei Technikern .	279
6.3.1	Lower Bound	280
6.3.2	Ganzzahlige Optimierung	280
6.3.3	Konstruierende Verfahren	281
6.3.4	Constraint Logic Programming	281
6.3.5	Hill Climbing	283
6.3.6	Fast Local Search und Guided Local Search	284
6.3.7	Simulated Annealing	285
6.3.8	Evolutionsstrategien	285
6.3.9	Genetische Algorithmen	287
6.3.10	Particle Swarm Optimization	288
6.3.11	Multiagentensysteme	290
6.3.12	Vergleich der Lösungsverfahren	292
6.4	Abschließende Beurteilung der Lösungsverfahren	295
7	Zusammenfassung	303
7.1	Fazit	303
7.2	Zukünftige Forschungsgegenstände	305
	Literaturverzeichnis	xxi
	Anhang	lx
7.3	Beispielproblem im XML-Format	lx
7.4	Ergebnistabellen	lxii

Abbildungsverzeichnis

2.1	Ursachen für Unproduktivität [314]	12
2.2	Über- und Unterdeckung im Personaleinsatz	13
2.3	Idealisierter Planungsprozess im Workforce Management	14
2.4	Rahmenbedingungen für den Planer	18
2.5	Nutzung von Tools zur Personaleinsatzplanung	22
3.1	Personalbedarf beim Logistikproblem im Zeitverlauf eines Tages bei neun Arbeitsplätzen (Ausschnitt 3.30–17.30 Uhr)	32
3.2	Starrer Personaleinsatz (drei Schichten) bei schwankendem Personalbedarf	41
3.3	Flexibler Personaleinsatz (automatische Arbeitszeitmodellerstellung) bei schwankendem Personalbedarf	42
3.4	Aufbau des Dynamic Schedulers	52
3.5	Position der Aufträge und Servicecenter	55
3.6	Ablauf des Gebots- und Rücknahmeprozesses	62
4.1	Abgrenzung der Komplexitätsklassen (in Anlehnung an [417])	73
4.2	Verlauf der Fehlerpunkte für ES (1,5) TypE + R bei Handel_2AP	76
4.3	Einordnung von Lösungsverfahren	82
4.4	Zusammenhang zwischen LP, MIP, NLP und MINLP	83
4.5	Grafische Darstellung eines ganzzahligen linearen Optimierungsproblems	85
4.6	Grafische Darstellung eines ganzzahligen linearen Optimierungsproblems mit Schnitt	86
4.7	Enumerationsbaum bei Branch & Bound	87
4.8	Variablenbelegung in Baumdarstellung	108
4.9	Variablenbelegung in Baumdarstellung nach Domänenreduktion	109
4.10	Nachbarschaft bzgl. des Mitarbeiters	128
4.11	Nachbarschaft bzgl. des Zeitintervalls	129
4.12	Shift-Move	130
4.13	Swap-Move	131
4.14	Move-Operatoren für Simulated Annealing	136
4.15	Vereinfachte Darstellung der Evolution in der Biologie	138
4.16	Schematischer Ablauf eines EA	139
4.17	Zusammenhang zwischen Fortschrittsgeschwindigkeit und Erfolgswahrscheinlichkeit der Mutation beim Korridor- und Kugelmodell	143
4.18	Evolutionenfenster	146
4.19	1-Punkt-Crossover bei zwei Individuen	161

4.20	Regeln der Boids	165
4.21	Methoden zur Behandlung von Dimensionsüberschreitungen	169
4.22	Nachbarschaftstopologien gBest, Circle ($k = 2$), Wheel, Cluster und Pyramide	171
4.23	Ablauf des Agentenansatzes für die untertägige Personaleinsatzplanung in der Logistik	191
4.24	Ablauf des Agentenansatzes für die automatische Arbeitszeitmodellerstellung im Handel mit zwei Arbeitsplätzen	195
4.25	Ablauf des Agentenansatzes von Kokkoras für die Kombination von Touren- und Personaleinsatzplanung	199
4.26	Position der Aufträge, Servicecenter und des Gravitationszentrums	202
4.27	Ablauf des eigenen Agentenansatzes für die Kombination von Touren- und Personaleinsatzplanung	203
4.28	Swap-Move für zwei Zeitintervalle	208
5.1	Planungsdialog mit An-/Abwesenheiten und Personalbedarf in ARROW	216
5.2	Ablehnungsbereiche für H_0 bei einseitigem (links) und zweiseitigem (rechts) Test	225
5.3	Stichprobenumfang (n_1 und $n_2 = 30$) bei einseitigem t -Test der Mittelwerte mit unabhängigen Stichproben	230
5.4	Beurteilungskriterien für Lösungsverfahren	231
6.1	Verlauf der Fehlerpunkte für ES bei Logistik_Complete	247
6.2	Verlauf der Fehlerpunkte für ES bei Logistik_S_Complete	247
6.3	Verlauf der Fehlerpunkte verschiedener Nachbarschaftstopologien für PSO bei Logistik_Complete	251
6.4	Verlauf der Fehlerpunkte verschiedener Nachbarschaftstopologien für PSO bei Logistik_S_Complete	251
6.5	Verlauf der Fehlerpunkte für ES und PSO bei Logistik_Complete	259
6.6	Verlauf der Fehlerpunkte für ES und PSO bei Logistik_S_Complete	259
6.7	Vergleich verschiedener Lösungsverfahren bei Logistik_Complete	261
6.8	Vergleich verschiedener Lösungsverfahren bei Logistik_S_Complete	261
6.9	Verlauf der Fehlerpunkte für ES bei Handel_1AP_Jan	270
6.10	Verlauf der Fehlerpunkte verschiedener Nachbarschaftstopologien für PSO (20) bei Handel_1AP_Jan	272
6.11	Vergleich verschiedener Lösungsverfahren bei Handel_2AP	276
6.12	Vergleich verschiedener Lösungsverfahren bei Handel_2AP_Jan	277
6.13	Vergleich verschiedener Lösungsverfahren bei Handel_1AP	277
6.14	Vergleich verschiedener Lösungsverfahren bei Handel_1AP_Jan	278
6.15	Verlauf der Fehlerpunkte für ES bei BT_Mod-250-118	287
6.16	Verlauf der Fehlerpunkte verschiedener Nachbarschaftstopologien für PSO bei BT_Mod-250-118	289
6.17	Verlauf der Fehlerpunkte für ES und PSO bei BT_Mod-250-118	293
6.18	Vergleich verschiedener Lösungsverfahren bei BT_Mod-250-118	294

Tabellenverzeichnis

3.1	Klassifikation nach Ernst u.a. und Anzahl der dort zuzuordnenden Arbeiten	26
3.2	Arbeitszeitmodelle beim Logistikproblem mit Arbeitsbeginn und -ende	33
3.3	Rahmenbedingungen der Logistikprobleme ohne Stammarbeitsplatzregelung	36
3.4	Rahmenbedingungen der Logistikprobleme mit Stammarbeitsplatzregelung	37
3.5	Effekte durch die Ausweitung der Anzahl an Arbeitszeitmodellen . . .	44
3.6	Rahmenbedingungen der Handelsprobleme	48
3.7	Servicecenter und ihre Kapazitäten	56
3.8	Lösungsverfahren anderer Autoren für das British Telecom-Problem .	61
3.9	Lösungsverfahren im ROADEF-Wettbewerb 2007	64
4.1	Abbildung der Problemstellung beim Logistikproblem	69
4.2	Abbildung der Problemstellung beim Handelsproblem	70
4.3	Abbildung der Problemstellung beim British Telecom-Problem	72
4.4	Übersicht über Lösungsverfahren und Anzahl Arbeiten	78
4.5	Unterschiede der vier EA-Hauptformen	140
4.6	Diskrete Rekombination	142
4.7	Intermediäre Rekombination	142
4.8	Rekombination nach Variante 5	151
4.9	Lösungsrepräsentation mit dem SPV	175
5.1	Fehlerarten bei Hypothesentests	225
5.2	Mittelwert und Standardabweichung der Experimente eins und zwei .	228
5.3	Einfluss verschiedener Effektgrößen auf den Stichprobenumfang . . .	229
6.1	Lower Bounds beim Logistikproblem	235
6.2	Lower Bounds beim Logistikproblem mit Stammarbeitsplatzregelung	236
6.3	Ergebnisse der MIP-Solver	237
6.4	Ergebnisse der MINLP-Solver	239
6.5	Ergebnisse verschiedener Nachbarschaftstopologien an den einzelnen Tagen des Logistikproblems ohne Stammarbeitsplatzregelung	252
6.6	Ergebnisse verschiedener Nachbarschaftstopologien an den einzelnen Tagen des Logistikproblems mit Stammarbeitsplatzregelung	253
6.7	t-Tests (einseitig) bei Logistik_Complete	258

6.8	t-Tests (einseitig) bei Logistik_S_Complete	258
6.9	t-Tests (einseitig) bei Handel_1AP	275
6.10	t-Tests (einseitig) bei Handel_1AP_Jan	275
6.11	t-Tests (einseitig) bei Handel_2AP	275
6.12	t-Tests (einseitig) bei Handel_1AP_Jan	276
6.13	t-Tests (einseitig) bei BT_Mod-250-118	293
7.1	Ergebnisse bei Logistik_Complete (Teil 1)	lxii
7.2	Ergebnisse bei Logistik_Complete (Teil 2)	lxiii
7.3	Ergebnisse bei Logistik_Mo (Teil 1)	lxiv
7.4	Ergebnisse bei Logistik_Mo (Teil 2)	lxv
7.5	Ergebnisse bei Logistik_Di (Teil 1)	lxvi
7.6	Ergebnisse bei Logistik_Di (Teil 2)	lxvii
7.7	Ergebnisse bei Logistik_Mi (Teil 1)	lxviii
7.8	Ergebnisse bei Logistik_Mi (Teil 2)	lxix
7.9	Ergebnisse bei Logistik_Do (Teil 1)	lxx
7.10	Ergebnisse bei Logistik_Do (Teil 2)	lxxi
7.11	Ergebnisse bei Logistik_Fr (Teil 1)	lxxii
7.12	Ergebnisse bei Logistik_Fr (Teil 2)	lxxiii
7.13	Ergebnisse bei Logistik_Sa (Teil 1)	lxxiv
7.14	Ergebnisse bei Logistik_Sa (Teil 2)	lxxv
7.15	Ergebnisse bei Logistik_So (Teil 1)	lxxvi
7.16	Ergebnisse bei Logistik_So (Teil 2)	lxxvii
7.17	Ergebnisse bei Logistik_S_Complete (Teil 1)	lxxviii
7.18	Ergebnisse bei Logistik_S_Complete (Teil 2)	lxxix
7.19	Ergebnisse bei Logistik_S_Mo (Teil 1)	lxxx
7.20	Ergebnisse bei Logistik_S_Mo (Teil 2)	lxxx1
7.21	Ergebnisse bei Logistik_S_Di (Teil 1)	lxxx2
7.22	Ergebnisse bei Logistik_S_Di (Teil 2)	lxxx3
7.23	Ergebnisse bei Logistik_S_Mi (Teil 1)	lxxx4
7.24	Ergebnisse bei Logistik_S_Mi (Teil 2)	lxxx5
7.25	Ergebnisse bei Logistik_S_Do (Teil 1)	lxxx6
7.26	Ergebnisse bei Logistik_S_Do (Teil 2)	lxxx7
7.27	Ergebnisse bei Logistik_S_Fr (Teil 1)	lxxx8
7.28	Ergebnisse bei Logistik_S_Fr (Teil 2)	lxxx9
7.29	Ergebnisse bei Logistik_S_Sa (Teil 1)	xc
7.30	Ergebnisse bei Logistik_S_Sa (Teil 2)	xc1
7.31	Ergebnisse bei Logistik_S_So (Teil 1)	xc2
7.32	Ergebnisse bei Logistik_S_So (Teil 2)	xc3
7.33	Ergebnisse bei Handel_1AP	xc4
7.34	Ergebnisse bei Handel_1AP_Jan	xc5
7.35	Ergebnisse bei Handel_2AP	xc6
7.36	Ergebnisse bei Handel_2AP_Jan	xc7
7.37	Ergebnisse bei BT_Mod-250-118	xc8

Tabellenverzeichnis

7.38	Ergebnisse bei BT_Rel-250-118	xcix
7.39	Ergebnisse bei BT_Tsang-250-118	xcix
7.40	Ergebnisse bei BT_Bristol-250-118	xcix

Abkürzungsverzeichnis

ALCD	Adjust Length Center Demand
ALCV	Adjust Length Center Value
ANSI	American National Standards Institute
ARMS	Automated Resource Management System
ARROW	Automatic Resource and Route Oriented Workforce Management
ASAP	Automated Scheduling, Optimisation and Planning Research Group
ASMR	Automatic Synchronising Multiple Communication Rounds
AZM	Arbeitszeitmodell
B	Balloon
BA	Breakout Algorithmus
BDMLP	Brooke, Drud and Meeraus Linear Program
CG	Compact Generation
CHIP	Constraint Handling in Prolog
CL	Closest First
CLP	Constraint Logic Programming
COIN-OR	Computational Infrastructure for Operations Research
CoinBonmin	Coin Basic Open-Source Nonlinear Mixed Integer Programming
CoinCbc	Coin Branch and Cut
CoinGlpk	Coin Gnu Linear Programming Kit
CoinIpopt	Coin Interior Point Optimizer
CoinScip	Coin Solving Constrained Integer Programs
CP	Constraint Programming
CPU	Central Processing Unit
CRM	Customer Relationship Management
CSPCONS	Communicating Sequential Prolog with Constraints
DAI	Distributed Artificial Intelligence
D-CLP	Distributed Constraint Logic Programming
DICOPT	Discrete and Continuous Optimizer
DFKI	Deutsches Forschungszentrum für Künstliche Intelligenz
DNLP	Discontinuous Nonlinear Programming
DPSO	Discrete Particle Swarm Optimization
ECP	Extended Cutting Plane
ECRC	European Computer-Industry Research Centre

Abkürzungsverzeichnis

ElipSys	ECRC Logic Inference Parallel System
EP	Evolutionäre Programmierung
ES	Evolutionsstrategie
ESO	Evolutionary Swarm Optimization
ETP	Employee Timetabling Problem
FC	Forward Checking
FC-CBJ	Forward Checking and Conflict-Directed Back Jumping
FLS	Fast Local Search
GA	Genetischer Algorithmus
GAMS	General Algebraic Modeling System
GBD	Generalized Benders Decomposition
GENET	Generic Neural Network
GLS	Guided Local Search
GP	Genetische Programmierung
GRASP	Greedy Randomized Adaptive Search Procedure
HC	Hill Climber bzw. Hill Climbing
H-PSO	Hierarchical Particle Swarm Optimization
HS	Heuristic Search
I	Initialisierungsvariante
ILP	Integer Linear Programming
IP	Integer Programming
JADE	Java Agent Development Framework
JFO	Jumping Frog Optimization
JPSO	Jumping Particle Swarm Optimization
JSP	Job-Shop Scheduling Problem
KV	Konstruierendes Verfahren
LISP	List Processing
LB	Least Busy First
LP	Linear Programming
M	Mixed
MAIS	Multi-Agent Information System
MAS	Multiagentensystem
MathML	Mathematical Markup Language
MAX-SAT	Maximum Satisfiability
MCHC	Min-Conflict Hill Climbing
M-DPSO	Multiphase-Discrete Particle Swarm Optimization
MDSDTCTSPTW	Multi Depot Side Dependent Time Constraint Traveling Salesman Problem with Time Windows
MINLP	Mixed Integer Nonlinear Programming
MIP	Mixed Integer Programming
MIQCP	Mixed Integer Quadratically Constrained Programming
MNS	Multi-Neighbourhood Search
MUSE	Mutation-Selektion

MWM	Mobile Workforce Management
NLP	Nonlinear Programming
OA	Outer Approximation
OJ	Optimise Job
PM	Pilot-Methode
PROLOG	Programmation en Logique
PSO	Particle Swarm Optimization
PPS	Produktionsplanung und -steuerung
QAP	Quadratic Assignment Problem
QCP	Quadratically Constrained Programming
R	Reparatur
RCPSP	Resource Constraint Project Scheduling Problem
RECONET	Retractable Contract NET
RHC	Random Hill Climbing
RLFAP	Radio Link Frequency Assignment Problem
RMINLP	Relaxed Mixed Integer Nonlinear Programming
RMIP	Relaxed Mixed Integer Programming
RMIQCP	Relaxed Mixed Integer Quadratically Constrained Programming
ROADEF	Société française de Recherche Opérationnelle et Aide à la Décision
SA	Simulated Annealing
SAT	Satisfiability
SD	Same Direction
SDTCTSPTW	Side Dependent Time Constraint Traveling Salesman Problem with Time Windows
SHC	Steepest Hill Climbing
SM	Schedule Manager
SPV	Smallest Position Value
STTL	Standard Timetabling Language
TFISP	Tactical Fixed Interval Scheduling Problem
TCTSP	Time Constraint Traveling Salesman Problem
TCTSPTW	Time Constraint Traveling Salesman Problem with Time Windows
TG	Tour Generation
TS	Tabu Search
TSP	Traveling Salesman Problem
TTB	Time Table Block
TTL	Timetabling Language
TTML	Timetabling Markup Language
Type	Mutation mit maximaler Entropie
TypG	Gauß-Mutation
VNS	Variable Neighbourhood Search
VRP	Vehicle Routing Problem
WFM	Workforce Management

Kapitel 1

Einleitung

In diesem Kapitel wird zunächst auf die Notwendigkeit von leistungsstarken Softwarelösungen für den effizienten Personaleinsatz eingegangen. In diesem Zusammenhang wird auch die bisherige Forschung zu diesem Thema beleuchtet, wobei einige Defizite aufgezeigt werden. Hieraus ergibt sich die Motivation für das Anfertigen dieser Arbeit. In Kapitel 1.2 wird anschließend die Methodik der Forschung dargestellt und eingeordnet. Dabei werden auch die Forschungsfragen dieser Arbeit hergeleitet. Dieses Kapitel endet mit einem Überblick über den Aufbau dieser Arbeit.

1.1 Motivation

Viele Unternehmen sind durch eine immer stärkere Kundenorientierung und einen zunehmend internationaler werdenden Wettbewerb geprägt. Um unter diesen Rahmenbedingungen bestehen zu können, ergreifen sie Maßnahmen zur Sicherung der Wettbewerbsfähigkeit. Auf der einen Seite versucht man den Umsatz auszuweiten, indem man sich verstärkt um Kundenbindung bemüht sowie neue Märkte und Themen erschließt. Auf der anderen Seite ist man um Kostensenkung bemüht. Ein bedeutender Hebel auf der Kostenseite ist beim Personal zu finden. Blickt man auf Hochlohnländer wie Deutschland, so machen die Personalkosten einen beachtlichen Teil der Gesamtkosten eines Unternehmens aus [356]. Durch den immer stärker werdenden Zwang zur Effizienzsteigerung schreitet seit Jahren auch die Flexibilisierung im Workforce Management (WFM) weiter voran. Typische Flexibilisierungsinstrumente sind u.a. Lebensarbeitszeitkonten, flexible Arbeitszeiten, Arbeit auf Abruf, der Einsatz von Leiharbeitskräften sowie die Schaffung einer homogenen Qualifikationsstruktur.

Trotz dieser Fortschritte ist die Optimierung beim Einsatz von Personal in der Vergangenheit eher stiefmütterlich behandelt worden. Die häufigsten Tools für die Personaleinsatzplanung sind selbst heute noch MS EXCEL™ & Co. [14]. Mit leistungsstarken Lösungen für den bedarfsorientierten Personaleinsatz gelingt es nachweislich, Überstun-

den zu vermeiden, Leerzeiten zu minimieren und den Servicelevel zu erhöhen [336, S. 238–241], [338, S. 59–65]. Das Ziel ist Umsatzsteigerung und Kostensenkung. Der effiziente Personaleinsatz ist eine große Chance, ja sogar eine Notwendigkeit, um auch in Zukunft die Wettbewerbsfähigkeit zu sichern.

Gerade die Möglichkeiten des hochflexiblen WFM bieten enormes Potenzial. Der Fokus liegt dort auf der möglichst passgenauen Deckung des Personalbedarfs, der meist stark schwanken kann – auch untertägig. Zudem sind die Planungsintervalle mit 15 bis 60 Minuten äußerst klein, was eine sehr gute Berücksichtigung des realen Arbeitsvolumens ermöglicht. Hierzu sind meist eine Abkehr von starren Schichten sowie ein Planungsvorgehen erforderlich, welches angrenzende Schritte mit einbezieht. So z.B. die Integration der Arbeitszeitmodellerstellung oder der Tourenplanung in die Personaleinsatzplanung. Problemstellungen des hochflexiblen WFM zeichnen sich durch eine Vielzahl an Zeitintervallen und komplexe Nebenbedingungen aus. Die Komplexität derartiger Planungsszenarien ist erheblich und manuell nicht praktikabel zu bewältigen. Der Einsatz von Optimierungsverfahren zur Erstellung von Einsatzplänen ist zur Realisierung der zuvor genannten Ziele zwingend erforderlich. Mit der vorliegenden Arbeit soll dieser Forderung nachgekommen werden, indem Lösungsverfahren aus den verschiedensten Kategorien an einer Vielzahl an Problemstellungen auf ihre Eignung hin untersucht werden.

Problemstellungen rund um den effizienten Personaleinsatz werden seit Jahrzehnten intensiv bearbeitet. Meist werden sie selbst konstruiert oder reale Problemstellungen werden sehr stark vereinfacht. Zwar lassen sich hierdurch zahlreiche Lösungsverfahren erfolgreich anwenden – die Probleme sind dann jedoch derart verändert, dass die Ergebnisse kaum einen praktischen Nutzen haben [279]. Daher stammen die Problemstellungen in dieser Arbeit aus realen Unternehmen und werden nicht vereinfacht, um die Praxistauglichkeit von Lösungsverfahren beurteilen zu können.

Blickt man auf die Veröffentlichungen der letzten Jahrzehnte, so steht man meist vor dem Dilemma, dass die Problemstellungen nicht öffentlich zugänglich sind und nur ein einziges bzw. sehr wenige Lösungsverfahren an einem konkreten Problem beurteilt werden. Auf die Eignung eines Lösungsverfahrens haben Größe, Komplexität, Nebenbedingungen, etc. der Problemstellung einen entscheidenden Einfluss. Daher sind Erkenntnisse, die an einem Problem gewonnen werden, nicht ohne Weiteres auf andere Probleme übertragbar.

Auch kann festgehalten werden, dass in der Literatur bisher nur wenige Arbeiten zu finden sind, in denen man sich Problemen des hochflexiblen WFM zuwendet. Aus diesen Gründen werden in dieser Arbeit diverse Lösungsverfahren an einer Vielzahl an Problemstellungen des hochflexiblen WFM verglichen und beurteilt. Zudem werden die Echtdaten und die Ergebnisse öffentlich zugänglich gemacht (siehe [153]).

Bei den Problemstellungen dieser Arbeit treten keine dynamischen Änderungen auf. Rescheduling ist daher nicht erforderlich. Untersuchungen in diese Richtung rechtfertigen

durchaus ein gesondertes Dissertationsvorhaben und werden daher nicht angestellt. Bei allen Problemstellungen ist CPU-Zeit kein begrenzender Faktor, denn die Berechnungen können in der Praxis über Nacht laufen.

Nicht alle Lösungsverfahren, die in dieser Arbeit untersucht werden, sind selbst implementiert worden. Gerade bei der Kombination von Personaleinsatz- und Tourenplanung wird auf Ergebnisse anderer Autoren zurückgegriffen. Einige dieser Verfahren wurden nicht auf andere Problemstellungen in dieser Arbeit übertragen. Dies hätte den Rahmen der Arbeit bei Weitem besprengt. In Anbetracht der Tatsache, dass bereits sehr viele Lösungsverfahren untersucht werden, lässt die Implementierung weiterer ähnlicher Lösungsverfahren auch nur einen geringen zusätzlichen Erkenntnisgewinn erwarten. An geeigneter Stelle wird jeweils ein Hinweis gegeben, ob das entsprechende Verfahren selbst implementiert wurde oder nicht und bei welchen Problemstellungen es zum Einsatz kommt.

Im folgenden Kapitel wird die Methodik der Forschung dargestellt und eingeordnet. In diesem Zusammenhang werden auch die Forschungsfragen hergeleitet.

1.2 Methodik der Forschung und Forschungsfragen

Zur Herleitung der Zielsetzung dieser Arbeit wird sich an den vier Stufen Theorien, Forschungsmethoden, Forschungsgegenstände und Forschungsziele orientiert [357, S. 4]. Die Haupttheorie dieser Arbeit ist die Entscheidungstheorie. Dabei werden Entscheidungen unter Sicherheit getroffen, da die eintretenden Situationen bekannt sind. Es gibt bei den untersuchten Problemstellungen z.B. keine zufälligen Ereignisse, die den Personalbedarf beeinflussen und auch keine Krankheitsquote oder sonstige ungeplante Abwesenheiten. Zusätzlich wird am Rande die Systemtheorie eingesetzt, die bei den Problemstellungen zur Beschreibung eines formalen Systems dient.

In der Wirtschaftsinformatik existiert eine Vielzahl an Forschungsmethoden [418]. Besonders empirische Untersuchungen finden in dieser Arbeit Anwendung, wobei diverse statistische Tests durchgeführt werden. Zur Durchführung der Experimente werden Prototypen entwickelt und getestet. Somit können Lösungsverfahren an die individuellen Anforderungen der Problemstellungen angepasst und diverse Konzepte erprobt werden. Soweit möglich, werden auf Basis der Experimente auch induktive Schlüsse gezogen. Es kommen zudem deduktive Analysen zum Einsatz, um zum einen geeignete Lösungsverfahren zu identifizieren, aber auch die Wahl der bestmöglichen Parameterabstimmung sowie die Modellierung der Problemstellung zu unterstützen. Hierzu wird die relevante Literatur eingehend studiert.

Der Forschungsgegenstand dieser Arbeit bezieht sich wie bereits erläutert zum einen auf

Problemstellungen des hochflexiblen WFM und zum anderen auf Lösungsverfahren für derartige Problemstellungen. Diese sind klassische OR-Verfahren, Metaheuristiken und Multiagentensysteme.

Insgesamt wird ein sehr facettenreiches Spektrum an Lösungsverfahren und Problemstellungen untersucht. Dies ist erforderlich, um die folgenden Forschungsfragen zu beantworten. Das Forschungsziel ist folglich die Beantwortung der Forschungsfragen. Diese stellen zugleich die Forschungslücken dar, die mit dieser Arbeit geschlossen werden sollen.

- Für Personaleinsatzplanungsprobleme existiert eine Vielzahl an Lösungsverfahren. Problemstellungen des hochflexiblen WFM haben im Vergleich zur klassischen Personaleinsatzplanung bzw. Dienstplanung aber erheblich mehr sowie kleinere Zeitintervalle und einen sehr viel variableren Personalbedarf. Zudem müssen zeitgleich z.T. auch Arbeitszeitmodelle und Touren erstellt werden, was die Probleme besonders komplex macht. Mit dieser Arbeit soll vor allem die Frage geklärt werden:

„Eignen sich eher klassische OR-Verfahren, Metaheuristiken oder Multiagentensysteme für Problemstellungen des hochflexiblen WFM?“

Die Beantwortung dieser Frage soll beispielhaft an drei Problemstellungen erfolgen. Zur Beurteilung der Verfahren werden die Kriterien Exaktheit, Rechenzeit und Aufwand herangezogen, welche in Kapitel 5.4 erläutert werden (Beantwortung in Kapitel 6).

- Aus der zuvor genannten Forschungsfrage ergibt sich eine weitere Frage. Nach Puppe u.a. [315] eignen sich zentrale Lösungsverfahren weniger gut für die Dienstplanung – besser schneiden Multiagentensysteme (dezentrale Ansätze) ab. In dieser Arbeit soll geprüft werden: Gilt die Aussage von Puppe u.a. ebenfalls für die hier untersuchten Problemstellungen des hochflexiblen WFM (Beantwortung in Kapitel 6)?
- Einige der erprobten Lösungsverfahren wurden ursprünglich für kontinuierliche Problemstellungen entwickelt. Es soll daher geklärt werden: Wie lassen sich diese Lösungsverfahren bestmöglich an die für diese Arbeit beispielhaft ausgewählten kombinatorischen Problemstellungen des hochflexiblen WFM anpassen (Beantwortung in Kapitel 4 – jeweils die Unterkapitel „Anpassung an hochflexibles Workforce Management“)?
- Alle selbst implementierten Lösungsverfahren werden eingehend untersucht und hinsichtlich ihrer Parametereinstellungen auf die jeweilige Problemstellung angepasst. Aufgrund der dabei gewonnen Erkenntnisse soll die Frage geklärt werden: Welche Parametereinstellungen sind bei welchen der hier untersuchten Problemstellungen des hochflexiblen WFM zu wählen (Beantwortung in Kapitel 6)?

- Alle Lösungsverfahren werden in dieser Arbeit an realen Problemstellungen erprobt, die direkt aus der Praxis stammen. Sie sind typisch für viele Branchen in Deutschland. Es soll geklärt werden: Wie gut eignen sich die getesteten Verfahren für die hier beispielhaft untersuchten Problemstellungen des hochflexiblen WFM aus der Praxis (Beantwortung in Kapitel 6)?
- Mit den untersuchten Problemstellungen werden recht neue Wege im WFM beschrieben. So sind die Planung einer Vielzahl untertägiger Arbeitsplatzwechsel, die automatische Arbeitszeitmodellerstellung sowie die kombinierte Personaleinsatz- und Tourenplanung heute bereits im praktischen Einsatz – gehören aber nicht zu den Standardinstrumenten zur Steigerung der Agilität im WFM. Hier dominieren noch flexible Arbeitszeiten sowie der Springereinsatz oder der Einsatz geringfügig Beschäftigter. Soweit die Datenlage es zulässt, soll daher in dieser Arbeit geklärt werden: Welche Effekte lassen sich bei den ausgewählten Problemstellungen mit untertägigen Arbeitsplatzwechseln, automatisch erstellten Arbeitszeitmodellen sowie mit der kombinierten Personaleinsatz- und Tourenplanung realisieren (Beantwortung in Kapitel 6)?

1.3 Aufbau der Arbeit

Kapitel 2 widmet sich wesentlichen Begrifflichkeiten und grundlegenden Zusammenhängen im Themenfeld des WFM. Dabei wird in Kapitel 2.1 auch auf hochflexibles WFM eingegangen – der Kern der Arbeit hinsichtlich der zu untersuchenden Problemstellungen. Nachdem in Kapitel 2.2 die betriebswirtschaftliche Relevanz von WFM im Fokus steht, wird anschließend auf die Phasen des WFM eingegangen. Dort wird auch die Zusammenlegung einzelner Phasen erörtert. Mit Ausführungen zu typischen Rahmenbedingungen und anschließend zu häufig genutzten Planungstools, die den Einsatz moderner Softwarelösungen nahe legen, endet Kapitel 2.

Nachdem die Grundlagen gelegt sind, werden in Kapitel 3 die zu untersuchenden Problemstellungen beschrieben. Einleitend erfolgt die Vorstellung einer Klassifikation, in die die Problemstellungen später eingeordnet werden. In den Unterkapiteln 3.2 bis 3.4 wird anschließend auf die drei Problemstellungen eingegangen. Dabei handelt es sich um die untertägige Personaleinsatzplanung in der Logistik, um die automatische Arbeitszeitmodellerstellung im Handel sowie um die kombinierte Personaleinsatz- und Tourenplanung bei Technikern im Außendienst der British Telecom. Jedes der Unterkapitel 3.2 bis 3.4 hat den gleichen Aufbau. Sie beginnen mit einer Einleitung, in der auf individuelle Besonderheiten des jeweiligen Planungsszenarios eingegangen wird. Anschließend wird die reale Problemstellung beschrieben und das mathematische Modell vorgestellt. Zu jeder Problemstellung existieren mehrere Varianten, um die Lösungsverfahren möglichst umfassend zu erproben. Diese Varianten werden ebenfalls erläutert. Nachdem die Pro-

blemstellung anschließend entsprechend der in Kapitel 3.1 vorgestellten Klassifikation eingeordnet ist, werden ähnliche Problemstellungen in der Literatur erläutert.

In Kapitel 4 wird umfassend auf die Lösungsverfahren eingegangen, die in dieser Arbeit Anwendung finden. Für die kombinierte Personaleinsatz- und Tourenplanung bei Technikern im Außendienst standen bereits einige Ergebnisse anderer Autoren zur Verfügung. Alle anderen Verfahren wurden entweder selbst oder im Rahmen von Diplomarbeiten am Fachgebiet Wirtschaftsinformatik für Dienstleistungen der TU Ilmenau (Tabu Search [178] und Teile von Evolutionsstrategien [35]) erstellt. Kapitel 4 beginnt zunächst mit Ausführungen zur geeigneten Abbildung der drei in Kapitel 3 vorgestellten Problemstellungen. In diesem Zusammenhang werden verschiedenen Varianten diskutiert. Anschließend wird in Kapitel 4.2 auf die Komplexität aller drei Problemstellungen eingegangen, wobei die Untauglichkeit einer vollständigen Enumeration hergeleitet wird. In Kapitel 4.3 wird das Abbruchkriterium der Problemstellungen dargestellt. Bevor einzelne Lösungsverfahren beschrieben werden, wird in Kapitel 4.4 allgemein auf typische Verfahren eingegangen. Anschließend werden in den Kapiteln 4.5 bis 4.16 Lösungsverfahren vorgestellt. Jedes Verfahren erhält ein eigenes Unterkapitel. Dabei handelt es sich um Ganzzahlige Optimierung, Konstruierende Verfahren (KV), Constraint Logic Programming (CLP), Hill Climbing (HC), Fast Local Search (FLS) und Guided Local Search (GLS), Tabu Search (TS), Simulated Annealing (SA), Evolutionsstrategien (ES), Genetische Algorithmen (GA), Particle Swarm Optimization (PSO), Multiagentensysteme (MAS) sowie Hybride Lösungsverfahren. Die Unterkapitel der einzelnen Lösungsverfahren sind meist gleich aufgebaut. Nach Ausführungen zum grundlegenden Ablauf wird deren Anwendung in der Literatur erläutert. Anschließend wird ausführlich auf die Anpassung des Verfahrens an die Problemstellungen dieser Arbeit eingegangen, bei denen Experimente durchgeführt werden. Nicht alle Verfahren werden bei allen drei Problemstellungen erprobt. Dies gilt vornehmlich für die Lösungsverfahren, die von anderen Autoren am Problem des Technikereinsatzes im Außendienst implementiert wurden, da in den Publikationen implementierungsrelevante Details fehlen. In Anbetracht der Gesamtzahl an Lösungsverfahren und der Anzahl der bereits selbst implementierten Verfahren hätten weitere Implementierungen auch den Rahmen dieser Arbeit gesprengt. Ein Erkenntnisverlust ist hierdurch jedoch nicht zu befürchten, da z.B. ein eigener Ansatz für ein Multiagentensystem sowie PSO, ES und HC bei allen drei Problemstellungen untersucht wurden. Zudem stehen Ergebnisse weiterer relevanter Verfahren, die als Benchmark dienen, für jede Problemstellung zur Verfügung.

Mit Kapitel 5 wird auf die Themen rund um die Implementierung sowie Durchführung der Experimente und Ergebnisinterpretation eingegangen. Fast alle Verfahren dieser Arbeit wurden in einem selbst entwickelten Framework (ARROW) implementiert. Dieses wird in Kapitel 5.1 kurz vorgestellt. Die Ausnahme bilden klassische OR-Verfahren, für die GAMS herangezogen wird, sowie die Verfahren anderer Autoren. Beim Einsatz von ARROW erfolgt das Speichern und Einlesen von Problemstellungen mit Hilfe eines selbst entwickelten XML-Formats. Dieses erleichtert das Arbeiten mit einer Vielzahl an Problemstellungen erheblich. Auf das Format wird in Kapitel 5.2 eingegangen. In dieser

Arbeit sind zahlreiche statistische Auswertungen erforderlich. Kapitel 5.3 widmet sich eingehend diesem Thema. Kapitel 5 endet mit Ausführungen zu Beurteilungskriterien für Lösungsverfahren, anhand derer die Bewertung vorgenommen wird.

Die Vorstellung der Ergebnisse und deren Interpretation erfolgt in Kapitel 6. Die ersten drei Unterkapitel stehen jeweils für eine der drei Problemstellungen. Diese enthalten die Ausführungen zu den einzelnen Lösungsverfahren und eine zusammenfassende Betrachtung über alle Verfahren an der konkreten Problemstellung. In Kapitel 6.4 wird dann der Vergleich der Lösungsverfahren über alle Probleme angestellt.

Diese Arbeit endet mit einer Zusammenfassung, in der die wesentlichen Erkenntnisse komprimiert dargestellt sind. Zwar werden zahlreiche Fragen geklärt. Es ergeben sich jedoch Neue, die ebenfalls in Kapitel 7.2 vorgestellt werden. Neben der Zusammenfassung ist am Ende der Arbeit auch ein Anhang zu finden, der u.a. alle Ergebnisse in tabellarischer Form enthält.

Kapitel 2

Begrifflichkeiten und Zusammenhänge

In diesem Kapitel werden die Grundlagen gelegt, die für das weitere Verständnis erforderlich sind. Zunächst wird auf relevante Begrifflichkeiten eingegangen, wobei auch hochflexibles WFM – das Anwendungsfeld dieser Arbeit – erläutert wird. Die Nutzenaspekte des WFM werden in Kapitel 2.2 dargestellt. Um eine genauere Vorstellung von den Abläufen des WFM zu erhalten, erfolgt in Kapitel 2.3 die Erläuterung des zugehörigen Phasenkonzepts. Hier wird ebenfalls auf die Möglichkeiten eingegangen, einzelne Phasen zusammenzulegen. So z.B. die automatische Arbeitszeitmodellerstellung, die als praktische Problemstellung in dieser Arbeit untersucht wird. Anschließend werden in Kapitel 2.4 die Rahmenbedingungen erläutert, die im Planungsprozess berücksichtigt werden müssen. Die Vielzahl und die Komplexität der Rahmenbedingungen machen klar, dass besondere Anforderungen an Tools für das WFM gestellt werden müssen. In Kapitel 2.5 wird auf dieses Thema abschließend eingegangen.

2.1 Begrifflichkeiten

2.1.1 Personalplanung

Personalplanung ist Teil der Unternehmensplanung und richtet sich nach der Unternehmensstrategie. „In der Personalplanung geht es darum, das erforderliche Personal am richtigen Ort [zum richtigen] Zeitpunkt zur Verfügung zu stellen.“ [172, S. 6] Demnach beschäftigt sie sich mit „Entscheidungen über die zukünftige Verwendung von Personal in der Unternehmung sowie der Vorbereitung und Kontrolle dieser Entscheidungen“ [104, S. 1759].

Die Aktionsfelder der Personalplanung werden in der Literatur nicht immer einheitlich

gesehen. Die folgenden Aufgaben werden jedoch häufig genannt [104], [292]:

- Personalbedarfsplanung,
- Personalbeschaffungsplanung,
- Personaleinsatzplanung,
- Personalentwicklungsplanung,
- Personalkostenplanung,
- Personalabbauplanung.

Bei Horsch [192] sind die Personalbestands- und Personalkostenplanung nicht enthalten. Drum [105, S. 197–198] sieht zudem ein Aktionsfeld der Personalplanung in der Sicherung ihrer Akzeptanz in Unternehmen. Er fügt bei einigen Aktionsfeldern auch einen qualitativen und quantitativen Aspekt ein. So ist für die Personalbeschaffungsplanung nicht nur die Anzahl der Vollzeitäquivalente maßgeblich, sondern auch die erforderlichen Qualifikationen.

Die zuvor aufgeführten Elemente der Personalplanung können nicht isoliert voneinander betrachtet werden, da Wechselwirkungen zwischen ihnen bestehen. Jedes Aktionsfeld wirkt sich beispielsweise auf die Personalkosten aus und die Personalbedarfsplanung hat einen Einfluss auf die Personalbeschaffungs- und -freisetzungsplanung.

2.1.2 Personaleinsatzplanung

Wie zuvor dargestellt, ist die Personaleinsatzplanung Bestandteil der Personalplanung. Mag sieht sie sogar als Kernstück der Personalplanung [254]. Horsch versteht unter Personaleinsatzplanung den „im Unternehmen vorhandenen Mitarbeitern konkrete Arbeitsaufgaben und Stellen zuzuweisen“ [192, S. 212]. Sie ist demnach die gedankliche Vorwegnahme der zukünftigen Einordnung des verfügbaren Personals in den Prozess der Leistungserbringung. Es geht darum, in welchem Umfang (quantitative Dimension), zu welcher Zeit (temporale Dimension), an welchem Ort (lokale Dimension) Arbeitskräfte mit gewissen Fähigkeiten (qualitative Dimension) bestimmte Stellen oder Tätigkeiten übernehmen. Aus dieser Definition heraus ergeben sich viele Überschneidungen mit dem Begriff der Personalplanung. In der Praxis bedient man sich häufig der „5 R“ [14, S. 6]:

- die *richtig* qualifizierten Mitarbeiter,
- in der *richtigen* Anzahl,

- zur richtigen Zeit,
- am richtigen Ort,
- zu den richtigen (optimalen) Kosten.

Personaleinsatzplanung ist bzgl. ihres Planungshorizontes kurz-, mittel- oder langfristig angelegt. So kann der Planungshorizont im Handel wenige Tage betragen – bei kurzfristigen Ereignissen auch wenige Stunden. In Kliniken, wo nicht von Personaleinsatzplanung, sondern von Dienstplanung gesprochen wird, wird meist ein kompletter Monat geplant. Und in der Papierindustrie sind oft ganzjährige Einsatzpläne mit rollierenden Schichtzyklen zu finden. Die Grenzen zwischen den Planungshorizonten sind fließend. Ebenso sind die Zeitintervalle, in denen geplant wird, häufig sehr verschieden. Das Spektrum reicht von 15-Minuten-Intervallen über die tageweise Einplanung bis hin zur Planung von Wochen- oder Monatsblöcken [106, S. 125], [170, S. 30].

Bei der Personaleinsatzplanung müssen zahlreiche Rahmenbedingungen beachtet werden. Z.B. Arbeitsverträge, Personalbedarfe, Verfügbarkeiten, Qualifikationen, etc. Auf einige dieser Rahmenbedingungen wird später in Kapitel 2.4 eingegangen.

2.1.3 Hochflexibles Workforce Management

Der Begriff WFM wird in seiner Bedeutung vielfältig gebraucht. Oft wird er synonym mit dem deutschen Begriff Personaleinsatzplanung verwendet. Zwar umfasst WFM in seinem Kern vor allem die Personaleinsatzplanung. Aber auch Personalbedarfsermittlung, Arbeitszeitmodellgestaltung und Arbeitszeitmanagement sind integraler Bestandteil – mit einem besonderen Fokus auf den bedarfsorientierten Personaleinsatz. Einige Autoren wie Feldmann sehen sogar die Zutrittskontrolle als Bestandteil von WFM [119, S. 14].

Wie gut es gelingt, die nachfolgend in Kapitel 2.2 dargestellten Ziele des WFM zu erreichen, hängt ganz entscheidend von der WFM-Agilität im Unternehmen ab. Also der Fähigkeit, den Personaleinsatz an sich ändernde Rahmenbedingungen anzupassen. Z.B. wenn das Arbeitsaufkommen für einen längeren Zeitraum um 30% sinkt, der Personalbedarf sehr variabel ist und untertägige Auftragsspitzen kostengünstig abgedeckt werden müssen. Hierfür steht Unternehmen ein ganzes Bündel an praxistauglichen Maßnahmen zur Verfügung. Eine größere Granularität in der Planung, flexible Arbeitszeiten, die Abkehr von starren Schichten, die Zusammenlegung von Planungsphasen des WFM (siehe Kapitel 2.3), Lebensarbeitszeitkonten, Arbeit auf Abruf, der Einsatz von Leiharbeitskräften und die Schaffung einer homogenen Qualifikationsstruktur sind nur einige Maßnahmen. Für vertiefende Ausführungen zum Einsatz von Flexibilisierungsinstrumenten zur Steigerung der WFM-Agilität sei auf die Arbeiten von Günther [149], [150] verwiesen.

Mit hochflexiblem WFM ist demnach der effiziente, bedarfsorientierte Personaleinsatz gemeint, der die in Kapitel 2.3 dargestellten Phasen berücksichtigt und bei dem Instrumente zur Steigerung der WFM-Agilität zum Einsatz kommen. „Hochflexibel“ meint in dieser Arbeit keine kurzfristigen Planänderungen (kein Rescheduling), sondern die Fähigkeit, Mitarbeiter sehr bedarfsorientiert in kleinen Planungsintervallen auf verschiedene Arbeitsplätze bzw. Aufträge zu setzen und in diesem Zusammenhang ggf. auch die Arbeitszeiten der Mitarbeiter passgenau an den Personalbedarf anzupassen. Derartige Planungsprobleme weichen stark von der klassischen Personaleinsatzplanung oder Dienstplanung mit starren Schichten ab. Sie sind durch sehr kleine und sehr zahlreiche Zeitintervalle (z.B. Planung in 15-Minuten-Intervallen), untertägige Arbeitsplatzwechsel, durch eine Kombination der in Kapitel 2.3 dargestellten Phasen des WFM (z.B. automatische Arbeitszeitmodellerstellung) oder durch die kombinierte Touren- und Personaleinsatzplanung gekennzeichnet.

2.2 Nutzen durch Workforce Management

Dass es im WFM enorme Potenziale gibt, zeigt die Studie von Proudfoot Consulting [314] (die Studie existiert bereits in einer neueren Auflage, wobei dort die unproduktive Arbeitszeit nicht mehr so detailliert herausgearbeitet ist). Mitarbeiter verbringen im Durchschnitt mehr als ein Drittel ihrer Arbeitszeit unproduktiv. Es werden also Kosten verursacht, ohne einen wertschöpfenden Beitrag zu leisten. Als Hauptursache konnte mangelnde Planung und Steuerung identifiziert werden, was in Abbildung 2.1 veranschaulicht wird.

Die Auswirkung eines unangepassten Personaleinsatzes an den Personalbedarf sind in Abbildung 2.2 dargestellt. Deutlich sind Phasen der Über- und Unterdeckung im Personaleinsatz zu erkennen. Eine Überdeckung bedeutet unnötige Personalkosten. Mitarbeiter sind anwesend, obwohl sie nicht benötigt werden. Sie füllen diese Zeit dann u.a. mit Alibi-Tätigkeiten und leisten keinen wertschöpfenden Beitrag. Auf der anderen Seite bewirkt eine Unterdeckung, dass der Servicelevel sinkt, Überstunden anfallen oder Aufträge womöglich nicht angenommen werden können. Es gilt also, den Personaleinsatz und den Personalbedarf anzugleichen.

Der effiziente Personaleinsatz darf heute nicht mehr nur eine Aufgabe einzelner Personen im Unternehmen sein, die völlig autark und ohne Berücksichtigung unternehmensweiter Gegebenheiten und Prozesse Mitarbeiter verplanen. Um heute wettbewerbsfähig zu sein, muss das WFM seine Rolle als zentrales Steuerungsinstrument im Unternehmen einnehmen. Nur so gelingt es, die wertvollste Ressource im Unternehmen – den Menschen im Arbeitsprozess – optimal einzusetzen. Mit WFM sollen folgende Ziele erreicht werden [336, S. 238–241], [338, S. 59–65]:

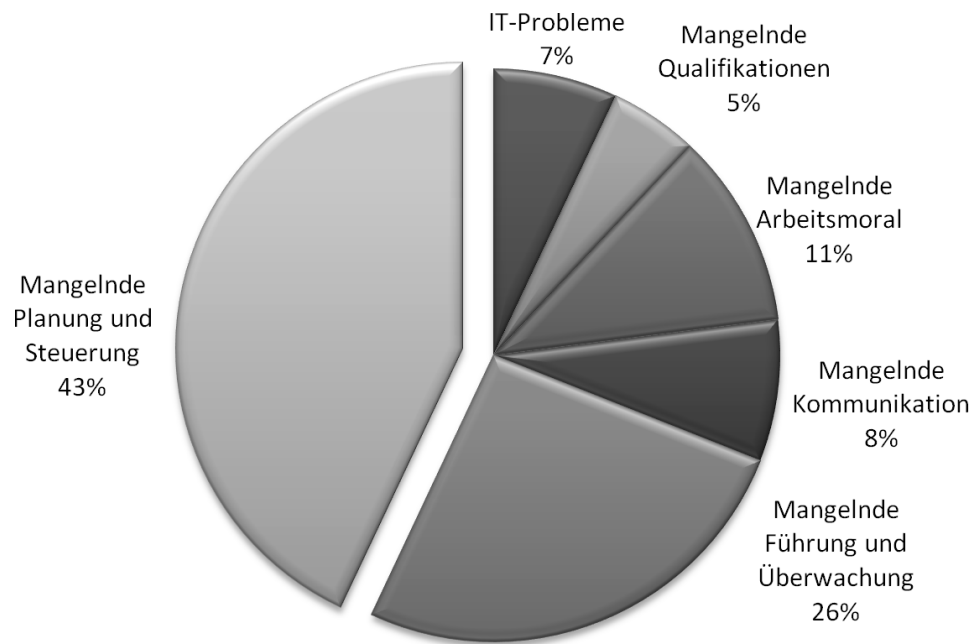


Abbildung 2.1: Ursachen für Unproduktivität [314]

- Kostensenkung,
- Umsatzsteigerung,
- Erhöhung des Servicelevels,
- mehr Transparenz im Unternehmen,
- schnellere Reaktion auf Bedarfsschwankungen,
- weniger Leerzeiten und Überstunden,
- reduzierter administrativer Aufwand,
- gezielter Einsatz von Leiharbeitskräften,
- effektive Nutzung flexibler Arbeitszeitmodelle,
- passgenauer Einsatz von Arbeit auf Abruf,
- Steigerung der Mitarbeitermotivation,
- Reduzierung gefahrener Kilometer,

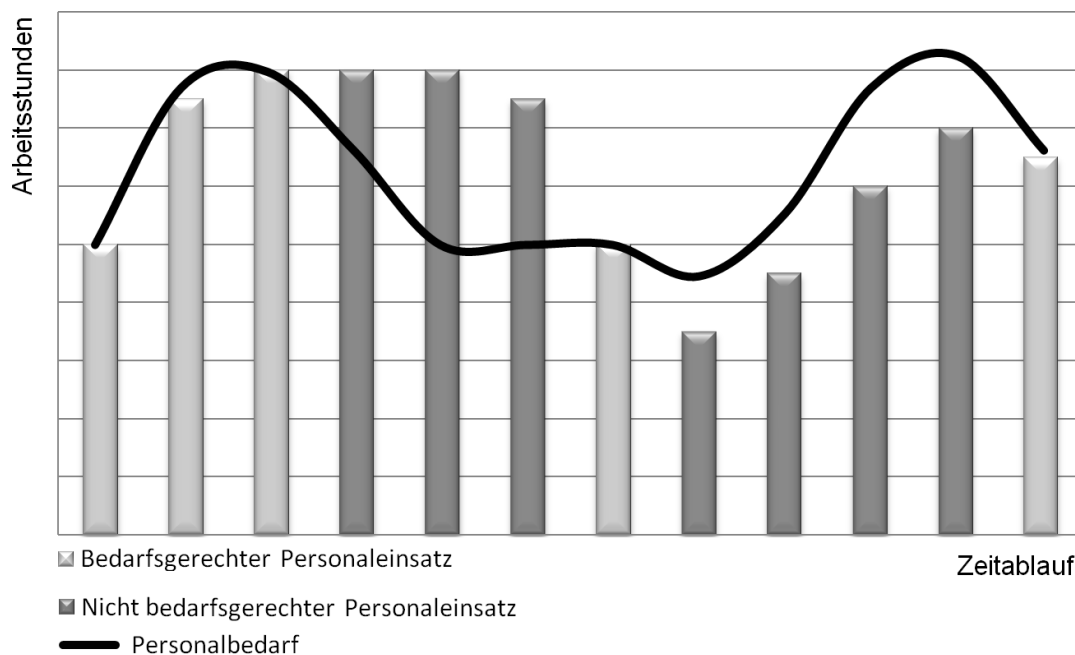


Abbildung 2.2: Über- und Unterdeckung im Personaleinsatz [337, S. 19]

- geringe Abnutzung der Betriebsmittel.

Die Priorität der Ziele und ihr Nutzenpotenzial sind dabei von Unternehmen zu Unternehmen verschieden. Sie hängen von der Unternehmensgröße, -organisation und -kultur sowie vom Geschäftsmodell und der Ist-Situation ab. Aber auch die Fähigkeit des Unternehmens, Veränderungsprozesse erfolgreich umzusetzen und Instrumente zur Steigerung der WFM-Agilität einzuführen, sind für den Grad der Zielerreichung ausschlaggebend. Einige konkrete Nutzenbeispiele durch die Einführung von WFM-Systemen sind nachfolgend aufgeführt [147, S. 137]:

- Thalia Bücher AG: Innerhalb von sieben Monaten wurden Überstunden um 75% reduziert.
- EDEKA Minden-Hannover Holding GmbH: Einsparung von 200.000€ pro Jahr durch die Fehlerreduzierung im Arbeitszeitmanagement bei 2.500 Mitarbeitern. Zudem eine Reduzierung von etwa 12.000 Stunden an administrativer Tätigkeit pro Jahr.
- Max Bahr Holzhandlung GmbH & Co. KG: Reduktion der Mehrarbeit von 50.000 auf 1.000 Stunden innerhalb eines Jahres.
- Telefónica O2 Germany GmbH & Co. OHG: Reduzierung der Überstunden innerhalb von 18 Monaten um 50% in den Shops.

2.3 Planungsprozess im Workforce Management

WFM ist ein zentraler Bestandteil unternehmerischen Handelns und ist eng mit vielen Prozessen im Unternehmen verwoben. Zahlreiche Softwareanbieter verfolgen den idealisierten Ansatz in Abbildung 2.3 [337, S. 20–29], [420, S. 344–353], [150, Kap. 14.6]. Dieses Vorgehen ist durchaus praktikabel und nutzenstiftend. Die Vielzahl der Unternehmen aller Branchen, die derartige Systeme erfolgreich einsetzen, sind der beste Beleg. Es muss jedoch gesagt werden, dass die Zusammenlegung einiger Phasen weitere Nutzenpotenziale birgt. Nachfolgend werden die einzelnen Phasen erläutert, wobei in Kapitel 2.3.5 auf die Zusammenlegung von Phasen eingegangen wird.



Abbildung 2.3: Idealisierter Planungsprozess im Workforce Management

2.3.1 Phase 1: Personalbedarfsermittlung

In dieser Phase wird ermittelt, wie viele Mitarbeiter wann mit welcher Qualifikation auf welchem Arbeitsplatz bzw. in welcher Funktion benötigt werden. Für eine effiziente Personaleinsatzplanung ist es notwendig, den Personalbedarf möglichst exakt zu bestimmen. Allein Fehler in dieser Phase können nach einer Studie von Miebach Consulting im Durchschnitt bis zu 15% höhere Personalkosten ausmachen [265, S. 4]. In der Praxis werden je nach Planungshorizont verschiedene Verfahren eingesetzt, wobei in der kurzfristigen Bedarfsermittlung die einfache Schätzung dominiert [265, S. 14–15].

Die Bedarfsermittlung lediglich aufgrund von Erfahrungswerten kann nicht immer optimal sein. Dem Planer mangelt es aber meist auch an leistungsfähigen Tools. Moderne Lösungen ermöglichen es z.B. in der Logistik, im Handel oder im Call Center das zu erwartende Arbeitsvolumen aufgrund von Vergangenheitsdaten zu prognostizieren. Dabei werden Werbeaktionen, kurzfristige Auftragsänderungen, Trends und sogar das Wetter berücksichtigt. Eine andere Form der Bedarfsermittlung ist ereignisorientiert. Ist im Vorfeld bekannt, wann z.B. Kleintransporter, LKW, Flugzeuge oder Züge eines bestimmten Typs eintreffen oder wieder abfahren, kann der Personalbedarf ermittelt werden. Ein LKW (3,5 t ohne Anhänger), der übermorgen um 13 Uhr an Laderampe drei erwartet wird, verursacht einen Personalbedarf von zwei Personen von 13–14 Uhr zur Entladung an Laderampe drei sowie eine Person von 12.50–13.10 Uhr an der Pforte (bei mehreren LKW jedoch nur max. zwei Personen an der Pforte). Durch das Einlasten verschiedener Ereignisse ergibt sich sehr exakt der Personalbedarf. Je nach Planungsbereich kann der

Personalbedarfsermittlung eine weitere Phase vorgeschaltet sein. Im Speditionsgeschäft ist dies häufig die Tourenplanung. In der Intralogistik z.B. ein PPS-System.

2.3.2 Phase 2: Arbeitszeitmodellerstellung

Ausgehend vom Personalbedarf werden Arbeitszeitmodelle erstellt, die den Personalbedarf möglichst gut abdecken. Gesetzliche und tarifliche Regelungen müssen Berücksichtigung finden. Der Aufwand für die Erstellung ist meist recht groß. Z.T. müssen sie auch vom Betriebsrat genehmigt werden. Gewöhnlich werden Arbeitszeitmodelle daher nur selten geändert und für spätere Planungen weiterverwendet. Ein absolutes Negativbeispiel für eine sehr unflexible Planung bei starken Bedarfsschwankungen wäre die Verwendung von lediglich einem Arbeitszeitmodell (Arbeitsbeginn: 7 Uhr, Arbeitsende: 15 Uhr, feste Pause: 30 Minuten, Gleitzeit: keine). Arbeiten alle Mitarbeiter nach diesem Modell, können untertägige, wöchentliche oder auch saisonale Schwankungen im Personalbedarf nicht durch den angepassten Personaleinsatz ausgeglichen werden. Unnötige Kosten durch Leerzeiten und Überstunden sowie ein schlechter Service und eine geringe Mitarbeitermotivation wären nur einige der Folgen.

2.3.3 Phase 3: Personaleinsatzplanung

Hier findet die eigentliche Planerstellung statt. Wie in Kapitel 2.1.2 dargestellt wurde, wird in dieser Phase bestimmt, welcher Mitarbeiter wann welchen Arbeitsplatz besetzen bzw. welchen Auftrag erledigen soll. Der Planungshorizont kann dabei unterschiedlich groß sein. Auch der Zeitpunkt, zu dem ein neuer Plan für die Mitarbeiter einsehbar ist, kann variieren. Bei der Planung werden diverse Rahmenbedingungen berücksichtigt. Qualifikationen, Mitarbeiterwünsche, Abwesenheiten, Saldenstände, gesetzliche/tarifliche Regelungen, etc. fließen in die Planung ein.

Leistungsstarke Planungstools ermöglichen eine vollautomatische Planvorschlagsgenerierung. Sogar die Erstellung verschiedener Planungsszenarien ist möglich. Ein kostenoptimaler Plan kann einer Planungsvariante mit maximalem Servicegrad oder einem Plan mit reduzierter Maschinenlaufzeit gegenübergestellt werden. Die Qualität dieser Planungsanschlüsse hängt neben der funktionalen Tiefe und der individuellen Parametrierung des Systems ganz entscheidend von den verwendeten Planungsalgorithmen ab.

2.3.4 Phase 4: Arbeitszeitmanagement

Arbeitszeitmanagement ist mehr als die Kommen- und Gezeiten der Mitarbeiter aufzuzeichnen. In dieser Phase werden z.B. die Saldenstände der Mitarbeiter aufgrund ihrer Arbeitszeiten bestimmt. Abwesenheiten werden geplant und überwacht. Zudem werden alle Daten ermittelt, die für die Abrechnung an das Lohn- und Gehaltssystem übergeben werden oder für Auswertungen erforderlich sind. Hierfür ist es unabdingbar, dass auch alle zu berücksichtigenden gesetzlichen/tariflichen Regelungen im System hinterlegt sind. Andernfalls leidet die Datenqualität und ein manueller Mehraufwand ist unausweichlich.

2.3.5 Zusammenlegung von Phasen

Grundsätzlich laufen die vier dargestellten Phasen meist nacheinander ab, wobei Phase 4 auch vor bzw. parallel zu anderen Phasen angeordnet werden kann. Man findet hier durchaus verschiedene Ansätze. Denn die Planung hat einen Einfluss auf das Arbeitszeitmanagement: z.B. durch garantierte Zuschläge – auch bei nachträglicher Umplanung, automatischer Schichterkennung bis hin zur Steuerung der Zutrittskontrolle je nach geplanter Arbeitszeit. Des Weiteren erhält der Planer über die hinterlegten Regelungen in der Zeitwirtschaft einen Überblick über die Auswirkungen seines Handelns direkt bei der Planerstellung. Aber auch die Daten des Arbeitszeitmanagements fließen in die Planung ein, wenn Abwesenheiten und Zeitsalden berücksichtigt werden. Allein die Berücksichtigung der Urlaubsplanung würde ein paralleles Vorgehen erfordern, da durch die Gewährung von Urlaub die Personalkapazität sinkt, ggf. veränderte Arbeitszeitmodelle erforderlich werden und sogar der Personalbedarf beeinflusst wird (Ablehnung/Verschiebung von Aufträgen durch die reduzierte Personalkapazität).

In Kapitel 2.3.3 wurde erläutert, dass Mitarbeitern im Rahmen der Personaleinsatzplanung zuvor erstellte Arbeitszeitmodelle zugewiesen werden. Hier gibt es Ausnahmen. Z.B. im Handel, wo man Arbeitszeitmodelle bei jeder Planung z.T. automatisch erstellt. Phase 2 und 3 laufen dort in einem Schritt ab. Es werden lediglich Regeln für die Erstellung der Arbeitszeitmodelle vorgegeben. So müssen meist minimale und maximale Längen der Arbeitszeitmodelle eingehalten werden. Diese Integration oder auch eine Vielzahl an verfügbaren Arbeitszeitmodellen erhöhen die Flexibilität der Einsatzplanung enorm. Natürlich mit den positiven Auswirkungen auf viele der genannten Ziele eines hochflexiblen WFM. Ein konkretes Praxisbeispiel wird in Kapitel 3.3 erläutert.

Eine ganzheitliche Berücksichtigung von Personalbedarfsermittlung (Phase 1) und Einsatzplanung (Phase 3) ist in der Praxis kaum zu finden. Sie birgt jedoch erhebliche Potenziale. Denn der Verlauf des Personalbedarfs hängt z.B. von den geplanten Touren oder vom geplanten Produktionsprogramm ab. Meist wird die Ressource Mensch

in diesem Zusammenhang nicht als begrenzender Faktor gesehen. Getreu der Devise: „Irgendwie wird man das Auftragsvolumen schon bewältigen“. Verkürzt man Tour eins beispielsweise um einen anzufahrenden Kunden und fügt ihn in Tour zwei ein, so mag dies aus Sicht der Tourenplanung nicht optimal sein. Die Verfügbarkeit an qualifiziertem Personal lässt jedoch keine andere Alternative zu – ohne Ruhezeiten zu verletzen oder teure Zuschläge zu bezahlen. An Echtdateien der British Telecom wird eine derartige Problemstellung in dieser Arbeit untersucht (siehe Kapitel 3.4). Eine ähnliche Problemstellung kann man z.T. in Call Centern finden, wo der Personalbedarf nicht losgelöst von der Personaleinsatzplanung betrachtet wird. Denn wenn der Servicelevel aufgrund einer zu geringen Besetzung schlecht ist, rufen einige Kunden in den Folgeperioden erneut an, was wiederum den Personalbedarf beeinflusst. Diese Zusammenhänge lassen sich nur in einem integrierten Planungsvorgehen oder wenigstens in einer zyklischen Planoptimierung lösen.

Neben der Zusammenführung von Phase 2 und 3 sowie von Phase 1 und 3 ist es ebenso denkbar, die ersten drei Phasen in einem Planungsschritt zu lösen. Hierbei würde es sich z.B. um das zuvor erwähnte Call Center-Beispiel handeln, bei dem zusätzlich Arbeitszeitmodelle automatisch erstellt werden.

2.4 Rahmenbedingungen des Workforce Managements

Die Vielzahl der Regelungen, die bei der Personaleinsatzplanung zu berücksichtigen sind, ist schier unüberschaubar. So ist es nicht verwunderlich, dass es in der Praxis oft Probleme bei der Planung und Steuerung von Personal gibt. Planer sind aufgrund der Datenflut regelmäßig überfordert. Zudem sind viele der Rahmenbedingungen unscharf und können sich schnell ändern. Abbildung 2.4 veranschaulicht beispielhaft die Fülle an Informationen, auf die ein Planer achten muss.

Wesentliche Rahmenbedingungen typischer Planungsszenarien werden nachfolgend allgemein beschrieben. Dabei werden auch wichtige Begrifflichkeiten erläutert.

Saldenstände Das Führen von Salden (z.B. Gleitzeit, Überstunden, Freischicht, etc.) ist einerseits für die Entlohnung der Mitarbeiter relevant. Auf der anderen Seite werden Salden im Rahmen der Arbeitszeitflexibilisierung benötigt. Sie unterliegen Regelungen für ihre Berechnung und dürfen meist ein bestimmtes Band nicht über- und unterschreiten. Das Erreichen des Nullwertes eines Saldos innerhalb eines bestimmten Zeitraumes oder zu einem bestimmten Stichtag, wie dem Jahresende, ist ebenfalls sehr häufig anzutreffen. Welchen Regeln Salden unterliegen und wie man im Einzelfall mit ihnen umgeht,

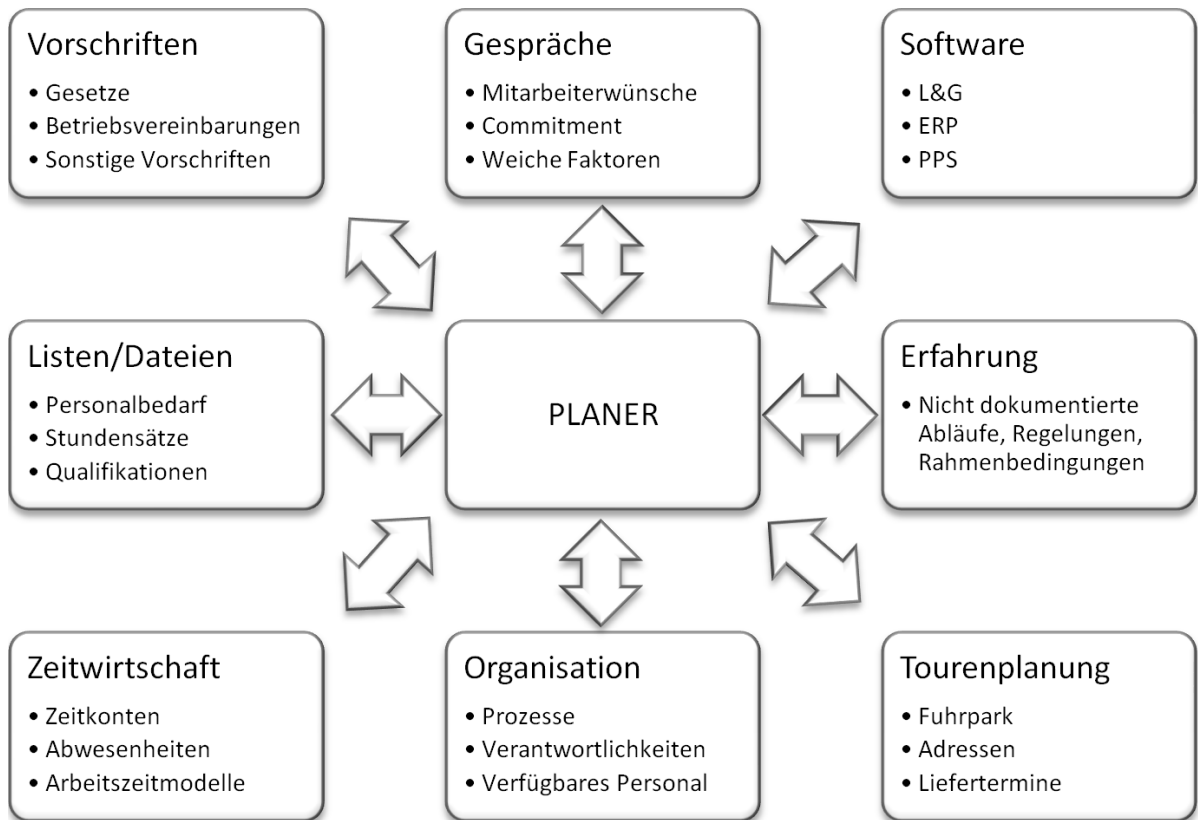


Abbildung 2.4: Rahmenbedingungen für den Planer

ist sehr individuell und wird meist unterschiedlich gehandhabt.

Ziel ist es oft, die Salden aller Mitarbeiter möglichst auf gleichem Niveau zu halten. Es ist also nicht erstrebenswert, dass ein Mitarbeiter z.B. 150 Gleitzeitstunden aufgebaut hat, während sein Kollege -60 Gleitzeitstunden führt. Derartige Konstellationen sind Ausdruck einer Überbelastung einzelner Mitarbeiter und wirken sich demotivierend auf die Belegschaft aus. U.U. fallen auch Kosten durch die Auszahlung von Gleitzeitstunden an. Ggf. gibt es Regelungen, Salden bei bestimmten Grenzen ersatzlos zu kappen. Dann werden z.B. bis zu 20 Überstunden am Monatsende ausbezahlt. Alle übrigen Stunden werden ersatzlos abgeschnitten und nur in besonderen Ausnahmefällen ausbezahlt.

Arbeitszeitmodelle Arbeitszeitmodelle regeln die Arbeitszeiten der Mitarbeiter. Sie definieren Arbeitsbeginn und Arbeitsende sowie ggf. Pausenregelungen, Kernzeiten sowie erlaubte Anwesenheitszeiten. Oft werden die Begriffe Arbeitszeitmodell und Schicht synonym gebraucht. Tatsächlich kann eine Schicht je nach Tagestyp mehrere Arbeitszeitmodelle beinhalten. So kann eine Frühschicht z.B. Montag bis Donnerstag von 5:00 bis 13:00 Uhr gehen, am Freitag jedoch nur von 5:00 bis 12:00 Uhr. Die Frühschicht beinhaltet in diesem Beispiel demnach zwei Arbeitszeitmodelle. In dieser Arbeit treten

derartige Fälle nicht auf, sodass die Begriffe Schicht und Arbeitszeitmodell bzgl. des täglichen Arbeitsbeginns und -endes synonym gebraucht werden.

Geplante Abwesenheiten Mitarbeiter haben geplante Abwesenheiten, die relativ frühzeitig bekannt sind. Zu diesen Abwesenheiten zählt Urlaub, Fortbildung, Gleitzeittag, etc. sowie geplante Kuren und Operationen. Abwesenheiten führen dazu, dass die Verfügbarkeit an Mitarbeitern sinkt. Auch Betriebsmittel unterliegen geplanten Abwesenheiten. So durchlaufen Transportmittel regelmäßig Serviceinspektionen. Diese Serviceinspektionen können ggf. um einen oder mehrere Tage verschoben werden. Ein bereits genehmigter Urlaub ist vom Planer jedoch nicht mehr leicht verschiebbar.

Neben den geplanten Abwesenheiten seien der Vollständigkeit halber auch ungeplante Abwesenheiten erwähnt. Bei ihrem Auftreten kann ein Rescheduling erforderlich sein, welches in dieser Arbeit jedoch keine Anwendung findet, da ungeplante Abwesenheiten bei den drei untersuchten Problemstellungen nicht berücksichtigt werden.

Personalbedarfsdeckung Die passgenaue Deckung des Personalbedarfs ist eine der Hauptaufgaben der Personaleinsatzplanung. Um alle anstehenden Aufgaben optimal bearbeiten zu können, muss der Bedarf an Personal und an Betriebsmitteln (Fahrzeuge, Ausrüstung, Ersatzteile) für die jeweiligen Planungsintervalle gedeckt sein. Eine Über- oder Unterdeckung ist zu vermeiden. Zu viel Personal führt zu unnötigen Kosten und langfristig wirkt dieser Zustand demotivierend. Zu wenig Personal macht Überstunden und Zuschläge erforderlich und zwingt die Mitarbeiter zu einer erhöhten Arbeitsleistung. Zudem können Umsatzpotenziale nicht genutzt werden und die Motivation und Kundenzufriedenheit sinkt. Ebenso negativ wirkt sich eine Über- bzw. Unterdeckung im Einsatz von Fahrzeugen, Ausrüstung, Ersatzteilen, etc. aus.

Teambildung Z.T. fahren Mitarbeiter in Fahrgemeinschaften zur Arbeit oder sie arbeiten in Teams zusammen. Hierzu ist es erforderlich, dass alle Gruppenmitglieder den gleichen Arbeitsbeginn und das gleiche Arbeitsende haben. Ein Einsatzplanungsproblem enthält dann nicht mehr nur die Dimensionen wer, wann, wo, sondern auch mit wem. In diesem Zusammenhang treten auch Stammarbeitsplatzregelungen auf, die für eine Teambildung sorgen, wobei zeitliche Aspekte keine Rolle spielen.

Wünsche der Mitarbeiter Mitarbeiter haben vor allem Wünsche in Bezug auf ihre Arbeitszeiten, Arbeitstage und Arbeitsplätze. Die Berücksichtigung dieser Wünsche trägt zur Mitarbeitermotivation und somit zur Produktivität bei.

Gesetzliche und tarifliche Regelungen Die Fülle und Bandbreite dieser Regelungen erscheint unerschöpflich, sodass an dieser Stelle keine vollständige Aufzählung aller Regelungen erfolgen kann. Es gibt allgemeine Regelungen für minimale und maximale Arbeitszeiten, für Pausen-, Lenk- und Ruhezeiten. Zudem gibt es Regelungen für besondere Personengruppen wie Schüler, Schwangere, Langzeitkranke, geringfügig Beschäftigte und Behinderte. Diese Regeln sind zwingend einzuhalten. Die Praxis zeigt jedoch, dass einige Regelungen auch „flexibel“ ausgelegt werden und man Verstöße bewusst in Kauf nimmt.

Kostengrenzen für Mitarbeiter Einige Mitarbeitergruppen dürfen ein bestimmtes Einkommen im Monat nicht überschreiten. Dies trifft auf 400€-Kräfte zu. Das Prinzip lässt sich jedoch auch auf andere Personengruppen mit anderen Maximaleinkommen übertragen.

Zuschläge Im Rahmen der Zeitwirtschaft wird Arbeitsbeginn und Arbeitsende der Mitarbeiter erfasst. Dabei werden auch besondere Ereignisse, wie Pausen und Fehlgründe dokumentiert. Für die spätere Entlohnung ist es nicht nur wichtig, wie viele Stunden eine Person gearbeitet hat, sondern wann sie gearbeitet hat. So kann an Wochenenden ein höherer Stundenlohn gewährt werden und Nachtarbeit kann zu Zuschlägen führen. Die Zuordnung der Arbeitszeiten zu den entsprechenden Lohnarten erfolgt in der Bezugsartenrechnung. Anschließend werden die Daten an das Lohnsystem übertragen, wo die auszahlenden Geldbeträge ermittelt werden. Nehmen Mitarbeiter nicht an der Zeiterfassung teil, so spricht man von Negativzeiterfassung oder von „Nichtstemplern“. Bei diesen Mitarbeitern werden die geplanten Arbeitszeiten als Ist-Arbeitszeiten angesehen. Lediglich Abweichungen vom Plan werden dokumentiert.

Öffnungszeiten Lager und Geschäfte haben Öffnungszeiten. Es macht also keinen Sinn, einen Einsatz- bzw. Auslieferungsort zu erreichen, wenn dieser noch nicht geöffnet hat oder bereits geschlossen ist. Ebenso hat das Auslieferungslager oder die Servicestation bestimmte Öffnungszeiten, die eingehalten werden müssen.

Qualifikationen Jeder Mitarbeiter verfügt über eine Menge an Qualifikationen. Entweder in der Form: vorhanden/nicht vorhanden oder mit einer Ausprägung. Z.B. kann Mitarbeiter eins die Qualifikation A zu 80 Punkten bzw. Prozent. Dabei kann auch eine Mindestpunktzahl in Bezug auf die Ausprägung der Qualifikationen gefordert sein. Zudem kann der Qualifikationsgrad auch einen Rückschluss auf die Bearbeitungsgeschwindigkeit von Aufträgen zulassen.

Ausstattung der Arbeitsmittel Ähnlich den Mitarbeiterqualifikationen, müssen auch die Arbeitsmittel gewissen Anforderungen genügen. So müssen beispielsweise Fahrzeuge für bestimmte Aufträge mit einer Hebebühne ausgestattet sein.

Transportkapazitäten Wenn es darum geht, eine bestimmte Menge auszuliefern, muss die maximal mögliche Zuladung berücksichtigt werden. Außerdem sind überdimensionierte Transportkapazitäten zu vermeiden, da sie zu erhöhten Betriebskosten führen.

Fahrzeuggeschwindigkeiten Die maximal zulässigen Höchstgeschwindigkeiten unterliegen gesetzlichen Einschränkungen. Zudem sind sie an bestimmte Verhältnisse anzupassen (Nebel, Eis, Straßenschäden). Die Fahrgeschwindigkeit kann durch Staus und zäh fließenden Verkehr beschränkt sein. Die Wahl der Route hat einen Einfluss auf die zurückzulegende Strecke und auch auf die Dauer der Fahrt.

Zeitfenster für Aufträge Das Zeitfenster für die Erfüllung eines Auftrags liegt gewöhnlich innerhalb der Öffnungszeiten. Privatpersonen haben keine Öffnungszeiten, wohl aber Zeiten zu denen sie angetroffen werden wollen/können. Dabei ist zu berücksichtigen, dass neben der Ankunftszeit auch das Bearbeitungsende des Auftrags innerhalb des Zeitfensters liegen muss. Ein nicht Einhalten des Zeitfensters kann dazu führen, dass der Auftrag nicht erfüllt werden kann oder es zu Strafzahlungen kommt. Auf alle Fälle sind derartige Verstöße negativ für den Service und die Kundenzufriedenheit. Zudem muss ggf. ein Folgetermin vereinbart werden, um die Arbeiten abzuschließen, was Mehrkosten nach sich zieht.

2.5 Typische Planungstools in der Praxis

Die Beherrschung der großen Komplexität im WFM ist aufgrund vielfältiger Arbeitszeitmodelle, Mitarbeiterverfügbarkeiten, Bedarfsschwankungen, Qualifikationen sowie der Beachtung von gesetzlichen/tariflichen Regelungen etc. nur durch den Einsatz von speziellen Softwarelösungen wirklich effizient möglich. Trotz der Komplexität und der vielen ungenutzten Potenziale werden Mitarbeiter heute meist noch mit Tabellenkalkulationsprogrammen oder auf „Papier“ verplant. Abbildung 2.5 veranschaulicht dies, wobei Mehrfachnennungen möglich waren.

Den Einsatz einer modernen Software aufgrund einer geringen Mitarbeiteranzahl von vornherein auszuschließen, ist mit Vorsicht zu genießen. Es gibt genügend Beispiele, in denen der Nutzen selbst bei sehr kleinen Planungseinheiten enorm ist [147, S. 137, 150]. Ausschlaggebend ist allein, welche Ziele verfolgt werden und ob man diese mit

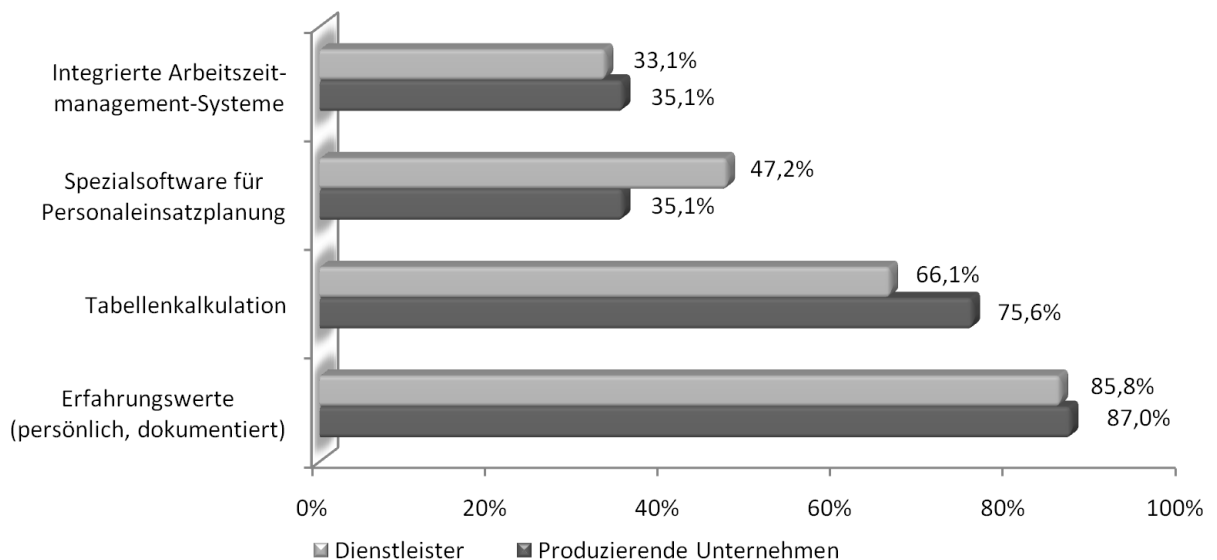


Abbildung 2.5: Nutzung von Tools zur Personaleinsatzplanung [14, S. 48]

einer Tabellenkalkulation erreichen kann oder eine geeignete Software benötigt. Eine Tabellenkalkulation eignet sich vor allem, wenn folgende Punkte eintreten:

- Es gibt wenig Regelungen und Einflussfaktoren bei der Planerstellung.
- Die Anzahl der Arbeitszeitmodelle ist überschaubar.
- Der Personalbedarf schwankt nicht oder nur sehr geringfügig.
- Die Planungsbereiche arbeiten sehr eigenständig und ein Austausch von Mitarbeitern und Informationen erfolgt nur selten oder gar nicht.
- Zugriffe, Änderungen und Einsichten in den Plan sind unkompliziert geregelt.
- Es werden kaum Maßnahmen zur Steigerung der WFM-Agilität eingesetzt.
- Der Nutzen aus einer leistungsstarken Software steht in einem ungünstigen Verhältnis zu den Kosten.
- Es sind einfach keine Kapazität und kein Budget vorhanden, um eine leistungsstarke Software einzuführen.
- Es gibt Personen im Unternehmen, die sich hervorragend mit der Tabellenkalkulation auskennen. MS EXCEL™ kann sehr mächtig sein. Die Fremddateizugriffe, Formeln, etc. müssen jedoch erst einmal in einer Planungsvorlage hinterlegt werden. Auch muss sichergestellt sein, dass auch später das Know-how da ist, um

nachträgliche Anpassungen durchzuführen.

Selbst in großen Unternehmen wird z.B. MS EXCEL™ erfolgreich eingesetzt. Und zwar genau dann, wenn die zuvor genannten Rahmenbedingungen zutreffen. MS EXCEL™ hat jedoch Defizite, die den Einsatz von Spezialsoftware für das WFM erforderlich machen:

- Kein detailliertes Berechtigungskonzept – lediglich Lese-/Schreibberechtigung auf die MS EXCEL™-Dateien. Keine expliziten Berechtigungen auf Fehlgrundeingaben, Mitarbeiter und Arbeitszeitmodelle.
- Umständlicher und z.T. fehlerbehafteter Zugriff auf Daten von Fremdsystemen (An-/Abwesenheiten, Zeitstempelungen, Saldenstände, Personalbedarfe, ...).
- Der Einsatzplan muss manuell erstellt werden. Nur über Umwege ist eine automatische Planerstellung aufgrund zuvor hinterlegter Regelungen innerhalb von MS EXCEL™ möglich.
- Auswertungen sind mit MS EXCEL™ meist sehr aufwändig und nicht im nötigen Detaillierungsgrad lieferbar.
- Keine Protokollierung von Änderungen im Plan (wer hat wann was geändert).
- Zu verplanende Mitarbeiter haben keinen Zugriff auf die MS EXCEL™-Tabelle. Daher sind Tauschbörsen und Mitarbeiterwünsche nicht ohne Weiteres elektronisch abbildbar.
- MS EXCEL™ ermöglicht nur eingeschränkte Plausibilitätsprüfungen des Einsatzplans. Verstöße gegen komplexere Regelungen werden u.U. nicht erkannt.
- Findet keine Zeiterfassung statt (Negativzeiterfassung), werden die Planwerte als Istwerte unterstellt. Voraussetzung ist, dass Änderungen wie Abwesenheiten oder Mehrarbeit im Plan nachgetragen werden. Die Übergabe aller MS EXCEL™-Daten an das Lohnsystem ist durch viele manuelle Tätigkeiten geprägt, sehr zeitraubend sowie fehleranfällig.

Die Komplexität des Planungsprozesses sowie die dargestellten Defizite von Tabellenkalkulationen verdeutlichen die Notwendigkeit effizienter Softwarelösungen für das WFM. Doch selbst wenn Tabellenkalkulationen abgeschafft und moderne Softwarelösungen eingesetzt werden, bereitet die manuelle Planerstellung oft große Probleme. Zwar werden die Planer nun umfassend unterstützt, vermeiden Fehler, können schneller sowie flexibler agieren und haben alle notwendigen Informationen online zur Verfügung. Aber die Komplexität der Personaleinsatzplanung wird so nicht reduziert. Die Planerstellung wird lediglich beschleunigt, da alle Informationen sofort zur Verfügung stehen und nicht aufwändig beschafft bzw. nachgerechnet werden müssen. Integraler Bestandteil muss

Kapitel 2 Begrifflichkeiten und Zusammenhänge

daher besonders bei großen Problemstellungen und/oder komplexen Regelungen eine automatische Planvorschlagsgenerierung sein.

In diesem Kapitel wurden grundlegende Begriffe und Prozesse dargestellt. Zudem wurde die Notwendigkeit funktional tiefer Softwarelösungen und effizienter Optimierungsverfahren erörtert. Im folgenden Kapitel wird auf drei Problemstellungen eingegangen, die direkt aus der Praxis stammen. An ihnen werden später Lösungsverfahren erprobt und beurteilt.

Kapitel 3

Problemstellungen für hochflexibles Workforce Management

In diesem Kapitel werden drei praktische Problemstellungen vorgestellt, die auf Echt-daten basieren. Zum einen handelt es sich um die untertägige Personaleinsatzplanung in der Logistik, um die automatische Arbeitszeitmodellerstellung im Handel und um die kombinierte Touren- und Personaleinsatzplanung beim Technikereinsatz der British Telecom. Alle drei Problemstellungen können dem hochflexiblen WFM zugeordnet werden, da die Planung in sehr kleinen Zeitintervallen erfolgt, zumeist untertägige Arbeitsplatz-/Auftragswechsel notwendig sind und z.T. eine Zusammenlegung von Phasen des WFM (Kapitel 2.3) erfolgt.

Bei der Auswahl der Problemstellungen wurde darauf geachtet, dass sie aus Branchen mit vielen Arbeitnehmern stammen und möglichst leicht auf andere Branchen übertragbar sind. Zudem werden verschiedene Varianten der Problemstellungen erstellt, um die gewonnenen Erkenntnisse auf ein möglichst breites Fundament zu stellen. Somit soll verhindert werden, dass Aussagen lediglich zu Nischenthemen gemacht werden.

Zunächst wird eine Klassifikation von Problemstellungen vorgestellt, in der die drei Problemstellungen später eingeordnet werden. In den Unterkapiteln 3.2 bis 3.4 werden die einzelnen Probleme anschließend erläutert. Neben der Beschreibung der individuellen Besonderheiten des jeweiligen Planungsszenarios wird in jedem der drei Unterkapitel auf die reale Problemstellung und das mathematische Modell eingegangen. Zudem werden diverse Varianten eines Problems vorgestellt, das Problem wird klassifiziert und es wird jeweils auf ähnliche Problemstellungen in der Literatur eingegangen.

3.1 Klassifikation von Problemstellungen

Zunächst wird ein Überblick über die Klassifikation von Problemstellungen gegeben, in die die drei Planungsprobleme dieser Arbeit eingeordnet werden können. Anschließend werden die einzelnen Klassen kurz erläutert.

Eine Einleitung in die Personaleinsatzplanung geben Nanda und Browne [280]. Sie gruppieren Personaleinsatzplanungsprobleme in drei Klassen: Shift Scheduling (Planung von Arbeits- und Pausenzeiten je Mitarbeiter), Day-Off Scheduling (Arbeits- und Ruhetagsplanung je Mitarbeiter) und Tour Scheduling (Kombination aus den beiden erstgenannten Klassen). Nanda und Browne geben auch einen Überblick über Arbeiten zu allen drei Klassen.

Ernst u.a. liefern 2004 eine Zusammenfassung von Arbeiten rund um das Thema Personaleinsatzplanung [114]. Sie haben hierzu etwa 700 Arbeiten von 1954 bis 2004 einbezogen. Zwar ist die Übersicht nicht vollständig – in Anbetracht der Tatsache, dass Arbeiten aus einem halben Jahrhundert betrachtet wurden, ist dies jedoch durchaus verständlich. Umfangreiche Quellen für weitere Übersichten sind ebenfalls angegeben. Ernst u.a. erweitern die Einteilung von Nanda und Browne, wobei sie sich an einer zweiten Arbeit orientieren, die sie ebenfalls 2004 veröffentlicht haben [115]. Tabelle 3.1 zeigt diese Klassifikation und die Anzahl der in [114] enthaltenen Arbeiten, die der entsprechenden Klasse zugeordnet werden können.

Tabelle 3.1: Klassifikation nach Ernst u.a. [114] und Anzahl der dort zuzuordnenden Arbeiten

Lösungsverfahren	Anzahl Arbeiten	Lösungsverfahren	Anzahl Arbeiten
Crew Scheduling	219	Task Based Demand	47
Tour Scheduling	185	Demand Modelling	40
Flexible Demand	107	Task Assignment	32
Workforce Planning	99	Shift Assignment	24
Crew Rostering	76	Disruption Management	16
Shift Scheduling	64	Other Classifications	12
Cyclic Roster	62	Stint Based Roster	9
Days Off Scheduling	61	Roster Assignment	6
Shift Based Demand	55		

Dass eine Zuordnung von Problemstellungen nicht immer überschneidungsfrei vorgenommen werden kann, zeigt die Arbeit von Schaerf und Meisels [335]. Sie wurde von Ernst u.a. dem Task Assignment zugeordnet, da dort eine Aufgabe eine komplette Schicht

ausfüllt. Zudem wurden Mitarbeitern zuvor noch keinen Schichten zugeteilt, sodass es sich auch um ein Shift Scheduling Problem handelt. Nachfolgend werden die einzelnen Klassen kurz erläutert.

Demand Modelling Es werden drei Gruppen unterschieden, die sich daran orientieren, für was der Bedarf angegeben wird. Hierzu gehört Task Based Demand, Flexible Demand und Shift Based Demand.

Task Based Demand Beim Task Based Demand wird er für Aufgaben oder Projekte angegeben, die durch Zusatzinformationen wie frühester Beginn, spätestes Ende, Dauer, Ort, benötigte Qualifikationen etc. charakterisiert sein können.

Flexible Demand Flexible Demand wird dann verwendet, wenn man wenig konkrete Informationen über Zeitpläne und anstehende Aufträge hat (z.B. im Inbound Call Center). Hier gibt man einen Bedarf pro Zeitintervall (15 Minuten, Vormittag, 1 Tag, ...) und Qualifikation an, der durch Forecastmodelle ermittelt wird. Es ist also bekannt, welches Arbeitsvolumen wann anfällt. Nicht aber für welchen Kunden und für welchen Auftrag.

Shift Based Demand Shift Based Demand ist oft bei der Dienstplanung zu finden. Der Bedarf an Personal wird pro Schicht angegeben, welche meist die tägliche Arbeitszeit eines Mitarbeiters darstellt.

Days Off Scheduling Days-Off Scheduling meint die Planung von Ruhetagen der Mitarbeiter innerhalb des Planungshorizonts. Diese Planungsart ist häufig in Zusammenhang mit flexiblen und schichtbasierten Bedarfen anzutreffen. Mit der gegenteiligen Planung, dem Working-Day Scheduling würde man die Arbeitstage der Mitarbeiter planen.

Shift Scheduling Hier werden die am besten geeigneten Schichten aus einem Pool von Schichten ausgewählt, um einen gegebenen Bedarf möglichst optimal abzudecken und dabei u.a. Über- und Unterdeckungen zu vermeiden.

Tour Scheduling Tour Scheduling kombiniert Days Off Scheduling und Shift Scheduling. Tour Scheduling lässt sich auf Shift Scheduling reduzieren, wenn der Planungshori-

zont ein Tag ist. Eine Reduzierung auf Days Off Scheduling ist dann möglich, wenn nur eine Schicht pro Tag existiert.

Shift Assignment Shift Assignment ist ein Spezialfall von Tour Scheduling, bei dem Arbeits- und Ruhetage bereits als Input gegeben sind, sodass nur noch die Schichten auf Mitarbeiter verplant werden müssen.

Task Assignment Beim Task Assignment werden Aufträge, die bestimmte Qualifikationen erfordern und in einem bestimmten Zeitraum ausgeführt werden müssen, auf Mitarbeiter verteilt. Den Mitarbeitern wurden zuvor Schichten zugeteilt, sodass deren An- und Abwesenheitszeiten sowie -tage bekannt sind.

Roster Assignment Mitarbeiter werden einem fertigen Einsatzplan für dessen komplette Länge zugeordnet, sodass jeder Mitarbeiter seine Arbeitszeiten und den Arbeitsplatz für den gesamten Planungshorizont kennt.

Crew Scheduling Das Crew Scheduling ist meist in Zusammenhang mit Transportsystemen zu finden (Flugzeug, Zug, Bus) und ist eines der meist bearbeiteten Gebiete der Personaleinsatzplanung. Hier werden Einsätze erzeugt, die aus mehreren Rundfahrten bestehen können. Wobei die Ermittlung gültiger Einsätze häufig recht komplex ist.

Crew Rostering Crew Rostering dient der Erzeugung eines optimalen Einsatzplans, in dem die erzeugten Einsätze des Crew Scheduling auf Mitarbeiter verplant werden.

Cyclic Roster Es wird ein Schichtzyklus erzeugt der aus mehreren Schichten besteht. Dieser Schichtzyklus kann später Mitarbeitern zugewiesen werden, die dann verschiedene Einsprungzeitpunkte haben. Schichtzyklen werden häufig bei wiederkehrenden Bedarfen verwendet.

Stint Based Roster Beim Stint Based Roster werden Schichtfolgen inkl. Abwesenheiten für einige Tage erzeugt.

Disruption Management Disruption Management wird nur selten berücksichtigt. Ziel ist es, Einsatzpläne zu erzeugen, die wenig anfällig für Unterbrechungen sind.

Workforce Planning Einen langfristigen Horizont hat Workforce Planning. Es dient der Ermittlung des Mitarbeiterstamms, der z.B. durch Neueinstellungen und Entlassungen beeinflusst werden kann.

3.2 Untertägige Personaleinsatzplanung in der Logistik

Im vorangegangenen Kapitel wurde ein Überblick über die Klassifikation von Problemstellungen gegeben. In diesem Kapitel wird zunächst auf die Besonderheiten der untertägigen Planung eingegangen. In Kapitel 3.2.2 wird anschließend eine konkrete Problemstellung aus der Praxis beschrieben. Um die Lösungsverfahren später an einer möglichst großen Zahl an Problemstellungen mit unterschiedlichen Eigenschaften zu erproben, werden in Kapitel 3.2.3 verschiedenen Varianten der untertägigen Einsatzplanung erstellt. Mit einer Klassifikation der Problemstellung anhand der Gruppierung in Kapitel 3.1 und Ausführungen zu ähnlichen Problemen endet dieses Kapitel.

3.2.1 Untertägige Personaleinsatzplanung

Trotz vieler Fortschritte in der Automatisierung ist die Logistik in weiten Teilen sehr arbeitsintensiv. Zur gleichen Zeit ist sie einem enormen Druck ausgesetzt. Logistiker sind gezwungen, Maßnahmen zu ergreifen, um weiterhin wettbewerbsfähig zu bleiben. Doch im Gegensatz zur Optimierung der Logistikprozesse wurde die Optimierung beim Einsatz von Personal in der Vergangenheit eher vernachlässigt. Eine bedeutende Stellschraube auf der Kostenseite ist vor allem beim Personal zu finden. Hier kann man u.a. mit der untertägigen Personaleinsatzplanung zur Flexibilisierung des Personaleinsatzes und somit zur Kostenreduktion beitragen.

Bei der ganztägigen Personaleinsatzplanung erfolgt die Zuweisung eines Mitarbeiters bzgl. Arbeitsplatz und Arbeitszeitmodell für den kompletten Tag. Oft werden bei der Planung gleich mehrere Tage gemeinsam geplant (Wochenplan, Monatsplan). Klassische Beispiele sind die Dienstplanung im Krankenhaus und die Schichtplanung in einem Produktionsunternehmen. Für nähere Erläuterungen zu Abläufen der ganztägigen Personaleinsatzplanung sei auf die einschlägige Literatur zu diesem Thema verwiesen: Meisel und Schaerf [258], Scherf [337] sowie Hesseln und Zander [183]. Im Gegensatz dazu werden Mitarbeiter bei der untertägigen Personaleinsatzplanung nicht mehr fest auf einen Arbeitsplatz für den ganzen Tag geplant. Sie können ihren Arbeitsplatz beliebig oft wechseln. Dies ist eine Anforderung, die besonders im Dienstleistungsbereich häufig gefordert ist, da dort gewöhnlich kurzzeitige Bedarfe an Arbeitskräften auf den verschiedenen Arbeitsplätzen auftreten. So arbeitet ein Mitarbeiter beispielsweise von

8.00–10.30 Uhr im Verkauf und von 10.30–13.15 Uhr an Kasse. Die Zeitintervalle, in denen ein Mitarbeiter auf einem Arbeitsplatz eingeplant ist, können beliebig klein sein. Aus praktischen Gründen machen jedoch Zeitintervalle kleiner als 15 Minuten wenig Sinn, wobei es auch Ausnahmen gibt. Personaleinsatzplanung mit einer untertägigen Planung kann in zwei Varianten erfolgen:

1. Der zu planende Zeitraum (z.B. eine Woche oder ein Monat) wird in einem einzigen Planungsschritt komplett untertägig geplant. Es erfolgt somit keine Zerlegung der Problemstellung in Teilprobleme wie dies bei der zweiten Variante der Fall ist. Die Komplexität derartiger Probleme ist daher meist ungleich größer. Das Handelsproblem mit zwei Arbeitsplätzen (siehe Kapitel 3.3) basiert u.a. auf diesem Vorgehen.
2. Zur Komplexitätsreduzierung wird das Planungsproblem zerlegt. Zunächst wird im Rahmen der ganztägigen Personaleinsatzplanung ein Wochen- oder Monatsplan erstellt. Dort werden den Mitarbeitern Arbeitszeitmodelle und Stammarbeitsplätze für die einzelnen Tage zugewiesen. Anschließend werden die Mitarbeiter in einem zweiten Planungsschritt untertägig auf einzelne Arbeitsplätze verplant, wobei die Informationen der ganztägigen Einsatzplanung genutzt werden. Im folgenden Kapitel wird das Logistikproblem vorgestellt, bei dem dieses Vorgehen angewendet wird.

Von der gewählten Planungsvariante hängt ab, welche Nebenbedingungen relevant sind. Wird die Planung in einem einzigen Schritt vollzogen, so sind zusätzlich zu den Nebenbedingungen der untertägigen Planung auch die der ganztägigen Planung zu berücksichtigen. Die Nebenbedingungen im Rahmen der ganztägigen Planung können äußerst komplex und umfangreich sein, wie die Arbeit von Vanden Berghe zur Dienstplanung beispielhaft zeigt [391, S. 34–66].

Sollte das Planungsproblem zur Komplexitätsreduzierung zunächst ganztägig und anschließend untertägig gelöst werden, ist für die untertägige Personaleinsatzplanung neben An-/Abwesenheiten der ganztägigen Planung vor allem die maximale Anzahl der untertägigen Arbeitsplatzwechsel wichtig. Zu häufige Arbeitsplatzwechsel sind nicht arbeitsergonomisch und werden von Mitarbeitern als störend und belastend empfunden. Zudem würden sich bei zu vielen Wechseln organisatorische Probleme ergeben. Es ist z.B. nicht sinnvoll, dass ein Mitarbeiter zehn Mal am Tag den Arbeitsplatz wechselt. Dies ist einerseits schwer zu realisieren und stellt aufgrund möglicher Wegzeiten und eventueller Einarbeitungszeiten eine Ressourcenverschwendung dar. In der Planung besteht daher die Herausforderung, den Bedarf bestmöglich zu decken und dabei möglichst wenigen Arbeitsplatzwechsel zu vollziehen.

3.2.2 Beschreibung der Problemstellung

Die vorliegende Problemstellung stammt von einem deutschen Logistikdienstleister. In diesem Unternehmen wird auf einem räumlich beschränkten Areal an 7 Tagen in der Woche jeweils 20 Stunden gearbeitet. Angenommen sei eine Menge an Mitarbeitern $\mathcal{E} = \{1, \dots, E\}$, eine Menge an Arbeitsplätzen $\mathcal{W} = \{1, \dots, W\}$ und ein diskreter Zeithorizont \mathcal{T} mit dem Index $t = 0, \dots, T - 1$. Jede Periode t des Zeithorizonts hat eine Länge l_t , die größer als null ist.

$$l_t > 0 \quad \forall t \in \mathcal{T} \quad (3.1)$$

Die Zuweisung eines der 65 Mitarbeiter zu einem von neun Arbeitsplätzen wird über die binäre Variable x_{ewt} gesteuert.

$$x_{ewt} = \begin{cases} 1, & \text{wenn Mitarbeiter } e \text{ dem Arbeitsplatz } w \\ & \text{in Periode } t \text{ zugewiesen wurde} \\ 0, & \text{sonst} \end{cases} \quad (3.2)$$

Der Personalbedarf d_{wt} an Mitarbeitern pro Arbeitsplatz und Periode ist weit im Voraus bekannt darf nicht negativ sein.

$$d_{wt} \geq 0 \quad \forall w \in \mathcal{W} \text{ und } \forall t \in \mathcal{T} \quad (3.3)$$

Der Personalbedarf wird ereignisorientiert ermittelt. Ein Ereignis löst verschiedene zeitlich feste Aktivitäten auf verschiedenen Arbeitsplätzen aus. Die Aktivitäten können vor und nach dem eigentlichen Ereignis beginnen und enden. Kurzfristige Veränderungen treten nur äußerst selten auf, sodass in diesem Punkt eine hohe Planungssicherheit besteht. In Abbildung 3.1 ist der Personalbedarf der neun Arbeitsplätze in 15-Minuten-Intervallen gegeben. Dort ist er beispielhaft für den Freitag in einem Ausschnitt von 3.30–17.30 Uhr dargestellt.

Da es keine flexiblen Arbeitszeitmodelle gibt, sind diese lediglich durch ihre Kommen- und Gehenzeiten definiert. Alle Arbeitszeitmodelle sind in Tabelle 3.2 dargestellt. Die Mitarbeiter sind hinsichtlich ihrer Arbeitszeiten recht flexibel, was sich in der Vielfältigkeit der Arbeitszeitmodelle niederschlägt. Für die untertägige Einsatzplanung der Mitarbeiter stehen ihre Abwesenheiten (z.B. Urlaub, Fortbildung) und zugeteilten Arbeitszeitmodelle aus der Monatsplanung fest. Die Verfügbarkeit der Mitarbeiter ist somit zu Beginn der untertägigen Planung bekannt und wird mit der binäre Variable a_{et} festgelegt.

$$a_{et} = \begin{cases} 1, & \text{wenn Mitarbeiter } e \text{ in Periode } t \text{ anwesend ist} \\ 0, & \text{sonst} \end{cases} \quad (3.4)$$

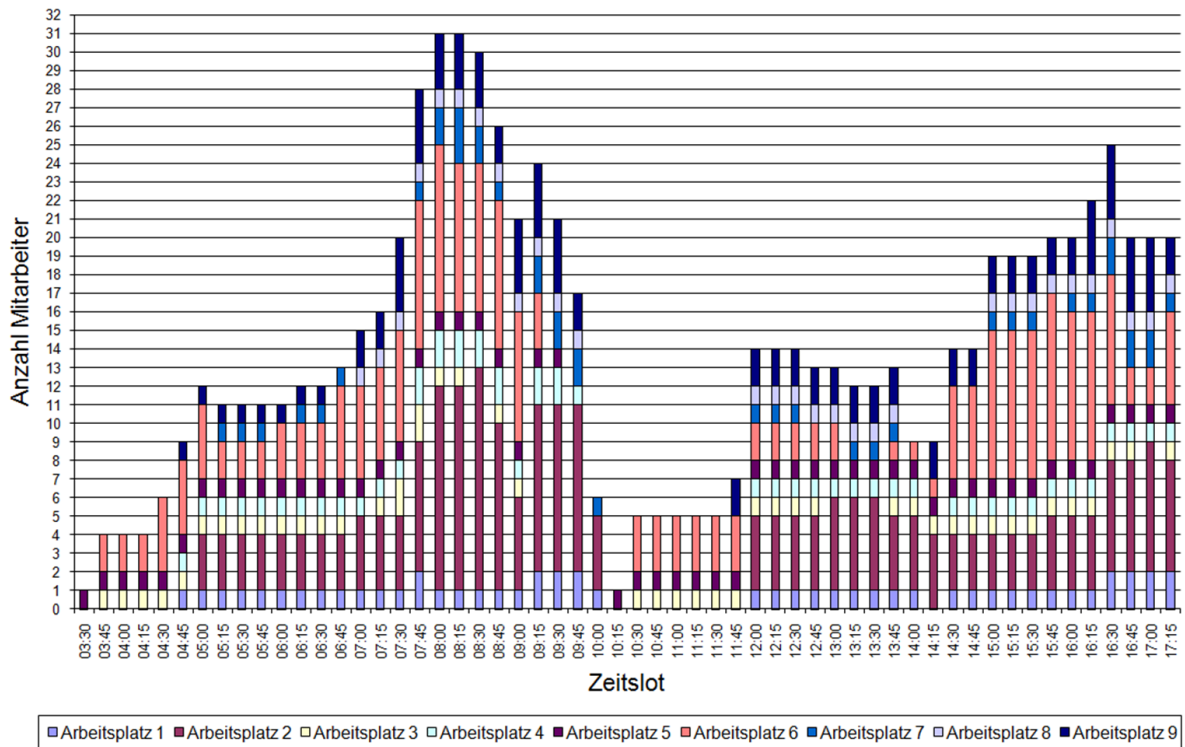


Abbildung 3.1: Personalbedarf beim Logistikproblem im Zeitverlauf eines Tages bei neun Arbeitsplätzen (Ausschnitt 3.30–17.30 Uhr)

Aktuell findet mit MS EXCEL™ eine manuelle Personaleinsatzplanung auf Monatsbasis statt, bei der Mitarbeiter ein Arbeitszeitmodell und meist einen festen Arbeitsplatz pro Tag zugewiesen bekommen. Dies geschieht unter Berücksichtigung von An- und Abwesenheiten, Mehrarbeitssalden, Qualifikationen sowie von Ruhezeiten. Auch die Abweichung von den Besetzungsvorgaben pro Arbeitsplatz wird bei der Planung möglichst minimiert. Der Personalbedarf der Arbeitsplätze unterliegt während des Tages meist großen Schwankungen. Mitarbeiter werden in der Personaleinsatzplanung jedoch vorwiegend ganztägig auf Arbeitsplätze verplant, sodass es weite Phasen der Über- und Unterdeckung an Mitarbeitern gibt. Dies lässt die Servicequalität und Mitarbeitermotivation sinken und führt zu unnötigen Personalkosten sowie Leerzeiten. Derzeit wird nur selten ein untertägiger Arbeitsplatzwechsel bereits in der Planung vollzogen, da man hier mit MS EXCEL™ schnell an Grenzen stößt. Größtenteils greifen Abteilungsleiter regulierend in die Abläufe vor Ort ein und besetzten Mitarbeiter auf Zuruf um. Eine wirklich bedarfsorientierte Personaleinsatzplanung lässt sich mit diesem Vorgehen jedoch nicht realisieren. Ziel soll es daher sein, ausgehend von der ganztägigen Personaleinsatzplanung auf Monatsbasis, einen bedarfsorientierten untertägigen Einsatzplan zu erstellen (Variante 2 in Kapitel 3.2.1). Damit soll der Personalbedarf besser abgedeckt werden, was zu weniger Leerzeiten und Überstunden sowie zu einer Reduzierung der Überbeanspruchung einzelner Mitarbeiter führt.

Tabelle 3.2: Arbeitszeitmodelle beim Logistikproblem mit Arbeitsbeginn und -ende

Arbeitszeitmodell	Arbeitsbeginn	Arbeitsende
1	3.30	9.45
2	4.30	10.00
3	5.00	13.00
4	6.45	15.00
5	7.30	13.00
6	7.30	15.30
7	9.45	17.45
8	13.00	21.00
9	15.00	20.30
10	15.00	21.45
11	15.00	23.15
12	15.30	23.30
13	17.45	23.30

Zu den harten Nebenbedingungen gehört, dass die Arbeitszeitmodelle festgeschrieben sind und bei der untertägigen Planung nicht mehr verändert werden dürfen. Zudem dürfen Arbeitsplatzwechsel nur in 15-Minuten-Intervallen stattfinden. Ein Mitarbeiter kann in einer Periode t nur dann einem Arbeitsplatz w zugeordnet werden, wenn er auch tatsächlich anwesend ist. Er kann auch immer nur einem Arbeitsplatz zur gleichen Zeit zugewiesen werden.

$$\sum_{w=1}^W x_{ewt} = a_{et} \quad \forall e \in \mathcal{E} \text{ und } \forall t \in \mathcal{T} \quad (3.5)$$

Im Gegensatz zu diesen harten Nebenbedingungen wird die Verletzung von weichen Nebenbedingungen mit Fehlerpunkten bestraft. Die exakte Bestimmung der Fehlerpunkthöhe ist in der Praxis ein iterativer Prozess aus Variation der Punkthöhe und anschließender Analyse des Einsatzplans, der so lange wiederholt wird, bis der Planer mit dem Ergebnis zufrieden ist. Die Höhe der gewählten Fehlerpunkte kann sich dabei nicht nur von Unternehmen zu Unternehmen unterscheiden, sondern sogar zwischen den Abteilungen eines Unternehmens. Die hier verwendeten Fehlerpunkte stammen aus einem Interview mit dem Logistikdienstleister und spiegeln dessen Anforderungen wider.

Besonders in Bezug auf Qualifikationen gibt es strenge Regelungen, da es beim Einsatz nicht qualifizierter Mitarbeiter schnell zu großen Sach- und Personenschäden kommen kann. Jeder Arbeitsplatz kann eine Menge an Qualifikationen Q_w erfordern. Auf der

anderen Seite können Mitarbeiter über eine Menge an Qualifikationen Q_e verfügen. Der Arbeitgeber investiert regelmäßig viel Geld und Zeit in Qualifikationsmaßnahmen, sodass viele Mitarbeiter auf unterschiedlichen Arbeitsplätzen arbeiten können. Bei der Besetzung müssen für sieben der neun Arbeitsplätze Qualifikationen berücksichtigt werden. Die beiden Arbeitsplätze ohne Qualifikationen befinden sich in keinem Sicherheitsbereich. Dort müssen lediglich einfache Pack- bzw. Bedientätigkeiten ausgeführt werden. Alle anderen Arbeitsplätze setzen z.T. kostspielige Schulungen und Ausbildungen voraus. Bei den Qualifikationen handelt es sich u.a. um Weisungsbefugnisse, erfolgreich absolvierte Schulungen, Zertifikate, Feuerwehr-, Ersthelfer- und Technikerausbildungen sowie um die Befähigung, Flurförderzeuge, Busse, Schlepper und dgl. zu fahren. Die Vielzahl der Qualifikationen wurde zu vier Qualifikationsgruppen zusammengefasst. Wird ein Mitarbeiter auf einen Arbeitsplatz geplant und verfügt nicht über alle erforderlichen Qualifikationen, so entstehen Fehlerpunkte P_q für die Dauer der Besetzung entsprechend der Fehlerpunkthöhe c_q . In diesem Fall ist c_q mit 200 Fehlerpunkten pro Minute bewertet. Somit ist diese Nebenbedingung im Vergleich zu allen folgenden Nebenbedingungen mit Abstand am stärksten gewichtet.

$$P_q = \sum_{t=0}^{T-1} \sum_{w=1}^W \sum_{e=1}^E c_q l_t x_{ewt} \begin{cases} c_q = 200, & \text{wenn Mitarbeiter } e \text{ nicht für} \\ & \text{Arbeitsplatz } w \text{ qualifiziert ist} \\ c_q = 0, & \text{sonst} \end{cases} \quad (3.6)$$

Das Ziel der untertägigen Personaleinsatzplanung ist die Reduzierung der Besetzungsabweichungen. Findet eine Abweichung von der Besetzungsvorgabe d_{wt} statt, so entstehen Fehlerpunkte P_d für die Dauer und Höhe der Fehlbesetzung entsprechend der Fehlerpunkthöhe. Dabei wird zwischen verschiedenen Fehlertypen unterschieden: c_{do} bei Überdeckung, wenn der Bedarf $d_{wt} > 0$ ist, c_{dn} bei Überdeckung, wenn der Bedarf $d_{wt} = 0$ ist sowie c_{du} bei einer Unterdeckung. Mit einer höheren Gewichtung von c_{dn} kann vermieden werden, dass Mitarbeiter auf Arbeitsplätze geplant werden, bei denen in der Periode t rein gar nichts zu tun ist.

$$P_d = \sum_{t=0}^{T-1} \sum_{w=1}^W (c_{do} + c_{dn} + c_{du}) l_t \left| \left(\sum_{e=1}^E x_{ewt} \right) - d_{wt} \right|, \quad (3.7)$$

mit:

- $c_{do} = 1$, wenn auf Arbeitsplatz w in Periode t eine Überdeckung im Personaleinsatz vorliegt mit $d_{wt} > 0$, sonst $c_{do} = 0$.
- $c_{dn} = 2$, wenn auf Arbeitsplatz w in Periode t eine Überdeckung im Personaleinsatz vorliegt mit $d_{wt} = 0$, sonst $c_{dn} = 0$.
- $c_{du} = 1$, wenn auf Arbeitsplatz w in Periode t eine Unterdeckung im Personaleinsatz vorliegt mit $d_{wt} > 0$, sonst $c_{du} = 0$.

Die letzte Nebenbedingung bezieht sich auf die Anzahl der Arbeitsplatzwechsel pro Mitarbeiter r_e . Um r_e nicht unbegrenzt anwachsen zu lassen, wird jeder Arbeitsplatzwechsel mit Fehlerpunkten c_r bestraft. Jeder Arbeitsplatzwechsel – egal wie lange die Tätigkeit dort dauert – wird mit genau einem Fehlerpunkt bewertet ($c_r=1$), sodass die Anzahl der Arbeitsplatzwechsel gegenüber der anderen Nebenbedingungen weit untergeordnet ist.

$$P_r = c_r \sum_{e=1}^E r_e \quad (3.8)$$

Die Zielfunktion, die es zu minimieren gilt, lautet:

$$\min P = P_q + P_d + P_r. \quad (3.9)$$

Zur Bearbeitung der Problemstellung wurden vom Logistikdienstleister Echtdaten einer Woche zur Verfügung gestellt. Um die Daten verwenden zu dürfen, wurden alle Informationen soweit anonymisiert, dass keine Rückschlüsse auf die Mitarbeiter und das Unternehmen gezogen werden können. Im folgenden Kapitel werden weitere Varianten der Problemstellung vorgestellt.

3.2.3 Varianten der Problemstellung

Lösungsverfahren sollen in dieser Arbeit an möglichst facettenreichen Ausprägungen von Problemstellungen beurteilt werden. Daher wird das zuvor beschriebene Problem mit 65 Mitarbeitern und 560 Zeitintervallen (7 Tage) in die einzelnen Tage a 80 Zeitintervalle zerlegt. Da nicht alle Mitarbeiter an allen sieben Tagen arbeiten, ist auch die Anzahl der zu verplanenden Mitarbeiter an den einzelnen Tagen geringer. Alle Wochentage haben weiterhin neun zu besetzende Arbeitsplätze.

Die Zerlegung der kompletten Woche in einzelne Tage ist problemlos möglich, da sich die Tage nicht gegenseitig beeinflussen. Denn auch hier sind die Schichten bereits vorgegeben – lediglich untertägige Arbeitsplatzwechsel sind zu planen. Mit dem Wochenproblem und den Problemen der einzelnen Tage ergeben sich in Summe acht Problemstellungen, deren Rahmenbedingungen in Tabelle 3.3 dargestellt sind. Logistik_Complete bezeichnet dabei das Wochenproblem. Die Anzahl der Dimensionen resultiert u.a. aus der Abbildung der Problemstellung. Dies wird in Kapitel 4.1 ausführlich erläutert. Bei Logistik_Complete ergeben sich z.B. die 36.400 Dimensionen aus 65 Mitarbeitern * 560 Zeitintervalle (7 Tage * 20 Stunden * 4 Intervalle pro Stunde = 560).

Die zuvor dargestellten acht Problemstellungen werden nun weiter modifiziert, sodass sich weitere acht zu untersuchende Problemstellungen ergeben. Die Modifikation resultiert aus den zusätzlichen Anforderungen eines zweiten Unternehmens. Dessen Unterneh-

Tabelle 3.3: Rahmenbedingungen der Logistikprobleme ohne Stammarbeitsplatzregelung

Bezeichnung der Problemstellung	Anzahl Mitarbeiter	Anzahl Zeitintervalle	Anzahl Dimensionen
Logistik_Complete	65	560	36.400
Logistik_Mo	43	80	3.400
Logistik_Di	38	80	3.040
Logistik_Mi	44	80	3.520
Logistik_Do	46	80	3.680
Logistik_Fr	45	80	3.600
Logistik_Sa	39	80	3.120
Logistik_So	42	80	3.360

mensstruktur, die Aufgaben etc. werden nachfolgend nicht detailliert erläutert. Relevant sind lediglich die Modifikationen des ursprünglichen Logistikproblems.

Beim zweiten Unternehmen legt man großen Wert auf das Arbeiten in Teams, deren Zusammensetzung möglichst nicht geändert werden soll. Welcher Mitarbeiter welchem Team angehört, wird über dessen individuellen Stammarbeitsplatz m_e geregelt, der hier einem der neun Arbeitsplätze entspricht. Die Anforderung lässt sich demnach so formulieren, dass ein Mitarbeiter nach Möglichkeit immer auf seinem Stammarbeitsplatz arbeiten soll. Verstöße werden unabhängig von ihrer Dauer mit Fehlerpunkten geahndet.

$$P_m = \sum_{t=0}^{T-1} \sum_{e=1}^E c_m \begin{cases} c_m = 1, & \text{wenn } m_e \neq \text{Arbeitsplatzzu-} \\ & \text{weisung bei } e \text{ zum Zeitpunkt } t \\ c_m = 0, & \text{sonst} \end{cases} \quad (3.10)$$

Trotz der Stammarbeitsplatzregelung ist es das eigentliche Ziel der untertägigen Einsatzplanung, den Personalbedarf auf den neun Arbeitsplätzen möglichst passend abzudecken. Hierzu muss Formel 3.7 geändert werden. Aufgrund des Fokus auf Stammarbeitsplätze macht es nun keinen Sinn mehr, erhöhte Strafpunkte zu vergeben, wenn auf einem Arbeitsplatz eine Überdeckung bei einem Bedarf von null besteht. Fehlerpunkte für Über- und Unterdeckung werden nun nach Formel 3.11 vergeben. Man erkennt, dass c_{dn} aus Formel 3.7 entfernt ist.

$$P_d = \sum_{t=0}^{T-1} \sum_{w=1}^W (c_{do} + c_{du}) l_t \left| \left(\sum_{e=1}^E x_{ewt} \right) - d_{wt} \right|, \quad (3.11)$$

mit:

3.2 Untertägige Personaleinsatzplanung in der Logistik

- $c_{do} = 1$, wenn auf Arbeitsplatz w in Periode t eine Überdeckung im Personaleinsatz vorliegt mit $d_{wt} \geq 0$, sonst $c_{do} = 0$.
- $c_{du} = 1$, wenn auf Arbeitsplatz w in Periode t eine Unterdeckung im Personaleinsatz vorliegt mit $d_{wt} > 0$, sonst $c_{du} = 0$.

Weitere Modifikationen der ursprünglichen Problemstellung sind nicht erforderlich, so dass sich die zu minimierende Zielfunktion wie folgt ergibt.

$$\min P = P_q + P_d + P_r + P_m \quad (3.12)$$

Das Planungsszenario des zweiten Unternehmens mit Stammarbeitsplätzen unterscheidet sich hinsichtlich der durchzuführenden Tätigkeiten, Anzahl Mitarbeitern und Arbeitsplätzen sowie bei den Schwankungen im Personalbedarf nicht wesentlich vom Problem in Kapitel 3.2.2. Da vom zweiten Unternehmen keine Echtdateien in der erforderlichen Granularität vorliegen, werden die Daten des ursprünglichen Logistikproblems verwendet. Dort ist auch bekannt, welcher Mitarbeiter welchem Stammarbeitsplatz zugeordnet ist. In Tabelle 3.4 sind die Rahmenbedingungen der acht Problemstellungen des zweiten Unternehmens aufgeführt. In der Bezeichnung der Problemstellungen wird jeweils $_S$ eingefügt.

Tabelle 3.4: Rahmenbedingungen der Logistikprobleme mit Stammarbeitsplatzregelung

Bezeichnung der Problemstellung	Anzahl Mitarbeiter	Anzahl Zeitintervalle	Anzahl Dimensionen
Logistik_S_Complete	65	560	36.400
Logistik_S_Mo	43	80	3.400
Logistik_S_Di	38	80	3.040
Logistik_S_Mi	44	80	3.520
Logistik_S_Do	46	80	3.680
Logistik_S_Fr	45	80	3.600
Logistik_S_Sa	39	80	3.120
Logistik_S_So	42	80	3.360

Mit der Einführung von Fehlerpunkten für Abweichungen vom Stammarbeitsplatz ist der Lösungsraum nun deutlich zerklüfteter als bei der originalen Problemstellung in Kapitel 3.2.2. So lange keine zusätzlichen Qualifikationsfehler oder Arbeitsplatzwechsel auftreten, war es dort egal, ob z.B. Mitarbeiter 1 oder Mitarbeiter 2 in einem Zeitslot auf Arbeitsplatz 1 arbeitet. Durch die Einführung von Stammarbeitsplätzen führt die

Auswahl von Mitarbeiter 1 oder 2 im entsprechenden Zeitslot viel eher zu verschiedenen Fehlerpunkten. Und auch die Auswirkungen auf die nachfolgenden Zeitintervalle des Tages sind nun größer.

3.2.4 Einordnung der Problemstellung

Bei der Problemstellung des Logistikdienstleisters ist der Personalbedarf von Aufträgen abhängig. Jeder Auftrag erzeugt innerhalb fester Start- und Endzeitpunkte einen Bedarf an Personal für einen oder mehrere Arbeitsplätze, die meist bestimmte Qualifikationen erfordern. Es gibt Mindestbedarfe und z.T. auch eine Begrenzung des Bedarfs, wenn sich mehrere Aufträge zeitlich überschneiden. Dem Task Based Demand ist die Problemstellung jedoch nicht zuzuordnen, da der Logistikdienstleister in einem zweiten Schritt alle Bedarfe der Aufträge pro Arbeitsplatz summiert und sie auf Basis von Zeitintervallen angibt. Dies ist deshalb möglich, da die Aufträge eine Besetzung von Arbeitsplätzen erfordern, die immer räumlich zusammenfallen. Ein Beispiel soll den Zusammenhang veranschaulichen: eine Person am Serviceschalter für Auftrag A und zwei Personen am Serviceschalter für Auftrag B ergeben drei Personen am Serviceschalter. Eine Differenzierung, wer für welchen Auftrag am Serviceschalter zuständig ist, ist bei diesem Problem nicht erforderlich. Daher ist die Problemstellung des Logistikdienstleisters der Gruppe des Task Based Demands zuzuordnen, sondern der Gruppe des Flexible Demands – auch wenn hier recht viele Informationen über den Personalbedarf und die zu Grunde liegenden Aufträge vorliegen. Doch der Bedarf wird in sehr kleinen Zeitintervallen angegeben und im Vorfeld errechnet.

Weiterhin kann die Problemstellung dem Task Assignment zugeordnet werden. Zwar ist der Bedarf nicht in Form von Aufträgen gegeben und es werden auch keine Wegzeiten einbezogen. Aber es müssen Mitarbeiter verplant werden, denen bereits Schichten zugewiesen wurden. Die Einsatzplanung erfolgt also untertäglich auf Arbeitsplätze für ein bestimmtes Zeitintervall, wobei u.a. Qualifikationen berücksichtigt werden.

3.2.5 Vergleichbare Problemstellungen in der Literatur

In der Literatur wurde keine Arbeit gefunden, die exakt der vorliegenden Problemstellung des Logistikdienstleisters entspricht. Vielmehr existieren einige interessante Ansätze, die Facetten der Problemstellung behandeln.

Die Arbeit von Vanden Berghe aus dem Jahre 2002 [391] stellt einen sehr interessanten untertägigen Planungsansatz dar (Planungsdaten und Benchmarktests sind in [88] zu finden). Bei diesem Ansatz gibt man für den Bedarf untertägige Zeitperioden an (Flexible Demand) und entkoppelt so den Personalbedarf von festen Schichten. Im Zuge der

Verplanung von Mitarbeitern auf Schichten, die nun aus den einzelnen Perioden bestehen, entstehen flexiblere Einsatzpläne mit weniger Leerzeiten. Es muss jedoch beachtet werden, dass konkrete Arbeitsplätze nicht vorgesehen sind. Eine untertägige Personaleinsatzplanung erfolgt daher nur auf Schichten (Zeit) und nicht auf Arbeitsplätze (Ort). Vanden Berghe verwendet für ihre Experimente u.a. Variable Neighbourhood Search (VNS), Hybrid Tabu Suche und einen Memetischen Algorithmus, wobei letzterer meist am besten abschnitt.

Meisels und Schaerf geben 2003 [258] eine allgemeingültige Definition eines Employee Timetabling Problems (ETP), wobei sie eine Problemstellung analog zu ihrer Arbeit von 1999 heranziehen [335], in der eine Planung von Aufgaben erfolgt. Es gibt Schichten und Aufgaben, wobei eine Schicht für mehrere Aufgaben gelten kann. Aufgaben haben immer die gleiche Länge der zugehörigen Schicht. Mitarbeiter werden den Schichten zugewiesen und übernehmen eine Aufgabe je nach Qualifikation. Da die Aufgabe für die Dauer der kompletten Schicht gilt, erfolgen keine untertägigen Wechsel der Aufgaben (bzw. Arbeitsplätze). Meisels und Schaerf experimentieren in ihrer Arbeit lediglich mit lokalen Suchverfahren.

Blöchliger gibt ebenfalls in 2003 einer sehr allgemeine Modellierungsbeschreibung, in der der Begriff Schicht nicht zur Modellierung verwendet wird – lediglich als erläuterndes Beispiel [37]. Stattdessen wird von einem Time Table Block (TTB) gesprochen, der eine individuelle Länge haben kann. Dabei könnte ein TTB selbstverständlich auch eine Schicht sein. Mitarbeiter können in diesem Modell nun mehreren aufeinanderfolgenden TTB zugeordnet werden, wodurch untertägige Zeitintervalle innerhalb einer Schicht abbildbar sind. Aufträge werden in der Arbeit von Blöchliger auch berücksichtigt. Ein Auftrag wird jedoch immer fest an einen TTB geknüpft, sodass die binäre Entscheidungsvariable des Modells x_{sb} für die Zuweisung eines Mitarbeiters (s) zu einem TTB (b) steht. Konkrete Lösungsverfahren werden nicht angewendet – es wird lediglich ein kurzer Überblick gegeben.

Bei der vorliegenden Problemstellung des Logistikdienstleisters ist aber bereits bekannt, welche Mitarbeiter in welchem Zeitraum arbeiten, da dies in der ganztägigen Einsatzplanung ermittelt wurde. Es muss bei der Planung des Logistikdienstleisters also nur entschieden werden, auf welchem Arbeitsplatz Mitarbeiter im jeweiligen TTB arbeiten. Im Prinzip stellt die Problemstellung des Logistikdienstleisters eine Kombination aus [258] (Zuweisung auf Aufträge in Zeitintervallen) und [37] (untertägige Zeitintervalle) dar, wobei die Arbeitszeiten der Mitarbeiter bereits vorgegeben sind.

Die Problemstellung des Logistikdienstleisters zur untertägigen Personaleinsatzplanung hat auch Ähnlichkeiten mit Abläufen in Call Centern. Gilat u.a. skizzieren das Optimierungstool der IBM speziell für Call Center, geht jedoch nicht auf Lösungsverfahren ein [141]. In der Arbeit wird auch von Intra-Day Scheduling gesprochen. In Call Centern gibt es Phasen, in denen Mitarbeiter für Inbound-Calls zur Verfügung stehen. Ebenso gibt es Phasen, in denen z.B. Schreibarbeiten erledigt werden müssen (meist als Volumen-

vorgabe pro Zeitintervall hinterlegt). Beim Intra-Day Scheduling werden Mitarbeiter auf verschiedene Tätigkeiten pro Schicht verplant. Ebenso kann beim Intra-Day Scheduling eine Pausenplanung entsprechend der gesetzlichen Regelungen erfolgen. Gilat u.a. gehen nicht auf Outbound Aktivitäten ein. Im Rahmen des Intra-Day Scheduling würden Mitarbeiter dann verschiedenen Outbound Projekten pro Schicht zugeordnet werden. Für einen sehr umfassenden und detaillierten Überblick über die Abläufe in einem Call Center sei auf die Arbeit von Gans u.a. [133] verwiesen.

3.3 Automatische Arbeitszeitmodellerstellung im Handel

Zuvor wurde die untertägige Personaleinsatzplanung in der Logistik beschrieben. In diesem Kapitel soll nun auf den Handel eingegangen werden, wobei die Planungskomplexität vor allem durch die automatische Erstellung von Arbeitszeitmodellen erhöht wird.

Nachfolgend wird die automatische Arbeitszeitmodellerstellung direkt in der Personaleinsatzplanung erläutert. Die konkrete Problemstellung aus dem Handel wird anschließend in Kapitel 3.3.2 beschrieben, woraufhin in Kapitel 3.3.3 die Varianten dieses Problems vorgestellt werden. Dieses Kapitel endet mit einer Klassifikation der Problemstellung und Ausführungen zu ähnlichen Problemstellungen.

3.3.1 Automatische Arbeitszeitmodellerstellung

Der Handel ist sehr personalintensiv und zählt zu einer der größten Branchen Deutschlands [355]. Zur gleichen Zeit ist diese Branche von einem starken Wettbewerb geprägt. Es findet ein Konsolidierungsprozess statt, der weiter anhalten wird. Zudem ist der Handel mit einer sinkenden Kaufkraft ihrer Kunden konfrontiert [264]. Stellenabbau, der vermehrte Einsatz geringfügig Beschäftigter, die Ausweitung der Verkaufsfläche oder des Waren- und Dienstleistungsangebots sind unter diesen Rahmenbedingungen nur einige Maßnahmen des Handels, die jedoch nicht immer erfolgreich sind [226].

Zur Kostensenkung wurde die Mitarbeiteranzahl oft auf ein Minimum reduziert und auch eine Flexibilisierung des Personaleinsatzes wurde umgesetzt. Trotz dieser Flexibilisierung ist es nicht immer möglich, die Arbeitszeiten der Mitarbeiter an den Personalbedarf anzupassen. Eine Überdeckung an Personal führt zu unnötigen Kosten, was den Gewinn schmälert. Eine Unterdeckung an Personal auf der anderen Seite bewirkt einen schlechten Servicelevel und führt somit zu einem Anstieg des Anteils an Nichtkäufern. Im Durchschnitt wird nur jede fünfte Person zu einem Käufer [291]. Diese Anzahl kann mit einem besseren Service reduziert werden. Mit einem schlechten Service wird der mögliche

Umsatz demnach nicht maximal abgeschöpft. Ein am Personalbedarf orientierter Personaleinsatz führt neben einer Kostenreduktion durch die Vermeidung von Überdeckungen zu einer Verbesserung des Servicelevels und kann somit zur Umsatzsteigerung beitragen. Bei steigendem Umsatz sinkt der Anteil der Personalkosten an den Gesamtkosten, was einer der größten Effekte eines WFM-Systems im Handel ist [138, S. 256–257].

Ein oft ungenutzter Hebel zur Steigerung der WFM-Agilität liegt in den meist sehr starren Schichtmodellen. In Abbildung 3.2 wird dieser Zusammenhang an konkreten Daten eines 3-Schicht-Betriebs veranschaulicht. Der Personalbedarf schwankt während der Woche sehr stark. Es gelingt mit den drei zur Verfügung stehenden Schichten jedoch nicht, Über- und Unterdeckungen im Personaleinsatz zu vermeiden. Überdeckungen führen zu Leerzeiten. Also Zeit, in der Mitarbeiter anwesend sind, aber keinen wertschöpfenden Beitrag leisten. Auf der anderen Seite müssen Unterdeckungen durch Überstunden oder durch den Einsatz von Leiharbeitskräften teuer ausgeglichen werden. U.U. drohen sogar Umsatzeinbußen, da Aufträge erst gar nicht angenommen werden können.

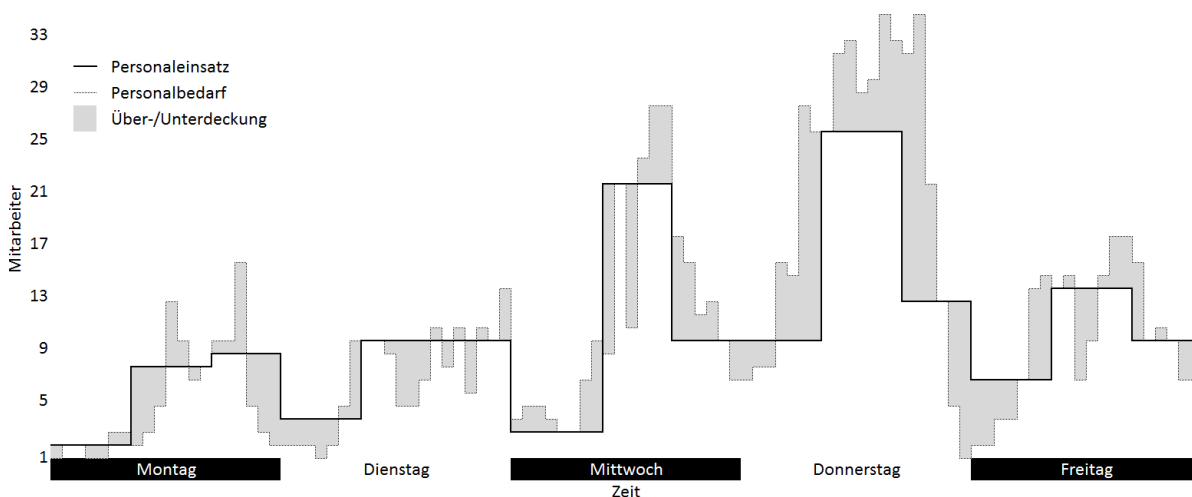


Abbildung 3.2: Starrer Personaleinsatz (drei Schichten) bei schwankendem Personalbedarf

Dieser Problemstellung kann mit der Einführung weiterer Arbeitszeitmodelle begegnet werden. In Abbildung 3.3 sieht man deutlich, wie der Personaleinsatz bedarfsorientiert gestaltet wird. Eine vollständige Angleichung des Personaleinsatzes an den Personalbedarf ist jedoch aufgrund diverser Rahmenbedingungen, wie Ruhezeiten, minimale/maximale Arbeitszeiten pro Tag und Grenzen der Zeitsalden, nicht möglich. Doch selbst mit diesen Restriktionen können durch die bessere Ausnutzung der Normalarbeitszeit der Mitarbeiter Über- und Unterdeckungen bei gleicher Arbeitsleistung drastisch reduziert werden (die Flächen unter den Kurven für den Personaleinsatz sind in beiden Abbildungen gleich groß). Welche Effekte sich daraus ergeben, liegt auf der Hand.

- Kostensenkung durch bessere Ausnutzung der Normalarbeitszeit der Belegschaft.

- Reduzierung von Überstunden und Leerzeiten.
- Verminderter Einsatz von Leiharbeitern und Aushilfen zur Peakabdeckung.
- Umsatzsteigerung und eine bessere Kundenzufriedenheit durch einen erhöhten Servicelevel.
- Erhöhte Mitarbeitermotivation durch Vermeidung von Lastspitzen.

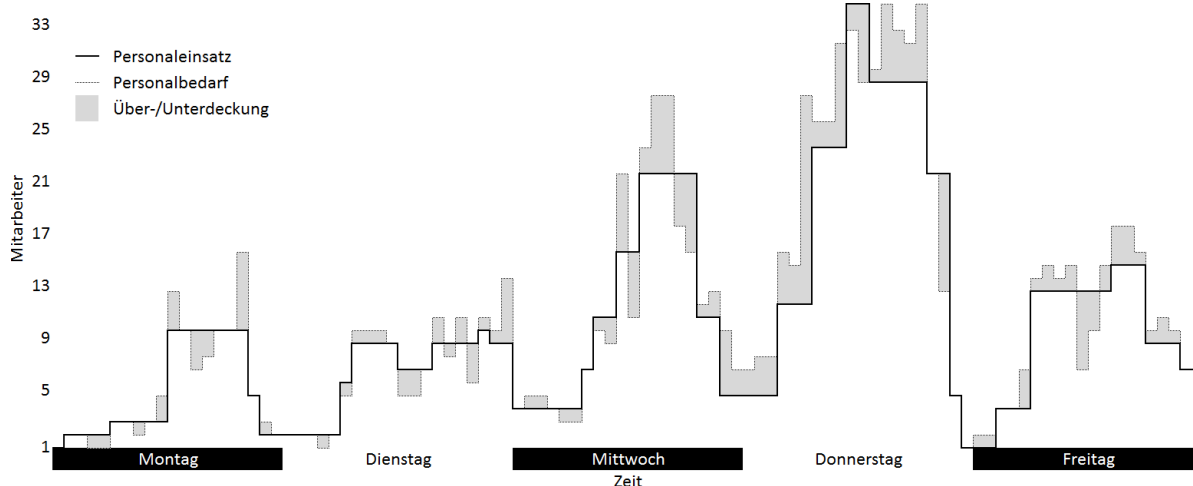


Abbildung 3.3: Flexibler Personaleinsatz (automatische Arbeitszeitmodellerstellung) bei schwankendem Personalbedarf

Mit einer Erhöhung der Anzahl möglicher Arbeitszeitmodelle kann selbst ein stark schwankender Personalbedarf kostengünstig abgedeckt werden. Selbstverständlich bedeutet diese Erhöhung einiges an Aufwand für die Erstellung und Pflege der Arbeitszeitmodelle. Per Hand oder mit einer Tabellenkalkulation ist dies nur sehr aufwändig zu bewältigen. Und an eine Erstellung optimaler Einsatzpläne ist ohne eine leistungsstarke Softwarelösung erst gar nicht zu denken. Besonders interessant sind Lösungen, die eine automatische Erzeugung von Arbeitszeitmodellen direkt in der Einsatzplanerstellung erlauben.

Automatische Arbeitszeitmodellerstellung meint folgendes: Man löst sich vollends von starren Schichten, da sie den anstehenden Personalbedarf nicht immer optimal decken. Der Planer gibt lediglich einige Regeln für die bedarfsgerechte Erstellung von Arbeitszeitmodellen an. Diese sind u.a. die minimale und maximale Dauer der Arbeitszeitmodelle sowie die Grenzen für deren Beginn und Ende. Bei der automatischen Erstellung des Einsatzplans werden individuelle Arbeitszeitmodelle für jeden Mitarbeiter generiert. Diese Arbeitszeitmodelle sind so gestaltet, dass sie den Personalbedarf bestmöglich abdecken. Selbstverständlich können bei der automatischen Planung neben den Regeln für die Arbeitszeitmodellerstellung z.B. auch Verfügbarkeiten, Mitarbeiterwünsche, Qualifikationen, Zeitsalden und Personalkosten berücksichtigt werden. Um den Nutzen greifbarer

zu machen, werden die Effekte durch eine automatische Arbeitszeitmodellerstellung an einem Praxisbeispiel erläutert (dies ist nicht das Praxisbeispiel aus dem Handel, an dem in dieser Arbeit Lösungsverfahren beurteilt werden).

In einem Unternehmen wird zwischen 6 und 19 Uhr gearbeitet. Zur Planung der 480 Mitarbeiter standen bisher fünf verschiedene Arbeitszeitmodelle zur Verfügung, womit man bereits recht flexibel war. Das Arbeitsvolumen unterliegt konjunkturellen sowie saisonalen Schwankungen. Zudem treten häufig starke untertägige Sprünge im Personalbedarf auf. Vor der Einführung des WFM-Systems wurde das Arbeitsvolumen nicht immer in der Normalarbeitszeit der Mitarbeiter geleistet, da der Personalbedarf an den Arbeitsplätzen zu stark schwankt. Häufig musste man Überstunden fahren und auf Leiharbeitskräfte zurückgreifen. Auch Leerzeiten waren ein Problem, die meist mit Alibitätigkeiten gefüllt wurden. Von einem WFM-System versprach man sich vor allem die Lösung dieser kostspieligen Probleme. Ein Vergleich wurde mit fünf, sechs sowie neun Arbeitszeitmodellen und mit der automatischen Arbeitszeitmodellerstellung angestellt. Alle Berechnungen wurden an einem kompletten Kalenderjahr vorgenommen, wobei eine kommerzielle Software (ATOSS Staff Efficiency Suite 5 [13, S. 991–1420]) mit ihrem konstruierenden Verfahren genutzt wurde. Der Fokus der Optimierung lag jeweils auf der Reduzierung der Über- und Unterdeckung im Personaleinsatz. Tendenziell werden automatische Arbeitszeitmodelle mit einer Länge von 7,5 Stunden erzeugt. Zur optimalen Bedarfsdeckung wurden in Ausnahmefällen auch Arbeitszeitmodelle mit einer minimalen Länge von 3 Stunden bis hin zu einer maximalen Länge von 9 Stunden zugelassen. In Tabelle 3.5 werden die Effekte einer gesteigerten Anzahl an Arbeitszeitmodellen veranschaulicht. Dabei muss angemerkt werden, dass es sich um die Ergebnisse einer Planung handelt. Tatsächlich kommt es in der Realität jedoch selbst bei der automatischen Arbeitszeitmodellerstellung zu Über- und Unterdeckungen, da sich die Auftragslage kurzfristig ändern oder Mitarbeiter erkranken können. Dennoch wird das Potenzial durch die Zahlen belegt.

3.3.2 Beschreibung der Problemstellung

Diese Problemstellung stammt aus einer Abteilung für Damenbekleidung eines Kaufhauses. Angenommen sei wie beim Logistikproblem eine Menge an Mitarbeitern $\mathcal{E} = \{1, \dots, E\}$, eine Menge an Arbeitsplätzen $\mathcal{W} = \{1, \dots, W\}$ und ein diskreter Zeithorizont \mathcal{T} mit dem Index $t = 0, \dots, T - 1$. Jede Periode t des Zeithorizonts hat eine Länge l_t , die größer als Null ist.

$$l_t > 0 \quad \forall t \in \mathcal{T} \quad (3.13)$$

Auch hier erfolgt die Zuweisung eines der 15 Mitarbeiter auf einen der beiden Arbeits-

Tabelle 3.5: Effekte durch die Ausweitung der Anzahl an Arbeitszeitmodellen

Arbeitszeitmodelle	Überdeckung (in h/Jahr)	Unterdeckung (in h/Jahr)	Summe der Abweichungen (in h/Jahr)
5 verschiedene Arbeitszeitmodelle	26.094	528	26.622
6 verschiedene Arbeitszeitmodelle	16.488	4.734	21.222
15 verschiedene Arbeitszeitmodelle	3.426	438	3.864
Automatische Arbeits- zeitmodellerstellung	0	0	0

plätze über die binäre Variable x_{ewt} .

$$x_{ewt} = \begin{cases} 1, & \text{wenn Mitarbeiter } e \text{ dem Arbeitsplatz } w \\ & \text{in Periode } t \text{ zugewiesen wurde} \\ 0, & \text{sonst} \end{cases} \quad (3.14)$$

Das Kaufhaus hat von Montag bis Sonnabend von 10 bis 20 Uhr geöffnet. Sonntags und an Feiertagen ist das Kaufhaus geschlossen. Weiterhin ist im Vorfeld bekannt, welcher Mitarbeiter wann Urlaub hat oder auf Fortbildung ist. Somit stehen die Verfügbarkeiten und Abwesenheiten der Mitarbeiter fest und werden mit der binären Variable a_{et} festgelegt.

$$a_{et} = \begin{cases} 1, & \text{wenn Mitarbeiter } e \text{ in Periode } t \text{ verfügbar ist} \\ 0, & \text{sonst} \end{cases} \quad (3.15)$$

Mitarbeiter werden auf zwei verschiedene Arbeitsplätze (Kasse und Verkauf) verplant, wobei sie so eingearbeitet werden, dass sie auf beiden Arbeitsplätzen arbeiten können. Qualifikationen werden bei der Planung demnach nicht berücksichtigt. Die Definition der Arbeitsplätze ist recht weit gefasst, sodass Nebentätigkeiten, wie Refill, Switchen oder Umbau im Personalbedarf der beiden Arbeitsplätze integriert werden. Der Personalbedarf d_{wt} ist für beide Arbeitsplätze in 1-Stunden-Intervallen gegeben. Er wurde von der Zentrale für jeden Arbeitsplatz aufgrund von Vergangenheitsdaten ermittelt. Als Bedarfstreiber (Vergangenheitsdaten) je Zeitintervall wurden Umsatzdaten und Anzahl Kassenbons für die Kassenarbeitsplätze sowie Umsatzdaten und Anzahl Kunden für die Verkaufsarbeitsplätze verwendet. Auch Feiertage, Brückentage sowie Werbeaktionen fanden bei der Bedarfsermittlung Berücksichtigung. Eine Minimal- und Maximalbesetzung

je Arbeitsplatz und Zeitintervall begrenzen die Ergebnisse der Personalbedarfsermittlung. Der Personalbedarf d_{wt} an Mitarbeitern pro Arbeitsplatz und Periode ist weit im Voraus bekannt darf nicht negativ sein.

$$d_{wt} \geq 0 \quad \forall w \in \mathcal{W} \text{ und } \forall t \in \mathcal{T} \quad (3.16)$$

Derzeit werden alle 15 Mitarbeiter mit MS EXCEL™ verplant. Der zuständige Floor-Manager ist mit der Einsatzplanung meist überlastet. Eine Vielzahl unterschiedlichster Faktoren müssen berücksichtigt werden. Besonders die starken Schwankungen im Personalbedarf lassen eine bedarfsorientierte Personaleinsatzplanung mit dem aktuellen Vorgehen nicht zu. Um das geschilderte Problem abzustellen, soll eine Einsatzplanung mit automatisch erstellten Arbeitszeitmodellen vollzogen werden, bei der auch untertägige Arbeitsplatzwechsel zulässig sind.

Bei dieser Art der Planung gibt der Planer lediglich die minimal und maximal zulässige Länge der Arbeitszeiten an. In der Planung werden dann Arbeitszeitmodelle generiert, die den Bedarf möglichst gut abdecken. Dabei werden Öffnungszeiten sowie die Verfügbarkeiten der Mitarbeiter und deren wöchentliche Sollarbeitszeit berücksichtigt. Aufgrund von Bedarfsschwankungen sind auch untertägige Arbeitsplatzwechsel zulässig. Zu den harten Nebenbedingungen gehört, dass Arbeitsplatzwechsel oder der Beginn bzw. das Ende eines Arbeitszeitmodells nur in 1-Stunden-Intervallen stattfinden darf. Ein Mitarbeiter kann in einer Periode t nur dann einem Arbeitsplatz w zugeordnet werden, wenn er auch tatsächlich verfügbar ist. Er kann auch immer nur einem Arbeitsplatz zur gleichen Zeit zugewiesen werden. Hierbei ist zu beachten, dass ein prinzipiell anwesender Mitarbeiter nicht zwingend einem Arbeitsplatz zugewiesen werden muss, da er zur Bedarfsdeckung nicht benötigt wird.

$$\sum_{w=1}^W x_{ewt} \leq a_{et} \quad \forall e \in \mathcal{E} \text{ und } \forall t \in \mathcal{T} \quad (3.17)$$

Es existieren auch weiche Nebenbedingungen, bei deren Verletzung Fehlerpunkte anfallen. Über ihre Höhe findet eine Gewichtung der Restriktionen statt. Besonders im Handel ist die Sicherstellung eines hohen Servicelevels für die Erreichung der Umsatzziele besonders wichtig. Daher sollen bei der Planerstellung Besetzungsabweichungen möglichst vermieden werden. Sobald eine Abweichung von der Besetzungsvorgabe d_{wt} auftritt, entstehen Fehlerpunkte P_d für die Dauer und Höhe der Fehlbesetzung entsprechend der Fehlerpunkthöhe. Dabei wird analog zum Logistikproblem zwischen verschiedenen Fehlertypen unterschieden: c_{do} bei Überdeckung, wenn der Bedarf $d_{wt} > 0$ ist, c_{dn} bei Unterdeckung, wenn der Bedarf $d_{wt} = 0$ ist sowie c_{du} bei einer Unterdeckung. Durch die höheren Fehlerpunkte bei c_{dn} wird die gegenseitige Unterstützung von Mitarbeitern gefördert.

$$P_d = \sum_{t=0}^{T-1} \sum_{w=1}^W (c_{do} + c_{dn} + c_{du}) l_t \left| \left(\sum_{e=1}^E x_{ewt} \right) - d_{wt} \right|, \quad (3.18)$$

mit:

- $c_{do} = 1$, wenn auf Arbeitsplatz w in Periode t eine Überdeckung im Personaleinsatz vorliegt mit $d_{wt} > 0$, sonst $c_{do} = 0$.
- $c_{dn} = 2$, wenn auf Arbeitsplatz w in Periode t eine Überdeckung im Personaleinsatz vorliegt mit $d_{wt} = 0$, sonst $c_{dn} = 0$.
- $c_{du} = 1$, wenn auf Arbeitsplatz w in Periode t eine Unterdeckung im Personaleinsatz vorliegt mit $d_{wt} > 0$, sonst $c_{du} = 0$.

Insgesamt existieren sechs verschiedene Arbeitsverträge, die sich in der zu leistenden wöchentlichen Sollarbeitszeit s_e von 10 bis 40 Stunden in einer normalen Woche unterscheiden. Es gibt also Vollzeit- und Teilzeitmitarbeiter sowie studentische Aushilfen. In Wochen mit Feiertagen wird die wöchentliche Sollarbeitszeit s_e um einen Faktor h_{Woche} reduziert. Ist z.B. der Montag in Woche eins ein arbeitsfreier Tag, so ist $h_1 = 5/6$. Die tatsächlich in einer Woche von einem Mitarbeiter geleistete Arbeitszeit i_e sollte nicht über der vertraglich fixierten wöchentlichen Sollarbeitszeit liegen. Jede zu viel geleistete Minute wird mit Fehlerpunkten c_w geahndet.

$$P_w = \sum_{Woche=1}^{52} \sum_{e=1}^E c_w (i_e - s_e * h_{Woche}), \quad (3.19)$$

mit:

- $c_w = 1$, wenn $i_e - s_e * h_{Woche} > 0$.
- $c_w = 0$, sonst.

Ein besonderes Augenmerk liegt auf den automatisch erstellten Arbeitszeitmodellen. Die tägliche Arbeitszeit eines Mitarbeiters soll nicht kürzer als drei Stunden und nicht länger als neun Stunden sein. Bei einem Verstoß gegen diese Regelungen fallen je Mitarbeiter und Tag jeweils Fehlerpunkten c_t an, die hier auf 1.000 gesetzt sind. Die Summe dieser Fehlerpunkte für den Planungszeitraum ist P_t . Zudem dürfen Arbeitszeitmodelle am Tag nicht unterbrochen sein. Jeder Verstoß führt je Mitarbeiter und Tag zu den Fehlerpunkten c_t , wobei c_t mit 10.000 bewertet ist. Die Summe aller Fehlerpunkte durch unterbrochene Arbeitszeitmodelle wird mit P_c bezeichnet.

Für eine optimale Bedarfsdeckung sind untertägige Arbeitsplatzwechsel zwar nötig – unnötige Wechsel sollen jedoch möglichst vermieden werden. Die Anzahl der Arbeitsplatzwechsel pro Mitarbeiter wird mit r_e bezeichnet. Jeder Arbeitsplatzwechsel wird mit Fehlerpunkten c_r bestraft, wobei c_r bei diesem Problem eins beträgt.

$$P_r = c_r \sum_{e=1}^E r_e \quad (3.20)$$

Die Zielfunktion, die es zu minimieren gilt, lautet:

$$\min P = P_d + P_w + P_t + P_c + P_r. \quad (3.21)$$

Für die geschilderte Problemstellung liegen historische Daten für das komplette Kalenderjahr 2006 vor, die anonymisiert wurden. Im praktischen Einsatz würde man lediglich die kommende Woche oder den kommenden Monat planen. Das komplette Jahr wird gewählt, um die Lösungsverfahren an einer möglichst komplexen Problemstellung zu testen.

3.3.3 Varianten der Problemstellung

Die in Kapitel 3.3.2 dargestellte Problemstellung für ein komplettes Jahr mit 2 Arbeitsplätzen und demzufolge untertägigen Arbeitsplatzwechseln wird nachfolgend als `Handel_2AP` bezeichnet. Diese Problemstellung umfasst 15 Mitarbeiter. Neben dem kompletten Jahr wird auch der Januar separat untersucht. Das zugehörige Problem trägt die Bezeichnung `Handel_2AP_Jan`.

Zwei weitere Problemvarianten werden aus `Handel_2AP` und `Handel_2AP_Jan` abgeleitet, indem nur der erste der beiden Arbeitsplätze berücksichtigt wird. Aus den Echt-daten ist bekannt, dass neun der 15 Mitarbeiter den ersten Arbeitsplatz als Stammarbeitsplatz haben. Somit ergeben sich zwei Problemstellungen, die als `Handel_1AP` und `Handel_1AP_Jan` bezeichnet werden. Sie umfassen neun Mitarbeiter und einen Arbeitsplatz mit dem entsprechenden Bedarf für ein Kalenderjahr bzw. den Januar. Da nur ein Arbeitsplatz existiert, gibt es auch keine untertägigen Arbeitsplatzwechsel. In Formel 3.21 ist P_r folglich nicht enthalten. Der Grund für die Erzeugung von `Handel_1AP` liegt darin, dass alle gefundenen Probleme in der Literatur (siehe Kapitel 3.3.5) immer nur einen Arbeitsplatz beinhalten. Zudem bestand die Möglichkeit, das in Kapitel 3.3.5 aufgeführte Lösungsverfahren von Sauer und Schumann [334] zu erproben, was die Modifikation von `Handel_2AP` und `Handel_2AP_Jan` erforderte.

Für die automatische Arbeitszeitmodellerstellung im Handel werden folglich vier Problemstellungen untersucht. Dabei ist die Erläuterung zur Abbildung der Problemstellungen, die einen Einfluss auf die Anzahl an Dimensionen hat, Kapitel 4.1 zu entnehmen. Nachfolgend sind die Rahmenbedingungen der vier Probleme in Tabelle 3.6 dargestellt.

Tabelle 3.6: Rahmenbedingungen der Handelsprobleme

Bezeichnung der Problemstellung	Anzahl Mitarbeiter	Anzahl Zeitintervalle	Anzahl Dimensionen
Handel_2AP	15	8.760	131.400
Handel_2AP_Jan	15	744	11.160
Handel_1AP	9	8.760	78.840
Handel_1AP_Jan	9	744	6.696

3.3.4 Einordnung der Problemstellung

Die Einordnung des Handelsproblems erfolgt in die von Ernst u.a. [114] dargestellten Klassen. Bzgl. des Bedarfs kann das Problem dem Flexible Demand zugeordnet werden. Der Bedarf wird aufgrund von Vergangenheitsdaten für 1-Stunden-Intervalle prognostiziert. Dabei werden Events, Öffnungszeiten und die Lage der Feiertage berücksichtigt.

Da keine Arbeitszeitmodelle vorgegeben sind, müssen diese in der Planung erstellt werden. Das Handelsproblem ist daher dem Shift Scheduling zuzuordnen. Bei den Problemen Handel_2AP und Handel_2AP_Jan muss zudem festgelegt werden, auf welchem der beiden Arbeitsplätze die Mitarbeiter arbeiten sollen. Sie werden daher zusätzlich dem Task Assignment zugeordnet.

3.3.5 Vergleichbare Problemstellungen in der Literatur

Ein Einsatzplanungsproblem aus dem Handel wird 2006 von Prüm [312] bearbeitet. In der Einsatzplanung findet gleichzeitig auch eine automatische Arbeitszeitmodellerstellung bei schwankendem, Personalbedarfen in 1-Stunden-Intervallen statt. Die Zuweisung von Mitarbeitern erfolgt unter Berücksichtigung von Verfügbarkeiten, vertraglichen Regelungen sowie aufgrund von Einschränkungen bei der minimalen und maximalen Dauer der täglichen Arbeitszeit. Prüm erprobt Simplex, Branch & Bound (B&B) sowie ein hybrides Verfahren. Das hybride Verfahren startet mit einem Simplex-Algorithmus, dessen Ergebnis mit zwei verschiedenen Heuristiken in eine ganzzahlige Form transformiert wird. Zur Transformation werden ALCV (Adjust Length Center Value) und ALCD (Adjust Length Center Demand) herangezogen. Für seine Untersuchungen verwendet er 20 Problemstellungen mit 4 bis 168 Zeitintervallen sowie 2 bis 576 Mitarbeitern. Die alleinige Anwendung von Simplex an der relaxierten Problemstellung führt meist recht schnell zu optimalen Ergebnissen. Die Ergebnisse sind aufgrund ihrer Reellwertigkeit aber unbrauchbar. B&B hingegen operiert auf ganzen Zahlen. Es liefert bei Prüm jedoch nur bei sehr kleinen Problemen die Optimallösung innerhalb von einer Stunde. Beim hybriden

Verfahren hingegen werden recht schnell Lösungen gefunden. Theoretisch können die Lösungen aber sehr schlecht sein. Im Durchschnitt sind sie 24% unter dem Optimum und nur gelegentlich optimal. ALCV und ALCD unterscheiden sich dabei nicht wesentlich voneinander. Unabhängig vom Lösungsverfahren wird das eigentliche Problem von Prüm jedoch sehr stark vereinfacht, um es als Lineares Programm darstellen und überhaupt in akzeptabler Zeit bearbeiten zu können. So gibt es z.B. nur einen Arbeitsplatz und somit auch keine Arbeitsplatzwechsel.

Sauer und Schumann stellen 2007 [334] ein konstruierendes Verfahren für ein ähnliches Personaleinsatzplanungsproblem im Handel vor. Der Personalbedarf steht in 15-Minuten-Intervallen zur Verfügung und auch Arbeitszeitmodelle werden automatisch erstellt, wobei der Planungshorizont maximal eine Woche a sieben Tage beträgt. Sauer und Schumann vollziehen die Planung für lediglich einen Arbeitsplatz unter Berücksichtigung von Öffnungszeiten, Feiertagen sowie Mitarbeiterverfügbarkeiten. Zudem ist es möglich, verschiedene Arbeitsverträge mit den entsprechenden Regelungen zur Arbeitszeit zu hinterlegen. Das Lösungsverfahren wurde als Modul ActivePeP in das WFM-System des Softwareherstellers Torex Retail Holdings Limited [376] implementiert. Derzeit wird diese Software von zahlreichen Einzelhändlern erfolgreich eingesetzt. Da ActivePeP als interaktives Einsatzplanungstool entwickelt wurde, hat man einen Greedy-Algorithmus aus zwei Gründen gewählt:

1. Expertenwissen kann einfach in den Algorithmus integriert werden, um so die Planungsqualität zu erhöhen.
2. Der Greedy-Algorithmus erstellt sehr schnell eine Lösung.

An den Problemen `Handel_1AP` und `Handel_1AP_Jan` wird in dieser Arbeit u.a. auch das konstruierende Verfahren von Sauer und Schumann beurteilt. Für eine genaue Beschreibung des konstruierenden Verfahrens sei auf Kapitel 4.6.3.1 verwiesen.

2010 veröffentlicht Stolletz [358] seine Arbeit zur Arbeitszeitmodellerstellung für Bodenpersonal eines Flughafens. Der Planungshorizont der drei untersuchten Problemstellungen mit 13 bis 65 Mitarbeitern beträgt 15 Tage, die in 30 Minuten-Intervalle zerlegt sind. Da jedoch nicht zu jedem Zeitpunkt gearbeitet wird, beinhalten die Problemstellungen lediglich 1.512 bis 7.650 Zeitintervalle a 30 Minuten. Zur Lösung nutzt Stolletz CPLEX 11.01. Auf einem PC mit 1,8 GHz und 2 GB RAM benötigt CPLEX je nach Problemstellung zwischen 23 und 900 Sekunden. Stolletz untersucht jedoch nur Problemstellungen ohne untertägige Arbeitsplatzwechsel, sodass es wie bei Prüm oder Sauer und Schumann nur um die Erstellung von sehr flexiblen Arbeitszeitmodellen geht.

Die zu untersuchenden Handelsprobleme dieser Arbeit sind deutlich größer und erheblich komplexer als die Probleme von Prüm oder von Sauer und Schumann. Prüm erstellt nur Pläne für einzelne Tage und im Höchstfall für eine Woche. Sauer und Schumann können im Maximum nur eine Woche planen. Zwar untersucht Stolletz recht umfang-

reich dimensionierte Problemstellungen. Diese sind jedoch immer auf exakt 15 Tage beschränkt. Im hier untersuchten Handelsproblem wird hingegen ein komplettes Jahr bzw. ein kompletter Monat herangezogen. Zudem ist bei den Problemen `Handel_2AP` und `Handel_2AP_Jan` die Komplexität durch die Berücksichtigung zweier Arbeitsplätze erheblich größer, da im Zuge der Einsatzplanung und automatischen Arbeitszeitmodell-erstellung auch untertägige Arbeitsplatzwechsel zu erzeugen sind, deren Anzahl jedoch gleichzeitig minimiert werden soll. Prüm, Sauer und Schumann sowie Stolletz können lediglich einen Arbeitsplatz berücksichtigen. Weiterhin verwenden Prüm und Stolletz nur lineare Nebenbedingungen und Zielfunktionen.

3.4 Kombinierte Personaleinsatzplanung und Tourenplanung bei Technikern

In diesem Kapitel wird die dritte Problemstellung dieser Arbeit analog zum Logistik- und Handelsproblem erläutert. Zunächst wird auf die Kombination von Touren- und Personaleinsatzplanung eingegangen. Die konkrete Problemstellung, an der bereits viele Lösungsverfahren beurteilt wurden, wird in Kapitel 3.4.2 erläutert. Von dieser Problemstellung existiert eine Vielzahl an Varianten, die in Kapitel 3.4.3 beschrieben werden. Anschließend folgt eine Klassifikation des Problems. Mit Ausführungen zu bisherigen Arbeiten an dieser und ähnlichen Problemstellungen endet dieses Kapitel.

3.4.1 Kombination von Touren- und Personaleinsatzplanung

Die Kombination von Touren- und Personaleinsatzplanung tritt in den verschiedensten Anwendungsfällen auf. Z.B. beim Technikereinsatz im Außendienst, beim Einsatz von Pflegekräften, bei Speditionen oder bei Sicherheitsdiensten. Jedes Anwendungsfeld hat besondere Eigenheiten. So sollen in der Pflege Kunden möglichst immer von den gleichen Mitarbeitern betreut werden oder bei Speditionen sind z.T. Rahmentouren vorgegeben. Besonders beim Technikereinsatz werden komplexe Anforderungen an die Planung gestellt. Hunderte von Qualifikationen, Zeitfenster für Aufträge, begrenzte Mitarbeiterverfügbarkeiten, individuelle Leistungskennziffern und ein komplexes Straßennetz sind nur einige der Rahmenbedingungen, die es zu berücksichtigen gilt.

Wie zuvor geschildert, ist das Anwendungsspektrum der Kombination von Touren- und Personaleinsatzplanung recht facettenreich. Grundsätzlich liegen die Problemstellungen Tourenplanung und Personaleinsatzplanung recht nahe beieinander. So überführen Beck u.a. [28] das Vehicle Routing Problems (VRP) in ein Job-Shop Scheduling Problem (JSP), wobei fünf Merkmale identifiziert wurden, die Routing- und Scheduling-Probleme unterscheiden (man geht jedoch davon aus, dass es noch weitere geben könnte).

Die Lösung der kombinierten Touren- und Personaleinsatzplanung kann in zwei Schritten ablaufen. Dies stellt noch heute ein gängiges Vorgehen dar. Nachfolgend werden die zwei Lösungsschritte skizziert.

1. Tourenplanung: Aufgrund der Auftragslage (aus Serviceintervallen, Bestellungen, Notrufen, etc.) werden Touren gebildet. Bei der Tourenplanung soll eine möglichst gute Zuordnung von Fahrzeugen zu Aufträgen inkl. optimaler Reihenfolge der Auftragsstandorte gefunden werden. Es gibt zwei Aspekte bei der Tourenplanung. Die Clusterung gibt an, welche Aufträge zu einer Tour zusammengefasst werden, und das Routing definiert, in welcher Reihenfolge die Punkte innerhalb einer Tour bedient werden. Zielsetzung einer Tourenplanung ist z. B. die Anzahl der eingesetzten Fahrzeuge, die zurückgelegte Strecke, die Einsatzzeit oder eine komplexeren Kostenfunktion zu minimieren.

Bei der Tourenplanung steht jedoch das Fahrzeug und weniger der Mitarbeiter im Vordergrund. Daher werden häufig keine Zeitsalden berücksichtigt und es ist auch meist nicht bekannt, ob zu einem bestimmten Zeitpunkt auch wirklich genügend Mitarbeiter zur Verfügung stehen. Es ist auch nicht sicher gestellt, dass die gebildeten Touren optimal in Bezug auf Zeitsalden, gesetzliche Regelungen und geplante An-/Abwesenheiten sind.

Durch die Tourenplanung im ersten Schritt wird der Personalbedarf für einen bestimmten Zeitraum festgelegt. Dieser Personalbedarf ist eine Eingangsgröße für den zweiten Schritt – die Personaleinsatzplanung.

2. Personaleinsatzplanung: Der Personalbedarf ist für einen bestimmten Zeitraum aus Schritt eins – der Tourenplanung – gegeben und wird in das System zur Personaleinsatzplanung übernommen. Dabei sind die zuvor ermittelten Touren festgeschrieben und werden im weiteren Planungsprozess nicht mehr geändert.

Beim zuvor dargestellten Lösungsansatz, bei dem die Optimierung in zwei getrennten Schritten erfolgt, werden die ermittelten Touren im zweiten Schritt nicht mehr verändert. Hieraus ergeben sich einige Probleme, die mit dem bisherigen Lösungsansatz nicht zufriedenstellend gelöst werden können. Selbst Rücksprünge von Schritt zwei zu Schritt eins führen nicht zwangsläufig zu optimalen Ergebnissen. So kann es beispielsweise sein, dass eine Tour in Bezug auf wenig zu fahrende Kilometer, wenig eingesetzte Fahrzeuge und geringe Kosten optimal ist. Bei der Personaleinsatzplanung stellt sich jedoch heraus, dass vier Mitarbeiter in der kommenden Woche stark Überstunden aufbauen und der Rest nicht voll ausgelastet ist. Es macht also Sinn, Touren- und Personaleinsatzplanung in einem Planungsschritt zu bearbeiten. Im folgenden Kapitel wird eine derartige Problemstellung erläutert.

3.4.2 Beschreibung der Problemstellung

Diese Problemstellung stammt von der British Telecom und umfasst ein Planungsszenario, in dem Techniker eine Vielzahl an räumlich verteilten Aufträgen erledigen müssen (Echtdaten sind in [44] zu finden). Die erste Softwarelösung, die die British Telecom in der Praxis für vollautomatisches WFM einsetzte hieß Work Manager. Mit dem System konnten bis zu 150.000 Aufträge auf 20.000 Techniker verplant werden. Umfassende Beschreibungen des Work Managers sind in [136] und [242] zu finden. Herzstück des Work Managers ist der Dynamic Scheduler [239], [240], [241]. Ursprünglich war der Dynamic Scheduler nur in der Lage, sehr kurzfristige Planungen zu vollziehen, wobei keine Qualifikationen berücksichtigt wurden. Die Ergebnisse waren äußerst schnell verfügbar und optimal – jedoch war eine weit in die Zukunft gerichtete Planung mit individuellen Qualifikationen der Mitarbeiter nicht möglich. Daher wurden der Dynamic Scheduler und der Work Manager überarbeitet. Sie haben nun den in Abbildung 3.4 dargestellten Aufbau. Inzwischen wird die Software ARMS (Automated Resource Management System) eingesetzt, die in Java entwickelt wurde. ARMS beinhaltet neben dem Dynamic Scheduler zusätzlich eine Personalbedarfsermittlung [297], [400]. Eine Beschreibung der Anforderungen an den Technikereinsatz bei der British Telecom ist in den Arbeiten von Kern u.a. [209] sowie von Azarmi und Smith [19] zu finden. Ähnliche Anforderungen sind auch beim Technikereinsatz der Firma Ericsson dargestellt [10].

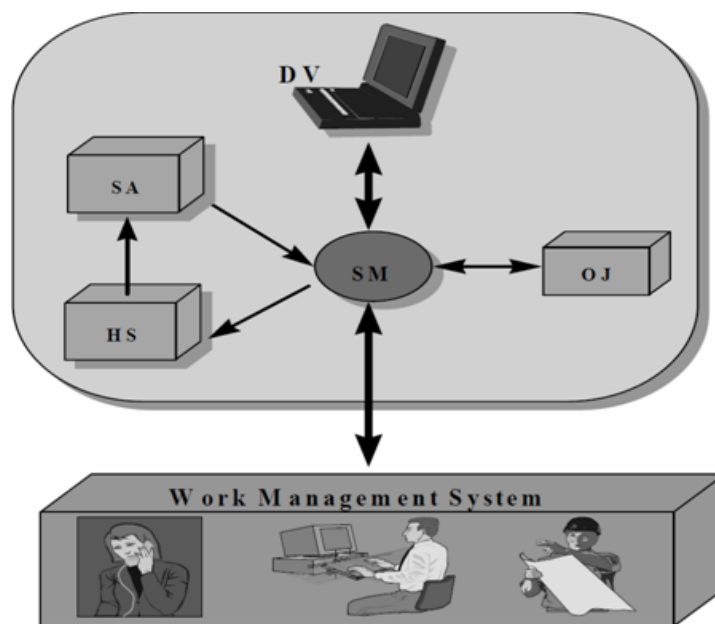


Abbildung 3.4: Aufbau des Dynamic Schedulers [242]

Die 5 Module des Dynamic Schedulers haben folgende Aufgabe:

- SM (Scheduler Manager): Zentrale Komponente, die die weiteren Module ansteuert und in regelmäßigen Zeitabständen Informationen vom Work Management System holt.
- HS (Heuristic Search): Komplexe und schwierige Aufgaben werden hier zugewiesen. Das Modul erzeugt einen vorläufigen Einsatzplan.
- SA (Simulated Annealing): Der vorläufige Einsatzplan wird mit allen restlichen, nun sehr einfach zuzuweisenden Aufgaben gefüllt.
- OJ (Optimise Job): Hier erfolgt eine Prüfung von Nebenbedingungen und ggf. eine Reparatur sowie eine Optimierung durch veränderte Bearbeitungsreihenfolgen der Aufträge.
- DV (Data Visualiser): Visualisierung der für den Planer relevanten Informationen.

Die Problemstellung, die in dieser Arbeit untersucht wird, stammt aus einer einzelnen Region der British Telecom. Sie umfasst daher lediglich 250 Aufträge $\mathcal{J} = \{1, \dots, J\}$ und 118 Techniker $\mathcal{E} = \{1, \dots, E\}$. Die Untersuchung der eigenen Lösungsverfahren an dieser einzelnen Region erfolgt vor dem Hintergrund, dass bereits sehr viele Verfahren anderer Autoren dort erprobt wurden. Weitere und größere Regionen werden daher nicht herangezogen. Das hier verwendete Verhältnis aus Aufträgen und Technikern ist nicht allgemeingültig, da in der Realität auch häufig Aufträge existieren, die mehrtägige Bearbeitungszeiten haben [18, S. 82]. Für die Lösung dieser Problemstellung würden im Work Manager die Module SM, HS, SA, OJ in Abbildung 3.4 zum Einsatz kommen. In der Problemstellung sind die Arbeitszeitmodelle der Techniker vorgegeben. Die Arbeitszeitmodelle sollen in der Planung nicht verändert werden. Lediglich der Aufbau von Überstunden ist möglich. Der Planungshorizont beträgt einen Tag.

Mit Hilfe der binären Variable x_{ejk} wird einem Mitarbeiter e ein Auftrag j in der Reihenfolgeposition $\mathcal{K} = \{1, \dots, K\}$ zugewiesen. Die Reihenfolge der abzuarbeitenden Aufträge bei einem Mitarbeiter ist wichtig, da sich so sein zu fahrender Wege ergibt. Da einem Mitarbeiter theoretisch alle 250 Aufträge zugeordnet werden können, ist k im Maximum 250.

$$x_{ejk} = \begin{cases} 1, & \text{wenn Mitarbeiter } e \text{ dem Auftrag } j \\ & \text{in Position } k \text{ zugewiesen wurde} \\ 0, & \text{sonst} \end{cases} \quad (3.22)$$

Ein Auftrag j kann immer nur von einem Techniker e allein ausgeführt werden, sodass eine Beschleunigung der Ausführungszeit durch eine gegenseitige Hilfestellung nicht mög-

lich ist. Zudem darf kein Wechsel des Technikers bei der Auftragsbearbeitung erfolgen.

$$\sum_{e=1}^E \sum_{k=1}^K x_{ejk} = 1 \quad \forall j \in \mathcal{J} \quad (3.23)$$

Jeder Auftrag umfasst 5 Tupel: Auftragsnummer j , X-Koordinate auf der Karte, Y-Koordinate auf der Karte, Dauer d_j und Auftragstyp. Bei einer Zuweisung kann es vorkommen, dass ein Auftrag gar nicht ausgeführt werden kann, weil der entsprechende Techniker sonst die maximale Länge seiner Arbeitszeit überschreiten würde. Um Aufträge kenntlich zu machen, die aufgrund ihrer Zuweisung zu einem Techniker nicht ausgeführt werden, wird die binäre Variable s_{ejk} verwendet. Sie gibt an, ob der Auftrag j aufgrund der aktuellen Zuweisung zu einem Techniker e in k ausgeführt werden kann oder nicht. Auf die Einflussgrößen zur Festlegung von s_{ejk} wird später konkret eingegangen. Klar ist jedoch bereits hier, dass s_{ejk} bereits immer dann null ist, wenn x_{ejk} auch null ist – der Auftrag j dem Techniker e in k nicht zugeordnet wurde.

$$s_{ejk} = \begin{cases} 1, & \text{wenn Auftrag } j \text{ ausgeführt werden kann und } x_{ejk} = 1 \\ 0, & \text{sonst} \end{cases} \quad (3.24)$$

Ebenso wie jeder Auftrag wird auch jeder Techniker durch mehrere Tupel charakterisiert: Technikernummer e , Uhrzeit für Arbeitsbeginn, Uhrzeit für Arbeitsende, max. Überstunden o_e , Erfahrung r_e , Qualifikationen q_{ej} , X-Koordinate seines Servicecenters, Y-Koordinate seines Servicecenters.

Zu beachten ist, dass ein Techniker e nur dann einem Auftrag zugewiesen werden darf, wenn er hierzu qualifiziert ist. Dies ist eine harte Nebenbedingung. Das Vorhandensein einer Qualifikation wird mit der binären Variable q_{ej} angezeigt. Die Anzahl der für einen Auftrag qualifizierten Techniker kann von nur einem bis zu 107 reichen.

$$q_{ej} = \begin{cases} 1, & \text{Techniker } e \text{ ist für Auftrag } j \text{ qualifiziert} \\ 0, & \text{sonst} \end{cases} \quad (3.25)$$

Die entsprechende Qualifikationsbedingung ist nachfolgend dargestellt.

$$x_{ejk} - q_{ej} \leq 0 \quad \forall e \in \mathcal{E}, \forall j \in \mathcal{J} \text{ und } \forall k \in \mathcal{K} \quad (3.26)$$

Insgesamt existieren elf Servicecenter. In Abbildung 3.5 sind die Positionen der Aufträge und der Servicecenter dargestellt. Die Zahlen rechts neben den Servicecenternummern stehen für die Anzahl der jeweils fest zugewiesenen Techniker.

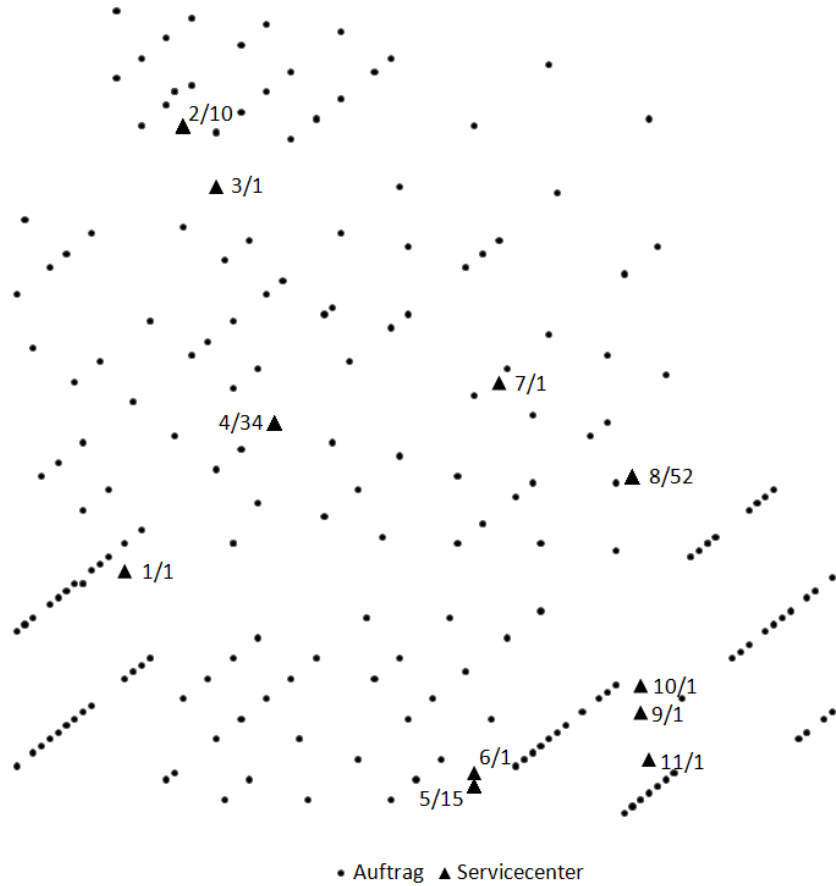


Abbildung 3.5: Position der Aufträge und Servicecenter [217, S. 14] (Zahlen: Servicecenter-ID/Anzahl Techniker)

Tabelle 3.7 zeigt für jedes Servicecenter die Anzahl der dort stationierten Techniker sowie die verfügbare Kapazität, die aus der Anzahl an Technikern mal 480 Minuten (acht Stunden) resultiert.

Mit Hilfe der X- und Y-Koordinaten von Aufträgen und Servicecentern kann die Fahrzeit l_{ek} eines Technikers e zu seinem Reiseziel in k ermittelt werden. In [383] ist diese Formel fehlerhaft. Hierbei handelt es sich um einen Tippfehler. Denn alle zugänglichen Ergebnisse wurden überprüft und ihre Richtigkeit kann bestätigt werden. Die korrekte Formel für l_{ek} lautet $\forall e \in \mathcal{E}, \forall j \in \mathcal{J}$ und $\forall k \in \mathcal{K}$ wie folgt.

$$l_{ek}((x_1, y_1), (x_2, y_2)) = \begin{cases} \frac{\frac{1}{2} * \Delta x + \Delta y}{8}, & \text{wenn } \Delta x > \Delta y \\ \frac{\frac{1}{2} * \Delta y + \Delta x}{8}, & \text{sonst} \end{cases} \quad (3.27)$$

Es müssen $\forall e \in E, \forall j \in J$ und $\forall k \in K$ lediglich die X- und Y-Koordinaten (hier: x_1, y_1 sowie x_2, y_2) der entsprechenden Aufträge oder Servicecenter in Formel 3.27 eingesetzt werden. Dabei sind nur Fahrten zu einem anderen Auftrag oder einem Servicecenter

Tabelle 3.7: Servicecenter und ihre Kapazitäten [217, S. 18]

Servicecenter-ID	1, 3, 6, 7, 9, 10, 11	2	4	5	8
Anzahl Techniker	1	10	34	15	52
Kapazität in Minuten	480	4.800	16.320	7.200	24.960

zulässig. Fahrten zwischen Servicecentern sind nicht erlaubt. Jeder Techniker startet bei Arbeitsbeginn von seinem Servicecenter und muss dieses nach Beendigung des letzten ausführbaren Auftrags innerhalb der Arbeitszeit wieder erreichen. All dies sind harte Nebenbedingungen. Es gelten also folgende Regeln für die Nutzung von Formel 3.27:

- Arbeitsbeginn: Fahrt vom Servicecenter des Technikers e zum ersten Auftrag j in k , wenn $k = 1$, $x_{ejk} = 1$ und $s_{ejk} = 1$.
- Folgeauftrag: Fahrt von einem Auftrag in $k - 1$ zum nächsten Auftrag j in k , wenn $k > 1$, x_{ejk-1} und $x_{ejk} = 1$ sowie $s_{ejk} = 1$.
- Arbeitsende: Fahrt von einem Auftrag in $k - 1$ zurück zum Servicecenter des Technikers e in k , wenn $k > 1$, $x_{ejk-1} = 1$, $x_{ejk} = 0$ und $s_{ejk-1} = 1$.
- Arbeitsende mit nicht ausgeführten Aufträgen: Fahrt von einem Auftrag in $k - 1$ zurück zum Servicecenter des Technikers e in k , wenn $k > 1$, x_{ejk-1} und $x_{ejk} = 1$ sowie $s_{ejk} = 0$.
- Sonst: Techniker e ist in k untätig und Formel 3.27 findet keine Anwendung. Daher ist $l_{ek} = 0$.

Mit Hilfe der Erfahrung r_e eines Technikers e und der durchschnittlichen Dauer d_j eines Auftrags j kann die tatsächliche Bearbeitungszeit rd_j des Auftrags j berechnet werden. Die Dauer d_j eines Auftrags j kann zwischen 10 und 514 Minuten betragen. In Formel 3.28 ist ihre Berechnung dargestellt. Hat ein Techniker z.B. die Erfahrung acht und dauert ein Auftrag durchschnittlich 20 Minuten, so beträgt die tatsächliche Bearbeitungszeit rd_j nur noch 16 Minuten.

$$rd_j = d_j * \frac{r_e}{10} \quad (3.28)$$

Wie bereits erwähnt, wird Aufträgen ein Typ zugeordnet, der Einfluss auf den Zeitpunkt der Ausführung hat. Unter einem Auftragsstyp ist zu verstehen, wann der Auftrag ausgeführt werden soll bzw. welche Priorität er hat. Diese Vorgabe ist eine harte Nebenbedingung, die es zwingend einzuhalten gilt. Es werden fünf Typen unterschieden:

- Erster: Der Auftrag muss beim jeweiligen Techniker der erste Auftrag des Ta-

3.4 Kombinierte Personaleinsatzplanung und Tourenplanung bei Technikern

ges sein. Ein Techniker kann immer nur maximal einen Auftrag vom Typ „Erster“ haben. Eine Zuweisung zu einem Techniker ist also nur zulässig, wenn $k = 1$ ist.

- Letzter: Der Auftrag muss beim jeweiligen Techniker der letzte Auftrag des Tages sein. Ein Techniker kann immer nur maximal einen Auftrag vom Typ „Letzter“ haben. Hier ist eine Zuweisung eines Auftrags zum Zeitpunkt k nur dann zulässig, wenn $x_{ejk} = 1$ und $x_{ejk+1} = 0$ sind oder x_{ejk} sowie $x_{ejk+1} = 1$ und $s_{ejk+1} = 0$ sind.
- Vormittags: Der Beginn des Auftrags muss vor 12 Uhr liegen.
- Nachmittags: Der Beginn des Auftrags muss nach 12 Uhr liegen.
- Keine Vorgabe: Hinsichtlich des Beginns und der Reihenfolge gibt es keine Vorgaben.

Aus der Regelung, dass ein Auftrag am Nachmittag erst nach 12 Uhr begonnen werden darf, ergibt sich ggf., dass ein Techniker e zu diesem Auftrag fährt und dort warten muss. Er ist also zu früh am Einsatzort. Die Summe der Wartezeit je Techniker e wird mit w_e bezeichnet.

Jeder Techniker kann für acht Stunden eingesetzt werden. Hierzu sind zwei verschiedene Arbeitszeitmodelle vorgegeben, die den Technikern im Vorfeld zugewiesen wurden. Arbeitsbeginn ist entweder 8.00 Uhr oder 8.30 Uhr, sodass das Arbeitsende 16.00 Uhr oder 16.30 Uhr ist. Es ist nicht vorgeschrieben, dass ein Techniker die vollen acht Stunden mit Reisezeit, dem Warten auf den Auftragsbeginn und der Erledigung von Aufträgen verbringen muss. Es besteht sogar die Möglichkeit, dass ein Techniker keinen Auftrag im Planungszeitraum übernimmt.

Arbeitsbeginn und Arbeitsende des Arbeitszeitmodells eines jeden Technikers sind fest vorgegeben und dürfen nicht verändert werden. Eine Ausweitung der Arbeitszeit ist lediglich im Rahmen der maximal zulässigen Überstundengrenze am Ende der acht Stunden möglich. Für jeden Techniker e wird diese Grenze o_e individuell vorgegeben. Sie kann maximal 60 Minuten betragen und darf nicht überschritten werden. Die tatsächlich geplanten Überstunden je Techniker e werden mit co_e bezeichnet. Für co_e gilt demnach die folgende Formel. Dabei wird deutlich, dass ein Techniker nicht ausführbare Aufträge ($s_{ejk} = 0$) zwar auf seinem Plan haben kann ($x_{ejk} = 1$), diese aber nicht berücksichtigt. Er würde andernfalls seine Arbeitszeit (inkl. möglicher Überstunden) überschreiten. Bei einem Techniker e mit $s_{ejk} = 1$ nimmt die binäre Variable s_{ejk} so lange den Wert eins an, bis eine Überschreitung der maximalen Arbeitszeit droht. Alle Folgeaufträge des Technikers können dann ebenfalls nicht ausgeführt werden – er muss zu seinem Servicecenter

zurück kehren (Arbeitsende mit nicht ausgeführten Aufträgen).

$$co_e = \begin{cases} -8 + \sum_{j=1}^J \sum_{k=1}^K s_{ejk} * x_{ejk} * (rd_j + l_{ek} + w_e) & \forall e \in \mathcal{E}, \text{ mit } co_e \leq o_e, \\ \text{wenn } \sum_{j=1}^J \sum_{k=1}^K s_{ejk} * x_{ejk} * (rd_j + l_{ek} + w_e) > 8 \\ 0, & \text{sonst} \end{cases} \quad (3.29)$$

Aufgrund der zuvor genannten Rahmenbedingungen und Restriktionen (Bearbeitungsdauer, Fahrtzeit, Prioritäten, Arbeitszeit, etc.) kann es möglich sein, dass nicht alle Aufträge im Planungszeitraum erledigt werden können. In der Zielfunktion soll das nicht erledigte Arbeitsvolumen minimiert werden. Hierzu wird auf die durchschnittliche Bearbeitungsdauer d_j eines Auftrags j ein Strafwert f addiert, der für dieses Problem auf 60 gesetzt wird. Diese Summe wird für jeden Auftrag j mit dem binären Wert von s_{ejk} multipliziert, der angibt, ob ein Auftrag ausgeführt wird oder nicht. Zudem sollen die Fahrtzeit und die Überstunden möglichst gering gehalten werden. Dies ist nahe liegend, da Fahrtzeit keine wertschöpfende Tätigkeit ist und Ressourcen verbraucht. Überstunden co_e führen zu Zuschlägen und sollen ebenfalls möglichst vermieden werden. Die gesamten Strafpunkte c , die es zu minimieren gilt, ergeben sich wie folgt.

$$\min c = \sum_{e=1}^E \sum_{k=1}^K l_{ek} + \sum_{e=1}^E co_e^2 + \sum_{j=1}^J (d_j + f) * s_{ejk} \quad (3.30)$$

3.4.3 Varianten der Problemstellung

Die zuvor in Kapitel 3.4.2 beschriebene Problemstellung wird nachfolgend als BT_250-118 bezeichnet. Sie wurde in diversen Arbeiten (siehe Kapitel 3.4.5) sehr individuell modifiziert. Die eigenen Experimente in dieser Arbeit werden an BT_Mod-250-118 durchgeführt, da an dieser Problemstellung die meisten Lösungsverfahren erprobt wurden und somit eine möglichst umfassende Vergleichbarkeit erzielt wird. Bei den eigenen Untersuchungen erfolgt die Abbildung der Problemstellung analog zum Vorgehen beim Logistik- und Handelsproblem. Daher hat BT_Mod-250-118 29.500 Dimensionen (alle weiteren Varianten der British Telecom-Probleme bei gleichartiger Abbildung auch). Einige Autoren nutzen ein anderes Vorgehen zur Abbildung des Problems. Umfassende Erläuterungen hierzu können Kapitel 4.1 entnommen werden.

Die verschiedenen Probleme, die von anderen Autoren ausgehend von BT_250-118 erstellt wurden, sind nachfolgend dargestellt. Sie unterscheiden sich von der originalen Problemstellung nur in den angegebenen Punkten – haben also immer 118 Techniker und 250 Aufträge.

- BT_250-118:

- Originale Problemstellung, die zuvor in Kapitel 3.4.2 beschrieben wurde.
- BT_Mod-250-118:
 - Überstunden sind für keinen Techniker erlaubt ($o_e = 0$).
 - Der Auftragsstyp „Erster“ zählt als Auftragsstyp „Vormittag“.
 - Der Auftragsstyp „Letzter“ zählt als Auftragsstyp „Nachmittag“.
- BT_Rel-250-118:
 - Überstunden sind für keinen Techniker erlaubt ($o_e = 0$).
- BT_Tsang-250-118:
 - Überstunden sind erlaubt aber auf maximal 10 Minuten je Techniker begrenzt ($o_e = 10$).
 - Der Auftragsstyp „Erster“ zählt als Auftragsstyp „Vormittag“.
 - Der Auftragsstyp „Letzter“ zählt als Auftragsstyp „Nachmittag“.
- BT_Bristol-250-118:
 - Überstunden sind für keinen Techniker erlaubt ($o_e = 0$).
 - Alle Aufträge vom Auftragsstyp „Nachmittag“, die länger als 260 Minuten dauern, werden in den Auftragsstyp „Letzter“ umgewandelt.

3.4.4 Einordnung der Problemstellung

Zunächst wird das British Telecom-Problem entsprechend der Klassifikation von Ernst u.a. [114] eingeordnet. Der Bedarf ergibt sich aufgrund von auszuführenden Aufträgen. Diese Aufgaben haben einen frühesten Beginn, eine Dauer und ein spätestes Ende. Es handelt sich daher um Task Based Demand. Die auszuführenden Aufträge werden in der Planung Mitarbeitern zugewiesen, wobei Qualifikationen berücksichtigt werden. Zudem wurden die Mitarbeiter zuvor Arbeitszeitmodellen zugewiesen, sodass ihre An- und Abwesenheiten bekannt sind. Das British Telecom-Problem gehört daher dem Task Assignment an.

Zusätzlich zur Klassifikation von Ernst u.a. für die Personaleinsatzplanung, kann das Problem auch im Kontext des Traveling Salesman Problems eingeordnet werden. Azarmi und Abdul-Hameed [18] ordnen es in die Klasse der multi Time Constraint Traveling Salesman Probleme (multi-TCTSP) ein. Ein Techniker muss eine Reihe von Orten in der kürzesten Reihenfolge anfahren, wobei er wieder zur Startposition zurückkehren muss. Hierbei müssen die Arbeitszeiten der Techniker eingehalten werden. Da mehrere Techniker vorhanden sind und jede Tour eines Technikers ein TCTSP ist, wird von einem multi-TCTSP gesprochen. Die Einordnung von Azarmi und Abdul-Hameed ist nicht vollständig. Denn es gibt Restriktionen bzgl. zeitlicher Nebenbedingungen der Aufträge. Daher handelt es sich zusätzlich zu den Ausführungen von Azarmi und Abdul-Hameed [18] um ein multi-TCTSPTW (TW = Time Windows). Da auch Qualifikationen berücksichtigt werden, handelt es sich weiterhin um ein multi-SDTCTSPTW (SD = Site Dependent). Schlussendlich existieren mehrere Servicecenter, sodass die Problemstellung als multi-MDSDTCTSPTW (MD = Multi-Depot) bezeichnet werden kann.

3.4.5 Vergleichbare Problemstellungen in der Literatur

An der zuvor dargestellten Problemstellung der British Telecom wurden in der Vergangenheit zahlreiche Lösungsverfahren erprobt. Diese werden in Kapitel 4 ausführlich beschrieben. Ihre Resultate werden in Kapitel 6 dargestellt. Nachfolgend sind die Verfahren in Tabelle 3.8 lediglich benannt. Für eine Erläuterung der Abkürzungen sei auf die entsprechenden Beschreibungen der Lösungsverfahren oder auf das Abkürzungsverzeichnis verwiesen. Zusammenfassend kann gesagt werden, dass Fast Guided Local Search (Fast GLS) von Tsang und Voudouris [383] die besten Resultate der in Tabelle 3.8 aufgeführten Lösungsverfahren liefert. Für weiterführende Ausführungen zu den Lösungsverfahren sei wie bereits erwähnt auf Kapitel 4 und 6 verwiesen.

Die Problemstellung der British Telecom ist zwar frei zugänglich im Internet auffindbar, leider jedoch ein wenig versteckt. Zudem ist sie lediglich als Aneinanderreihung von Zahlen und Trennzeichen in verschiedenen Dateien hinterlegt. Ohne umfassende Auseinandersetzung mit den Dateien und der in Tabelle 3.8 aufgeführten Literatur sind die Daten kaum verständlich. Dies mag ein Grund sein, warum das Problem nach 2003 nicht mehr bearbeitet wurde. Zudem sind einige Arbeiten im Rahmen eines Forschungsprojektes der British Telecom entstanden, welches ein zeitlich definiertes Ende hatte.

Nachfolgend wird auf Problemstellungen und deren Lösungsverfahren eingegangen, die dem British Telecom-Problem ähnlich sind.

Zur Lösung von British Telecom-Problemen mit dynamischen Änderungen stellen Tsang u.a. 2005 [379] und 2008 [380] eine agentenbasierte Applikation ASMR (Automatic Synchronising Multiple Communication Rounds) vor. Sie ist in Java entwickelt und basiert auf dem RECONET-Protokoll (Retractable Contract NET), welches die Nutzung von

Tabelle 3.8: Lösungsverfahren anderer Autoren für das British Telecom-Problem

Verfahren	Autor/en	Jahr	Quelle
SA	Baker	1993	[22]
GA	Muller, Magill, Prosser und Smith	1993	[277]
CLP	Azarmi und Abdul-Hameed	1995	[18]
CLP	Yang	1996	[428]
D-CLP	Kokkoras	1996	[217]
HC	Tsang und Voudouris	1997	[383]
FLS	Tsang und Voudouris	1997	[383]
Fast GLS	Tsang und Voudouris	1997	[383]
D-CLP	Kokkoras und Gregory	1998	[218]
D-CLP	Sakellariou, Kokkoras und Vlahavas	2003	[330]

Metaheuristiken erlaubt. Experimente werden an zufällig erzeugten Problemstellungen vorgenommen, die die Struktur des British Telecom-Problems nachempfinden. ASMCR löst die getesteten Problemstellungen in 5–15 Minuten. Über den verwendeten Rechner wurden jedoch keine Aussagen gemacht. Während Kokkoras u.a. [217], [218], [330] für jedes Servicecenter einen Agenten erzeugen, werden nun für Servicecenter (Verkäufer von Leistungen = Service Seller) und für Regionen mit Aufträgen (Käufer von Leistungen = Service Buyer) Agenten erzeugt. Ein Agent kann immer mehrere Servicecenter bzw. Techniker umfassen. Zudem gibt es einen übergeordneten Agenten (Manager), der die Service Seller und Service Buyer steuert. Bei der Lösung werden Verträge zwischen Service Buyern und Service Sellern geschlossen. In einem Vertrag ist definiert, wer den Auftrag erledigt. Dabei wird wie folgt vorgegangen:

1. Service Buyer laden bestimmte Service Seller dazu ein, ein Gebot für einen Auftrag abzugeben.
2. Die angesprochenen Service Seller geben ein Gebot an den Service Buyer ab.
3. Der Service Buyer entscheidet sich für ein Gebot.
4. Der Service Buyer schließt einen Vertrag mit dem Service Seller.

Z.T. kann es besser sein, wenn ein Service Buyer einen Vertrag auflöst und stattdessen einen anderen Service Seller beauftragt. Z.B. wenn ein neuer Auftrag hinzukommt, ein anderer Service Buyer einen Vertrag mit dem Service Seller schließen möchte oder sich ein Techniker verspätet. Der Ablauf ist in Abbildung 3.6 dargestellt. Die Schritte 1 und 2 dienen dem Aushandeln eines Vertrages. In Schritt 3 stellt Service Buyer 1 eine Anfrage

an Service Buyer 2, in der er um die Auflösung des Vertrages von Service Buyer 2 mit dem Service Seller bittet. Hierzu bietet Service Buyer 1 eine Kompensationszahlung an. Bevor Service Buyer 2 entscheidet, versucht er einen anderen Service Seller zu finden, der den Auftrag übernehmen könnte. Daraufhin entscheidet er, ob die Kompensationszahlung genügt und löst den entsprechenden Vertrag auf oder nicht.

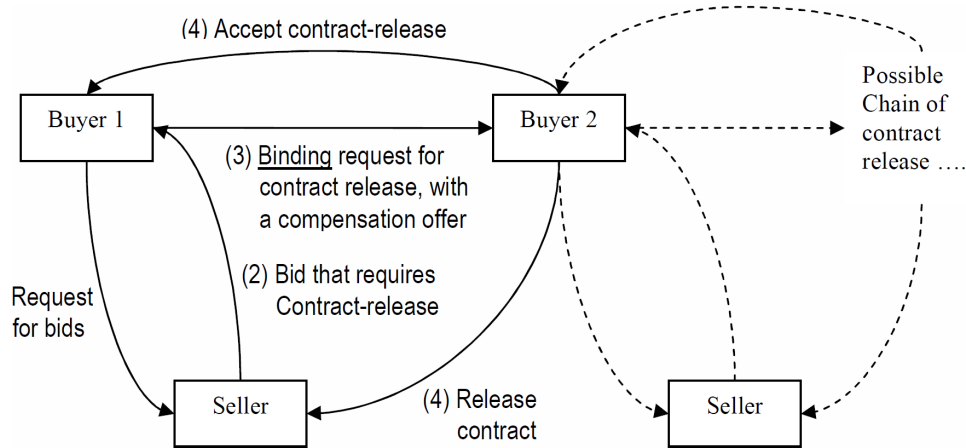


Abbildung 3.6: Ablauf des Gebots- und Rücknahmeprozesses [379, S. 495]

Ein weiteres Multiagentensystem stellen Tsang u.a. 2008 [382] vor, welches sich am British Telecom-Problem orientiert und die Mitarbeiterselbstbestimmung fördern soll. Es basiert auf ASMR und enthält ebenfalls Service Buyer und Service Seller, die in Verhandlung treten. In ihrer Arbeit erläutern sie jedoch lediglich die Idee – ohne Experimente durchzuführen.

In der Problemstellung der British Telecom sind Mitarbeiter bereits recht differenziert abgebildet. Es gibt unterschiedliche Leistungskennziffern und Qualifikationsprofile. Kliem und Anderson gehen 1996 [215] noch einen Schritt weiter und untersuchen die richtige Teambildung, damit die Erfolgswahrscheinlichkeit eines Projektes möglichst groß ist. Es wurden Zusammenhänge zwischen Erfolg eines Projektes und den Persönlichkeiten der Teammitglieder untersucht. Persönlichkeitstypen wurden identifiziert, die bei unterschiedlichen Projektarten verschieden erfolgreich sind. Das Ziel muss es also sein, immer die Personen mit der richtigen Persönlichkeit für das Team eines bestimmten Projekts auszuwählen. Die Planungsdimensionen Mitarbeiter, Zeitintervall, Auftrag/Arbeitsplatz wird also um die Dimension Team erweitert.

2007 gehen Naveh u.a. [281] bei der individuellen Abbildung der Mitarbeiter nicht so weit wie Kliem und Anderson [215], legen auf diesen Punkt aber ebenfalls einen Fokus. Die zu lösende Aufgabe ist dort nicht einfach in der Erstellung eines Einsatzplans zu sehen, bei dem gesetzliche Regelungen sowie An- und Abwesenheiten berücksichtigt werden. Vielmehr betreibt man neben der Planung auch ein Matching, um den richtigen Mitarbeiter

mit individuellen Eigenschaften auf den am besten passenden Auftrag zu planen. In diesem Zusammenhang wird von Identification & Assignment gesprochen. Es zeigt sich, dass Constraint Programming (CP) sehr gut für derartige Identification & Assignment-Probleme geeignet ist. Vor allem daher, da man laut Naveh u.a. reale Zusammenhänge nicht exakt in einer Kostenfunktion abbilden kann, die es zu optimieren gilt. Wie sollen z.B. die Kosten wegen Unzufriedenheit bewertet werden, wenn ein sehr gut qualifizierter Mitarbeiter einfache Packtätigkeiten ausführen muss. Oder wenn ein Mitarbeiter eine erforderliche Sprache nicht ausreichend beherrscht, der Kunde unzufrieden ist und sogar etwaige Nacharbeiten erforderlich werden. CP eignet sich nach Aussage von Naveh u.a. besonders, da man derartige Regeln recht gut in geeigneten Tools abbilden kann und nicht als mathematische Formeln hinterlegen muss.

Beim Wettbewerb ROADEF 2007 [87] (Société française de Recherche Opérationnelle et Aide à la Décision) wurden verschiedene Lösungsverfahren für drei Datensätze mit mehreren Instanzen der France Telecom untersucht (zwei der drei Datensätze wurden veröffentlicht [87]). Die Instanzen schwanken von 5 Mitarbeitern bis 150 und von 5 Aufträgen bis 800. In diesen Problemstellungen gibt es drei Qualifikationen, die von den Mitarbeitern auf einer Skala von eins bis vier unterschiedlich gut erfüllt werden. Auf der anderen Seite erfordern Aufträge eine bestimmte Anzahl an Mitarbeitern mit einer bestimmten Qualifikation und der entsprechenden Ausprägung. Weiterhin können Aufträge eine Bearbeitungsreihenfolge erfordern und eine Priorität besitzen. Zudem ist die Anzahl an Fahrzeugen begrenzt und Aufträge können extern vergeben werden. Einmal erstellte Teams zu Beginn der Planung werden für den Planungszeitraum nicht mehr aufgelöst und erledigen alle Aufträge gemeinsam. In diesem Punkt unterscheidet sich dieses Problem erheblich von Problem der British Telecom, wo Techniker immer allein arbeiten. Weitere Informationen sind in der Problembeschreibung von Dutot, Laugier und Bustos [109] zu finden. Im ROADEF-Wettbewerb war Rechenzeit ein begrenzender Faktor. Daher wurden die Problemstellungen zur Komplexitätsreduzierung z.T. stark zerlegt. So wurde sie von Balev und Gaci [23] und von Gavranović [137] in einzelne Tage geteilt. Bei Balev und Gaci war sie dann noch immer nicht zufriedenstellend zu bearbeiten, sodass man sie anschließend noch in Teams je Tag zerlegt hat. Recht häufig werden Konstruierende Verfahren verwendet, wobei sich die Verfahren im Einzelnen stark unterscheiden, was auch die Verteilung der Platzierungen über die gesamte Skala erklärt. Die Tabelle 3.9 zeigt die Bewertung der verschiedenen Lösungsverfahren (einige Verfahren sind unbekannt, da die Arbeiten nicht veröffentlicht wurden – zudem war die Arbeit [43] in französischer Sprache nicht verständlich).

Das Ranking beim ROADEF-Wettbewerb ergibt sich aus mehreren gewichteten Summanden. Zum einen ist in der Bewertung die gesamte Dauer des Einsatzplans enthalten und zum anderen ist jeweils die Dauer enthalten, bis alle Aufträge mit gleicher Priorität abgearbeitet sind [109].

In diesem Kapitel wurden die drei zu untersuchenden Problemstellungen vorgestellt, wobei auch vielfältige Problemvarianten erörtert wurden. Nachfolgend wird auf relevante

Tabelle 3.9: Lösungsverfahren im ROADEF-Wettbewerb 2007

Team	Platzierung	Lösungsverfahren	Quelle
28	1	Konstruierendes Verfahren	[195]
25	2	Lokale Suche	[116]
36	3	Konstruierendes Verfahren + Lokale Suche	[81]
44	4	Konstruierendes Verfahren + Lokale Suche	[177]
34	5	Lokale Suche	[305]
8	6	Konstruierendes Verfahren	[220]
24	10	Lokale Suche	[100]
26	11	Column Generation	[137]
2	12	Konstruierendes Verfahren	[23]
15	13	Konstruierendes Verfahren	[197]
1	14	Konstruierendes Verfahren	[301]
11	16	Konstruierendes Verfahren	[38]

Lösungsverfahren und deren Anpassung an Problemstellungen des hochflexiblen WFM eingegangen.

Kapitel 4

Lösungsverfahren für hochflexibles Workforce Management

In diesem Kapitel wird auf die Lösungsverfahren eingegangen, die an wenigstens einer der drei Problemstellungen aus Kapitel 3 erprobt werden. Zunächst wird die geeignete Abbildung der Problemstellungen erläutert, wobei verschiedenen Varianten diskutiert werden. Anschließend wird die Komplexität aller drei Problemstellungen in Kapitel 4.2 hergeleitet. Die Untauglichkeit der vollständigen Enumeration wird hierbei deutlich. Nachdem das Abbruchkriterium der Problemstellungen in Kapitel 4.3 dargestellt wird, werden in Kapitel 4.4 Vergleiche von Lösungsverfahren für das WFM aus der Literatur erörtert. In den Kapiteln 4.5 bis 4.16 werden die Lösungsverfahren vorgestellt, die in dieser Arbeit Anwendung finden. Die Unterkapitel der einzelnen Lösungsverfahren sind meist gleich aufgebaut. Nach Ausführungen zum grundlegenden Ablauf wird auf die Anwendung bei ähnlichen Problemstellungen eingegangen. Anschließend wird die Anpassung des Verfahrens an die Problemstellungen erläutert, bei denen Experimente durchgeführt werden. In diesem Zusammenhang wird die Forschungsfrage beantwortet wie die Lösungsverfahren bestmöglich an die drei beispielhaften Problemstellungen des hochflexiblen WFM angepasst werden sollten. Nicht alle Verfahren werden bei allen drei Problemstellungen erprobt. Dies gilt vornehmlich für die Lösungsverfahren, die von anderen Autoren/Unternehmen implementiert wurden. Z.B. am Handelsproblem oder am Problem der British Telecom.

4.1 Abbildung der Problemstellungen

Um die verschiedenen Lösungsverfahren anwenden zu können, müssen die Problemstellungen in geeigneter Weise abgebildet werden. Zur Modellierung der Problemstellungen stehen verschiedene Wege offen, die nachfolgend beschrieben werden. Aufgrund der Vor- und Nachteile der einzelnen Varianten wird darauf hin die Wahl der hier verwendeten

Form begründet. Anschließend wird die Abbildung der einzelnen Problemstellungen beschrieben.

4.1.1 Vorüberlegungen

In diesem Kapitel werden zunächst drei mögliche Varianten zur Abbildung der Problemstellungen aufgeführt, worauf hin die Auswahl der geeignetsten Variante begründet wird.

Variante 1 Diese Variante wurde in den 70er Jahren u.a. von Warner [409], von Miller, Pierskalla und Rath [266], von Kostreva, Leszcynski und Passini [223] sowie 1991 von Kostreva und Jennings [222] angewendet. Bei der Modellierung wird für jeden Mitarbeiter zunächst jede mögliche individuelle Einsatzplanungskombination in einer binär codierten Form hinterlegt, in der er für den zu planen Zeitraum arbeiten kann. Dieses Vorgehen macht die Modellierung sehr speicherintensiv – ermöglicht jedoch die Abbildung hoch komplexer Nebenbedingungen. Das Planungsproblem besteht nun darin, genau die Einsatzplanungskombinationen der Mitarbeiter auszuwählen, sodass der Plan optimal ist und durch das Auswählen von Einsatzplanungskombinationen weiterer Mitarbeiter keine harten Nebenbedingungen verletzt werden. Mit Hilfe einer binären Entscheidungsvariablen x_{ej} wird für jeden Mitarbeiter e festgelegt, welche seiner Einsatzplanungskombination j in den Plan gezogen wird. Dabei kann für jeden Mitarbeiter immer nur eine Einsatzplanungskombination gewählt werden:

$$\sum_{j=1}^J x_{ej} = 1 \quad \forall e \in \mathcal{E} \quad (4.1)$$

Variante 2 Eine weitere Variante zur Abbildung der Problemstellungen wird von Arthur und Ravindran [11] sowie von Hui [194] in den 80er Jahren genutzt. Sie verwenden eine dreidimensionale Matrix. Über die binäre Entscheidungsvariable x_{ewt} wird bestimmt, welcher Mitarbeiter e im jeweiligen Zeitintervall t auf welchem Arbeitsplatz w arbeiten soll. Das Planungsproblem besteht demnach darin, in jedem Zeitintervall für jeden Mitarbeiter den richtigen Arbeitsplatz zu wählen, sodass die Fehlerpunkte der weichen Nebenbedingungen minimiert und die harten Nebenbedingungen eingehalten werden. In dieser Variante kann die Abbildung komplexer Nebenbedingungen u.U. sehr schwierig oder nur über die Einführung einer Vielzahl an Ungleichungen realisierbar sein. Dafür ist sie jedoch recht speichereffizient.

Meisels und Schaerf [258] wenden sich 2003 von der binären Entscheidungsvariable mit drei Dimensionen ab und geben eine allgemeingültige Definition eines Personaleinsatz-

planungsproblems. Sie nutzen nun eine zweidimensionale Matrix. Die Zeilen repräsentieren Mitarbeiter und die Spalten Schichten. Ein Matrixelement stellt die Zuweisung einer Aufgabe/eines Arbeitsplatzes für einen Mitarbeiter in der entsprechenden Schicht dar.

Variante 3 Prinzipiell wäre es auch möglich, statt einer mehrdimensionalen Matrix eine Permutation aller Arbeitsplatz- oder Auftragszuweisungen zu verwenden. Tsang u.a. [383] wählen dieses Vorgehen 1997 u.a. für das Problem BT_Mod-250-118 (für eine detaillierte Beschreibung siehe Kapitel 4.9.3). Dieses Vorgehen erfordert weniger Plausibilitätsprüfungen und Korrekturmechanismen im Vergleich zu einer zwei- oder dreidimensionalen Matrix. Die Permutation aller Aufträge wird im Zuge der Bewertung durch die Fitnessfunktion in einen Einsatzplan transformiert.

Auswahl einer Variante Der Vorteil von Variante 1 besteht darin, dass in einem ersten Schritt die Einsatzplankombinationen der Mitarbeiter außerhalb der eigentlichen Optimierung erzeugt werden. Da dort lediglich Folgen von Arbeitsplatzzuweisungen je Mitarbeiter hinterlegt werden, können deutlich komplexere Nebenbedingungen berücksichtigt werden, die in Variante 2 gar nicht oder nicht in linearer Form abbildbar wären. Problematisch ist bei der ersten Variante jedoch, dass Thonton und Sattar [372] bei relativ moderat dimensionierten Plänen mit 25 bis 37 Mitarbeitern, 3 Schichten und 14 Zeitintervallen (somit 1.050 bis 1.554 binäre Entscheidungsvariablen) an die Speicherplatzgrenzen eines Rechners mit 8 MB RAM geraten. Das Problem besteht bei ihnen darin, dass viele Mitarbeiter Teilzeitkräfte mit wenigen Restriktionen sind und bei ihnen somit tausende möglicher Einsatzplankombinationen existieren. Auch Stolletz [358], der reale Problemstellungen mit bis zu 65 Mitarbeitern und mehreren tausend Zeitintervallen untersucht, kann die erste Variante nicht verwenden. Die Anzahl der Schichtabfolgen liegt dort pro Mitarbeiter im Maximum bei 331^{15} und ist daher nicht beherrschbar.

Bei Warner [409], der Variante 1 verwendet, existierten lediglich 10 bis 20 Kombinationsmöglichkeiten je Mitarbeiter, was eine starke Vereinfachung realer Problemstellungen darstellt [371]. Aus diesem Grund modellieren Thornton und Sattar die Problemstellung in einer Kombination aus Variante 1 und 2. Teilzeitkräfte werden über Variante 2 abgebildet, die deutlich speicherplatzsparender ist und alle anderen Mitarbeiter werden über Variante 1 abgebildet. So werden die Vorteile beider Modellierungsformen effizient genutzt und ihre Nachteile möglichst minimiert.

Wie zuvor erwähnt, stoßen Thornton und Sattar [372] mit Variante 1 bereits bei sehr kleinen Problemen an Speicherplatzgrenzen. Die Anzahl möglicher Kombinationsmöglichkeiten ist einfach zu groß. Moderne Rechner ermöglichen selbstverständlich die Umsetzung erheblich komplexerer Probleme. Doch allein das Wochenproblem des Logistik-

dienstleisters hat 65 Mitarbeiter, 560 Zeitintervalle (80 Intervalle am Tag a 15 Minuten * 7 Tage) und 9 Arbeitsplätze. Es wird deutlich, dass bei 327.600 binären Entscheidungsvariablen die Anzahl der möglichen Kombinationsmöglichkeiten je Mitarbeiter um ein Vielfaches höher ist. Beim Problem Handel_2AP sind es 262.800 und beim Problem BT_Mod-250-118 7.375.000 binäre Entscheidungsvariablen. Die Anwendung der ersten Variante ist demnach nicht praktikabel. Auch eine Kombination aus Variante 1 und 2 macht keinen Sinn. So haben Thornton und Sattar nur diejenigen Mitarbeiter mit Variante 2 abgebildet, die besonders viele Einsatzplanungskombinationen haben. Bei den drei verschiedenen Problemstellungen dieser Arbeit besitzen jedoch alle Mitarbeiter ähnlich viele Kombinationsmöglichkeiten. Alle Mitarbeiter sind gut ausgebildet und können daher auf vielen Arbeitsplätzen arbeiten. Zudem sind sie in vielen Zeitintervallen der Planungszeiträume anwesend bzw. verfügbar, sodass es auch hier keine nutzbaren Einschränkungen gibt.

Neben Variante 1 wird auch Variante 3 nicht bevorzugt. Bei der Transformation der Permutation im Rahmen der Bewertung durch die Fitnessfunktion können Bereiche des Lösungsraums von vornherein ausgeschlossen sein. Dies geschieht z.B. beim British Telecom-Problem in der Arbeit von Tsang u.a. [383]. Dort können kürzere Wege nicht immer gefunden werden. Denn z.T. macht es Sinn, einen Auftrag oder einen Arbeitsplatz nicht gleich dem ersten qualifizierten Mitarbeiter in einer festen Liste zuzuweisen. Es ist also recht schwer, eine geeignete Transformationsregel zu verwenden. Zudem arbeitet man nicht direkt auf dem eigentlichen Einsatzplan (der Matrix), was das Erkennen von Zusammenhängen hinsichtlich einer möglichen Fehlerreparatur erschwert. Daher wird dieses Vorgehen in dieser Arbeit nicht angewendet, obwohl es bei den Lösungsverfahren eine Vielzahl an Plausibilitätsprüfungen und Korrekturmechanismen reduzieren würde.

Aufgrund der genannten Schwächen werden die Problemstellungen komplett nach dem Vorbild der Variante 2 umgesetzt. Grundsätzlich wird sich am Konzept der zweidimensionalen Matrix von Meisels und Schaerf [258] orientiert, da im Vergleich zu einer dreidimensionalen Matrix die Anzahl an Entscheidungsvariablen reduziert ist. Lediglich die Abbildung in GAMS (siehe Kapitel 4.5) erfolgt über binäre Entscheidungsvariablen in einer dreidimensionalen Matrix. Der Hintergrund ist, dass diese Form in GAMS leichter abzubilden ist und GAMS bei der eigentlichen Modellerstellung völlig eigenständig vorgeht. An dieser Stelle sei angemerkt, dass einige Autoren beim British Telecom-Problem nicht auf Variante 2, sondern auf Modifikationen von Variante 3 setzen.

4.1.2 Untertägige Personaleinsatzplanung in der Logistik

Beim Logistikproblem findet eine zweidimensionale Matrix Anwendung. Dabei stehen die Zeilen der Matrix für Mitarbeiter. Die Spalten stehen nicht mehr wie bei Meisels und Schaerf [258] für Schichten, sondern für einzelne Zeitperioden der Länge $l_t > 0$. Hier wird

das Konzept der Time Tabling Blocks von Blöchliger [37] verwendet. Um Zeiten kenntlich zu machen, in denen ein Mitarbeiter laut Arbeitszeitmodell nicht verfügbar ist, wird ein Dummy-Arbeitsplatz eingeführt. In Tabelle 4.1 ist dies Arbeitsplatz 0. Dort ist Mitarbeiter zwei in der ersten Periode nicht verfügbar (daher auch nicht anwesend) und arbeitet anschließend auf Arbeitsplatz zwei. Um einen Verstoß gegen die harte Verfügbarkeitsnebenbedingung zu verhindern, dürfen nur Nicht-Dummy-Arbeitsplätze geändert werden. Somit wird gewährleistet, dass kein Mitarbeiter eingeplant wird, der nicht verfügbar ist. Der Dummy-Arbeitsplatz dient ebenfalls dazu, Zeiten kenntlich zu machen, in denen das Unternehmen nicht tätig ist und demzufolge kein Personal einzuplanen ist.

Tabelle 4.1: Abbildung der Problemstellung beim Logistikproblem

Mitarbeiter	Zeitperiode						
	1	2	3	4	5	6	...
1	1	1	1	1	1	1	
2	0	2	2	2	2	2	
3	0	1	1	2	2	2	
4	0	6	6	6	6	2	
5	3	3	2	2	0	0	
...							

Die Anzahl der Dimensionen ergibt sich beim Logistikproblem folglich aus der Anzahl an Zeilen (Mitarbeiter) * Anzahl an Spalten (Zeitintervalle). Bei Logistik_Complete und Logistik_S_Complete sind es z.B. 36.400 Dimensionen (65 Mitarbeiter * 560 Zeitintervalle (15 Minuten-Intervalle bei 7 Tagen a 20 Stunden)).

Zur Reduzierung der Komplexität macht es Sinn, die Anzahl der Dimensionen zu verringern. Im Fall des Logistikproblems lässt sich dies über die geeignete Abbildung der Zeit realisieren. Innerhalb des zu planenden Tages wird die Zeit daher als zeitdiskretes Modell dargestellt, was die Anzahl an Dimensionen deutlich verringert. Ein Eventpoint, bei dem ein neuer Zeitintervall beginnt, tritt dann ein, wenn sich der Bedarf an einem der Arbeitsplätze oder die Verfügbarkeit eines Mitarbeiters ändern. Bei den Logistikproblemen der kompletten Woche kann die Anzahl an Zeitintervallen z.B. um 21,25% reduziert werden. Mit diesem Vorgehen sind die Zeitintervalle jedoch nicht mehr gleich lang, so dass deren Länge l_t gesondert hinterlegt werden muss. Denn die Verletzung bestimmter Nebenbedingungen wird minutengenau bewertet.

4.1.3 Automatische Arbeitszeitmodellerstellung im Handel

Für das Handelsproblem findet ebenfalls die zweidimensionale Matrix analog zum Logistikproblem Anwendung. Auch hier stehen die Zeilen der Matrix für Mitarbeiter und die Spalten für einzelne Zeitperioden der Länge $l_t > 0$. Wie beim Logistikproblem macht der Dummy-Arbeitsplatz kenntlich, dass ein Mitarbeiter nicht verfügbar ist oder dass das Unternehmen nicht tätig ist – z.B. Zeitintervalle außerhalb der Öffnungszeiten oder Feiertage (in Tabelle 4.2 Arbeitsplatz 0).

Zusätzlich muss beim Handelsproblem ein Arbeitsplatz eingeführt werden, der kenntlich macht, dass der Mitarbeiter zwar verfügbar wäre aber nicht arbeitet – und daher auch nicht im Unternehmen erscheinen soll (in Tabelle 4.2 Arbeitsplatz 3). Also anders als beim Dummy-Arbeitsplatz, bei dem der Mitarbeiter nicht verfügbar ist oder das Unternehmen nicht tätig ist. Dieses Vorgehen ist erforderlich, da beim Handelsproblem auch die Arbeitszeitmodelle erstellt werden. Es muss ersichtlich sein, dass ein Mitarbeiter nicht arbeitet, aber prinzipiell eingeplant werden könnte.

Tabelle 4.2: Abbildung der Problemstellung beim Handelsproblem

Mitarbeiter	Zeitperiode						
	1	2	3	4	5	6	...
1	1	1	1	1	1	1	
2	0	2	2	2	2	2	
3	0	1	1	2	2	2	
4	0	0	0	0	0	0	
5	3	2	2	2	0	0	
...							

Die Dimensionen des Handelsproblems berechnen sich wiederum aus der Anzahl an Zeilen (Mitarbeiter) * Anzahl an Spalten (Zeitintervalle). Bei Handel_2AP sind es z.B. 131.400 Dimensionen (15 Mitarbeiter * 8.760 Zeitintervalle (1 h-Intervalle bei 365 Tagen a 24 Stunden)). Zwar hat das Geschäft nicht 24 Stunden am Tag geöffnet, die Modellierung der Problemstellung soll aber so flexibel geschehen, dass etwaige Erweiterungen wegen Inventur, verlängerten Öffnungszeiten etc. problemlos möglich sind. Zudem wird dieses Vorgehen von den beiden kommerziellen Softwarelösungen gefordert, die ebenfalls am Handelsproblem getestet werden (siehe Kapitel 4.6.3.1).

Alle selbst implementierten Lösungsverfahren gehen so vor, dass sie Zuweisungen mit einem Dummy-Arbeitsplatz (in Tabelle 4.2 Arbeitsplatz 0) unberücksichtigt lassen. Dort finden also keinerlei Änderungen oder Berechnungen statt. Eine Komprimierung der Pro-

blemstellung wie beim Logistikproblem hat für die Zeitintervalle mit Dummy-Arbeitsplatz keinen Nutzen – denn sie werden einfach übersprungen. Bei allen anderen Zeitintervallen ist die Komprimierung nicht möglich, da die Arbeitszeitmodelle (Anwesenheiten der Mitarbeiter) noch nicht feststehen und sich bei jedem Zeitintervall ein Eventpoint ergeben würde.

4.1.4 Kombination von Touren- und Personaleinsatzplanung

Zur Abbildung des British Telecom-Problems muss ein etwas anderes Vorgehen als beim Logistik- und Handelsproblem gewählt werden. Denn bei diesen Problemen stehen die Zeilen einer zweidimensionalen Matrix für die Mitarbeiter und die Spalten für die Zeitintervalle. Wobei die Zeit diskret abgebildet wird. Jedes Matrixelement enthält eine Zuweisung, auf welchem Arbeitsplatz der Mitarbeiter im jeweiligen Zeitintervall arbeitet. Dabei werden auch Verfügbarkeiten abgebildet.

Würde man dieses Konzept auf das British Telecom-Problem übertragen, so müsste man Zeitintervalle in 1-Minuten-Intervallen hinterlegen, da die Wegzeiten und die Bearbeitungszeiten der Aufträge in Minuten angegeben werden und stark schwanken können. Das Problem hätte damit 118 Zeilen (Anzahl Techniker) und 510 Zeitintervalle (Anzahl Minuten von 8.00–16.30 Uhr). Die so entstehende Matrix würde 60.180 Dimensionen umfassen.

Nach Möglichkeit soll der Lösungsraum, in dem tatsächlich Änderungen und Berechnungen stattfinden, so klein wie möglich gehalten werden, um die Komplexität des Problems zu reduzieren. Daher wird beim British Telecom-Problem zwar eine zweidimensionale Matrix, aber ein etwas anderes Vorgehen bei den Spalten der Matrix gewählt.

Für jeden Techniker wird eine Permutation der von ihm zu bearbeitenden Aufträge hinterlegt. Die Permutation kann alle 250 Aufträge beinhalten, was praktisch jedoch nicht eintritt (rechnerisch hat ein Techniker im Durchschnitt 2,1 Aufträge). Es ergibt sich somit eine Matrix, in der die Zeilen für Techniker und die Spalten für die Abfolge von Aufträgen stehen. Jedes Matrixelement enthält eine Auftragsnummer, wobei die Auftragsreihenfolge in der jeweiligen Permutation bestimmt, wann der Auftrag erledigt wird. Die exakten Zeiten ergeben sich aus Fahrt- und Bearbeitungszeiten sowie ggf. aus Restriktionen für den Beginn am Nachmittag. Die Zeiten sind nicht in der Matrix hinterlegt. Sie werden in der Fitnessfunktion berechnet. Matrixelemente ohne Auftrag erhalten einen einheitlichen Dummy-Wert (0 in Tabelle 4.3). Jeder Auftrag muss einem Techniker zugeteilt sein. Er darf jedoch nur einmal einem der 118 Techniker zugeteilt werden. Es ist auch möglich, dass ein Techniker keinen Auftrag erledigen muss. Sollte ein Techniker mehr Aufträge haben, als er erledigen kann, so bewertet die Fitnessfunktion die überschüssigen Aufträge als unerledigt. Mit diesem Vorgehen ergeben sich lediglich 29.500 Dimensionen, wovon 250 (die Aufträge) nicht mit einem Dummy-Wert belegt

sind.

Vom Grundgedanken her (zweidimensionale Matrix) entspricht dieses Vorgehen der Abbildung des Logistik- und Handelsproblems. Jedoch mit einer abgewandelten Berücksichtigung der Zeit. Eine Komprimierung wie beim Logistikproblem ist auch hier nicht möglich, da die Auftragszuweisung und -reihenfolge (Bedarf) nicht feststeht und somit keine Eventpoints gebildet werden können.

Tabelle 4.3: Abbildung der Problemstellung beim British Telecom-Problem

Mitarbeiter	Nummer in der Permutation						
	1	2	3	4	5	6	...
1	56	59	103	24	25	89	
2	0	0	0	0	0	0	
3	207	201	0	0	0	0	
4	203	217	209	36	12	204	
5	125	0	0	0	0	0	
...							

4.2 Komplexität

In der Komplexitätstheorie kann eine Klassifikation von Problemstellungen aufgrund der Zeitkomplexität erfolgen. Hierzu wird die maximale Anzahl an Berechnungsschritten in Beziehung zur Eingabelänge des Problems gesetzt.

Wenn die Rechenzeit mit der Eingabelänge nicht stärker als mit einer Polynomfunktion wächst, spricht man polynomialer Zeit. Kann ein Problem in polynomialer Zeit von einer deterministischen Turingmaschine gelöst werden, so liegt es in der Menge P. Existiert zur Lösung jedoch nur eine nichtdeterministische Turingmaschine, so spricht man von NP. Da ein Problem in P auch von einem nichtdeterministischen Algorithmus gelöst werden kann, gilt $P \subseteq NP$. Ob auch $P = NP$ gilt, konnte bisher nicht beantwortet werden. Ein Problem wird als NP-vollständig bezeichnet, wenn es in NP liegt und sämtliche bekannten NP-Probleme in polynomialer Zeit auf dieses Problem reduziert werden können. Bisher konnte kein NP-vollständiges Problem mit einem polynomiellen Algorithmus gelöst werden. NP-vollständige Probleme sind auch gleichzeitig NP-schwer. NP-schwere Probleme liegen nicht zwangsläufig in NP und sämtliche bekannten NP-Probleme sind in polynomialer Zeit auf dieses Problem reduzierbar. Daher existieren NP-schwere Probleme, die nicht in NP liegen und somit nicht NP-vollständig sind. [361, S. 12–13], [413,

S. 37–52] In Abbildung 4.1 sind die Komplexitätsklassen P, NP, NP-vollständig und NP-schwer dargestellt.

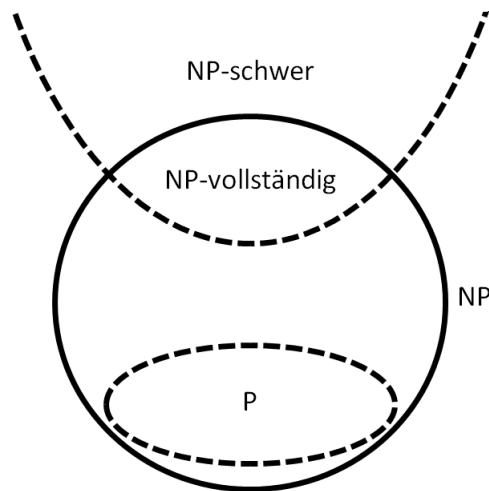


Abbildung 4.1: Abgrenzung der Komplexitätsklassen (in Anlehnung an [417])

Bei den drei verschiedenen Problemstellungen dieser Arbeit handelt es sich um kombinatorische Probleme. Während Lineare Programme (LP) in Polynomialzeit gelöst werden können, führt die Ganzzahligkeitsbedingung bei einer oder bei allen Variablen in der Regel zu NP-schweren Problemen. Dabei macht es keinen Unterschied, ob die Ganzzahligkeitsbedingung nur für eine Variable (MIP = Mixed Integer Programming) oder im Spezialfall für alle Variablen (IP = Integer Programming) gilt. Die Komplexitätsklasse ist immer dieselbe. Auch der Fall, dass ganzzahlige Variablen nur binäre Werte annehmen können (0-1 Integer Programming) ändert nichts an der Komplexitätsklasse. Ausgehend von Cooks Beweis der NP-Vollständigkeit des SAT-Problems [78] konnte Karp 1972 [202] zeigen, dass 0-1 Integer Programming und 20 weitere Probleme ebenfalls NP-vollständig sind. Neben der linearen Optimierung und seinen Sonderformen wie MIP, treten besonders bei praktischen Problemstellungen Nichtlinearitäten in der Zielfunktion und/oder den Nebenbedingungen auf. Die nichtlineare ganzzahlige Optimierung (MINLP = Mixed Integer Nonlinear Programming) ist besonders schwer zu lösen, da sie die Schwierigkeiten beim Lösen von MIP und NLP (Nonlinear Programming) vereint. Also die Komplexität kombinatorischer Probleme (MIP) und die Schwierigkeit, nichtlineare Probleme zu lösen (NLP). Da MIP- und NLP-Probleme aus komplexitätstheoretischer Sicht NP-schwer sind, ist die Schwierigkeit beim Lösen von MINLP-Problemen nicht verwunderlich [63].

Bezogen auf die Personaleinsatzplanung zeigen Garey und Johnson 1979 [135], dass selbst einfachste Formen des Staff Scheduling NP-vollständig sind. Zwei Jahre später konnte Bartholdi [24] die NP-Vollständigkeit für die Personaleinsatzplanung mit Schichtzyklen beweisen, bei der Mitarbeiter mit Unterbrechung zur Verfügung stehen. 1982 zeigen Tien und Kamiyama [374], dass praktische Personnel Scheduling Probleme noch komplexer sind als das Travelling Salesman Problem (TSP), welches an sich schon NP-vollständig

ist. Aus einer experimentellen Sicht legen die Arbeiten von Easton und Rossin [111] sowie von Brusco und Jacobs [54] nahe, dass allgemeine Personaleinsatzplanungsprobleme schwierige Optimierungsprobleme sind. Für die Stundenplanerstellung haben Cooper und Kingston 1995 [79] und 1996 [80] gezeigt, dass es sich ebenfalls um NP-vollständige Probleme handelt. Und 1998 haben Kragelund und Kabel [227, S. 12–15] für das allgemeine Employee Timetabling Problem bewiesen, dass es NP-schwer ist.

Trotz der Komplexität derartiger Zuordnungsprobleme wären sehr kleine Problemstellungen prinzipiell mit einer vollständigen Enumeration lösbar. Daher wird nachfolgend auf die Anzahl an Kombinationsmöglichkeiten aller drei Problemstellungen eingegangen. Unterstellt man bei allen drei Problemen, dass die Berechnung von 100.000 Lösungen auf einem aktuellen Rechner lediglich eine Minute benötigt, wird die Notwendigkeit geeigneter Lösungsverfahren nachfolgend deutlich. Es wird klar, dass eine vollständige Enumeration bei den drei Problemstellungen ausgeschlossen werden kann. Die Rechenzeit wäre nicht praktikabel.

Die Anzahl der Dimensionen ist bei den drei Problemstellungen dieser Arbeit recht groß. Die Problemstellung des Logistikdienstleisters hat in beiden Varianten des Wochenproblems mit 65 Mitarbeitern und sieben Tagen a 300 Zeitintervalle 36.400 Dimensionen. Die Komplexität ist W^D [333, S. 52–55]. W steht für die Anzahl an Arbeitsplätzen und D für die Anzahl an änderbaren Dimensionen. Bei neun zur Verfügung stehenden Arbeitsplätzen ergeben sich ohne Berücksichtigung von Qualifikationen und Verfügbarkeiten (aufgrund der Schichtpläne) $9^{36.400}$ Kombinationsmöglichkeiten. Berücksichtigt man Qualifikationen und Verfügbarkeiten, so sind es noch $9^{5.584}$ Kombinationsmöglichkeiten.

Beim Handelsproblem mit 2 Arbeitsplätzen, 15 Mitarbeitern und 8.760 Zeitintervalle sind es bereits 131.400 Dimensionen und $2^{131.400}$ Kombinationsmöglichkeiten. Unter Berücksichtigung von Verfügbarkeiten, die sich aus Öffnungszeiten und Feiertagen ergeben, sind es noch 51.150 Dimensionen (15 Mitarbeiter und 3.410 Zeitintervalle) und somit $2^{3.410}$ Kombinationsmöglichkeiten.

Das Problem der British Telecom umfasst 118 Mitarbeiter und 250 Aufträge, was zu 29.500 Dimensionen führt. Die Komplexität der Problemstellung ist J^S , wobei J die Anzahl der Aufträge und S die Anzahl der Aufträge ist, für die ein Techniker im Durchschnitt qualifiziert ist. Daraus ergeben sich 250^{27} Kombinationsmöglichkeiten (ca. 10^{56}). [428]

4.3 Abbruchkriterium

In diesem Kapitel werden die Abbruchkriterien der drei Problemstellungen beschrieben. Grundsätzlich lässt sich eine Vielzahl an Kriterien finden. Hansen und Mladenovic [175, S. 148] nennen einige Kriterien. Nachfolgend ist eine umfangreichere Auflistung dargestellt. Prinzipiell kann auch mit mehreren Abbruchkriterien gleichzeitig gearbeitet werden.

- Erreichen einer Zeitobergrenze.
- Erreichen einer maximalen Anzahl an Fitnessbewertungen.
- Erreichen einer maximalen Anzahl an Iterationen/Generationen.
- Unterschreitung eines bestimmten Fitnesswertes.
- Keine Verbesserung nach einer bestimmten Anzahl an Iterationen/Generationen oder Fitnessbewertungen oder Zeit etc.
- Zu geringe absolute Verbesserung der Fitness nach einer bestimmten Anzahl an Iterationen/Generationen oder Fitnessbewertungen oder Zeit etc.
- Zu geringe prozentuale Verbesserung der Fitness nach einer bestimmten Anzahl an Iterationen/Generationen oder Fitnessbewertungen oder Zeit etc.
- Bei einem populationsbasierten Verfahren ist die Standardabweichung der Fitness aller Partikel bzw. Individuen seit einer bestimmten Anzahl an Iterationen/Generationen oder Fitnessbewertungen oder Zeit etc. gleich null.
- Manueller Abbruch durch den Anwender.

Die Wahl des geeigneten Abbruchkriteriums für ein Lösungsverfahren hat einen entscheidenden Einfluss auf dessen Ergebnisqualität und die benötigte CPU-Zeit. Mit einem zu frühen Abbruch bleiben bessere Lösungen unentdeckt. Ein zu großzügig gewähltes Abbruchkriterium kostet unnötig CPU-Zeit, da das Verfahren längst konvergiert ist oder die Wahrscheinlichkeit, eine bessere Lösung zu finden, bereits verschwindend klein ist. CPU-Zeit ist in dieser Arbeit aber nicht der begrenzende Faktor. Daher soll das Abbruchkriterium so bemessen sein, dass es auf keinen Fall zu klein dimensioniert ist.

In dieser Arbeit wird einheitlich die Anzahl an berechneten Fitnessfunktionen als Abbruchkriterium gewählt. Dies ermöglicht eine sehr gute Vergleichbarkeit der Lösungsverfahren. Für die drei Problemstellungen dieser Arbeit werden die folgenden Abbruchkriterien gewählt. Diese Werte gelten für alle Varianten der jeweiligen Problemstellungen.

Bei den einzelnen Tagen des Logistikproblems z.B. hätten sie durchaus kleiner gewählt werden können. Rechenzeit ist bei allen Problemstellungen aber nicht der begrenzende Faktor, sodass dort keine Unterschiede gemacht werden.

- Untertägige Personaleinsatzplanung in der Logistik: 400.000 Berechnungen der Fitnessfunktion.
- Automatische Arbeitszeitmodellerstellung im Handel: 400.000 Berechnungen der Fitnessfunktion.
- Kombination von Touren- und Personaleinsatzplanung bei der British Telecom: 20.000.000 Berechnungen der Fitnessfunktion.

Zur Ermittlung der Abbruchkriterien wurden äußerst umfassend dimensionierte Testläufe durchgeführt. In Abbildung 4.2 erkennt man deutlich, dass ES beim Problem Handel_2AP schnell konvergiert (die Skalierung des Diagramms wurde bewusst nicht verändert, um das ausreichend dimensionierte Abbruchkriterium zu verdeutlichen). Bereits bei 200 Berechnungen der Fitnessfunktion sind die Fehlerpunkte sehr stark gesunken. Ab dann verläuft die Kurve in Abbildung 4.2 fast waagrecht. Die Wahrscheinlichkeit, dass das Verfahren bei 100.000+1 Berechnungen der Fitnessfunktion eine bessere Lösung findet, ist fast null. Weitere Berechnungen machen daher keinen Sinn. Dieser Effekt kann auch bei PSO und HC beobachtet werden. Um kein Risiko einzugehen, zu früh abzubrechen, wird eine Toleranz hinzugefügt, was die 400.000 Berechnungen der Fitnessfunktion beim Handelsproblem erklärt. Diese Testläufe wurden bei allen Problemstellungen durchgeführt.

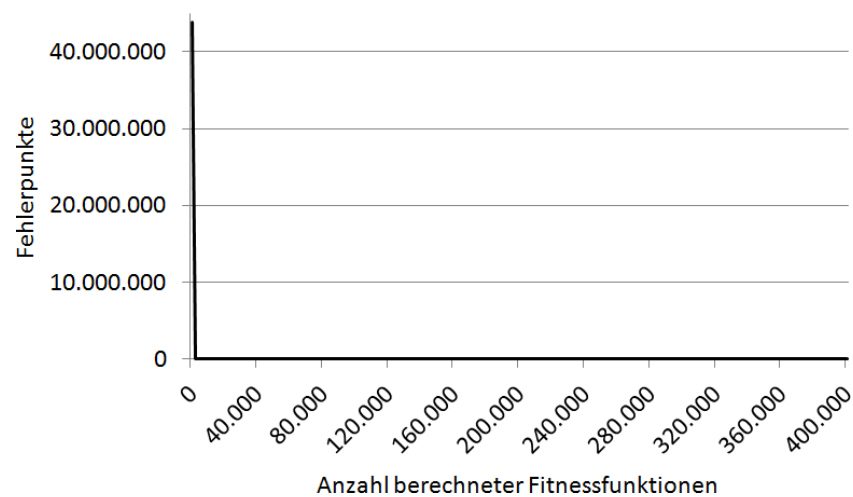


Abbildung 4.2: Verlauf der Fehlerpunkte für ES (1,5) TypeE + R bei Handel_2AP

Bei der Ganzzahligen Optimierung in Kapitel 4.5 wird vom einheitlichen Abbruchkriterium abgewichen. Hier werden acht Stunden für das Logistikproblem und die klei-

neren Problemausschnitte gewählt (andere Problemstellungen brauchen aufgrund der Erkenntnisse am Logistikproblem nicht untersucht zu werden). Alle anderen getesteten Lösungsverfahren benötigen beim Logistikproblem gewöhnlich weniger als eine Stunde – der Agentenansatz sogar weniger als eine Sekunde.

Eine weitere Ausnahme bilden die Agentenansätze, die eine Lösung konstruieren. Bei ihnen wird die Fitness für das Logistik- und Handelsproblem lediglich einmal am Ende berechnet. Beim British-Telecom-Problem muss sie mehrfach berechnet werden. Das Verfahren bricht ab, wenn keine Verbesserungen mehr erzielt werden können. Ein Kriterium bzgl. der CPU-Zeit ist nicht erforderlich, da sie immer im Sekundenbereich liegt.

4.4 Allgemeiner Überblick über Lösungsverfahren

Bereits in Kapitel 3 wurden Lösungsverfahren für drei Problemstellungen des hochflexiblen Workforce Managements dargestellt. Mit dem ROADEF-Wettbewerb in Kapitel 3.4.5 wurde auch ein Vergleich von Lösungsverfahren vorgestellt. Sie wurden an einheitlichen Problemstellungen und unter einheitlichen Testbedingungen beurteilt. An dieser Stelle sollen weitere Vergleiche von Lösungsverfahren dargestellt werden, wobei deren Anwendungsfeld über das Thema hochflexibles Workforce Management z.T. hinaus geht.

Seit einigen Jahrzehnten werden Problemstellungen zu den Themen Personaleinsatzplanung, Schichtplanung, Abwesenheitsplanung, etc. bis hin zur universitären Stundenplanerstellung intensiv bearbeitet. Neue Verfahren und verfeinerte Problemstellungen, die z.T. direkt aus der Praxis stammen, stehen dabei immer wieder im Zentrum der Betrachtung. Für Übersichten über Lösungsverfahren sei zusätzlich zu den äußerst umfassenden Nennungen von Ernst u.a. [114] auf die Arbeiten von Appelblad und Lönn [10] sowie Aickelin und Li [7] verwiesen. Zudem ist die Arbeit von Tsang [377] mit allgemeinen Ausführungen zu Lösungsverfahren sehr interessant, obwohl es sich dort bereits um das Job Shop Scheduling Problem handelt.

Bei den Lösungsverfahren für Problemstellungen rund um das Workforce Managements können zwei Typen unterschieden werden. Die eine Gruppe sind Algorithmen, die explizit für spezielle Problemstellungen entwickelt wurden. Z.B. in der Arbeit von Burke und Soubeiga [61] für den Catering-Bereich, wo eine nicht zusammenhängende Zielfunktion einen Lösungsraum mit mehreren Plateaus bildet. Bei der Lösung geht man in zwei recht simplen Schritten vor und hat den Algorithmus bereits an Echtdateen getestet. Zur Güte gibt es jedoch nur die Aussage, dass der verantwortliche Planer im Unternehmen mit den Ergebnissen sehr zufrieden ist. Neben speziell entwickelten Algorithmen finden auf der anderen Seite flexibel einsetzbare Methoden Anwendung. Ausgehend von Ernst u.a. [114] zeigt Tabelle 4.4 zusammenfassend verschiedene Lösungsverfahren und die

zugehörige Anzahl der Arbeiten.

Tabelle 4.4: Übersicht über Lösungsverfahren und Anzahl Arbeiten [114, S. 41]

Lösungsverfahren	Anzahl Arbeiten	Lösungsverfahren	Anzahl Arbeiten
Branch & Bound	14	Lagrangean Relaxation	32
Branch & Cut	9	Linear Programming	35
Branch & Price	30	Matching	36
Column Generation	48	Mathematical Programming	27
Constraint Logic Programming	46	Network Flow	38
Constructive Heuristic	133	Other Meta-Heuristic	11
Dynamic Programming	17	Other Methods	35
Enumeration	13	Queueing Theory	32
Evolution	4	Set Covering	58
Expert Systems	15	Set Partitioning	72
Genetic Algorithms	28	Simple Local Search	39
Goal Programming	19	Simulated Annealing	20
Integer Programming	139	Simulation	31
Iterated Randomised Construction	5	Tabu Search	16

Traditionell werden oft Lösungsverfahren aus dem Operations Research verwendet. Derartige Verfahren überwiegen in Tabelle 4.4. Z.B. untersucht Thornton 1995 [371] verschiedene Lösungsverfahren an 52 Problemstellungen der Dienstplanung. Hierzu gehören ein Verfahren für Integer Linear Programming (ILP), zwei Varianten von SA sowie zwei Varianten des zyklischen Descent-Algorithmus nach Miller u.a. [409]. Alle Verfahren erzeugten bessere Ergebnisse als die manuelle Planung. Das ILP-Verfahren konnte 34 der 52 Probleme lösen und sogar Probleme mit bis zu 3.000 Variablen. Bei größeren Problemen war es nicht mehr anwendbar. Die Ergebnisse von SA und des zyklischen Descent-Algorithmus waren nicht signifikant schlechter als die Optimalwerte. Der zyklische Descent-Algorithmus hatte die kürzeste Rechenzeit und da das ILP-Verfahren bei großen Problemen streikt, ist der zyklische Descent-Algorithmus aus der Sicht von Thornton zu bevorzugen. Als Kritikpunkte der mathematischen Optimierung führt er Probleme bei der Abbildung komplexer Nebenbedingungen und Schwierigkeiten beim Lösen großer Problemstellungen an.

Kroon, Salomon und van Wassenhove wenden 1997 [230] Lagrange-Relaxation und einen Greedy-Algorithmus an zufällig erzeugten Datensätzen einer Problemstellung eines Flughafens an. Die Problemstellung, die in die Klasse der Tactical Fixed Interval Scheduling Problems (TFISP) eingeordnet wird, sieht so aus, dass Aufgaben mit festem Beginn und Ende verplant werden müssen, wobei auch Qualifikationen zu berücksichtigen sind. Im Durchschnitt lagen die Ergebnisse beider Heuristiken nur 0–7% unter den Optimallösungen, die sehr rechenaufwändig ermittelt wurden.

Auch moderne Metaheuristiken werden angewendet. Ein Vergleich der Verfahren ist mit der Arbeit von Dowsland [102] sowie mit der Arbeit von Aickelin und Dowsland [6] möglich. Dabei wird das Dienstplanungsproblem in drei Teilprobleme zerlegt. Mit Phase eins wird sichergestellt, dass ausreichend Mitarbeiter zur Verfügung stehen, um die Nebenbedingungen zu erfüllen. In der zweiten Phase werden Mitarbeiter auf die einzelnen Arbeitstage verteilt, wobei ein Arbeitstag in Tag und Nacht unterschieden wird. In der dritten Phase erfolgt die Schichtzuweisung auf die Mitarbeiter. Für die zweite Phase wird von Aickelin und Dowsland [6] ein GA verwendet. Da sich der einfache kanonische GA für die untersuchte Problemstellung nicht eignet, wird ein koevolutionärer GA verwendet. Dort wird zur Lösung eine Hierarchie von Spezies verwendet, sodass Spezies auf Level zwei aus den Komponenten von Spezies auf Level eins bestehen. Zudem wurde an der Problemstellung mit 52 Datensätzen IP untersucht, wobei einige Datensätze auch nach erheblicher Laufzeit nicht gelöst werden konnten. Ebenso erwies sich SA als wenig erfolgreich und fand oft keine gültige Lösung. Das erfolgreichste Verfahren stellt TS dar. Variiert man jedoch die Parameter der Datensätze zufällig, so dominiert der koevolutionäre GA deutlich in Bezug auf Geschwindigkeit, Einfachheit und Robustheit.

Ebenso wurde CP und CLP für das Lösen von WFM-Problemen genutzt. Meisels und Lusternik haben 1998 [260] Personaleinsatzplanungsprobleme als Constraint-Netzwerke mit binären und nicht binären Nebenbedingungen abgebildet und zu deren Lösung eine Technik des Constraint Processings, Forward Checking and Conflict-Directed Back Jumping (FC-CBJ) verwendet. Dabei stellte man fest, dass bei einer Vergrößerung der durchschnittlichen Dimensionsgröße der Variablen auch die Anzahl an Lösungen des Problems stieg. Am aufwändigsten waren dann diejenigen Netzwerke zu lösen, bei denen eine Mischung aus lösbaren und nicht lösbaren Problemen vorherrschte. Entfernt man sich von dieser Mischung, so sinkt auch der Lösungsaufwand. Vergleichende Tests mit einem GA führten zu ähnlichen Resultaten. Das Ergebnis dieser Untersuchung war, dass es einen starken Zusammenhang zwischen der durchschnittlichen Dimensionsgröße der Variablen und des Lösungsaufwands gibt. Ähnliche Ergebnisse hatten Tsang u.a. [378] erzielt. Hier hat man GLS, TS und GA getestet, wobei GLS erheblich besser abschnitt. Es muss jedoch gesagt werden, dass TS und GA nur rudimentär implementiert wurden, sodass ein Vergleich dieser Heuristiken nicht allgemeingültig ist. Allgemein wurde hier auch gesagt, dass das Handling und das Lösen von Bedingungserfüllungsproblemen recht schwer ist, da der Anwender über spezielles Wissen verfügen muss. Z.B. wie er das Problem am besten formuliert, welches Lösungsverfahren sich am besten eignet, wie er problemspezifisches Wissen integrieren kann. Selbst am Markt verfügbare Tools wie der

ILOG Solver verlangen ein Höchstmaß an Wissen vom Anwender.

Lourenço, Paixão und Portugal erproben 2001 [250] diverse Lösungsverfahren an fünf realen Problemen des Bus-Driver Scheduling. Sie nutzen Lineare Programmierung (LP), TS, zwei GA sowie GRASP (Greedy Randomized Adaptive Search Procedure) – eine Kombination aus KV und Lokaler Suche. Die größte der fünf Problemstellungen konnten mit LP nicht gelöst werden. Dort schnitt TS am besten und GRASP am schlechtesten ab.

Häufig werden auch hybride Verfahren eingesetzt. Burke, Li und Qu stellen 2010 [60] ein derartiges Verfahren für die Dienstplanung vor. Das Problem wird zunächst vereinfacht und mit IP von CPLEX 10.0 gelöst. Bei der Vereinfachung bleiben lediglich die Nebenbedingungen erhalten, die unbedingt erforderlich sind und wenig Komplexität erzeugen und daher wenig Rechenzeit erfordern. Im zweiten Teil des hybriden Verfahrens findet VNS Anwendung [176], bei der Blöcke, bestehend aus Schichten, im Einsatzplan zwischen Mitarbeitern ausgetauscht werden. Das Verfahren operiert nicht mehr am vereinfachten Problem, sondern an der kompletten Problemstellung. Mit VNS sollen im zweiten Schritt all die Nebenbedingungen erfüllt werden, die bei IP unberücksichtigt blieben. Optimierte man das Problem allein mit CPLEX 10.0, so werden bei den zwölf Problemstellungen auch nach zwei Stunden keine akzeptablen Ergebnisse gefunden. Ebenso ist die alleinige Anwendung von VNS nicht zielführend, da bei keiner Problemstellung eine gültige Lösung ermittelt werden konnte. Die hybride Variante aus IP und VNS liefert jedoch sehr gute Ergebnisse. Als Benchmark wurden ein hybrider GA [307] und eine hybride VNS [56] der ORTEC Logiplan GmbH [294] getestet. Das Verfahren in [60] war bei allen zwölf Problemstellungen signifikant besser als die beiden Verfahren von ORTEC. Im Durchschnitt 15,2%.

Es ist erkennbar, dass generelle Aussagen zur Eignung von Lösungsverfahren nicht ohne Weiteres gemacht werden können. Größe, Komplexität, Struktur, . . . sowie die Rahmenbedingungen für die Lösung haben einen zu großen Einfluss auf die Auswahl eines Verfahrens. Daher verwenden Cowling, Kendall und Soubeiga eine Hyperheuristik [83], die den Einsatz von Heuristiken aufgrund von nicht problemspezifischen Informationen wie CPU-Zeit oder die Veränderungen der Fitnessfunktion steuert. In ihrer Arbeit wird die Hyperheuristik für die Meetingplanung einer Verkaufsveranstaltung (Sales Summit) genutzt, wobei verschiedene Auswahlverfahren untersucht werden. Zudem sind dort weitere Ausführungen zu heuristisch orientierten Lösungsverfahren zu finden.

In diesem Kapitel konnte nicht auf alle Lösungsverfahren eingegangen werden, die nachfolgend detailliert erläutert werden. Die Auswahl der Verfahren, die in dieser Arbeit an wenigstens einer der drei Problemstellungen getestet werden, resultiert aus den Forschungsfragen. So werden klassische OR-Verfahren, Metaheuristiken und Multiagentensysteme untersucht. Die konkrete Auswahl der Lösungsverfahren basiert auf zwei Kriterien.

1. Verfahren, die in der Literatur bereits häufig für ähnliche Problemstellungen eingesetzt wurden und dort sehr gute Ergebnisse erzielt haben.
2. Verfahren, von denen zu erwarten ist, dass sie sehr gute Ergebnisse erzielen werden (auch wenn sie heute noch nicht umfassend für derartige Problemstellungen eingesetzt werden).

Die in dieser Arbeit untersuchten Lösungsverfahren sind in Abbildung 4.3 mit Buchstaben markiert. Dabei steht das L für das Logistikproblem, das H für das Handelsproblem und das B für das British Telecom-Problem. Befindet sich ein Buchstabe im Kästchen auf der linken Seite, so wurde das zugehörige Verfahren beim jeweiligen Problem selbst implementiert. Steht es auf der rechten Seite, so haben andere Autoren eigene Experimente vorgenommen. Dies ist zumeist beim British Telecom-Problem erfolgt. Die Experimente zu TS von Helbig [178] wurden von ihm im gleichen Framework (ARROW siehe Kapitel 5.1) wie die eigenen Experimente vorgenommen. ES basiert beim Logistikproblem ohne Stammarbeitsplatzregelung auf der Arbeit von Birnstiel [35]. Anpassungen, Erweiterungen und die Experimente zu ES in dieser Arbeit wurden jedoch selbst durchgeführt.

In Abbildung 4.3 sind MAS nicht als kleines Kästchen dargestellt, sondern am Rand über die Lösungsverfahren gespannt. Diese Darstellungsart wird gewählt, da MAS eine Art Architektur sind, die Algorithmen verwenden. Ebenso sind die Hybride Verfahren über den unteren Rand gespannt, da sie mehrere der oben dargestellten Verfahren beinhalten können. Entsprechend der zentralen Forschungsfrage dieser Arbeit sind in Abbildung 4.3 die drei Gruppen: Klassische OR-Verfahren, Metaheuristiken und MAS zu erkennen. Es soll geklärt werden, wie sich Lösungsverfahren dieser drei Gruppen für die drei beispielhaften Problemstellungen des hochflexiblen WFM eignen. Die zusätzlich in Abbildung 4.3 aufgeführten Verfahren liefern sinnvolle Vergleichswerte, sind z.T. Bestandteil kommerzieller Softwarelösungen oder werden häufig in Zusammenhang mit ähnlichen Problemstellungen (vorherige Ausführungen und Kapitel 3.2.5, 3.3.5, 3.4.5) verwendet.

4.5 Ganzzahlige Optimierung

Eine zentrale Forschungsfrage dieser Arbeit ist, ob sich eher klassische OR-Verfahren, Metaheuristiken oder Multiagentensysteme für Problemstellungen aus dem Bereich des hochflexiblen WFM eignen. Zur Beantwortung der Frage werden beispielhaft drei Problemstellungen herangezogen. Nachfolgend wird auf die erste Gruppe der Lösungsverfahren – die klassischen OR-Verfahren – eingegangen.

Die ganzzahlige Optimierung findet in der Praxis eine breite Anwendung. So kann eine

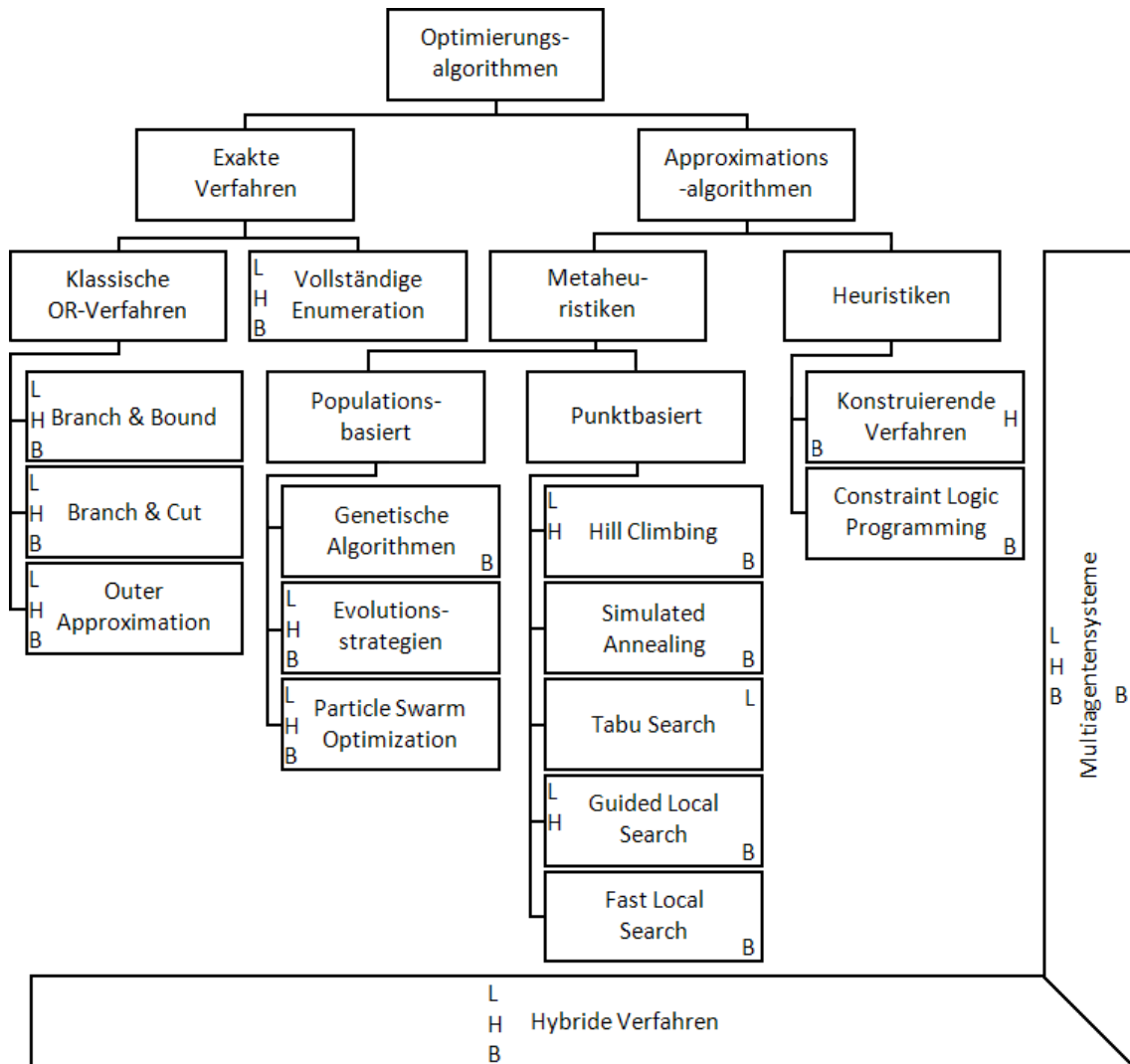


Abbildung 4.3: Einordnung von Lösungsverfahren (erweitert nach [46, S. 10])

Werft keine 1,37 Schiffe eines Typs herstellen, ein Mitarbeiter kann in einem Zeitintervall nur auf Arbeitsplatz eins oder Arbeitsplatz zwei arbeiten und ein LKW-Fahrer muss sich entscheiden, ob er zunächst Kunden A oder Kunden B anfahren will.

Obwohl bei der nachfolgenden Modellierung der Problemstellung keine reellwertigen Zahlen verwendet werden, wird aufgrund der Zugehörigkeit zur gleichen Komplexitätsklasse von der gemischt ganzzahligen linearen (MIP) bzw. nichtlinearen (MINLP) Optimierung gesprochen. Zumal die verwendete Modellierungsumgebung GAMS auch nur diese Begriffe im Rahmen der Solverauswahl unterstützen. Der geschilderte Zusammenhang wird in Abbildung 4.4 visualisiert.

In dieser Arbeit soll die Problemstellung des Logistikdienstleisters mit MIP und MINLP

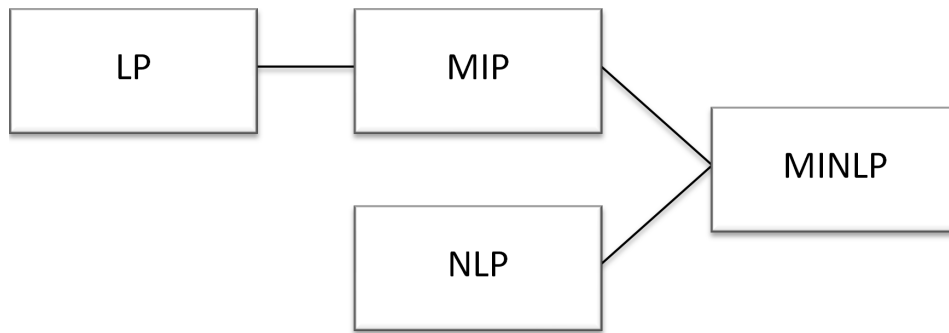


Abbildung 4.4: Zusammenhang zwischen LP, MIP, NLP und MINLP

gelöst werden. Hierzu wird die Entwicklungsumgebung GAMS 22.8 (General Algebraic Modelling System) [132] genutzt. GAMS ermöglicht eine sich stark an der mathematischen Syntax orientierende Modellierung von Problemstellungen und verfügt über einen leistungsfähigen Modellcompiler sowie eine beeindruckende Zahl an MIP- und MINLP-Solvern. Daher beschränkt sich die Arbeit mit GAMS im Wesentlichen auf die Umsetzung des Modells und die Parametrierung der Solver – den Rest erledigt GAMS. Sehr gute Einführungen in GAMS sind in [328] und etwas umfangreicher in [257] zu finden. Auf die Modellierung der beiden anderen Problemstellungen dieser Arbeit wird verzichtet, da die Schlussfolgerungen anhand des Logistikproblems auch auf das Handels- und das British Telecom-Problem übertragen werden können.

4.5.1 Grundlegender Ablauf

In diesem Kapitel wird zunächst auf die gemischt ganzzahlige lineare Optimierung (MIP) eingegangen. In diesem Zusammenhang werden Schnittebenenverfahren, Branch & Bound (B&B) sowie Branch & Cut (B&C) dargestellt. Anschließend folgen in Kapitel 4.5.1.2 Erläuterungen zur gemischt ganzzahligen nichtlinearen Optimierung (MINLP). Hier wird ebenfalls auf B&B eingegangen. Aber auch auf Outer Approximation (OA) sowie auf Generalized Benders Decomposition (GBD). Dieses Kapitel endet mit Ausführungen zu verschiedenen GAMS-Solvern, die in dieser Arbeit erprobt werden.

4.5.1.1 MIP-Verfahren

Die grundlegende Darstellung eines MIP-Problems sieht wie folgt aus:

$$P \begin{cases} \text{Min} Z = f(x, y) \\ \text{u.d.N.} & g_j(x, y) \leq 0 \quad j \in J \\ & x \in X \cap \mathbb{R}^n, y \in Y \cap \mathbb{Z}^m \end{cases} \quad (4.2)$$

Die Funktionen f und g sind konvex, differenzierbar und linear. Der Raum X ist kompakt und konvex. Der Raum Y ist eine durch Hyperebenen begrenzte Menge ganzer Zahlen: $Y = \{y \mid y \cap \mathbb{Z}^m, A_y \leq a\}$.

Eine Funktion f ist konvex, falls keine Sekante zu zwei beliebigen Punkten unterhalb des Graphen liegt. Liegt die Sekante zweier Punkte unterhalb von f , so ist f nichtkonvex. Konvexität liegt vor, wenn für alle $\lambda \in [0, 1]$ gilt:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \quad \forall x_1, x_2. \quad (4.3)$$

Für Konkavität gilt \geq statt \leq . Bei mehrdimensionalen Funktionen lässt sich die obige Formel analog verwenden. Im besten Fall ergibt sich die mehrdimensionale Funktion als Summe von eindimensionalen Funktionen. Dabei ist die Summe konvexer Funktionen ebenfalls konvex und die Summe konkaver Funktionen ist konkav. Die Richtigkeit dieser Aussage folgt direkt aus der Formel für konvex bzw. konkav. Zum Beweis siehe [180, S. 413–415].

In Abbildung 4.5 wird ein ganzzahliges lineares Optimierungsproblem grafisch veranschaulicht. Die gepunktete Linie zeigt die Zielfunktion. Bei einem LP würde sich der Optimalpunkt im Kreuzungspunkt der beiden Nebenbedingungen (durchgezogene Linien) befinden (hier x^*). Der Punkt x^* in Abbildung 4.5 ist für das ganzzahlige Optimierungsproblem jedoch ungültig, da gegen die Ganzzahligkeitsbedingung verstoßen wird. Optimal ist daher der Punkt $x_1 = 3; x_2 = 4$. Der von den gestrichelten Linien innerhalb der Nebenbedingungen und den Achsen eingefasste Bereich stellt die zulässige Menge des ganzzahligen Optimierungsproblems dar. Dabei sind alle Eckpunkte ganzzahlig und es ergibt sich ein konvexer Polyeder. Diverse Lösungsverfahren für MIP-Probleme werden nachfolgend beschrieben.

Schnittebenenverfahren Die ersten Verfahren zur Lösung ganzzahliger Optimierungsprobleme sind die Schnittebenenverfahren von Gomory [167]. Bei diesen Verfahren wird anstelle des ganzzahligen Problems eine Folge linearer Probleme gelöst. In den Abbildungen 4.5 und 4.6 wird dieser Zusammenhang veranschaulicht. Der dunkle Bereich stellt dabei die zulässige Menge des relaxierten ganzzahligen Problems dar. Ist das Ergebnis des so entstehenden linearen Problems nicht ganzzahlig, wird eine Schnittebene als zusätzliche Restriktion eingeführt. Ausgehend von Abbildung 4.5, bei der x^* (das Optimum des relaxierten Problems) im Kreuzungspunkt der Nebenbedingungen liegt, wird ein Schnitt eingefügt, der den Lösungsraum des relaxierten Problems weiter einschränkt. Der Schnitt wird dabei so gewählt, dass bei einem erneuten Lösen des relaxierten Problems ein neues x^* gefunden wird. Der Schnitt entfernt also die Optimallösung des relaxierten Problems und bei einem konvexen Lösungspolyeder nicht die Optimallösung des ganzzahligen Problems. Nachdem der Schnitt hinzugefügt wurde, wird das lineare Problem erneut berechnet. Dieses Vorgehen aus Schneiden und Berechnung des relaxierten

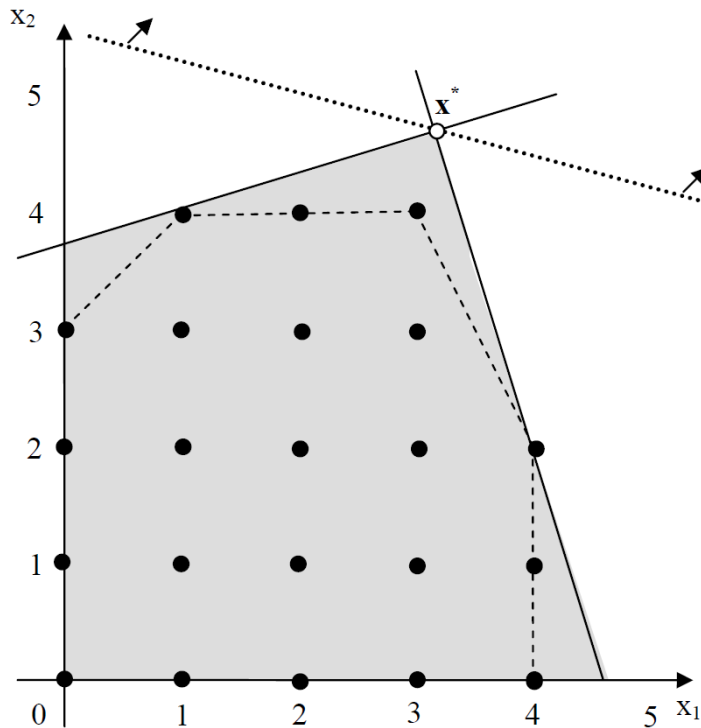


Abbildung 4.5: Grafische Darstellung eines ganzzahligen linearen Optimierungsproblems [412, S. 31]

Problems wird so lange wiederholt, bis eine ganzzahlige Lösung vorliegt oder festgestellt wird, dass keine optimale ganzzahlige Lösung existiert. Bei einem Minimierungsproblem erhöhen die Schnitte den Zielfunktionswert, bei einem Maximierungsproblem wird der Zielfunktionswert reduziert. Schnittebenenverfahren können sehr ungünstige Laufzeiten haben, da bei ihnen die Anzahl an Iterationen sehr hoch sein kann. Oft werden diese Verfahren daher mit B&B-Verfahren kombiniert, um diese zu beschleunigen.

Branch & Bound Bei B&B [90] wird eine Problemstellung gelöst, indem ein Enumerationsbaum (siehe Abbildung 4.7) aufgebaut wird, der aus Teilproblemen besteht. Die Elemente des Verfahrens sind die Initialisierung, die Verzweigung (Branching) und die Begrenzung (Bounding).

Bei der Initialisierung wird zunächst eine gültige Startlösung ermittelt. Der Wert der Startlösung definiert die obere Schranke des Problems. Ausgehend von der Startlösung wird ein relaxiertes LP erstellt, welches deutlich einfacher zu lösen ist. Z.B. mit Simplex- oder Innere-Punkt-Verfahren. Beim relaxierten LP wird eine zu fixierende Variable ausgewählt, die noch nicht ganzzahlig ist. Für diese Variable wird nun Ganzzahligkeit vorgeschrieben. Der Wertebereich dieser Variablen wird in zwei oder mehrere disjunkte Teilmengen zerlegt, sodass sich am Beispiel von Abbildung 4.7 ausgehend vom Problem

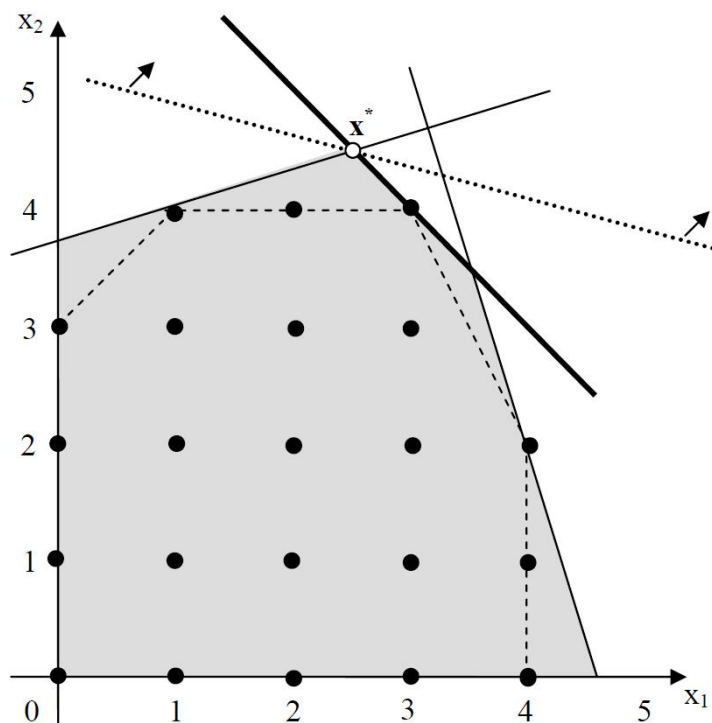


Abbildung 4.6: Grafische Darstellung eines ganzzahligen linearen Optimierungsproblems mit Schnitt [412, S. 31–32]

P_0 zwei Teilprobleme P_1 und P_2 ergeben (Branching). Durch das Lösen dieser relaxierten Teilprobleme werden ihre Untergrenzen ermittelt. Liegt eine der Untergrenzen über der Obergrenze, so kann dieser Zweig aus dem Enumerationsbaum entfernt werden (Bounding). Dies ist damit zu begründen, dass das Optimum des ganzzahligen Problems niemals besser als das Optimum des relaxierten Problems sein kann. Daher kann in der Optimallösung des ganzzahligen Problems die Obergrenze nicht unter der Untergrenze liegen. Abgeschnittene Zweige werden aus der weiteren Berechnung ausgeschlossen, sodass sich die Rechenzeit verkürzt. Eine neue Obergrenze wird dann gefunden, wenn die Lösung eines Teilproblems von der Wurzel bis zum Blatt des Baumes besser als die bisherige Obergrenze ist. Die Lösung ist dann zwangsläufig ganzzahlig. Fallen Ober- und Untergrenze zusammen, bricht das Verfahren ab, da dann eine ganzzahlige Lösung gefunden wurde, dessen Optimalität bewiesen ist.

Der Rechenaufwand wird durch ein geschicktes Verzweigen der Teilprobleme und eine frühzeitige Terminierung nicht Erfolg versprechender Teilbäume reduziert. Hierzu werden verschiedene Knotenauswahlregeln genutzt [412, S. 72–75]:

- statische Regeln,
- auf Schätzungen basierende Regeln,

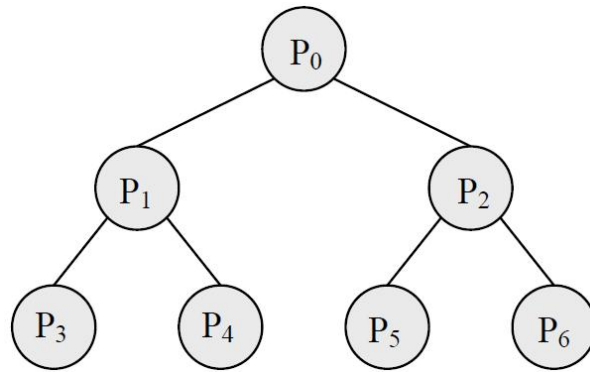


Abbildung 4.7: Enumerationsbaum bei Branch & Bound [412, S. 66]

- zwei-Phasen Regeln,
- backtracking Regeln.

So ist es vorteilhaft, schnell eine möglichst gute Lösung zu finden, um die Obergrenze herunter zu setzen und den Baum einzuschränken. Auf der anderen Seite macht es Sinn, die Untergrenze heraufzusetzen, um die Optimalität einer gefundenen Lösung zu beweisen. Im ungünstigsten Fall kann bei B&B jedoch auch eine vollständige Enumeration erfolgen. Bei umfangreichen Problemen kann ebenfalls der Speicherplatzbedarf kritisch sein [184, S. 6].

Branch & Cut Dieses Verfahren basiert auf der Kombination des B&B-Verfahrens mit dem Schnittebenenverfahren [283]. Die Kombination kann auf vielfältige Weise geschehen. So können Schnitte lediglich am Ausgangsknoten des B&B-Verfahrens in das relaxierte LP-Problem eingefügt werden, ehe das B&B-Verfahren startet. Es ist jedoch auch möglich, in einigen oder allen Knoten des Enumerationsbaums zu prüfen, ob weitere Schnittebenen eingefügt werden können.

B&C-Verfahren sind in den einzelnen Knoten rechenaufwändiger als B&B-Verfahren, wenn dort weitere Schnittebenen eingeführt werden sollen. Hierdurch lässt sich aber gewöhnlich die obere und untere Schranke besser abschätzen, wodurch sich die Anzahl der zu untersuchenden Knoten reduziert und der Rechenaufwand in Summe meist zurück geht. Der Speicherbedarf ist bei B&C geringer als beim Schnittebenenverfahren, wenn die Schnittebenen für den gesamten Baum gelten. Sie können aber ebenso nur für einen Teilbaum gelten.

4.5.1.2 MINLP-Verfahren

Ein MINLP-Problem hat die folgende Gestalt:

$$P \begin{cases} \text{Min}Z = f(x, y) \\ \text{u.d.N. } g_j(x, y) \leq 0 \quad j \in J \\ x \in X \cap \mathbb{R}^n, y \in Y \cap \mathbb{Z}^m \end{cases} \quad (4.4)$$

Dabei sind f und g konvexe und differenzierbare Funktionen, wobei f und/oder g nicht-linear sind. Der Raum X ist kompakt und konvex. Der Raum Y ist eine durch Hyper-ebenen begrenzte Menge ganzer Zahlen: $Y = \{y \mid y \in \mathbb{Z}^m, A_y \leq a\}$. Würde $y \in Y \cap \mathbb{R}^m$ statt $y \in Y \cap \mathbb{Z}^m$ gelten, so würde sich P von MINLP auf NLP reduzieren. Wären f und g linear, so reduziert sich P von MINLP auf MIP. Treten beide Fälle auf, so würde sich P von MINLP auf ein klassisches LP reduzieren [238]. Weiterführende Ausführungen zu MINLP sind der Arbeit von Floudas [121] zu entnehmen.

Bei den Lösungsverfahren für MINLP-Probleme macht man sich die Erfahrungen bei der Lösung von MIP und NLP zu Nutze, bei denen man in der Vergangenheit bereits sehr gute Fortschritte erzielt hat. Nachfolgend werden Verfahren zur Lösung von MINLP-Problemen beschrieben.

Branch & Bound Bei B&B orientiert sich das Lösen von MINLP-Problemen am Vorgehen bei MIP. Ausgehend von einem Ausgangsknoten, der eine zulässige ganzzahlige Lösung darstellt und die Obergrenze definiert, werden zwei oder mehrere Teilbäume gebildet. In jedem Teilbaum werden ganzzahlige Variablen sukzessive fixiert. Die Besonderheit ist, dass bei MINLP kontinuierliche NLP-Relaxationen der Teilprobleme erzeugt werden, deren Lösungen die unteren Schranken der Teilbäume definieren. Bei der Lösung wird sukzessive ein Baum mit ganzzahligen Variablen als Knoten aufgebaut. Aus diesem Enumerationsbaum werden Zweige entfernt, wenn dort die Untergrenze über der Obergrenze liegt. Der Algorithmus bricht ab, sobald Ober- und Untergrenze zusammen fallen. B&B wird vor allem dann angewendet, wenn die NLP-Teilprobleme effizient zu lösen sind oder nur wenige dieser Probleme gelöst werden müssen. Dies ist meist dann der Fall, wenn es nur wenige ganzzahlige Variablen gibt.

Outer Approximation Duran und Grossmann beschrieben das OA-Verfahren erstmals 1986 [108]. Im Gegensatz zu B&B sollen bei OA weniger Berechnungen von NLP-Problemen erfolgen. Krall [199, S. 255] skizziert OA-Verfahren wie folgt: „[Sie] beschreiben den zulässigen Bereich eines beschränkten Optimierungsproblems als Schnitt eines Ensembles von einfach strukturierten Mengen, z.B. Polyeder.“ Mit diesem Verfahren lässt sich aus dem ursprünglichen Problem ein „einfacheres“ Problem erstellen. So wird

bei diesem Verfahren ein Problem durch die abwechselnde Lösung zweier Hilfsprobleme gelöst. Zum einen das nichtlineare kontinuierliche NLP-Teilproblem und zum anderen das lineare ganzzahlige MIP-Masterproblem. Bei den NLP-Teilproblemen wird jeweils eine ganzzahlige Variable fixiert. Die Lösung des kontinuierlichen Teilproblems liefert die obere Schranke des MINLP-Problems. Die Lösung des MIP-Masterproblems liefert die untere Schranke des MINLP-Problems und eine neue zu fixierende Variable für das NLP-Teilproblem in der nächsten Iteration. Dieses Vorgehen wird so lange wiederholt, wie die untere Grenze kleiner als die obere Grenze ist. Das Verfahren bricht nach endlicher Zeit entweder mit einer optimalen ganzzahligen Lösung ab oder mit der Gewissheit, dass es keine optimale ganzzahlige Lösung gibt [169]. Werden nichtkonvexe MINLP-Probleme mit OA-Verfahren gelöst, so können besonders folgende Fehler auftreten:

- Die NLP-Hilfsprobleme können u.U. keine globalen Minima liefern.
- Das Masterproblem kann bei der Linearisierung ggf. Lösungen abschneiden.
- Die Masterprobleme liefern keine rigorosen Schranken.

Mit Equality Relaxation und Augmented Penalty soll den Effekten bei nichtkonvexen MINLP-Problemen weitestgehend entgegen gewirkt werden. DICOPT (Discrete and Continuous Optimizer) ist z.B. ein von GAMS unterstützter Solver, der die beiden Verfahren enthält. Da für die Nutzung von DICOPT eine Lizenz benötigt wird, die für diese Arbeit nicht vorliegt, muss auf Experimente mit diesem Solver verzichtet werden.

Generalized Benders Decomposition Ähnlich wie bei OA ist auch der Ablauf des Verfahrens GBD gestaltet [169], [121]. Der Hauptunterschied zwischen GBD und OA besteht in der Formulierung des MIP-Masterproblems. Bei GBD werden weniger Nebenbedingungen in das MIP-Masterproblem eingefügt als bei OA, sodass folglich die untere Schranke weniger eng gefasst wird. Es gilt: untere Schranke GBD < untere Schranke OA. Somit ist davon auszugehen, dass GBD deutlich mehr Iterationen für die Lösung benötigt als OA. Da es bei GBD weniger Nebenbedingungen und auch weniger Variablen gibt, sind bei GBD jedoch die Iterationen günstiger als bei OA.

4.5.1.3 GAMS-Solver

Die in dieser Arbeit genutzten Solver für MIP und MINLP stammen vor allem aus der COIN-OR-Bibliothek. COIN-OR steht für Computational Infrastructure for Operations Research und beinhaltet diverse Open Source Projekte [73]. Im Rahmen des GAMSlinks-Projektes von COIN-OR stehen folgende Solver in GAMS 22.8 zur Verfügung (die vom Solver lösbaren Modelle sind ebenfalls aufgeführt – zuvor nicht erläuterte Abkürzungen sind dem Abkürzungsverzeichnis zu entnehmen):

- CoinBonmin 0.99: LP, RMIP, MIP, DNLP, NLP, RMINLP, MINLP, QCP, RMI-QCP, MIQCP
- CoinCBC 2.1: LP, MIP, RMIP
- CoinGlpk 4.26: LP, MIP, RMIP
- CoinIpopt 3.4: LP, RMIP, DNLP, NLP, RMINLP, QCP, RMIQCP
- CoinScip 1.00: LP, MIP, RMIP

In dieser Arbeit werden der MINLP- und MIP-Solver CoinBonmin sowie die MIP-Solver CoinCBC und CoinGlpk genutzt. Alle anderen MIP- und MINLP-Solver müssen in GAMS über eine Lizenz freigeschaltet werden, die nicht zur Verfügung steht. Eine Ausnahme bildet der LP- und MIP-Solver BDMLP, der ebenfalls frei in GAMS verfügbar ist. Vergleichende Untersuchungen von einigen MINLP-Solvern, die hier nicht untersucht werden konnten, wurden von Bussieck, Drud und Meeraus [62] an 250 Problemstellungen der MINLPLib vorgenommen. Dort schnitt AlphaECP am besten ab und fand die meisten Lösungen, wobei SBB und DICOPT generell schneller waren. Linderoth und Ralphs [247] untersuchten 2005 sechs nichtkommerzielle MIP-Solver, von denen jedoch nur CoinCBC und CoinGlpk in GAMS enthalten sind. Zwar ist keine eindeutige Aussage möglich, welcher Solver zu bevorzugen ist, CoinCBC und CoinGlpk sind jedoch allgemein im Mittelfeld anzusiedeln. Bei besonders schwer zu lösenden Problemen zeigt CoinGlpk vor CoinCBC die zweitbeste Performance.

Nachfolgend werden die GAMS-Solver beschrieben, die in dieser Arbeit erprobt werden.

BDMLP Der kostenfrei in GAMS verfügbare LP- und MIP-Solver BDMLP (Brooke, Drud and Meeraus Linear Program) wurde ursprünglich von der Weltbank von Brooke, Drud, und Meeraus entwickelt und wird nun von der GAMS Development Corporation weiter gepflegt. Das B&B-Verfahren für MIP wurde von Bussieck und Drud hinzugefügt. BDMLP eignet sich für moderat dimensionierte Problemstellungen, solange sie nicht zu stark degeneriert [200] und gut skaliert sind. Laut GAMS ist BDMLP nicht mit der Geschwindigkeit kommerzieller MIP-Solver vergleichbar [131]. Dies wird u.a. auch von Lastusilta, Bussieck und Westerlund [232] gezeigt, wo kommerzielle Solver eine bessere Wahl sind.

CoinCBC CoinCBC (COIN-OR Branch and Cut) ist ein in C++ geschriebener frei verfügbarer Solver des COIN-OR-Projektes [73]. Er wurde vor allem von Forrest und

Lougee-Heimer entwickelt. Wie der Name CBC schon nahelegt, handelt es sich um einen B&C-Algorithmus. Er kann für LP-, MIP- und RMIP-Probleme angewendet werden.

CoinGlpk Ein ebenfalls für LP-, MIP- und RMIP-Probleme geeigneter Solver im COIN-OR-Projekt ist CoinGlpk (Gnu Linear Programming Kit). Er wurde von Makhorin in ANSI C geschrieben und ist in Form von Bibliotheken organisiert. Für die Lösung des MIP wird ein B&B-Algorithmus verwendet. Dieser ist in der Lage, 100 bis 200 ganzzahlige Variablen zu verarbeiten. Je nach Problemgröße und Komplexität waren bisher auch bis zu 1.000 ganzzahlige Variablen möglich. Eine umfangreiche Beschreibung von Glpk ist im Handbuch der Free Software Foundation zu finden [125].

CoinBonmin CoinBonmin (Basic Open-source Nonlinear Mixed Integer programming) [40] wurde im Rahmen eines Gemeinschaftsprojektes der IBM an der Carnegie Mellon Universität in C++ entwickelt und ist ebenfalls in das COIN-OR-Projekt integriert. Mit seiner Hilfe sollen u.a. MIP- und MINLP-Probleme gelöst werden, wobei man bei der Entwicklung von CoinBonmin besonders auf MINLP fokussiert. Hierfür sind fünf verschiedene Algorithmen implementiert:

- CoinBonmin-BB: einfacher B&B-Algorithmus, der auf der Lösung eines NLPs an jedem Knoten des Baums beruht [108].
- CoinBonmin-OA: auf OA basierender Dekompositions-Algorithmus [171], [120].
- CoinBonmin-QG: auf OA basierender B&C-Algorithmus [316]. CoinBonmin-QG ähnelt einer Variante von CoinBonmin-Hyb, in der einige Funktionen deaktiviert sind.
- CoinBonmin-Hyb: ein hybrider B&C-Algorithmus, der auf OA und einem B&B-Algorithmus basiert [40].
- CoinBonmin-Ecp: auf ECP-Schnitten (Extended Cutting Plane) basierender B&C-Algorithmus nach FilmINT [4].

Die aufgeführten Bonmin-Algorithmen sind exakt, wenn das zu lösende MINLP konvex ist. Bei nichtkonvexen MINLPs können sie dennoch angewendet werden – sind dann aber Heuristiken, bei denen keine Optimalität der Lösung garantiert ist.

4.5.2 Anwendungen in der Literatur

Naidu u.a. geben in [279] einen Überblick über Lösungsverfahren für die Einsatzplanung. U.a. auch Verfahren der linearen und ganzzahligen Optimierung. Derartige Lösungsverfahren liefern optimale Ergebnisse in Bezug auf die Minimierung des Mitarbeiterereinsatzes, die Reduzierung der Überstunden und anderer Ziele. Die hierzu notwendigen Modellvereinfachungen gehen jedoch auf Kosten der Anwendbarkeit und demzufolge auf die Qualität des Einsatzplans. So sagen Naidu u.a., dass mit mathematischen Modellen arbeitsergonomische Restriktionen wie z.B. Wünsche nicht abgebildet werden können. Negative Auswirkungen auf die Mitarbeitermotivation, Fluktuations- und Krankheitsquote sowie auf die Produktivität und den Servicelevel sind vorprogrammiert. Der Begriff der „optimalen Lösung“ ist laut Naidu u.a. daher kritisch zu betrachten. Die Aussage, dass Wünsche in mathematischen Modellen nicht abgebildet werden können, ist mit Vorsicht zu genießen. So können Mitarbeiterwünsche für Schichten oder Arbeitsplätze sehr wohl in linearer Form abgebildet werden. Hierzu sind lediglich zusätzliche Fehlerpunkte bei Nichterfüllung des Wunsches für jede Entscheidungsvariable und eine Erweiterung der Zielfunktion nötig. Grundsätzlich steht aber fest, dass die Modellierung einiger Nebenbedingungen äußerst komplex und z.T. nicht in linearer Form möglich sein kann.

Eine sehr umfangreiche Bibliografie liefern Ernst u.a. in [114] mit etwa 700 Arbeiten zum Rostering. In einem sehr großen Teil der Arbeiten findet IP Anwendung und dominiert neben den ebenfalls häufig verwendeten KV alle anderen Verfahren bzgl. der Anwendungshäufigkeit. Ernst u.a. stellen fest, dass die Komplexität der in den Arbeiten untersuchten Problemstellungen im Laufe der Zeit nicht zugenommen hat. Lediglich bei den entsprechenden Modellen ist eine stärkere Orientierung an der Realität festzustellen, was an leistungsfähigeren Lösungsverfahren und Rechnern liegt.

Kostreva u.a. orientieren sich 1976 [266] und 1991 [222] an der Modellierungsform von Warner [409] und erzeugen ein MIP. Dieses lösen sie, indem sie zwei Phasen iterativ durchlaufen und das Ergebnis sukzessive verbessern. In der ersten Phase wird heuristisch ein Einsatzplan erzeugt, der die Nebenbedingungen des Problems erfüllt. Jeder Mitarbeiter hat im Vorfeld Strafpunkte für bestimmte Schichtkonstellationen verteilt. In der zweiten Phase werden daher die Strafpunkte minimiert. Bei diesem Verfahren ist jedoch keine Optimalität garantiert, wie dies bei Warner der Fall ist. Die Qualität der Lösung hängt nämlich vom heuristisch erstellten initialen Plan ab. Wird das Verfahren zu einem vorzeitigen Abbruch gezwungen, so wurde aber meist schon ein recht gutes Ergebnis erzielt. Bei Warner führt ein vorzeitiger Abbruch zu einem unvollständigen Plan. Leider wurde das Verfahren von Kostreva u.a. nicht sonderlich detailliert beschrieben und es wurden auch keine vergleichenden Tests mit anderen Verfahren angestellt, sodass weiterführende Aussagen nicht möglich sind.

In der Arbeit von Alfares [9] wird 2002 eine Problemstellung des Days-Off Scheduling beleuchtet. In einem Zeitraum von 21 Tagen stehen jedem Mitarbeiter 7 freie Tage zu.

Da die Mitarbeiter kostspielig vom und zum Einsatzort transportiert werden müssen, ist es das Ziel, die Anzahl eingeplanter Mitarbeiter zu reduzieren und die Transportkosten zu minimieren. Das Problem wurde als duales LP abgebildet. In Vortests mit reduzierten Problemstellungen wurden die Tools HyperLindo und ExcelSolver eingesetzt, um IP als Lösungsverfahren zu testen. Einige Testfälle konnten auch nach mehrstündiger Laufzeit nicht gelöst werden, sodass sich IP nicht als effektives Lösungsverfahren eignete. Daher hat man die Problemstellung weiter reduziert und ein simples Verfahren in drei Schritten entwickelt, das den Zusammenhang von dualem und primalem Problem im Optimum berücksichtigt.

Ein Schichtplanungsproblem, in dem Schichten mit untertägigen Arbeitsplatzwechseln in einem Call Center erzeugt werden, wird 2008 von Bhulai, Koole, und Pot [32] untersucht. Tatsächlich finden in den Schichten keine physischen Arbeitsplatzwechsel statt. Ein Arbeitsplatzwechsel ist in Form eines Teamwechsels definiert, bei dem der Call Center Agent an seinem Arbeitsplatz verbleibt. Die Berücksichtigung von Teams mit bestimmten Qualifikationen macht die Problemstellung im Vergleich zur Berücksichtigung von nur einer Qualifikation recht komplex [31]. Hinzu kommt, dass ein Servicelevel eingehalten werden muss, was das Problem ebenfalls deutlich schwerer macht. Diesem Umstand wird Rechnung getragen. So untersuchen Bhulai, Koole, und Pot eine integrierte Optimierung sowie eine zyklische Optimierung mit zwei Phasen. In Phase eins wird die minimale Besetzungstärke je Qualifikationsgruppe ermittelt, um einen bestimmten Servicelevel zu erreichen. Hierzu wird der Algorithmus von Pot, Bhulai und Koole [308] genutzt, der Lagrange Relaxation verwendet. Dieser wird mit einem Schnittebenenverfahren von Cezik und L'Ecuyer [67] verglichen, welches sich am Verfahren von Atlason, Epelman und Henderson [12] orientiert. Lagrange Relaxation liefert fast die gleichen Ergebnisse, wie das Schnittebenenverfahren. Letzteres benötigte an den zehn getesteten Problemstellungen jedoch bis zu dreizehnmal mehr Rechenzeit, da die Schnitte deutlich mehr Simulationsläufe des Call Centers erfordern. In Phase zwei der zyklischen Optimierung erfolgt dann die Modellierung optimaler Schichten. Zur Lösung des zyklischen Vorgehens mit IP wird der Solver SA-Opt der Universität Amsterdam verwendet. Über den Solver selbst sind leider keine Details bekannt. Zwar garantiert das zyklische Vorgehen im Gegensatz zur integrierten Optimierung keine Optimalität – das Ergebnis ist jedoch nur 8% vom Optimum entfernt. Der Versuch, das in Phase eins angewendete Verfahren für die integrierte Optimierung zu erweitern, ist aufgrund der Komplexität der Problemstellung gescheitert. Dabei ist das Problem mit 14 Zeitintervallen à 1 Stunde, 2 Qualifikationstypen (somit 3 Teams bzw. Arbeitsplätze) und 35 Mitarbeitern noch recht moderat.

Klinkert löst 2008 [216] MIP-Probleme für das Days-Off Scheduling mit Hilfe von CPLEX 11.0. Diese Probleme stammen aus dem Ground Handling eines Flughafens und umfassen 500–800 Mitarbeiter, 100–300 zu besetzende Schichten pro Tag sowie 50 verschiedene Qualifikationskombinationen. Daher ergeben sich meist Problemstellungen mit etwa 100.000 binären Variablen. Bevor der Solver gestartet wird, werden zusätzliche Nebenbedingungen eingefügt, um auf besonders kritische Situationen zu reagieren. Z.B. wenn

in einer Schicht die Differenz zwischen Personalverfügbarkeit und -bedarf sehr gering ist. Bzgl. der Laufzeit wurde kein konkreter Wert angegeben. Es ist lediglich bekannt, dass CPLEX 11.0 einige Stunden benötigt, um ein Problem zu lösen.

Zahlreiche Tests wurden von Bonamia u.a. ebenfalls in 2008 [40] an einer Vielzahl konvexer MINLP-Problemstellungen durchgeführt. Dabei wurden folgende Erkenntnisse bei einigen GAMS-Solvern gewonnen: CoinBonmin-Hyb ist grundsätzlich zu bevorzugen, da es die meisten Probleme innerhalb von drei Stunden Rechenzeit löst. Nichts desto trotz gibt es Fälle, in denen CoinBonmin-OA oder CoinBonmin-BB schneller als CoinBonmin-Hyb waren. Für nichtkonvexe MINLPs wird empfohlen, CoinBonmin-BB zu verwenden. Im Vergleich zu den drei Problemen dieser Arbeit sind die Probleme von Bonamia u.a. erheblich kleiner.

Verschiedene Varianten eines Branch & Price-Verfahrens werden von Maenhout und Vanhoucke [253] an einer Vielzahl von Dienstplanungsproblemen der NSPLib [392] erprobt. Unter anderem wurden verschiedene Branching-Strategien untersucht, wobei die 0/1-Strategie am besten abschnitt. Auch konnte gezeigt werden, dass der Algorithmus deutlich verbessert wird, wenn eine geeignete Startlösung über eine vorgeschaltete Heuristik ermittelt wird. Diverse Techniken zur Reduzierung des Lösungsbaums wurden ebenfalls untersucht. Bei den durchgeführten Experimenten fand man das Ergebnis meist nach wenigen Minuten.

Der Vollständigkeit halber ist an dieser Stelle die Arbeit von Prüm aus dem Jahre 2006 [312] zu einem Einsatzplanungsproblem im Handel erwähnt. Für eine detaillierte Beschreibung sei auf Kapitel 3.3.5 verwiesen. Dort befinden sich auch Ausführungen zur Arbeit von Stolletz, der 2010 [358] CPLEX erfolgreich angewendet hat.

Insgesamt zeigt sich, dass klassische OR-Verfahren recht häufig eingesetzt werden. Meist werden die Problemstellungen jedoch modifiziert, um diese Verfahren anwenden zu können. Einige Autoren gehen auch dazu über, das eigentliche Problem zu relaxieren und das so gewonnene Ergebnis mit Hilfe von Heuristiken zu bearbeiten. Dies jedoch nicht immer mit großem Erfolg (siehe Prüm [312]). Ein derartiges Vorgehen wird in dieser Arbeit nicht weiter verfolgt. Denn die Problemstellung von Prüm, wo er z.T. nur recht schlechte Ergebnisse erzielen konnte, kommt den eigenen Problemen am nächsten. Zudem haben die meisten eigenen Probleme im Gegensatz zur Arbeit von Prüm eine Vielzahl an Arbeitsplätzen. Beim Logistikdienstleister z.B. ist das Ergebnis des relaxierten Problems (alle Nichtlinearitäten sind dort entfernt) absolut unbrauchbar. Der Plan hat extrem viele Arbeitsplatzwechsel. Hierfür eine Heuristik zu erstellen, die die entfernten Nebenbedingungen bestmöglich berücksichtigt, ist kaum möglich, da die Komplexität des Problems gerade aus der Reduzierung der Arbeitsplatzwechsel bei gleichzeitiger Minimierung der Besetzungsabweichungen (inkl. Qualifikationen etc.) liegt.

4.5.3 Anpassung an hochflexibles Workforce Management

Wie bereits erwähnt, soll in diesem Kapitel nur die Anpassung an die untertägige Personaleinsatzplanung in der Logistik in GAMS geschildert werden. Denn die Erkenntnisse, die hinsichtlich der Anwendbarkeit ganzzahliger Optimierungsverfahren beim Logistikproblem gewonnen werden, sind auf die beiden anderen Problemstellungen übertragbar. Wie das Logistikproblem sind auch das Handels- und British Telecom-Problem durch eine Vielzahl an Dimensionen und Entscheidungsvariablen gekennzeichnet. Zudem treten bei beiden Problemen sehr komplexe Nebenbedingungen und z.T. Nichtlinearitäten in der Zielfunktion auf.

4.5.3.1 Untertägige Personaleinsatzplanung in der Logistik

In diesem Kapitel werden zwei Modelle für das Logistikproblem vorgestellt. Zum einen handelt es sich um das MIP-Modell, bei dem das originale Problem Logistik_Complete stark vereinfacht abgebildet wird. Zum anderen um das MINLP-Modell. Dort ist die originale Problemstellung Logistik_Complete dargestellt. Da sich die dort gemachten Erkenntnisse auch auf die Varianten mit Stammarbeitsplatzregelung übertragen lassen, wird auf die Modellierung von Logistik_S_Complete als MIP- und MINLP-Modell verzichtet.

MIP-Modell Das MIP-Modell des Problems Logistik_Complete wird durch die nachfolgend aufgelisteten Eigenschaften charakterisiert:

- Einsatzplanung in definierten Zeitintervallen.
- Jeder Mitarbeiter wird nur einem Arbeitsplatz pro Zeitintervall zugewiesen.
- Berücksichtigung von Verfügbarkeiten der Mitarbeiter aufgrund der ganztägigen Monatsplanung.
- Berücksichtigung von Qualifikationen.
- Minimierung von Über-/Unterdeckung der Arbeitsplatzbesetzungen.
- Keine Berücksichtigung von erhöhten Strafpunkten bei einer Arbeitsplatzzuweisung, während dort ein Bedarf von null herrscht.
- Keine Berücksichtigung von Arbeitsplatzwechseln.

In den letzten beiden Punkten weicht das MIP-Modell von der ursprünglichen Problemstellung des Logistikdienstleisters ab. Aufgrund dieser Abweichungen wird das MIP-Modell von Logistik_Complete als Logistik_Complete_MIP bezeichnet. Die Berücksichtigung der letzten beiden Nebenbedingungen, die im Modell ausgelassen sind, würde jeweils zur Nichtlinearität der Zielfunktion führen.

Bei der in Logistik_Complete_MIP gewählten Form der Modellierung war es erforderlich, einen weiteren Dummy-Arbeitsplatz einzuführen, um den Überhang an Mitarbeitern in Phasen der Überdeckung aufzufangen. Arbeitsplätze werden bei der Überdeckung also mit qualifizierten und verfügbaren Mitarbeitern besetzt und der Überhang an Mitarbeitern wird auf den Dummy-Arbeitsplatz geplant. Ohne die Einführung des Dummy-Arbeitsplatzes würde der Überhang an Mitarbeitern nicht eingeplant werden. Die Mitarbeiter hätten frei, was bei dieser Problemstellung jedoch nicht erfolgen soll. Bei einem Mitarbeiter, der wegen Überdeckung auf den Dummy-Arbeitsplatz geplant wurde, kann man jedoch nicht prüfen, ob er tatsächlich auf einem Arbeitsplatz mit einem Bedarf größer Null unterstützt werden kann oder ob er wegen fehlender Qualifikationen auf einem Arbeitsplatz mit einem Bedarf von Null geplant worden wäre. Als Konsequenz aus dem Dummy-Arbeitsplatz muss daher die Regel entfallen, nach der eine Überdeckung bei einem Arbeitsplatz ohne Personalbedarf mit höheren Fehlerpunkten bewertet wird. Neben dem Dummy-Arbeitsplatz gibt es auch Dummy-Mitarbeiter, die auf nicht besetzte Arbeitsplätze verplant werden, um so eine Unterdeckung abzufangen. Anhand der verplanten Dummy-Mitarbeiter und des Dummy-Arbeitsplatzes kann man also die Über- und Unterdeckung pro Zeitintervall ablesen. Um die Linearität der Zielfunktion zu gewährleisten, musste die Berücksichtigung von Arbeitsplatzwechseln ebenfalls entfallen. Nachfolgend wird das Modell Logistik_Complete_MIP kurz charakterisiert, wobei statistische Daten aus GAMS aufgeführt sind.

- Logistik_Complete_MIP: Eine Vereinfachung der originalen Problemstellung mit 560 Zeitintervallen, 65 Mitarbeitern und 9 Arbeitsplätzen. Zur Vermeidung der Nichtlinearität musste auf die Berücksichtigung von Arbeitsplatzwechseln und erhöhten Fehlerpunkten bei einem Bedarf von null verzichtet werden.
 - Zeilen der Matrix: 57.793
 - Spalten der Matrix: 517.441
 - Elemente ungleich 0 in der Matrix: 1.205.839
 - Nicht lineare Elemente in der Matrix: 0

Im folgenden Listing ist der GAMS-Code für das Modell Logistik_Complete_MIP abgebildet. Aufgrund der eigentlichen Länge wurde er hier stark reduziert.

Listing 4.1: Logistik_Complete_MIP (gekürzt)

```

1: Sets m mitarbeiter / MA1, MA2, [...] DummyMA /
2:      a arbeitsplatz / AP1, AP2, [...] DummyAP /
3:      t zeit (15 Min) / 1*672 /
4:
5: Table verfügbar(m,t) Verfügbarkeit
6:      1 2 3 4 [...]
7: MA1 0 1 1 1 [...]
8:
9: DummyMA 0 1 1 1 [...]
10:
11: Table demand(a,t) Bedarf
12:      1 2 3 4 [...]
13: AP1 0 0 1 1 [...]
14:
15: DummyAP 0 1 0 0 [...]
16:
17: Table ErrorDemand(a,t) Fehlerpunkte (Besetzungen)
18:      1 2 3 4 [...]
19: AP1 15 15 15 15 [...]
20:
21: DummyAP 0 0 0 0 [...]
22:
23: Table ErrorQuali(a,m) Fehlerpunkte (Qualifikationen)
24:      MA1 MA2 DummyMA [...]
25: AP1 3000 0 0 [...]
26:
27: DummyAP 0 0 0 [...]
28:
29: Table ErrorDummy(m,a,t) Fehlerpunkte bei Falschplanung
30:      1 2 3 4 [...]
31: MA1.AP1 0 0 0 0 [...]
32:
33: DummyMA.AP9 9999 9999 0 0 [...]
34: DummyMA.DummyAP 0 0 0 0 [...]
35:
36: Positive Variable abw(t,a) Abweichung vom Bedarf;
37: Binary Variable x(t,a,m) Entscheidungsvariable;
38: Variable z Zielfunktion (minimiere);
39:
40: Equations verfügbarkeit(t,m) Verfügbarkeit der MA
41:      abweichung(t,a) Besetzungsabweichung
42:      cost Kosten;

```

```

43:
44:   verfu_gbarkeit (t ,m) .. sum (a ,x (t , a ,m) )=e=verfu_gbar (m, t) ;
45:   abweichung (t , a) .. sum (m ,x (t , a ,m) )-abw (t , a)=e=demand (a , t) ;
46:
47:   cost .. z =e= sum ((t , a) ,ErrorDemand (a , t) *abw (t , a)) +
                sum ((t , a ,m) ,ErrorDummy (m , a , t) *x (t , a ,m)) +
                sum ((t , a ,m) ,ErrorQuali (a ,m) *x (t , a ,m)) +
48:
49:   Model modell1 / all /;
50:   Solve modell1 using mip minimizing z ;

```

Nachfolgend wird der GAMS-Code für das Modell Logistik_Complete_MIP zeilenweise erläutert.

- 1–3: Definition der Mengen (Mitarbeiter, Arbeitsplätze und Zeitintervalle).
- 5–9: Festlegung der Verfügbarkeit von Mitarbeitern (1 = verfügbar). Im ersten Zeitintervall ist niemand verfügbar, da dies die Berechnung von Arbeitsplatzwechseln erleichtert.
- 11–15: Der Personalbedarf je Arbeitsplatz und Zeitintervall wird festgelegt.
- 17–21: Fehlerpunkte für Besetzungsabweichungen.
- 23–27: Fehlerpunkte bei Qualifikationsverletzungen (0 = Mitarbeiter ist qualifiziert). Dummy-Mitarbeiter sind für alle Arbeitsplätze qualifiziert.
- 29–34: Dummy-Mitarbeiter sollen Unterdeckungen abfangen. Jedoch nur dann, wenn es auch tatsächlich eine Unterdeckung gibt. Dies wird über Fehlerpunkte geregelt.
- 36–38: Einführung verschiedener Variablen.
- 40–42: Benennung von Gleichungen bzw. Ungleichungen.
- 44: Mitarbeiter werden nur eingeplant, wenn sie verfügbar sind. Dann müssen sie immer einem Arbeitsplatz zugewiesen werden und haben nicht frei. Zudem darf jeder Mitarbeiter nur einen Arbeitsplatz pro Zeitintervall haben.
- 45: Nebenbedingung für die Reduzierung der Besetzungsabweichung.
- 47: Zielfunktion: Fehler für Besetzungsabweichungen, Fehler, wenn ein Dummy-Mitarbeiter ohne Notwendigkeit eingeplant wurde und Qualifikationsfehler.

- 49: Benennung des Modells sowie der zu berücksichtigenden Gleichungen und Ungleichungen.
- 50: Aufruf des Solvers.

MINLP-Modell An dieser Stelle wird das Problem Logistik_Complete ohne Vereinfachungen als MINLP modelliert. Das Modell entspricht also exakt der Problemstellung des Logistikdienstleisters für die komplette Woche. Es wird durch folgende Eigenschaften charakterisiert, welche sich lediglich in den letzten beiden Punkten vom Modell Logistik_Complete_MIP unterscheiden.

- Einsatzplanung in definierten Zeitintervallen.
- Jeder Mitarbeiter wird nur einem Arbeitsplatz pro Zeitintervall zugewiesen.
- Berücksichtigung von Verfügbarkeiten der Mitarbeiter aufgrund der ganztägigen Monatsplanung.
- Berücksichtigung von Qualifikationen.
- Minimierung von Über-/Unterdeckung der Arbeitsplatzbesetzungen.
- Berücksichtigung von erhöhten Strafpunkten bei einer Arbeitsplatzzuweisung, während dort ein Bedarf von null herrscht.
- Minimierung von Arbeitsplatzwechseln.

Bei diesem Modell wurde die Berücksichtigung von Über- und Unterdeckungen im Personaleinsatz in die Zielfunktion gezogen. Hierdurch können der Dummy-Arbeitsplatz und die Dummy-Mitarbeiter, wie sie im MIP-Modell verwendet wurden, entfallen. Diese Art der Modellierung führt nun dazu, dass eine Überdeckung bei einem Bedarf von null mit höheren Strafpunkten bewertet werden kann. Ebenso werden nun Arbeitsplatzwechsel in der Zielfunktion berücksichtigt. Es wurden drei Varianten des MINLP-Modells erstellt, die sich in der Anzahl der Zeitintervalle unterscheiden. Das erste Modell entspricht exakt der Problemstellung des Logistikdienstleisters (Logistik_Complete). Die anderen beiden sind deutlich kleiner gehalten und enden in ihrer Bezeichnung auf die Anzahl an Zeitintervallen, die sie umfassen. Die beiden kleineren Probleme dienen der Beurteilung, bis zu welchen Problemgrößen MINLP-Verfahren angewendet werden können. Andere Lösungsverfahren werden an diesen beiden Problemen nicht untersucht. Nachfolgend werden die drei Varianten kurz beschrieben, wobei auch die statistischen Werte aus GAMS angegeben werden, die Anhaltspunkte für die Größe und Komplexität der Problemstellung geben.

- Logistik_Complete: Die originale Problemstellung mit 560 Zeitintervalle, 65 Mitarbeitern und neun Arbeitsplätzen.
 - Zeilen der Matrix: 43.681
 - Spalten der Matrix: 393.121
 - Elemente ungleich 0 in der Matrix: 786.241
 - Nicht lineare Elemente in der Matrix: 393.120
 - Code Länge in GAMS: 7.632.722

- Logistik_Complete_20: Eine reduzierte Problemstellung. Es gibt weiterhin 65 Mitarbeiter und neun Arbeitsplätze. Die originale Problemstellung wurde jedoch bis auf die ersten 20 Zeitintervalle gekürzt. Daher sind von den 65 Mitarbeitern nur sechs anwesend und können verplant werden.
 - Zeilen der Matrix: 1.301
 - Spalten der Matrix: 11.701
 - Elemente ungleich 0 in der Matrix: 23.401
 - Nicht lineare Elemente in der Matrix: 11.700
 - Code Länge in GAMS: 224.251

- Logistik_Complete_4: Eine reduzierte Problemstellung. Die Problemstellung umfasst die ersten vier Zeitintervalle von Logistik_Complete. Es gibt 65 Mitarbeiter, von denen lediglich 2 verfügbar sind und somit verplant werden können. Die Arbeitsplätze wurden von neun auf die ersten zwei reduziert.
 - Zeilen der Matrix: 261
 - Spalten der Matrix: 521
 - Elemente ungleich 0 in der Matrix: 1.041
 - Nicht lineare Elemente in der Matrix: 520
 - Code Länge in GAMS: 9.372

Auch der GAMS-Code für Logistik_Complete ist nachfolgend stark gekürzt.

Listing 4.2: Logistik_Complete (gekürzt)

```

1:  Sets  m  mitarbeiter / MA1, MA2, [...] MA65 /
2:        a  arbeitsplatz / AP1, AP2, [...] AP9 /
3:        t  zeit (15 Min) / 1*672 /
4:
5:  Table  verfügbar(m,t)      Verfügbarkeit
6:        1      2      3      4      [...]
7:  MA1    0      1      1      1      [...]
8:        [...]
9:  MA9    0      1      1      1      [...]
10:
11: Table  demand(a,t)        Bedarf
12:        1      2      3      4      [...]
13:  AP1    0      0      1      1      [...]
14:        [...]
15:  AP9    0      1      2      4      [...]
16:
17: Table  ErrorDemand(a,t)    Fehlerpunkte (Besetzungen)
18:        1      2      3      4      [...]
19:  AP1    30     30     15     15     [...]
20:        [...]
21:  AP9    0      15     15     15     [...]
22:
23: Table  ErrorQuali(a,m)     Fehlerpunkte (Qualifikationen)
24:        MA1     MA2     MA3     [...]
25:  AP1    3000    0      0      [...]
26:        [...]
27:  AP9    0      3000    0      [...]
28:
29: Scalars ErrorRot Fehlerpunkte (Arbeitsplatzwechsel) / 1 /
30:
31: Binary Variable  x(t,a,m)      Entscheidungsvariable;
32: Variable         z              Zielfunktion (minimiere);
33:
34: Equations  verfügbarkeit(t,m) Verfügbarkeit der MA
35:            cost                Kosten;
36:
37: verfügbarkeit(t,m) .. sum(a, x(t,a,m)) =e= verfügbar(m,t);
38:
39: cost .. z =e= sum((t,a),ErrorDemand(a,t)*abs(demand(a,t)-
                sum(m,(x(t,a,m)))))+sum(t,sum((a,m),
                x(t,a,m)*quali(a,m)))+sum((t,m),

```

```
sum(a, ErrorRot / 2 * (abs(x(t, a, m) - x
(t + 1, a, m)))) - 288;
40:
41: Model modell1 / all / ;
42: Solve modell1 using minlp minimizing z;
```

Nachfolgend wird der zuvor abgebildete GAMS-Code für Logistik_Complete zeilenweise erläutert.

- 1-3: Definition der Mengen (Mitarbeiter, Arbeitsplätze und Zeitintervalle).
- 5-9: Festlegung der Verfügbarkeit von Mitarbeitern (1 = verfügbar). Im ersten Zeitintervall ist niemand verfügbar, da dies die Berechnung von Arbeitsplatzwechseln erleichtert.
- 11-15: Der Personalbedarf je Arbeitsplatz und Zeitintervall wird festgelegt.
- 17-21: Fehlerpunkte für Besetzungsabweichungen.
- 23-27: Fehlerpunkte bei Qualifikationsverletzungen (0 = Mitarbeiter ist qualifiziert). Dummy-Mitarbeiter sind für alle Arbeitsplätze qualifiziert.
- 29: Fehlerpunkte für einen Arbeitsplatzwechsel.
- 31-32: Einführung verschiedener Variablen.
- 34-35: Benennung von Gleichungen bzw. Ungleichungen.
- 37: Mitarbeiter werden nur eingeplant, wenn sie verfügbar sind. Dann müssen sie immer einem Arbeitsplatz zugewiesen werden und haben nicht frei. Zudem darf jeder Mitarbeiter nur einen Arbeitsplatz pro Zeitintervall haben.
- 39: Zielfunktion: Fehler für Besetzungsabweichungen, wobei Arbeitsplätze mit einem Bedarf von null mit höheren Fehlerpunkten im jeweiligen Zeitintervall belegt sind. Zudem Qualifikationsfehler und Fehler durch Arbeitsplatzwechsel.
- 41: Benennung des Modells sowie der zu berücksichtigenden Gleichungen und Ungleichungen.
- 42: Aufruf des Solvers.

4.6 Konstruierende Verfahren

In diesem Kapitel wird zunächst der grundlegende Ablauf Konstruierender Verfahren (KV) skizziert. Anschließend werden in Kapitel 4.6.2 Anwendungen dieses Verfahrens an verschiedenen Planungsproblemen erörtert, die in der Literatur zu finden sind. KV werden nicht am Logistikproblem erprobt. Für alle Varianten des Handelsproblems kann auf ein KV eines Softwareherstellers für WFM zurückgegriffen werden. Zusätzlich bestand die Möglichkeit, das Verfahren eines weiteren Softwareherstellers an den Varianten mit nur einem Arbeitsplatz zu erproben. Beide Verfahren werden in Kapitel 4.6.3.1 erläutert. Eine eigene Implementierung eines KV erfolgt für das British Telecom-Problem. Diese ist abschließend in Kapitel 4.6.3.2 dargestellt.

4.6.1 Grundlegender Ablauf

KV können sehr facettenreich gestaltet sein, weshalb an dieser Stelle lediglich exemplarisch auf den Greedy Algorithmus eingegangen wird.

Zu den KV gehört klassischerweise der Greedy Algorithmus. „A greedy algorithm always makes the choice that looks best at the moment.“ [82, S. 370] Der Algorithmus ist also „gierig“ und wählt schrittweise den Folgezustand aus, der den größten Nutzen bringt. Bekannte Beispiele für einen Greedy Algorithmus sind die Algorithmen von Kruskal [231], Prim [311] und Dijkstra [98].

Da es bei KV nicht notwendig ist, in jedem Schritt eine Fitnessfunktion zu berechnen, sind sie meist sehr schnell. Die Fitnessfunktion wird gewöhnlich nur einmal berechnet, wenn das Verfahren beendet ist.

4.6.2 Anwendungen in der Literatur

Mitte der 60er Jahre wurden bereits KV von Broder [45] und Cole [74] für die universitäre Stundenplanerstellung erstellt. Derartige Verfahren wurden Ende der 60er Jahre von Wood [421] sowie von Foxley und Lockyer [124] in individuelle Softwarelösungen für Universitäten implementiert.

Kilby [211] wendete auf zwei Problemstellungen des Staff Scheduling mit je 100 erzeugten Testinstanzen ein KV an, bei der die Auswahl des zuzuweisenden nächsten Elementes über ein „Maß des Bedauerns“ erfolgt. Der Planungshorizont beträgt 14 Tage, wobei den 30 bzw. 22 Mitarbeitern 200 Aufträge zugewiesen werden müssen. Jeder Auftrag ist durch ein Startdatum, eine Startzeit und eine Dauer charakterisiert. Qualifikationen werden

über drei Eigenschaften mit einer bestimmten Punktzahl abgebildet. Eine Restriktion der Problemstellungen ist, dass einem Mitarbeiter pro Tag nur ein Auftrag zugewiesen werden kann. Würde man diese Restriktion aufweichen und kleinere Zeitintervalle für Zuweisungen zulassen, so erhielte man eine untertägige Planung.

Di Gasparo testet 2003 [96] an einem moderat dimensionierten Schedulingproblem mit beschränkten Ressourcen zwei Heuristiken, die auf einem KV basieren. Zusätzlich implementiert er TS, welches die Ergebnisse des KV als Startlösung nimmt. Im Vergleich schneidet die hybride Variante mit TS bis zu 45% besser ab als die alleinige Verwendung des KV.

Für die Planung des Technikereinsatzes wurden KV beim ROADEF-Wettbewerb in 2007 [87] verwendet. Von zwölf Teams ist bekannt, welches Verfahren sie beim Wettbewerb einsetzten. Acht davon setzten auf KV. Nähere Erläuterungen zum ROADEF-Wettbewerb und den verwendeten Verfahren wurden bereits in Kapitel 3.4.5 gemacht.

Ein weiteres Verfahren, das in dieser Arbeit auch an den Problemen `Handel_1AP` und `Handel_1AP_Jan` angewendet wird, wurde von Sauer und Schumann [334] entwickelt. Für eine Beschreibung des Verfahrens sei nachfolgend auf Kapitel 4.6.3.1 verwiesen.

4.6.3 Anpassung an hochflexibles Workforce Management

In diesem Kapitel werden lediglich Ausführungen zur automatischen Arbeitszeitmodell-erstellung im Handel und zur Kombination von Touren- und Personaleinsatzplanung gemacht. Beim Logistikproblem wird ein KV im Rahmen eines MAS erprobt, weshalb an dieser Stelle auf Kapitel 4.15.3.2 verwiesen wird.

Die beiden KV, die am Handelsproblem angewendet werden, stammen jeweils aus kommerziellen Softwarelösungen. Sie sind nicht in der Lage, alle Nebenbedingungen des Logistikproblems abzubilden und sind dort somit nicht einsetzbar. Zudem kann der Ansatz von Sauer und Schumann nur mit einem Arbeitsplatz umgehen, sodass ihr Verfahren nur bei `Handel_1AP` und `Handel_1AP_Jan` in Frage kommt. Die beiden kommerziellen Lösungen sind zudem nicht für die Tourenplanung geeignet und daher nicht auf das Problem der British Telecom anwendbar.

Beim British Telecom-Problem wird ein eigenes Verfahren entwickelt. Dies ist nicht beim Logistik- und Handelsproblem anwendbar. Es wurde vor allem auf regionale Zuordnungen von Aufträgen zu Servicecentern und verschiebbare Ausführungszeiten der Aufträge ausgerichtet. All dies ist bei den beiden Problemstellungen nicht zu finden. Allgemein zeigt sich bereits hier, dass KV scheitern können, wenn sie nicht explizit für bestimmte Problemstellungen entwickelt wurden. Kleine Änderungen eines Problems können genügen, um ein zuvor sehr gut anwendbares KV unbrauchbar zu machen.

4.6.3.1 Automatische Arbeitszeitmodellerstellung im Handel

Für die Probleme Handel_1AP und Handel_1AP_Jan stehen die beiden folgenden KV zur Verfügung. Da das erste Verfahren nur Probleme mit einem Arbeitsplatz unterstützt, können Handel_2AP und Handel_2AP_Jan lediglich vom zweiten Verfahren bearbeitet werden.

Ansatz von Sauer und Schumann Dieses KV stammt von Schumann und Sauer [334] und ist als Modul ActivePeP Bestandteil einer kommerziellen Softwarelösung für die Personaleinsatzplanung mit automatischer Arbeitszeitmodellerstellung im Handel. Es wird in dieser Arbeit als KV 1 bezeichnet. Der Ablauf des Lösungsverfahrens ist nachfolgend dargestellt.

Algorithmus 1 KV 1 von Sauer und Schumann bei der automatischen Arbeitszeitmodellerstellung im Handel

```

1: while Noch nicht alle Tage geplant do
2:   Bestimme den schwersten Tag
3:   Bestimme den maximalen Zuweisungszeitraum
4:   Ermittle die möglichen Mitarbeiter
5:   Ordne die möglichen Mitarbeiter
6:   if Mitarbeiter existiert then
7:     Berechne das Intervall für die Zuweisung
8:     Weise den Mitarbeiter zu
9:     Aktualisiere alle Daten
10:  else
11:    Reduziere den Personalbedarf
12:  end if
13: end while

```

Sauer und Schumann erstellen den Einsatzplan unter der Annahme, dass nicht immer genügend Personal zur Verfügung steht und Unterdeckungen im Personaleinsatz somit nicht immer vermeidbar sind. Daher wird eine Heuristik verwendet, die engpassorientiert arbeitet. Die wichtigsten Entscheidungen, die der Algorithmus treffen muss, ist die Auswahl des zu planenden Tages und die Auswahl der passenden Mitarbeiter. Die Gewichtung der weichen Nebenbedingungen geschieht über Fehlerpunkte. Das Verfahren läuft in drei Schritten ab.

1. Auswahl des Tages: Ein Tag kann nur dann ausgewählt werden, wenn dort ein Bedarf an Personal besteht. Wenn mehr als ein Tag in Frage kommt, wird bestimmt, wie schwer die einzelnen Tage zu besetzen sind. Hierzu wird der Bedarf an Sollstunden des Tages in das Verhältnis zu den Sollstunden der entsprechenden

Woche gesetzt. Der Tag mit dem verhältnismäßig größten Bedarf wird zuerst genommen. Bei Gleichstand wird zufällig entschieden. Anschließend wird der größte zusammenhängende Zeitraum mit einem Bedarf größer als null ermittelt, der noch nicht gedeckt ist. Für diesen Zeitraum wird nun ein Mitarbeiter gesucht.

2. Auswahl des Mitarbeiters: Grundsätzlich stehen nur die Mitarbeiter zur Verfügung, die noch nicht zugewiesen wurden und an dem Tag nicht abwesend sind. Die Mitarbeiter werden nun bewertet und in eine Reihenfolge gebracht. Bei der Bewertung werden Nebenbedingungen wie die wöchentliche Arbeitszeit und die minimale oder maximale Länge der Arbeitszeit berücksichtigt. Es kann daher sein, dass Mitarbeiter wegen Verstößen gegen die Nebenbedingungen aus der Liste entfernt werden. Der am besten bewertete Mitarbeiter wird zugewiesen.
3. Im nächsten Schritt wird die maximal mögliche Arbeitszeit des Mitarbeiters innerhalb des in Schritt eins ermittelten Zeitraums bestimmt. Ist diese kleiner als der Zeitraum aus Schritt eins, so wird zusätzlich die beste Lage der Arbeitszeit berechnet. Anschließend erfolgt die Zuweisung und alle Daten werden aktualisiert.

Ansatz eines Softwareherstellers Dieses KV, welches nachfolgend als KV 2 bezeichnet wird, ist ein Modul eines Softwareherstellers für das WFM. Die Software ist branchenneutral aufgestellt und etwa 300 Kunden des Anbieters, auch aus dem Handel, nutzen dieses Modul zur automatischen Erstellung von Einsatzplänen. Der Anbieter darf in dieser Arbeit nicht namentlich genannt werden. Ebenso dürfen keine Details über das Lösungsverfahren veröffentlicht werden. Es muss in dieser Arbeit daher als black Box betrachtet werden.

Mit dem Verfahren können beliebig viele Arbeitsplätze, Zeitintervalle und Mitarbeiter berücksichtigt werden. Zudem ist es möglich, etwa 50 Nebenbedingungen mittels Parametrierung einzufügen und mit Fehlerpunkten zu bewerten. In einem gesonderten Dialog wird die Berechnung gestartet. Dort sind auch alle Verstöße gegen Nebenbedingungen und der Fitnesswert ersichtlich.

4.6.3.2 Kombination von Touren- und Personaleinsatzplanung

Beim KV für die Kombination von Touren- und Personaleinsatzplanung werden zum einen harte Nebenbedingungen wie Verfügbarkeiten oder Qualifikationen eingehalten. Zum anderen soll die Zuteilung der Aufträge auf Techniker in Abhängigkeit von ihrer noch verfügbaren Kapazität und der Entfernung des Auftrags von ihrem Servicecenter erfolgen. Die exakten Entfernungen von Auftrag zu Auftrag finden in der Berechnung der Kapazität keine Anwendung, da dies bereits über die regionale Zuordnung des Auftrags abgebildet wird – also über die Entfernung des Auftrags zum Servicecenter des

Technikers. Dieses Verfahren wird als KV 3 bezeichnet. Der Ablauf ist in Algorithmus 2 dargestellt.

Algorithmus 2 KV 3 bei der Kombination von Touren- und Personaleinsatzplanung

```

1: Setze die Kapazität  $Kap_e$  aller Techniker  $e$  auf 0
2: repeat
3:   Wähle einen zufälligen Auftrag  $j$ 
4:   Schreibe alle für Auftrag  $j$  qualifizierten Techniker in die Liste
     Quali
5:   Berechne für jeden Techniker die Fahrzeit  $ct_e$  von seinem
     Servicecenter  $i$  zum Auftrag  $j$ 
6:   while Quali ist nicht leer do
7:     Wähle Techniker  $e$  mit der geringsten Fahrzeit  $ct_e$  zum Auftrag  $j$ 
8:     Entferne Techniker  $e$  aus Quali
9:     Berechne die Bearbeitungsdauer  $rd_j$  durch Techniker  $e$ 
10:    if  $Kap_e + rd_j < 8$  Stunden then
11:      Füge Auftrag  $j$  an das Ende der Permutation von Techniker  $e$  ein
12:      Entferne alle Techniker aus Quali
13:      Aktualisiere  $Kap_e$ 
14:    end if
15:  end while
16:  if Auftrag konnte nicht zugewiesen werden then
17:    Wähle einen Techniker  $e$  aus, der für Auftrag  $j$  qualifiziert ist
18:    Füge Auftrag  $j$  an das Ende der Permutation von Techniker  $e$  ein
19:    Aktualisiere  $Kap_e$ 
20:  end if
21: until Alle Aufträge sind zugewiesen

```

4.7 Constraint Logic Programming

Constraint Logic Programming (CLP) – oder auch Logikprogrammierung mit Nebenbedingungen – wird in dieser Arbeit nicht selbst implementiert. Für das British Telecom-Problem liegen jedoch Ergebnisse mit CLP von Azarmi und Abdul-Hameed [18], von Yang [428] sowie von Kokkoras u.a. [217], [218], [330] vor. Es werden demnach keine Experimente am Logistik- oder Handelsproblem durchgeführt. Dieses Kapitel soll daher nur einen kleinen Überblick über CLP geben.

Zunächst wird der grundlegende Ablauf erläutert, worauf hin auf Anwendungen in der Literatur eingegangen wird. In Kapitel 4.7.3 werden dann die CLP-Ansätze für das British Telecom-Problem dargestellt.

4.7.1 Grundlegender Ablauf

CLP entstand Mitte der achtziger Jahre aus der Zusammenführung zweier deklarativer Paradigmen: Constraint Solving und Logic Programming [128, S. 3]. Hofstedt und Wolf definieren den Begriff Constraint wie folgt: „Aus formaler Sicht sind Constraints spezielle prädikatenlogische Formeln, mit deren Hilfe man Eigenschaften von Problemen und deren Lösungen beschreibt. Das können z.B. Gleichungen oder Ungleichungen über Zahlen sein, aber auch andere Ausdrücke über Zahlen, Booleschen Werten oder beliebigen anderen Mengen wie Buchstaben oder Wörtern.“ [185, S. 53]

Nachfolgend wird beispielhaft der Ablauf von Propagierung und Domänenreduktion erläutert. Hierzu wird ein Beispiel von Freese [126, S. 9] herangezogen: gesucht sind ganze Zahlen für x, y und z , so dass $x, y, z \in [1, 3]$ und $y < z$, $x - y = 1$ und $x \neq z$ gilt. Auf Basis dieser Angaben kann ein Suchbaum erstellt werden (siehe Abbildung 4.8), wobei der Pfad von der Wurzel zum Blatt eine Belegung der Variablen x, y und z darstellt. Aus $x, y, z \in [1, 3]$ folgt, dass die Domänen auf $D = \{\{1, 2, 3\}, \{1, 2, 3\}, \{1, 2, 3\}\}$ beschränkt sind.

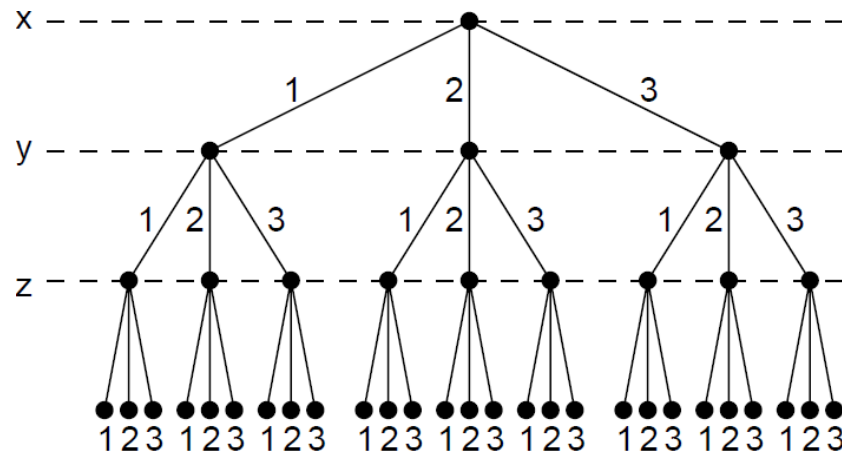


Abbildung 4.8: Variablenbelegung in Baumdarstellung [126, S. 9]

Die Lösungsidee ist, dass ein Suchraum mit Hilfe von Constraints aktiv eingeschränkt wird. Dies wird als Propagierung bzw. Constraintpropagierung bezeichnet. Es werden also Werte aus dem Bereich von Variablen entfernt, die in keiner Lösung vorkommen können [426, S. 11–12]. Im dargestellten Beispiel sieht die Propagierung wie folgt aus.

- Der Wert 1 für z kann wegen $y < z$ ausgeschlossen werden.
- Der Wert 3 für y kann wegen $y < z$ ausgeschlossen werden.
- Der Wert 1 für x kann wegen $x - y = 1$ ausgeschlossen werden.

Die Domänen wurden somit auf $D = \{\{2, 3\}, \{1, 2\}, \{2, 3\}\}$ eingeschränkt. Die Variablenbelegung nach der Domänenreduktion ist in Abbildung 4.9 dargestellt.

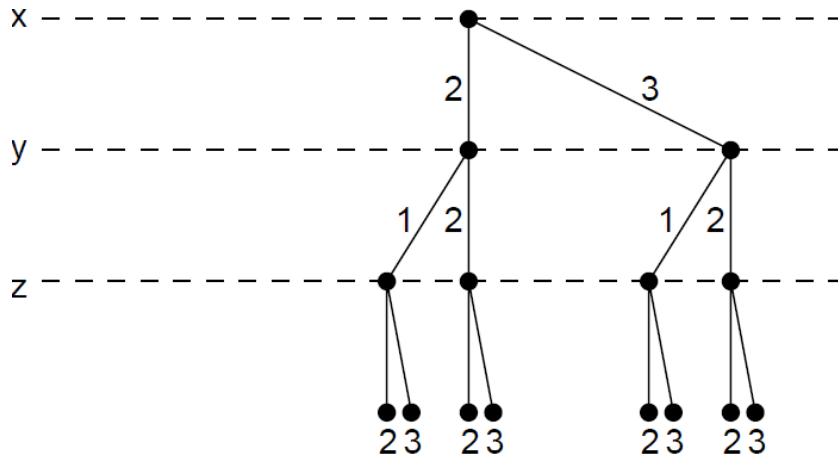


Abbildung 4.9: Variablenbelegung in Baumdarstellung nach Domänenreduktion [126, S. 11]

Um nun eine gültige Lösung zu finden, wird ein Wert aus der reduzierten Domäne einer Variablen gewählt. Für die Reihenfolge, mit der Werte ausgewählt werden, gibt es verschiedene Ansätze. Bei der lexikalischen Reihenfolge wird zuerst x mit 2 belegt. Durch eine erneute Propagierung liegen alle Variablen im Beispiel fest: $x = 2$, $y = 1$ und $z = 3$. Sollte die Festlegung von Variablen dazu führen, dass Constraints nicht erfüllbar sind oder Domänen leer sind, kann Backtracking eingesetzt werden. Dabei wird zunächst versucht, eine andere Auswahl zu treffen, die nicht falsch ist. Ist dies nicht möglich, wird die vorherige Festlegung rückgängig gemacht usw. Ist ein Punkt erreicht, an dem keine vorherige Festlegung rückgängig gemacht werden kann, existiert keine gültige Lösung. [126, S. 9–13] Backtracking kann demnach zur Suche nach weiteren Lösungen verwendet werden [333, S. 64].

Bei vielen Problemstellungen wie dem hochflexiblen WFM sucht man jedoch nicht bloß eine gültige Lösung. Vielmehr ist man an einer optimalen Lösung interessiert, der eine Fitnessfunktion zu Grunde liegt. Mit dem zuvor dargestellten Ansatz können alle gültigen Lösungen einer Problemstellung identifiziert werden. Bei äußerst vielen möglichen Lösungen (z.B. bei den in dieser Arbeit untersuchten Problemstellungen) ist ein systematisches Abarbeiten aller Lösungen nicht mehr praktikabel. In diesem Fall kann auf Heuristiken ausgewichen werden. Diese sind für die Auswahl der Variablen und der Werte in deren Domäne verantwortlich. Grundsätzlich gilt zu beachten: je aufwändiger die Propagierung, umso kleiner kann der Suchbaum werden. Da die Propagierung jedoch Rechenzeit beansprucht, muss ein geeigneter Mittelweg zwischen Suche und Propagierung gefunden werden. [15, S. 10–12]

4.7.2 Anwendungen in der Literatur

Eine kurze Einführung in CLP und dessen Anwendung im Bereich Scheduling und Planung geben Lever u.a. [244]. Für eine allgemeine Übersicht sei auf die Sammlung von Jampel verwiesen, in der er 911 Arbeiten zum Thema CLP auflistet [196].

Wie bereits erwähnt, wurde CLP auf zwei Varianten des British Telecom-Problems angewendet. Zunächst 1995 von Azarmi und Abdul-Hameed [18]. Anschließend von Yang [428] im Jahre 1996 und in mehreren Arbeiten von 1996 bis 2003 durch Kokkoras u.a. [217], [218], [330]. Die CLP-Ansätze dieser Autoren werden nachfolgend in Kapitel 4.7.3.1 dargestellt.

Im Rahmen des ConPlan-Projekts haben das DFKI und die SIEDA GmbH 1997 Constrainttechniken speziell für die Dienstplanung im Krankenhaus entwickelt [16]. Aus diesem Projekt ist der OC:Planner hervorgegangen – eine kommerzielle Anwendung, die Bestandteil des Lösungsportfolios der SIEDA GmbH ist und von mehr als 60 Krankenhäusern genutzt wird [375].

4.7.3 Anpassung an hochflexibles Workforce Management

In diesem Kapitel werden die CLP-Ansätze diverser Autoren für das British-Telecom-Problem dargestellt. Eigene Implementierungen am Logistik-, Handels- oder British Telecom-Problem werden nicht durchgeführt.

4.7.3.1 Kombination von Touren- und Personaleinsatzplanung

Nachfolgend werden drei Varianten von CLP erläutert, die am British Telecom-Problem erprobt wurden. Zunächst wird auf die beiden Varianten von Azarmi und Abdul-Hameed [18] eingegangen, worauf hin das Verfahren von Yang [428] erläutert wird.

Ansatz von Azarmi und Abdul-Hameed Azarmi und Abdul-Hameed wenden 1995 [18] verschiedene Varianten von CLP auf das Problem BT_Mod-250-118 an. Zudem untersuchen sie eine weitere Problemstellung der British Telecom mit 434 Aufträgen und 120 Technikern, deren Daten jedoch nicht zugänglich sind. Des Weiteren werden die CLP-Varianten an diversen Ausprägungen eines Vehicle Routing Problems (VRP) der Universität Strathclyde untersucht. Azarmi und Abdul-Hameed entscheiden sich für die Anwendung von CLP, da es umfangreiche Modellierungssprachen gibt, die mächtige Lösungsverfahren beinhalten. Laut Azarmi und Abdul-Hameed ist die Entwicklungszeit mit

CLP – auch für kommerzielle Anwendungen – relativ gering und die Problemstellungen lassen sich mit CLP leicht modifizieren.

Azarmi und Abdul-Hameed stellen zwei CLP-Varianten vor. Zum einen Tour Generation (CLP TG + R), welches mit CHIP [99], [389] umgesetzt wurde. CHIP ist eine CLP-Sprache, die auf PROLOG basiert. Zum anderen wurde Compact Generation (CLP CG + R) mit ElipSys [309] umgesetzt, dass eine Erweiterung von CHIP darstellt und eine parallele Ausführung ermöglicht. CHIP und ElipSys wurden vom European Computer-Industry Research Centre (ECRC) entwickelt. Nachfolgend werden CLP TG + R und CLP CG + R vorgestellt. Das R steht für eine zusätzliche Reparatur der Lösung. Eine ausführliche Beschreibung von CLP TG + R gibt Abdul-Hameed [1]. Und für eine umfangreichere Version der Arbeit von Azarmi und Abdul-Hameed [18] sei auf die Masterarbeit von Abdul-Hameed [2] verwiesen.

Die erste Variante, Tour Generation (CLP TG + R) wurde von Azarmi und Abdul-Hameed in CHIP implementiert. Hier steht der Auftrag im Mittelpunkt. Die Modellierung der Problemstellung erfolgt demnach für jeden Auftrag durch eine Zuweisung eines Technikers, wobei nicht zu erledigende Aufträge mit einem Dummy-Mitarbeiter besetzt werden. Neben der Zuweisung eines Technikers auf einen Auftrag wird auch die Startzeit dessen Bearbeitung durch den jeweiligen Techniker definiert. CLP TG + R geht in zwei Schritten vor. In der ersten Phase werden Techniker den Aufträgen unter Berücksichtigung der Zielfunktion und der Nebenbedingungen zugewiesen. Der Ablauf von Phase eins ist in Algorithmus 3 abgebildet. Da es nach Abschluss dieser Phase Aufträge gibt, die nicht bearbeitet werden, wird in Phase zwei ein Verschieben bestehender Aufträge vorgenommen, um Raum für die restlichen Aufträge zu schaffen. Phase zwei repariert den Plan und dient der Reduzierung bisher nicht zugewiesener Aufträge.

Bei diesem Ablauf können an zwei Stellen Heuristiken eingesetzt werden. Zum einen bei der Auswahl des Auftrags j in Zeile 3 und bei der Auswahl des Technikers e in Zeile 6.

Bei der Auswahl des Auftrags (Zeile 3) wird ein First-fail-Ansatz nach van Hentenryck [388] verwendet. Es wird zuerst der Auftrag gewählt, für den die geringste Anzahl an Technikern qualifiziert ist. Schwer zuzuweisende Aufträge sollen somit schon frühzeitig abgearbeitet werden. Aufträge, für die eine Vielzahl an Technikern qualifiziert sind, haben eine größere Chance, auch später noch erfolgreich zugewiesen zu werden. In Experimenten konnte mit First-fail die Anzahl an nicht zugewiesenen Aufträgen nicht signifikant reduziert werden. Jedoch wurden die Fehlerpunkte für nicht zugewiesene Aufträge deutlich reduziert. In Bezug auf die Auswahl des Technikers (Zeile 6) wird ein B&B-Algorithmus ohne Backtracking nach French [127] verwendet (Closest Engineer). Dieser Ansatz zeigte in Experimenten die gewünschte Wirkung. Am besten schnitt in den Experimenten die gemeinsame Verwendung von First-fail und Closest-Engineer ab.

Kann ein Auftrag nicht erfüllt werden, wird ihm in Phase eins ein Dummy-Techniker zugewiesen. Um die Anzahl an unerledigten Aufträgen zu reduzieren, werden in Phase

Algorithmus 3 CLP TG + R

```
1: Erstelle die Liste RemainingUnallocated aller Aufträge
2: while RemainingUnallocated ist nicht leer do
3:   Wähle einen Auftrag  $j$  aus RemainingUnallocated
4:   Erstelle die Liste Qualified qualifizierter Techniker für Auftrag  $j$ 
5:   if Qualified ist nicht leer then
6:     Wähle einen Techniker  $e$  aus der Liste Qualified
7:     Entferne  $e$  aus Qualified
8:   else
9:     Weise Auftrag  $j$  den Dummy-Techniker zu
10:    Gehe zu Zeile 23
11:  end if
12:  if Techniker  $e$  erfüllt alle Nebenbedingungen then
13:    if Techniker  $e$  erfüllt seine lokalen Nebenbedingungen then
14:      Weise Techniker  $e$  auf den Auftrag  $j$  zu
15:      Weise dem Auftrag die früheste mögliche Startzeit zu
16:      Propagiere die temporalen Nebenbedingungen aller nicht
        zugewiesenen Aufträge
17:    else
18:      Gehe zu Zeile 5
19:    end if
20:  else
21:    Gehe zu Zeile 5
22:  end if
23:  Entferne Auftrag  $j$  aus der Liste RemainingUnallocated
24: end while
```

zwei Aufträge vertauscht und verschoben, um Raum für noch nicht erledigte Aufträge zu schaffen. Dabei wird eine Liste der noch nicht erledigten Aufträge abgearbeitet. Nach der Abarbeitung kann die Liste sich verändert haben, sodass ein mehrfacher Durchlauf von Phase zwei zu weiteren Verbesserungen führen kann.

Die zweite Variante von Azarmi und Abdul-Hameed ist Compact Generation (CLP CG + R). Sie stellt eine Erweiterung von CLP TG + R dar, die in ElipSys implementiert wurde. Auch bei CLP CG + R wurden die beiden Heuristiken First-fail und Closest-Engineer implementiert. Der Hauptunterschied zu CLP TG + R besteht darin, dass die Tour eines Technikers immer in eine optimale Reihenfolge gebracht wird, sobald ein neuer Auftrag hinzugefügt wird. Ein Auftrag braucht also nur an das Ende der bestehenden Tour angefügt werden und wird dann in die optimale Reihenfolge gebracht. CLP CG + R besteht ebenfalls aus zwei Phase, wobei in Phase eins ein Plan erzeugt wird, der in Phase zwei repariert wird. Der Ablauf von Phase eins bei CLP CG + R ist in Algorithmus 4 dargestellt.

Algorithmus 4 CLP CG + R

```

1: Erstelle die Liste RemainingUnallocated aller Aufträge
2: while RemainingUnallocated ist nicht leer do
3:   Wähle einen Auftrag  $j$  aus RemainingUnallocated
4:   Erstelle die Liste Qualified qualifizierter Techniker für Auftrag  $j$ 
5:   if Qualified ist nicht leer then
6:     Wähle einen Techniker  $e$  aus der Liste Qualified
7:     Entferne  $e$  aus Qualified
8:   else
9:     Weise Auftrag  $j$  den Dummy-Techniker zu
10:    Gehe zu Zeile 19
11:  end if
12:  Erweitere die Tour von Techniker  $e$  um Auftrag  $j$ 
13:  Bringe die Tour von Techniker  $e$  in die optimale Reihenfolge
14:  if Alle Nebenbedingungen werden erfüllt then
15:    Gehe zu Zeile 19
16:  else
17:    Gehe zu Zeile 5
18:  end if
19:  Entferne Auftrag  $j$  aus der Liste RemainingUnallocated
20: end while

```

Für die Berechnung der optimalen Tour wird ein B&B-Algorithmus verwendet. Gewöhnlich sind einem Mitarbeiter nur wenige Aufträge zugeordnet. Bei sehr vielen Aufträgen kann der B&B-Algorithmus sehr viel Rechenzeit benötigen und zum Flaschenhals von CLP CG + R werden. Daher wird ab einer bestimmten Anzahl an Aufträgen bei einem Techniker eine Einfüge-Heuristik für einen neuen Auftrag verwendet. Für alle gültigen Positionen des Auftrags in der Tour werden die Fehlerpunkte berechnet. Der Auftrag wird dort eingefügt, wo die geringsten Fehlerpunkte anfallen.

Ansatz von Yang 1996 untersucht Yang [428] CLP an den Problemen BT_Rel-250-118, BT_Mod-250-118 und BT_Bristol-250-118. Er implementiert CLP mit und ohne Forward Checking in Andorra-I [430], [429], [332].

Beim Forward Checking wird die freie Zeit eines Technikers berechnet. Übernimmt ein Techniker einen Auftrag, reduziert sich seine freie Zeit. Er kann aus der Liste der möglichen Mitarbeiter für einen Auftrag entfernt werden, wenn seine freie Zeit nicht mehr zur Erfüllung des Auftrags genügt.

Bei der Reihenfolge der Techniker erprobt Yang drei verschiedene Heuristiken:

1. Closest First (CL): Der Techniker, dessen Servicecenter am nächsten am Auftrag

liegt, wird zuerst gewählt.

2. Same Direction First (SD): Techniker fahren nicht nur vom Servicecenter zum Auftrag und zurück, sondern bewegen sich auch zwischen Aufträgen. Es wird daher zwischen Aufträgen unterschieden, die in Fahrtrichtung des Technikers liegen und Aufträgen, die nicht in Fahrtrichtung liegen. Entsprechend der Fahrtrichtung werden die Techniker ausgewählt. Bei SD gehört ein Techniker entweder der Gruppe an, die in der richtigen Richtung liegen oder nicht. Er ist in der Gruppe mit der richtigen Richtung, wenn eine der drei folgenden Voraussetzungen zutrifft. Dabei liegen zwei Aufträge dann in derselben Richtung, wenn der Winkel zwischen dem Servicecenter und den beiden Aufträgen kleiner als 45° ist.
 - a) Der Techniker ist bisher untätig.
 - b) Der Techniker wurde bereits einem Auftrag zugeordnet, der in der gleichen Richtung liegt.
 - c) Der Techniker hat bereits mehrere Aufträge, von denen jedoch zwei in derselben Richtung wie der neue Auftrag liegen.
3. Least Busy First (LB): Damit möglichst wenig Techniker untätig sind, wird der Techniker gewählt, der am wenigsten zu tun hat. Die Reduzierung der Fahrtkosten steht nicht im Fokus.

Yang liefert für die Laufzeit seines CLP-Ansatzes mit Forward Checking eine Laufzeit von $O(J * Q)$. Dabei ist J die Anzahl an Aufträgen und Q die durchschnittliche Anzahl an Aufträgen, für die ein Techniker qualifiziert ist.

4.8 Hill Climbing

Mit Hill Climbing (HC) wird eine erste Metaheuristik aufgegriffen, mit der die zentrale Forschungsfrage beantwortet werden soll, ob sich eher klassische OR-Verfahren, Metaheuristiken oder Multiagentensysteme für Problemstellungen des hochflexiblen WFM eignen (beispielhaft wurden drei typische Problemstellungen ausgewählt).

Eigene Implementierungen von HC werden in dieser Arbeit am Logistik- und am Handelsproblem erprobt. Beim British Telecom-Problem kann auf HC von Tsang und Voumouris [383] zurück gegriffen. Nach Ausführungen zum grundlegenden Ablauf von HC und dessen Anwendung bei Einsatzplanungsproblemen in der Literatur, wird die Anpassung von HC an die drei Problemstellungen in Kapitel 4.8.3 erläutert.

4.8.1 Grundlegender Ablauf

HC ist eine punktbasierte Heuristik aus der Gruppe der Lokalen Suche. HC legt nahe, dass es immer ein Maximum im Lösungsraum zu finden gilt. Selbstverständlich kann das Verfahren auch auf Minimierungsprobleme angewendet werden. Es funktioniert nach dem Prinzip eines Wanderers, der von Nebel umgeben ist und den Gipfel eines Berges sucht. Dabei kann er nicht in die Ferne schauen, sondern nur seine Umgebung für den nächsten Schritt betrachten. Folgt der Wanderer Schritt für Schritt dem Weg mit dem größten Anstieg, so kann er u.U. den Gipfel erreichen. Womöglich kommt er aber auch auf einem Berg unterhalb des Gipfels an, da er zum Gipfel ein Tal hätte durchwandern müssen. HC hat also das Problem, dass es nicht aus einem lokalen Optimum entkommen kann. Nach di Gasparo [96, S. 14] sind die wesentlichen Elemente der Lösungsraum, die Nachbarschaft und die Fitnessfunktion.

Beim HC existiert eine Vielzahl an Variationen. Drei weit verbreitete sind nachfolgend dargestellt [96, S. 16–17].

1. Steepest Hill Climbing (SHC): Es wird immer die komplette Nachbarschaft untersucht. Der Schritt wird dann zu dem Nachbarn gemacht, der zur größten Verbesserung führt. SHC bricht ab, wenn keine Verbesserung möglich ist. Alternativ kann SHC auch abbrechen, wenn nur Verschlechterungen möglich sind. Dann sind auch Seitwärtsbewegungen im Lösungsraum möglich.
2. Random Hill Climbing (RHC): Hier wird ein Nachbar zufällig aus der Nachbarschaft ausgewählt. Er wird für einen Schritt akzeptiert, wenn die Lösung verbessert wird bzw. nicht schlechter wird. Kommt der zufällig gewählte Nachbar nicht für den Schritt in Frage, wird ein anderer Nachbar zufällig gewählt. RHC bricht ab, wenn keiner der Nachbarn gewählt werden kann.
3. Min-Conflict Hill Climbing (MCHC): Dieses Verfahren wurde von Minton u.a. [272] entwickelt. Es läuft im Wesentlichen in zwei Schritten ab. Zunächst wird eine Variable zufällig gewählt, die gegen eine der Nebenbedingungen verstößt. Im zweiten Schritt wird dann der Wert dieser Variable innerhalb ihres Wertebereichs so geändert, dass die Anzahl der Verstöße gegen Nebenbedingungen minimal ist.

Aus den Beschreibungen von SHC, RHC und MCHC wird deutlich, dass die Verfahren immer dann abbrechen, wenn sie sich in einem lokalen Optimum bzw. bereits auf einem Plateau (wenn keine Seitwärtsschritte möglich sind) befinden. Dies ist der große Nachteil aller HC-Verfahren. Sie sind nicht in der Lage, aus einem lokalen Optimum bzw. Plateau zu entkommen. Um dem zu begegnen, wurden diverse Metaheuristiken wie GLS (Kapitel 4.9), SA (Kapitel 4.11) oder TS (Kapitel 4.10) entwickelt.

4.8.2 Anwendungen in der Literatur

Für das Airline Crew Scheduling Problem der Federal Express Corporation nutzen Baker u.a. 1979 [21] Heuristiken mit lokalen Suchoperatoren, die Lösungsverfahren für das Vehicle Scheduling entlehnt sind.

Jones stellt 1989 [198] ein Planungssystem für das Crew Rostering der American Airlines vor. Er geht zur Lösung in zwei Phasen vor. Zunächst generiert er einen Plan, der noch Fehler enthält. Dieser Plan wird mit Hilfe eines HC in Phase zwei verbessert.

Wren geht 1991 [425] ebenfalls in zwei Schritten vor. Zunächst wird eine Lösung zufällig konstruiert, die dann mit Hilfe eines HC verbessert wird.

Gewöhnlich wird beim Tour Scheduling immer nur eine einzige Lösung von den Verfahren erzeugt. Für den Planer ist es jedoch von Vorteil, weitere gleich gute Lösungen zu sehen. Easton und Rossin nutzen 1991 [111] daher einen HC, um weitere Lösungen zu finden.

Collins und Sisley wenden sich 1994 [75] der Aufgabenzuweisung bei Servicetechnikern zu. Reparaturaufträge werden über ein Call Center entgegen genommen und sofort Technikern zugewiesen. Ein HC findet dabei Anwendung, der die Aufträge in den bestehenden Einsatzplan einfügt.

Ein HC wird 1997 von Tsang und Voudouris [383] an den British Telecom-Problemen BT_Mod-250-118 und BT_Tsang-250-118 erprobt. Der HC, der die Basis für ihre Experimente mit FLS und Fast GLS ist, wird nachfolgend in Kapitel 4.8.3.1 genau beschrieben.

Stützle erprobt 1997 [360] MCHC und verschiedene Erweiterungen dieser Heuristik. In einer Erweiterung werden auch zufällige Schritte gemacht, die Verschlechterungen in Kauf nehmen. Eine weitere Variante verfügt über Elemente von TS. Zur Beurteilung zieht er verschiedene Problemstellungen wie lösbare CSP und das MAX-CSP heran. Alle Erweiterungen von MCHC schnitten bei CSP und MAX-CSP signifikant besser ab als MCHC selbst. Besonders die Variante mit TS scheint nach Stützle besonders vielversprechend für sehr schwere Problemstellungen zu sein.

Schaerf und Meisels stellen 1999 [335] diverse HC für verschiedene Employee Timetabling Probleme vor. Eine ähnliche Arbeit veröffentlichen sie 2003 [258].

Williams und Dozier wenden 2001 [419] verschiedene Varianten von MCHC auf ein Graphenfärbeproblem an. Sie nutzen den HC von Minton u.a. [272], den HC von Wallace und Freuder [407], den Breakout Algorithmus (BA) von Morris [275], [276] sowie einen eigenen BA. Williams und Dozier führen Experimente durch und kommen zu dem Schluss, dass ein BA für derartige Problemstellungen am besten geeignet ist.

Bei HC ist die Wahl der Nachbarschaften entscheidend. Mit VNS soll eine bessere Erkundung des Lösungsraums gefördert werden. VNS wurde 2004 erfolgreich bei der Dienstplanung von Burke u.a. [57] angewendet. Sie erweitern HC und TS um verschiedene Varianten von VNS. Fast alle Verfahren schneiden gleich gut ab. Burke u.a. unterstreichen jedoch die Bedeutung besonders einfacher HC, die in einer Vielzahl an Nachbarschaften suchen.

4.8.3 Anpassung an hochflexibles Workforce Management

Die nachfolgend dargestellten HC für das Logistik- und Handelsproblem wurden selbst entwickelt. Für das Problem der British Telecom wird auf die Ergebnisse von Tsang und Voudouris [383] zurück gegriffen.

4.8.3.1 Untertägige Personaleinsatzplanung in der Logistik

Der Ablauf des HC ist bei allen Logistikproblemen mit und ohne Stammarbeitsplatzregelung identisch. Beim verwendeten HC werden die Arbeitsplatzzuweisungen in der bisher besten Lösung nacheinander systematisch verändert. Nach jeder Änderung wird der Fitnesswert berechnet. Sobald die Fehlerpunktzahl nicht größer wird, wird die Modifikation der Arbeitsplatzzuweisung übernommen – ansonsten wieder verworfen. Die Veränderung der Arbeitsplatzzuweisung erfolgt für alle Elemente der zweidimensionalen Matrix (siehe Kapitel 4.1.2), in denen der jeweilige Mitarbeiter anwesend ist. Ein Matrixelement ist also der Kreuzungspunkt aus Zeitintervall und Mitarbeiter. Nachfolgend ist der Kern des HC in Algorithmus 5 dargestellt.

Der Dummy-Arbeitsplatz zeigt dabei an, dass der entsprechende Mitarbeiter nicht anwesend ist. Dort dürfen keine Änderungen vorgenommen werden. Im dargestellten Pseudocode wird der Index i für die Elemente der zweidimensionalen Matrix und der Index j für den Arbeitsplatz je Schleifendurchlauf erhöht. Bei den Matrixelementen ist zu beachten, dass sich der HC immer Spalte für Spalte (also beim gleichen Mitarbeiter) durch die Matrix bewegt und nach dem letzten Zeitintervall der Matrix zur nächsten Zeile springt. Es ist auch möglich, den Index der Matrixelemente oder des Arbeitsplatzes vom Größten beginnend rückwärts zu zählen. Aus den Kombinationsmöglichkeiten der Laufrichtungen beider Indizes ergeben sich insgesamt vier Durchläufe für den verwendeten HC. Grundsätzlich gilt, dass eine Nachbarschaft immer nur einen Nachbarn enthält. Steht in einem Matrixelement der Arbeitsplatz drei, so ist der einzige Nachbar (abhängig von der Zählrichtung bei Arbeitsplätzen) ein Einsatzplan mit dem Arbeitsplatz vier im gleichen Matrixelement (bzw. zwei, wenn bei den Arbeitsplätzen abwärts statt aufwärts gezählt wird). Die Matrixelemente werden also recht systematisch durchlaufen, was wegen der angestrebten Reduzierung der Arbeitsplatzwechsel auch notwendig ist. Aufgrund der

Algorithmus 5 Hill Climber bei der untertägigen Personaleinsatzplanung in der Logistik

```
1: Erzeuge eine zufällige Startlösung
2: Berechne die Fitness
3: for i = 0 bis Anzahl Matrixelemente-1 do
4:   if Arbeitsplatz in Matrixelement i  $\neq$  Dummy-Arbeitsplatz then
5:     for w = 1 bis Anzahl Arbeitsplätze do
6:       AktuellerAP = Arbeitsplatz in Matrixelement i
7:       Arbeitsplatz in Matrixelement i = w
8:       Berechne die Fitness
9:       if f(neuer Position)  $\leq$  f(aktuelle Position) then
10:        // neuen Bestwert gefunden
11:        f(aktuelle Position) = f(neue Position)
12:       else
13:         Arbeitsplatz in Matrixelement i = AktuellerAP
14:       end if
15:     end for
16:   end if
17: end for
```

vielen Plateaus im Lösungsraum sind Seitwärtsbewegungen ebenfalls erforderlich. Daher wird ein Schritt auch dann vollzogen, wenn sich der Fitnesswert nicht verändert hat.

Da die vier Durchläufe deutlich weniger Berechnungen der Fitnessfunktion erfordern als beim Abbruchkriterium vorgesehen ist, wird HC bei Beendigung mit einer neuen zufälligen Startlösung gestartet. Dieses Vorgehen wird so lange wiederholt, bis das Abbruchkriterium erreicht ist. Es handelt sich also um einen HC mit Multistart. Für die Bestimmung der Startlösung stehen zwei Varianten zur Verfügung, die jeweils separat getestet werden.

1. Zufällige Zeitintervalle (I1): Alle Dimensionen, in denen ein Mitarbeiter anwesend ist, werden mit zufällig ermittelten Arbeitsplätzen belegt, wobei auf Qualifikationen Rücksicht genommen wird.
2. Zufällige Tage (I2): Jedem Mitarbeiter wird für den kompletten Tag jeweils ein zufällig ermittelter Arbeitsplatz für die Zeit seiner Anwesenheit zugewiesen. Dabei werden die Qualifikationen der Mitarbeiter berücksichtigt.

4.8.3.2 Automatische Arbeitszeitmodellerstellung im Handel

Das Grundprinzip des HC für das Handelsproblem unterscheidet sich nicht wesentlich vom HC für das Logistikproblem. Der Unterschied liegt einzig und allein in der Be-

rücksichtigung eines zusätzlichen Arbeitsplatzes – zusätzlich zum Dummy-Arbeitsplatz – der die Verfügbarkeit von Mitarbeitern kenntlich macht. Der zusätzliche Arbeitsplatz gibt an, dass der Mitarbeiter zwar verfügbar ist, aber nicht arbeitet. Bei Handel_2AP und Handel_2AP_Jan mit zwei Arbeitsplätzen werden vom HC somit drei Arbeitsplätze durchlaufen. Beim Handel_1AP und Handel_1AP_Jan sind es zwei Arbeitsplätze. Der Dummy-Arbeitsplatz wird nie durchlaufen, da der Mitarbeiter dort z.B. aufgrund von Krankheit oder Urlaub abwesend ist oder der Shop geschlossen hat.

Für die Initialisierung kann nur auf die zweite Variante (I2) des HC für das Logistikproblem zurück gegriffen werden. Die erste Variante (I1) muss entfallen. Die Berücksichtigung des zusätzlichen Arbeitsplatzes, der kenntlich macht, dass ein Mitarbeiter verfügbar ist aber nicht arbeitet, führt in Variante eins zu einer schier unüberschaubaren Zahl an fehlerhaften Arbeitszeitmodellen. Bei einer derart schlechten Startlösung hat der HC erhebliche Probleme und erzeugt keine brauchbaren Ergebnisse.

4.8.3.3 Kombination von Touren- und Personaleinsatzplanung

Tsang und Voudouris entwickeln 1997 [383] einen HC für das British Telecom-Problem und testen ihn an den Varianten BT_Mod-250-118 und BT_Tsang-250-118. Ausgehend von diesem HC entwickeln sie FLS und Fast GLS. Nachfolgend wird der HC erläutert, der sehr einfach gestaltet ist.

Zur Abbildung der Problemstellung nutzen Tsang und Voudouris eine Permutation der zu erledigenden Aufträge (siehe Kapitel 4.1.1). Zur Erzeugung der Startlösung werden die Aufträge nach der Anzahl an Technikern geordnet, die für den jeweiligen Auftrag qualifiziert sind. Aufträge, für die nur sehr wenige Techniker in Frage kommen, werden als erstes in der Permutation platziert. Der HC geht anschließend so vor, dass er zwei Aufträge in der Permutation vertauscht, wenn die Fitness durch das Vertauschen besser wird. Weitere Ausführungen waren zu diesem Verfahren nicht zu finden.

4.9 Fast Local Search und Guided Local Search

In diesem Kapitel wird zunächst auf den grundlegenden Aufbau der beiden Metaheuristiken Fast Local Search (FLS) und Guided Local Search (GLS) eingegangen. Kapitel 4.9.2 enthält Ausführungen zur Anwendung von FLS und GLS bzw. der Kombination von beidem: Fast GLS. Anschließend wird die Anpassung von FLS und Fast GLS durch Tsang und Voudouris [383] am British Telecom-Problem erläutert. Beim Logistik- und Handelsproblem wurden lediglich einige Tests angestellt, da FLS und Fast GLS für diese

Probleme ungeeignet sind. Weitere Ausführungen hierzu sind Kapitel 4.9.3 zu entnehmen.

4.9.1 Grundlegender Ablauf

4.9.1.1 Fast Local Search

FLS ist eine allgemein anwendbare Strategie, um die Nachbarschaft in Optimierungsverfahren einzuschränken und die Suche somit zu beschleunigen. Grundsätzlich muss für den Geschwindigkeitsvorteil jedoch die Gefahr in Kauf genommen werden, dass gute Ergebnisse unberücksichtigt bleiben. Zudem dient das Verfahren nicht dem Entkommen aus einem lokalen Optimum.

FLS arbeitet wie folgt: Für jeden Auftrag existiert für dessen Positionen in der Permutation je ein Aktivierungsbit, welches die Werte null oder eins annehmen kann. Dieses Aktivierungsbit wird wie folgt manipuliert:

1. Bei der Initialisierung werden alle Aktivierungsbits auf eins gesetzt (an).
2. Das Aktivierungsbit einer Auftragsposition wird auf null gesetzt (aus), wenn sich durch Vertauschen dieses Auftrags in der aktuellen Permutation mit allen anderen Aufträgen kein besseres Ergebnis finden lässt.
3. Ein Aktivierungsbit wird auf eins gesetzt, wenn der Auftrag in der entsprechenden Position in einen Tausch involviert ist, der zu einem besseren Ergebnis führt und somit übernommen wird.

Nur wenn das Aktivierungsbit in der entsprechenden Position der Permutation auf eins steht, zählt das Suchverfahren diese Position zu seiner Nachbarschaft. Das bedeutet, dass Positionen, die zuvor nicht zu einer Verbesserung beitragen konnten, ignoriert werden, wodurch die Nachbarschaft erheblich reduziert wird.

4.9.1.2 Guided Local Search

GLS wird auf ein Lokales Suchverfahren gesetzt. Es wurde von GENET (Generic Neural Network) abgeleitet, das 1991 von Tsang und Wang entwickelt wurde [408], [384]. Das grundsätzliche Konzept von GLS ist dem Operations Research entlehnt. Hierzu sei auf die Arbeiten von Koopman [219] und Stone verwiesen [359]. Der Ablauf von GLS ist in Algorithmus 6 dargestellt.

Algorithmus 6 Guided Local Search

```

1: Erzeuge eine Startlösung
2: Initialisiere alle Strafpunkte  $p_f$  mit 0
3: repeat
4:   Führe die Lokale Suche für  $h(s)$  aus
5:   if Lokales Optimum erreicht then
6:     Berechne  $util_f$  für jedes Feature
7:      $p_{f++}$  für jedes Feature mit maximalem  $util_f$ 
8:   end if
9: until Abbruchkriterium tritt ein
10: Gib das beste Ergebnis aus, dass für  $g(s)$  gefunden wurde

```

Bei GLS wird eine Fitnessfunktion derart modifiziert, sodass die Lokale Suche aus einem lokalen Optimum entkommen kann. Statt der ursprünglichen Fitnessfunktion $g(s)$ wird bei GLS nun $h(s)$ verwendet, die $g(s)$ um zusätzliche Strafpunkte erweitert. Die modifizierte Fitnessfunktion hat folgende Gestalt.

$$h(s) = g(s) + \lambda \sum_{f=1}^F p_f I_f(s) \quad (4.5)$$

Dabei steht λ für einen Parameter, der den Einfluss der Strafpunkte steuert. Gewöhnlich ist λ relativ unempfindlich gegenüber Änderungen, was die Experimente von Tsang und Voudouris [383] belegen. F ist die Anzahl der Features, die bei GLS berücksichtigt werden. Ein Feature ist z.B. ein Auftrag, ein Arbeitsplatz, ein Fahrzeug, etc. – also Dinge, die in der Fitnessfunktion zu Fehlern führen können. $I_f(s)$ ist eine binäre Variable und zeigt an, ob in der aktuellen Lösung gegen das Feature f verstoßen wird.

$$I_f(s) = \begin{cases} 1, & \text{wenn } s \text{ gegen das Feature } f \text{ verstößt} \\ 0, & \text{sonst} \end{cases} \quad (4.6)$$

Die Strafpunkte je Feature f werden mit p_f abgebildet. Sie werden mit null initialisiert. Befindet sich die Lokale Suche in einem lokalen Optimum, werden bestimmte Strafpunkte p_f derart erhöht, sodass sich die Zielfunktion $h(s)$ und somit das Zielfunktionsgebirge verändert und ein Entkommen aus dem lokalen Optimum ermöglicht wird. Die Erhöhung von p_f geschieht wie folgt: Für jedes Feature wird die $util_f(s_*)$ berechnet, wobei s_* für die Lösung eines lokalen Optimums steht.

$$util_f(s_*) = I_f(s_*) \frac{c_f}{1 + p_f} \quad (4.7)$$

Features f , die höhere Kosten c_f verursachen und gegen die bisher nicht sehr häufig verstoßen wurde, erhalten besonders hohe util-Werte. Bei dem/den Feature/s f mit

dem größten util-Wert wird p_f um 1 erhöht.

$$p_f = p_f + 1 \tag{4.8}$$

Durch die Berücksichtigung von Kosten c_f und Strafpunkten p_f bei einem Verstoß gegen ein Feature f in einem lokalen Optimum wird die Suche in andere Bereiche des Lösungsraums gelenkt. Bei Features mit hohen Kosten c_f werden die Strafpunkte p_f häufiger erhöht. In diesem Fall steigt die Chance bei anderen Features, dass auch bei ihnen p_f erhöht wird [398, S. 40–41]. Ein ähnlicher Ansatz wie bei GLS wird 1992 von Thrun [373] verwendet.

4.9.2 Anwendungen in der Literatur

Guided Local Search (GLS) wurde vornehmlich von Tsang, Voudouris und Mills in zahlreichen Arbeiten an diversen kombinatorischen, diskreten sowie kontinuierlichen Problemstellungen erprobt. Einen allgemeinen Überblick geben Voudouris und Tsang 2001 [405] und 2003 [406]. Voudouris veröffentlicht 1997 seine Dissertation [398] zu GLS. Dort geht er auf kontinuierliche Testfunktionen, die Frequenzzuweisung (RLFAP), das Problem BT_Mod-250-118, das TSP und auf das quadratische Zuweisungsproblem (QAP) ein. Ergänzende Anwendungen von GLS auf die Problemstellungen seiner Dissertation sind in folgenden Quellen zu finden.

- Kontinuierliche Testfunktionen: Voudouris und Tsang 1995 [401] sowie Voudouris 1998 [399].
- Frequenzzuweisung: Voudouris und Tsang 1996 [402] sowie 1998 [403].
- BT_Mod-250-118: Tsang und Voudouris 1997 [383].
- Traveling Salesman Problem: Voudouris und Tsang 1999 [404].
- Quadratisches Zuweisungsproblem: Mills, Tsang und Ford 2003 [271].

Mills und Tsang wenden 1999 [269] GLS auf das SAT-Problem an und 2000 [270] auf das MAX-SAT-Problem. In seiner Dissertation erweitert Mills 2002 [268] GLS um ein Aspirationskriterium und um Random Moves. Diese erprobt er an den Problemen SAT, MAX-SAT und QAP, wo er die Ergebnisse des ursprünglichen GLS verbessern kann. Eine Kurzfassung wurde 2002 von Tsang, Mills und Ford [381] erstellt.

Am British Telecom-Problem erproben Tsang und Voudouris 1997 [383] u.a. FLS und Fast GLS. Beide Verfahren werden nachfolgend genauer beschrieben.

Neben der Lokalen Suche wurde das Prinzip von GLS zum Entkommen aus lokalen Optima auch bei GA implementiert. Der Guided GA wurde von Lau [233] entwickelt und erfolgreich bei diversen Problemstellungen angewendet. 1998 entwickelt Lau einen Guided GA für das allgemeine Zuweisungsproblem [234] und für die Prozessorkonfiguration [235]. Zudem wendet er den Guided GA später für das Problem der Frequenzzuweisung an [236], [237].

4.9.3 Anpassung an hochflexibles Workforce Management

Umfassende Experimente mit FLS und GLS wurden in dieser Arbeit nicht am Logistik- und Handelsproblem vorgenommen. GLS lässt bei beiden Problemstellungen im Vergleich zu HC keine Vorteile erkennen. Es wurden auch einige Tests am Logistikproblem durchgeführt, deren Erkenntnisse sich auf das Handelsproblem übertragen lassen. Eine Begründung, warum GLS beim Logistik- und Handelsproblem nicht sinnvoll anwendbar ist, kann Kapitel 6.1.4 und 6.2.5 entnommen werden. Daher ist nachfolgend nur das Vorgehen für FLS und Fast GLS am British Telecom-Problem dargestellt.

4.9.3.1 Kombination von Touren- und Personaleinsatzplanung

In ihrer Arbeit [383] wenden Tsang und Voudouris FLS und Fast GLS auf die Probleme BT_Mod-250-118 und BT_Tsang-250-118 an und konnten die zum damaligen Zeitpunkt besten Ergebnisse erzielen.

Abbildung der Problemstellung Wie bereits in Kapitel 4.1.1 dargestellt, wird das Problem von Tsang und Voudouris [383] als Permutation aller 250 Aufträge dargestellt. Um aus dieser Permutation nun einen Einsatzplan zu erstellen und somit die Fehlerpunkthöhe zu erhalten, wird wie folgt vorgegangen.

1. Ordne für jeden der 250 Aufträge die qualifizierten Techniker in aufsteigender Reihenfolge, wobei sich die Reihenfolge aus den Fahrtkosten zwischen Auftrag und Servicecenter des Technikers ergibt (kleinste zuerst).
2. Durchlaufe die Permutation der 250 Aufträge und tue für jeden Auftrag folgendes: Versuche den Auftrag demjenigen Techniker zuzuweisen, der zuerst in der Liste der qualifizierten Techniker steht. Ist dies möglich, berechne bei diesem Techniker die optimale Reihenfolge seiner Aufträge (im Durchschnitt hat ein Techniker 2,2 Aufträge). Ist dies nicht möglich, versuche den Auftrag dem nächsten Techniker in der Liste der qualifizierten Techniker zuzuweisen (es ist möglich, dass ein Auftrag keinem Techniker zugeordnet werden kann).

3. Sobald alle Aufträge abgearbeitet sind, wird die Fitness des so erzeugten Einsatzplans berechnet.

Diese Art der Problemdarstellung ermöglicht eine sehr einfache Veränderung der Lösung durch das Vertauschen zweier Aufträge in der Permutation der 250 Aufträge. Dabei müssen keine Regelungen hinsichtlich Qualifikationen, Fahrtzeiten, Arbeitsende, Auftragsstypen, etc. berücksichtigt werden, da dies erst bei der Erstellung des eigentlichen Einsatzplans und der Berechnung der Fehlerpunkte erfolgt. Problematisch ist an dieser Darstellung jedoch, dass womöglich gute Einsatzpläne nicht erzeugt werden können. Der Lösungsraum enthält Bereiche, die mit der verwendeten Darstellung nicht erreicht werden können. Dies liegt daran, dass ein Auftrag immer dem ersten qualifizierten Techniker zugeteilt wird, der den Auftrag übernehmen kann. U.U. würde jedoch die Zuteilung zu einem anderen Techniker mehr Sinn machen, da dadurch die Fahrtzeit zu einem Folgeauftrag reduziert wird. Dieser Aspekt wird nicht berücksichtigt, da immer nur die Fahrtkosten zwischen Auftrag und Servicecenter berücksichtigt werden.

Initialisierung Für die Ermittlung der Startposition testen Tsang und Voudouris [383] verschiedenen Varianten.

1. Zufällige Reihenfolge der Aufträge.
2. Auftragsreihenfolge in Abhängigkeit von der Auftragsdauer geteilt durch die Anzahl qualifizierter Techniker (größte Wert zuerst, da diese Aufträge am schwierigsten zuzuweisen sind).
3. Auftragsreihenfolge in Abhängigkeit von der Anzahl qualifizierter Techniker (kleinste Werte zuerst, da diese Aufträge am schwierigsten zuzuweisen sind).

Gibt es in Variante zwei und drei mehrere Alternativen, wird zufällig entschieden, welcher Auftrag als nächstes in die Permutation genommen wird. In Experimenten erzeugt Variante drei die besten Startlösungen. Die Ergebnisse von FLS und Fast GLS sind auch mit dieser Variante am besten. Daher wird von Tsang und Voudouris Variante drei für die Experimente gewählt.

Fast Local Search Die Nachbarschaft jedes Zustands umfasst beim von Tsang und Voudouris [383] verwendeten HC $O(J^2)$ Nachbarn, wobei J für die Anzahl an Aufträgen steht. Bei 250 Aufträgen ergibt sich somit in jeder Iteration eine Vielzahl an Berechnungen der Fitnessfunktion. Die Aktivierungsbits werden für die Positionen der Aufträge in der Permutation gesetzt. Steht das Aktivierungsbit eines Auftrags auf 1, wählt das Suchverfahren diesen Auftrag für einen Tausch mit einem anderen Auftrag aus.

Features für Guided Local Search Im British Telecom-Problem gibt es 250 Aufträge. Diese Aufträge beeinflussen das Ergebnis entscheidend, da nicht erledigte Aufträge zu Fehlerpunkten führen. Sie werden daher von Tsang und Voudouris [383] als Feature genommen. Fahrtkosten hingegen werden nicht als relevante Features angesehen, da sie im Zuge der Reduzierung nicht ausgeführter Aufträge minimiert werden. Denn je besser die Touren gestaltet sind, umso mehr Zeit steht für die Erledigung weiterer Aufträge zur Verfügung.

4.10 Tabu Search

Analog zu den bisherigen Kapiteln wird auch für die Metaheuristik Tabu Search (TS) zunächst auf den grundlegenden Ablauf und anschließend auf relevante Literatur eingegangen. TS wurde im Rahmen einer Diplomarbeit von Helbig [178] am Logistikproblem ohne Stamarbeitsplatzregelung angepasst (Kapitel 4.10.3). Weitere Implementierungen von TS werden in dieser Arbeit nicht vorgenommen.

4.10.1 Grundlegender Ablauf

TS ist eine iterative punkt-basierte Metaheuristik, die in die Gruppe der Lokalen Suche eingeordnet wird. Sie wurde 1986 unabhängig voneinander von Glover [142] und Hansen [174] entwickelt. Der grundlegende Ablauf ist in Algorithmus 7 dargestellt.

Algorithmus 7 Tabu Search

- 1: Erzeuge Startlösung X
 - 2: **while**
 - 3: Abbruchkriterium nicht erfüllt **do**
 - 4: Bestimme den Nachbarn \hat{X} von X mit der besten Fitness, der nicht tabu ist oder das Aspirationskriterium erfüllt
 - 5: $X = \hat{X}$
 - 6: Aktualisiere die Tabuliste
 - 7: **end while**
-

Mit TS soll die Suche so gesteuert werden, dass ein Entkommen aus einem lokalen Optimum möglich ist. Im Gegensatz zu SA erfolgt die Steuerung jedoch nicht stochastisch, sondern deterministisch. Auch bei TS werden Verschlechterungen der Lösung bewusst in Kauf genommen. Im Prinzip wählt der Algorithmus für den nächsten Schritt immer die beste Lösung aus der Nachbarschaft aus – auch wenn diese zu einer Verschlechterung führt.

Würde man dieses Vorgehen konsequent umsetzen, so würde nach einem Schritt zu einer schlechteren Lösung sofort wieder zur vorherigen, besseren Lösung gesprungen werden. Zyklen oder Schleifen wären die Folge. Um dies zu verhindern, erhält das Verfahren ein Kurzzeitgedächtnis – genauer gesagt eine Tabuliste. Dort sind bereits getätigte Schritte bzw. Informationen über bisherigen Schritte für eine bestimmte Dauer gespeichert und somit „tabu“ gesetzt. Die Anzahl der Nachbarn wird hierdurch eingeschränkt, da Schritte aus der Tabuliste nicht erlaubt sind. In welchem Umfang Informationen gespeichert werden, kann sehr unterschiedlich sein. Es können ganze Lösungen, Eigenschaften von Lösungen oder Eigenschaften von Schritten gespeichert werden. Wichtig ist, dass genügend Informationen gespeichert werden, um Zyklen zu verhindern. Auf der anderen Seite bedeutet dies auch einen großen Speicher- und Rechenbedarf. Werden ganze Lösungen gespeichert, so sind Zyklen absolut ausgeschlossen. Der Speicher- und Rechenbedarf ist dann jedoch entsprechend groß. [34, S. 44–53]

In jeder Iteration wird die Tabuliste aktualisiert. Neben der Aufnahme von Einträgen können bereits vorhandene Einträge nach Ablauf ihrer Tabudauer T aus der Liste wieder entfernt werden. Jeder Eintrag in der Tabuliste hat eine eigene Tabudauer. Die Wahl dieser Dauer ist ein entscheidender Parameter und erheblich für die Ergebnisqualität mitverantwortlich. Eine zu kurze Dauer verhindert Zyklen nicht effektiv und eine zu lange Dauer schließt gute Lösungen möglicher Weise zu lange aus, was die Suche behindert [364, S. 447]. Daher wird häufig eine variable Tabulistenlänge verwendet, wobei es keine generelle Aussage über die optimale Wahl der Tabudauer T gibt [34, S. 50]. Zwei Ansätze einer variablen Tabulistenlänge sind Robust TS von Taillard [364] und Reactive TS von Battiti und Tecchioli [25].

Wird nicht die komplette Lösung in der Tabuliste gespeichert, kann die Tabuliste sehr mächtig sein. Sie kann Lösungen ausschließen, die noch gar nicht besucht wurden, da auch sie Eigenschaften aus der Tabuliste besitzen. Um zu vermeiden, dass bessere Lösungen nicht unerreichbar sind, werden Aspirationskriterien eingeführt. Sie dienen dazu, einen tabu gesetzten Schritt dennoch auszuführen [139, S. 44]. Glover und Laguna geben eine sehr umfassende Übersicht über Aspirationskriterien [143, S. 50–54]. Bei Aspiration by Objective wird ein tabu gesetzter Schritt dennoch ausgeführt, wenn sich die Lösung verbessern würde. Denn dort ist sichergestellt, dass diese Lösung zuvor besucht wurde. Ein anderes Beispiel ist Aspiration by Default. Es findet Anwendung, wenn alle Nachbarn tabu gesetzt sind. Die Lösung würde sich dann zwangsläufig festfahren. Um dies zu verhindern, wird die Lösung mit der geringsten Verschlechterung gewählt.

4.10.2 Anwendungen in der Literatur

Bereits beim allgemeinen Überblick über Lösungsverfahren (siehe Kapitel 4.4) und bei HC (siehe Kapitel 4.8.2) wurden einige Arbeiten vorgestellt, in denen u.a. TS verwendet wurde. Nachfolgend wird auf weitere Arbeiten eingegangen.

TS wird häufig beim Scheduling und Routing verwendet. So auch bei der Transportplanung 1995 von Cao und Uebe [64] und 1998 von Sun u.a. [363].

1999 nutzen Cavique Rego und Themido 1999 [65] einen TS-Ansatz für die Dienstplanung von sechs Notaufnahmen. Dabei sind besondere Regelungen hinsichtlich der Schichtzyklen zu berücksichtigen. Sie testen ihren Ansatz an Echtdateien und konnten die manuellen Pläne verbessern.

Besonders häufig wird TS in Verbindung mit VNS verwendet. Di Gaspero und Schaerf arbeiten 2000 [97] mit wechselnden Nachbarschaften am Problem der Prüfungsorganisation. Sie benutzen zwei lokale Suchtechniken mit unterschiedlichen Nachbarschaften. Sie nennen das Verfahren Tandem Search, da nach jeder Iteration automatisch zwischen beiden Nachbarschaften gewechselt wird.

Auf dieser Grundlage entwickelten sie 2003 [96] Multi-Neighbourhood Search (MNS) bzw. Token Ring Search für die universitäre Stundenplanerstellung. Auch hier wird kontinuierlich zwischen mehreren Nachbarschaften gewechselt. Bei der universitären Stundenplanerstellung müssen Kombinationen aus Kursen, Räumen und Zeiten gefunden werden. Daher entscheiden sich Di Gaspero und Schaerf für die Basis-Nachbarschaften Zeit und Raum. Daraus abgeleitet ergeben sich zwei zusätzliche Nachbarschaften: Union (Zeit oder Raum) und Composition (Zeit und Raum). Insgesamt erhalten sie somit vier Nachbarschaften, die kombiniert mit HC und TS insgesamt acht Suchverfahren bilden. Diese acht Suchverfahren werden wie in einem Kreis kontinuierlich nacheinander durchlaufen.

2005 wird VNS bei TS ebenfalls für die universitäre Stundenplanerstellung durch Abdullah, Burke und Mc Collumbei [3] angewendet. Sie waren bei den elf Testproblemen jedoch nicht in der Lage, die bisherigen Bestwerte mit ihrem Ansatz zu unterbieten.

4.10.3 Anpassung an hochflexibles Workforce Management

TS wird in dieser Arbeit lediglich an der Problemstellung des Logistikdienstleisters ohne Stammarbeitsplatzregelung untersucht. Dies ist 2009 durch Helbig im Rahmen einer Diplomarbeit an der TU Ilmenau erfolgt [178]. Weitere Implementierungen für andere Probleme werden nicht vorgenommen.

4.10.3.1 Untertägige Personaleinsatzplanung in der Logistik

Nachfolgend wird die Anpassung von TS auf das Problem der untertägigen Personaleinsatzplanung in der Logistik geschildert. Untersuchungen an der Variante mit Stammar-

beitsplatzregelung werden nicht angestellt.

Initialisierung Zunächst werden die beiden Varianten (I1 und I2) für die Erzeugung der Startlösungen übernommen, die beim HC in Kapitel 4.8.3.1 vorgestellt wurden. Dort wird in Variante 1 (I1) für jedes Zeitintervall, in dem ein Mitarbeiter anwesend ist, unter Berücksichtigung von Qualifikationen ein Arbeitsplatz zufällig bestimmt. Bei Variante 2 (I2) sind die Zuordnungen sehr viel geordneter, da einem Mitarbeiter unter Berücksichtigung seiner Qualifikationen ein und derselbe Arbeitsplatz für den kompletten Tag zugewiesen wird, insofern er anwesend ist.

Bei TS wird eine dritte Initialisierungsvariante (I3) eingeführt. Da der manuell erstellte Einsatzplan bekannt ist, kann auch er als Startlösung genommen werden. Wie in Variante 2 sind dort den anwesenden Mitarbeitern Arbeitsplätze für den kompletten Tag zugewiesen. Untertägige Arbeitsplatzwechsel sind daher nicht eingeplant. Beim HC mit Multistart hätte die Initialisierung mit Variante 3 keinen Sinn gemacht, da HC deterministisch abläuft und ein mehrfaches Starten aus dem gleichen Punkt im Lösungsraum heraus keinen Sinn macht.

Nachbarschaft Die Fehlerpunkte ergeben sich beim Logistikproblem aus Qualifikationsverletzungen, Abweichungen von Personalbedarf und Personaleinsatz sowie aufgrund von Arbeitsplatzwechseln. Diese Fehler treten bei der Prüfung entweder bzgl. eines Mitarbeiters (Qualifikationen, Arbeitsplatzwechsel) oder bzgl. eines Zeitintervalls (Über-/Unterdeckungen im Personaleinsatz) auf. In Kapitel 4.10.2 wurde das Token Ring-Verfahren von di Gaspero und Schaerf [96] vorgestellt. Dort wird abwechselnd mit den Basis-Nachbarschaften Zeit und Raum gearbeitet. Da di Gaspero und Schaerf sehr gute Resultate erzielen, soll dieser Ansatz auch auf die untertägige Personaleinsatzplanung übertragen werden. Es werden also zwei Nachbarschaften erzeugt. Mitarbeiter und Zeit, die in Abbildung 4.10 und 4.11 dargestellt sind [178, S. 40].

	Periode 1	Periode 2	Periode 3	Periode 4	Periode 5
Mitarbeiter 1	0	0	1	1	1
Mitarbeiter 2	0	0	0	0	1
Mitarbeiter 3	0	1	1	1	2
Mitarbeiter 4	0	0	0	1	1
Mitarbeiter 5	0	0	0	1	1
Mitarbeiter 6	0	0	1	1	2

Abbildung 4.10: Nachbarschaft bzgl. des Mitarbeiters [178, S. 40]

Im Laufe der Suche wird kontinuierlich zwischen den beiden Nachbarschaften gewechselt. Wird in einem Matrixelement ein Wert geändert, so wird nachfolgend in der jeweils

	Periode 1	Periode 2	Periode 3	Periode 4	Periode 5
Mitarbeiter 1	0	0	1	1	1
Mitarbeiter 2	0	0	0	0	1
Mitarbeiter 3	0	1	1	2	2
Mitarbeiter 4	0	0	0	1	1
Mitarbeiter 5	0	0	1	1	1
Mitarbeiter 6	0	0	1	1	2

Abbildung 4.11: Nachbarschaft bzgl. des Zeitintervalls [178, S. 40]

anderen Nachbarschaft gesucht, die ebenfalls dieses Matrixelement enthält. Z.B. wird in Abbildung 4.10 in der Nachbarschaft des Mitarbeiters drei gesucht. Das beste Matrixelement ist dort der Zeitintervall vier, wo der Arbeitsplatz von eins auf zwei geändert wird. Anschließend wird die Nachbarschaftsform gewechselt, sodass nun im Zeitintervall vier gesucht wird (siehe Abbildung 4.11).

Problematisch an diesem Vorgehen ist jedoch, dass an sieben Tagen in der Woche fast rund um die Uhr gearbeitet wird. Die Mitarbeiter arbeiten jedoch nur eine gewisse Zeit an bestimmten Tagen. Es gibt daher viele unzusammenhängende Bereiche in der Matrix (siehe Abbildung 5.1 – grün sind Anwesenheiten und rot Abwesenheiten). Ein kreuzartiges Vorgehen garantiert nicht, dass die Suche in alle Teile des Lösungsraums vordringt. Daher wird eine dritte Nachbarschaft eingeführt, welche alle Matrixelemente als Nachbarn beinhaltet, in denen ein Mitarbeiter anwesend ist. Diese Nachbarschaft wird in einer Iteration mit einer geringen Wahrscheinlichkeit von 0,5% genutzt. Eine größere Wahrscheinlichkeit wird nicht verwendet, da diese Nachbarschaft sehr rechenintensiv ist.

Zusätzlich zum zuvor dargestellten Vorgehen wird auch VNS implementiert. Dies erfolgt jedoch ausschließlich für das Wochenproblem. Denn die Idee beruht darauf, dass sich die einzelnen Tage im Wochenproblem nicht gegenseitig beeinflussen. Daher können Nachbarschaften jeweils auf einen der sieben Tage begrenzt werden. Es wird bei VNS nun nicht mehr im kompletten Wochenproblem gesucht. Das Problem wird virtuell in Tage zerlegt. Die Suche beginnt beim ersten Tag. TS springt zum nächsten Tag der Woche, wenn in der letzten Iteration keine Verbesserung erzielt wurde. Beim Sonntag angekommen, wird wieder zum Montag gesprungen. Diese TS-Variante wird in den Ergebnistabellen im Anhang um die Bezeichnung VNS ergänzt.

Schritt Mit einem Schritt wird die Änderung der Lösung bezeichnet. Für Personaleinsatzplanungsprobleme werden bei TS meist nur zwei Formen verwendet. Zum einen der Shift-Move (siehe Abbildung 4.12) und zum anderen der Swap-Move (siehe Abbildung 4.13).

Bei einem Swap-Move wird die Zuordnung zweier Mitarbeiter zu Arbeitsplätzen ver-

	Periode 1	Periode 2	Periode 3	Periode 4	Periode 5
Mitarbeiter 1	0	0	1	2	1
Mitarbeiter 2	0	0	0	0	1
Mitarbeiter 3	0	1	1	1	1
Mitarbeiter 4	0	0	0	1	1
Mitarbeiter 5	0	0	0	1	1
Mitarbeiter 6	0	0	1	1	2
Bedarf auf AP 1	0	1	4	4	5
Bedarf auf AP 2	0	0	0	1	1
Defizit	/	/	-1	/	/



	Periode 1	Periode 2	Periode 3	Periode 4	Periode 5
Mitarbeiter 1	0	0	1	2	1
Mitarbeiter 2	0	0	0	0	1
Mitarbeiter 3	0	1	1	1	1
Mitarbeiter 4	0	0	0	1	1
Mitarbeiter 5	0	0	1	1	1
Mitarbeiter 6	0	0	1	1	2
Bedarf auf AP 1	0	1	4	4	5
Bedarf auf AP 2	0	0	0	1	1
Defizit	/	/	/	/	/

Abbildung 4.12: Shift-Move [178, S. 29]


tauscht. Hierdurch lassen sich Arbeitsplatzwechsel und Qualifikationsverletzungen reduzieren. Über- und Unterdeckungen können mit einem Swap-Move nicht verringert werden, da die Anzahl an Zuweisungen zu einem Arbeitsplatz nicht verändert wird – lediglich die Zuordnung, wer auf welchem Arbeitsplatz arbeitet. Um Über- und Unterdeckungen im Personaleinsatz zu reduzieren, müssen Shift-Moves durchgeführt werden, bei denen lediglich die Arbeitsplatzzuweisung eines Mitarbeiters geändert wird.

Da beim Logistikproblem ein großer Fokus auf der optimalen Besetzung liegt, kommen ausschließlich Shift-Moves zum Einsatz, wobei auch Qualifikationen und Anwesenheiten berücksichtigt werden. Das Suchverfahren durchläuft somit die komplette Nachbarschaft und prüft für alle anwesenden Mitarbeiter nacheinander alle Arbeitsplatzzuweisungen, für die der jeweilige Mitarbeiter qualifiziert ist. Der Schritt wird dann bei dem Mitarbeiter mit dem Arbeitsplatz ausgeführt, der am besten ist. Die Tabuliste und das Aspirationskriterium finden dabei natürlich Anwendung.

Akzeptanzkriterium Bzgl. des Akzeptanzkriteriums werden gewöhnlich zwei Varianten unterschieden [295, S. 4–5].

- Best Accept: Es wird die Lösung aus der Nachbarschaft ausgewählt, die das beste

	Periode 1	Periode 2	Periode 3	Periode 4	Periode 5
Mitarbeiter 1	0	0	1	2	1
Mitarbeiter 2	0	0	0	0	1
Mitarbeiter 3	0	1	1	1	1
Mitarbeiter 4	0	0	0	1	1
Mitarbeiter 5	0	0	0	1	1
Mitarbeiter 6	0	0	1	1	2
Bedarf auf AP 1	0	1	3	4	5
Bedarf auf AP 2	0	0	0	1	1
Defizit	/	/	/	/	/



	Periode 1	Periode 2	Periode 3	Periode 4	Periode 5
Mitarbeiter 1	0	0	1	1	1
Mitarbeiter 2	0	0	0	0	1
Mitarbeiter 3	0	1	1	1	1
Mitarbeiter 4	0	0	0	1	1
Mitarbeiter 5	0	0	1	1	1
Mitarbeiter 6	0	0	1	2	2
Bedarf auf AP 1	0	1	3	4	5
Bedarf auf AP 2	0	0	0	1	1
Defizit	/	/	/	/	/

Abbildung 4.13: Swap-Move [178, S. 30]

Ergebnis liefert und nicht tabu gesetzt ist.

- First Accept: Es wird die erste Lösung aus der Nachbarschaft ausgewählt, die zu einer Verbesserung führt. Mögliche bessere Nachbarschaftslösungen bleiben dadurch unberücksichtigt. Die Anzahl an Berechnungen je Iteration wird jedoch reduziert. Sollte keine Lösung gefunden werden, die zu einer Verbesserung führt, so greift Best Accept.

Da die Suchintensität möglichst groß sein soll, wird für das Logistikproblem Best Accept als Akzeptanzkriterium gewählt.

Aspirationskriterium In der Tabuliste werden keine kompletten Lösungen hinterlegt, sondern lediglich die Ausgangswerte und Positionen geänderter Matrixelemente. Dadurch können bessere Lösungen jedoch versehentlich tabu gesetzt werden. Um dem zu begegnen, wird das Kriterium Aspiration by Objective verwendet. Somit ist es möglich, einen eigentlich tabu gesetzten Schritt dennoch auszuführen, wenn er zu einer Verbesserung der Lösung führt.

Grundsätzlich kann der Zustand eintreten, dass alle Nachbarn tabu gesetzt sind. Dies

ist ein Zeichen dafür, dass diese Region des Lösungsraums bereits intensiv durchsucht wurde. Um die Diversifikation zu erhöhen, wird in diesem Fall ein zufälliger Zeitintervall und ein zufälliger Mitarbeiter gesucht, der in diesem Zeitintervall anwesend ist. Dieser Punkt im Lösungsraum stellt den Startpunkt für die weitere Suche dar.

Tabudauer Für die Tabudauer wird kein fester Wert, sondern die dynamische Variante Robust TS von Taillard [365] gewählt. Diese wurde in der jüngeren Vergangenheit häufig bei kombinatorischen Problemstellungen angewendet. Bei Robust TS wird für jeden Eintrag in der Tabuliste eine Zufallszahl berechnet, die innerhalb einer minimalen T_{min} und maximalen T_{max} Tabudauer liegt.

Für das Logistikproblem wurden Tests durchgeführt, um die optimalen Grenzen zu finden, da in der Literatur keine allgemeingültigen Aussagen zu finden sind. Es zeigt sich, dass es bis zu einer Tabudauer von 150 Iterationen keine wesentlichen Unterschiede in den Ergebnissen gibt. Erst ab diesem Wert werden die Ergebnisse schlechter, da die Tabuliste die Suche dann zu sehr einschränkt. Daher wird T_{min} auf 10 und T_{max} auf 150 gesetzt.

Ablauf Nachfolgend ist der Ablauf von TS in Algorithmus 8 dargestellt. Er enthält alle zuvor geschilderten Anpassungen an das Logistikproblem. Nach der zufälligen Initialisierung der Startlösung wird jeweils mit Shift Moves in der aktuellen Nachbarschaft gesucht. Die beste Lösung wird aus \hat{X} ausgewählt. Der Schritt wird durchgeführt, wenn \hat{X} nicht tabu ist. Sollte \hat{X} tabu sein, so wird der Schritt dennoch getätigt, wenn er zu einer neuen Bestlösung führt. Am Ende jeder Iteration wird die Tabuliste aktualisiert. Die neue Position – bestehend aus den Informationen Mitarbeiter, Zeitintervall und Arbeitsplatz – wird in der Tabuliste abgelegt. Die Tabudauer T wird zufällig in den Grenzen $T_{min} = 10$ und $T_{max} = 150$ berechnet. Zudem werden alle Tabudauern um eins reduziert. Außerdem werden für die nächste Iteration der neue Startpunkte und die Ausgangslösung festgelegt.

Das Abbruchkriterium beträgt wie bei HC, ES und PSO 400.000 Berechnungen der Fitnessfunktion (siehe Kapitel 4.3). Da TS bei den Tagesproblemen jedoch erheblich besser abschneidet als bei der kompletten Woche, werden Experimente am Wochenproblem zusätzlich mit 2,8 Millionen Berechnungen der Fitnessfunktion ($400.000 * 7$ Tage) durchgeführt. Diese TS-Variante trägt die Bezeichnung TS*.

Algorithmus 8 Tabu Search bei der untertägigen Personaleinsatzplanung in der Logistik

```

1: Erzeuge Startlösung  $X$ 
2: Beste Lösung  $B = X$ 
3: Random Startpunkt  $I$  aus Mitarbeiter und Zeitintervall
4: while
5: Abbruchkriterium nicht erfüllt do
6:   Generiere alle Lösungen  $\hat{X}$  aus  $N$  von  $I$  durch Shift Move
7:   Fitness von lokaler Bestlösung  $LB = \infty$ 
8:   Berechne die Fitness aller  $\hat{X}$ 
9:   Wähle  $\hat{X}$  mit minimaler Fitness
10:  if  $f(\hat{X}) \leq f(LB)$  &&  $\hat{X} \neq \text{tabu}$  then
11:     $f(LB) = f(\hat{X})$ 
12:  else if  $f(\hat{X}) < f(B)$  &&  $\hat{X} = \text{tabu}$  then
13:     $f(LB) = f(\hat{X})$ 
14:  end if
15:  Aktualisiere Tabuliste bei  $\hat{I}$  mit  $T = \text{random}(T_{min}; T_{max})$ 
16:  Reduziere alle  $T$  um 1
17:  if  $f(LB) < f(B)$  then
18:     $B = LB$ 
19:  end if
20:  if  $\hat{I} = I$  then
21:    random Startpunkt  $I$  aus Mitarbeiter und Zeitintervall
22:  else
23:     $I = \hat{I}$ 
24:  end if
25:   $X = \hat{X}$ 
26: end while

```

4.11 Simulated Annealing

Simulated Annealing (SA) wird zunächst grundlegend erläutert. Anschließend wird in Kapitel 4.11.2 auf Anwendungen von SA in der Literatur eingegangen. Baker [22] wendet diese Metaheuristik auf das British Telecom-Problem an (Kapitel 4.11.3). Bei anderen Problemstellungen dieser Arbeit wird SA nicht erprobt.

4.11.1 Grundlegender Ablauf

SA ist eine punktbasierte naturanaloge Heuristik, die der Gruppe der Lokalen Suche zugeordnet wird. Sie wurde unabhängig voneinander von Kirkpatrick, Gelatt und Vecchi

[213] sowie von Cerny [66] entwickelt. Die Idee von SA orientiert sich an der Abkühlung von glühendem Metall. Die Atome bewegen sich zunächst sehr stark, da ihre Bindungen untereinander gelöst sind. Bei zunehmender Abkühlung lässt die Bewegung nach, bis das Kristallgitter schließlich vollständig erstarrt und einen energiearmen Zustand einnimmt. Übertragen auf die Lokale Suche werden Verschlechterungen bei einem Schritt mit einer bestimmten Wahrscheinlichkeit p in Kauf genommen. Sie sinkt im Laufe der Zeit, bis keinerlei Verschlechterungen mehr zulässig sind. SA ist also prinzipiell in der Lage, aus einem lokalen Optimum zu entkommen.

Die Parameter, die die Eigenschaften von SA bestimmen, sind die Abkühlrate, die Starttemperatur T_0 und die Anzahl an Schritten für die die Temperatur T_t beibehalten wird. SA beginnt mit einer Startlösung. Anschließend wird eine Schleife durchlaufen, bis ein Abbruchkriterium greift. In der Schleife werden Schritte zu anderen Lösungskandidaten berechnet. Führen sie zu keiner Verschlechterungen, werden sie sofort übernommen. Würde jedoch eine Verschlechterung eintreten, so wird die Wahrscheinlichkeit p berechnet, mit der dieser Schritt dennoch ausgeführt wird. Die Berechnung geschieht wie folgt.

$$p = \exp\left(\frac{f(\text{aktuellePosition}) - f(\text{neuePosition})}{T_t}\right) \quad (4.9)$$

Dabei steht T_t für die Temperatur zum Zeitpunkt t . Für sehr hohe Werte von T_t ist p fast eins – ein Schritt wird nahezu immer ausgeführt. Es ist ersichtlich, dass die Wahrscheinlichkeit p nicht nur von T_t , sondern auch von der Differenz beider Fitnesswerte abhängt. Kleine Verschlechterungen werden demzufolge mit einer größeren Wahrscheinlichkeit angenommen. Mathematisch haben van Laarhoven and Aarts 1987 [390] bewiesen, dass SA bei unendlich kleinen Temperaturänderungen immer das globale Minimum findet. Im praktischen Einsatz ist dies jedoch nicht hilfreich, da ein Ergebnis nach endlicher Zeit vorliegen muss. Daher erfolgt die Abkühlung in größeren Schritten, wobei die beiden folgenden Varianten am häufigsten zu finden sind. Meist wird die Temperatur T_t durch eine feste Abkühlrate α reduziert, wobei α gewöhnlich im Bereich $0,8 < \alpha < 1$ liegt. Aber auch Ansätze mit variablen Abkühlraten sind zu finden. Durch die Abkühlung wird die Temperatur immer weiter reduziert. Erreicht sie den Nullwert, handelt es sich bei dem Optimierungsverfahren nur noch um einen HC.

4.11.2 Anwendungen in der Literatur

SA wurde 1987 von van Laarhoven und Aarts [390] und 1989 von Aarts und Korst umfangreich untersucht.

Auf die Problemstellung der British Telecom wendet Baker 1993 [22] SA an. Die Heuristik war Teil des Dynamic Schedulers, der von der British Telecom zur Steuerung

mehrerer tausend Techniker genutzt wurde. Das Verfahren von Baker wird nachfolgend in Kapitel 4.11.3.1 beschrieben.

Brusco und Jacobs wenden sich 1993 [53] mit SA der Erzeugung optimaler Schichtzyklen zu. Sie vergleichen ihren Ansatz mit einigen Heuristiken und mit IP, wobei SA am besten abschneidet.

Im gleichen Jahr veröffentlichen Brusco und Jacobs ihre Arbeit [54], in der sie SA für Problemstellungen des Tour Schedulings anwenden. Die Problemstellungen sind dabei durch die Vernachlässigung vieler Nebenbedingungen gekennzeichnet, was die Einsatzplanung sehr flexibel gestaltet. Nach Brusco und Jacobs eignet sich SA sehr gut für derartige Anwendungsfälle.

Für die Dienstplanung mit sehr individuellen Mitarbeiterverfügbarkeiten und Arbeitsverträgen entwickelt Thompson 1996 [370] einen SA-Ansatz. Er testet die Suche unter verschiedenen Nachbarschaftskonzepten an 114 Testproblemen. An 20 Testproblemen vergleicht er seinen Ansatz mit IP, wobei die Ergebnisse von SA im Durchschnitt nur 0.29% schlechter sind – dafür jedoch nur 8.4% der Rechenzeit benötigen, um einen Plan zu erstellen.

Bailey, Garner und Hobbs wenden sich 1997 [20] ebenfalls der Dienstplanung in einem Krankenhaus zu. 27 Krankenschwestern, Teambildung und Qualifikationen sind dabei u.a. zu berücksichtigen. Sie vergleichen einen GA und SA an der Problemstellung. Beide Verfahren produzieren in sehr kurzer Zeit Ergebnisse, die optimal sind bzw. sehr nahe am Optimum liegen.

Dowling u.a. nutzen 1997 [101] SA für die Personaleinsatzplanung im Ground Handling. Konkret geht es um die Einsatzplanung für eine Fluggesellschaft mit 500 Mitarbeitern auf einem internationalen Flughafen. Die Arbeit von Dowling u.a. hat einen starken Anwendungsfokus. So beschreiben sie sehr detailliert das Einsatzplanungssystem und die Problemstellung der Airline, welche äußerst komplex ist. Der Planungshorizont beträgt einen Monat, in dem Dienste und Abwesenheiten geplant werden. Zudem müssen Aufgaben verplant und Servicelevel eingehalten werden. Und auch kurzfristige Änderungen müssen behandelt werden.

Eine weitere Arbeit im Aviation-Bereich wurde 1997 von Proksch [313] veröffentlicht. Er führt sehr umfangreiche Untersuchungen an drei Crew Scheduling Problemen durch und vergleicht SA mit verschiedenen Parametereinstellungen. Zudem zieht er einige einfache Lokale Suchverfahren zum Vergleich heran, die schlechtere oder gleich gute Ergebnisse wie SA liefern. Zusammenfassend stellt er fest, dass sich SA sehr gut für die drei Problemstellungen eignet und auch für erheblich umfangreichere Probleme überaus gut anwendbar sein dürfte. Problematisch ist aus seiner Sicht, dass recht aufwändige Parametereinstellungen notwendig sind, um gute Ergebnisse zu erzielen. Emden-Weinert und Proksch gehen 1999 [113] ebenfalls auf dieses Thema ein.

4.11.3 Anpassung an hochflexibles Workforce Management

SA wird in dieser Arbeit lediglich am British Telecom-Problem angewendet. Dabei wird auf die Implementierung durch Baker [22] zurück gegriffen. Eine Implementierung beim Logistik- und Handelsproblem findet nicht statt.

4.11.3.1 Kombination von Touren- und Personaleinsatzplanung

Baker setzt 1993 SA zur Lösung des Problems BT_Mod-250-118 ein [22]. Die Arbeit von Baker [22] war nicht zu beschaffen, sodass Details zur konkreten Umsetzung, die über das hier Dargestellte hinaus gehen, nicht verfügbar sind. Eine umfassende Beschreibung des Einsatzes dieses Verfahren bei der British Telecom ist jedoch in der Arbeit von Lesaint u.a. [239] zu finden.

Baker bildet die Problemstellung als Menge von Touren ab, wobei jedem Techniker eine Tour zugeordnet ist, die auch leer sein kann. Nicht ausführbare Aufträge werden auf einen Dummy-Techniker geplant. Für die Erzeugung eines Moves stehen die vier Aktionen 2-opt, relocate, exchange und cross zur Verfügung (siehe Abbildung 4.14).

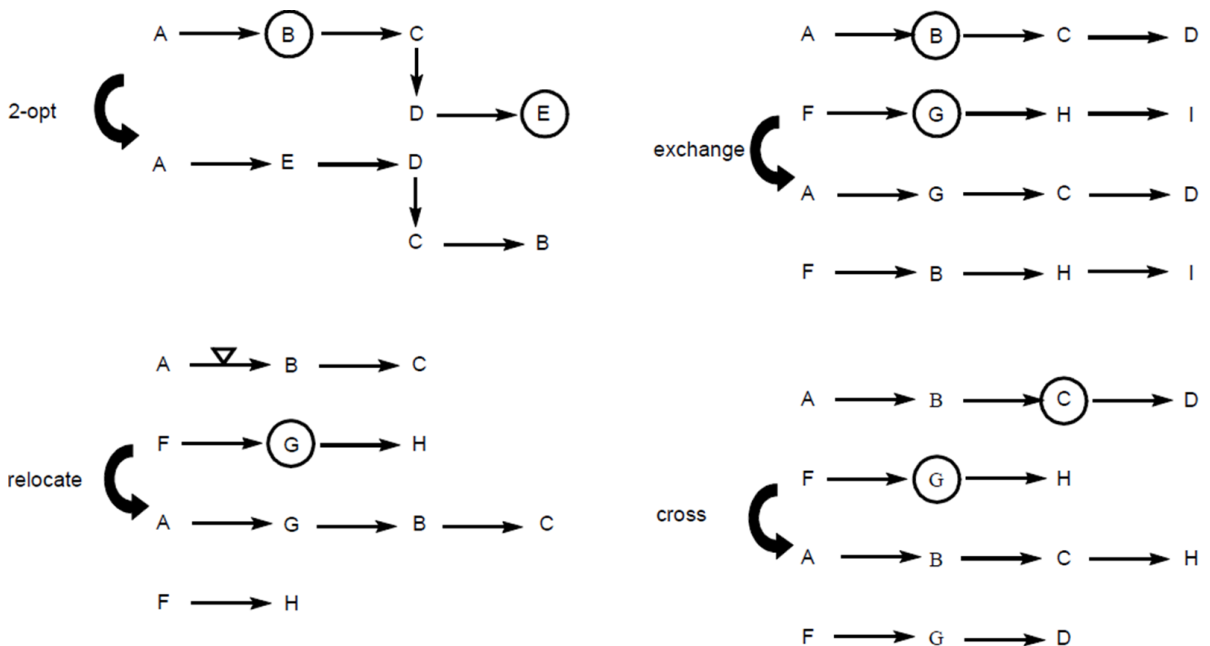


Abbildung 4.14: Move-Operatoren für Simulated Annealing [239, S. 23]

Lesaint u.a. [239] messen der Aktion relocate im Vergleich zu den anderen Aktionen die größte Bedeutung zu. Sie sagen, dass die anderen Aktionen ihre Effizienz bei den Ne-

benbedingungen des British Telecom-Problems verlieren. Relocate besetzt einen Auftrag um. Dies kann innerhalb derselben Tour oder in einer anderen Tour geschehen.

4.12 Evolutionsstrategien

Evolutionsstrategien (ES) werden in dieser Arbeit sehr umfassend untersucht. Zunächst wird auf grundlegende Zusammenhänge dieser Metaheuristik u.a. im Rahmen der Evolution eingegangen. In Kapitel 4.12.2 werden anschließend der Ablauf und in Kapitel 4.12.3 die Anwendungen von ES in der Literatur erläutert. Eigene Implementierungen werden bei allen drei Problemstellungen durchgeführt. Die entsprechenden Ausführungen hierzu sind in Kapitel 4.12.4 zu finden.

4.12.1 Grundlegende Zusammenhänge

1859 erschien Darwins berühmtes Buch „On the Origin of Species By Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life“ [91]. Dort beschreibt er die Theorie der Evolution durch Selektion. Mit diesem Werk setzt er einen Meilenstein in der modernen Evolutionstheorie, obgleich er das Substantiv „Evolution“ nicht benutzt [77, S. 6].

Ridley definiert Evolution wie folgt [327, S. 4]: „Evolution means change, change in the form and behavior of organisms between generations.“ Eine vereinfachte Darstellung der Evolution in der Biologie ist in Abbildung 4.15 zu finden. Man erkennt dort drei grundlegende Prinzipien: Selektion, Rekombination und Mutation.

- Selektion: Hier erfolgt die Auswahl der Individuen, die Eltern der nächsten Generation sein werden. Also zum einen durch eine gute Anpassung an die Umwelt überleben und sich somit durchsetzen. Zum anderen werden Eltern in der Selektion ausgewählt, die jeweils an der Rekombination teilnehmen. Die Selektion tritt in Abbildung 4.15 daher zweimal auf.
- Rekombination: Die Rekombination (oder Crossover) repräsentiert den Fortpflanzungsprozess. Es werden einzelne Gene oder Genabschnitte der Eltern zur Erzeugung von Nachkommen durchmischt.
- Mutation: Sie sorgt für Diversität in der Population. Das Erbgut wird dabei ungerichtet geändert, wobei kleine Änderungen wahrscheinlicher sind als große.

Die Evolution in der Biologie ist kein Optimierungsprozess – vielmehr ist sie ein An-

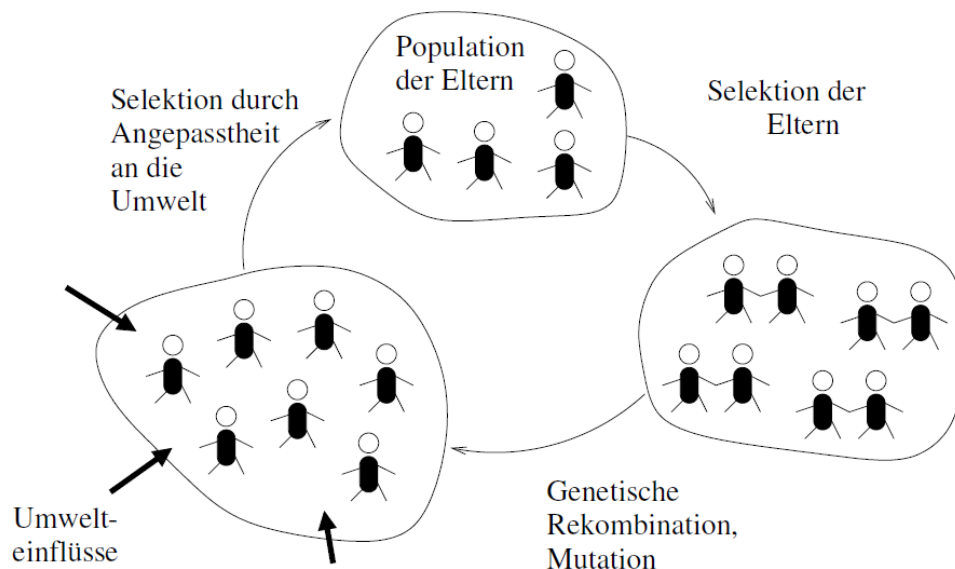


Abbildung 4.15: Vereinfachte Darstellung der Evolution in der Biologie [414, S. 28]

passungsprozess. Evolutionäre Algorithmen (EA) nehmen sich die biologische Evolution als Vorbild und haben das Ziel, eine Problemstellung zu lösen. Nachfolgend ist der von Weicker entwickelte schematische Ablauf für einen EA in Abbildung 4.16 dargestellt.

Bei EAs lassen sich vier Hauptformen unterscheiden, die kurz erläutert werden [284, S. 13–180], [414, S. 29–30], [415, S. 127–154].

- Genetische Algorithmen (GA): Dieses Verfahren geht auf Holland zurück [186], [187], [188]. Jedes Individuum stellt eine Lösung der Problemstellung dar, wobei diese binär oder reell codiert ist. Ein Gen nimmt dann entweder den Wert null oder eins an. Die Operatoren von GA arbeiten also auf dem Genotyp (codierte Lösung). Die Selektion erfolgt stochastisch und fitnessproportional. Daher haben auch schlechte Individuen eine Chance zu überleben. Die Mutation hat eine nachrangige Bedeutung bei GA. Sie verfügt auch über keine selbstadaptierenden Strategieparameter wie bei ES. Der wichtigste Suchoperator für GA ist die Rekombination.
- Genetische Programmierung (GP): Diese Variante wurde im Wesentlichen von Koza geprägt [224], [225]. Bei GP wird ein Individuum als eigenes Programm aufgefasst. Die Suche erfolgt in einem Lösungsraum aller Programme. Damit nimmt man zwangsläufig einen sehr großen Suchraum in Kauf. Koza arbeitet mit Baumstrukturen für die Darstellung der Programme, für die er die Programmiersprache LISP wählt.
- Evolutionsstrategien (ES): ES gehen auf Rechenberg [318], [319] und Schwefel [342] zurück. So wendet Rechenberg ES 1973 erstmalig für die Entwicklung widerstandsminimaler Strömungskörper an. ES arbeitet direkt auf dem Phänotyp (decodierte

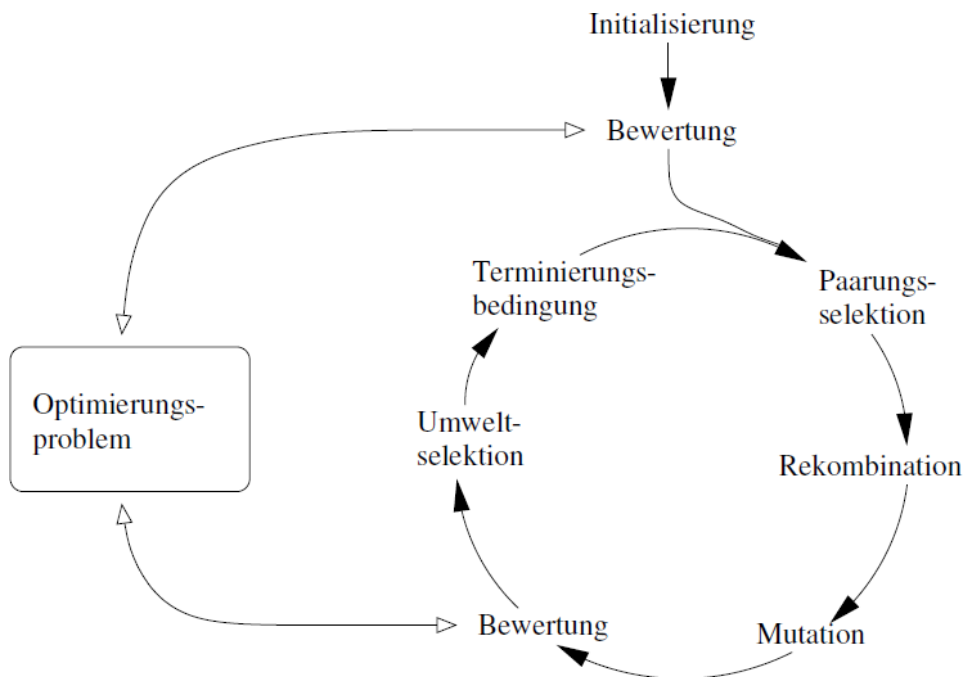


Abbildung 4.16: Schematischer Ablauf eines EA [414, S. 29]

Lösung). Die Lösung jedes Individuums ist also nicht binärcodiert, wie dies meist bei GA der Fall ist. In der Umweltselektion werden deterministisch die besten Individuen ausgewählt. Der bedeutendste Suchoperator ist die Mutation – wobei die Strategieparameter zur Steuerung der Mutation meist selbstadaptierend sind. ES ist ohne Rekombination anwendbar – ihre Einbeziehung kann sich jedoch als vorteilhaft erweisen. Bei Anwendung der Rekombination werden die jeweiligen Eltern zufällig gewählt.

- Evolutionäre Programmierung (EP): Fogel u.a. haben diese Form entwickelt [122]. EP hat viele Gemeinsamkeiten mit ES. So arbeitet EP ebenfalls auf dem Phänotyp. Denn das Problem soll möglichst natürlich dargestellt werden. Es existiert jedoch keine Vorschrift für die Lösungsrepräsentation. Bei EP liegt der Fokus ebenfalls auf der Mutation. Wie bei ES existieren auch hier selbstadaptierende Strategieparameter. Zudem gibt es eine Selektion – jedoch keine Rekombination. Die Idee ist, dass die Population nicht aus Individuen, sondern aus Spezies besteht. Da es zwischen Spezies keine Rekombination gibt, wird sie auch nicht bei EP benutzt.

Nissen liefert eine Gegenüberstellung der vier Hauptformen anhand der wichtigsten Kriterien, die nachfolgend in Tabelle 4.5 dargestellt ist.

Tabelle 4.5: Unterschiede der vier EA-Hauptformen [286, S. 242]

Kriterium	GA	GP	ES	EP
Lösungsrepräsentation	oft binär, auch reell	Programm	reell	reell
Fitnessmaß	skalierter Zielfunktionswert	skalierter Zielfunktionswert	Zielfunktionswert	(evtl. skalierter) Zielfunktionswert
Selektion	stochastisch, nicht diskriminierend	stochastisch, nicht diskriminierend	deterministisch, diskriminierend	stochastisch, diskriminierend
Mutation	Hintergrundoperator	uneinheitlich verwendet	Hauptsuchoperator	einziger Suchoperator
Rekombination	Hauptsuchoperator	Hauptsuchoperator	wichtig v.a. für Selbstadaptivität	nicht verwendet
Selbstadaptivität	noch wenig Erfahrung	noch wenig Erfahrung	Standardabweichung, Rotationswinkel	Standardabweichung (in Meta-EP)

4.12.2 Grundlegender Ablauf

Zuvor wurden ES bereits kurz in der Gegenüberstellung mit anderen EA-Hauptformen skizziert. In diesem Kapitel wird der Ablauf von ES umfassend erläutert. Er ist in Algorithmus 9 dargestellt, wobei auf die einzelnen Operatoren gesondert in den nachfolgenden Unterkapiteln eingegangen wird.

Algorithmus 9 ES

- 1: Intitialisiere die Population mit μ Individuen
 - 2: Berechne die Fitness der μ Individuen
 - 3: **repeat**
 - 4: Rekombination zur Erzeugung von λ Nachkommen
 - 5: Mutation der λ Nachkommen
 - 6: Berechne die Fitness der Nachkommen
 - 7: Selektion, um die neue Generation mit μ Individuen zu erzeugen
 - 8: **until** Abbruchkriterium tritt ein
-

Zunächst erfolgt die Initialisierung der Population mit μ Individuen. Dies sind die El-

tern für die kommende Generation. Nach der Initialisierung wird die Fitness der Individuen mit Hilfe der Fitnessfunktion berechnet. Die nun folgenden Aktionen befinden sich in einer Schleife, die endet, sobald das Abbruchkriterium greift. Zunächst werden λ Nachkommen durch Rekombination der Eltern erzeugt. Diese Nachkommen werden anschließend mutiert und ihre Fitness wird berechnet. Die Mutation ist bei ES der wichtigste Suchoperator. Sie dient auch dem Entkommen aus lokalen Optima. Meist wird die Mutation durch einen oder mehrere selbstadaptierende Strategieparameter gesteuert. Die Idee ist, dass ein Individuum nicht nur eine Lösung, sondern auch einen oder mehrere Strategieparameter enthält. Diese werden ebenfalls mutiert und rekombiniert. Gute Strategieparameter führen zu einer guten Mutationsschrittweite, die eher zu guten Lösungen führt, sodass das Individuum folglich mit einer höheren Wahrscheinlichkeit überlebt. Auf diese Weise setzen sich gute Individuen mit guten Strategieparametern durch (selbstverständlich existieren auch Mutationsvarianten ohne Strategieparameter – siehe Kapitel 4.12.2.2). Als letzter Schritt, bevor erneut mit der Rekombination begonnen wird, findet die deterministische Selektion statt, bei der nur die besten Individuen überleben. Hier werden die μ Eltern der neuen Generation ermittelt. Auf die Rekombination, Mutation und Selektion wird in den folgenden Unterkapiteln detailliert eingegangen.

4.12.2.1 Rekombination

Die Rekombination dient dazu, aus zwei oder mehr Eltern Nachkommen zu erzeugen. Bei ES werden für ein Kind nicht nur die Werte der Entscheidungsvariablen sondern auch die Werte der Strategieparameter festgelegt.

Verbreitete Formen der Rekombination werden von Nissen [284] genannt. Einen guten Überblick über die Rekombination liefert Pohlheim [304, S. 34–45]. Er geht auf Varianten der Rekombination bei unterschiedlichen Lösungsräumen ein. Zwei wichtige Rekombinationsformen mit zwei Eltern sind die diskrete Rekombination, die häufig für Entscheidungsvariablen verwendet wird sowie die intermediäre Rekombination, die häufig für Strategieparameter genutzt wird [284, S. 152]. Der beispielhafte Ablauf beider Formen ist in den Tabellen 4.6 und 4.7 dargestellt. Bei der diskreten Rekombination wird für jedes Vektorelement eine Zufallszahl ermittelt, mit der entweder der Wert aus Elter eins oder Elter zwei auf das Kind übertragen wird. In der intermediären Rekombination wird auf das Kind der Durchschnittswert beider Eltern im jeweiligen Vektorelement geschrieben. Für den beispielhaften Ablauf einer intermediären Rekombination und einer diskreten Multirekombination sei auf die Arbeit von Biethahn u.a. verwiesen [34, S. 82–83].

Tabelle 4.6: Diskrete Rekombination [34, S. 76–77]

Vektorelement	1	2	3	4	5	6
Elter 1	1,8	0,8	3,0	0,6	1,2	2,1
Elter 2	3,8	2,2	3,0	0,8	0,2	3,1
Nachkomme	3,8	2,2	3,0	0,8	1,2	2,1

Tabelle 4.7: Intermediäre Rekombination [34, S. 76–77]

Vektorelement	1	2	3	4	5	6
Elter 1	1,8	0,8	3,0	0,6	1,2	2,1
Elter 2	3,8	2,2	3,0	0,8	0,2	3,1
Nachkomme	2,8	1,5	3,0	0,7	0,7	2,6

4.12.2.2 Mutation

Die Mutation ist der entscheidende Suchoperator bei ES. Dabei kommt der richtigen Wahl der Mutationsschrittweite eine entscheidende Bedeutung zu. Ist sie zu klein, sind die Unterschiede zwischen Eltern und Nachkommen gering. Der Suchfortschritt ist demnach sehr langsam. Ist die Mutationsschrittweite sehr groß, wird das Nachkommen zu stark geändert, was zu Verschlechterungen der Fitness führen kann. Auch eine konstante Mutationsschrittweite hemmt den Suchfortschritt, da dies zu einer Zufallssuche führt [341, S. 20].

Nachfolgend werden Mutationsvarianten beschrieben, mit denen eine gerichtete Suche möglich ist.

Mutation mit der 1/5-Erfolgsregel Die Mutation mit der 1/5-Erfolgsregel ist eine Mutation mit adaptiver Schrittweitensteuerung. Hier sind keine Strategieparameter erforderlich. Die 1/5-Erfolgsregel wurde von Rechenberg für eine (1 + 1)-ES aufgestellt [318]. Er untersuchte den Zusammenhang zwischen Fortschrittsgeschwindigkeit ϕ und Erfolgswahrscheinlichkeit der Mutation W_e bei zwei Problemstellungen mit dem Index e . Unter der Erfolgswahrscheinlichkeit versteht man die Wahrscheinlichkeit, mit der ein Nachkomme eine bessere Fitness hat als sein Elter. In Abbildung 4.17 ist dieser Zusammenhang für das Korridor- und das Kugelmodell abgebildet. Beim Korridormodell ist $W_{Korr} \approx 0,184$ also etwa ein Sechstel und das Kugelmodell $W_{Kug} \approx 0,270$ also etwa ein Viertel. Aus dem Durchschnitt von ein Sechstel und ein Viertel ergibt sich ein Fünftel. [221, S. 117–125], [27, S. 5–6]

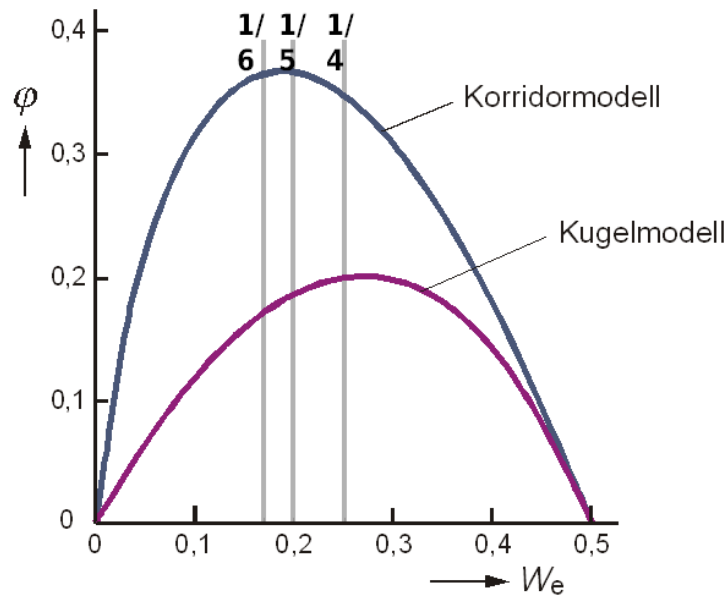


Abbildung 4.17: Zusammenhang zwischen Fortschrittsgeschwindigkeit und Erfolgswahrscheinlichkeit der Mutation beim Korridor- und Kugelmodell [320, S. 26]

Rechenberg definiert die 1/5-Erfolgsregel wie folgt [318, S. 122]: „Das Verhältnis der erfolgreichen Mutationen zu allen Mutationen sollte 1/5 sein. Wenn es größer ist als 1/5, erhöhe die Varianz, wenn es weniger ist, verringere die Mutationsvarianz.“ Die Idee ist also, die Mutationsschrittweite zu verringern, je näher man dem möglichen Optimum kommt.

Zur Änderung der Mutationsschrittweite gibt Schwefel eine Handlungsvorschrift [342]. So muss die Erfolgswahrscheinlichkeit der Mutation über mehrere Mutationen betrachtet werden. Nach n Mutationen soll daher geprüft werden, wie viele Verbesserungen in den letzten $10n$ Mutationen aufgetreten sind. Bei weniger als $2n$ Verbesserungen wird die Mutationsschrittweite um den Faktor 0,85 erhöht. Bei mehr als $2n$ Verbesserungen wird sie um den Faktor 0,85 herab gesetzt. Mit diesem Vorgehen wird die Mutationsschrittweite nach der 1/5-Erfolgsregel dynamisch angepasst. [221, S. 118]

Klassische Gauß-Mutation Bei der Mutation mit der 1/5-Erfolgsregel ist die Schrittweitensteuerung sehr starr. Mit der nachfolgend dargestellten mutativen Schrittweitensteuerung lässt sich die Mutation problemabhängig anpassen [341, S. 22]. Der Grundgedanke ist u.a., dass in der Biologie Kinder ihren Eltern meist sehr ähnlich sehen. Kleine Änderungen in der Mutation treten demnach häufiger auf als große. Mit Hilfe der Normalverteilung (oder Gauß-Verteilung nach Carl Friedrich Gauß) wird dem Rechnung getragen. Bei der Mutation wird zunächst der Strategieparameter σ – also die

Standardabweichung für die Mutationsschrittweite – verändert [27, S. 4].

$$\sigma'_i = \sigma_i * \exp(\tau'_1 * N(0, 1) + \tau_2 * N_i(0, 1)) \quad (4.10)$$

In Formel 4.10 sind τ'_1 und τ_2 exogene Konstanten, wobei beide Werte häufig auf eins gesetzt werden. Der Index i steht für das jeweilige Vektorelement des zu mutierenden Kindes. Man erkennt, dass $\tau'_1 * N(0, 1)$ für alle Strategieparameter einheitlich gilt und $\tau_2 * N_i(0, 1)$ individuell für den Strategieparameter mit dem Index i gilt. Nach der Mutation der Strategieparameter werden die Werte der Entscheidungsvariablen mutiert. Die Formel für die Mutation der Entscheidungsvariable X_i lautet wie folgt (in Anlehnung an [284, S. 142]).

$$x'_i = x_i + N(0, \sigma'_i) \quad (4.11)$$

Auf die Variable x_i wird demnach eine normalverteilte Zufallszahl mit der Standardabweichung σ'_i addiert. Dieses Vorgehen wird für alle Entscheidungsvariablen vollzogen. Auf diese Weise ergibt sich eine selbstanpassende Steuerung der Mutation.

Mutation mit dem Konzept der maximalen Entropie Die Problemstellungen in dieser Arbeit sind kombinatorisch und haben die folgende Form.

$$\min \{f(x) : x \in M \subseteq \mathbb{N}_0^n\} \quad (4.12)$$

Es gilt zu beachten, dass der Lösungsraum beschränkt ist, da sich die Menge M der einzuplanenden Arbeitsplätze oder Aufträge immer auf tatsächlich existierende Arbeitsplätze bzw. Aufträge bezieht. Weil die Entscheidungsvariable nur Werte der Menge \mathbb{N}_0 annehmen kann, sollte auch das Vorgehen bei der Lösung auf den Lösungsraum der natürlichen Zahlen abgestimmt sein.

Rudolph wendet 1994 [329] $ES(\mu + \lambda)$ in einem unbegrenzten Lösungsraum mit ganzen Zahlen an fünf Problemstellungen an. Für die Mutation nutzt er das Konzept der maximalen Entropie, um eine geeignete Wahrscheinlichkeitsfunktion für die verschiedenen Problemstellungen zu finden. Ein Individuum besteht aus einem Vektor $x \in \mathbb{Z}^n$ und aus einem Parameter $s \in \mathbb{R}_+$, mit dem die Mutation gesteuert wird. In jeder Generation wird s mutiert: $\hat{s} = s * \exp(N)$, wobei N eine normalverteilte Zufallsvariable mit dem Mittelwert null und der Varianz $\tau^2 = 1/n$ ist. Die untere Grenze für \hat{s} ist eins. Mit diesem Vorgehen passt sich s kontinuierlich an, da davon ausgegangen werden kann, dass Individuen mit einer guten Schrittweite am ehesten überleben. Der zur Mutation erforderliche Parameter p ergibt sich nach folgender Formel:

$$p = 1 - \frac{S/n}{(1 + (S/n)^2)^{1/2} + 1} \quad (4.13)$$

Bei der Mutation von x wird auf jedes Vektorelement die Differenz zweier geometrischer Zufallsvariablen Z addiert. Z ergibt sich aus der Differenz von G_1 und G_2 . Dabei hängen G_1 und G_2 von p und somit von s ab. Zudem ist U gleichverteilt mit Mittelwert null und Varianz eins und wird bei jedem Vektorelement und für jedes G neu berechnet. Die Subtrahenden G_1 und G_2 ergeben sich wie folgt:

$$G_i = \left\lfloor \frac{\log(1 - U)}{\log(1 - p)} \right\rfloor \quad (4.14)$$

Rudolph nimmt in der Rekombination von Elter eins und zwei den Durchschnitt von S_1 und S_2 für das Kind. Der Vektor x für das Kind wird mittels Uniform Crossover erzeugt.

Evolutionsfenster Zuvor wurden Varianten der Mutation dargestellt. Bei ihnen hat die Mutationsschrittweite – die mittlere Größe der Änderungen in der Mutation – einen entscheidenden Einfluss auf die Fortschrittsgeschwindigkeit von ES. Eine zu kleine Mutationsschrittweite hemmt den Fortschritt. Es besteht die Gefahr der Stagnation. Eine zu große Mutationsschrittweite ist ebenfalls kontraproduktiv. Viele Änderungen können zu Rückschritten führen. Die Mutationsschrittweite muss folglich gut gewählt werden, um bestmögliche Ergebnisse zu erzielen [303, S. 2]. Rechenberg bezeichnet diesen schmalen Bereich der Mutationsschrittweite als Evolutionsfenster. Dieses ist beispielhaft in Abbildung 4.18 dargestellt. Dort stellt die Kurve die Fortschrittsgeschwindigkeit in Abhängigkeit von der Mutationsschrittweite dar. Gelingt es ES häufig, mit der Mutationsschrittweite im Evolutionsfenster zu liegen, ist der Fortschritt besonders groß. Für die exakte Wahl der Mutationsschrittweite bei beliebigen Zielfunktionen existiert keine Handlungsempfehlung. Zumal die richtige Mutationsschrittweite auch von der Position des Individuums im Lösungsraum abhängt. Hieraus leitet sich die Notwendigkeit der selbstadaptiven Mutationsschrittweite ab. [K1, S. 116–117]

Im übertragenen Sinn kann man sagen, dass links vom Evolutionsfenster erkonservative Parteien sitzen. Bei ihnen kommt es zur Stagnation sie wollen keine Veränderungen. Rechts vom Evolutionsfenster sitzen hingegen revolutionäre Parteien. Diese fordern radikale, tiefgreifende Änderungen, die zu einem Rückschritt führen. Der größte Fortschritt wird bei den Parteien erzielt, die im Evolutionsfenster sitzen. [341], [321]

4.12.2.3 Selektion

Bei ES existieren zwei Selektionsvarianten, die sich darin unterscheiden, ob zusätzlich zu den Nachkommen auch deren Eltern an der Selektion teilnehmen und somit überleben können oder nicht. Aus den Individuen, die für die Selektion in Frage kommen, werden dann die besten μ Individuen als Eltern der nächsten Generation ausgewählt.

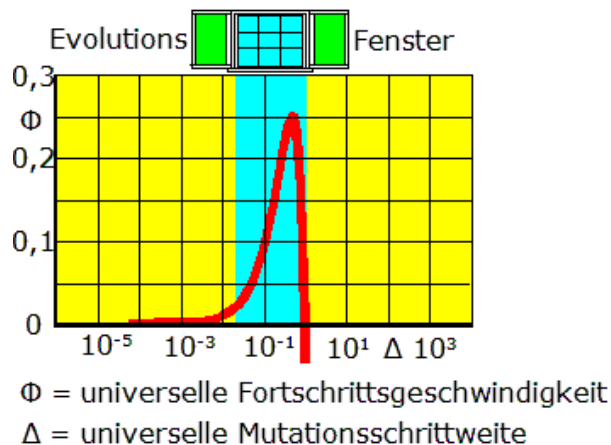


Abbildung 4.18: Evolutionsfenster [321]

- **Plusselektion:** Bei der Plusselektion werden die μ Eltern nicht aus der Selektion ausgeschlossen. Es nehmen Eltern und Nachkommen gemeinsam an der Selektion teil, wobei μ Eltern für die neue Generation ermittelt werden. Theoretisch kann ein Elter unendlich lang überleben, wenn seine Fitness besser als die des Nachkommen ist. Dies bewirkt, dass der beste Fitnesswert niemals schlechter werden kann als der beste Fitnesswert in der vorherigen Generation. Das Entkommen aus lokalen Optima kann so erschwert bzw. unmöglich sein, da die Eltern im lokalen Optimum nie die Population verlassen. Dies ist auch hinsichtlich selbstadaptierender Strategieparameter problematisch. Nämlich dann, wenn ein Individuum mit sehr schlechten Strategieparametern zufällig ein lokales Optimum gefunden hat. Mit hoher Wahrscheinlichkeit sind die mutierten Standardabweichungen der Nachkommen ebenfalls schlecht, sodass ihre Fitness sehr wahrscheinlich ebenfalls schlecht sein wird. Die Plusselektion hat aber auch einen Vorteil. Gute Lösungen gehen nicht verloren, was die Suche beschleunigen kann. [341, S. 17–18], [339, S. 10–11]
- **Kommaselektion:** In der Kommaselektion konkurrieren nur die Nachkommen miteinander. Eltern werden aus der Selektion ausgeschlossen. Sie werden gelöscht und haben in den kommenden Generationen keine Bedeutung. Jedes Individuum lebt nur für eine Generation. So kann die beste Fitness in einer Generation bei der Kommaselektion schlechter sein als in der vorherigen Generation. Hierdurch ist ein Entkommen aus lokalen Optima möglich. Ebenso tritt das bei der Plusselektion geschilderte Problem bzgl. der Strategieparameter nicht auf. [341, S. 17–18], [339, S. 9–10]

Selektionsdruck Mit Hilfe des Selektionsdrucks s kann das Verhalten von ES variiert werden. In der Natur wird der Selektionsdruck z.B. durch natürliche Feinde verändert. Er ergibt sich wie folgt.

$$s = \mu/\lambda \quad (4.15)$$

Er lässt sich durch Veränderung von μ und λ zwischen den Extremwerten 0 (starker Selektionsdruck) und 1 (schwacher Selektionsdruck) variieren. Ist λ erheblich größer als μ , können nur verhältnismäßig wenig Nachkommen in die neue Generation übernommen werden. Die Auslese ist sehr stark und der Selektionsdruck groß. Ist λ jedoch fast so groß wie μ , findet kaum Auslese statt. Der Selektionsdruck ist demnach gering. [273]

Populationswelle Eine weitere Möglichkeit, das Verhalten von ES zu beeinflussen, liegt in der Veränderung der Populationsgröße. Dies wird als Populationswelle bezeichnet. Man erreicht die Änderung in (μ, λ) - oder $(\mu + \lambda)$ -Strategien, indem μ geändert wird. Soll dabei der Selektionsdruck s unverändert bleiben, so muss λ proportional zu μ angepasst werden. [273]

4.12.2.4 ES-Varianten

Nachfolgend werden diverse ES-Varianten vorgestellt. Sie beinhalten eine unterschiedliche Anzahl an Eltern und Nachkommen sowie verschiedene Selektionsstrategien. Es wird unterstellt, dass immer eine Rekombination eingesetzt wird, wenn dies möglich ist. Auf die Angabe von ρ (Mischungszahl für Individuen) oder ρ' (Mischungszahl für Populationen) wird nachfolgend verzichtet, da in allen Experimenten dieser Arbeit eine Rekombination mit zwei Eltern ($\rho = 2$) vorgenommen wird, soweit diese möglich ist. Experimente mit Populationen werden nicht angestellt.

- (1+1)-ES: Die (1+1)-Selektion war die erste Variante, die für ES eingesetzt wurde. Ein Elter erzeugt genau ein Nachkommen. Bei der (1+1)-Selektion findet keine Rekombination statt, da es lediglich ein Elter gibt. Das Nachkommen wird durch Klone erzeugt.
- $(\mu+1)$ -ES: Hier existieren μ Eltern, die genau ein Nachkommen erzeugen. Wegen der Plusselektion nehmen alle Eltern und das Nachkommen an der Selektion teil, in der die μ Eltern der nächsten Generation ermittelt werden. Da $\mu > 1$ gilt, wird das Nachkommen über Rekombination erzeugt.
- $(\mu+\lambda)$ -ES: Neben den μ Eltern existieren λ Nachkommen, die durch Rekombination erzeugt werden. Alle Eltern und Nachkommen nehmen an der Selektion teil.
- (μ, λ) -ES: Hier existieren ebenfalls μ Eltern und λ Nachkommen. Dabei ist $\mu < \lambda$, da bei $\mu = \lambda$ keine Selektion erfolgt und eine Zufallssuche stattfindet [193, S. 44]. Wegen der Kommaselektion nehmen nur die λ Nachkommen an der Selektion der μ Eltern für die nächste Generation teil.

- $(\mu'+/\lambda' (\mu+/\lambda)^\gamma)$ -ES: Diese Form wird auch als geschachtelte ES bezeichnet. Sie ist in den Varianten $(\mu'+\lambda' (\mu+\lambda)^\gamma)$, $(\mu'+\lambda' (\mu,\lambda)^\gamma)$, $(\mu',\lambda' (\mu+\lambda)^\gamma)$ und $(\mu',\lambda' (\mu,\lambda)^\gamma)$ anwendbar. Hinter dieser ES-Variante steckt das Populationskonzept. Hier ist die Population und nicht mehr das Individuum Gegenstand der Selektion. Der Ablauf sieht so aus, dass zunächst μ' Populationen λ' neue Populationen erzeugen, welche für γ Generationen isoliert werden. Die so isolierten Populationen bestehen jeweils aus μ Eltern und λ Nachkommen. Nach Ablauf der γ Generationen werden die μ' besten isolierten Population als neue Startpopulationen gewählt, die wiederum λ' Populationen erzeugen. Im übertragenen Sinn kann dies mit Gipfelklettern oder Gipfelspringen verglichen werden [322]. Geschachtelte ES können bei komplexen multimodalen Problemstellungen erfolgreicher sein als die zuvor aufgeführten ES-Varianten. [273]

Bzgl. des optimalen Verhältnisses von μ/λ wurden von Schwefel Untersuchungen angestellt [343], [344]. Er hat sich zunächst $(1,\lambda)$ -ES zugewendet und die Ergebnisse auf ES mit $\mu > 1$ übertragen. Dabei stellt sich heraus, dass ein Verhältnisses für μ/λ von ein Fünftel bis ein Siebtel optimal ist. Dieses Verhältnis wird nachfolgend für die eigenen Experimente beibehalten.

An dieser Stelle sei der Vollständigkeit halber auf die ES-Variante von Aday verwiesen, welche er weitestgehend unabhängig von Rechenberg und Schwefel entwickelt hat [5]. Er verwendet die MUSE-Strategie, die für Mutations-Selektions-Strategie steht. Hierbei handelt es sich um eine Ansammlung von evolutionären Modulen, die für eine konkrete Problemstellung kombiniert und angepasst werden [284, S. 162–166].

4.12.3 Anwendungen in der Literatur

Anwendungen von ES für kombinatorische oder ganzzahlige Problemstellungen sind nicht sehr häufig zu finden. Nachfolgend wird kurz auf einige Arbeiten eingegangen.

Nissen wendet ES 1994 [284], [285] erfolgreich auf verschiedene Instanzen des quadratischen Zuordnungsproblems an. Neben ES mit angepasster Mutation und einer hybriden ES-Variante erprobt er EP und zwei GA-Varianten, wobei sich ES und EP als besser gegenüber den GA-Varianten erweisen. ES schneidet auch im Vergleich zu anderen Verfahren wie 2-Opt deutlich besser ab. Im Vergleich zu TS kann ES vor allem in der hybriden Variante mit anschließender 2-Opt-Heuristik überzeugen. Vielfach liefert ES allein auch bereits sehr gute Werte.

1994 entwickelt Rudolph [329] eine Mutation für diskrete Problemstellungen. Er nutzt das Konzept der maximalen Entropie, um für die Mutation eine geeignete Wahrscheinlichkeitsfunktion zu finden, die jeweils an die entsprechende Problemstellung angepasst ist. Da Rudolph sehr gute Ergebnisse erzielt hat, wird dieses Konzept in dieser Ar-

beit aufgegriffen und für kombinatorische Problemstellungen angepasst (siehe Kapitel 4.12.2.2).

Ende der 90er wendet Homberger [189], [190] ES mit einer Kommaselektion auf 56 Instanzen des Vehicle Routing Problem mit Zeitfenstern (VRPTW) von Solomon [353] an. Er entwickelt zwei ES-Varianten. In der ersten Variante existieren keine Strategieparameter. Vielmehr verfügt jedes Individuum über eine Mutationsvorschrift, die rekombiniert wird. Diese enthält eine Permutation der Aufträge, wobei jeder Auftrag genau zweimal aufgeführt ist. Während der Mutation wird die Permutation durchlaufen. Beim erstmaligen Auftreten eines Auftrags wird dieser Auftrag aus dem Kind ausgefügt. Beim zweiten Auftreten des Auftrags wird er wieder in das Kind eingefügt. Das Ausfügen und Einfügen unterliegt bestimmten Regeln. Die zweite ES-Variante läuft wie folgt ab: Es gibt keine Mutationsvorschrift. Die Mutation erfolgt über Or-Opt-Moves [293]. Dort wird mindestens ein Auftrag aus einer Tour ausgefügt und wieder in eine Tour eingefügt. Homberger verwendet dabei zwei Regeln, die alternierend angewendet werden. Die Anzahl der Or-Opt-Moves wird über einen Strategieparameter gesteuert. Beide ES-Varianten schneiden sehr gut ab. Sie erzielen bei 50 der 56 Instanzen exakt die bisher bekannten Bestwerte. Variante eins liefert bei weiteren vier Instanzen neue Bestwerte und Variante zwei findet bei fünf weiteren Instanzen neue Bestwerte.

Für das kombinatorische Survivable-Network-Design-Problem erproben Nissen und Gold 2008 [290] drei ES-Varianten. Dabei passen sie die Mutation an und implementieren eine Reparatur. Insgesamt führen sie Experimente an sechs Probleminstanzen durch und können für alle Probleme neue Bestwerte erzielen. ES ist in der Lage, die bisherigen GA zu schlagen.

Weitere Anwendungen von ES für ganzzahlige Problemstellungen sind z.B. die Arbeit von Herdy [182] für einen diskreten Lösungsraum oder die Arbeit von Bäck [26] für einen binären Lösungsraum. Auf kombinatorische Probleme wird ES zudem von Beyer und Schwefel [30] sowie von Schindler, Rothlauf und Pesch [340] angewendet. Insgesamt wird deutlich, dass sich ES durchaus für kombinatorische und ganzzahlige Problemstellungen eignen kann.

4.12.4 Anpassung an hochflexibles Workforce Management

In diesem Kapitel werden die Implementierungen von ES bei allen drei Problemstellungen dieser Arbeit beschrieben. ES mit klassischer Gauß-Mutation beim Logistikproblem ohne Stamarbeitsplatzregelung wurde von Birnstiel im Rahmen einer Diplomarbeit an der TU Ilmenau implementiert [35].

4.12.4.1 Untertägige Personaleinsatzplanung in der Logistik

Für die untertägige Personaleinsatzplanung in der Logistik muss ES stark modifiziert werden. Die Modifikation erfolgt für beide Problemvarianten mit und ohne Stammarbeitsplatzregelung einheitlich. Lediglich die optionale Reparatur muss zusätzlich an die Erfordernisse von Stammarbeitsplätzen angepasst werden. Nachfolgend ist der Ablauf in Algorithmus 10 dargestellt.

Algorithmus 10 ES mit Reparatur

```
1: Initialisiere die Population mit  $\mu$  Individuen
2: Repariere die Population
3: Berechne die Fitness der  $\mu$  Individuen
4: repeat
5:   Rekombination zur Erzeugung von  $\lambda$  Nachkommen
6:   Mutation der  $\lambda$  Nachkommen
7:   Repariere die  $\lambda$  Nachkommen
8:   if ES-Strategie =  $(\mu+\lambda)$  then
9:     Repariere die  $\mu$  Eltern
10:  end if
11:  Berechne die Fitness aller reparierter Individuen
12:  Selektion, um die neue Generation mit  $\mu$  Individuen zu erzeugen
    ( $(\mu+\lambda)$  oder  $(\mu, \lambda)$ )
13: until Abbruchkriterium tritt ein
```

Initialisierung Bei der Initialisierung erhält jedes Individuum eine eigene Startposition. Hierzu werden die zwei Initialisierungsstrategien (I1 und I2) getestet, die bereits beim HC in Kapitel 4.8.3.1 Anwendung finden. Variante 3 (I3), die in Kapitel 4.10.3.1 bei TS vorgestellt wurde, findet von vorne herein keine Anwendung. Denn der Start aller Individuen aus einem Punkt heraus behindert bei ES die Suche.

In Tests hat sich gezeigt, dass I1 sehr schlecht abschneidet. Der Plan enthält deutlich zu viele Arbeitsplatzwechsel, mit denen ES Probleme hat. Experimente mit I1 werden daher nicht durchgeführt. I2 hingegen liefert gute Ergebnisse und wird daher zur Initialisierung gewählt.

Rekombination Beim Logistikproblem wurden insgesamt elf verschiedenen Rekombinationsvarianten getestet, auf die im Einzelnen nicht eingegangen werden soll. Am besten schnitt die Variante fünf ab (siehe Tabelle 4.8). Für jedes Kind wird dort zufällig eine individuelle Grenze in der zeitlichen Dimension (Spalte) ermittelt, bis der die Werte aus Elter eins übernommen werden. Der Rest des Kindes wird mit den Werten

von Elter zwei befüllt. Die ermittelte Grenze ist für alle Mitarbeiter (Zeilen) gleich. Folgender Zusammenhang dürfte die Überlegenheit dieser Rekombinationsvariante im Vergleich zu allen anderen erklären. Strafpunkte werden für Qualifikationsverletzungen, Anzahl Arbeitsplatzwechsel und vor allem für Über-/Unterdeckung der Arbeitsplatzbesetzungen vergeben. Die Zuweisung der Werte aus Elter eins bis zu einer einheitlichen Zeitgrenze (Spalte) für alle Mitarbeiter (Zeilen) ist wahrscheinlich effektiver als andere Rekombinationsvarianten, da eine gewisse spaltenweise Ordnung hinsichtlich der Über- und Unterdeckung bei der Rekombination nicht verworfen wird.

Tabelle 4.8: Rekombination nach Variante 5

Vektorelement	1	2	3	4	5	6
Elter 1	0	2	2	3	3	3
Elter 2	0	7	7	1	1	1
Nachkomme	0	2	7	1	1	1

Klassische Gauß-Mutation Die klassische Gauß-Mutation erfolgt für das Logistikproblem nach dem in Algorithmus 11 dargestellten Ablauf. Sie wird nachfolgend als TypG bezeichnet. Für jeden Mitarbeiter existiert eine Liste mit den Arbeitsplätzen für die er qualifiziert ist. Stehen nun der zufällig ermittelte Zeitintervall und der Mitarbeiter für die Mutation fest, so wird dort ein Arbeitsplatz aus der entsprechenden Qualifikationsliste des Mitarbeiters eingesetzt.

Algorithmus 11 ES mit klassischer Gauß-Mutation bei der untertägige Personaleinsatzplanung in der Logistik

```

1:  $\tau = 0,5$ 
2:  $\hat{\sigma} = \sigma * \exp(\tau * N(0,1))$ 
3: Drift =  $N(0, \hat{\sigma})$ 
4: Count = runde |Drift|
5: if Count < 1 then
6:   Count = 1
7: end if
8: for i = 1 bis Count do
9:   Wähle einen Mitarbeiter  $e$  mit  $p = 1/|E|$ 
10:  Wähle Zeitintervall  $t$ , in dem Mitarbeiter  $e$  anwesend ist mit  $p =$ 
    1/Anzahl Zeitintervalle, in denen  $e$  anwesend ist
11:  Wähle Arbeitsplatz  $w$  aus der Liste der Arbeitsplätze für die
    Mitarbeiter  $e$  qualifiziert ist mit  $p = 1/\text{Anzahl Elemente der Liste}$ 
12:  Neuer Arbeitsplatz von  $e$  in  $t = w$ 
13: end for

```

Mutation mit dem Konzept der maximalen Entropie Bei der Mutation mit maximaler Entropie – nachfolgend als TypE bezeichnet – bewegt sich der Wert eines Matrixelements durch die Addition von Z in eine bestimmte Richtung. Da dieses Z in der vorliegenden Problemstellung meist null ist und $|Z|$ fast nie Werte größer als zwei annimmt, ist die Mutation nicht immer zielführend. Bestimmte Regionen werden nicht erreicht, da auf dem Weg zum Optimum ungünstige Arbeitsplatzzuweisungen auftreten und die schlechtesten Individuen nicht selektiert werden. Das Konzept von Rudolph ist demnach nicht für kombinatorische Problemstellungen des hochflexiblen WFM geeignet und muss modifiziert werden.

Um in der Mutation die Chance zu haben, ausgehend von einem beliebigen Wert jeden anderen zulässigen Wert zu erreichen, wurde Z mit der Anzahl an Arbeitsplätzen minus eins ($8 = 9 - 1$) sowie einer gleichverteilten Zufallsvariable v multipliziert, mit $v \sim V[0, 1]$. Z ergibt sich nun wie folgt.

$$Z = \text{round}(8v(G_1 - G_2)) \quad (4.16)$$

Neben dieser Anpassung wurde auch τ verändert und auf 0,05 gesetzt. τ^2 ergibt sich nun aus $17,07/n$. In seiner Arbeit hat Rudolph für $\tau^2 = 1/n$ gewählt. Mit dieser Varianz hat er fünf Problemstellungen untersucht, wobei n jeweils recht kleine Werte zwischen 5 und 30 annimmt. Für eine deutlich größere Anzahl an Dimensionen ist dieses Vorgehen scheinbar ungeeignet. Im Vergleich zum Vorgehen von Rudolph müssen noch weitere Anpassungen beim Logistikproblem vorgenommen werden, da der Lösungsraum nicht unbeschränkt ist und Verfügbarkeits- und Qualifikationsrestriktionen beachtet werden müssen. Der finale Ablauf hat die in Algorithmus 12 dargestellte Gestalt. Die Mutation mit maximaler Entropie von Rudolph muss also folgende Modifikationen erhalten, um beim Logistikproblem anwendbar zu sein.

- Begrenzung der Dimensionsgrenzen wegen begrenzter Anzahl an Arbeitsplätzen.
- Beschränkung der Mutation auf Matrixelemente mit anwesenden Mitarbeitern.
- Zuweisung von Arbeitsplätzen unter Berücksichtigung der Qualifikationen.
- Größere Streuung von Z durch $Z = \text{round}(8v(G_1 - G_2))$.
- $\tau^2 = 17,07/n$.
- Rekombination Variante 5.

Algorithmus 12 ES mit dem Konzept der maximalen Entropie bei der untertägige Personaleinsatzplanung in der Logistik

```

1:  $\tau = \sqrt{17,07/n}$ 
2:  $\hat{\sigma} = \sigma * \exp(\tau * N(0,1))$ 
3: if  $\hat{\sigma} < 1$  then
4:    $\hat{\sigma} = 1$ 
5: end if
6: Berechne  $p$ 
7: for  $e = 1$  bis Anzahl Mitarbeiter do
8:   for  $t = 1$  bis Anzahl Zeitintervalle do
9:     Berechne  $Z$ 
10:    while aktueller Arbeitsplatz +  $Z$  ist unzulässig do
11:      Berechne  $Z$ 
12:    end while
13:    neuer Arbeitsplatz = aktueller Arbeitsplatz +  $Z$ 
14:   end for
15: end for

```

Reparatur Mit einer Reparatur der Individuen soll die Lösung verbessert werden. Sie ist bei ES nicht zwingend erforderlich und kann grundsätzlich auch weggelassen werden. Die Reparatur selbst verursacht einen gewissen Rechenaufwand. In $(\mu + \lambda)$ -Strategien werden nicht nur die Nachkommen, sondern auch die Eltern repariert. Eine mehrfache Reparatur eines Individuums kann zu weiteren Verbesserungen führen. Auf eine mehrfache Reparatur aller Individuen wird aufgrund der dann deutlich erhöhten Rechenzeit verzichtet. Die Reparatur korrigiert folgende Fehler in der hier angegebenen Reihenfolge:

1. Qualifikationsverletzungen: Mitarbeiter, die für den aktuell zugewiesenen Arbeitsplatz nicht qualifiziert sind, werden auf einen Arbeitsplatz umbesetzt, für den sie die Qualifikation besitzen (ungeachtet möglicher Über-/Unterdeckungen im Personaleinsatz).
2. Bedarf = 0: Mitarbeiter, die einem Arbeitsplatz zugewiesen sind, bei dem in einem Zeitintervall ein Bedarf von null besteht, werden nach Möglichkeit umbesetzt. Dabei werden die Qualifikationen der Mitarbeiter berücksichtigt. So kann es vorkommen, dass eine Umbesetzung nicht möglich ist, da zu einem Zeitpunkt alle Arbeitsplätze, für die der Mitarbeiter qualifiziert wäre, einen Bedarf von null haben.
3. Über-/Unterdeckung: Sind in einem Zeitintervall zu wenig Mitarbeiter eingeplant, so wird versucht, Mitarbeiter von Arbeitsplätzen umzusetzen, auf denen eine Überdeckung zu diesem Zeitpunkt herrscht. Somit wird gleichzeitig auch die Überdeckung minimiert. Auch hier werden Qualifikationen berücksichtigt, sodass die

Über-/Unterdeckung nicht immer vermieden werden kann. Auch gibt es Phasen, in denen der Personalbedarf über bzw. unter dem Personalbestand liegt, sodass es zwangsläufig zu Phasen der Über-/Unterdeckung kommt.

In Algorithmus 13 ist der Ablauf der Reparatur eines Individuums dargestellt, wobei $\text{Value}[t,e]$ für den Arbeitsplatz des Mitarbeiters e in Zeitintervall t steht.

Algorithmus 13 Reparatur bei der untertägige Personaleinsatzplanung in der Logistik ohne Stammarbeitsplatzregelung

```
1: for  $e = 1$  bis Anzahl Mitarbeiter do
2:   Erstelle für jeden Mitarbeiter  $e$  die Liste  $\text{WpList}_e$  mit den
   Arbeitsplätzen, für die er qualifiziert ist
3:   for  $t = 1$  bis Anzahl Zeitintervalle do
4:     // Entferne Qualifikationsfehler
5:     if Mitarbeiter  $e$  ist nicht für  $\text{Value}[t,e]$  qualifiziert then
6:        $\text{Value}[t,e] = \text{random Arbeitsplatz aus } \text{WpList}_e$ 
7:     end if
8:     // Entferne Fehler bei einem Bedarf von 0
9:     if Bedarf auf Arbeitsplatz  $\text{Value}[t,e] = 0$  then
10:       $\text{Value}[t,e] = \text{random Arbeitsplatz aus } \text{WpList}_e$ , bei dem der
      Bedarf  $> 0$  ist (wenn so ein Arbeitsplatz existiert)
11:    end if
12:  end for
13: end for
14: // Reduziere Über-/Unterdeckungen
15: for  $t = 1$  bis Anzahl Zeitintervalle do
16:   for  $w = 1$  bis Anzahl Arbeitsplätze do
17:     if Arbeitsplatz  $z$  in Zeitintervall  $t$  unterdeckt ist then
18:       for  $e = 1$  bis Anzahl Mitarbeiter do
19:         if  $\text{Value}[t,e]$  ist überdeckt &&  $e$  ist für  $w$  qualifiziert then
20:            $\text{Value}[t,e] = w$ 
21:         end if
22:       end for
23:     end if
24:   end for
25: end for
```

Für die Problemvariante mit Stammarbeitsplatzregelung wird die Reparatur speziell angepasst. Zunächst werden Mitarbeiter bei Qualifikationsverletzungen auf ihren Stammarbeitsplatz gesetzt. Da bei dieser Problemvariante keine erhöhten Fehlerpunkte bei einem Bedarf von null anfallen, entfällt der zuvor dargestellte Reparaturschritt zwei. Zusätzlich wird jedoch an das Ende der bisherigen Reparatur ein zusätzlicher Schritt eingeführt, bei dem Mitarbeiter auf ihren Stammarbeitsplatz gesetzt werden, wenn dies nicht zu

zusätzlichen Fehlern bei Qualifikationen oder Besetzungsabweichungen führt. Auch bei Problemstellungen mit Stammarbeitsplatz ist die Reparatur in ES optional und muss nicht zwingend eingesetzt werden.

4.12.4.2 Automatische Arbeitszeitmodellerstellung im Handel

Der Ablauf von ES für die automatische Arbeitszeitmodellerstellung im Handel ist fast analog zu ES beim Logistikproblem. Es gibt lediglich die nachfolgend dargestellten Modifikationen. Alle anderen Elemente und Parameter bleiben bestehen.

Initialisierung Im Gegensatz zum Logistikproblem ist hier kein Monatsplan mit den täglichen Arbeitszeiten gegeben. Denn dieser soll erst in der Planung erstellt werden. Es ist jedoch bekannt, zu welchen Uhrzeiten das Geschäft geöffnet ist, an welchen Tagen es aufgrund von Feiertagen geschlossen ist und wann Mitarbeiter Urlaub haben. Dieses problemspezifische Wissen soll für die Initialisierung der Population genutzt werden. Verschiedene Initialisierungsvarianten wurden erprobt, wobei Variante 1 (I1) des Logistikproblems am besten abschnitt (siehe Kapitel 4.12.4.1). I2 eignet sich weniger gut, da Mitarbeiter bei der Initialisierung ganztägig auf Arbeitsplätze geplant worden wären, was immer Überschreitungen der maximalen Arbeitszeit nach sich zieht. Zudem haben Mitarbeitern mit ganztägiger Dummy-Arbeitsplatzzuweisung später kaum die Möglichkeit, eingeplant zu werden.

Die Initialisierung sieht demnach so aus, dass Mitarbeitern, die in einem Zeitintervall verfügbar sind, jeweils ein zufälliger Arbeitsplatz (inkl. Dummy-Arbeitsplatz) in jedem dieser Zeitintervalle zugewiesen wird. Hierdurch können gelegentlich Fehler bei den Arbeitszeitmodellen auftreten, die mit der Reparatur sofort behoben werden.

Reparatur Während beim Logistikproblem die Reparatur nicht zwingend erforderlich ist, ist sie beim Handelsproblem unerlässlich, um akzeptable Ergebnisse zu erzielen. In Vortests wurde festgestellt, dass ohne Reparatur keine praktikablen Ergebnisse geliefert werden können. Es gibt in den finalen Einsatzplänen zu viele Verstöße gegen die Nebenbedingungen, die beim Handelsproblem – im Vergleich zum Logistikproblem – zahlreicher und vor allem erheblich komplexer sind. Daher wurde eine Reparatur implementiert, die Verstöße möglichst ausbessert. Die Reparatur korrigiert folgende Fehler in der hier angegebenen Reihenfolge:

1. Bedarf = 0: Mitarbeiter, die einem Arbeitsplatz zugewiesen sind, bei dem in einem Zeitintervall ein Bedarf von null besteht, werden nach Möglichkeit umbesetzt.

2. Über-/Unterdeckung: Sind in einem Zeitintervall zu wenig Mitarbeiter eingeplant, so wird versucht, Mitarbeiter von Arbeitsplätzen umzubesetzen, auf denen eine Überdeckung zu diesem Zeitpunkt herrscht. Somit wird gleichzeitig auch die Überdeckung minimiert. Über- und Unterdeckung können nicht immer vermieden werden, da es Phasen gibt, in denen der Personalbedarf über bzw. unter dem Personalbestand liegt, sodass es zwangsläufig zu Besetzungsfehlern kommt.
3. Mehr als ein Arbeitszeitmodell pro Tag: Hat ein Mitarbeiter mehr als ein Arbeitszeitmodell pro Tag (z.B. 1. Beginn 10 Uhr, 1. Ende 14 Uhr, 2. Beginn 16 Uhr, 2. Ende 20 Uhr), so wird versucht, eines der beiden Arbeitszeitmodelle auf einen anderen Mitarbeiter zu übertragen, der im entsprechenden Zeitraum noch nicht eingeplant ist. Man könnte die Situation mehrere Arbeitszeitmodelle pro Tag auch so beschreiben, dass ein Arbeitszeitmodell Lücken hat, was zu äußerst hohen Fehlerpunkten führt.
4. Minimale Länge der Arbeitszeitmodelle: Unterschreitet ein Arbeitszeitmodell die minimale Länge, so wird versucht, das Arbeitszeitmodell eines anderen Mitarbeiters zu kürzen und an zu kurzen Arbeitszeitmodell anzudocken. Gelingt dies nicht, so wird versucht, das zu kurze Arbeitszeitmodell an das Arbeitszeitmodell eines anderen Mitarbeiters anzudocken.
5. Maximale Länge der Arbeitszeitmodelle: Wird die maximal zulässige Länge eines Arbeitszeitmodells überschritten, so wird es in zwei zulässige Teile zerlegt. Anschließend wird versucht, eines der Teile auf einen anderen Mitarbeiter zu übertragen bzw. an ein bestehendes Arbeitszeitmodell eines anderen Mitarbeiters anzudocken.
6. Arbeitsplatzwechsel: Arbeitsplatzwechsel dienen der bedarfsgerechten Personaleinsatzplanung. Unnötige Wechsel sind jedoch zu vermeiden. Pro Zeitintervall wird für jeden Mitarbeiter geprüft, ob das Vertauschen seines Arbeitsplatzes mit dem eines anderen Mitarbeiters zu einer Reduzierung der Arbeitsplatzwechsel führt. Ist dies möglich, so erfolgt der Tausch.
7. Maximale wöchentliche Arbeitszeit: Jeder Mitarbeiter hat laut Arbeitsvertrag eine maximale wöchentliche Arbeitszeit, die nicht überschritten werden sollte. Gibt es Überschreitungen, wird die Tagesbelegung eines Mitarbeiters mit zu vielen Stunden mit der Tagesbelegung eines Mitarbeiters mit zu wenig Stunden vertauscht. Gewöhnlich hat der Mitarbeiter mit zu vielen Stunden dann frei.
8. Finale Reparatur der Arbeitszeitmodelle: Nachdem alle Reparaturen durchlaufen wurden, kann es noch immer vorkommen, dass die minimale oder maximale Länge der Arbeitszeitmodelle nicht eingehalten wurde. Oder es gibt noch Lücken in den Arbeitszeitmodellen. Die Reparatur war zuvor also nicht in der Lage, Teile von Arbeitszeitmodellen bei anderen Mitarbeitern anzudocken oder zu verschie-

ben. Die Einhaltung der Regelungen für die Arbeitszeitmodelle hat jedoch oberste Priorität und ist daher mit den höchsten Fehlerpunkten für Verstöße belegt. Daher werden an dieser Stelle die fehlerhaften Arbeitszeitmodelle ohne Rücksicht auf Über-/Unterdeckungen sowie wöchentliche Arbeitszeiten repariert. Zu kurze Arbeitszeitmodelle werden verlängert bzw. komplett gelöscht – je nachdem, was weniger Fehlerpunkte verursacht. Zu lange Arbeitszeitmodelle werden verkürzt. Lücken in den Arbeitszeitmodellen werden aufgefüllt oder Enden werden gelöscht – je nachdem, was weniger Fehlerpunkte verursacht.

Die dargestellte Reparatur gilt für die Probleme Handel_2AP und Handel_2AP_Jan. Beim Handel_1AP und Handel_1AP_Jan entfällt Schritt 6, da es keine Arbeitsplatzwechsel geben kann. Zudem erfolgen in Schritt eins und zwei nur noch Umbesetzungen zwischen Arbeitsplatz eins und dem Dummy-Arbeitsplatz, der kenntlich macht, dass ein Mitarbeiter zwar verfügbar ist aber nicht arbeitet.

4.12.4.3 Kombination von Touren- und Personaleinsatzplanung

Bereits beim Logistik- und Handelsproblem wurden ES angepasst. Das British Telecom-Problem erfordert aufgrund der veränderten Abbildung der Problemstellung und des anders strukturierten Lösungsraums weitere Modifikationen. Die Lösungsräume sind beim Logistik- und Handelsproblem eher durch flache Bereiche geprägt, bei denen vereinzelt Vertiefungen und Erhebungen auftreten. Beim British Telecom-Problem existieren diese flachen Bereiche nicht sehr zahlreich. Der Lösungsraum ist deutlich zerklüfteter, da die Techniker an einem Auftrag verschieden schnell arbeiten und auch die Anfahrt- und Abfahrtszeiten verschieden lang sein können. Am British Telecom-Problem führen Lösungsverfahren womöglich weniger Seitwärtsschritte aus, sind weniger in Bewegung und konvergieren schneller in einem lokalen Optimum. Der grundlegende Ablauf erfolgt analog zum Ablauf beim Logistikproblem und ist in Algorithmus 10 dargestellt.

Initialisierung Die Initialisierung der Population soll nicht völlig zufällig erfolgen, sondern bereits problemspezifische Aspekte berücksichtigen. Daher wird KV 3 aus Kapitel 4.6.3.2 für die Initialisierung der Individuen gewählt. Für jedes Individuum erzeugt KV 3 eine eigene Startlösung.

Rekombination Bei der Rekombination zweier Eltern zur Erzeugung eines Nachkommens wird wie folgt vorgegangen:

1. Ermittle zufällig einen Techniker (bei jeder Rekombination ein neuer Techniker).

2. Kopiere die komplette Matrix von Elter eins, beginnend von Techniker eins bis zu dem zufällig ermittelten Techniker, auf den Nachkommen.
3. Kopiere die komplette Matrix von Elter zwei, beginnend mit dem zufällig ermittelten Techniker bis zum letzten Techniker, auf den Nachkommen. Bereits vorhandene Aufträge von Elter A werden nicht kopiert.
4. Durchlaufe die Liste der nicht zugewiesenen Aufträge und hänge jeden Auftrag an die Permutation eines qualifizierten Mitarbeiters.

Für die Mutation der Nachkommen werden zwei Varianten erprobt, die nachfolgend erläutert werden.

Klassische Gauß-Mutation Die Mutation mit der Gauß-Verteilung (TypG) ist nachfolgend in Algorithmus 14 dargestellt. Die Matrix wird nicht systematisch durchlaufen. Für jedes Count wird eine von zwei Aktionen zur Veränderung des Individuums zufällig ausgewählt, wobei Count im Minimum 1 und im Maximum 250 (Anzahl Aufträge) betragen darf.

Algorithmus 14 ES mit klassischer Gauß-Mutation bei der Kombination von Touren- und Personaleinsatzplanung

```
1:  $\tau = 0,5$ 
2:  $\sigma = \sigma * \exp(\tau * N(0,1))$ 
3: Count = runde  $|N(0,\sigma)|$ 
4: if Count < 1 then
5:   Count = 1
6: end if
7: if Count > 250 then
8:   Count = 250
9: end if
10: for i = 1 bis Count do
11:   Berechne eine Zufallszahl  $v = N(1,100)$ 
12:   if  $v \leq 20$  then
13:     Verschiebe einen Auftrag
14:   else
15:     Vertausche zwei Aufträge
16:   end if
17: end for
```

In Zeile 12 wird ein Auftrag verschoben. Dies läuft wie folgt ab: Es wird ein Techniker zufällig ausgewählt. Für dessen letzten Auftrag in der Permutation wird ein qualifizierter

Mitarbeiter gesucht. Ist dieser gefunden, so wird ihm der Auftrag als letzter Auftrag seiner Permutation zugewiesen. Wird kein qualifizierter Techniker gefunden, passiert nichts. Das Vertauschen in Zeile 15 läuft wie folgt ab: Es werden zwei zufällige Aufträge ausgewählt und die zugehörigen Techniker gesucht. Die Techniker tauschen anschließend die Auftragszuweisungen aus, wobei der neue Auftrag jeweils an die gleiche Stelle des alten Auftrags geschrieben wird. Qualifikationen werden bei dieser Aktion berücksichtigt.

Mutation mit dem Konzept der maximalen Entropie Bei der Mutation mit maximaler Entropie (TypeE) wird die Matrix systematisch durchlaufen. In jedem Feld, in dem sich eine Auftragszuweisung befindet, wird Z berechnet. Sobald Z ungleich null ist, wird eine von zwei möglichen Aktionen zur Veränderung des Individuums zufällig ausgewählt. Im Gegensatz zur Mutation mit Gauß-Verteilung betrifft die Änderung das konkrete Matrixelement, bei dem Z berechnet wird. Die Mutation läuft nach dem in Algorithmus 15 dargestellten Ablauf ab.

Reparatur Qualifikationen sind beim British Telecom-Problem besonders kritisch und deren Einhaltung eine harte Nebenbedingung. Eine Lösung ist also nur ohne Qualifikationsfehler gültig. Um etwaig auftretende Qualifikationsfehler zu vermeiden, was gelegentlich auftreten kann, werden diese repariert. In der Reparatur werden Qualifikationsverletzungen gesucht. Ein fehlerhaft zugewiesener Auftrag wird dann einem zufällig ermittelten, qualifizierten Mitarbeiter zugewiesen. Bei der Einordnung in dessen Permutation wird auf ein mögliches Zeitfenster des Auftrags geachtet. Eine ggf. entstehende Lücke in der Auftragsabfolge beim nicht qualifizierten Techniker wird geschlossen, sodass die Lösung gültig ist.

Die Reparatur bezieht sich also nur auf Qualifikationen. Derart umfangreiche Reparaturen wie beim Logistik- oder Handelsproblem sind nicht erforderlich. Da Qualifikationen harte Nebenbedingungen sind, kann ohne den Einsatz der Reparatur nicht garantiert werden, dass keine Qualifikationsfehler in der Lösung enthalten sind. Sie ist für den erfolgreichen Einsatz von ES demnach zwingend erforderlich.

4.13 Genetische Algorithmen

Wie bereits in Kapitel 4.12.1 gezeigt wurde, gehören Genetische Algorithmen (GA) ebenfalls in die Gruppe der EA. Für grundlegende Ausführungen zur Evolution, zu EA oder der Abgrenzung von ES und GA sei auf Kapitel 4.12.1 verwiesen. Nachfolgend wird zunächst der grundlegende Aufbau von GA beschrieben. In Kapitel 4.13.2 wird auf Anwendungen von GA in der Literatur eingegangen. Dies erfolgt nicht in dem Detaillierungsgrad wie bei ES, da GA in dieser Arbeit nicht selbst implementiert wurde. Muller

Algorithmus 15 ES mit dem Konzept der maximalen Entropie bei der Kombination von Touren- und Personaleinsatzplanung

```
1:  $\tau = 0,05$ 
2:  $\hat{\sigma} = \sigma * \exp(\tau * N(0,1))$ 
3: if  $\hat{\sigma} < 1$  then
4:    $\hat{\sigma} = 1$ 
5: end if
6: Berechne  $p$ 
7:  $p_{Log} = \text{Log}(1 - p)$ 
8: for  $j = 1$  bis Anzahl Aufträge do
9:   for  $e = 1$  bis Anzahl Techniker do
10:    Berechne  $Z$ 
11:    if Auftrag  $j - Z$  bei Techniker  $e > 250$  then
12:       $Z = 0$ 
13:    end if
14:    if Auftrag  $j - Z$  bei Techniker  $e < 1$  then
15:       $Z = 0$ 
16:    end if
17:    if  $Z \neq 0$  then
18:      Berechne eine Zufallszahl  $v = N(1,100)$ 
19:      if  $v \leq 20$  then
20:        Verschiebe den Auftrag  $j$  bei Techniker  $e$  an das Ende der
          Permutation eines qualifizierten Technikers
21:      else
22:        Vertausche den Auftrag  $j$  bei Techniker  $e$  mit einem zufälligen
          Auftrag eines für Auftrag  $j$  qualifizierten Technikers
23:      end if
24:    end if
25:   end for
26: end for
```

u.a. [277] haben diese Metaheuristik auf das Problem der British Telecom angewendet. In Kapitel 4.13.3.1 wird auf ihre Implementierung eingegangen. An den anderen Problemstellungen dieser Arbeit wird GA nicht untersucht.

4.13.1 Grundlegender Ablauf

GA gehen auf Holland zurück [186], [187], [188]. Ebenso wie bei ES, stellt jedes Individuum eine Lösung dar, die jedoch meist binär codiert ist. Da die Suchoperatoren auf dem Genotyp arbeiten, können u.U. ungültige Lösungen entstehen, die aufwändig repariert werden müssen. Der grundlegende Ablauf eines GA ist in Algorithmus 16 dargestellt.

Algorithmus 16 GA

-
- 1: Initialisiere die Population
 - 2: Berechne die Fitness der Individuen
 - 3: **repeat**
 - 4: Selektiere Elternpaare
 - 5: Rekombination zur Erzeugung von Nachkommen
 - 6: Mutation der Nachkommen
 - 7: Berechne die Fitness der Nachkommen
 - 8: Ersetze die Individuen der aktuellen Generation durch die Nachkommen
 - 9: **until** Abbruchkriterium tritt ein
-

Selektion Gewöhnlich wird aus zwei Eltern ein Nachkommen erzeugt. Grundsätzlich ist es aber auch möglich, mehr als zwei Eltern zu rekombinieren oder ein Nachkommen durch Klonen eines Elter zu generieren.

Die Auswahl eines Elternpaares kann z.B. mit dem Roulette-Wheel-Schema erfolgen. Langfristig wird mit diesem Ansatz der Gedanke verfolgt, dass sich die Güte der Population von Generation zu Generation steigert. Denn sehr gute Individuen haben beim Roulette-Wheel-Schema eine größere Wahrscheinlichkeit, ihre Erbinformationen weiterzugeben als schlechte Individuen. Eltern werden mit der Wahrscheinlichkeit ausgewählt, die proportional zu dessen Fitnesswert ist. So haben auch schlechte Individuen eine Chance – wenn auch eine kleine – ihre Erbinformationen weiterzugeben. [255, S. 7, 9]

Rekombination Der wichtigste Suchoperator bei GA ist die Rekombination. Sie dient dazu, die Suche zu steuern. Es werden immer so viele Nachkommen wie Eltern erzeugt, um die Populationsgröße konstant zu halten. Es existiert eine Fülle an Rekombinationsvarianten. Beispielhaft ist in Abbildung 4.19 die Rekombination mit dem 1-Punkt-Crossover dargestellt. Dort liegt die Bruchstelle bei $q = 4$. [255, S. 10]

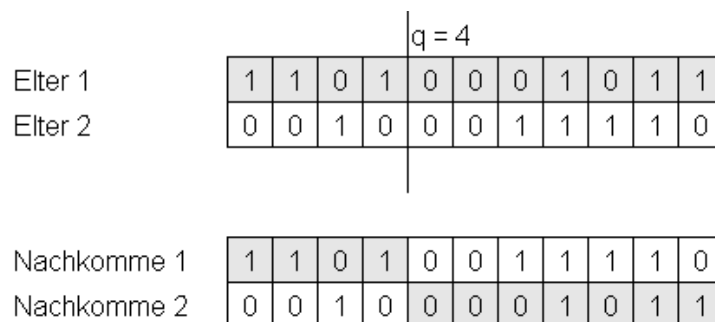


Abbildung 4.19: 1-Punkt-Crossover bei zwei Individuen

Die Tatsache, dass hier beispielhaft der 1-Punkt-Crossover aufgeführt ist, soll kein Indikator für die generelle Eignung dieser Rekombinationsvariante sein.

Mutation Bei der Mutation wird Bit für Bit durchlaufen. Dabei existiert eine sehr kleine Wahrscheinlichkeit, dass das Bit invertiert wird. Die Mutation hat eine nachrangige Bedeutung bei GA. Dies legt bereits die kleine Eintrittswahrscheinlichkeit für die Bit-Invertierung nahe. Die Mutation verfügt im Gegensatz zu ES meist über keine selbstadaptierenden Strategieparameter. Sie dient dazu, eine vorzeitige Konvergenz zu verhindern, da sie garantiert, dass kein Allel (Wert) endgültig verloren geht. [94, S. 22–23]

GA haben in der Vergangenheit sehr viele Erweiterungen und Modifikationen erfahren. In dieser Arbeit wurde kein eigener GA implementiert. Daher wird auf weitere grundlegende Ausführungen zu GA verzichtet. Im folgenden Kapitel wird zunächst auf einige Anwendungen von GA in der Literatur eingegangen. Anschließend wird die Anpassung von GA für das British Telecom-Problem kurz dargestellt.

4.13.2 Anwendungen in der Literatur

Wie auch ES werden GA nicht sehr häufig für kombinatorische Problemstellungen angewendet. Meist kommen sie bei besonders schwer lösbaren Problemstellungen zum Einsatz. Z.B. 1993 bei Easton und Mansour [110], 1995 bei Wren und Wren [424] sowie 2000 bei Aickelin und Dowsland [6].

1993 erproben Muller u.a. [277] einen verteilt arbeitenden GA am Problem der British Telecom. In Kapitel 4.13.3 wird gesondert auf diesen Ansatz eingegangen.

Awad und Chinnek [17] verwenden 1998 einen GA für die auftragsbezogene Planung im Rahmen der Prüfungsorganisation. Es werden 120 saisonal beschäftigte Prüfungsaufseher verplant, wobei die Anzahl der zu belegenden Zeitintervalle recht begrenzt ist. Über einen Zeitraum von 12 bis 15 Tagen werden pro Tag drei Zeitintervalle mit einer maximalen Länge von drei Stunden angeboten, in denen Prüfungen stattfinden können. Die Startpopulation des GA wird mit einer problemspezifischen Heuristik ermittelt. Im Echtbetrieb wurde die Berechnung in 30 Minuten vollzogen. Zuvor benötigte man per Hand 35 Stunden. Der zuständige Planer äußerte sich sehr positiv über das Planungstool, da er auch mit den Ergebnissen und den Auswertungsmöglichkeiten sehr zufrieden war.

Büttcher nutzt 2002 [55] GA, um Probleme der Dienstplanung und der Stundenplanerstellung zu lösen. Dabei zeigt sich, dass auch nach mehrstündigen Berechnungen zahlrei-

che harte Nebenbedingungen verletzt werden. Erst durch den zusätzlichen Einsatz von Reparaturmechanismen und Lokaler Suche kann dieses Problem behoben werden.

4.13.3 Anpassung an hochflexibles Workforce Management

GA wird in dieser Arbeit lediglich am British Telecom-Problem angewendet. Dabei werden die Ergebnisse von Muller u.a. [277] genutzt. Eine Implementierung beim Logistik- und Handelsproblem findet nicht statt.

4.13.3.1 Kombination von Touren- und Personaleinsatzplanung

Muller u.a. stellen 1993 in ihrer Arbeit [277] verteilt arbeitende GA für das Problem BT_Mod-250-118 vor. Bei den GA orientieren sie sich am Vorgehen von Clearwater, Huberman und Hogg [70].

Es existieren zeitgleich mehrere GA. Jeder GA kann eine eigene Art haben, ein Chromosom in einen Einsatzplan zu transformieren, um anschließend die Fitness zu berechnen. Zum einen werden von Muller u.a. ein servicecenterorientierter Ansatz nach Fox und Smith [123] und ein ressourcenbasierter Ansatz nach Smith u.a. [350] verwendet. Neben diesen Transformationsansätzen können sich die GA auch bzgl. ihres Verhaltens unterscheiden. Aufgrund verschiedener Regeln gibt es GA, die sehr schnell Lösungen liefern. Andere GA benötigen deutlich mehr Zeit, können dafür aber bessere Ergebnisse aufweisen. Damit möchte man erreichen, dass es zu jeder Zeit ein möglichst akzeptables Ergebnis gibt. Auch wenn der Anwender vorzeitig abbricht. Optional verfügt jeder GA auch über eine Reparatur.

Es gibt einen gemeinsamen Speicher, in dem das jeweils beste Chromosom jedes GA hinterlegt ist. Muller u.a. wollen so eine Art Kommunikation zwischen den GA herstellen, da jeder GA auf diesen Pool der „Super-Individuen“ zugreifen und sie in seine lokale Population nehmen kann. So können GA auf Chromosomen zugreifen, die aus verschiedenen Lösungsansätzen heraus entstanden sind. Weitere Details haben Muller u.a. nicht veröffentlicht.

4.14 Particle Swarm Optimization

Eine weitere populationsbasierte Metaheuristik ist Particle Swarm Optimization (PSO). Nachfolgend werden zunächst grundlegende Zusammenhänge erläutert, worauf hin der

Ablauf von PSO in Kapitel 4.14.2 dargestellt wird. PSO wird in dieser Arbeit sehr umfangreich an allen drei Problemstellungen untersucht. Die jeweiligen Ausführungen hierzu sind in Kapitel 4.14.3 zu finden. Ein eigenes Kapitel zur Anwendung von PSO in der Literatur gibt es nicht, da derartige Ausführungen für PSO passender in den einzelnen Unterkapiteln angesiedelt sind.

Wenn nachfolgend PSO mit einem zusätzlichen Wert in Klammern dargestellt ist, z.B. PSO (20), so handelt es sich bei der Zahl um die Schwarmgröße. In diesem Beispiel sind es 20 Partikel.

4.14.1 Grundlegende Zusammenhänge

Schwarmintelligenz Schwarmintelligenz ist in der Natur in vielfältigen Ausprägungen zu finden. Blattschneiderameisen überwinden große Entfernungen auf markierten Wegen, und Termiten errichten in ihren Nestern ein Kühlsystem [39, S. 1–5]. In Schwärmen existiert kein Individuum, das die Aufsicht übernimmt und Anweisungen erteilt. Dennoch sind Schwärme in der Lage, komplexe Aufgaben zu lösen, die ein einzelnes Individuum nicht bewältigen könnte. Der Grund für diese Fähigkeit, die Ausprägung der Schwarmintelligenz ist, liegt in der Selbstorganisation des Schwarms und in der Arbeitsteilung [396, S. 46]. Das Konzept der Schwarmintelligenz wurde auch bei Roboterschwärmen verwendet, indem sich die Agenten durch Interaktion mit ihrem nächsten Nachbarn selbst organisieren [29]. Der Begriff der Schwarmintelligenz umfasst alle Tätigkeiten, die auf Basis des Kollektivverhaltens von Insektenkolonien und anderen Gemeinschaften arbeiten [39, S. 7]. Schwarmintelligenz liegt vor, wenn die fünf Eigenschaften von Millonas durch den Schwarm erfüllt sind [267, S. 418–419].

- **Adaptability:** Wenn es sich lohnt, dann soll das Verhalten geändert werden.
- **Diverse Response:** Viele Möglichkeiten nutzen und nicht auf nur eine setzen.
- **Proximity:** Durchführung einfacher zeitlicher und räumlicher Berechnungen.
- **Quality:** Reaktion auf qualitative Umweltfaktoren.
- **Stability:** Veränderte Umwelt führt nicht gleich zu neuem Verhalten.

Zwischen den sich widersprechenden Prinzipien Adaptability und Stability, die für sich allein genommen einen Schwarm in Chaos bzw. in Ordnung ergeben, muss eine sinnvolle Balance gefunden werden. Der Schwarm, der bei PSO Anwendung findet, besitzt alle fünf Eigenschaften von Millonas und unterliegt somit dem Konzept der Schwarmintelligenz [112, S. 212–226].

Sozialverhalten Schwarmverhalten kann in der Natur bei Vögeln, Fischen und in hoch organisierten Insektenpopulationen beobachtet werden [387, S. 27], [282, S. 513]. Dabei bewegen sich die Schwärme oft synchron, lösen sich plötzlich auf und gruppieren sich neu. Diese Bewegungen sind das Resultat des Sozialverhaltens der Schwarmmitglieder, welches Parallelen zu menschlichem Sozialverhalten aufweist.

PSO wurde entwickelt, um menschlichen Sozialverhalten und Denken zu simulieren. Ein Unterscheidungsmerkmal des Sozialverhaltens bei Menschen und bei Tieren ist in der Abstraktion zu sehen. So passen Fische, Vögel oder Insekten ihr Verhalten an Raubtiere, Nahrungsquellen und anderen Umweltbedingungen an. Menschen hingegen beziehen zusätzlich kognitive und erfahrungsbezogene Parameter mit ein. Dies ist auch der Grund, warum sich Menschen gewöhnlich nicht in synchronen Formationen bewegen. Sie neigen eher dazu, Verhalten und Einstellung an die Personen in ihrer Nähe anzupassen. Ein weiteres Unterscheidungsmerkmal im Sozialverhalten von Tiergemeinschaften und Menschen ist die Kollision. Zwei Vögel oder Fische sind nicht in der Lage, die gleiche Position im physischen Raum einzunehmen, wogegen zwei Individuen mit der gleichen Einstellung nicht zusammenstoßen können. Ihr Verhalten und ihre Einstellung manifestieren sich in einem abstrakten mehrdimensionalen Raum. [204, S. 1942–1943] In diesem Fall kollidiert das Denken der Individuen nicht – es stimmt miteinander überein [206, S. 111].

Von ersten Simulationsmodellen zur PSO Ursprünglich entwickelte Reynolds [325, S. 25] das Boids-Modell, um koordinierte Schwarmbewegung für Computergrafiken und Filme zu simulieren. Ein Boid stellt ein einzelnes zu simulierendes Individuum dar. In Abbildung 4.20 sind die drei Regeln der Boids dargestellt, die das Schwarmverhalten beeinflussen.

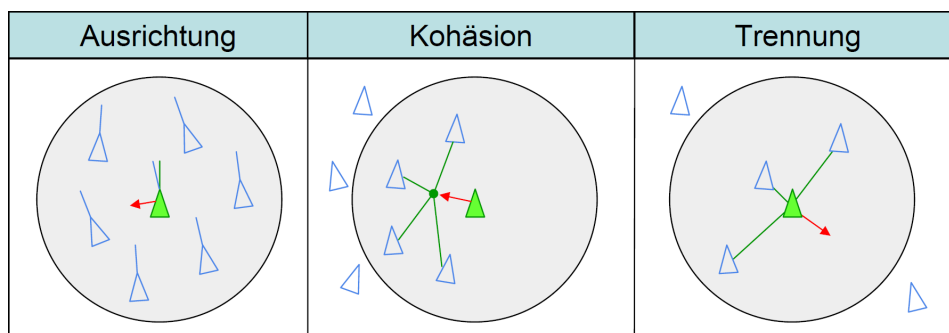


Abbildung 4.20: Regeln der Boids [326]

- Ausrichtung: Boids bewegen sich in die Richtung, in die sich auch ihre Nachbarn bewegen.

- Kohäsion: Boids bewegen sich zur mittleren Position einer Gruppe.
- Trennung: Um Gedränge zu vermeiden streben Boids nach außen.

Zur Vermeidung einer allzu gleichförmigen Bewegung der Boids fügten Heppner und Grenander [181, S. 233–238] Rastplätze hinzu, die angesteuert werden können. Dabei hat jedes Subjekt einen Faktor, der dem Wunsch zu rasten entspricht und dem Wunsch im Schwarm zu bleiben entgegenwirkt. Der Wunsch zu rasten wird dabei umso größer, je näher der Rastplatz ist. Entdeckt ein Individuum einen Rastplatz, so zieht es seine Nachbarn mit sich, die dann wiederum ihre Nachbarn in Richtung des Rastplatzes lenken, bis der komplette Schwarm gelandet ist.

Kennedy und Eberhart nutzten 1995 den Ansatz der Boids und erweiterten das Modell von Reynolds um ein von den Partikeln zu findendes Optimum, das sie „Cornfield Vector“ nannten [204, S. 1944]. Von der Fähigkeit einen bestimmten Punkt zu finden und dabei Erfahrungen auszutauschen, leiteten sie ihr Modell der Particle Swarm Optimization ab [206, S. 114]. Die Schwarmmitglieder sind nun masselose, kollisionsfreie Partikel. Sie müssen daher nicht mehr auf den Abstand zu anderen Partikeln achten. Statt eines Rastplatzes suchen die Partikel in einem Lösungsraum nach dem Optimum.

Die Gestalt des Lösungsraums ist durch eine Fitnessfunktion definiert [204, S. 1943, 1947]. Dabei stellt jedes einzelne Partikel mit seiner Position eine Lösung des Problems dar [394, S. 111]. Bei der Suche nach dem Optimum orientieren sich die Partikel nicht nur an ihren eigenen Erfahrungen, sondern auch an den Erfahrungen ihrer Nachbarn [129, S. 25]. Sie tauschen Informationen aus, was die Entwicklung einer Population in sozialen Systemen günstig beeinflusst kann [300, S. 235].

4.14.2 Grundlegender Ablauf

Zunächst wird der ursprüngliche PSO-Algorithmus von Kennedy und Eberhart vorgestellt [204], der in einem Lösungsraum der Gestalt \mathbb{R}^d operiert. Jedes Partikel hat einen Vektor für die eigene Position (eine Lösung der Problemstellung) und die eigene Geschwindigkeit. Zudem kennt jedes Partikel seine persönlich beste Position (pBest), die es jemals gefunden hat und die beste Position seiner Nachbarschaft (gBest bzw. lBest – Nachbarschaftstopologien werden in Kapitel 4.14.2.1 erläutert) [387, S. 21–22]. Nachfolgend ist der Ablauf in Algorithmus 17 dargestellt.

Die Initialisierung des Schwarms mit Startposition und -geschwindigkeit erfolgt gewöhnlich zufällig [395, S. 1584]. Anschließend beginnt der iterative Optimierungsprozess. Die Formeln für die Berechnung der neuen Geschwindigkeit \vec{v}_i und Position \vec{x}_i eines Partikels i zum Zeitpunkt t sehen wie folgt aus. Dabei steht \vec{p}_g für den Positionsvektor von gBest,

Algorithmus 17 PSO in Anlehnung an [204] für ein Minimierungsproblem in einer gBest-Topologie

```

1: Initialisiere den Schwarm
2: Berechne die Fitness der Partikel
3: Bestimme pBest und gBest
4: repeat
5:   for i = 1 bis Anzahl Partikel do
6:     Berechne die neue Position des Partikels
7:     Berechne die Fitness des Partikels
8:     if f(neue Position) < f(pBest) then
9:       pBest = neue Position
10:    end if
11:    if f(pBest) < f(gBest) then
12:      gBest = pBest
13:    end if
14:  end for
15: until Abbruchkriterium tritt ein

```

\vec{p}_i für den Positionsvektor von pBest des Partikels i und $\vec{x}_i(t-1)$ für den Positionsvektor des Partikels i zum Zeitpunkt $(t-1)$.

$$\vec{v}_i(t) = \vec{v}_i(t-1) + \varphi_1(\vec{p}_i - \vec{x}_i(t-1)) + \varphi_2(\vec{p}_g - \vec{x}_i(t-1)) \quad (4.17)$$

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (4.18)$$

Zur Gewichtung des individuellen und sozialen Verhaltens eines Partikels werden die beiden gleich verteilte Zufallsvariablen φ_1 und φ_2 verwendet, die eine obere Grenze haben. Denn in Formel 4.17 fließt die eigene bisher beste Position (pBest) und die beste Position aus der Nachbarschaft des Partikels (gBest bzw. lBest) ein. Mit der Berücksichtigung von pBest wird das individuelle Verhalten einbezogen. Mit der besten Position aus der Nachbarschaft hingegen fließt die Erfahrung des Schwarms bzw. der Nachbarschaft in die Berechnung der Geschwindigkeit mit ein (soziales Verhalten). Sollte bei einem Partikel pBest gleich gBest bzw. lBest sein, entfällt bei diesem Partikel der soziale Einfluss auf die Geschwindigkeit. Die Geschwindigkeiten der Partikel tendiert mit der Zeit zur Explosion. Um eine unbegrenzte Zunahme zu unterbinden, kann die folgende Formel verwendet werden [206, S. 295, 312–313]. Dort ist v_{id} die Geschwindigkeit des Partikels i in der Dimension d und V_{max} die maximal zulässige Geschwindigkeit.

$$\begin{aligned} & \text{if } (v_{id} > V_{max}) \text{ then } v_{id} = V_{max} \\ & \text{else if } (v_{id} < -V_{max}) \text{ then } v_{id} = -V_{max} \end{aligned} \quad (4.19)$$

Der PSO-Algorithmus bricht im Allgemeinen ab, wenn ein bestimmter Fitnesswert über-

bzw. unterschritten ist, der Algorithmus eine vorgegebene Anzahl an Iterationen absolviert hat oder andere Kriterien eintreten [346, S. 8].

Inertia Weight Mit dem Inertia Weight legt man einen Gewichtungsfaktor für die vorherige Partikelgeschwindigkeit fest. Zusätzlich zu den dargestellten Geschwindigkeitsbegrenzungen der einzelnen Dimensionen kann somit die neue Partikelgeschwindigkeit und somit das Verhalten des Schwarms beeinflusst werden. Ein sehr kleines Inertia Weight führt dazu, dass der Schwarm stärker zu einer lokalen Suche tendiert, was die Konvergenz des Schwarms beschleunigt. Hierdurch kann er eher in einem lokalen Optimum konvergieren. Mit einem großen Inertia Weight neigt der Schwarm auf der anderen Seite zu einer globalen Erkundung des Lösungsraums. Dies fördert die Robustheit, kann jedoch das Finden des Optimums erschweren. Das Verhalten des Schwarms hängt jedoch nicht allein vom Inertia Weight ab, sondern auch von anderen Einflussfaktoren wie V_{min} , V_{max} , φ_1 , φ_2 , der Schwarmgröße und der Fitnessfunktion. Laut Kennedy, Eberhart und Shi hat sich ein Inertia Weight von 0,8 bei bestimmten Problemstellungen bewährt. Die Formel zur Berechnung der neuen Geschwindigkeit mit einem Inertia Weight w ist nachfolgend dargestellt [206, S. 339–342].

$$\vec{v}_i(t) = w * \vec{v}_i(t-1) + \varphi_1(\vec{p}_i - \vec{x}_i(t-1)) + \varphi_2(\vec{p}_g - \vec{x}_i(t-1)) \quad (4.20)$$

Die Wahl des Inertia Weights kann das Verhalten der Partikel stark beeinflussen und ein starrer Wert mag nicht immer gut geeignet sein. Ein dynamisches Inertia Weight hat den Vorteil, dass der Schwarm zuerst den Lösungsraum global erkundet und im Zeitablauf zu einer lokalen Erkundung tendiert. Das Inertia Weight beginnt mit einem großen Wert und sinkt im Zeitablauf. Robustheit und schnelle Konvergenz können so kombiniert werden. Problematisch ist, dass der Schwarm womöglich schon lokal sucht, obwohl er sich noch nicht in der Nähe des Optimums befindet. [347], [348, S. 1947–1948]

Constriction Factor Die Einführung eines Beschränkungsfaktors ist eine weitere Möglichkeit, die Berechnung der Partikelgeschwindigkeit zu beeinflussen. Der Constriction Factor χ dämpft die Geschwindigkeit. In Untersuchungen von Clerc und Kennedy hat sich ein Constriction Factor von 0,73 bewährt, der sich nach der folgenden Formel aus $\varphi = 4,1$ und $k = 1$ berechnet. Die folgende Formel ist nur für $\varphi > 4$ anzuwenden [71].

$$\chi = \frac{2k}{\left| \varphi - 2 + \sqrt{\varphi^2 - 4\varphi} \right|} \quad (4.21)$$

Ein kleiner Constriction Factor kann die Geschwindigkeit derart verringern, dass der Schwarm den Lösungsraum nicht komplett erkundet und in einem lokalen Optimum

konvergiert. Bei einem großen Wert verhält sich der Schwarm eher explorativ. Die Robustheit wird erhöht, aber die konkrete Bestimmung des Optimums erschwert. Auch hier hängt das Verhalten des Schwarms nicht allein vom Constriction Factor ab. So haben V_{min} , V_{max} , φ_1 , φ_2 , die Schwarmgröße und die Fitnessfunktion ebenfalls einen Einfluss. Nachfolgend ist die Formel für die Berechnung der Geschwindigkeit mit einem Constriction Factor aufgeführt [432, S. 528–531], [206, S. 338–339].

$$\vec{v}_i(t) = \chi * (\vec{v}_i(t-1) + \varphi_1(\vec{p}_i - \vec{x}_i(t-1)) + \varphi_2(\vec{p}_g - \vec{x}_i(t-1))) \quad (4.22)$$

Behandlungsmethoden für Dimensionsüberschreitungen Beim zuvor dargestellten Vorgehen der Positionsneuberechnung eines Partikels wurde nicht auf mögliche Dimensionsgrenzen geachtet. Um ungültige Positionen außerhalb der Dimensionsgrenzen zu vermeiden, müssen derartige Verstöße behandelt werden. Besonders wenn das Optimum in mindestens einer Dimension auf einer Dimensionsgrenze liegt oder $V_{max} = X_{max} - X_{min}$ ist, nehmen Verstöße zu.

In Abbildung 4.21 sind die Methoden Stick, Wrap und Bounce zur Behandlung von Dimensionsüberschreitungen dargestellt. Mit dem Verfahren Stick wird das Partikel auf die Dimensionsgrenze gesetzt, die überschritten wurde. In Abbildung 4.21 würde die neue Partikelposition in der entsprechenden Dimension eigentlich 12 sein. Da X_{max} jedoch 10 ist, wird die neue Partikelposition ebenfalls auf 10 gesetzt. Während die Partikelposition bei Stick auf die Dimensionsgrenze gesetzt wird, wird sie bei Bounce an der überschrittenen Dimensionsgrenze „zurückgeworfen“. In Abbildung 4.21 erhält sie somit den Wert 8. Bei Wrap tritt das Partikel bei einer Überschreitung an der gegenüberliegenden Dimensionsgrenze in den Lösungsraum ein und wird daher in Abbildung 4.21 auf 2 gesetzt.

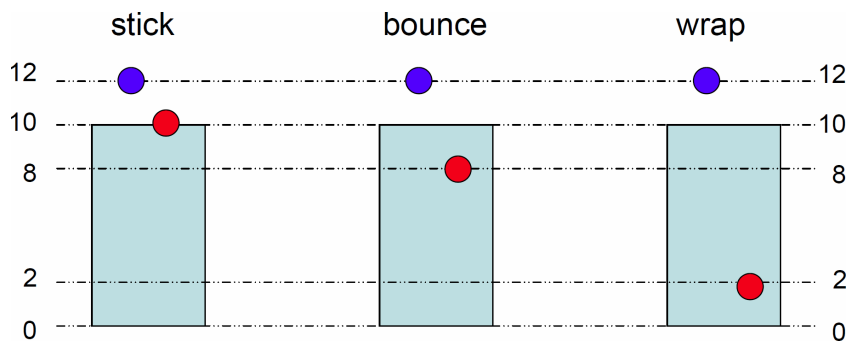


Abbildung 4.21: Methoden zur Behandlung von Dimensionsüberschreitungen [47, S. 17]

4.14.2.1 Nachbarschaftstopologien

In der Vergangenheit wurden zahlreiche Modifikationen und Erweiterungen am klassischen PSO-Algorithmus von Kennedy und Eberhart [204] vorgenommen. So zum Beispiel die Einführung des Inertia Weights, des Constriction Factors oder die Anpassung an binäre und ganzzahlige Lösungsräume sowie an kombinatorische Problemstellungen. Aber auch die Wahl der geeigneten Nachbarschaftstopologie (Soziometrie) spielt eine wichtige Rolle für die Effektivität der PSO. Es gibt jedoch keine generelle Aussage darüber, welche Nachbarschaftstopologie besser und welche schlechter ist, da ihre Effektivität u.a. von der zu lösenden Problemstellung abhängt. Bei PSO wird das Verhalten der Partikel (also die neue Partikelposition in jeder Iteration) u.a. durch den individuellen Einfluss und den sozialen Einfluss „anderer Partikel“ bestimmt. Der individuelle Einfluss steht für die beste Position, die das jeweilige Partikel bisher gefunden hat. Was im Rahmen des sozialen Einflusses als „anderer Partikel“ bezeichnet werden kann, hängt von der Nachbarschaftstopologie des Schwarms ab. Das soziale Verhalten bezieht sich demnach auf die beste von der Nachbarschaft bisher gefundene Position. Watts hat gezeigt, dass der Informationsfluss in sozialen Strukturen durch verschiedene Eigenschaften beeinflusst wird [411], [410]:

- Zusammenschluss (k): Die Größe der Nachbarschaft eines Partikels.
- Gruppierung (C): Die Anzahl an Nachbarn eines Partikels, die gleichzeitig Nachbarn voneinander sind.
- Entfernung (L): Der durchschnittliche kürzeste Abstand zwischen zwei Nachbarn. Dieser wird stark von k und C beeinflusst.

Die besonders häufig in der ursprünglichen Form des PSO verwendeten Topologien sind die gBest- und lBest-Topologie. Bei gBest sind die Schwarmmitglieder derart miteinander verbunden, dass jedes Partikel der Nachbar jedes anderen Partikels ist. Somit ist in der gBest-Topologie jedem Partikel sofort der bisher beste gefundene globale Wert bekannt. Alle Partikel werden bei der Positionsberechnung von gBest angezogen. Befindet sich das globale Optimum nicht nahe genug am besten Partikel, hat es der Schwarm schwer, andere Gebiete des Lösungsraums zu erkunden und kann möglicher Weise in einem lokalen Optimum konvergieren. Der Schwarm tendiert also zu einer relativ schnellen Konvergenz in einem Punkt, wobei dies der Robustheit schaden kann [262].

Mit der lBest-Topologie, bei der ein Partikel nur mit seinen Nachbarn verbunden ist, soll die vorzeitige Konvergenz in einem Suboptimum verhindert werden. Der Parameter k steht für die Anzahl der Nachbarn eines Partikels. Bei $k = 2$ handelt es sich um eine Circle-Topologie (bzw. Ring). Wird k auf die Partikelanzahl minus 1 erhöht, ergibt sich wiederum eine gBest-Topologie. Jedes Partikel besitzt lediglich Informationen über sich selbst und über seine Nachbarn. Der Schwarm konvergiert langsamer als bei gBest, hat

dafür aber eine größere Chance, das globale Optimum zu finden [206]. Im Vergleich zur ursprünglichen Form ändert sich durch die Verwendung von lBest am grundlegenden Ablauf der PSO nichts. Zur Berechnung der aktuellen Geschwindigkeit eines Partikels wird lediglich gBest durch lBest ersetzt [129].

Bzgl. der Bildung von Nachbarschaften können zwei Varianten genutzt werden. So besteht die Nachbarschaft aus einem nummerierten Feld, dem Partikel zugeordnet sind [203]. Die Bildung der Nachbarschaft ist also extern bestimmt und ist nicht von den Partikelpositionen im Lösungsraum abhängig. Eine weitere Variante geht nicht von der Anordnung in einem nummerierten Feld aus, sondern von der Distanz zweier Partikel im Lösungsraum. Zur Nachbarschaft eines Partikels gehören also all die Partikel, welche eine bestimmte Entfernung zu diesem Partikel unterschreiten. Die Nachbarschaften sind also nicht fest vorgegeben, sondern können sich bei jeder Iteration verändern. Der Nachteil dieser Methode ist der Rechenaufwand für die notwendige Ermittlung der Distanzen bei jeder Iteration [261].

Eine weitere Nachbarschaftsform ist Wheel (bzw. Star). Es existiert ein zentrales Partikel, welches mit allen anderen Partikeln verbunden ist. Jedes Partikel hat also nur das zentrale Partikel als Nachbarn und ist von allen anderen Partikeln isoliert. Durch diese Anordnung wird ein neues gBest nicht sofort im gesamten Schwarm verbreitet. Das zentrale Partikel wirkt wie ein Filter, ähnlich den Organisationsstrukturen in Unternehmen, wo es ein zentrales Management und Mitarbeiter gibt.

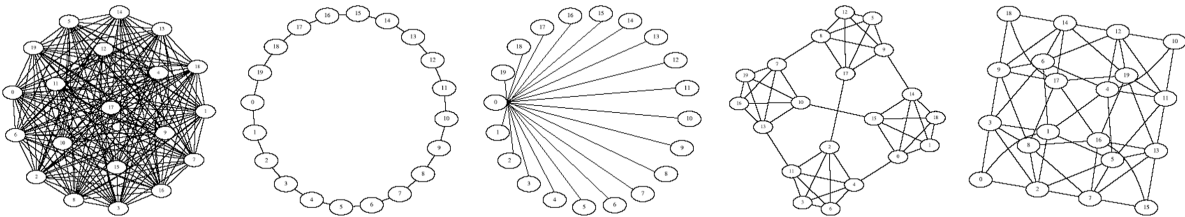


Abbildung 4.22: Nachbarschaftstopologien gBest, Circle ($k = 2$), Wheel, Cluster und Pyramide [263]

Zur Vermeidung der frühzeitigen Konvergenz in einem lokalen Optimum wurden in der Vergangenheit diverse Modifikationen und Neuerungen bzgl. der Nachbarschaftstopologien erprobt. Besonders kontinuierliche Testfunktionen wie Sphere, Rastrigin, Griewank, Rosenbrock, f_6 , etc. standen dabei im Fokus. Nachfolgend wird kurz auf verschiedene Arbeiten eingegangen, wobei auch Untersuchungen an kombinatorischen Problemstellungen dargestellt werden.

Suganthan verändert 1999 die Nachbarschaftsgröße kontinuierlich [362]. Er startet in einer lBest-Topologie mit $k = 1$. Die Nachbarschaft besteht anfangs nur aus dem Parti-

kel selbst. Im Zeitablauf wird k kontinuierlich erhöht, bis eine gBest-Topologie erreicht wird. Dabei wird die Nachbarschaft über die Hamming-Distanz der Partikel bestimmt. An mehreren kontinuierlichen Testfunktionen mit 20, 30 und 50 Dimensionen erprobt Suganthan die so modifizierte PSO mit 50 Partikeln. Im Vergleich zur Standard-PSO konnten meist bessere Ergebnisse an den Testproblemen erzielt werden.

In ihrer Arbeit aus dem Jahre 2002 untersuchen Kennedy und Mendes [207] verschiedene Nachbarschaftstopologien an kontinuierlichen Testfunktionen. Vier mit 30 und eine mit zwei Dimensionen. Neben den bereits vorgestellten Topologien wurden die Pyramide-, die Small-, die von Neumann-Topologie sowie zufällig generierte Topologien verwendet. Pyramide und von Neumann schnitten am besten ab. Wheel und gBest am schlechtesten.

Blackwell und Bentley [36] nutzen im gleichen Jahr PSO, um Musik zu erzeugen. Sehr interessant ist, dass sie eine zusätzliche Beschleunigung in die Berechnung der neuen Partikelposition addieren, um die Anziehungskraft des Schwarmzentrums zu reduzieren. Durch die Vermeidung von Kollisionen wird die Konvergenzgeschwindigkeit verlangsamt.

Einen ähnlichen Ansatz verfolgen Krink, Vesterstrom und Riget 2002 [229], indem sie einen Radius um jedes Partikel legen. Wenn zwei Partikel kollidieren würden, setzt man sie auseinander, um eine Kollision und somit eine Gruppierung zu vermeiden.

In ihrer Arbeit fügen Løvbjerg und Krink 2002 [251] zu jedem Partikel eine Variable (Critical Value) hinzu. Diese Variable repräsentiert einen kritischen Wert, der erhöht wird, wenn sich Partikel einander nähern. Überschreitet der kritische Wert eine bestimmte Schwelle, so erfolgt eine Umbesetzung der entsprechenden Partikel. An vier kontinuierlichen Testfunktionen mit jeweils 30 Dimensionen konnten mit diesem Vorgehen bessere Ergebnisse erzielt werden.

Xie, Zhang und Yang [427] integrieren in 2002 negative Entropie in den Schwarm, wodurch sie Chaos in der Partikelgeschwindigkeit und der Partikelposition einführen. Somit soll verhindert werden, dass der Schwarm in einem Punkt konvergiert. An zwei kontinuierlichen Testfunktionen mit jeweils 10, 20 und 30 Dimensionen wird diese Dissipative PSO mit sehr guten Resultaten angewendet.

2008 benutzt Hamdan [173] mehrere Nachbarschaften zeitgleich. Je Iteration prüft jedes Partikel den Fitnesswert seiner neuen Position für eine Star-, Circle- und von Neumann-Topologie. Das Partikel verwendet dann die Topologie, die das beste Ergebnis liefert. An 9 kontinuierlichen Testfunktionen mit 2 bis 30 Dimensionen wurde dieses Verfahren erprobt. Es lieferte bei fast allen Testfunktionen neue Bestwerte. Ein erhöhter Rechenbedarf muss jedoch für dieses Vorgehen in Kauf genommen werden.

Alle zuvor aufgeführten Experimente mit verschiedenen Nachbarschaftstopologien wurden an kontinuierlichen Testfunktionen mit relativ wenigen Dimensionen durchgeführt.

Für diese Arbeit sind jedoch besonders Untersuchungen an kombinatorischen Problemstellungen interessant.

Sivanandam, Visalakshi und Bhuvaneswari [349] nutzen 2007 u.a. PSO in einer gBest- und einer lBest-Topologie, um die Problemstellung des Multiprozessor Scheduling zu bearbeiten. Die gBest-Topologie schneidet dabei besser ab als die lBest-Topologie.

Czogalla und Fink [89] untersuchten 2009 die gBest-, lBest-, Wheel- und eine hierarchische Topologie (H-PSO) an einer kombinatorischen Problemstellung. Am Beispiel von vier Problemen der Projektplanung (RCPS) mit 30 bis 120 Aktivitäten wurden Experimente durchgeführt. Dabei beträgt die Schwarmgröße je nach Problemstellung und Nachbarschaftstopologie zwischen 20 und 70 Partikel. Man hat gezeigt, dass die Topologien lBest und H-PSO bessere Ergebnisse als Wheel und gBest liefern. Am schlechtesten schneidet gBest ab.

Insgesamt wird deutlich, dass sich im Vergleich all der genannten Arbeiten keine der Nachbarschaftstopologien uneingeschränkt gegen andere durchsetzt.

4.14.2.2 Anpassung an binäre und diskrete Lösungsräume

In Kapitel 4.14.2 wurde nur auf den PSO-Algorithmus für stetige Variablen eingegangen. Grundsätzlich ist PSO aber auch für binäre und diskrete Variablen geeignet. Auch eine gemeinsame Verwendung ist möglich, wobei sich PSO auch hier als sehr effektiv erweist [130]. Durchquert ein Partikel einen diskreten Lösungsraum, bleiben die Bereiche zwischen den ganzzahligen Werten unberücksichtigt [179].

Die erste Form für Discrete PSO (DPSO) mit binären Variablen wurde 1997 von Kennedy und Eberhart [205] vorgestellt. Dort wird die Geschwindigkeit eines Partikels in der jeweiligen Dimension als Wahrscheinlichkeit für die Änderung des binären Wertes genutzt. Untersuchungen von Kennedy und Spears haben gezeigt, dass ein derartiger PSO-Algorithmus bei einigen Problemstellungen effizienter ist als GA [208].

Al-Kazemi und Mohan [8] stellen 2002 einen PSO-Algorithmus für binäre Lösungsräume vor. Dort sind Partikel in zwei Gruppen geteilt und durchlaufen wiederholt zwei Phasen (M-DPSO = Multiphase-DPSO). In der ersten Phase bewegen sich die Partikel der ersten Gruppe in Richtung gBest und die der zweiten Gruppe in die Gegenrichtung. In Phase zwei werden die Laufrichtungen der Gruppen vertauscht. Im Vergleich zu einem GA und DPSO schnitt M-DPSO bei Al-Kazemi und Mohan besser ab.

Für diskrete Lösungsräume wäre es prinzipiell möglich, jede Variable in eine binäre Form zu transformieren, um Problemstellungen mit dem binären PSO-Algorithmus zu lösen. Dieses Vorgehen erhöht jedoch die Anzahl an Dimensionen und z.T. die Komplexität.

Zudem sind spezielle Methoden erforderlich, um die Dimensionsgrenzen der Variablen auch nach der Transformation in binäre Variablen einzuhalten. Aus diesen Gründen wurde das ursprüngliche Konzept von Kennedy und Eberhart für binäre Variablen auf Variablen aus dem Dezimalsystem übertragen [393, S. 142].

Ein Verfahren, welches in der Vergangenheit erfolgreich angewendet wurde und es PSO erlaubt, weiterhin stetige Variablen zu verwenden, besteht darin, die berechnete Position der Partikel zu runden, bevor sie zugewiesen wird. Somit lässt sich die ursprüngliche Form der PSO sehr leicht modifizieren. Venter und Sobieszczanski-Sobieski [395, S. 1587] benutzen diese Variante 2003 für Berechnungen an einem freitragenden Ausleger. An sieben Problemstellungen der ganzzahligen Programmierung mit 2 bis 30 Dimensionen wird von Parsopoulos und Vrahatisin [300, S. 282–288] ein PSO-Algorithmus genutzt, bei dem die Werte gerundet werden. Günther [145] und später Brodersen und Schumann [48] untersuchen mit diesem Vorgehen erfolgreich ein stochastisches Lagerhaltungsproblem. PSO eignete sich für diese Problemstellungen sehr gut. Das Problem, dass durch das Runden womöglich Informationen verloren gehen, wurde in den aufgeführten Arbeiten nicht beobachtet. Ein Kritikpunkt an diesem Vorgehen besteht jedoch darin, dass sich die Partikel beim Runden u.U. weiter vom Optimum entfernen. Zudem besteht die Gefahr, durch das Runden Werte zu erzeugen, die gegen Restriktionen verstoßen.

4.14.2.3 Anpassung an kombinatorische Problemstellungen

Nachfolgend werden Ansätze aus der Literatur vorgestellt, bei denen PSO für kombinatorische Problemstellungen angepasst wird. Bei der Routenplanung unbemannter Aufklärungsflugzeuge verwendet Secrest [345] Übergangswahrscheinlichkeiten im PSO. Also ein ähnlicher Ansatz wie beim DPSO von Kennedy und Eberhart [205]. Ausgehend von einer Startposition wird eine Wahrscheinlichkeit ermittelt, mit der ein zufälliger Ort, ein Ort aus gBest oder ein Ort pBest angefliegen werden soll. Ist dieser neue Ort bestimmt, wird wiederum eine Wahrscheinlichkeit für den Weiterflug erzeugt. In jeder Iteration wird die Lösung also Schritt für Schritt erstellt, wobei es bestimmte Regeln gibt. Z.B. um Zyklen zu vermeiden.

Tasgetiren u.a. [367] verwenden 2004 den Smallest Position Value (SPV), der für alle Formen der Reihenfolgeplanung geschaffen wurde. Mit ihm wird der ursprünglich von Kennedy und Eberhart für kontinuierliche Problemstellungen entwickelte PSO-Algorithmus für kombinatorische Problemstellungen angepasst. Dabei kann gezeigt werden, dass sich der SPV durchaus eignet, um PSO beim Scheduling anwenden zu können. Das Verfahren wurde auch an einem Job Shop Scheduling Problem [368], an einem Flow Shop Sequencing Problem [385], [366] und an einer Sammlung von Traveling Salesman Problemen (TSPLIB [323]) [369] angewendet. Dort waren die Ergebnisse durchweg positiv. Z.B. konnte gezeigt werden, dass PSO bei den Problemen der TSPLIB neben dem GA und den Heuristiken FST-Lagr sowie FST-Root mit am besten abschneidet. In Tabelle 4.9

wird das Vorgehen beim SPV visualisiert, wobei jede Dimension für einen Auftrag steht. Mit -1,20 hat die fünfte Dimension den kleinsten Wert aller Dimensionen, sodass der Auftrag fünf an die erste Stelle gesetzt wird. Dieses Vorgehen wird analog für alle weiteren Dimensionen durchgeführt, bis alle Aufträge in eine Reihenfolge gebracht sind.

Tabelle 4.9: Lösungsrepräsentation mit dem SPV

j	1	2	3	4	5	6
x_{ij}^k	1,80	-0,99	3,01	-0,72	-1,20	2,15
v_{ij}^k	3,89	2,94	3,08	-0,87	-0,20	3,16
s_{ij}^k	5	2	4	1	6	3

Zur Lösung des Traveling Salesman Problems gehen Pang u.a. [299] anders vor und kombinieren einen für kombinatorische Problemstellungen modifizierten PSO-Algorithmus mit einer Fuzzy Matrix. Der Algorithmus wurde ebenfalls an Problemstellungen der TSPLIB [323] getestet und zeigte dort sehr gute Ergebnisse. Für weitere Anwendungen dieses Verfahrens auf das Quadratische Zuweisungsproblem sei auf die Arbeiten von Liu u.a. [248], [249] verwiesen.

Li [245] untersucht 2005 eine Vielzahl unterschiedlichster PSO-Varianten, um Bedingungserfüllungsprobleme zu lösen. Je nach Problemstellung schneiden die einzelnen Verfahren verschieden gut ab, wobei grundsätzlich festgestellt wird, dass die Variante `bcspZigzagHop` am geeignetsten ist. Schlechter schneidet die Variante `genericZigzagHop` ab.

Fealko [118] wendet den kanonischen PSO-Algorithmus 2005 bei der Prüfungsorganisation an. Der Algorithmus wird nur wenig verändert. Lediglich eine Auswahl bei der Zuweisung neuer Partikelpositionen wird hinzugefügt. Faelko nutzt zur Abbildung der Problemstellung eine zweidimensionale Matrix mit ganzen Zahlen und rundet die berechneten Partikelpositionen auf ganzzahlige Werte. Er zeigt, dass sich die kanonische Form des PSO-Algorithmus mit Rundung der Positionen nicht für die kombinatorische Problemstellung der Prüfungsorganisation eignet. In ersten Experimenten am Problem der untertägigen Personaleinsatzplanung in der Logistik und am Handelsproblem wurden die gleichen Erfahrungen gemacht, weshalb der Ansatz in dieser Arbeit nicht weiter fortgeführt wird.

2006 haben Chu u.a. [69] PSO ebenfalls an einer Problemstellung der Prüfungsorganisation untersucht. Für die Anpassung der PSO an kombinatorische Problemstellungen wurde das bisherige Vorgehen derart geändert, dass keine Geschwindigkeit mehr berechnet wird, die zur Bestimmung der neuen Position des Partikels dient. Die neue Position eines Partikels ergibt sich in jeder Iteration durch Vertauschen zweier Zuweisungen in einem Partikel sowie durch Kopieren einer Zuweisung aus der persönlich oder global

besten Position in die neue Partikelposition. Es kann gezeigt werden, dass PSO in dieser modifizierten Form ein effizientes Lösungsverfahren für die Prüfungsorganisation ist. Algorithmus 18 zeigt den Ablauf dieser PSO-Variante.

Algorithmus 18 PSO nach Chu u.a. [69] für ein Minimierungsproblem in einer gBest-Topologie

```
1: Initialisiere den Schwarm
2: Berechne die Fitness der Partikel
3: Bestimme pBest und gBest
4: repeat
5:   for i = 1 bis Anzahl Partikel do
6:     Berechne die neue Position des Partikels mit Hilfe von 3 Aktionen
7:     Berechne die Fitness des Partikels
8:     if f(neue Position) < f(pBest) then
9:       pBest = neue Position
10:    end if
11:    if f(pBest) < f(gBest) then
12:      gBest = pBest
13:    end if
14:  end for
15: until Abbruchkriterium tritt ein
```

Beim Schritt der Erzeugung einer neuen Lösung (Zeile 6) geschieht in jeder Iteration folgendes:

1. Jedes Partikel muss zwei Dimensionen seiner Position zufällig ermittelten und deren Werte vertauschen.
2. Danach ermittelt jedes Partikel eine Dimension in seiner Position zufällig und kopiert an diese Stelle den entsprechenden Wert aus pBest.
3. Anschließend ermittelt jedes Partikel wiederum eine Dimension in seiner Position zufällig und kopiert an diese Stelle den entsprechenden Wert aus gBest.

Ausgehend von dieser Modifikation für die Prüfungsorganisation wird die universitäre Stundenplanerstellung von Brodersen und Schumann in 2007 [47] sowie von Brodersen in 2008 [46] untersucht. Am dargestellten Ablauf von Chu u.a. für die Prüfungsorganisation ändert sich nichts. Lediglich der Schritt zur Erzeugung einer neuen Lösung wird weiter modifiziert, sodass nun die nachfolgend aufgeführten Aktionen für die Erzeugung einer neuen Partikelposition zur Verfügung stehen. Dabei werden nicht immer alle Aktionen benutzt. Vielmehr hat man Testläufe durchgeführt, bei denen einige der Aktionen gänzlich unberücksichtigt bleiben. Als beste Variante stellt sich die Einbeziehung von Aktion 2 und 4 heraus (Aktion 1 und 3 bleiben unberücksichtigt). Mit diesem Ansatz

kann der händisch erzeugte Stundenplan wesentlich verbessert werden, wenngleich es nicht gelingt, eine gültige Lösung zu erzeugen.

1. Jedes Partikel muss ein Zeitfenster mit den zugehörigen Veranstaltungen aus der aktuellen Position vertauschen.
2. Jedes Partikel wählt eine Dimension in seiner Position zufällig aus und kopiert an diese Stelle den entsprechenden Wert aus pBest.
3. Jedes Partikel muss ein Zeitfenster mit den zugehörigen Veranstaltungen aus gBest vertauschen.
4. Jedes Partikel wählt eine Dimension in seiner Position zufällig aus und kopiert an diese Stelle den entsprechenden Wert aus gBest.

In einem Verfahren, welches Jumping Particle Swarm Optimization (JPSO) bzw. Jumping Frogs Optimization (JFO) genannt wird, wird ebenfalls auf die Elemente Geschwindigkeit und Trägheit bei der Positionsberechnung der Partikel verzichtet. Vielmehr wird eine Bewegung in Form von Sprüngen vollzogen. Die Gewichte bei der Berechnung einer neuen Position werden als Wahrscheinlichkeiten für das Verhalten eines Partikels gesehen. So kann sich jedes Partikel in jeder Iteration zufällig bewegen oder von den besten Punkten im Lösungsraum angezogen werden. Die Anziehungskraft dieser Punkte bewirkt beim Partikel eine zufällig ermittelte Anzahl an Sprüngen in die jeweilige Richtung – ähnlich einem Frosch, der von Seerosenblatt zu Seerosenblatt in den Teich springt. Das Partikel springt dabei in die Richtung des entsprechenden Anziehungspunktes – nicht zwangsläufig auf diesen Punkt. Nachfolgend ist die Formel für die Positionsberechnung bei der JPSO dargestellt.

$$x_i = c_1 * x_i \otimes c_2 * b_i \otimes c_3 * g_i \otimes c_4 * g^* \quad (4.23)$$

Dabei stehen c_1 für die Wahrscheinlichkeit eine zufällige Bewegung zu machen und c_2 für die Wahrscheinlichkeit einen Sprung in Richtung pBest zu machen. Mit c_3 wird die Wahrscheinlichkeit angegeben, einen Sprung in Richtung der besten Position in der aktuellen Iteration i über alle Partikel des Schwarms zu machen. Der Sprung in Richtung gBest wird über die Wahrscheinlichkeit c_4 eingeleitet. Moreno Pérez u.a. wenden JPSO 2007 [310] und 2008 [134] erfolgreich am p-Median-Problem an, das bekanntlich NP-schwer ist. Mit diesem Verfahren (mit und ohne Lokaler Suche) hat man bessere Ergebnisse als mit einem DPSO erzielt. Consoli u.a. [76] erproben den JPSO 2008 am Steinerbaumproblem, welches NP-vollständig ist. Im Vergleich zu einem exakten Verfahren, der Pilot-Methode (PM) und einem einfachen Multistartverfahren (mit und ohne Lokaler Suche) schneidet JPSO in Bezug auf Ergebnisqualität und Lösungsgeschwindigkeit deutlich besser ab. Es wird gezeigt, dass sich JPSO für komplexe kombinatorische Problemstellungen eignet.

4.14.3 Anpassung an hochflexibles Workforce Management

In diesem Kapitel wird die Anpassung von PSO an die drei Problemstellungen aus Logistik, Handel und Technikereinsatz bei der British Telecom dargestellt.

4.14.3.1 Untertägige Personaleinsatzplanung in der Logistik

Die Veränderungen des PSO-Algorithmus für kombinatorische Problemstellungen in Kapitel 4.14.2.2 lieferten viel versprechende Ergebnisse. An dem Konzept des JPSO [134], [310], [76] soll sich jedoch nur bedingt orientiert werden. Die Sprünge eines Partikels gehen beim JPSO immer nur in Richtung eines Anziehungspunktes – nicht zwangsläufig direkt auf diesen Punkt. Am Beispiel der Personaleinsatzplanung wird dieses Verhalten veranschaulicht: Zu einem bestimmten Zeitpunkt ist ein Mitarbeiter fälschlicherweise dem Arbeitsplatz zwei zugewiesen. Nur ein einziger Arbeitsplatz, Nummer neun, würde keine Fehlerpunkte erzeugen. Arbeitsplatz zwei und alle übrigen Arbeitsplätze verursachen jedoch Fehlerpunkte in gleicher Höhe. Daher springt das Partikel beim JPSO in Richtung Arbeitsplatz neun. Diese Sprünge führen jedoch zu keiner Verbesserung, da es völlig unerheblich ist, ob der Mitarbeiter dem Arbeitsplatz zwei, drei, ... oder acht zugewiesen ist. Lediglich Arbeitsplatz neun wäre korrekt. Mit Sprüngen besteht daher die Gefahr, dass Arbeitsplatz neun niemals erreicht wird, da die Sprünge vorher eingestellt werden können. Statt mehrerer Sprünge in Richtung Arbeitsplatz neun macht es deutlich mehr Sinn, den Mitarbeiter direkt auf Arbeitsplatz neun zu setzen.

Es zeigt sich, dass Anwendungen von PSO im Rahmen des Timetablings bisher wenig verbreitet sind. 2007 hat Poli die IEEE Xplore Datenbank für die thematische Gruppierung von PSO-Anwendungen analysiert [306]. Von den über 1.100 Veröffentlichungen zu PSO waren etwa 700 anwendungsorientiert, worunter nur sechs Prozent dem Scheduling zuzuordnen sind. Fokussiert man bei der Analyse von Poli stärker auf das Timetabling, so ist lediglich eine einzige Arbeit in diesem Gebiet anzusiedeln [69]. Genauer gesagt wurde mit PSO die Prüfungsorganisation untersucht. Der PSO-Algorithmus für das Logistikproblem soll ausgehend von den Modifikationen durch Chu u.a. bei der Prüfungsorganisation [69] und den darauf aufbauenden Arbeiten zur universitären Stundenplanerstellung von Brodersen und Schumann [47], [46] angepasst werden. Der Ablauf ist bei beiden Problemvarianten mit und ohne Stammarbeitsplatz identisch. In Algorithmus 19 ist der Ablauf für eine gBest-Topologie dargestellt.

In den Arbeiten [69] sowie [47] und [46] wird in jeder Iteration die neue Partikelposition mittels einer zuvor fest definierte Abfolge aus Tausch- und Kopieraktionen für zufällig ausgewählte Dimensionen ermittelt. Da das Verhalten des PSO-Algorithmus bei der Personaleinsatzplanung sehr stark von der Häufigkeit einzelner Aktionen pro Iteration abhängt, wird von dieser festen Abfolge einzelner Aktionen bei zufällig ermittelten Di-

Algorithmus 19 PSO für die untertägige Personaleinsatzplanung in der Logistik in einer gBest-Topologie

```

1: Initialisiere den Schwarm
2: Berechne die Fitness der Partikel
3: Bestimme pBest und gBest
4: repeat
5:   for i = 1 bis Anzahl Partikel do
6:     Berechne die neue Position des Partikels mit Hilfe von 4 Aktionen
7:     Repariere das Partikel
8:     Berechne die Fitness des Partikels
9:     if f(neue Position) < f(pBest) then
10:      pBest = neue Position
11:      Repariere pBest
12:      Berechne die Fitness von pBest
13:     end if
14:     if f(pBest) < f(gBest) then
15:      gBest = pBest
16:      Repariere gBest
17:      Berechne die Fitness von gBest
18:     end if
19:   end for
20: until Abbruchkriterium tritt ein

```

mensionen abgewichen. Der Ablauf wird so geändert, dass jetzt alle Dimensionen durchlaufen werden. Je Iteration wird für alle Dimensionen eines Partikels eine Eintrittswahrscheinlichkeit ermittelt, mit der eine der nachfolgend aufgeführten Aktionen ausgeführt wird. Die optimalen Eintrittswahrscheinlichkeiten wurden in Vortests bestimmt. Nachfolgend sind die Eintrittswahrscheinlichkeiten für das Problem Logistik_Complete und Logistik_S_Complete angegeben.

1. 9,7%: Keine Änderung der Partikeldimension. Der bereits zugewiesene Arbeitsplatz bleibt erhalten.
2. 0,3%: Zufällig wird ein Arbeitsplatz ermittelt und zugewiesen. Die Wahrscheinlichkeitsfunktion zur Ermittlung des Arbeitsplatzes ist gleichverteilt.
3. 30%: Der Partikeldimension wird der entsprechende Arbeitsplatzes aus \vec{p}_i zugewiesen. Somit wird die individuelle Komponente von PSO berücksichtigt.
4. 60%: Der Partikeldimension wird der entsprechende Arbeitsplatz aus \vec{p}_g (bzw. \vec{p}_i , wenn es sich nicht um eine gBest-Nachbarschaftstopologie handelt) zuweisen. Mit einer derartigen Zuweisung wird das soziale Verhalten des Schwarms einbezogen.

Inwieweit der Schwarm eher individuell oder sozial agiert, wird durch die Eintrittswahrscheinlichkeiten bestimmt, mit der ein Arbeitsplatz aus pBest oder gBest bzw. lBest zugewiesen wird. Erwähnenswert ist an dieser Stelle, dass die Eintrittswahrscheinlichkeiten stark von der Anzahl an Dimensionen abhängen. Bei einem kleineren Planungsproblem mit weniger Dimensionen müsste Aktion zwei deutlich höher gewichtet werden, um gute Ergebnisse zu erzielen. So beträgt bei den Tagesproblemen des Logistikdienstleisters die Wahrscheinlichkeit für Aktion zwei bereits 1,25% und für Aktion eins dementsprechend 8,85%. Die anderen beiden Wahrscheinlichkeiten bleiben bestehen.

Mit den dargestellten Modifikationen wurden die Charakteristika von PSO nicht verändert. Lediglich die Art, eine neue Partikelposition zu bestimmen, wurde angepasst. Daher entfällt die Festlegung von φ_1, φ_2 sowie eines Inertia Weights oder Constriction Factors. Weiterhin ist es durch die jetzige Form der Positionsermittlung nicht mehr erforderlich, etwaige Dimensionsüberschreitungen zu behandeln. Alle anderen Eigenheiten von PSO hinsichtlich des individuellen oder sozialen Verhaltens bleiben erhalten. Auch alle bei der kontinuierlichen Optimierung genannten Nachbarschaftstopologien können uneingeschränkt genutzt werden.

Initialisierung Für PSO wurden diverse Initialisierungsstrategien getestet, die schon bei Multistart HC (Kapitel 4.8.3.1) und TS (Kapitel 4.10.3.1) verwendet wurden.

In Tests hat sich gezeigt, dass Variante 1 (I1) am schlechtesten abschneidet. Der Plan enthält deutlich zu viele Arbeitsplatzwechsel, mit denen PSO Probleme hat. Variante 2 (I2) liefert bereits sehr gute Ergebnisse. Am besten schneidet jedoch Variante 3 (I3) als Startposition aller Partikel ab. Da diese Startlösung (der manuell erstellte Einsatzplan) im Vergleich zu den bisher gefundenen Bestwerten einen recht schlechten Fitnesswert hat, bewegen sich die Partikel sofort nach der Initialisierung in verschiedene Richtungen. Eine Ballung der Partikel im Lösungsraum tritt bereits nach kurzer Zeit nicht mehr auf. Zumal die Problemstellung recht viele Dimensionen hat.

Reparatur In Kapitel 4.12.4.1 wurde die Reparatur bei ES geschildert. Die gleiche Reparatur findet auch bei PSO für die jeweilige Problemvariante Anwendung. Im Algorithmus 19 fällt bezüglich der Einführung der Reparatur in PSO ein Unterschied zu ES auf. Bei ES wird die Reparatur bereits direkt nach der Initialisierung vorgenommen. Bei PSO musste die Reparatur an dieser Stelle entfallen. PSO initialisiert alle Partikel mit einer einheitlichen Startlösung, da die Endergebnisse dann besser sind als bei einer zufälligen Initialisierung. Eine Reparatur dieser Startlösung direkt nach der Initialisierung führte jedoch dazu, dass sich der Schwarm in den folgenden Iterationen nicht ausreichend über den Lösungsraum verteilt. Mit der hier vorgestellten Form (Reparatur erst innerhalb der For-Schleife) wird die Verschiedenartigkeit der Partikel gewahrt. Bei ES war das beschriebene Verhalten des PSO nicht zu beobachten, sodass direkt nach der

Initialisierung die Reparatur erfolgen kann.

Eine weitere Besonderheit liegt bei PSO darin, dass die Reparatur nicht nur nach jeder Neuberechnung der Partikelposition angestoßen wird. Immer dann, wenn ein neues pBest oder gBest gefunden wird, erfolgt eine erneute Reparatur von pBest bzw. gBest. Der Hintergrund ist folgender: Bei der Reparatur wird der Vektor der Entscheidungsvariable von links nach rechts durchlaufen. Nach einer erfolgten Reparatur ist es mit großer Wahrscheinlichkeit zu Änderungen in der Partikelposition gekommen. Wegen dieser Änderungen kann eine erneute Reparatur durchaus zu einer weiteren Verbesserung der Lösung führen, wenn auch nur zu einer sehr kleinen. Somit ist aber sichergestellt, dass pBest und gBest zum jeweiligen Zeitpunkt eine möglichst gute Lösung haben. Das Auftreten der zusätzlichen Reparaturen ist sehr gering – besonders, wenn die Optimierung weiter fortgeschritten ist. Daher kann der zusätzliche Rechenaufwand in Kauf genommen werden. Auf eine mehrfache Reparatur aller Partikel wurde aufgrund der dann deutlich erhöhten Rechenzeit verzichtet.

4.14.3.2 Automatische Arbeitszeitmodellerstellung im Handel

PSO für die automatische Arbeitszeitmodellerstellung im Handel läuft fast analog zur untertägigen Personaleinsatzplanung in der Logistik ab und kann Algorithmus 19 entnommen werden.

Es gibt jedoch Unterschiede in den Eintrittswahrscheinlichkeiten der vier Aktionen zur Berechnung der neuen Partikelposition, da die Eintrittswahrscheinlichkeit für Aktion zwei stark von der Anzahl an Dimensionen abhängt. Die Eintrittswahrscheinlichkeiten sehen wie folgt aus.

1. 9,98%: Keine Änderung der Partikeldimension. Der bereits zugewiesene Arbeitsplatz bleibt erhalten.
2. 0,02%: Zufällig wird ein Arbeitsplatz ermittelt und zugewiesen. Die Wahrscheinlichkeitsfunktion zur Ermittlung des Arbeitsplatzes ist gleichverteilt.
3. 30%: Der Partikeldimension wird der entsprechende Arbeitsplatzes aus \vec{p}_i zugewiesen. Somit wird die individuelle Komponente der PSO berücksichtigt.
4. 60%: Der Partikeldimension wird der entsprechende Arbeitsplatz aus \vec{p}_g (bzw. \vec{p}_l , wenn es sich nicht um eine gBest-Nachbarschaftstopologie handelt) zuweisen. Mit einer derartigen Zuweisung wird das soziale Verhalten des Schwarms gesteuert.

Die Initialisierung des Schwarms und die Reparatur der Partikel entspricht exakt dem Vorgehen bei ES (siehe Kapitel 4.12.4.2). Also eine zufällige Initialisierung nach Variante

1 (I1). Eine Reparatur der Partikel ist ebenfalls erforderlich, da die Endergebnisse sonst zu viele Verstöße im Bereich der Arbeitszeitmodelle enthalten.

Außer der veränderten Initialisierung, den geänderten Eintrittswahrscheinlichkeiten bei den Aktionen eins und zwei sowie der erweiterten Reparatur, gibt es keine Unterschiede zu PSO für das Logistikproblem.

4.14.3.3 Kombination von Touren- und Personaleinsatzplanung

Die Anpassungen des PSO-Algorithmus wurden zuvor am Logistik- und am Handelsproblem dargestellt. Das British Telecom-Problem erfordert auch bei PSO aufgrund der veränderten Abbildung der Problemstellung und des anders strukturierten Lösungsraums einige Modifikationen im Vergleich zu PSO beim Logistik- und Handelsproblem. Der Ablauf wird nachfolgend erläutert, wobei die Initialisierung der Population und die erforderliche Reparatur der Qualifikationsfehler wie bei ES ablaufen (siehe Kapitel 4.12.4.3). Zur Initialisierung wird also ebenfalls KV 3 aus Kapitel 4.6.3.2 genutzt.

Um zu verhindern, dass der PSO-Algorithmus vorzeitig in einem lokalen Optimum konvergiert, existieren verschiedenen Möglichkeiten. Eine Möglichkeit liegt in der Wahl der Nachbarschaftstopologie. Eine andere besteht darin, jedes Partikel mit der Fähigkeit auszustatten bei der Bewegung seine neue Position zu betrachten und darauf hin zu entscheiden, ob diese Position lohnend ist oder nicht. Positionen, die zu Verschlechterungen führen, können abgelehnt werden. Dies ist sinnvoll, da der PSO in Experimenten in Regionen vorgedrungen ist, die zu immer weiteren Verschlechterungen führten. Nach einigen Iterationen war das Partikel derart verändert, dass ein Entkommen aus dem lokalen Optimum verwehrt blieb. Der PSO wurde daher modifiziert. Die Partikel kommunizieren nun nicht mehr nur miteinander. Sie können nun auch „sehen“. Mit einer geringen Wahrscheinlichkeit nimmt das Partikel jedoch auch Verschlechterungen in Kauf, um zu verhindern, dass es vollends in einem lokalen Optimum festsetzt und keine Bewegungen im Lösungsraum ausführt. Der Ablauf für eine gBest-Topologie ist in Algorithmus 20 dargestellt. Dieser Ansatz ist neuartig und wurde bisher nicht in der Literatur gefunden.

Bestimmung der neuen Partikelposition Der Ablauf des PSO-Algorithmus beim British Telecom-Problem unterscheidet sich nicht wesentlich vom Ablauf beim Logistik- und Handelsproblem. Der Unterschied liegt in der Einführung einer Schleife mit dem Index w . Dies ist notwendig, da im Gegensatz zum Logistik- und Handelsproblem die Matrix eines Partikels nicht mehr systematisch durchlaufen wird aber dennoch mehrere Änderungen je Iteration ausgeführt werden sollen. Für den Index w wurde in Tests der Wert 300 ermittelt. Tatsächlich werden jedoch keine 300 Änderungen vorgenommen, da diese vom Partikel z.T. nicht angenommen oder aufgrund fehlender Qualifikationen

Algorithmus 20 PSO für die Kombination von Touren- und Personaleinsatzplanung in einer gBest-Topologie

```

1: Initialisiere den Schwarm
2: Berechne die Fitness der Partikel
3: Bestimme pBest und gBest
4: repeat
5:   for i = 1 bis Anzahl Partikel do
6:     for w = 1 bis 300 do
7:       Berechne die neue Position des Partikels mit Hilfe von 6
         Aktionen
8:     end for
9:     Repariere das Partikel
10:    Berechne die Fitness des Partikels
11:    if f(neue Position) < f(pBest) then
12:      pBest = neue Position
13:    end if
14:    if f(pBest) < f(gBest) then
15:      gBest = pBest
16:    end if
17:  end for
18: until Abbruchkriterium tritt ein

```

nicht durchgeführt werden. Ebenso kann es beim Kopieren eines Wertes aus pBest oder gBest vorkommen, dass dies zu keiner Änderung in der Partikelposition führt, da dieser Wert bereits an der entsprechenden Position vorhanden war. Eine weitere Modifikation von PSO betrifft die Berechnung der neuen Partikelposition. Es gibt nun die sechs folgenden Aktionen, von denen pro Durchlauf der Schleife mit dem Index w jeweils eine zufällig ausgewählt wird. Im Gegensatz zum Logistik- und Handelsproblem wird die Parametrierung des PSO-Algorithmus ein wenig komplexer, da die Eintrittswahrscheinlichkeiten für die verschiedenen Bewegungsaktionen mit aufwändigeren Tests bestimmt werden müssen.

1. 0,05%: Vertausche zwei Auftragszuweisungen, ohne diesen Schritt abzulehnen. Es werden zwei zufällige Aufträge ausgewählt und die aktuell zugewiesenen Techniker gesucht. Die Techniker tauschen anschließend die Auftragszuweisungen aus, wobei der neue Auftrag in der Permutation des Technikers jeweils an die gleiche Stelle des alten Auftrags geschrieben wird. Qualifikationen werden bei dieser Aktion berücksichtigt. Dieser Schritt darf vom Partikel auch bei Verschlechterung der Fitness nicht abgelehnt werden.
2. 24,95%: Vertausche zwei Auftragszuweisungen, wobei dieser Schritt abgelehnt werden darf. Der Ablauf ist analog zu Schritt eins, mit dem Unterschied, dass das Partikel diesen Schritt nicht ausführt, wenn sich dadurch die Fitness verschlechtert.

3. 0,25%: Verschiebe den letzten Auftrag eines Technikers an das Ende der Permutation eines anderen Technikers, ohne diesen Schritt abzulehnen. Es wird ein Techniker zufällig ausgewählt. Für dessen letzten Auftrag in der Permutation wird ein qualifizierter Mitarbeiter gesucht. Ist dieser gefunden, wird ihm der Auftrag als letzter Auftrag seiner Permutation zugewiesen. Wird kein qualifizierter Techniker gefunden, passiert nichts. Dieser Schritt darf vom Partikel auch bei Verschlechterung nicht abgelehnt werden.
4. 14,75%: Verschiebe den letzten Auftrag eines Technikers an das Ende der Permutation eines anderen Technikers, wobei dieser Schritt abgelehnt werden darf. Der Ablauf ist analog zu Schritt drei, mit dem Unterschied, dass das Partikel diesen Schritt nicht ausführt, wenn sich dadurch die Fitness verschlechtert.
5. 20%: Füge einen Wert aus pBest ein, wobei dieser Schritt abgelehnt werden darf. Wähle aus pBest einen zufälligen Techniker und einen zufälligen Auftrag seiner Permutation. Füge diesen Auftrag in die neue Partikelposition an gleicher Stelle wie in pBest ein. Befindet sich dort bereits ein Auftrag, so verschiebe den kompletten Rest der Permutation um einen Slot nach rechts (also später), um Platz zu schaffen. Sollte der Auftrag in der Permutation isoliert sein, so verschiebe ihn so lange nach links, bis er an einen Slot mit einem Auftrag angrenzt. Dieser Schritt kann vom Partikel bei Verschlechterung der Fitness abgelehnt werden.
6. 40%: Füge einen Wert aus gBest ein, wobei dieser Schritt abgelehnt werden darf. Analoges Vorgehen zu Schritt fünf. Mit dem Unterschied, dass es sich hier um einen Wert aus gBest (bzw. lBest in einer anderen Nachbarschaftstopologie) statt aus pBest handelt.

Im Gegensatz zum Logistik- und Handelsproblem erfolgt hier nur einmal eine Reparatur. Eine erneute Reparatur, wenn pBest oder gBest gefunden wurde ist nicht erforderlich, da bereits alle Qualifikationsfehler durch die einmalige Reparatur behoben sind.

4.15 Multiagentensysteme

Multiagentensysteme (MAS) sind neben klassischen OR-Verfahren und Metaheuristiken die dritte Gruppe an Lösungsverfahren, die in dieser Arbeit untersucht werden. Sie werden an allen drei Problemstellungen mit eigenen Implementierungen untersucht. Beim British Telecom-Problem kann zusätzlich auf die Arbeiten von Kokkoras [217], von Kokkoras und Gregory [218] sowie von Sakellariou, Kokkoras und Vlahavas [330] zurück gegriffen werden. Die entsprechenden Ausführungen sind in Kapitel 4.15.3 zu finden. Zuvor wird in Kapitel 4.15.1 auf den grundlegenden Ablauf von MAS sowie in Kapitel 4.15.2 auf Anwendungen in der Literatur eingegangen.

4.15.1 Grundlegender Ablauf

Im Gegensatz zu Optimierungsverfahren, wie z.B. TS, ES oder PSO, die einen zentralen Problemlösungsansatz verfolgen, beruht der Lösungsansatz bei MAS auf Dezentralisierung. Hierzu bedient man sich autonomer Agenten, die mit ihrer Umwelt interagieren und kommunizieren. Es existiert keine allgemein akzeptierte Definition, was ein Agent ist [416]. Nach Wooldridge ist ein Agent ein Computersystem, welches zu autonomen Handlungen fähig ist, um seine Ziele zu erreichen [422, S. 15]. Wooldridge und Jennings charakterisieren Agenten mit folgenden Eigenschaften [423, S. 118].

- **Autonomie:** Agenten handeln ohne direkte Beeinflussung von Menschen oder anderen Entitäten. Sie haben eine gewisse Kontrolle über ihr Handeln und ihren Zustand.
- **Soziale Fähigkeit:** Agenten interagieren miteinander oder mit Menschen über eine Kommunikationssprache.
- **Reaktivität:** Agenten nehmen ihre Umwelt wahr und reagieren zeitnah auf Veränderungen der Umwelt.
- **Proaktivität:** Agenten reagieren nicht nur auf Veränderungen in ihrer Umwelt, sie sind auch in der Lage, zielgerichtet zu handeln und die Initiative zu ergreifen.

Diese Charakterisierung ist recht einfach und wird von zahlreichen Autoren erweitert. So z.B. um die Eigenschaften Lernfähigkeit und Mobilität, Zusammenarbeit, Kontinuität, Persönlichkeit sowie der Fähigkeit, Schlussfolgerungen anzustellen. All diese Eigenschaften können – müssen aber nicht – bei einem Agenten auftreten.

MAS bestehen aus mehreren Agenten, wobei deren Anzahl je nach Problemstellung stark variieren kann. MAS sind ein Forschungsgebiet der verteilten künstlichen Intelligenz (DAI = Distributed Artificial Intelligence). Nach Weiss ist DAI „[...] the study, construction, and application of multiagent systems, that is, systems in which several interacting, intelligent agents pursue some set of goals or perform some set of tasks.“ [416, S. 1]. Vlasis erläutert die Unterschiede zwischen einem MAS und Systemen mit einem Agenten [397, S. 1–3]. In diesem Zusammenhang identifiziert er die Kriterien Aufbau, Umwelt, Wahrnehmung, Steuerung, Wissen und Kommunikation.

Nach Kirn, Herzog und Lockemann bestehen MAS aus heterogenen Agenten, die proaktiv ihre eigene Ziele verfolgen oder gemeinsam mit anderen Agenten Aufgaben erledigen. Dabei gibt es keine zentrale Steuerung. Vielmehr ist Interaktion zwischen den Agenten erforderlich, da die Wahrnehmung der Umwelt dezentral erfolgt, das Problem in Teilprobleme zerlegt ist und sich Lösungen überschneiden können. Die Ausgestaltung der Eigenschaften beschreiben sie wie folgt: „The properties of MAS appear to be almost

infinite regarding the number of agents, their flexibility, the ways of interaction, the initial state of the system, etc.“ [214, S. 3]. Bzgl. des grundlegenden Ablaufs eines MAS gibt es kein einheitliches Vorgehen. Müller stellt 1996 vier verschiedene Architekturvarianten für Agenten dar: beratende, reaktive, interagierende und hybride Agenten [274, S. 17–41], [296, S. 28–30].

4.15.2 Anwendungen in der Literatur

Puppe u.a. bearbeiten 2000 [315] zwei Schedulingprobleme im Krankenhaus. Zum einen die Dienstplanung des Pflegepersonals und zum anderen die Untersuchungs- bzw. Behandlungsplanung. Ziel der Arbeit ist es nicht, die bestmögliche Lösung zu finden, sondern verschiedene Schedulingstrategien zu vergleichen, die in der Praxis schnell übernommen werden können. Es werden zwei Agentenkonzepte vorgestellt. Die patientenorientierte Sichtweise [256]: Je Patientenuntersuchung wird ein eigener Agent kreiert. Und die ressourcenorientierte Sichtweise [95]: Die Ressource selbst bzw. die für sie verantwortliche Organisationseinheit existiert als Agent. Der Ressourcenansatz wird eher bei bekannten Rahmenbedingungen angewendet, wogegen der Patientenansatz eher bei der dynamischen Planung Anwendung findet. Für die Dienstplanung bietet sich der Ressourcenansatz an. Es existieren Agenten, die dieselben Aufgaben übernehmen können. Es können auch Agenten existieren, die gleichartige (daher austauschbare) Ressourcen zur Verfügung haben. Wird einem Agenten eine Aufgabe übertragen, die er nur unter Verletzung von Restriktionen erfüllen kann, so versucht der Agent, diese Aufgabe auf einen anderen Agenten zu übertragen. Ebenso kann ein Agent noch freie Ressourcen anbieten. Bei der Untersuchungs- und Behandlungsplanung findet der Patientenansatz Anwendung. Patienten, Personal und Organisationseinheiten sind die Agenten. Ein Agent kann Ressourcen besitzen oder Ressourcen für die Aufgabenerfüllung benötigen. Jeder Agent besitzt einen Stundenplan, der die übernommenen Aufgaben und die Verfügbarkeiten enthält. Kann ein Agent die Situation nicht lösen, weil ein Terminkonflikt besteht, so kann er bei der zuständigen Schedulinginstanz um eine Terminänderung bitten. Aufgaben können jedoch nicht wie bei der Dienstplanung unter verschiedenen Schedulinginstanzen (Organisationseinheiten) ausgetauscht werden, da die Aufgaben nicht gleichartig sind. Denn bei der Dienstplanung werden Schedules für gleichartige Aufgaben und Ressourcen erstellt. Puppe u.a. stellen Schedulingagenten mit einfacher Verhandlungsfähigkeit und Agenten mit rückgekoppelter Verhandlungsfähigkeit vor. Für die Dienstplanung eignen sich erstgenannte Agenten. Zudem fordern Puppe u.a. die Trennung von Problem (Art), Lösungsverfahren und Probleminstanz.

Ebenfalls für die Dienstplanung im Krankenhaus nutzt Pfohl 2000 [302] einen agentenbasierten elektronischen Markt. Mit diesem Ansatz soll die Akzeptanz des automatisch erstellten Dienstplanes bei den Mitarbeitern gesteigert werden. Das bisherige Planungssystem soll jedoch nicht völlig ersetzt werden. Vielmehr wird in zwei Phasen vorgegangen. In Phase eins wird mit dem bisherigen System der Sollplan erzeugt, der jedoch noch

Lücken enthält. Diese Lücken werden in der zweiten Phase mit Hilfe eines agentenbasierten Marktes geschlossen, sodass dann ein vollständiger Plan entsteht. Dieses Konzept wurde beispielhaft in Java implementiert.

Eine Terminplanung in einem Akutkrankenhaus unter Berücksichtigung der Arbeitsabläufe und der Personalstruktur wird 2002 von Krempels mit Hilfe von Agenten vorgenommen [228]. Das System wurde mit JADE (Java Agent Development Framework) implementiert. Zur Lösung des Problems wird in zwei Phasen vorgegangen. Zunächst erstellt der Planer einen Terminplan, wobei er keine Präferenzen berücksichtigt. Anschließend versucht der Planer das Ergebnis unter Berücksichtigung der Präferenzen zu verbessern. Die zweite Phase ist so gestaltet, dass eine Wissensbasis die relevanten Eigenschaften der Ressourcen enthält. Ein Expertensystem, in dem sich der Planer befindet, nutzt diese Wissensbasis. Findet der Scheduler einen Konflikt, wird für jeden der betroffenen Mitarbeiter ein Agent im Agentensystem gestartet und mit allen relevanten Informationen initialisiert. Anschließend startet die Verhandlung zwischen den Agenten. Der Planer wird über das Ergebnis der Verhandlung (immer genau ein gewinnender Agent) informiert, sodass eine Änderung im Plan vorgenommen werden kann. Es kommen präferenzbasierte Agenten (Policy-Agenten) zum Einsatz, wobei Krempels besonders auf die Abbildung der Präferenzen und der hierarchischen Strukturen achtet. So gibt es für jeden Mitarbeiter eine Gewichtsmasse, die die Stellung in der Organisation repräsentiert. Ein Oberarzt hat eine größere Gewichtsmasse als eine Krankenschwester. Mit der Verteilung der Gewichtsmasse auf die Termine können diese gewichtet werden.

De Causmaecker, Ouelhadj und Vanden Berghe gehen 2003 [93] auf Verhandlungsschemata beim Course Timetabling ein. Sie fordern, dass beim Informationsaustausch zwischen Agenten nicht alle möglichen Informationen ausgetauscht werden. Als Beispiel wird ein Lehrer genannt, der Gründe für eine Präferenz hat – diese Gründe jedoch nicht kommuniziert. Zudem darf der Verhandlungsprozess nicht unbegrenzt lange dauern und nach Ablauf der Frist muss eine Entscheidung vorliegen. Gelegentlich gibt es die Regel, dass bei jeder Anfrage sofort eine Entscheidung vorliegen muss. Wichtig sind bei Verhandlungen im Timetabling Dinge wie Fairness, Kontinuität und Effizienz. In einer globalen Zielfunktion müssen diese Dinge messbar sein, um sie zu berücksichtigen. De Causmaecker, Ouelhadj und Vanden Berghe verwenden ein Maß der Sympathie. Ein alternatives Vorgehen würde beaufsichtigende Agenten vorsehen, die für Fairness etc. sorgen. Dies würde jedoch dem grundlegenden Prinzip eines verteilten Systems widersprechen.

Im gleichen Jahr wenden sich Meisels und Kaplansky [259] verteilten Timetabling Problemen zu. In einer späteren Veröffentlichung [201] sind ein wenig umfangreichere Ausführungen zu finden. Meisels und Kaplansky stellen ein Modell vor, in dem es einen zentralen Agenten (Central Agent) gibt. Dieser koordiniert den Suchprozess aller Planungsagenten (Scheduling Agent) – ist also eine Art beaufsichtigender Agent, deren Einsatz de Causmaecker, Ouelhadj und Vanden Berghe ablehnen [93]. Das Modell wurde mit verschiedenen Konfigurationen an Problemen für die Dienstplanung im Krankenhaus

erprobt. Dabei wurde vor allem der zentrale Agent untersucht. Die Wahl des Suchverfahrens hat einen entscheidenden Einfluss auf die Laufzeit. Die besten Ergebnisse wurde bei einem zentralen Agenten mit Lokaler Suche und einem dynamischen Verhalten erzielt. Dort werden Nebenbedingungen dynamisch eingefügt und gelöscht.

Ausgehend von ihrer Arbeit in 2003 beschreiben de Causmaecker u.a. 2005 [92] ein agentenbasiertes Lösungsverfahren für das Personnel Scheduling. Die Umsetzung erfolgt mit der Programmiersprache Oz. Grundsätzlich wird angestrebt, die realen Zusammenhänge möglichst genau abzubilden. Die Delegation von Aufgaben dorthin, wo sie anfallen, ist eine Management Strategie. So kann dem Flaschenhals einer zentralen Planung begegnet werden, da die dezentralen Entscheidungseinheiten als agile Agenten handeln und schnell reagieren. Es werden zwei Probleme bei der Erstellung der Anwendung aufgeführt. Zunächst muss eine entsprechende Architektur für das Modell erstellt werden. Dabei werden nicht alle Informationen ausgetauscht, sondern nur Teilinformationen bzw. Variablen. Die Definition einer gemeinsamen Sprache ist wichtig, da die Agenten einen erheblichen Kommunikationsbedarf haben (mehr Kommunikation als bei einem zentralen Ansatz). Als zweites müssen Optimierungsverfahren für die Agenten und ein Verhandlungsprotokoll gefunden werden. Das Verhandlungsprotokoll enthält Deadlines, die die Algorithmen in der Entscheidungsfreiheit einschränken, um eine gute Performance zu gewährleisten.

In den bisherigen Arbeiten wurden Agenten lediglich für einzelne Problemstellungen eingesetzt. Chiu, Cheung und Leung stellen in 2005 [68] eine Multi-Agent Information System (MAIS) für Mobile Workforce Management (MWM) vor. Der WFM Prozess besteht bei ihnen aus den sechs folgenden Phasen, die meist umfangreichen Problemstellungen entsprechen. Es werden folgende Agentencluster gebildet, die aus verschiedenen Agenten bestehen, um die Ziele der sechs Phasen zu erreichen:

- Task Formulation Agent Cluster: Erzeugung von Aufträgen. Es gibt unzählige Quellen für Aufträge (Call Center, Internet, Wartungsarbeiten, CRM-Systeme, ...). Agenten fassen alle Informationen zusammen, ergänzen fehlende Informationen und bewerten die Aufträge in ihrer Dringlichkeit.
- Matchmaking Agent Cluster: Agenten identifizieren alle möglichen Mitarbeiter für eine Aufgabe. Sie bilden die Reihenfolge der Mitarbeiter. Anschließend wird eine Shortlist von Mitarbeitern erzeugt.
- Broker Agent Cluster: Agenten verhandeln mit der Shortlist von Mitarbeitern, um den besten Mitarbeiter auszuwählen.
- Commuting Agent Cluster: Wenn der Mitarbeiter anreisen muss, wird dieser Cluster aktiv, ansonsten wird zum nächsten Cluster gesprungen. Agenten berücksichtigen den Standort des Mitarbeiters, mögliche Verkehrsmittel und ermitteln die Anfahrt.

- **Service Support Agent Cluster:** Agenten in diesem Cluster unterstützen die Mitarbeiter in der Durchführung des Auftrags. Zum Beispiel die Koordination, wenn von mehreren Mitarbeitern an demselben Dokument gearbeitet wird. Zudem sind Agenten für den sicheren Zugang zu zentralen Systemen zuständig sowie für das Monitoring der Aktivitäten, sodass bei Verzögerungen schnell reagiert werden kann.
- **Workforce and User Agent Cluster:** Dieser Cluster ist erforderlich, um Mitarbeiter in ihrer Arbeit von Agenten zu unterstützen. Es gibt Agenten mit Erinnerungs- oder Alarmfunktion, es werden Wünsche unterstützt und Datenformate zur Kommunikation transformiert.

Gershman u.a. [140] nutzen in 2008 Agenten zur Meetingplanung. Dabei dienen die Agenten dazu, den Konflikt zwischen globalen Zielen (z.B. möglichst viele Meetings) und individuellen Zielen (z.B. Terminpräferenzen) zu entschärfen. Zeit ist bei der Problemstellung ein kritischer Faktor, da die Meetingplanung online erfolgen soll. Experimente mit je zehn Problemstellungen werden an drei verschiedenen Intervalllängen vorgenommen. Der Agentenansatz, der ein lokales Suchverfahren nutzt, liegt sehr nahe an den Optimalwerten, die B&B liefert.

Wie bereits erwähnt, wurden an der Problemstellung der British Telecom bereits ein Agentenansatz getestet. Hierbei handelt es sich um den Ansatz von Kokkoras u.a. [217], [218], [330], in dem CLP eingesetzt wird. Der Ansatz wird in Kapitel 4.15.3.4 eingehend erläutert.

In Kapitel 3.4.5 wurden zudem die Arbeiten von Tsang u.a. [379], [380], [382] vorgestellt, in denen sie MAS an zufälligen Instanzen des British Telecom-Problems mit dynamischen Änderungen untersuchen.

4.15.3 Anpassung an hochflexibles Workforce Management

In diesem Kapitel werden die selbst entwickelten Agentenansätze für das Logistik-, Handels- und British Telecom-Problem beschrieben. Bei letzterer Problemstellung wird zusätzlich der Agentenansatz von Kokkoras u.a. [217], [218], [330] erläutert.

4.15.3.1 Vorüberlegungen

Zentrale Ansätze zum Scheduling sind laut Puppe u.a. [315] schwer anwendbar. Zum einen ist Scheduling äußerst komplex und in der Unternehmensorganisation gibt es viele Verantwortlichkeitsstrukturen. Daher verteilen sich die Schedulingaufgaben auf ver-

schiedene Organisationseinheiten, welche Wechselwirkungen untereinander haben. Dies erschwert ein zentrales Scheduling. Zudem gibt es verschiedene Akteure, die z.T. konfliktäre Ziele haben. U.U. können die individuellen Ziele auch vom Organisationsziel abweichen. Die Abbildung der Ziele in einer zentralen Schedulingstrategie kann laut Puppe u.a. schwer sein.

Bei den hier untersuchten Problemstellungen sind viele sich gegenseitig behindernde Regelungen zu berücksichtigen. Im Wesentlichen gehen diese Regelungen von zwei Seiten aus. Zum einen ist dies der Arbeitgeber, der die anfallende Arbeit bestmöglich erledigt haben möchte. Dies bewirkt, dass eine Reduzierung der Personalkosten, ein passgenauer Mitarbeiterereinsatz, ein hoher Servicelevel, etc. angestrebt wird. Auf der anderen Seite ist die Gruppe der Arbeitnehmer, die ihre Rechte vertreten. Es gibt z.B. diverse gesetzliche und tarifliche Regelungen. So möchten Mitarbeiter z.B. nur dann arbeiten, wenn sie auch verfügbar sind und sie haben Regelungen zu minimalen und maximalen Arbeitszeiten sowie zum Umgang mit untertägigen Arbeitsplatzwechseln.

Vor diesem Hintergrund soll ein Agentensystem zur Personaleinsatzplanung auch diese beiden Gruppen in einem Planungsprozess abbilden. Dieser soll sich möglichst stark an der Realität orientieren. Zunächst sollen Abteilungsleiter unabhängig voneinander Mitarbeiter aus einem Pool verplanen. Etwaige Ineffizienzen in Form von nicht verplanten Mitarbeitern werden anschließend korrigiert. In der zweiten Phase wird den Mitarbeitern der Einsatzplan zugestellt, worauf hin sie versuchen, ihre Rechte durchzusetzen. Ein ähnlicher zweiphasiger Ablauf wurde von Krempels [228] vorgestellt. Dort gibt es in einer Verhandlung immer nur einen Gewinner und keine Kompromisse. Dies soll auch im hier dargestellten Ansatz umgesetzt werden. Eine Verhandlung muss zudem nach endlicher Zeit abgeschlossen sein [93], [92]. Für den hier verwendeten Agentenansatz gilt die Forderung, dass ein Agent sofort eine Antwort auf eine Anfrage erhält. Eine weitere Forderung, die auch von de Causmaecker u.a. [93], [92] gemacht wird, besteht darin, dass Agenten nur die notwendigen Informationen austauschen. Es wird von einem Mitarbeiter also nur eine Anfrage nach einem Tausch gestellt, ohne diese zu begründen oder andere nicht zwingend benötigte Informationen mitzugeben.

Überträgt man die Aussagen von Puppe u.a. [315] zu ihren beiden vorgestellten Agentenansätzen auf die vorliegenden Problemstellungen, so würden beim Ressourcenansatz Arbeitsplätze bzw. Abteilungsleiter als Agent existieren und Mitarbeiter verplanen. Zudem würden Mitarbeiter als Agenten existieren, auf die Aufgaben übertragen werden können. Dabei treten Mitarbeiter in Verhandlung, wenn es zu Konflikten kommt. Der Patientenansatz würde bei den dargestellten Problemstellungen so aussehen, dass jede Aufgabe – also jedes geforderte Vollzeitäquivalent pro Zeitintervall – einen Agenten darstellt, der einen Mitarbeiter benötigt. Zudem würden Mitarbeiter als Agenten existieren, die einer Aufgabe zugeteilt werden können. In dieser Arbeit wurde sich für den Ressourcenansatz entschieden, da dieser sich besonders für nicht veränderliche Rahmenbedingungen eignet. Zudem führt der Patientenansatz bei Problemstellungen, die einen Zeithorizont von einem Kalenderjahr umfassen, zu einer zu großen Anzahl an Agenten.

In einer Forschungsfrage dieser Arbeit soll am Beispiel der drei ausgewählten Problemstellungen geklärt werden, ob sich eher zentrale oder dezentrale Lösungsansätze für hochflexibles WFM eignen. Wie bereits erwähnt sind laut Puppe u.a. [315] bei der Dienstplanung nämlich eher dezentrale Ansätze zu bevorzugen. Diese werden nachfolgend für die drei Problemstellungen dieser Arbeit erläutert. Dabei basieren die eigenen MAS auf der Konstruktion von Lösungen. Der Einsatz von PSO, ES oder TS macht beim gewählten Vorgehen der MAS keinen Sinn. Für eine detaillierte Begründung sei auf Kapitel 4.16 verwiesen. Lediglich für das British Telecom-Problem wurde CLP von anderen Autoren innerhalb eines MAS erstellt.

4.15.3.2 Untertägige Personaleinsatzplanung in der Logistik

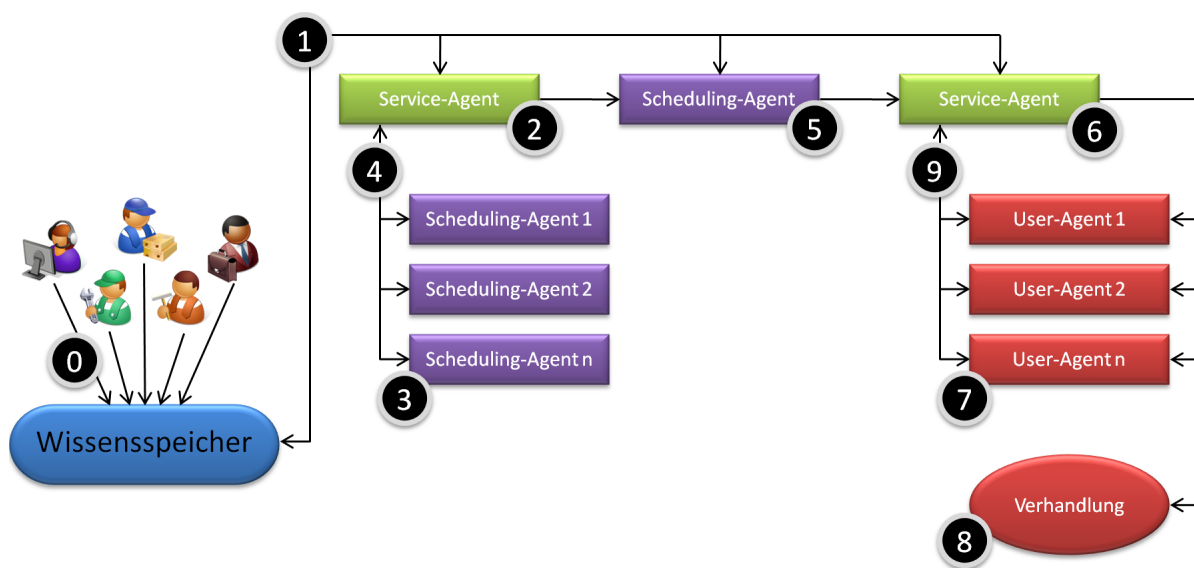


Abbildung 4.23: Ablauf des Agentenansatzes für die untertägige Personaleinsatzplanung in der Logistik

In Abbildung 4.23 wird der Agentenansatz für das Logistikproblem schematisch dargestellt. Dieser wird nachfolgend für das Problem ohne Stammarbeitsplatzregelung beschrieben. Im Anschluss an diese Ausführungen wird auf einige Modifikationen für zu berücksichtigende Stammarbeitsplätze eingegangen.

- Zunächst werden die Eigenschaften der existierenden Ressourcen, die aktuellen Anforderungen und die Rahmenbedingungen der Problemstellung in der Wissensbasis abgelegt (0). Hierzu gehören An-/Abwesenheiten der Mitarbeiter, Qualifikationen der Mitarbeiter, erforderliche Qualifikationen der Arbeitsplätze, Fehlerpunkthöhen für den Verstoß gegen Restriktionen, Länge der Planungsintervalle, Personalbedarfe je Zeitintervall, etc.

- Die Informationen des Wissensspeichers werden an die drei Agenten (2), (5) und (6) geliefert (1). Zudem erfolgen Rückmeldungen von den drei Agenten an den Wissensspeicher.
- Zu Beginn der Planung initialisiert Service-Agent (2) den Einsatzplan, indem alle Mitarbeiter auf einen Dummy-Arbeitsplatz geplant werden. Dieser kennzeichnet, dass der jeweilige Mitarbeiter noch keinem realen Arbeitsplatz zugeordnet wurde. Die Zuweisung auf den Dummy-Arbeitsplatz erfolgt nur, wenn der Mitarbeiter im jeweiligen Zeitintervall anwesend ist.
- Anschließend ermittelt Service-Agent (2) das Ranking der neun Arbeitsplätze. Arbeitsplätze, für die nur wenige Mitarbeiter qualifiziert sind, erhalten die höchste Priorität. Sollten für zwei oder mehrere Arbeitsplätze die gleiche Anzahl an Mitarbeitern zur Verfügung stehen, wird zufällig entschieden, welcher Arbeitsplatz die höhere Priorität erhält.
- Entsprechend der Priorität (höchste zuerst) werden die Scheduling-Agenten (3) nacheinander vom Service-Agenten (2) initialisiert. Jeder Scheduling-Agent (3) entspricht einem der neun Arbeitsplätze. Der Scheduling-Agent, für den die wenigsten Mitarbeiter qualifiziert sind, beginnt und plant qualifizierte Mitarbeiter ein, die anwesend sind und noch nicht verplant wurden. Dies geschieht mit dem Fokus auf die exakte Deckung des Personalbedarfs. Über- und Unterdeckung sollen minimiert werden, soweit dies möglich ist. Das Planungsergebnis des ersten Scheduling-Agenten wird dem Service-Agenten (2) mitgeteilt (4), der wiederum die Rückmeldung über den Plan an den Wissensspeicher gibt (1). Anschließend startet der Service-Agent (2) den nächsten Scheduling-Agenten (3), der nun ebenfalls versucht, den Personalbedarf bestmöglich zu decken. Dabei darf er nicht auf bereits verplante Mitarbeiter zugreifen. Der Service-Agent (2) startet so lange nacheinander Scheduling-Agenten (3), bis alle neun Arbeitsplätze abgearbeitet sind.
- Nachdem für alle Arbeitsplätze je ein Scheduling-Agent (3) die Einsatzplanung vorgenommen hat und diese an den Service-Agenten (2) gemeldet hat, können immer noch Mitarbeiter in einigen Zeitintervallen nicht verplant sein. Zudem können Umbesetzungen der Mitarbeiter auf andere Arbeitsplätze zu einer besseren Bedarfsdeckung führen. Daher wird nun vom Service-Agenten (2) ein Scheduling-Agent (5) aufgerufen, der ebenfalls mit der Wissensbasis verbunden ist (1). Scheduling-Agent (5) finalisiert nun den Einsatzplan. Also die Verplanung aller noch nicht zugewiesener Mitarbeiter, wobei zwangsläufig Überdeckungen in Kauf genommen werden. Zudem wird geprüft, ob Umbesetzungen von Mitarbeitern zu einer besseren Bedarfsdeckung führen, was entsprechend vollzogen wird.
- Die bisherige Einsatzplanung ist bisher aus der Unternehmenssicht erfolgt: Einhaltung der harten Nebenbedingungen und ein möglichst passgenauer Personaleinsatz. Dies geschah unter Vernachlässigung der Mitarbeiterbelange, in diesem Fall

unter Vernachlässigung der Reduzierung von Arbeitsplatzwechseln je Mitarbeiter. Daher ruft Scheduling-Agent (5) nach Erstellung des Einsatzplans den Service-Agenten (6) auf, um die Mitarbeiterbelange in der Planung zu berücksichtigen. Service-Agent (6) ist ebenfalls mit der Wissensbasis verbunden (1).

- Service-Agent (6) untersucht jeden Zeitintervall des Einsatzplans und prüft für jeden Mitarbeiter, ob ein Arbeitsplatzwechsel vorliegt. Trifft dies zu, so werden für den entsprechenden Zeitintervall alle Mitarbeiter identifiziert, die für die Verhandlung in Frage kommen. Ein Mitarbeiter kommt dann für die Verhandlung in Frage, wenn er in dem Zeitintervall anwesend ist und für einen möglichen Tausch qualifiziert ist. Service-Agent (6) erzeugt für die relevanten Mitarbeiter je einen User-Agenten (7). Im Gegensatz zu den Scheduling-Agenten (3) existieren die erzeugten User-Agenten (7) zeitgleich.
- Die User-Agenten (7) verhandeln nun über einen Tausch der Arbeitsplatzzuweisung (8). Dies läuft folgendermaßen ab: Der fehlerhafte User-Agent (bei ihm hat der Service-Agent (6) den Arbeitsplatzwechsel gefunden) fragt nacheinander bei allen anderen User-Agenten an, ob sie tauschen wollen. Der User-Agent, bei dem aktuell die Anfrage liegt und der fehlerhafte User-Agent, kennen jeweils ihre aktuellen Arbeitsplatzzuweisungen zum Zeitpunkt t und die Zuweisungen in $t-1$ und $t+1$. Sie tauschen lediglich die Information zu ihrer Arbeitsplatzzuweisung im Zeitpunkt t aus und können so ohne Berechnung der Fitnessfunktion entscheiden, ob ein Tausch zu weniger Fehlerpunkten führen würde. Wenn dem so ist, stimmt sie einem Tausch zu und teilt dies dem Service-Agenten (6) mit (9), woraufhin der Tausch vollzogen wird und alle User-Agenten gelöscht werden. Andernfalls fragt der fehlerhafte User-Agent beim nächsten User-Agenten an, bis alle abgefragt wurden. Als Ergebnis findet demnach entweder ein Tausch zwischen zwei der User-Agenten statt oder es findet kein Tausch statt, wenn sich das Planungsergebnis verschlechtern würde.
- Service-Agent (6) untersucht wiederum jeden Zeitintervall des Einsatzplans und prüft für jeden Mitarbeiter, ob ein Arbeitsplatzwechsel vorliegt. Neben der Verhandlung (8) zwischen den User-Agenten (7) erfolgt in (8) auch eine Verhandlung zwischen dem User-Agenten, bei dem der Arbeitsplatzwechsel festgestellt wurde und dem Service-Agenten (6). Ziel dieser Verhandlung ist es, keinen Tausch zwischen anderen User-Agenten zu vollziehen, sondern zum Zeitpunkt t einen Arbeitsplatzwechsel auf den Arbeitsplatz durchzuführen, auf dem der entsprechende Mitarbeiter zum Zeitpunkt $t-1$ oder $t+1$ arbeitet. Auch so wird die Anzahl der Arbeitsplatzwechsel reduziert. Der Service-Agent (6) stimmt dem Tausch nur dann zu, wenn es hierdurch in Summe nicht zu mehr Fehlerpunkten kommt. Das Ergebnis der Verhandlung ist entweder die Umplanung auf einen anderen Arbeitsplatz zum Zeitpunkt t und somit die Reduzierung der Arbeitsplatzwechsel oder keine Umplanung. Dieses Ergebnis wird dem Service-Agenten (6) mitgeteilt (9) und entsprechend vollzogen.

- Der Service-Agent (6) wiederholt die drei zuvor genannten Schritte so lange, bis keine Verbesserung eintritt. Den Einsatzplan meldet er an die Wissensbasis (1).

Sollen Stammarbeitsplätze berücksichtigt werden, so müssen die drei folgenden Aktionen an vorletzter Stelle in den zuvor geschilderten Ablauf eingefügt werden. In der letzten Aktion des Agentenansatzes werden dann folglich vom Service-Agenten (6) die letzten sechs zuvor genannten Schritte so lange wiederholt, bis keine Verbesserung eintritt.

- Service-Agent (6) untersucht jeden Zeitintervall des Einsatzplans und prüft für jeden Mitarbeiter, ob er auf seinem Stammarbeitsplatz arbeitet. Triff dies nicht zu, so werden für den entsprechenden Zeitintervall alle Mitarbeiter identifiziert, die für die Verhandlung in Frage kommen. Ein Mitarbeiter kommt dann für die Verhandlung in Frage, wenn er in dem Zeitintervall auf dem gewünschten Stammarbeitsplatz arbeitet (folglich anwesend ist) und für einen möglichen Tausch qualifiziert ist. Service-Agent (6) erzeugt für die relevanten Mitarbeiter je einen User-Agenten (7). Im Gegensatz zu den Scheduling-Agenten (3) existieren die erzeugten User-Agenten (7) zeitgleich.
- Die User-Agenten (7) verhandeln nun über einen Tausch der Arbeitsplatzzuweisung (8). Dies läuft folgendermaßen ab: Der fehlerhafte User-Agent (bei ihm hat der Service-Agent (6) den Stammarbeitsplatzfehler gefunden) fragt nacheinander bei allen anderen User-Agenten an, ob sie tauschen wollen. Der User-Agent, bei dem aktuell die Anfrage liegt und der fehlerhafte User-Agent, kennen jeweils ihre aktuellen Arbeitsplatzzuweisungen zum Zeitpunkt t und die Zuweisungen in $t-1$ und $t+1$. Sie tauschen lediglich die Information zu ihrer Arbeitsplatzzuweisung im Zeitpunkt t aus und können so ohne Berechnung der Fitnessfunktion entscheiden, ob ein Tausch zu weniger Fehlerpunkten führen würde. Wenn dem so ist, stimmt sie einem Tausch zu und teilt dies dem Service-Agenten (6) mit (9), woraufhin der Tausch vollzogen wird und alle User-Agenten gelöscht werden. Andernfalls fragt der fehlerhafte User-Agent beim nächsten User-Agenten an, bis alle abgefragt wurden. Als Ergebnis findet demnach entweder ein Tausch zwischen zwei der User-Agenten statt oder es findet kein Tausch statt, wenn sich das Planungsergebnis verschlechtern würde.
- Service-Agent (6) untersucht wiederum jeden Zeitintervall des Einsatzplans und prüft für jeden Mitarbeiter, ob ein Arbeitsplatzwechsel vorliegt. Neben der Verhandlung (8) zwischen den User-Agenten (7) erfolgt in (8) auch eine Verhandlung zwischen dem User-Agenten, der nicht auf seinem Stammarbeitsplatz arbeitet und dem Service-Agenten (6). Ziel dieser Verhandlung ist es, keinen Tausch zwischen anderen User-Agenten zu vollziehen, sondern zum Zeitpunkt t einen Arbeitsplatzwechsel auf den Stammarbeitsplatz des User-Agenten durchzuführen. Auch so wird die Anzahl der Arbeitsplatzwechsel reduziert. Der Service-Agent (6) stimmt dem Tausch nur dann zu, wenn es hierdurch in Summe nicht zu mehr Fehlerpunkten kommt. Das Ergebnis der Verhandlung ist entweder die Umplanung auf einen an-

deren Arbeitsplatz zum Zeitpunkt t oder keine Umplanung. Dieses Ergebnis wird dem Service-Agenten (6) mitgeteilt (9) und entsprechend vollzogen.

4.15.3.3 Automatische Arbeitszeitmodellerstellung im Handel

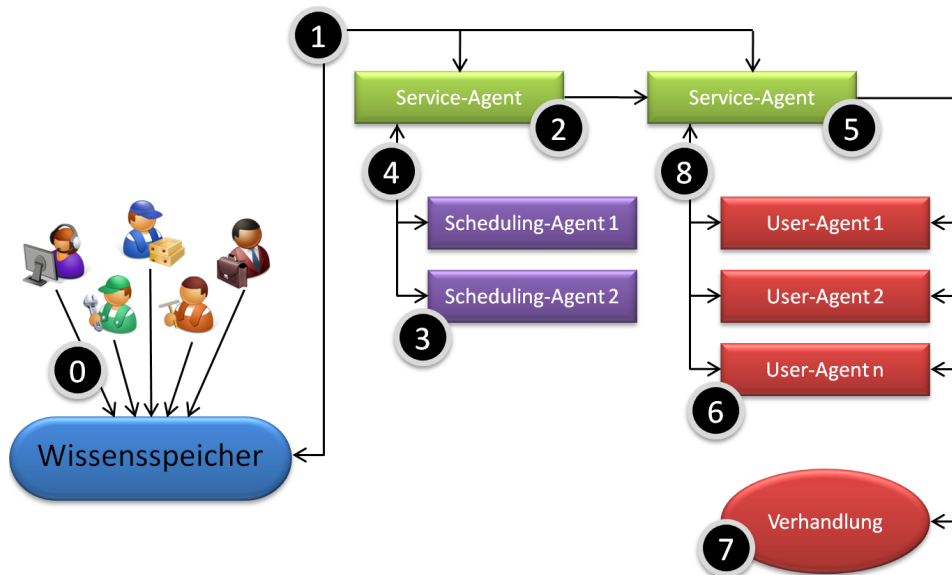


Abbildung 4.24: Ablauf des Agentenansatzes für die automatische Arbeitszeitmodellerstellung im Handel mit zwei Arbeitsplätzen

Der Agentenansatz für die beiden Handelsprobleme mit zwei Arbeitsplätzen und untertägigen Arbeitsplatzwechseln ist in Abbildung 4.24 schematisch dargestellt. Der Unterschied zu den Problemstellungen mit nur einem Arbeitsplatz liegt darin, dass in diesem Fall nur ein Scheduling-Agent (3) erzeugt wird und keine Priorisierung der Arbeitsplätze durch den Service-Agenten (2) vorgenommen werden muss. Zudem entfallen Arbeitsplatzwechsel, da nur ein Arbeitsplatz existiert. Auf die Beschreibung des Ablaufes für einen Arbeitsplatz wird an dieser Stelle verzichtet, da er dem Ablauf mit zwei (oder mehr) Arbeitsplätzen sehr ähnelt.

In Abbildung 4.15.3.3 erkennt man deutlich, dass sich der Ansatz stark am Vorgehen beim Logistikproblem orientiert. Die wesentlichen Unterschiede zwischen dem Agentenansatz für das Handelsproblem und dem Agentenansatz für das Logistikproblem sehen wie folgt aus:

- Scheduling-Agent (5) im Logistikproblem (Abbildung 4.23) wird im Handelsproblem nicht benötigt, da Arbeitszeitmodelle mit generiert werden. Es ist also explizit vorgesehen, dass Mitarbeiter nicht in allen Zeitintervallen, in denen sie verfügbar sind, verplant werden. Auch der Austausch von Mitarbeitern ist nicht nötig, da

keine Qualifikationen berücksichtigt werden müssen. Scheduling-Agent (5) in Abbildung 4.23 wurde daher im Handelsproblem entfernt.

- Im Handelsproblem müssen keine Qualifikationen berücksichtigt werden, sodass der Service-Agent (2) in Abbildung 4.24 bei der Priorisierung der Arbeitsplätze eine zufällige Auswahl trifft.
- Aufgrund fehlender Qualifikationen im Handelsproblem müssen die Scheduling-Agenten (3) und die User-Agenten (6) in Abbildung 4.24 nicht auf Qualifikationen achten.
- Im Handelsproblem treten deutlich mehr Restriktionen auf, die die Mitarbeiterbelange betreffen. Dies liegt daran, dass bei der Planung Arbeitszeitmodelle erstellt werden. Daher ist der Aufwand für Verhandlungen (7) in Abbildung 4.24 deutlich höher als beim Logistikproblem, wo lediglich Arbeitsplatzwechsel und Besetzungsabweichungen verhandelt werden.
- Statt einem, können nun mehrere Zeitintervalle Grundlage der Verhandlung (7) sein.

Nachfolgend wird der Ablauf des Handelsproblems mit zwei Arbeitsplätzen erläutert.

- Zunächst werden die Eigenschaften der existierenden Ressourcen, die aktuellen Anforderungen und die Rahmenbedingungen der Problemstellung in der Wissensbasis abgelegt (0). Hierzu gehören An-/Abwesenheiten der Mitarbeiter, Öffnungszeiten, Feiertage, minimale und maximale Längen der Arbeitszeitmodelle, Fehlerpunkthöhen für den Verstoß gegen Restriktionen, Länge der Planungsintervalle, Personalbedarfe je Zeitintervall, etc.
- Die Informationen des Wissensspeichers werden an die zwei Agenten (2) und (5) geliefert (1). Zudem erfolgen Rückmeldungen von den beiden Agenten an den Wissensspeicher.
- Zu Beginn der Planung initialisiert Service-Agent (2) den Einsatzplan, indem alle Mitarbeiter auf einen Dummy-Arbeitsplatz geplant werden. Dieser kennzeichnet, dass der jeweilige Mitarbeiter noch keinem reellen Arbeitsplatz zugeordnet wurde und daher nicht eingeplant ist. Die Zuweisung auf den Dummy-Arbeitsplatz erfolgt nur, wenn der Mitarbeiter auch verfügbar ist und der Shop geöffnet hat.
- Anschließend entscheidet Service-Agent (2) zufällig, welcher der beiden Arbeitsplätze im nächsten Schritt beginnen darf und daher die höhere Priorität erhält.
- Entsprechend der Priorität (höchste zuerst) werden nacheinander zwei Scheduling-Agenten (3) vom Service-Agenten (2) initialisiert. Jeder Scheduling-Agent (3) ent-

spricht einem der beiden Arbeitsplätze. Der Scheduling-Agent mit der höchsten Priorität beginnt und plant Mitarbeiter ein, die anwesend sind und noch nicht verplant wurden – sich also noch auf dem Dummy-Arbeitsplatz befinden. Dies geschieht mit dem Fokus auf die exakte Deckung des Personalbedarfs. Über- und Unterdeckung sollen minimiert werden, soweit dies möglich ist. Das Planungsergebnis des ersten Scheduling-Agenten wird dem Service-Agenten (2) mitgeteilt (4), der wiederum die Rückmeldung über den Plan an den Wissensspeicher gibt (1). Anschließend startet der Service-Agent (2) den zweiten Scheduling-Agenten (3), der nun ebenfalls versucht, den Personalbedarf bestmöglich zu decken. Dabei darf er nicht auf bereits verplante Mitarbeiter zugreifen.

- Die bisherige Einsatzplanung ist aus der Unternehmenssicht erfolgt: Einhaltung der harten Nebenbedingungen und ein möglichst passgenauer Personaleinsatz. Dies geschah unter Vernachlässigung der Mitarbeiterbelange. In diesem Fall unter Vernachlässigung der Reduzierung von Arbeitsplatzwechseln je Mitarbeiter und unter Vernachlässigung aller Regelungen für die Arbeitszeitmodelle. Zwar ist der Plan aus Unternehmenssicht sehr gut. Aus Mitarbeitersicht ist er nicht akzeptabel, zumal gegen diverse Regelungen der Betriebsvereinbarung verstoßen wurde. Daher ruft Service-Agent (2) nach Erstellung des Einsatzplans den Service-Agenten (5) auf, um die Mitarbeiterbelange in der Planung zu berücksichtigen. Service-Agent (5) ist ebenfalls mit der Wissensbasis verbunden (1).
- Service-Agent (5) untersucht jeden Zeitintervall des Einsatzplans und prüft für jeden Mitarbeiter, ob ein Arbeitsplatzwechsel, eine Besetzungsabweichung, ein Verstoß gegen die minimale oder maximal zulässige Länge der Arbeitszeiten vorliegt oder ob ein Mitarbeiter mehr als ein Arbeitszeitmodell pro Tag hat. Trifft dies zu, so werden für die entsprechenden Zeitintervalle alle Mitarbeiter identifiziert, die für eine Verhandlung in Frage kommen. Ein Mitarbeiter kommt dann für die Verhandlung in Frage, wenn er in den relevanten Zeitintervallen anwesend ist. Service-Agent (5) erzeugt für die relevanten Mitarbeiter je einen User-Agenten (6). Im Gegensatz zu den Scheduling-Agenten (3) existieren die erzeugten User-Agenten (6) zeitgleich.
- Die User-Agenten (6) verhandeln nun über einen Tausch der Arbeitsplatzzuweisung (7) in einem oder in mehreren Zeitintervallen. Der Dummy-Arbeitsplatz, bei dem ein Mitarbeiter zwar verfügbar aber nicht eingeplant ist, ist ebenfalls eine Tauschoption. Die Verhandlung läuft folgendermaßen ab: Der fehlerhafte User-Agent (bei ihm hat der Service-Agent (5) einen Fehler gefunden) fragt nacheinander bei allen anderen User-Agenten an, ob sie tauschen wollen. Der User-Agent, bei dem aktuell die Anfrage liegt und der fehlerhafte User-Agent, kennen jeweils ihre aktuellen Arbeitsplatzzuweisungen für die in Frage kommenden Zeitintervalle und die Zuweisungen der angrenzenden Zeitintervalle. Sie tauschen lediglich die Information zu ihren Arbeitsplatzzuweisungen im in Frage kommenden Zeitraum aus und können so ohne Berechnung der Fitnessfunktion entscheiden, ob ein Tausch zu weniger Fehlerpunkten führen würde. Wenn dem so ist, stimmt sie einem Tausch

zu und teilen dies dem Service-Agenten (5) mit (8), woraufhin der Tausch vollzogen wird und alle User-Agenten gelöscht werden. Andernfalls fragt der fehlerhafte User-Agent beim nächsten User-Agenten an, bis alle abgefragt wurden. Als Ergebnis findet demnach entweder ein Tausch zwischen zwei der User-Agenten statt oder es findet kein Tausch statt, wenn sich das Planungsergebnis verschlechtern würde.

- Neben der Verhandlung (7) zwischen den User-Agenten (6) erfolgt in (7) auch eine Verhandlung zwischen einem User-Agenten und dem Service-Agenten (5). Ziel dieser Verhandlung ist es, keinen Tausch zwischen anderen User-Agenten zu vollziehen, sondern z.B. die Länge von Arbeitszeitmodellen zu variieren. Da dies Auswirkungen auf die Über- und Unterdeckung im Personaleinsatz hat, muss der User-Agent (6) dies mit dem Service-Agenten (5) verhandeln (7). Derartige Verhandlungen finden auch statt, um erzeugte Besetzungsabweichungen möglichst zu korrigieren.
- Der Service-Agent (5) wiederholt die drei zuvor genannten Schritte so lange, bis keine Verbesserung eintritt. Den Einsatzplan meldet er an die Wissensbasis (1).

4.15.3.4 Kombination von Touren- und Personaleinsatzplanung

In diesem Kapitel werden zwei Agentenansätze für das British Telecom-Problem vorgestellt. Zunächst der Lösungsansatz von Kokkoras [217], [218], [330], der CLP von Yang [428] einsetzt. Anschließend wird ein eigener Agentenansatz vorgestellt, der sich am Vorgehen beim Logistik- und Handelsproblem orientiert und bei dem das Problem möglichst stark zerlegt wird.

Ansatz von Kokkoras Kokkoras verwendet 1996 [217] CLP von Yang [428], um das Problem BT_Rel-250-118 zu lösen. Dabei geht er in drei Schritten vor. Zunächst wird das Problem mit Hilfe von kooperierenden Agenten in Teilprobleme zerlegt. Die Zerlegung erfolgt servicecenterorientiert, sodass es für jedes Servicecenter einen Agenten gibt. Jedes der erzeugten Teilprobleme wird anschließend vom zugehörigen Agenten individuell gelöst, der das CLP-Verfahren von Yang [428] nutzt. In einer dritten Phase wird die so entstandene Lösung durch Kooperation der Agenten verändert, um möglichst viele noch nicht zugewiesene Aufträge auf Techniker zuzuteilen. Dieser Ansatz trägt nachfolgend die Bezeichnung D-CLP1. Der Ablauf ist in Abbildung 4.25 dargestellt.

In Phase eins (Clustering in Abbildung 4.25) wird das Problem in Teilprobleme zerlegt, um die Komplexität zu reduzieren und die CPU-Zeit für das Lösen zu verringern. Die reduzierte CPU-Zeit ermöglicht es, Algorithmen zu verwenden, die bei Anwendung auf die gesamte Problemstellung nicht praktikabel wären. U.U. kann bei zerlegten Problemen jedoch auch zusätzliche CPU-Zeit benötigt werden, um Konflikte bei Nebenbedingungen

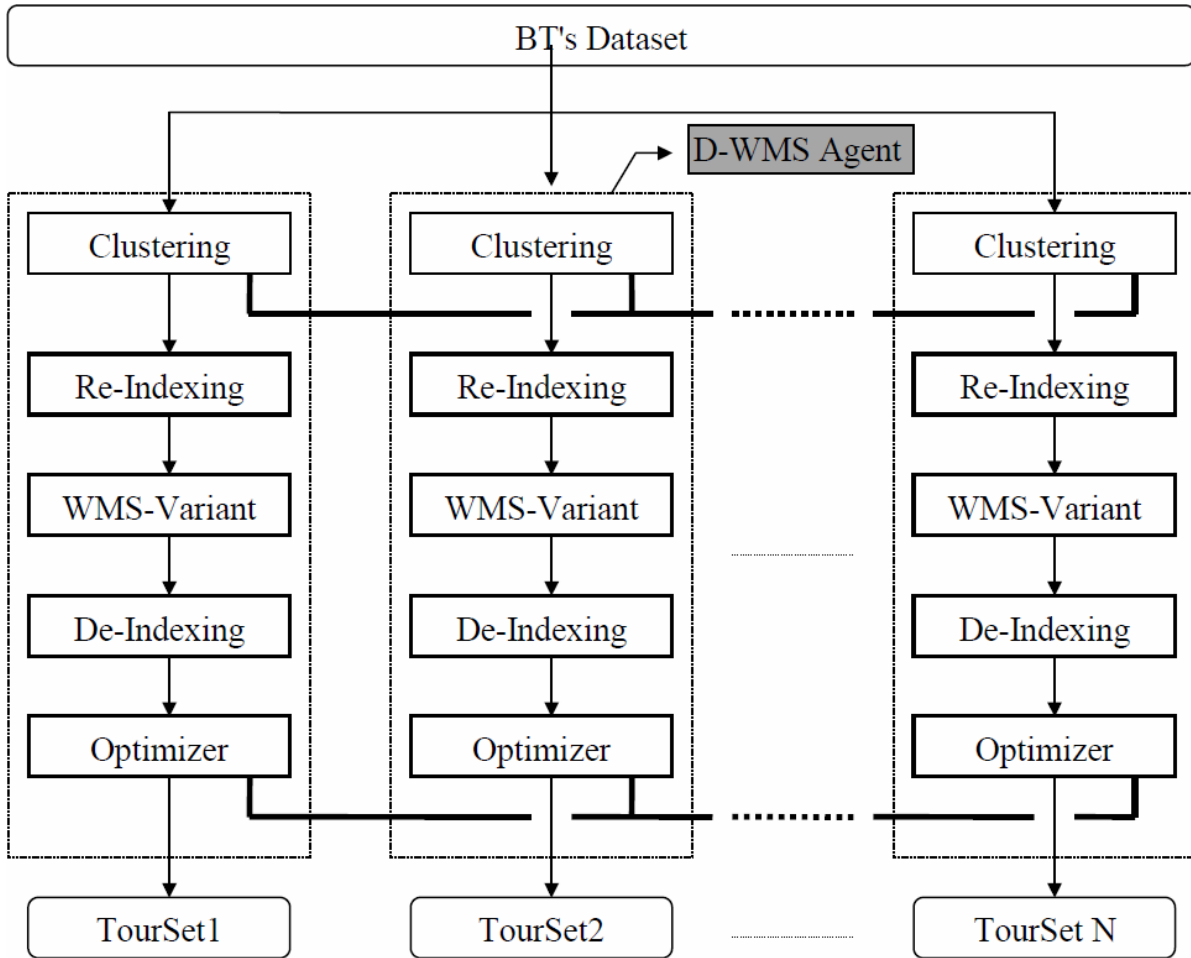


Abbildung 4.25: Ablauf des Agentenansatzes von Kokkoras für die Kombination von Touren- und Personaleinsatzplanung

zu lösen, die mehr als ein Teilproblem betreffen. Bei der Zerlegung in Phase eins wird sich an der Kapazität der Servicecenter (in Minuten) im Verhältnis zur Ausführungszeit von Aufträgen d_j orientiert. Hierzu wird eine Virtual Service Capacity (VSC_i) für jedes Servicecenter i berechnet.

$$VSC_i = \frac{\sum_{s=1}^S F_s}{F_i} * 2 \sum_{j=1}^J d_j \quad (4.24)$$

Dabei steht F für die noch freie Kapazität des jeweiligen Servicecenters. Die verschiedenen Auftragstypen werden nicht berücksichtigt, da Aufträge verschiedenen Typs recht gleichmäßig verteilt sind. Für die Zerlegung der Problemstellung wird zudem die Sharing Rate (SR_i) für jedes Servicecenter i benötigt. E_i steht für die Anzahl an Technikern im

Servicecenter i .

$$SR_j = \begin{cases} 2, & \text{wenn } E_i = 1 \\ 2 * \lceil \ln(E_i) \rceil, & \text{wenn } E_i > 1 \end{cases} \quad (4.25)$$

Der Ablauf von Phase eins ist in Algorithmus 21 dargestellt. Er wird als Balloon (B) bezeichnet.

Algorithmus 21 Ablauf von Balloon nach Kokkoras [217]

```

1: Ordne in Inputpooli für jedes Servicecenter  $i$  alle Aufträge  $j$  in
   aufsteigender Reihenfolge nach ihrer Entfernung zum Servicecenter
2: Berechne  $VSC_i$ 
3: Ordne in Centerlist alle Servicecenter  $i$  in aufsteigender Reihenfolge
   nach  $VSC_i$ 
4: while Noch nicht alle Aufträge zugewiesen do
5:   if  $SR_i = 0$  then
6:     Setze  $SR_i$  auf den Startwert
7:     Gehe zum nächsten Servicecenter  $i$  in Centerlist
8:   else if ( $VSC_i < 0 \ \&\& \text{Outputpool}_i < 6$ ) || ( $VSC_i > 0$ ) then
9:     Schiebe den 1. Auftrag  $j$  aus Inputpooli in Outputpooli
10:     $SR_i = SR_i - 1$ 
11:     $VSC_i = VSC_i - d_j$ 
12:    Entferne  $j$  aus Inputpooli aller anderen Servicecenter
13:   else
14:     Gehe zum nächsten Servicecenter  $i$  in Centerlist
15:   end if
16: end while

```

Der Algorithmus bricht ab, wenn alle Aufträge einem Servicecenter zugeordnet sind. Dies ist möglich, da die Gesamtkapazität aller Servicecenter größer als die Ausführungsdauer aller Aufträge ist (Fahrzeiten bleiben unberücksichtigt). Die Laufzeit des Algorithmus ist $O(J^2)$.

In Phase zwei wird CLP von Yang [428] genutzt, um jedes der Teilprobleme zu lösen. Hierzu ist eine Reindizierung der Auftrags- und Techniker-IDs notwendig. Nach der Lösung der Teilprobleme werden die Indizes wieder auf das Gesamtproblem übertragen. In Phase zwei erprobt Kokkoras ebenfalls die drei von Yang verwendeten Heuristiken Closes First, Same Direction First und Least Busy, um die Reihenfolge der Techniker zu bestimmen.

Nach Abschluss von Phase zwei gibt es noch sehr viele untätige Techniker und noch nicht zugewiesene Aufträge. Daher wird in Phase drei (Optimizer in Abbildung 4.25) versucht,

die Anzahl der nicht zugewiesenen Aufträge zu reduzieren. Die Idee ist, nicht zugewiesene Aufträge in Lücken zu platzieren, in denen ein Techniker untätig ist. Dabei geben Agenten Gebote für passende Aufträge ab und reduzieren infolge von Auftragszuweisungen die Gesamtfehlerpunkte. Der Ablauf der dritten Phase ist so gestaltet, dass die Kommunikation zwischen den Agenten möglichst gering ausfällt. Kokkoras wiederholt Phase drei so lange, bis sich keine Verbesserung mehr einstellt.

Kokkoras und Gregory erweitern 1998 [218] die Master-Arbeit von Kokkoras [217]. Ihre zweite CLP-Variante wird nachfolgend als D-CLP2 bezeichnet. Kokkoras und Gregory entwickeln für Phase eins zwei weitere Methoden, um das Gesamtproblem in Teilprobleme zu zerlegen. Die erste Methode trägt den Namen Center of Gravity. Zunächst werden alle Servicecenter so zu einer Fläche miteinander verbunden, dass die Servicecenter die Ecken darstellen. Das Gewicht der Ecken ergibt sich aus den dort stationierten Technikern. Anschließend wird der Schwerpunkt dieser Fläche bestimmt (Gravitationszentrum in Abbildung 4.26). Die Aufteilung des Gesamtproblems in Teilprobleme erfolgt nun wie folgt: Alle Aufträge werden auf Servicecenter zugeteilt. Dabei werden die Aufträge zuerst zugeteilt, die am weitesten vom Gravitationszentrum entfernt sind. Der Auftrag wird dann jeweils dem Servicecenter i zugewiesen, welches am nächsten ist, so lange folgendes für das Servicecenter i gilt: ($VSC_i < 0$ AND Aufträge in Outputpool $_i < 6$) OR ($VSC_i > 0$). Die zweite Methode heißt Mixed (M). Zunächst werden alle sehr weit von einem Servicecenter entfernten Aufträge mit Balloon (B) zugewiesen. Anschließend wird für die restlichen Aufträge Center of Gravity angewendet.

Eine weitere Erweiterung (D-CLP3) von D-CLP1 [217] und D-CLP2 [218] stellt die Arbeit von Sakellariou, Kokkoras und Vlahavas [330] dar, die sie 2003 veröffentlicht haben. Der wesentliche Unterschied zu D-CLP1 und D-CLP2 besteht in der Verwendung von CSPCONS (Communicating Sequential Prolog with Constraints), sodass jeder Agent als CSPCONS-Prozess modelliert wird. CSPCONS ist eine Erweiterung von CSP-II (Communicating Sequential Prolog II). Es ermöglicht eine Kommunikation der Agenten über TCP/IP.

Eigener Ansatz Zuvor wurde bereits ein Agentenansatz dargestellt. Während Kokkoras u.a. [217], [218], [330] für jedes Servicecenter einen Agenten erzeugen, wird hier ein anderer Ansatz verwendet. Nun soll die Problemstellung analog zum Logistik- und Handelsproblem möglichst stark zerlegt werden. Jeder Auftrag und jeder Techniker erhält einen Agenten. Aufträge wollen erfüllt und Techniker – mit Blick auf die Fahrtkosten – bestmöglich ausgelastet werden, wobei harte Nebenbedingungen eingehalten werden. Der Ablauf ist in Abbildung 4.27 dargestellt und wird nachfolgend beschrieben.

- Zunächst werden die Eigenschaften der existierenden Ressourcen, die aktuellen Anforderungen und die Rahmenbedingungen der Problemstellung in der Wissensbasis abgelegt (0). Hierzu gehören die Arbeitszeiten der Techniker, zugewiesene Service-

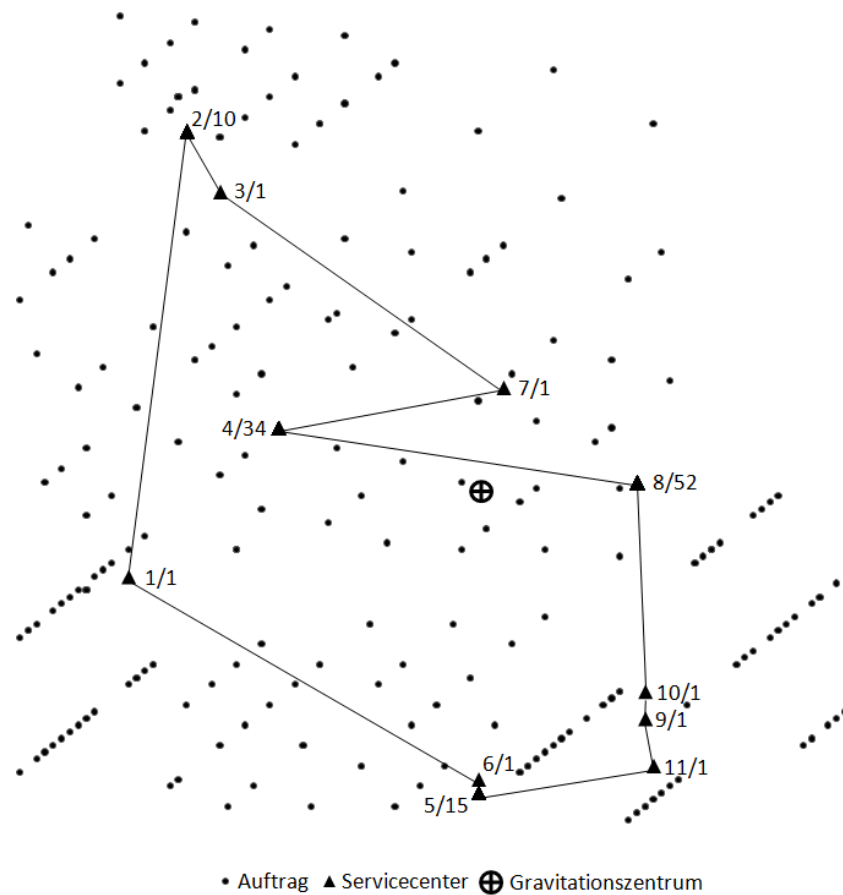


Abbildung 4.26: Position der Aufträge, Servicecenter und des Gravitationszentrums [218, S. 9] (Zahlen: Servicecenter-ID/Anzahl Techniker)

center der Techniker, Qualifikationen der Techniker, erforderliche Qualifikationen der Aufträge, Geoinformationen, Zeitfenster der Aufträge, Fehlerpunkthöhen für den Verstoß gegen Restriktionen, etc.

- Die Informationen des Wissensspeichers werden an die drei Agenten (2), (5) und (6) geliefert (1). Zudem erfolgen Rückmeldungen von den drei Agenten an den Wissensspeicher.
- Zu Beginn wird Service-Agent (2) aktiv, der eine Auftragssicht verfolgt. Er initialisiert nacheinander Job-Agenten (3), wobei jeder Job-Agent für einen Auftrag steht. Es kann immer nur ein Job-Agent zum gleichen Zeitpunkt existieren. Die Reihenfolge, in der diese Agenten von (2) initialisiert werden, ist zufällig.
- Ein Job-Agent (3) ist auf der Suche nach einem Techniker. Hierzu fragt jeder Job-Agent beim Service-Agenten (2) die Distanz aller qualifizierten Techniker von sich zum Servicecenter des jeweiligen Technikers ab (4). Service-Agent (2) erhält diese

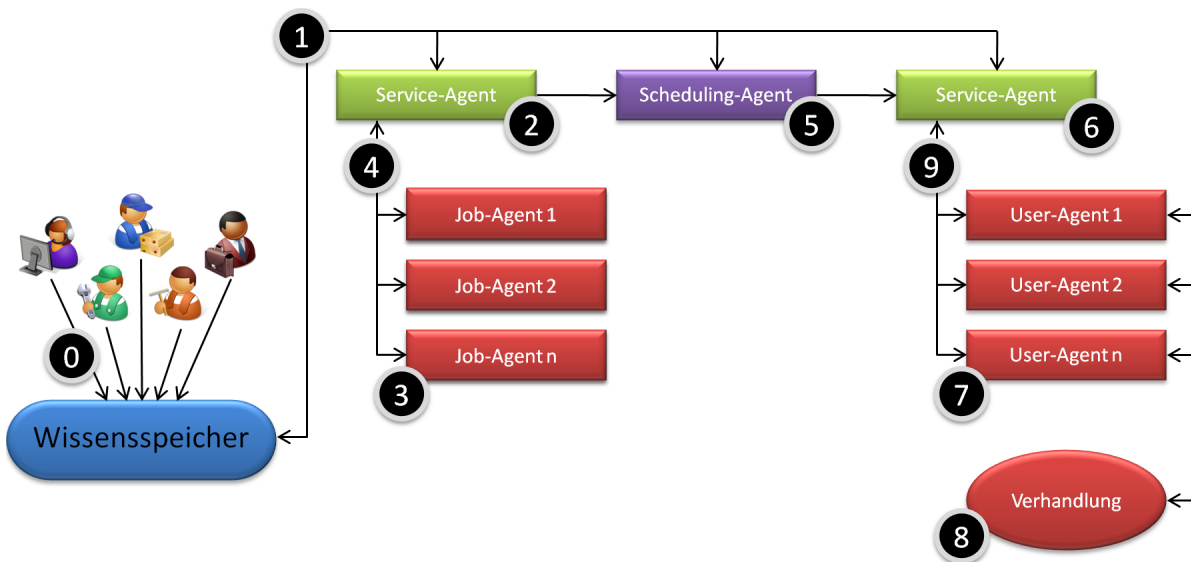


Abbildung 4.27: Ablauf des eigenen Agentenansatzes für die Kombination von Touren- und Personaleinsatzplanung

Information aus dem Wissensspeicher (1) und liefert sie an den Job-Agenten (3). Dieser wählt den Techniker mit der kürzesten Distanz, der noch freie Kapazitäten hat. Bei mehreren gleich guten Alternativen wird zufällig entschieden.

- Sollte es keinen qualifizierten Techniker mit freier Kapazität geben, so wird der Auftrag an einen zufällig ermittelten qualifizierten Techniker gegeben.
- Anschließend wird die Kapazität des Technikers im Wissensspeicher (1) aktualisiert. Hier wird nur die individuelle Bearbeitungszeit durch den Techniker berücksichtigt. Exakte Fahrtzeiten sind nicht bekannt, da nur die Entfernung zum Servicecenter berücksichtigt wurde. Auftragsreihenfolgen und somit Fahrtzeiten spielen daher noch keine Rolle.
- Nachdem bei allen Aufträgen je ein Job-Agent (3) einen Techniker zugewiesen hat, treten u.U. Verstöße gegen die Zeitfenster der Aufträge auf. Verstöße gegen diese harte Nebenbedingung müssen beseitigt werden. Daher wird nun vom Service-Agenten (2) ein Scheduling-Agent (5) aufgerufen, der ebenfalls mit der Wissensbasis verbunden ist (1). Scheduling-Agent (5) korrigiert nun mögliche Fehler auf zwei Wegen. Zunächst versucht der Scheduling-Agent (5) einen fehlerhaft platzierten Auftrag innerhalb der Tour eines Technikers zu verschieben. Hilft dies nicht, so wird der Auftrag bei einem qualifizierten Techniker an der korrekten Stelle platziert.
- Nachdem bisher der Fokus auf Aufträgen lag, die einen Techniker suchen, liegt der Fokus nun auf den Technikern, die untereinander verhandeln, um die Fehler-

punkte für nicht erledigte Aufträge zu reduzieren und die Wegezeiten zu verkürzen. Daher ruft der Scheduling-Agent (5) nach Erstellung des Einsatzplans den Service-Agenten (6) auf. Service-Agent (6) ist ebenfalls mit der Wissensbasis verbunden (1).

- Zunächst soll versucht werden, alle nicht erledigte Aufträge in freie Slots anderer Techniker zu platzieren. Hierzu prüft Service-Agenten (6) nacheinander alle Aufträge. Kann ein Auftrag von einem Techniker nicht erledigt werden, initialisiert Service-Agent (6) für den Techniker einen User-Agenten (7), der den nicht erledigten Auftrag enthält. Der User-Agent (7) meldet nun alle für den Auftrag qualifizierten Techniker (9) an den Service-Agenten (6), der diese ebenfalls als User-Agenten (7) initialisiert.
- Bei allen initialisierten User-Agenten (7) veranlasst der Service-Agent (6), dass die User-Agenten zunächst ihre Tour optimieren. Dies geschieht mittels einfachem Entfernen und neu Einfügen eines Auftrags innerhalb ihrer Tour. Für jeden Auftrag einer Tour wird der Fitnesswert aller möglichen Positionen berechnet. Die beste Position wird übernommen. Anschließend erfolgt die Berechnung für den nächsten Auftrag usw. Da jede Tour meist nur sehr wenige Aufträge beinhaltet, hält sich die Rechenzeit in Grenzen.
- In einer Verhandlung (8) können anschließend alle User-Agenten (7) ein Gebot für die Ausführung des bisher nicht ausgeführten Auftrags abgeben. Ist kein User-Agent (7) in der Lage, den Auftrag auszuführen, verbleibt der Auftrag beim ursprünglichen User-Agenten und die Verhandlung bricht ab. Andernfalls erhält der User-Agent den Zuschlag, der das niedrigste Gebot (entsprechend der verursachten Fehlerpunkte) abgibt. Das Ergebnis der Verhandlung wird dem Service-Agenten (6) mitgeteilt (9) und entsprechend vollzogen. Die User-Agenten (7) werden anschließend gelöscht.
- Bei der Verhandlung wurde nur darüber entschieden, ob ein Auftrag zu einem anderen Techniker gegeben werden kann. U.U. würde es Sinn machen, dass ein Techniker einen nicht erfüllten Auftrag von einem anderen Techniker übernimmt und ihm dafür einen anderen Auftrag zurück gibt. Sollte der Auftrag in der vorherigen Phase also nicht verschoben worden sein, wird Service-Agent (6) erneut aktiv. Hierzu initialisiert Service-Agent (6) den User-Agenten (7), der noch immer den nicht erledigten Auftrag enthält. User-Agent (7) meldet erneut alle für den Auftrag qualifizierten Techniker (9) an den Service-Agenten (6), der diese erneut als User-Agenten (7) initialisiert.
- In einer zweiten Verhandlung (8) können anschließend alle User-Agenten (7) ein Gebot für die Ausführung des bisher nicht ausgeführten Auftrags abgeben, wobei sie einen ihrer Aufträge an den Techniker mit dem nicht ausgeführten Auftrag abgeben. Es wird nur ein Auftrag abgegeben, für den der empfangende Techniker auch

qualifiziert ist. Existieren mehrere dieser Aufträge, so wird der Auftrag gewählt, der zu den geringsten Gesamtfehlerpunkten führt. Kann kein Auftrag abgegeben werden, scheidet dieser User-Agent aus der Verhandlung aus. Bei der Verhandlung erhält der User-Agent den Zuschlag, der zu den niedrigsten Gesamtfehlerpunkten beiträgt. Auf die Gesamtfehlerpunkte haben der entsprechende User-Agent und der User-Agent mit dem bisher nicht ausgeführten Auftrag einen Einfluss, da er einen Auftrag empfängt. Sollten sich die Gesamtfehlerpunkte verschlechtern, wird kein Tausch ausgeführt und die Verhandlung gilt als gescheitert. Das Ergebnis der Verhandlung wird dem Service-Agenten (6) mitgeteilt (9) und entsprechend vollzogen. Die User-Agenten (7) werden anschließend gelöscht.

- Service-Agent (6) wiederholt nach seinem Aufruf durch den Scheduling-Agenten (5) die letzten fünf Punkte, bis alle nicht erledigten Aufträge ausgeführt werden oder keine Verbesserung eintritt. Den Einsatzplan meldet er an die Wissensbasis (1).

In Kapitel 4.15.2 wurden diverse Arbeiten aus der Literatur dargestellt. Bei den eigenen Agentenansätze der drei Problemstellungen wurde gezeigt, dass sich diese an einigen der genannten Arbeiten orientieren. So wird der von Puppe u.a. [315] aufgeführte Ressourcenansatz verfolgt. Des Weiteren wird in zwei Phasen wie bei Krempels [228] vorgegangen. Auch der Ansatz von Krempels, in der zweiten Phase Agenten zu starten, wenn es zu einem Konflikt kommt, wird umgesetzt. Die eigenen Agentenansätze unterstützen ebenfalls die Forderungen von de Causmaecker, Ouelhadj und Vanden Berghe [93], dass beim Informationsaustausch zwischen Agenten nicht alle möglichen Informationen ausgetauscht werden. Und auch der Verhandlungsprozess dauert nicht unbegrenzt lange.

4.16 Hybride Lösungsverfahren

Hybride Lösungsverfahren wurden in den vorherigen Kapiteln z.T. bereits erwähnt. An dieser Stelle werden Implementierungen für die drei Problemstellungen dieser Arbeit erläutert (Kapitel 4.16.2). Zuvor wird kurz auf Arbeiten in der Literatur eingegangen. Ein Kapitel zum grundlegenden Ablauf entfällt, da Hybride Verfahren derart facettenreich sind, dass es den Rahmen dieser Arbeit sprengen würde. Für die einzelnen Teilverfahren der hybriden Ansätze sei jeweils auf die entsprechenden Kapitel dieser Arbeit verwiesen.

4.16.1 Anwendungen in der Literatur

Burke u.a. nutzen 1999 [58] TS für die Dienstplanung. Bei der Problemstellung existieren verschiedene Mitarbeitergruppen mit bestimmten Qualifikationen. Burke u.a. verwenden die Initialisierungsvarianten: aktueller Dienstplan, Dienstplan der Vorwoche und zufälliger Dienstplan. Ein Schritt wird vollzogen, indem die Schicht eines Mitarbeiters auf einen anderen Mitarbeiter übertragen wird. Der hybride TS schnitt besser ab als TS und HC für sich.

Dowland und Thompson nutzen 2000 [103] zwei klassische OR-Verfahren und TS für 17 reale Problemstellungen der Dienstplanung. Weder mit den klassischen OR-Verfahren noch mit TS können alle praktischen Erfordernisse an einen Dienstplan erfüllt werden. Erst mit einem hybriden Ansatz aus allen drei Verfahren, deren Kern TS ist, sind zufriedenstellende Ergebnisse zu erzielen.

Ein ähnliches Vorgehen wenden Valouxis und Housos 2000 [386] für die monatliche Dienst- und Pausenplanung an. Zunächst wird eine Lösung mit ILP erstellt. Diese wird anschließend mit einer Lokalen Suche verbessert. Daraufhin findet TS Anwendung, um die Lösung noch weiter zu verbessern. Valouxis und Housos ziehen CLP mit dem ILOG-Solver zum Vergleich heran. Der hybride Ansatz konnte CLP in Bezug auf Geschwindigkeit und Lösungsqualität schlagen.

4.16.2 Anpassung an hochflexibles Workforce Management

Nachfolgend wird auf die Anpassung von hybriden Lösungsverfahren an die Problemstellungen dieser Arbeit eingegangen. Die Ausführungen zu TS + HC basieren auf der Arbeit von Helbig [178, S. 39–51]. Die zweite Gruppe der Hybriden Verfahren basiert auf der Kombination von MAS mit ES bzw. PSO. Zudem werden weitere Verfahrenskombinationen erörtert.

4.16.2.1 Untertägige Personaleinsatzplanung in der Logistik

Hill Climbing und Tabu Search An dieser Stelle wird die Kombination von HC mit TS beschrieben, wobei Untersuchungen nur für die Probleme ohne Stammarbeitsplatzregelung angestellt werden. Detailliertere Ausführungen sind der Diplomarbeit von Helbig [178, S. 39–51] zu entnehmen. Für den hybriden Ansatz wird TS aus Kapitel 4.10.3.1 verwendet. Dieser Ansatz hat die Eigenschaft, bereits viele grobe Verstöße beim Logistikproblem zu beheben. Besonders bei der Reduzierung der Arbeitsplatzwechsel hat dieses Verfahren jedoch Probleme. TS nutzt Shift-Moves für Schritte in der Nachbarschaft.

Diese sind nicht sehr effektiv, wenn es um die Reduzierung von Arbeitsplatzwechseln geht. Die Aufgabe von HC ist es also, eine bereits gute Lösung weiter zu verbessern und vor allem Arbeitsplatzwechsel zu reduzieren. Nachfolgend ist der grundlegende Ablauf des hybriden Verfahrens von Helbig aus HC und TS in Algorithmus 22 dargestellt.

Die Zeilen 1–3 beinhalten die Initialisierung des Verfahrens. Die Bestimmung der Startlösung kann auf zwei verschiedenen Wegen erfolgen. Beim hybriden Ansatz aus HC und TS werden die Initialisierungen I2 und I3 eingesetzt, die bereits bei TS für das Logistikproblem Anwendung finden. Für weitere Ausführungen sei auf das Kapitel 4.10.3.1 verwiesen. Die Variable t enthält die Nummer der aktuellen Iteration und die Variable v die Nummer der Iteration, bei der die letzte Verbesserung erzielt wurde.

Die Zeilen 5–11 beinhalten den HC. Dieser wird ausgelöst, wenn die Differenz aus t und v ein Vielfaches von 50 erreicht. Somit wird erreicht, dass der HC erst dann startet, wenn TS seit längerem keine Verbesserung erzielen konnte. Nach einer Verbesserung erhält TS auch genügend Zeit, im Lösungsraum nach weiteren Verbesserungen zu suchen. Beim HC werden die Schritte mit dem Swap Move vollzogen. Dabei wird immer von der aktuellen Lösung X und nicht von der besten Lösung B ausgegangen. Andernfalls würde es bei Verwendung von B zu Zyklen kommen, wenn der HC keine Verbesserung findet.

Für die Nachbarschaft des HC werden alle Dimensionen gewählt, in denen ein Mitarbeiter anwesend ist. Die Nachbarschaft des HC entspricht also der dritten Nachbarschaft von TS, die mit der Wahrscheinlichkeit $p = 0,5\%$ aufgerufen wird. Diese Nachbarschaft wird gewählt, da die Lösung bereits recht gut ist und Verbesserungen nur nach einer intensiven Suche gefunden werden.

Da HC vor allem unnötige Arbeitsplatzwechsel reduzieren soll, kommen Swap-Moves zum Einsatz. Hierbei kann es Situationen geben, in denen der Austausch zwischen lediglich einem Zeitintervall zweier Mitarbeiter zu keiner Verbesserung führt. Z.T. müssen mehrere Zeitintervalle gleichzeitig getauscht werden. Beim HC kommt daher der Swap Move nach dem Vorbild von Burke u.a. [57, S. 10] zum Einsatz, bei dem für mehrere Schichten geprüft wird, ob ein Wechsel lohnend ist oder nicht. In Abbildung 4.28 ist das gewählte Vorgehen dargestellt. Würde dort nur ein Tausch zwischen Mitarbeiter drei und sechs in Periode drei oder in Periode vier erfolgen, würde dies die Anzahl an Arbeitsplatzwechseln nicht reduzieren – bei Periode drei sogar erhöhen. Erst der komplette Tausch der Perioden drei und vier führt zu einer Verbesserung.

Der Swap Move erfolgt für alle Mitarbeiter, die anwesend sind in genau einem Zeitintervall. Anschließend wird der Tausch von einem Zeitintervall auf zwei, drei, . . . , bis zu acht Zeitintervallen ausgedehnt. Mehr als acht Zeitintervalle werden nicht ausgetauscht, da dann die Wahrscheinlichkeit unnötige Arbeitsplatzwechsel zu reduzieren sehr gering ist. Zudem steigt ohne eine Begrenzung der Rechenaufwand stark an und ab acht Zeitintervallen überschneiden sich die Anwesenheiten von Mitarbeitern aufgrund der Vielzahl an Arbeitszeitmodellen nur noch selten. Ein Schritt wird auch dann akzeptiert, wenn

	Periode 1	Periode 2	Periode 3	Periode 4	Periode 5
Mitarbeiter 1	0	0	1	1	1
Mitarbeiter 2	0	0	0	0	1
Mitarbeiter 3	0	1	2	2	1
Mitarbeiter 4	0	0	0	1	1
Mitarbeiter 5	0	0	0	1	1
Mitarbeiter 6	0	0	1	1	2



	Periode 1	Periode 2	Periode 3	Periode 4	Periode 5
Mitarbeiter 1	0	0	1	1	1
Mitarbeiter 2	0	0	0	0	1
Mitarbeiter 3	0	1	1	1	1
Mitarbeiter 4	0	0	0	1	1
Mitarbeiter 5	0	0	0	1	1
Mitarbeiter 6	0	0	2	2	2

Abbildung 4.28: Swap-Move für zwei Zeitintervalle [178, S. 46]

der Fitnesswert gleich bleibt. Dies soll die Diversifikation fördern.

Wie bei HC für das Logistikproblem existieren auch hier mehrere Durchlaufrichtungen (siehe Kapitel 4.8.3.1). Es ist für den Verlauf der Suche nicht unerheblich, ob zunächst die Mitarbeiter eins und zwei einen Tausch ausführen oder die Mitarbeiter 1 und 65. So kann für den Tausch am Anfang oder am Ende der Mitarbeiterliste begonnen werden. Ebenso kann für die Wahl des Tauschpartners am Anfang oder am Ende der Mitarbeiterliste begonnen werden. Welche der sich so ergebenden vier Variante der Durchlaufrichtung aktuell gewählt wird, wird in jeder Iteration zufällig bestimmt.

In den Zeilen 12–31 von Algorithmus 22 ist TS dargestellt. Es wird in jeder Iteration gestartet. Unabhängig davon, ob der HC aktiv war oder nicht. Der Ablauf von TS erfolgt exakt nach dem Vorgehen in Kapitel 4.10.3.1.

Beim hybriden Ansatz aus HC und TS werden zusätzlich zwei Varianten erprobt, die auf die Arbeit von Helbig zurückzuführen sind [178]. Sie werden ausschließlich am Wochenproblem angewendet. An den Tagesproblemen sind die Ergebnisse bereits derart gut, dass keine weiteren Modifikationen erforderlich sind.

- Um die Anzahl an Berechnungen der Fitnessfunktion zu reduzieren, wird eine hybride Variante erprobt, in der das Akzeptanzkriterium First Accept statt Best Accept verwendet wird. Zudem kommt dort kein Aspirationskriterium zum Einsatz. Diese TS-Variante wird nachfolgend durch die Bezeichnungen (F) für First Accept und (-A) für das Weglassen des Aspirationskriteriums kenntlich gemacht.
- Eine zweite Variante erweitert den zuvor genannten Ansatz. Da sich die einzelnen

Tage im Wochenproblem nicht gegenseitig beeinflussen, können Nachbarschaften auf einen der sieben Tage begrenzt werden. Es wird also nicht im kompletten Wochenproblem gesucht. Die Suche beginnt beim ersten Tag. Nach einer definierten Anzahl an bewerteten Fitnessfunktionen (Abbruchkriterium / 7) springt die Suche zum nächsten Tag und sucht lediglich in diesem Tag, bis sie weiterspringt. Diese TS-Variante trägt zusätzlich die Bezeichnung /T.

Hybride Verfahren mit Agentensystemen In Kapitel 4.15.3.2 wurde ein MAS für das Logistikproblem vorgestellt. An dieser Stelle werden Möglichkeiten für hybride Ansätze erörtert.

Das MAS liefert sehr schnell eine Lösung. Es liegt daher nahe, Startlösungen für ES oder PSO vom Agentenansatz erzeugen zu lassen. In Tests wurde dieser hybride Ansatz erprobt. Da die Ergebnisse des MAS bereits recht gut sind, befinden sich die Individuen bzw. Partikel jedoch in lokalen Optima. Ein Entkommen gestaltet sich derart schwer, dass PSO und ES die Startlösungen des MAS nicht verbessern können. Dieser Ansatz muss also verworfen werden. Es würde lediglich mehr Rechenzeit und einen höheren Implementierungs- und Parametrierungsaufwand bedeuten, jedoch zu keiner Ergebnisverbesserung beitragen.

Die umgekehrte Variante, ES oder PSO erzeugen eine Startlösung, woraufhin ein MAS eingesetzt wird, ist ebenfalls nicht anwendbar. Der Agentenansatz unterstützt dieses Vorgehen nicht. Denn er konstruiert eine Lösung und setzt hierzu zunächst alle Mitarbeiter auf einen Dummy-Arbeitsplatz, der kenntlich macht, dass der jeweilige Mitarbeiter noch zugewiesen werden kann. Die Startlösung ist damit verloren. Auch diese Variante führt zu keiner Ergebnisverbesserung. Lediglich zu mehr Rechenzeit und einem größeren Aufwand für Implementierung und Parametrierung.

Ein dritter Ansatz sieht vor, das Verhandlungsergebnis von einem Optimierungsverfahren wie z.B. HC, ES oder PSO berechnen zu lassen. Da die Agenten über recht viel problemspezifisches Wissen verfügen, sind derartige Optimierungsverfahren nicht erforderlich. Agenten kennen immer die optimalen Entscheidungen, um die Fitness weiter zu verbessern. Daher muss auch dieser hybride Ansatz verworfen werden.

Evolutionary Swarm Optimization Unter der Bezeichnung Evolutionary Swarm Optimization (ESO) wird nachfolgend eine Kombination aus ES und PSO verstanden. Ein derartiger Ansatz ist bisher nicht in der Literatur zu finden. Die Idee sieht folgendermaßen aus: Der hier verwendete PSO-Algorithmus wird vornehmlich durch die Eintrittswahrscheinlichkeit der Aktion zwei bei der Positionsneubestimmung gesteuert. Er ist relativ unempfindlich gegen Änderungen der anderen drei Eintrittswahrscheinlichkeiten für die Aktionen eins, drei und vier. Im Gegensatz zu ES sterben die Partikel bei PSO

nicht. PSO verfügt über keine Rekombination und über keine Selektion. Die Mutation von ES ist in den Grundzügen vergleichbar mit der Positionsneuberechnung von PSO. Ein ESO-Algorithmus kombiniert die Elemente von ES und PSO. Es wird ein Schwarm erzeugt, der in jeder Iteration (jetzt Generation) mittels Rekombination Nachkommen erzeugt. Diese werden mit Hilfe der Positionsneuberechnung des ursprünglichen PSO verändert, wobei die Eintrittswahrscheinlichkeit für Aktion zwei selbstanpassend ist. Im Rahmen der anschließenden Selektion wird die Population auf die vorgegebene Schwarmgröße reduziert. Anschließend startet eine neue Generation. Da das Verhalten von PSO sehr stark von der Eintrittswahrscheinlichkeit für Aktion zwei abhängt, soll ESO vor allem von der Selbstanpassung dieser Wahrscheinlichkeit profitieren. Die anderen drei Wahrscheinlichkeiten ergeben sich regelbasiert in Abhängigkeit von der selbstangepassten Wahrscheinlichkeit für Aktion zwei.

PSO ist eine schwarmbasierte Metaheuristik, die Zusammenhänge in der Natur nachbildet. Mit dem ESO können Partikel (z.B. Vögel) nun nicht mehr nur sprechen, hören, sich bewegen oder wie beim British Telecom-Problem auch sehen. Nun können sie zusätzlich Nachkommen erzeugen und sterben.

Im Rahmen dieser Arbeit wurden verschiedene Tests mit ESO angestellt. Bei keinem der Experimente am Logistikproblem konnte ESO die Ergebnisse von PSO oder ES schlagen. Die Kombination von PSO und ES erweist sich zumindest dort als unbrauchbar. Anscheinend behindern sich die Suchoperatoren gegenseitig. Zudem ist die Parametrierung von ESO im Vergleich zu PSO oder ES ungleich komplexer. Auf ausführliche Experimente wurde daher verzichtet. Da der Implementierungs- und Parametrierungsaufwand für ESO doch erheblich ist, wurde dieser Ansatz nicht beim Handels- oder British Telecom-Problem umgesetzt.

4.16.2.2 Automatische Arbeitszeitmodellerstellung im Handel

Hybride Verfahren mit Agentensystemen Analog zum Logistikproblem wird auch hier der Ansatz verfolgt, Startlösungen für ES oder PSO vom MAS erstellen zu lassen. Beim Logistikproblem ist dieser Ansatz daran gescheitert, dass die Startlösungen bereits sehr gut waren und die Individuen bzw. Partikel nicht aus den lokalen Optima entkommen konnten. Hier ist das MAS nicht mehr derart gut. Daher können ES und PSO die Startlösungen auch verbessern. Sie sind jedoch nicht in der Lage, die Ergebnisse mit ihrer ursprünglichen Initialisierungsvariante zu schlagen. Auf weiterführende Experimente wurde daher verzichtet.

Wie beim Logistikproblem ist die umgekehrte Variante, bei der ES oder PSO eine Startlösung erzeugen und dann der Agentenansatz einsetzt, nicht anwendbar. Der Agentenansatz unterstützt ein derartiges Vorgehen nicht. Er ist recht nah an das Vorgehen des Agentensystems für das Logistikproblem angelehnt und konstruiert eine Lösung. Dabei

werden zu Beginn alle Mitarbeiter auf einen Dummy-Arbeitsplatz gesetzt. Dieser macht kenntlich, dass der jeweilige Mitarbeiter noch zugewiesen werden kann. Die Startlösung von ES oder PSO ist damit verloren, sodass es in dieser hybriden Form zu keiner Ergebnisverbesserung gegenüber der alleinigen Verwendung des Agentensystems kommt. Es ist lediglich ein Anstieg der Rechenzeit und des Aufwands für Implementierung und Parametrierung zu verzeichnen.

Die Agenten besitzen sehr viel problemspezifisches Wissen und treffen zu jedem Zeitpunkt optimale Entscheidungen. Der Einsatz von Optimierungsverfahren im Rahmen von Verhandlungen ist damit nicht notwendig.

4.16.2.3 Kombination von Touren- und Personaleinsatzplanung

Hybride Verfahren mit Agentensystemen Die folgenden Ausführungen gelten für das eigene Agentensystem, da auf das Verfahren von Kokkoras u.a. [217], [218], [330] kein Zugriff besteht.

Zunächst wird die Variante diskutiert, in der das MAS die Startlösungen für ES und PSO erzeugt. Wie beim Logistik- und Handelsproblem ist auch hier dieser Ansatz unbrauchbar. Die Ergebnisse des MAS können zwar verbessert werden. Die Ergebnisse von ES und PSO mit ihrer ursprünglichen Initialisierungsvariante können jedoch nicht geschlagen werden. Anscheinend sind die Ergebnisse des MAS derart strukturiert, dass ES und PSO damit Probleme haben. Experimente mit diesem Ansatz machen daher keinen Sinn.

Da sich das MAS am Ablauf der Ansätze beim Logistik- und Handelsproblem orientiert, ist die hybride Variante, in der eine Startlösung z.B. mit ES oder PSO erzeugt wird und anschließend Agenten eingesetzt werden, nicht anwendbar. Auch hier wird vom Agentenansatz eine Lösung konstruiert, die alle vorherigen Ergebnisse überschreiben würde. Daher ist dieses Vorgehen nicht zielführend und bedeutet lediglich mehr Rechenzeit sowie einen größeren Implementierungs- und Parametrierungsaufwand.

Das MAS verfügt bei der Kombination von Touren- und Personaleinsatzplanung über deutlich weniger problemspezifisches Wissen als die Ansätze für das Logistik- oder Handelsproblem. Dies liegt an der deutlich höheren Komplexität der Problemstellung, was die Erstellung von Agenten mit tiefem Wissen erschwert. Agenten wissen nun nicht sofort, ob eine Entscheidung lohnend ist oder nicht. Sie müssen immer erst die Fitness berechnen. Aufgrund dieser Berechnungen treffen sie Entscheidungen, die für sich genommen immer optimal sind. Die Optimalität der Entscheidung ist garantiert, da alle Zuweisungsmöglichkeiten für einen Auftrag geprüft werden. Die hierfür erforderlichen Berechnungen halten sich in Grenzen, da meist nicht sehr viele Techniker für einen Auftrag qualifiziert sind und demzufolge in Frage kommen. Der Einsatz von Optimie-

rungsverfahren wie z.B. HC, ES oder PSO ist daher nicht zur Entscheidungsfindung erforderlich.

Da sich die hybriden Ansätze mit MAS nicht eignen oder nicht umsetzbar sind, wird in Kapitel 6.2 nicht mit einem gesonderten Unterkapitel auf hybride Verfahren eingegangen.

Hybride Verfahren mit einer konstruierten Startlösung In den Kapiteln 4.12.4.3 und 4.14.3.3 wurden ES und PSO für das British Telecom-Problem modifiziert. Die Startlösung beider Verfahren wird jeweils mit einem KV erzeugt. Prinzipiell handelt es sich daher bei diesen ES- und PSO-Ansätzen um ein hybrides Verfahren mit KV. Tsang und Voudouris [383] gehen beim British Telecom-Problem ähnlich vor und sprechen z.B. bei ihrem HC ebenfalls nicht von einem hybriden Ansatz aus KV und HC. Auch beim Logistik- und Handelsproblem werden ES und PSO nicht als hybride Verfahren bezeichnet, obwohl mit I1 und I2 prinzipiell eine Lösung – wenn auch auf sehr einfachem Weg – konstruiert wird. Eigene Unterkapitel für hybride ES- und PSO-Varianten würden den Leser zu stark verwirren, da dann Lücken in der Gliederung dieser Arbeit entstehen. ES und PSO sind ja die zentralen Metaheuristiken dieser Arbeit. Nachfolgend wird daher nur von ES und PSO gesprochen. Aus diesem Grund und da hybride Ansätze mit MAS nicht zielführend bzw. nicht umsetzbar sind, existiert in Kapitel 6.3 kein eigenes Unterkapitel zu hybriden Verfahren.

Nachdem zahlreiche Lösungsverfahren vorgestellt und deren Anpassung an die Problemstellungen dieser Arbeit erläutert wurden, wird im folgenden Kapitel auf implementierungsrelevante Sachverhalte eingegangen.

Algorithmus 22 HC + TS bei der untertägigen Personaleinsatzplanung in der Logistik

```

1: Iteration  $t = 0$  und Verbesserung  $v = 0$ 
2: Erzeuge Startlösung  $X$  und bestimme beste Lösung  $B$ 
3: Random Startpunkt  $I$  aus Mitarbeiter und Zeitintervall
4: while Abbruchkriterium nicht erfüllt do
5:   if  $(t - v) = h$  then
6:     Generiere alle Lösungen  $\hat{X}$  aus  $N$  von  $I$  durch Swap Move
7:     Wähle  $\hat{X}$  mit minimaler Fitness
8:     if  $f(\hat{X}) \leq f(X)$  then
9:        $X = \hat{X}$  und  $I = \hat{I}$ 
10:    end if
11:  end if
12:  Generiere alle Lösungen  $\hat{X}$  aus  $N$  von  $I$  durch Shift Move
13:  Fitness von lokaler Bestlösung  $LB = \infty$ 
14:  Wähle  $\hat{X}$  mit minimaler Fitness
15:  if  $f(\hat{X}) \leq f(LB)$  &&  $\hat{X} \neq \text{tabu}$  then
16:     $f(LB) = f(\hat{X})$ 
17:  else if  $f(\hat{X}) < f(B)$  &&  $\hat{X} = \text{tabu}$  then
18:     $f(LB) = f(\hat{X})$ 
19:  end if
20:  Aktualisiere Tabuliste bei  $\hat{I}$  mit  $T = \text{random}(T_{min}; T_{max})$ 
21:  Reduziere alle  $T$  um 1
22:  if  $f(LB) < f(B)$  then
23:     $B = LB$  und  $v = t$ 
24:  end if
25:   $t = t + 1$ 
26:  if  $\hat{I} = I$  then
27:    random Startpunkt  $I$  aus Mitarbeiter und Zeitintervall
28:  else
29:     $I = \hat{I}$ 
30:  end if
31:   $X = \hat{X}$ 
32: end while

```

Kapitel 5

Rahmenbedingungen für Implementierung und Experimente

In diesem Kapitel wird zunächst die selbst entwickelte Softwarelösung beschrieben, die für fast alle Experimente dieser Arbeit genutzt wird. Eine Ausnahme bilden die klassischen OR-Verfahren. Bei ihnen wird GAMS herangezogen, da GAMS zahlreiche Solver bereit stellt und lediglich die Problemstellung vom Anwender modelliert werden muss. Anschließend wird in Kapitel 5.2 das selbst entwickelte XML-Format beschrieben, welches zum Speichern und Einlesen von Problemstellungen genutzt wird. Das Arbeiten mit einer Vielzahl an Problemstellungen wird so erheblich erleichtert. In Kapitel 5.3 wird ausführlich auf diverse statistische Tests eingegangen, die in dieser Arbeit Anwendung finden. Mit Ausführungen zu Beurteilungskriterien für Lösungsverfahren endet dieses Kapitel.

5.1 Die Softwarelösung ARROW

Speziell für diese Arbeit wurde die Softwarelösung ARROW entwickelt. Ziel des Systems ist es, die Experimente dieser Arbeit optimal zu unterstützen. ARROW steht für Automatic Resource and Route Oriented Workforce Management. Es ist vollständig in C# implementiert und umfasst etwa 10 MB an Quellcode. Das System besteht aus zwei Teilen. Zum einen die Logik für die Lösungsverfahren, die Verarbeitung von Problemstellungen und die Ausgabe von Ergebnissen nach MS EXCEL™. Zum anderen die grafische Oberfläche, in der Lösungsverfahren parametrisiert und Problemstellungen bearbeitet werden können.

ARROW wird genutzt, um schnell und einfach verschiedenste Problemstellungen des hochflexiblen WFM einzulesen oder zu erstellen, zu bearbeiten sowie zu speichern. Zudem ermöglicht es die Nutzung diverser Lösungsverfahren, die über einen separaten Dialog parametrisiert und gestartet werden können. Dabei können mehrere Konfigurationen

zeitgleich gestartet werden, was bei Mehrkernprozessoren sehr effektiv ist.

Für den praktischen Einsatz ist ARROW in der jetzigen Form nicht geeignet. Individuelle Anpassungen müsste der Planer meist noch programmieren und kann diese nicht parametrieren. Das hierfür erforderliche Wissen ist doch recht tief. Die Umsetzung einer voll parametrierbaren Oberfläche würde etwa 25 Manntage in Anspruch nehmen.

In Abbildung 5.1 ist der Planungsdialog von ARROW dargestellt. Das System verfügt über weitere Dialoge, die für die Bearbeitung von Problemstellungen und die Steuerung der Lösungsverfahren genutzt werden können. ARROW unterstützt folgende Funktionalitäten:

- Automatische Erstellung eines Einsatzplans mit diversen Lösungsverfahren.
- Berücksichtigung diverser harter und weicher Nebenbedingungen.
- Automatische Arbeitszeitmodellerstellung.
- Optimierung von Auftragsreihenfolgen.
- Planung untertägiger Arbeitsplatzwechsel.
- Beliebige Anzahl an Mitarbeitern, Arbeitsplätzen, Funktionen oder Aufträgen.
- Beliebige kleine Zeitfenster für die Planung und den Personalbedarf.
- Berücksichtigung untertägiger Bedarfsschwankungen.
- Berücksichtigung von Qualifikationen.
- Berücksichtigung individueller Arbeitsverträge.
- Ausgabe des Einsatzplans nach MS EXCEL™.
- Import und Export von Problemstellungen im XML-Format.
- Modifikation von Problemstellungen über eine grafische Oberfläche (anlegen/ändern/löschen von Mitarbeitern, Arbeitsplätzen, Qualifikationen, Arbeitsverträgen, Schichten).
- Steuerung von Lösungsverfahren über eine grafische Oberfläche.

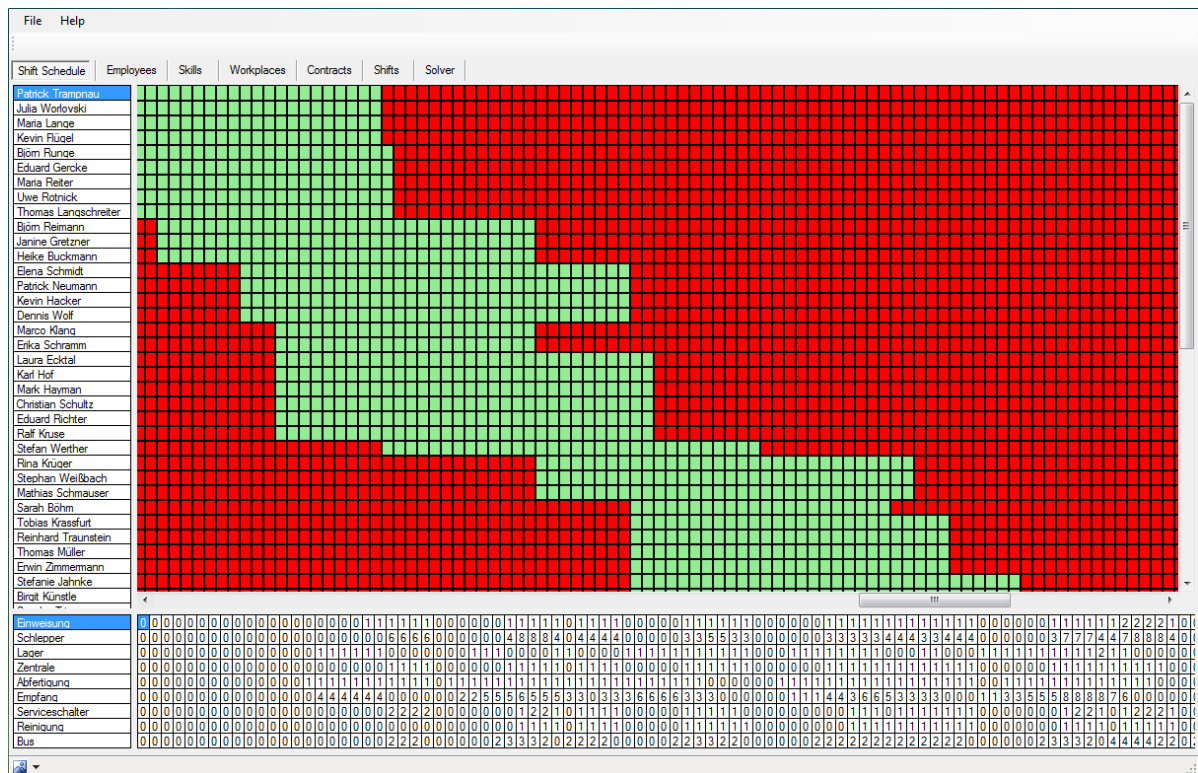


Abbildung 5.1: Planungsdialog mit An-/Abwesenheiten und Personalbedarf in ARROW

5.2 XML-Format für Workforce Management-Probleme

Das Vorhandensein eines Datenformates ist für eine intensive Forschung an Problemstellungen unabdingbar. Nur so ist es möglich, Planungsprobleme einer Vielzahl an Forschern zugänglich zu machen, sodass an ihnen vielfältige Benchmarktests vorgenommen werden können. Betrachtet man die Vielzahl an Veröffentlichungen der letzten Dekaden zum Thema Personaleinsatzplanung, zeigt sich, dass Vergleiche der vorgestellten Lösungsverfahren wegen der Verschiedenartigkeit der Problemstellungen nur schwerlich möglich sind.

Reis und Oliveira [324, S. 328] fordern, dass mit einem Datenformat School, Exam und University Timetabling Probleme abbildbar sein müssen. Diese Forderung steht beispielhaft für die aktuelle Situation der Datenformate im Timetabling. Bislang liegt der Fokus dieser Formate auf dem Educational Timetabling. Zwar wird zumeist eine Allgemeingültigkeit der Datenformate gefordert, doch die Besonderheiten des hochflexiblen WFM – z.B. kleine Zeitslots, untertägige Auftrags- oder Arbeitsplatzwechsel – blieben weitestgehend unberücksichtigt. Daher soll ein Datenformat für die Abbildung von Problemen des hochflexiblen WFM erstellt werden. Dabei soll kein komplett neues Format entste-

hen. Vielmehr hat es sich gezeigt, dass das Format der ASAP (Automated Scheduling, Optimisation and Planning Research Group) bereits recht gut geeignet ist, sodass es erweitert wird.

Im Folgenden wird zunächst ein Überblick über verschiedene Datenformate gegeben, die bisher entwickelt wurden. Wegen ihrer Relevanz für das neue Datenformat wird gesondert auf TTML und auf das Format der ASAP eingegangen. Das neue Datenformat soll gewissen Anforderungen genügen. Diese werden in Kapitel 5.2.2 vorgestellt. Anschließend werden die Modifikationen am Format der ASAP beschrieben. In Anhang 7.3 ist eine Problemstellung im neuen XML-Format beispielhaft aufgeführt.

5.2.1 Existierende Datenformate

In diesem Kapitel wird ein Überblick über verschiedene Datenformate für Planungsprobleme gegeben, wobei besonders auf zwei der vielversprechendsten Formate eingegangen wird.

5.2.1.1 Allgemeiner Überblick

In der Vergangenheit wurde eine Vielzahl an Formaten entwickelt. Doch bisher hat sich keine universelle Sprache zur Abbildung von Timetabling-Problemen durchgesetzt [324, S. 323]. Anfangs verwendete man kommaseparierte oder tabellarische Formaten, die aus mehreren Dateien bestehen [258, S. 11–19]. Im Rahmen von Bestrebungen zur Schaffung eines einheitlichen Formates wurden einfache Grammatiken formuliert [59], [80]. Weitere Entwicklungen zeigen einen Trend hin zu XML-Formaten wie Tablix [351], [352], TTML [431] oder das Format der ASAP [88]. Zumeist liegt der Fokus dieser Formate auf dem Educational Timetabling. Nachfolgend sind verschiedene Datenformate inkl. Quellen aufgelistet, die bisher besondere Beachtung fanden:

- Format der ASAP [88],
- STTL [212], [317],
- Tablix [351], [352],
- TTML [431],
- TTL [80],
- UniLang [324].

In den beiden folgenden Kapiteln wird etwas tiefer auf TTML und das Format der ASAP geblickt, da sie am besten geeignet für die Abbildung von Problemen des hochflexiblen WFM erscheinen.

5.2.1.2 TTML: Timetabling Markup Language

TTML ist ein sehr flexibles Datenformat in XML [431], das auf MathML [331] basiert. Obwohl es vornehmlich auf Educational Timetabling fokussiert, können auch Probleme des WFM abgebildet werden. Die Struktur von TTML setzt ein gewisses Problem- und Modellierungswissen des Anwenders voraus. Es genügt nicht, einfach Mitarbeiter, Qualifikationen, Bedarfe, usw. anzugeben. Vielmehr muss auch exakt spezifiziert werden, was Variablen und Ressourcen sind und wie die Zeit abgebildet werden soll. Des Weiteren werden Nebenbedingungen sehr detailliert hinterlegt. Zum einen schließt dies Missverständnisse zwischen Anwendern aus. Zum anderen sind TTML-Dateien auf den ersten Blick nicht leicht verständlich und deren Formulierung ist mit erheblichem Aufwand verbunden.

Obwohl dieses Datenformat bereits in 2003 erstmalig vorgestellt wurde, hat es bisher kaum Anwendung gefunden. Eigene Erfahrungen haben gezeigt, dass die manuelle Abbildung einer Problemstellung in TTML etwa doppelt so lange dauert wie mit dem Format der ASAP, welches nachfolgend beschrieben wird.

5.2.1.3 Format der ASAP

Die Wurzeln dieses Datenformates liegen im Nurse Rostering [50, S. 8–9], [391, S. 34–66]. Aus diesem Grund weist es einige Schwächen in Bezug auf eine arbeitsplatzbezogene und untertägige Personaleinsatzplanung auf. Im Nurse Rostering werden Mitarbeitern zumeist ganztägig Schichten zugewiesen. Mit einigen Modifikationen kann das XML-Format jedoch leicht erweitert werden, sodass diese Mängel abgestellt werden. Das Datenformat ist leicht verständlich und neue Problemstellungen lassen sich relativ einfach abbilden. Besonders positiv fallen ein zugehöriges Softwaresystem für das Nurse Rostering sowie die umfangreichen Dokumentationen, Publikationen und Echtdaten ins Gewicht, die für dieses Datenformat zur Verfügung stehen [88]. Des Weiteren werden diese Ressourcen intensiv von der ASAP gepflegt, sodass eine langfristige Betreuung und ein großer Bekanntheitsgrad gewährleistet sind.

Aus den zuvor genannten Gründen wird dieses Format für die Abbildung von Problemen des hochflexiblen WFM präferiert. In Kapitel 5.2.3 werden die Modifikationen beschrieben, um vor allem eine arbeitsplatzbezogene und untertägige Personaleinsatzplanung zu

ermöglichen. Zunächst werden jedoch die Anforderungen an das neue Datenformat in Kapitel 5.2.2 dargestellt.

5.2.2 Anforderungen an ein Datenformat

Das Datenformat für Problemstellungen des hochflexiblen WFM muss bestimmten Anforderungen genügen. Diese wurden schon von Burke u.a. [59, S. 214–215], Cumming und Paechter [86] sowie von Reis und Oliveira [324, S. 327–328] gefordert und lauten:

- Unabhängigkeit von Implementierungsdetails und Lösungsverfahren.
- Allgemeingültigkeit.
- Leichte Überführung existierender Problemstellungen in das neue Format.
- Leichte Erweiterbarkeit, die die Abwärtskompatibilität nicht einschränkt.
- Leicht lesbar und verständlich für den Anwender.
- Einfache Syntax und dabei so präzise wie möglich.
- Große Bekanntheit.

Die Allgemeingültigkeit des Formates, wie sie bereits von Burke u.a. [59, S. 214–215] gefordert wird, soll ebenfalls für Problemstellungen aus dem Bereich des hochflexiblen WFM bestehen. Eine Ausweitung auf alle Formen des Timetablings ist nicht angedacht. Dies würde die Komplexität deutlich erhöhen, da sich die verschiedenen Arten des Timetablings stark unterscheiden können. Ein großer Bekanntheitsgrad des Datenformates soll durch die Veröffentlichung einer Suite mit Beispielpunkten und Benchmarkdaten im Internet erreicht werden. Diese Suite beinhaltet die verschiedensten Probleme, die sich hinsichtlich der Anzahl an Mitarbeitern und Arbeitsplätzen sowie hinsichtlich Nebenbedingungen und Branchen unterscheiden. Sie ist auf der Homepage des Fachgebietes Wirtschaftsinformatik für Dienstleistungen an der TU Ilmenau eingestellt [153] und auch die Homepage der ASAP [88] verlinkt auf diese Suite.

Der von Cumming und Paechter [86] sowie von Reis und Oliveira [324, S. 327] genannten Forderung, dass auf Informationen verzichtet wird, die nicht in direktem Zusammenhang mit der Problemstellung stehen (z.B. Namen der Mitarbeiter), wird nicht vollständig nachgekommen. Vielmehr soll optional die Möglichkeit bestehen, Zusatzinformationen bei der Problemformulierung anzugeben. Für eine spätere Lösungsdarstellung kann es durchaus sinnvoll sein, ergänzende Informationen mit auszugeben.

Alle anderen Forderungen an das Datenformat werden bei dessen Konstruktion berücksichtigt. Dabei wird XML genutzt, was eine strukturierte und flexible Problemformulierung ermöglicht. Auch die Überführung bestehender Problemstellungen von einem Format in ein anderes ist dank XML gut handhabbar. Zudem sind XML-Dokumente plattformunabhängig, textorientiert und relativ leicht verständlich. Im nächsten Kapitel werden die erforderlichen Erweiterungen des Formates der ASAP beschrieben, um die Problemstellungen dieser Arbeit abbilden zu können.

5.2.3 Erweiterung des Datenformates der ASAP

Nachfolgend werden lediglich die Erweiterungen des Datenformates der ASAP erläutert. Eine Beschreibung aller Elemente ist der umfassenden HTML-Dokumentation [153] zu entnehmen, die ebenfalls auf der Homepage des Fachgebietes Wirtschaftsinformatik für Dienstleistungen an der TU Ilmenau eingestellt ist. In Anhang 7.3 ist beispielhaft ein Problem im erweiterten XML-Format der ASAP zu finden. Die hier dargestellten Erweiterungen ermöglichen nun die Abbildung von Problemen des hochflexiblen WFM. Umfassende Änderungen wurden auch mit der Einführung von Arbeitsplätzen vorgenommen.

Zusatzinformationen Einleitend werden Informationen zum Autor, Erstellungsdatum, Beschreibung des Planungsproblems sowie zu Quellen ergänzt. Eine spätere Nutzung der abgebildeten Problemstellungen durch Dritte soll so vereinfacht werden. Die Syntax für Zusatzinformationen ist in TTML sehr gut umgesetzt und wird daher übernommen. Zusatzinformationen werden im Tag `<MetaInformation>` eingeschlossen.

Zeit Für untertägige Planungsprobleme ist es erforderlich, eine Vielzahl an Zeitintervallen pro Tag darzustellen. Das Format der ASAP bietet hierzu den Tag `<Period>` in `<DayPeriods>`. Man will damit den Personalbedarf von Schichten entkoppeln und ihn für einzelne Perioden angeben. Im Zuge der Verplanung von Mitarbeitern auf Schichten, die aus einzelnen Perioden bestehen, sollen flexiblere Einsatzpläne mit weniger Leerzeiten entstehen [391, S. 37–38, 161–178]. Schon sehr kleine untertägige Planungsprobleme werden mit diesem Konstrukt jedoch unübersichtlich, sehr umfangreich und schwer zu lesen. Da dies den zuvor genannten Anforderungen an ein Datenformat nicht genügt, wird eine andere Form der Umsetzung gewählt. Im Tag `<DayPeriods>` kann nun `<Date>` stehen. Für einen oder mehrere Tage wird hier die Anzahl an Intervallen und die Intervalllänge angegeben. Sollte es unterschiedlich lange Intervalle in den Rohdaten geben, so werden sie im Vorfeld auf das kleinste gemeinsame Vielfache ihrer Länge reduziert.

Arbeitsplätze Das Konzept der ASAP sieht folgender Maßen aus: Um einen definierten Bedarf an Mitarbeitern zu decken, werden diese auf zuvor definierten Schichten verplant. Der Personalbedarf kann für Schichten, Schichtgruppen, Zeitperioden und Qualifikationen bestehen. Arbeitsplätze existieren im Datenformat der ASAP nicht. Daher ist deren Konzept nur dann unproblematisch anwendbar, wenn es sich lediglich um die Abbildung eines einzigen Arbeitsplatzes handelt. Bei mehr als einem Arbeitsplatz stößt das Konzept an Grenzen, da verschiedene Arbeitsplätze, bei denen bestimmte Schichten zulässig sind, nicht problemlos abgebildet werden können.

Theoretisch bestünde die Möglichkeit, Schichten nur für bestimmte Arbeitsplätze zu definieren. Eine konkrete Bindung der Schicht an den jeweiligen Arbeitsplatz erfolgt dabei nicht, da diese im Datenformat nicht existieren. Die Zuweisung ist lediglich ein gedanklicher Prozess. Ein Beispiel: Es gibt zwei Arbeitsplätze und eine Nachtschicht. Man würde z.B. die beiden Nachtschichten N1 und N2 erstellen, um die beiden Arbeitsplätze getrennt abzubilden. Dieses Vorgehen ist jedoch nicht optimal, da es sehr komplex ist und durch die Bindung des Personalbedarfs an Schichten die Realität nicht immer korrekt wiedergibt. Gerade im Handel wird der Personalbedarf für Arbeitsplätze nämlich je Zeitintervall ermittelt.

Eine Koppelung des Personalbedarfs an Schichten macht nur dann Sinn, wenn im Vorfeld ein Workforce Scheduling [278], [144] stattgefunden hat. In der Praxis ist das jedoch nicht immer der Fall und der eher langfristige Planungscharakter erlaubt keine kurzfristigen Arbeitsplatzwechsel, wie sie bei der untertägigen Personaleinsatzplanung zu finden sind. Zudem werden Mitarbeiter nach dem Konzept der ASAP nur einer Schicht pro Tag zugeteilt – selbst das zuvor geschilderte Vorgehen würde keine untertägigen Arbeitsplatzwechsel ermöglichen.

Die Definition von Arbeitsplätzen soll nachfolgend so flexibel gestaltet sein, dass sie auch als Aufträge im Rahmen des Technikereinsatzes aufgefasst werden können. In diesem Fall werden zusätzlich Geoinformationen und Ausführungszeiten ergänzt.

Aus diesen Gründen wird das Modell der ASAP um Arbeitsplätze erweitert. Schichten und Arbeitsplätze sind somit getrennt. Es gibt Arbeitsplätze, die bestimmte Qualifikationen erfordern können und für die ein Personalbedarf definiert wird. Zudem haben sie einen Namen, eine Abkürzung und eine eindeutige Identifikationsnummer. Auf der anderen Seite gibt es Schichten, die für bestimmte Arbeitsplätze zulässig sind. In der Schicht werden Pausen, Kernzeiten sowie Arbeitsbeginn und -ende festgelegt.

Sondertage Bisher ist es lediglich möglich, bestimmte Tage als Wochenenden festzulegen und Tage als Feiertage anzugeben. Zusätzlich sollen Sondertage festgelegt werden können. Hilfreich sind diese Sondertage, wenn es besonderen Ereignissen gibt. Beispielsweise ein Tag mit verlängerten Öffnungszeiten, Tage, an denen kein Urlaub genommen

werden darf, Messetage, etc.

Nebenbedingungen Nebenbedingungen können global in <MasterWeights> oder individuell für Mitarbeiter in <Contracts> hinterlegt werden. Individuelle Nebenbedingungen dominieren die globalen Nebenbedingungen. Einige Nebenbedingungen z.B. für Über- und Unterbesetzungen werden direkt in <MasterWeights> hinterlegt.

Nebenbedingungen können im Datenformat der ASAP nicht als hart oder weich definiert werden, da einige von vornherein als hart angesehen werden [50, S. 4], [52, S. 4], [51, S. 6]. Dieses Vorgehen ist nicht flexibel genug. Aus diesem Grund wird zukünftig im jeweiligen Tag für die Nebenbedingung mit „Type=hard/soft“ angegeben, ob es sich um harte oder weiche Nebenbedingungen handelt. Wird nichts angegeben, so wird „soft“ unterstellt.

Mitarbeiter sind so einzuplanen, dass der Personalbedarf nicht überschritten wird. Es kann im Format der ASAP eine Maximalbesetzung oder eine präferierte Besetzungsvorgabe als Bezugsgröße dienen. Sollten zu viele Mitarbeiter eingeplant sein, wird dort jedoch nicht unterschieden, ob ein Bedarf von Null oder mehr als Null Mitarbeitern vorliegt. Allgemein ist eine Überplanung der Besetzungsvorgabe zu vermeiden. Jedoch wiegt die Einplanung eines Mitarbeiters auf einen Arbeitsplatz, bei dem rein gar nichts zu tun ist, schwerer als eine Überplanung bei einem Arbeitsplatz mit einem Bedarf größer als Null. In den <MasterWeights> wird dieser Zusammenhang mit den zusätzlichen Tags <PrefOverStaffingNoRequirements> und <MaxOverStaffingNoRequirements> abgebildet. Es werden Überdeckungen von der präferierten und von der maximal zulässigen Besetzung berücksichtigt. Ebenso werden Tags für Stammarbeitsplatzfehler, Fahrtzeiten, etc. eingefügt.

Dank der Modifikationen bei <DayPeriods> ist es nun möglich, Über- und Unterbesetzungen einfach und komfortabel zu bewerten. Für jede Abweichung von der Besetzungsvorgabe fallen Strafpunkte je Zeitintervall an, wobei auch die Länge des Zeitintervalls berücksichtigt wird. Bei den entsprechenden Nebenbedingungen kann mit Unit=„sek/min/hour/day“ die geltende Zeiteinheit angegeben werden. Dann gelten die angegebenen Fehlerpunkte für jede Zeiteinheit, in der gegen die Bedingung verstoßen wird. Ansonsten beziehen sich die Fehlerpunkte auf die Tatsache eines Verstoßes. Also auf den jeweiligen Zeitintervall (der in der Problembeschreibung angegeben ist – nicht der aus einem später erzeugten zeitdiskreten Modell). Beispielsweise entstehen 200 Fehlerpunkte, wenn zwei Mitarbeiter zu viel in einem einzigen Zeitslot ohne angegebene Zeiteinheit arbeiten (bei 100 Fehlerpunkten). Wird eine Zeiteinheit angegeben und die Dauer der Über- bzw. Unterbesetzung entspricht einem nicht ganzzahligen Vielfachen dieser Zeiteinheit, so wird diese auf ein ganzzahliges Vielfaches gerundet (z.B. Unit=„hour“: 110 Minuten = 2 Stunden; 29 Minuten = 0 Stunden). Anschließend werden die Fehlerpunkte ermittelt.

Bei den `<MasterWeights>` wird der Tag `<NoSkill>` ergänzt. Eine Nebenbedingung wird verletzt, wenn der Mitarbeiter wenigstens eine der geforderten Qualifikationen nicht hat.

Es gibt Nebenbedingungen, die die Anzahl der Schichtwechsel berücksichtigen, jedoch keine, die sich auf Arbeitsplätze beziehen. Daher werden `<MinWorkstationsPerDay>` und `<MaxWorkstationsPerDay>` ergänzt. Diese Nebenbedingungen können global in `<MasterWeights>` oder mitarbeiterindividuell in `<Contracts>` stehen.

Qualifikationen Tätigkeiten erfordern oft bestimmte Qualifikationen, die Mitarbeiter erfüllen müssen, um auf ihnen eingeplant werden zu können. Nur in wenigen Situationen werden Planungsbereiche mit Mitarbeitern mit sehr homogener Qualifikationsstruktur besetzt, so dass es von dieser Seite her relativ gleichgültig ist, welcher Mitarbeiter auf welchem Arbeitsplatz arbeitet.

Das bloße Vorhandensein einer Qualifikation genügt in der Praxis nicht immer. Daher wird die Ausprägung einer Qualifikation als optionale Größe ergänzt. So müssen keine zusätzlichen Qualifikationen angelegt werden, die höhere Fähigkeitsniveaus symbolisieren und das Planungsproblem schnell unübersichtlich sowie die Handhabung wenig praktikabel machen. Mit Hilfe des Ausprägungsgrades ist es möglich, den jeweils am besten geeigneten Mitarbeiter einzusetzen, eine individuelle Bearbeitungszeit zu ermitteln oder eine Mindest- oder Maximalausprägung für einen Arbeitsplatz zu fordern. Zudem erhält jede Qualifikation eine Identifikationsnummer, sowie eine Lang- und eine Kurzbezeichnung. Die Identifikationsnummer kann an einen Mitarbeiter, einen Arbeitsplatz, usw. geknüpft werden und die zusätzlichen Bezeichnungen dienen der übersichtlicheren Ausgabe.

Personalbedarf Der Bedarf an Personal kann für Schichten, Schichtgruppen, Zeitintervalle und Qualifikationen in `<CoverRequirements>` angegeben werden. Mit der Einführung von Arbeitsplätzen ist nun auch die Möglichkeit gegeben, Bedarfe für Arbeitsplätze (bzw. Aufträge) festzulegen. Dies erfolgt in `<Cover>` für die jeweilige `<WorkstationID>`. So kann auch die Dimension Zeitintervall und Arbeitsplatz (bzw. Aufträge) zusammengeführt werden. Z.B. auf Arbeitsplatz eins ein Bedarf von fünf Mitarbeitern von 10–11 Uhr, von 2 Mitarbeitern von 11–12 Uhr etc.

Präferenzen Für Mitarbeiter können bisher Präferenzen für Schichten und Tage angegeben werden, zu denen sie arbeiten oder nicht arbeiten möchten. Für eine bessere Spezifikation der Präferenzen macht es Sinn, nicht nur Schichten anzugeben, sondern optional den gewünschten Arbeitsbeginn und/oder das Arbeitsende. Diese Zeiten schränken dann

die Schichtauswahl entsprechend ein und führen zu Strafpunkten, wenn sie verletzt werden.

Ganztägige Zuweisungen Sollte die untertägige Personaleinsatzplanung zur Komplexitätsreduzierung im Anschluss an eine ganztägige Planung erfolgen (siehe Kapitel 3.2), sind für die untertägige Planung einige notwendige Informationen zu hinterlegen. Hier kann die schon existierende Syntax der ASAP für die Abbildung der Lösung herangezogen werden. In <Assignments> wird jedem Mitarbeiter der geplante ganztägige Stammarbeitsplatz über die Arbeitsplatz-ID zugewiesen und über die zugeteilten Schichten sind Pausen, Arbeitsbeginn und -ende sowie An- und Abwesenheit der Mitarbeiter bekannt.

5.3 Statistische Grundlagen

Nachfolgend werden einige Grundlagen und verschiedene statistische Verfahren beschrieben. Die dargestellten Verfahren in den Kapiteln 5.3.2 und 5.3.3 dienen in dieser Arbeit vor allem der Beurteilung:

- welches von zwei gleichen Lösungsverfahren mit verschiedenen Parametereinstellungen im Mittel bessere Ergebnisse liefert
- oder welches von zwei verschiedenen Lösungsverfahren im Mittel bessere Ergebnisse liefert.

Der Verfahrensablauf ist bei beiden Untersuchungsgegenständen identisch und für die Umsetzung wird die Software PASW Statistics 18 (ehem. SPSS) genutzt. In Kapitel 5.3.4 wird abschließend auf die Berechnung des optimalen Stichprobenumfangs eingegangen. Der Aufbau der Kapitel 5.3.1 bis 5.3.4 orientiert sich an der Arbeit von Brodersen [46, S. 63–70] – die Ausführungen wurden um relevante Aspekte erweitert und vor allem auf die Anforderungen der hier durchgeführten Experimente angepasst.

5.3.1 Hypothesen und Irrtumswahrscheinlichkeiten

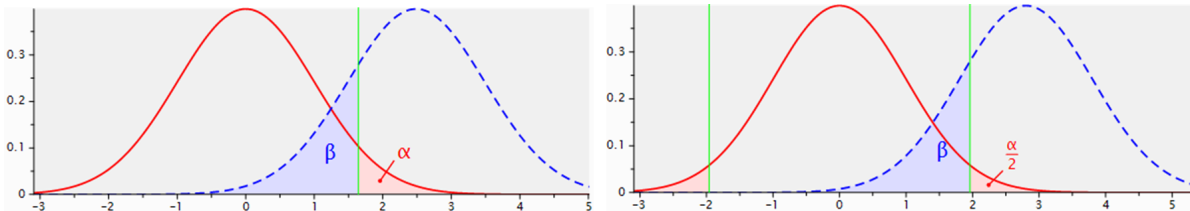
Gegenstand der nachfolgenden Erörterungen sind statistische Aussagen über Grundgesamtheiten, aus denen mittels Zufallsstichproben Beobachtungen erhoben werden. Um eine statistische Aussage über eine Grundgesamtheit zu beurteilen, wird eine Nullhypothese H_0 formuliert. Zum Beispiel „Die Mittelwerte zweier Zufallsstichproben sind gleich“ oder andere Aussagen über die Verteilung von Zufallsvariablen und deren Parameter. Kann in einem Signifikanztest die Nullhypothese nicht angenommen werden, so

wird man H_1 , die Alternativhypothese, als wahrscheinlich erachten. Zum Beispiel „Die Mittelwerte zweier Zufallsstichproben sind verschieden“. Die Wahrscheinlichkeit, mit der H_0 abgelehnt wird, obwohl sie wahr ist, wird als Fehler 1. Art bzw. als α -Fehler bezeichnet. Üblicherweise wird für α 0,1, 0,05 oder 0,01 gewählt. Die Annahme von H_0 , obwohl sie falsch ist, wird hingegen als Fehler 2. Art bzw. als β -Fehler bezeichnet. Tabelle 5.1 veranschaulicht diesen Zusammenhang.

Tabelle 5.1: Fehlerarten bei Hypothesentests nach [46, S. 64]

	H_0 ablehnen	H_0 nicht ablehnen
H_0 ist wahr	✗ Fehler 1. Art (α)	✓ ($1 - \alpha$)
H_1 ist wahr	✓ ($1 - \beta$)	✗ Fehler 2. Art (β)

In Abbildung 5.2 werden die Ablehnungsbereiche des einseitigen und des zweiseitigen Hypothesentests grafisch veranschaulicht.

Abbildung 5.2: Ablehnungsbereiche für H_0 bei einseitigem (links) und zweiseitigem (rechts) Test

In Abhängigkeit davon, ob eine Hypothese gerichtet oder ungerichtet ist, wird der ein- oder zweiseitige Hypothesentest durchgeführt. Ein einseitiger t -Test, der in Kapitel 5.3.3 näher beleuchtet wird, hätte die folgende Null- und Alternativhypothese. Er kann verwendet werden, wenn bereits Informationen über die Mittelwerte vorhanden sind, was bei den Experimenten dieser Arbeit der Fall ist.

- $H_0 : \mu_1 = \mu_2$.
- $H_1 : \mu_1 < \mu_2$.

Im Gegensatz dazu hätte ein zweiseitiger Test zum Beispiel die folgende Null- und Alternativhypothese:

- $H_0 : \mu_1 = \mu_2$.
- $H_1 : \mu_1 \neq \mu_2$.

Für weitere Ausführungen sei an dieser Stelle auf die Arbeiten von Bortz und Döring [41] sowie von Bortz und Weber [42] verwiesen.

Nachfolgend wird der Levene-Test beschrieben, mit dem die Varianzhomogenität geprüft wird. Die Varianzhomogenität ist für den t -Test relevant, der in Kapitel 5.3.3 beschrieben wird.

5.3.2 Levene-Test

Grundsätzlich sollte bei einem t -Test die Kenntnis über eine vorhandene Varianzhomogenität in die Ausführung des Tests einbezogen werden, da ein solcher Test teststärker ist als ein Test, bei dem keine Varianzhomogenität zugrunde liegt. Das Aufdecken eines Unterschiedes in den Mitten der Grundgesamtheiten kann mit einem t -Test, der auf der Voraussetzung der Varianzhomogenität basiert, eher erfolgen. Mit dem Levene-Test kann geprüft werden, ob die Varianzen zweier oder mehrere Gruppen gleich sind. Die Prüfgröße W des Signifikanztests ist F -verteilt mit $k - 1$ und $n - k$ Freiheitsgraden:

$$W = \frac{(n - k) \sum_{i=1}^k n_i (Z_{i.} - Z_{..})^2}{(k - 1) \sum_{i=1}^k \sum_{j=1}^{n_i} (Z_{ij} - Z_{i.})^2} \quad \text{mit} \quad (5.1)$$

- n : Umfang der Stichprobe,
- k : Umfang der Unterstichproben (Gruppen) n_i ,
- $Z_{ij} = |Y_{ij} - \bar{Y}_i|$ mit Y_i als dem arithmetischen Mittel der i -ten Gruppe,
- $Z_{i.}$: die Gruppenmittelwerte von Z_{ij} ,
- $Z_{..}$: die Gruppenmittelwerte von Z_{ij} über alle Gruppen.

Beispielhaft soll nun der Levene-Test durchgeführt werden. Beim untertägigen Personaleinsatzplanungsproblem des Logistikdienstleisters wurden in Experiment eins 30 Wiederholungen mit einem PSO-Algorithmus durchgeführt, der eine Schwarmgröße von 20 Partikeln hat. Dabei wurden folgende Ergebnisse erzielt.

71.035, 69.680, 70.201, 69.715, 69.483, 69.780, 69.821, 69.113, 70.686, 69.772, 69.929, 69.311, 70.197, 70.438, 68.049, 68.978, 70.311, 69.304, 70.050, 69.517, 70.221, 69.184, 69.823, 69204, 70067, 70405, 69.653, 70.167, 69.120, 70.024

Bei 30 weiteren Wiederholungen des PSO-Algorithmus mit 40 Partikeln an derselben Problemstellung wurden in Experiment zwei folgende Ergebnisse ermittelt.

79.846, 87.763, 78.515, 90.682, 84.233, 81.860, 78.172, 83.599, 81.657, 87.828, 88.721, 91.587, 87.637, 88.215, 79.188, 82.109, 97.376, 89.675, 85.592, 94.058, 85.947, 82.403, 79.658, 88.639, 87.694, 79.639, 84.644, 79.851, 81.523, 79.320

Die Null- und Alternativhypothesen lauten:

- $H_0 : \sigma_1^2 = \sigma_2^2$.
- $H_1 : \sigma_1^2 \neq \sigma_2^2$.

Für den Test wird das Signifikanzniveau α auf 0,05 gesetzt. Der Levene-Test liefert als Prüfgröße den Wert $L = 60,086$ und als p -Wert 0,0. Wegen $0,0 < 0,05$ muss die Nullhypothese, dass die Varianzen gleich sind, abgelehnt werden.

In dieser Arbeit wird nachfolgend ein Testniveau von 95% für den Levene-Test vorgegeben. Dieses wird im Rahmen von t -Tests daher nicht immer mit angegeben. In Abhängigkeit davon, ob Varianzhomogenität oder -inhomogenität vorliegt, werden die relevanten Werte für den t -Test in PASW abgelesen. Für vertiefende Ausführungen zum Levene-Test und zu dessen Robustheit sei auf die Arbeiten von Brown und Forsythe [49], Levene [243] sowie Lim und Loh [246] verwiesen.

5.3.3 t -Test für unabhängige Stichproben

Mit einem Zwei-Stichproben- t -Test kann geprüft werden, ob die Mittelwerte zweier unabhängiger Stichproben x_1, \dots, x_n und y_1, \dots, y_m als identisch angesehen werden können. Voraussetzung für die Nutzung des t -Tests sind Stichproben aus normalverteilten Grundgesamtheiten. Da eine gleiche, aber unbekannte Varianz der Stichproben unterstellt wird, ergibt sich die gewichtete Varianz aus:

$$s^2 = \frac{(n-1)s_x^2 + (m-1)s_y^2}{n+m-2}. \quad (5.2)$$

Die Prüfgröße ist t -verteilt mit $n+m-2$ Freiheitsgraden und lautet:

$$t = \sqrt{\frac{nm}{n+m}} \frac{\bar{x} - \bar{y}}{s}. \quad (5.3)$$

Da die Experimente in dieser Arbeit mit jeweils gleicher Anzahl an Wiederholungen erfolgen, vereinfacht sich die Formel zur Berechnung der Prüfgröße:

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{(s_x^2 + s_y^2)/n}}. \quad (5.4)$$

Die in Kapitel 5.3.2 aufgeführten Experimente sollen beispielhaft einem einseitigen t -Test unterzogen werden. In Kapitel 5.3.2 konnte gezeigt werden, dass keine Varianzhomogenität vorliegt. PASW stellt im Falle der Varianzhomogenität ein Korrekturverfahren zur Verfügung, sodass die Ergebnisse des t -Tests dennoch interpretiert werden können [210, S. 395]. Mit einem Shapiro-Wilk-Test wurde zudem gezeigt, dass das Testergebnis nicht gegen die Annahme normalverteilter Stichproben spricht. Auf den Shapiro-Wilk-Test wird in dieser Arbeit nicht eingegangen. Die folgende Tabelle stellt die Mittelwerte und die Standardabweichungen beider Experimente dar.

Tabelle 5.2: Mittelwert und Standardabweichung der Experimente eins und zwei

Experiment	Beobachtungen	Mittelwert	Standardabweichung
Experiment 1 (PSO 20)	30	69.774,60	595,32
Experiment 2 (PSO 40)	30	84.921,03	4.972,50

Die absolute Differenz der Mittelwerte beträgt 15.146,43. Da eine Aussage über die Mittelwerte der Experimente getroffen werden kann, kann ein einseitiger t -Test vorgenommen werden. Die Null- und Alternativhypothese des einseitigen t -Tests lautet:

- $H_0 : \mu_1 = \mu_2$.
- $H_1 : \mu_1 < \mu_2$.

Als t -Wert ergibt sich -16,57 und als p -Wert 0,0. Bei einem Signifikanzniveau von 95% ist der Mittelwertunterschied signifikant. Da die Alternativhypothese in einem einseitigen t -Test akzeptiert wird, kann geschlossen werden, dass Experiment eins im Mittel kleinere Mittelwerte liefert als Experiment zwei.

5.3.4 Poweranalyse

Für die Poweranalyse wird in dieser Arbeit die kostenfreie Software G*Power 3 verwendet, mit der a priori sowie post hoc Analysen durchgeführt werden können [117]. Beim t -Test für unabhängige Stichproben wurde in Kapitel 5.3.3 bisher nur eine Aussage darüber gemacht, ob unter der Nullhypothese die Mittelwerte identisch oder ob unter der Alternativhypothese beide Mittelwerte verschieden sind (bzw. einer größer oder kleiner ist). Um eine Aussage darüber zu treffen, ab welcher Differenz zweier Mittelwerte der eine als besser angesehen werden kann, wird die Effektgröße d eingeführt. Die spezifische Alternativhypothese lautet:

- $H_1 : \mu_1 < \mu_2 + d$.

Nach Cohen wird bei $|d| = 0,2$ von einem schwachen (kleinen) Effekt, bei $|d| = 0,5$ von einem mittleren Effekt und bei $|d| = 0,8$ von einem starken (großen) Effekt gesprochen [72]. Tabelle 5.3 zeigt für $\alpha = 0,05$ und $\epsilon = 0,8$ die erforderlichen Stichprobenumfänge bei einseitigem t -Test mit unabhängigen Stichproben in Abhängigkeit von d . Dabei wird ϵ als Power oder Teststärke bezeichnet und entspricht der Gegenwahrscheinlichkeit einen Fehler 2. Art zu begehen ($1-\beta$). Gewöhnlich wird $\epsilon = 0,8$ gewählt [107]. Der Stichprobenumfang wird in Tabelle 5.3 mit dem Faktor zwei angegeben, da zwei Stichproben gezogen werden.

Tabelle 5.3: Einfluss verschiedener Effektgrößen auf den Stichprobenumfang bei einseitigem t -Test der Mittelwerte mit unabhängigen Stichproben sowie $\alpha = 0,05$ und $\epsilon = 0,8$

Effektgröße d	Stichprobenumfang	Effektgröße d	Stichprobenumfang
0,1	2 * 1.238	0,6	2 * 36
0,2	2 * 310	0,7	2 * 26
0,3	2 * 139	0,8	2 * 21
0,4	2 * 78	0,9	2 * 16
0,5	2 * 51	1,0	2 * 14

Ein Signifikanztest sollte nicht auf Basis einer Gelegenheitsstichprobe durchgeführt werden, sondern auf Basis einer Zufallsstichprobe mit optimalem Stichprobenumfang [210, S. 319–323]. Ist die Effektgröße, das Testniveau und die Teststärke vorgegeben, lässt sich der optimale Stichprobenumfang a priori mit einer Poweranalyse berechnen. So können die Risiken, einen Fehler 1. oder 2. Art zu begehen, schon im Vorhinein minimiert werden. In Abbildung 5.3 ist für $|d| = 0,65$, $\alpha = 0,05$ und $\epsilon = 0,8$ die Berechnung des Stichprobenumfanges mit G*Power 3 dargestellt, der für alle Experimente dieser Arbeit verwendet wird. Für alle Experimente dieser Arbeit ergibt sich somit ein Stichprobenumfang von 30 Replikationen.

Dem Beispiel in Kapitel 5.3.3 konnte entnommen werden, dass Experiment eins im Mittel kleinere Mittelwerte liefert als Experiment zwei. Die Nullhypothese wurde also verworfen. An dieser Stelle interessiert nun, mit welchem Risiko man die Nullhypothese bei einem Stichprobenumfang von n_1 und $n_2 = 30$ fälschlicher Weise akzeptiert hätte (Fehler 2. Art). In einer post hoc Analyse mit G*Power 3 ergibt sich für die im Experiment ermittelte Effektgröße $|d| = 4,28$ und unter Beibehaltung aller übrigen Parameter eine Wahrscheinlichkeit für einen Fehler 2. Art von 0,0%. D.h., dass das Risiko, die Nullhypothese unter den gegebenen Rahmenbedingungen fälschlicher Weise zu akzeptieren, gegen null geht.

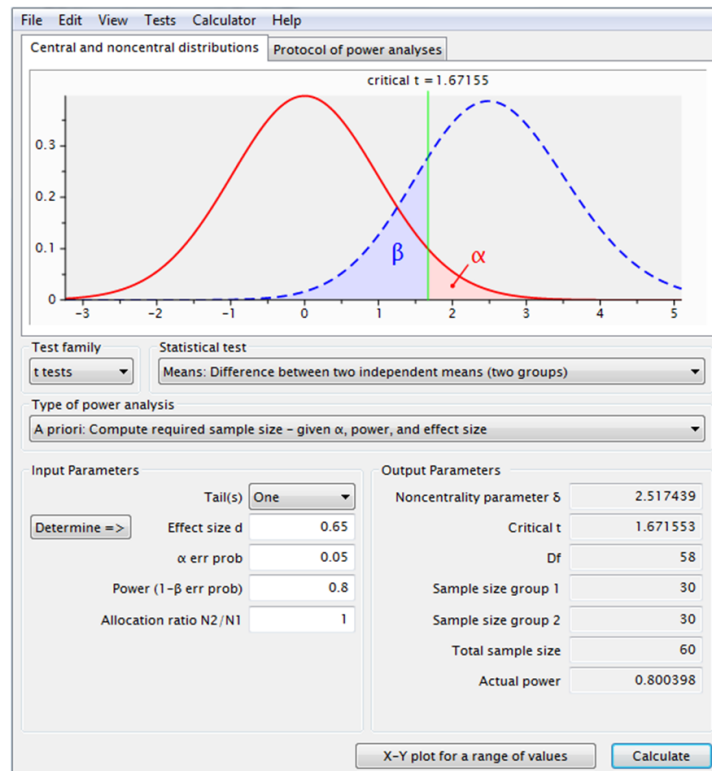


Abbildung 5.3: Stichprobenumfang (n_1 und $n_2 = 30$) bei einseitigem t -Test der Mittelwerte mit unabhängigen Stichproben

5.4 Beurteilungskriterien

Das Ergebnis eines Vergleichs von Lösungsverfahren wird durch die Wahl der Beurteilungskriterien stark beeinflusst [286, S. 244]. Daher wird in diesem Kapitel ausführlich auf die verwendeten Kriterien eingegangen. Die nachfolgenden Ausführungen orientieren sich an den umfassenden Darstellungen von Sondergeld [354] und Brodersen [46, S. 71–75] zu diesem Thema. Sondergeld fasst messbare Beurteilungskriterien zusammen, die in Abbildung 5.4 dargestellt sind. In seinen Ausführungen berücksichtigt er die Arbeiten von Crowder, Dembo und Mulvey [85], [84], von Greenberg [168] sowie von Hooker [191].

Sondergeld und Brodersen gewichten die Accuracy gegenüber den anderen Kriterien in Abbildung 5.4 am stärksten. Biethahn u.a. nennen ähnliche Kriterien [33, S. 190], [34, S. 121–122] und wählen zur Beurteilung die ungewichteten Kriterien: Güte des erreichten Zielwertes, Rechenzeit, Programmieraufwand sowie die Sicherheit, das Optimum zu finden. Ausgehend von diesen Kriterien werden in dieser Arbeit die Beurteilungskriterien Exaktheit, Rechenzeit und Aufwand zum Vergleich der Lösungsverfahren verwendet. Diese werden nachfolgend beschrieben.

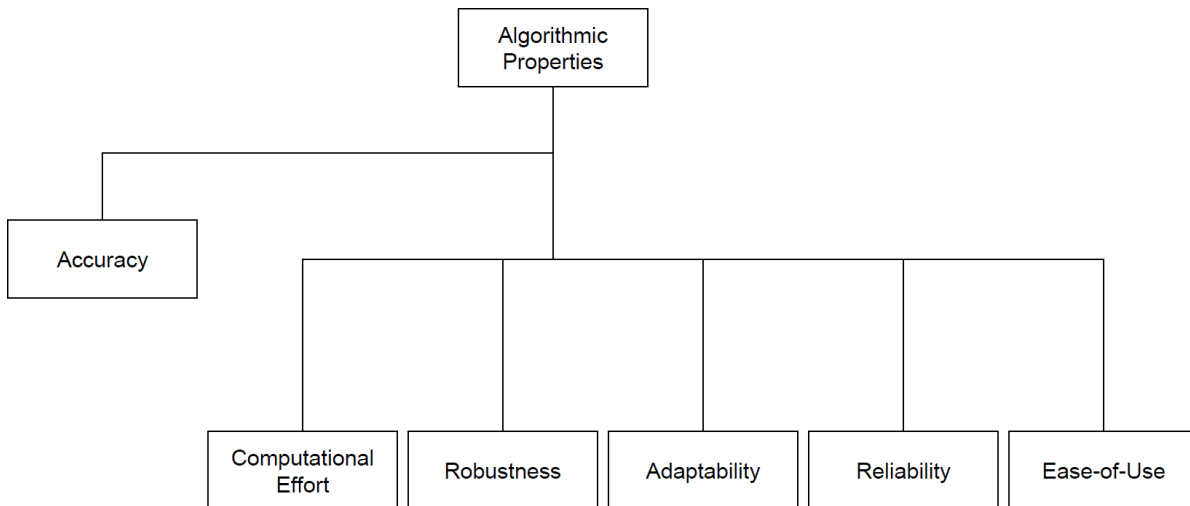


Abbildung 5.4: Beurteilungskriterien für Lösungsverfahren [354, S. 8]

Exaktheit Mit der Exaktheit wird ausgesagt, wie gut die gefundene Lösung des Algorithmus ist. Hierzu sind Anhaltspunkte notwendig. Daher werden Überlegungen zum Lower Bound der Problemstellung angestellt, was besonders beim Logistikproblem sehr gut möglich ist. Weiterhin werden Vergleiche mit bereits existierenden Referenzlösungen vorgenommen, wie sie z.B. für das Problem der British Telecom verfügbar sind.

Die Exaktheit eines Lösungsverfahrens hängt meist stark von der Intensität ab, mit der es an die jeweilige Problemstellung angepasst wurde [191, S. 35]. Die TS-Ergebnisse und einige Grundlagen von ES stammen aus Diplomarbeiten. Die Autoren haben die entsprechenden Verfahren umfassend modifiziert und im Vorfeld der Experimente aufwändigen Tests unterzogen. Das Gleiche gilt für alle selbst implementierten Verfahren.

Für die Beurteilung der Exaktheit nennt Sondergeld verschiedene Ansätze [354]. Die Nachfolgenden werden in dieser Arbeit verwendet. Die Beurteilung erfolgt an den durchschnittlichen Fehlerpunkten (soweit vorhanden), wobei der Vergleichswert zur Berechnung der prozentualen Abweichung die bisher gefundenen minimalen Fehlerpunkte sind.

- Absolute Werte (minimale und durchschnittliche Fehlerpunkte).
- Abstand einer Lösung zu der besten bekannten Lösung (absolut und prozentual).
- Vergleich von Mittelwerten anhand von Ergebnismengen mit Hilfe von t-Tests.

Rechenzeit Wenn möglich wird die Rechenzeit in dieser Arbeit als Zeitmessung der CPU-Zeit bei einer vorgegebenen Anzahl an bewerteten Fitnessfunktionen (Abbruchkriterium) verstanden. Alle Berechnungen – außer zu TS am Logistikproblem und zum KV

von Sauer und Schumann – wurden mit einem Intel Core Quad 4 x 2,66 GHz und 4 GB RAM unter Windows Vista 64-Bit durchgeführt. Für TS ist lediglich bekannt, dass die Berechnungen auf handelsüblichen Rechnern durchgeführt wurden. Sauer und Schumann verwendeten für ihr KV einen Intel Core Duo 2 x 2,0 GHz und 3 GB RAM mit Windows Vista 32-Bit. Obwohl bei den Experimenten exakte Zeiten aufgezeichnet wurden, genügt eine weniger granulare Angabe völlig. Die Angabe der Rechenzeit in Sekunden würde eine Genauigkeit vorgaukeln, die so nicht vorhanden ist. Denn die Berechnungen wurden z.T. parallel ausgeführt, es wurden zusätzlich erhebliche Datenmengen während der Berechnungen gespeichert und diverse Programme im Hintergrund ausgeführt. All dies beeinflusst die CPU-Zeit nicht unwesentlich.

Aufwand Unter Aufwand wird zum einen verstanden, wie einfach das Lösungsverfahren an eine Problemstellung angepasst werden kann. Hierauf hat die Anzahl der einzustellenden Parameter einen Einfluss. Zum anderen wird unter diesem Beurteilungskriterium der Programmieraufwand verstanden, um ein Lösungsverfahren zu implementieren und auf ein Problem anzuwenden.

Die Beurteilung des Aufwands ist subjektiv. Da alle selbst implementierten Lösungsverfahren (außer die Ganzzahlige Optimierung in GAMS) in ARROW mit C# realisiert wurden, ist ein grober Vergleich dennoch gut möglich. Dabei wird unterstellt, dass der Algorithmus bei jeder Problemstellung jeweils komplett neu implementiert wird. Lediglich die Fitnessfunktion und die Problemstellung sind bereits vorhanden. Es wird also nur die Zeit gewertet, die für die Programmierung und die Anpassung des Algorithmus notwendig ist. Zudem wird davon ausgegangen, dass man sich sehr gut mit ARROW, C# und GAMS sowie mit den jeweiligen Lösungsverfahren und Problemstellungen auskennt.

Nachdem alle Voraussetzungen für Experimente geschaffen sind, werden im folgenden Kapitel die entsprechenden Ergebnisse vorgestellt und erörtert.

Kapitel 6

Experimente und Ergebnisse

In diesem Kapitel werden die Ergebnisse der verschiedenen Lösungsverfahren erläutert. Dabei wird ein Großteil der in Kapitel 1.1 vorgestellten Forschungsfragen beantwortet. Vor allem die zentrale Forschungsfrage dieser Arbeit, ob sich eher klassische OR-Verfahren, Metaheuristiken oder MAS für die drei beispielhaften Problemstellungen des hochflexiblen WFM eignen. In diesem Zusammenhang wird auch geprüft, ob sich die Aussage von Puppe u.a. [315], nach der sich zentrale Lösungsverfahren weniger gut für die Dienstplanung eignen als MAS (dezentral), auch auf die beispielhaft ausgewählten Problemstellungen des hochflexiblen WFM übertragen lässt. Bei allen Experimenten werden äußerst umfangreiche Parametertests durchgeführt. Daher wird in den nachfolgenden Kapiteln ebenfalls auf die Forschungsfrage eingegangen, welche Parametereinstellungen bei welchen der hier untersuchten Problemstellungen zu wählen sind. Da alle Lösungsverfahren an realen Problemstellungen erprobt werden, wird in diesem Kapitel die Forschungsfrage geklärt, wie gut sich die Verfahren für derartige Probleme eignen. Im Rahmen dieses Kapitels sind auch die erzielten betriebswirtschaftlichen Effekte durch hochflexibles WFM ableitbar, was eine weitere Forschungsfrage ist.

Für jede der drei Problemstellungen Logistik, Handel und Technikereinsatz bei der British Telecom existiert ein eigenes Unterkapitel, in dem separat auf die Ergebnisse der individuell angepassten Algorithmen eingegangen wird. Die vollständige Enumeration als Lösungsverfahren wird nachfolgend nicht mehr beleuchtet. Bereits in Kapitel 4.2 wurde gezeigt, dass sie für die Problemstellungen des hochflexiblen WFM aufgrund der unakzeptablen Rechenzeiten ungeeignet ist. Jedes der Unterkapitel endet mit einem zusammenfassenden Vergleich der Lösungsverfahren. Eine abschließende Beurteilung der verwendeten Lösungsverfahren über alle drei Problemstellungen hinweg ist in Kapitel 6.4 zu finden. Für eine kurze Übersicht, welches Verfahren bei welcher Problemstellung erprobt wird, sei auf Abbildung 4.3 verwiesen.

6.1 Untertägige Personaleinsatzplanung in der Logistik

Nachfolgend werden die Ergebnisse verschiedener Lösungsverfahren für das Logistikproblem dargestellt. Im Rahmen der Experimente wurden zahlreiche Daten gesammelt und aufbereitet. Ein Großteil dieser Daten ist im Anhang in den Tabellen 7.1 bis 7.32 zu finden, um dieses Kapitel nicht zu stark aufzublähen. Ergebnisse, die der Übersicht halber eine andere Formatierung haben oder nicht im Anhang stehen, sind im entsprechenden Kapitel dargestellt.

Die Ergebnisse zu TS stammen von Helbig im Rahmen seiner Diplomarbeit [178]. Birnstiel hat sich in seiner Diplomarbeit mit ES TypG für das Logistikproblem beschäftigt [35]. Basierend auf seiner Vorarbeit wurden die hier dargestellten Ergebnisse für ES TypG berechnet.

Bevor auf die Ergebnisse der einzelnen Lösungsverfahren eingegangen wird, erfolgt zunächst eine Betrachtung des Lower Bounds.

6.1.1 Lower Bound

Für die nachfolgenden Experimente ist es wichtig zu wissen, wie groß die Fehlerpunkte im Optimum sind. Wie im folgenden Kapitel 6.1.2 gezeigt wird, ist es nicht möglich, die Problemstellung des Logistikdienstleisters optimal als MINLP-Modell zu lösen. Die Problemstellungen sind zu groß und zu komplex. Daher wird ein Lower Bound – eine untere Schranke für den Fitnesswert – ermittelt, den die Lösungsverfahren auf keinen Fall unterschreiten können.

Die Lösung als MINLP-Modell ist nicht möglich. In Kapitel 4.5.3.1 wurde jedoch ein MIP-Modell vorgestellt, bei dem die Problemstellung vereinfacht wurde. Arbeitsplatzwechsel und erhöhte Fehlerpunkte, wenn ein Arbeitsplatz ohne Bedarf besetzt wird, mussten dort entfallen. Auf Basis dieses MIP-Modells kann ein Lower Bound berechnet werden. Dies wird beispielhaft für das Problem Logistik_Complete beschrieben.

Wird das Problem Logistik_Complete_MIP gelöst, so wird 491 mal einem Dummy-Mitarbeiter ein Arbeitsplatz zugewiesen. Das bedeutet konkret, dass es in 491 Zeitintervallen zu Unterdeckungen kommt. Fehler für Unterdeckungen werden mit einem Punkt je Minute bestraft. Jeder Zeitintervall hat im MIP-Modell die Dauer von 15 Minuten, was insgesamt zu einer Unterdeckung in Höhe von 7.365 Fehlerpunkten führt.

Ebenso treten Überdeckungen auf. So werden Mitarbeiter 2.376 mal auf den Dummy-

Arbeitsplatz geplant, was einer Überdeckung gleich kommt. 2.376 Zeitintervalle mal 15 Minuten ergibt eine Überdeckung von 35.640 Minuten. Problematisch ist hier, dass Überdeckungen beim Logistikproblem ohne Stammarbeitsplatzregelung in zwei Gruppen geteilt werden: Bedarf auf Arbeitsplatz > 0 (ein Fehlerpunkt je Minute) und Bedarf auf Arbeitsplatz $= 0$ (zwei Fehlerpunkte je Minute). Es kann also nicht unterstellt werden, dass es lediglich 35.640 Fehlerpunkte bei der Überdeckung gibt, da in vielen Zeitintervallen Arbeitsplätze einen Bedarf von 0 haben. Interessant ist, dass alle PSO- und ES-Varianten mit Reparatur ebenfalls in Summe eine Überdeckung von exakt 35.640 Minuten ermitteln. Da die Reparatur garantiert eine optimale Aufteilung der Überdeckung in die beiden Gruppen vornimmt, können dort die Optimalwerte abgelesen werden. So ergeben sich 28.395 Fehlerpunkte bei einem Bedarf > 0 und 7.245 Fehlerpunkte bei einem Bedarf $= 0$. Die Reparatur nimmt immer eine optimale Aufteilung vor, da Mitarbeiter, die auf einem Arbeitsplatz mit einem Bedarf $= 0$ arbeiten, sofort umgeplant werden, wenn es in diesem Zeitintervall einen anderen Arbeitsplatz mit einem Bedarf > 0 gibt, für den der Mitarbeiter qualifiziert ist. Das Verfahren ist in Kapitel 4.12.4.1 beschrieben. In den bisher dargestellten Fehlerpunkten fehlen lediglich die minimalen Fehler für Arbeitsplatzwechsel. Diese sind leider nicht berechenbar bzw. herleitbar. Klar ist jedoch: Arbeitsplatzwechsel sind notwendig, um den Personalbedarf bestmöglich zu decken. Daher wird der Lower Bound immer ohne Fehler für Arbeitsplatzwechsel angegeben. In Tabelle 6.1 sind die Lower Bounds der einzelnen Probleme aufgeführt. Die Herleitung der Lower Bounds für die einzelnen Tage erfolgt analog zum Wochenproblem. Die Optimalwerte werden zwangsläufig noch etwas höhere Fehlerpunkte durch Arbeitsplatzwechsel haben.

Tabelle 6.1: Lower Bounds beim Logistikproblem

Problem	Lower Bound	Unterdeckung in Minuten	Überdeckung in Minuten	
			Bedarf >0	Bedarf $=0$
Logistik_Complete	50.250	7.365	28.395	7.245
Logistik_Mo	7.500	900	4.680	960
Logistik_Di	5.700	1.320	2.940	720
Logistik_Mi	7.935	1.965	4.560	705
Logistik_Do	8.040	1.185	4.335	1.260
Logistik_Fr	5.310	750	3.540	510
Logistik_Sa	8.655	675	4.290	1.845
Logistik_So	7.110	570	4.050	1.245

Das Wochenproblem kann problemlos in einzelne Tage zerlegt werden, da sich die Tage nicht gegenseitig beeinflussen. Addiert man die Minimalwerte der sieben Tage, die in dieser Arbeit von Lösungsverfahren gefunden werden, so ergeben sich für die komplette

Woche in der Variante ohne Stammarbeitsplatzregelung 51.419 Fehlerpunkte. Dieses ist bereits sehr nahe am Lower Bound. Der Wert beinhaltet exakt die in Tabelle 6.1 angegebenen Fehlerpunkte für Besetzungsabweichungen und zusätzlich nur noch 1.169 Fehlerpunkte für Arbeitsplatzwechsel. Die 51.419 Fehlerpunkte werden von keinem Verfahren beim Wochenproblem gefunden. Es zeigt sich also, dass eine Zerlegung der Problemstellung zur Komplexitätsreduzierung durchaus Sinn macht.

Die Berechnung der Lower Bounds für die Probleme mit Stammarbeitsplatzregelung orientiert sich am zuvor dargestellten Vorgehen. Da keine erhöhten Fehlerpunkte bei einem Bedarf = 0 anfallen, können die Minimalwerte für Über- und Unterdeckungen direkt aus den MIP-Experimenten abgelesen werden. Eine untere Grenze für Stammarbeitsplatzfehler kann nicht ermittelt werden. Ebenso ist es nicht möglich, einen Lower Bound für Arbeitsplatzwechsel herzuleiten. Diese Teilaspekte müssen folglich ausgelassen werden. Nachfolgend sind die Lower Bounds der einzelnen Probleme in Tabelle 6.2 aufgeführt.

Tabelle 6.2: Lower Bounds beim Logistikproblem mit Stammarbeitsplatzregelung

Problem	Lower Bound	Unterdeckung in Minuten	Überdeckung in Minuten
Logistik_S_Complete	43.005	7.365	35.640
Logistik_S_Mo	6.540	900	5.640
Logistik_S_Di	4.980	1.320	3.660
Logistik_S_Mi	7.230	1.965	5.265
Logistik_S_Do	6.780	1.185	5.595
Logistik_S_Fr	4.800	750	4.050
Logistik_S_Sa	6.810	675	6.135
Logistik_S_So	5.865	570	5.295

Auch beim Wochenproblem mit Stammarbeitsplatzregelung beeinflussen sich die einzelnen Tage nicht gegenseitig. Die Woche kann also problemlos zerlegt werden. Addiert man die Minimalwerte der sieben Tage, die in dieser Arbeit von Lösungsverfahren gefunden werden, so ergeben sich 46.594 Fehlerpunkte. Auch hier zeigt sich, dass Probleme zur Verbesserung der Ergebnisqualität möglichst weit zerlegt werden sollten, wenn sich die einzelnen Teilprobleme nicht gegenseitig beeinflussen.

6.1.2 Ganzzahlige Optimierung

Die folgenden Experimente sind hinsichtlich der Anzahl an Iterationen nicht begrenzt und hinsichtlich der Laufzeit auf maximal acht Stunden eingeschränkt. Wie später zu sehen ist, ist die Einschränkung auf acht Stunden bei Logistik_Complete unproblematisch. Denn es wird gezeigt, dass dort eine um ein Vielfaches höhere Rechenzeit notwendig wäre, die keine realistische Anwendung zulässt.

MIP-Solver In Tabelle 6.3 sind die Ergebnisse der MIP-Solver beim MIP-Modell des Wochenproblems ohne Stammarbeitsplatzregelung aufgeführt.

Tabelle 6.3: Ergebnisse der MIP-Solver

Problem	Solver	Ergebnis
Logistik_ Complete_MIP	BDMLP	Optimales Ergebnis nach 1.431 s
	CoinCbc	Optimales Ergebnis nach 34 s
	CoinGlpk	Optimales Ergebnis nach 118 s
	CoinBonmin-BB	Kein optimales Ergebnis nach 8 h
	CoinBonmin-OA	Optimales Ergebnis nach 8.067 s
	CoinBonmin-QG	Optimales Ergebnis nach 8.349 s
	CoinBonmin-Hyb	Optimales Ergebnis nach 8.164 s
	CoinBonmin-Ecp	Optimales Ergebnis nach 8.005 s

Das Problem Logistik_Complete_MIP mit mehr als 1,2 Mio. Matrixelementen $\neq 0$ konnte in wenigen Sekunden gelöst werden. Dabei schnitt CoinCbc mit 34 Sekunden, gefolgt von CoinGlpk mit 118 Sekunden am besten ab. Eine etwas höhere Rechenzeit weist BDMLP auf, da dieser Solver eher für kleine Probleme geeignet ist, die nicht zu stark degeneriert sind [131], [232]. Die fünf CoinBonmin-Solver sind weniger gut geeignet. Dies liegt an ihrer Ausrichtung auf MINLP-Modelle. Zwar finden vier CoinBonmin-Algorithmen eine optimale Lösung. Dies jedoch erst nach einer ungleich größeren Rechenzeit als CoinCbc. CoinBonmin-BB mit B&B löst das Problem auch nach 8 Stunden nicht. B&B scheint aufgrund der vielen Variablen in der Problemstellung weniger gut geeignet zu sein. Dies zeigt auch der Vergleich von CoinCbc (B&C) mit CoinGlpk (B&B) und BDML (B&B), bei dem CoinCbc das Ergebnis deutlich schneller liefert.

Trotz des Erfolges beim Einsatz von CoinCbc muss MIP als ungeeignet für die Problemstellung des Logistikdienstleisters erachtet werden. Denn das Problem ist als MIP-Modell stark vereinfacht. So werden Nebenbedingungen wie die Reduzierung unnötiger Arbeitsplatzwechsel oder erhöhte Fehlerpunkte bei einem Bedarf von null vernachlässigt. Das

Modell entspricht daher nicht dem eigentlichen Problem. Dies gilt für alle Problemvarianten mit und ohne Stammarbeitsplatzregelung. Demnach ist MIP für das vorliegende Problem nicht anwendbar.

Es wurde jedoch gezeigt, dass MIP-Solver für ähnliche Problemstellungen überaus gut geeignet sind, die sich ohne Modellierungseinschränkungen als lineares Modell darstellen lassen. Voraussetzung ist eine recht einfache Problemstellung ohne Nichtlinearitäten und eine großen Anzahl optimaler Lösungen. In den Arbeiten von Thornton [371], Prüm [312], Burke, Li und Qu [60] sowie Bhulai, Koole und Pot [32] kommt man an deutlich komplexeren MIP-Problemen jedoch genau zur gegenteiligen Aussage bzgl. der Anwendbarkeit von MIP-Solvern. Die zuvor gemachte Aussage soll daher nicht generalisiert werden.

Sollte es gelingen, die beiden vernachlässigten Restriktionen des Logistikproblems doch linear abzubilden, wird die Komplexität des Modells erheblich ansteigen. Die Anwendung von MIP-Solvern wird dann wahrscheinlich nicht zielführend sein. Durch die Vernachlässigung zweier Nebenbedingungen ist bei Logistik_Complete_MIP die Anzahl optimaler Lösungen erheblich erhöht worden. Alle Zeitintervalle a 15 Minuten lassen sich unabhängig voneinander lösen, da sie sich aufgrund der fehlenden Nebenbedingung für Arbeitsplatzwechsel nicht gegenseitig beeinflussen. Beim MIP-Problem handelt es sich daher lediglich um die Zuweisung von verfügbaren Mitarbeitern auf Arbeitsplätze unter Berücksichtigung von Qualifikationen. Da die Mitarbeiter recht gut qualifiziert sind und sich die Zeitintervalle nicht beeinflussen, gibt es sehr viele optimale Zuweisungen. Prinzipiell könnte man Logistik_Complete_MIP daher alternativ mit einer sehr einfachen Heuristik innerhalb einer Sekunde optimal konstruieren lassen.

Die gemachten Aussagen sind analog auf die MIP-Modelle der Tagesprobleme und die Varianten mit Stammarbeitsplatzregelung übertragbar. Bei den Tagesproblemen ist die CPU-Zeit entsprechend geringer als bei der kompletten Woche.

MINLP-Solver In Tabelle 6.4 kann man deutlich erkennen, dass bei den drei MINLP-Modellen nur für Logistik_Complete_4 ein optimales Ergebnis gefunden wird. Und dies in recht kurzer Zeit, wobei CoinBonmin-BB ein wenig länger zur Lösung benötigt. Dies liegt daran, dass CoinBonmin-BB aufgrund der großen Anzahl an Variablen viele NLP-Teilprobleme lösen muss, was zur erhöhten Rechenzeit führt. Die anderen vier CoinBonmin-Algorithmen nutzen B&C-Techniken, die den Lösungsraum beschneiden und so die Rechenzeit verkürzen. Logistik_Complete_4 ist im Vergleich zu den anderen MINLP-Modellen sehr klein. Es gibt nur 4 Zeitintervalle a 15 Minuten, 2 Arbeitsplätze und es sind lediglich 2 der 65 Mitarbeiter anwesend und somit verplanbar.

Bei den beiden anderen MINLP-Problemen Logistik_Complete_20 und Logistik_Complete mit 23.401 bzw. 786.241 Matrixelementen $\neq 0$ wird auch nach einer Rechenzeit von acht Stunden keine optimale Lösung gefunden. Das originale Problem des Logistikdienstlei-

Tabelle 6.4: Ergebnisse der MINLP-Solver

Problem	Solver	Ergebnis
Logistik_ Complete	CoinBonmin-BB	Kein optimales Ergebnis nach 8 h
	CoinBonmin-OA	Kein optimales Ergebnis nach 8 h
	CoinBonmin-QG	Kein optimales Ergebnis nach 8 h
	CoinBonmin-Hyb	Kein optimales Ergebnis nach 8 h
	CoinBonmin-Ecp	Kein optimales Ergebnis nach 8 h
Logistik_ Complete_20	CoinBonmin-BB	Kein optimales Ergebnis nach 8 h
	CoinBonmin-OA	Kein optimales Ergebnis nach 8 h
	CoinBonmin-QG	Kein optimales Ergebnis nach 8 h
	CoinBonmin-Hyb	Kein optimales Ergebnis nach 8 h
	CoinBonmin-Ecp	Kein optimales Ergebnis nach 8 h
Logistik_ Complete_4	CoinBonmin-BB	Optimales Ergebnis nach 190 s
	CoinBonmin-OA	Optimales Ergebnis nach < 1 s
	CoinBonmin-QG	Optimales Ergebnis nach < 1 s
	CoinBonmin-Hyb	Optimales Ergebnis nach < 1 s
	CoinBonmin-Ecp	Optimales Ergebnis nach < 1 s

sters Logistik_Complete kann mit Hilfe von MINLP-Solvern nicht in akzeptabler Zeit gelöst werden. Selbst das immerhin deutlich reduzierte Problem MINLP_20 ist innerhalb von acht Stunden nicht lösbar. MINLP ist demnach nicht geeignet, die vorliegende Problemstellung des Logistikdienstleisters zu lösen. Nicht verwunderlich, wenn man berücksichtigt, dass das Problem NP-schwer ist. Die Größe und die Komplexität der Problemstellung führen zu unakzeptablen Rechenzeiten. Lediglich äußerst kleine Probleme ähnlich Logistik_Complete_4 lassen sich lösen. Sie sind jedoch derart klein, dass sie keinerlei praktische Relevanz haben. Ein Planer könnte Logistik_Complete_4 auf einen Blick optimal lösen. Diese Erkenntnisse decken sich mit den in Kapitel 4.5.2 aufgeführten Ergebnissen. So haben Bonamia u.a. [40] CoinBonmin-Solver an sehr kleinen MINLP-Problemen erprobt und Ergebnisse erst nach drei Stunden erhalten. Eine um ein Vielfaches größere Problemstellung wie Logistik_Complete benötigt zwangsläufig eine erheblich höhere Rechenzeit.

Da die Problemstellungen mit Stammarbeitsplatzregelung ebenfalls nichtlineare Zielfunktionen aufweisen und sich nur marginal von der ursprünglichen Problemstellung unterscheiden, sind bei ihren MINLP-Modellen ähnliche Ergebnisse zu erwarten. Die hier gemachten Erkenntnisse lassen sich ebenfalls auf die MINLP-Modelle aller Tagesprobleme übertragen. Auch sie sind deutlich größer als Logistik_Complete_20 und daher

nicht in akzeptabler Zeit lösbar.

Allgemein ist zu beachten, dass die Modellierung von Problemstellungen in GAMS sehr aufwändig ist. Für jeden einzelnen Zeitintervall muss die richtige Anzahl an Dummy-Mitarbeitern und der korrekte Bedarf des Dummy-Arbeitsplatzes – auch in Abhängigkeit von Qualifikationen – bestimmt werden. Eine manuelle Transformation des Problems ist in der hier vorliegenden Größenordnung nicht zweckmäßig und übersteigt den Aufwand einer manuellen Planerstellung bei Weitem. Für den effizienten Einsatz wäre eine automatische Transformation dringend angeraten.

Bzgl. der Bewertung des Aufwandes für die Implementierung und die Anpassung der MINLP-Solver hat der Modellierungsaufwand jedoch keine Relevanz, da davon ausgegangen wird, dass die Problemstellung und die Fitnessfunktion bereits vorliegen. Es wurden lediglich Parametereinstellungen an den Solvern vorgenommen, die nicht zielführend waren, da für Logistik_Complete kein Ergebnis geliefert wurde. Da die MIP- und MINLP-Solver nicht selbst erstellt wurden und die Entwickler keine Aussagen zum Aufwand gemacht haben, kann zu diesem Punkt keine Bewertung erfolgen.

6.1.3 Hill Climbing

Die Ergebnisse für HC sind im Anhang in den Tabellen 7.1 bis 7.15 dargestellt. I1 und I2 bezeichnen dort die Initialisierungsvariante eins und zwei. Für die beiden Problemvarianten der kompletten Woche benötigt HC etwa 15 Minuten. Bei den einzelnen Tagen jedoch nur ein Siebtel dieser Zeit.

Gewöhnlich leidet HC darunter, nicht aus einem lokalen Optimum entkommen zu können. Besonders bei praktischen Anwendungen, wo die Fitnesslandschaft meist sehr flach ist, stellt dies ein großes Problem dar [96, S. 125]. Auch bei den Varianten des Logistikproblems gestaltet sich die Fitnesslandschaft sehr flach. Ein Beispiel: Es ist bei Logistik_Complete völlig unerheblich, ob Mitarbeiter eins auf Arbeitsplatz eins und Mitarbeiter zwei auf Arbeitsplatz zwei arbeitet oder ob sie ihre Arbeitsplätze vertauschen. Beide sind qualifiziert und tragen zur Bedarfsdeckung bei. Derartige Situationen treten häufig in den Logistikproblemen auf, sodass es weite Plateaus im Lösungsraum gibt. Ein wenig zerklüfteter gestalten sich die Problemvarianten mit Stammarbeitsplatzregelung, da sofort Fehlerpunkte entstehen, wenn ein Mitarbeiter nicht auf seinem Stammarbeitsplatz arbeitet.

Trotz dieses Umstands gelingt es HC in beiden Initialisierungsvarianten bei allen Problemstellungen des Logistikdienstleisters ohne Stammarbeitsplatzregelung die Überdeckung im Personaleinsatz bei einem Bedarf von null auf das Optimum zu reduzieren. Dies liegt am systematischen Vorgehen bei der Suche. Es werden daher immer alle Verstöße dieser Art gefunden und behoben. Probleme gibt es bei den anderen Nebenbedingungen

hinsichtlich des passgenauen Personaleinsatzes und der Arbeitsplatzwechsel. Hier wären für eine weitere Verbesserung erheblich umfangreichere Umplanungen erforderlich. Mit der Änderung eines einzelnen Matrixelementes in jeder Iteration ist ein Entkommen aus einem lokalen Optimum nicht möglich.

Im Gegensatz dazu gelingt es HC an den Problemvarianten mit Stammarbeitsplatzregelung sogar, das Minimum der Über- und Unterdeckung am Freitag, Samstag und Sonntag zu erreichen. Bei den übrigen Tagen und dem Wochenproblem sind die Ergebnisse recht nahe an den Optimalwerten. Die anderen Nebenbedingungen werden ein wenig schlechter eingehalten, wobei die Anzahl der Arbeitsplatzwechsel durchweg sehr gering ist. Der Verzicht auf zusätzliche Fehlerpunkte bei einem Bedarf von null scheint das Problem für HC bzgl. des passgenauen Personaleinsatzes zu vereinfachen. Auffällig ist, dass HC bei den Problemvarianten mit Stammarbeitsplatzregelung fast immer die gleichen Fehlerpunkte je Problemstellung (aber verschiedene Einsatzpläne) liefert. Dies ist eine Auswirkung des doch recht systematischen Vorgehens, was besonders bei den kleineren Tagesproblemen hervortritt.

Nicht nur die Art der Berechnung des nächsten Schrittes, auch die Art der Initialisierung hat einen Einfluss auf die Reduktion der Überdeckung und Unterdeckung im Personaleinsatz. Besonders auffällig ist dies bei der Problemvariante ohne Stammarbeitsplatzregelung. Dort gelingt es der Initialisierungsvariante 1 (I1) deutlich besser als der Variante 2 (I2) die Abweichungen des Personaleinsatzes vom Personalbedarf zu reduzieren. Dafür hat HC mit I1 jedoch mehr unnötige Arbeitsplatzwechsel. Bei HC mit I1 wird jedem Mitarbeiter bei der Initialisierung ein zufälliger Arbeitsplatz für jeden Zeitintervall zugewiesen, in dem er anwesend ist. Die Startlösung hat folglich extrem viele Arbeitsplatzwechsel, die nicht im nötigen Umfang reduziert werden können. Im Gegensatz dazu erhält ein Mitarbeiter bei HC mit I2 einen zufälligen Arbeitsplatz für den kompletten Tag in der Startlösung, sodass dort keine Arbeitsplatzwechsel auftreten. Dafür fällt es HC mit I2 jedoch schwerer, ausgehend von einer sehr geordneten Lösung, Über- und Unterdeckungen zu reduzieren. HC mit I1 führt im Gegensatz zu HC mit I2 gleich zu Beginn deutlich mehr Änderungen im Plan durch, da sich diese aufgrund der vielen Arbeitsplatzwechsel häufiger lohnen bzw. zu keinen Verschlechterungen führen. Die Suche ist daher nicht so eingeschränkt und kann in vielversprechende Regionen vordringen.

Mit Hilfe von t-Tests und jeweils 30 unabhängigen Replikationen von HC I1 und HC I2 kann geprüft werden, ob HC I1 tatsächlich signifikant besser als HC I2 ist. Für das Wochenproblem ohne Stammarbeitsplatzregelung zeigt der Levene-Test bei einem Signifikanzniveau von 0,05, dass die Standardabweichungen homogen sind ($L = 3,808$; $p = 0,056$). Für den t-Test ergibt sich ein signifikanter Mittelwertunterschied bei einem Signifikanzniveau von 95% ($T = -4,203$; $p < 0,001$). Analog hierzu wurden t-Tests für die Tagesprobleme durchgeführt. HC I1 ist an fünf Tagen signifikant besser als HC I2. Am Dienstag gibt es keinen signifikanten Unterschied der Mittelwerte und am Samstag ist HC I2 signifikant besser.

Bei den Problemvarianten mit Stammarbeitsplatzregelung ist der zuvor dargestellte Effekt der Initialisierungsvarianten nur am relativ groß dimensionierten Wochenproblem zu beobachten. Dort hat HC mit I1 mehr Arbeitsplatzwechsel als HC mit I2 – dafür aber weniger Fehler für Besetzungsabweichungen. Dennoch schneidet HC mit I2 signifikant besser ab und liefert sogar den Bestwert von allen getesteten Lösungsverfahren. Dies wird mit einem t-Test ($\alpha = 0,05$; $T = 8,967$; $p < 0,001$) belegt – Varianzhomogenität ist dort bei einem Signifikanzniveau von 0,05 nicht gegeben. An den einzelnen Tagen sind die Besetzungsabweichungen bereits immer auf das Optimum reduziert. Unterschiede in der Anzahl an Arbeitsplatzwechseln und in Stammarbeitsplatzfehlern sind bei beiden Initialisierungsvarianten kaum feststellbar. Die Art der Initialisierung ist bei den kleineren Problemen mit Stammarbeitsplatzregelung demnach nicht relevant. So zeigen t-Tests, dass HC I1 und HC I2 an 4 Wochentagen keine signifikanten Mittelwertunterschiede aufweisen ($\alpha = 0,05$). Am Dienstag ist HC I2 signifikant besser und für den Mittwoch und Sonntag kann kein t-Test durchgeführt werden, da die Standardabweichungen der Stichproben null sind. Die Mittelwerte von HC I1 und HC I2 sind in diesen beiden Fällen identisch.

Zusammenfassend kann festgestellt werden, dass die Ergebnisse von HC in beiden Problemvarianten – sei es am Wochenproblem oder an den einzelnen Tagen – recht gut sind. Zumal HC an sich äußerst simpel gestaltet ist.

Der Aufwand für die Implementierung ist aufgrund des einfachen Ablaufs trivial und in wenigen Mannstunden vollzogen. Parameter sind nicht vorhanden und müssen somit nicht eingestellt werden.

6.1.4 Guided Local Search

In Tests wurde die Anwendbarkeit von GLS erprobt. Es zeigt sich, dass GLS bei beiden Varianten des Logistikproblems keine Vorteile gegenüber der einzelnen Anwendung des zu Grunde liegenden Lokalen Suchverfahrens hat. Auf umfassende Experimente wurde daher verzichtet.

Die Ursache, warum GLS zu keinem besseren Ergebnis führt, ist recht einfach und in der Größe der Problemstellung begründet. GLS benötigt Features f , die verändert werden. Beim Logistikproblem sind dies die einzelnen Matrixelemente der Lösung. Wird gegen eine Nebenbedingung verstoßen, so wird dort u.U. (entsprechend der Bewertungsformel) eine Anpassung vorgenommen. Da die Problemstellung viele tausend Matrixelemente hat, die als Feature in Frage kommen, sind extrem viele Iterationen nötig, um positive Effekte durch GLS zu erzielen und aus einem lokalen Optimum zu entkommen. Ein Abbruchkriterium von 400.000 Berechnungen der Fitnessfunktion ist hier derart klein bemessen, dass beim Logistikproblem kein Unterschied zu den Ergebnissen des HC festgestellt werden kann. GLS kann seine Wirkung aufgrund der Vielzahl an Features

einfach nicht entfalten – selbst bei einem deutlich größeren Abbruchkriterium von mehr als 400.000 Berechnungen der Fitnessfunktion.

Im Vergleich hierzu können Tsang und Voudouris [383] GLS am Problem der British Telecom mit sehr guten Ergebnissen anwenden (siehe Kapitel 6.3.1). Dort gibt es auch nur 250 Features (Anzahl Aufträge) und nicht mehrere tausend wie beim Logistikproblem.

Der Aufwand für Implementierung und Parametrierung hält sich mit maximal einem Manntag in Grenzen. Es muss lediglich die Fitnessfunktion angepasst werden und auch der Parameter λ von GLS ist relativ unempfindlich gegenüber Änderungen, sodass er nicht aufwändig eingestellt werden braucht.

6.1.5 Tabu Search

Die folgenden Ausführungen basieren auf der Diplomarbeit von Helbig [178] und den dort durchgeführten Experimenten. Diese beziehen sich ausschließlich auf die Problemvarianten ohne Stamarbeitsplatzregelung. In den Ergebnistabellen 7.1 bis 7.15 im Anhang sind die Ergebnisse von TS dargestellt. TS I2 bzw. I3 steht dort für die Initialisierung mit der Variante 2 bzw. 3, VNS steht für Variable Neighbourhood Search und T* für die Ausweitung des Abbruchkriteriums auf 2,8 Mio. statt 400.000 Berechnungen der Fitnessfunktion (nur beim Wochenproblem). Alle anderen TS-Varianten sind hybride Formen (siehe Kapitel 6.1.9).

In Tests hat sich herausgestellt, dass die TS-Ergebnisse mit der Initialisierungsvariante I1 zu sehr schlechten Ergebnissen führen, da TS mit einer sehr ungeordneten Startlösung, die sehr viele Arbeitsplatzwechsel enthält, nicht zurechtkommt. Variante I1 wird daher nicht für TS verwendet. Im Vergleich der Initialisierungsvarianten I2 und I3 schneidet I3 bei den Wochentagen immer signifikant besser ab. In t-Tests mit einem Signifikanzniveau von 95% wurde die H_0 -Hypothese mit p -Werten kleiner als 0,001 bei allen Tagen immer abgelehnt. Es gibt also einen Zusammenhang zwischen der Startlösung und dem Ergebnis von TS. Die Startlösung sollte für TS beim Logistikproblem möglichst wenig Arbeitsplatzwechsel enthalten, da deren Minimierung in Kombination mit den weiteren Nebenbedingungen sehr viel Rechenzeit benötigt und zudem nicht trivial ist.

TS liefert sehr gute Ergebnisse bei den einzelnen Tagen und liegt immer sehr nahe an den Bestwerten von PSO sowie ES. Z.T. schlägt TS auch die besten Minimal- und/oder Durchschnittswerte anderer Verfahren. Es findet aber nicht bei allen Tagen die Minimalwerte bei der Über- und Unterdeckung im Personaleinsatz. Diese zusätzlichen Fehler sind zwar sehr klein, machen aber deutlich, dass TS auch bei einer fortgeschrittenen Anzahl an Iterationen nicht immer in alle Regionen des Lösungsraums vordringt. Denn das Verfahren ist so konstruiert, dass es derartige Fehler beheben würde, wenn es auf sie trifft. Eine Reparatur, wie sie bei PSO und ES verwendet wird, könnte diese Fehler

ganz beseitigen und wäre eine sinnvolle Ergänzung. Lediglich die CPU-Zeit steigt dann erheblich an.

Für eine weitere Verbesserung von TS am Wochenproblem wird VNS eingeführt. Das Problem wird virtuell in Tage zerlegt, zumal sich die einzelnen Tage im Wochenproblem nicht gegenseitig beeinflussen. Somit wird die Nachbarschaft um ein Siebtel reduziert. TS beginnt am ersten Tag der Woche und springt zum nächsten Tag, wenn in der letzten Iteration keine Verbesserung erzielt wurde. Beim Sonntag angekommen, wird wieder zum Montag gesprungen. TS verharrt nun nicht mehr in einer Nachbarschaft, die kaum Verbesserungen verspricht, sondern springt sofort in eine andere Nachbarschaft, die womöglich besser ist. Diese Variante verbessert TS erheblich. TS VNS I3 hat nur noch 56.489,5 Fehlerpunkte im Durchschnitt, womit es gegenüber TS I3 mit durchschnittlich 116.797,9 Fehlerpunkten signifikant besser ist ($\alpha = 0,05$; $T = -16,917$; $p < 0,001$). Dieser Wert kommt jedoch nicht an die Bestwerte von PSO und ES heran.

Im Gegensatz zu den Ergebnissen an den einzelnen Tagen ist TS am deutlich größeren Wochenproblem sehr viel schlechter. TS hat bei allen Fehlerpunktarten recht große Probleme. Selbst Qualifikationsfehler werden nicht völlig beseitigt. Während das Abbruchkriterium von 400.000 Bewertungen der Fitnessfunktion bei PSO, ES und HC mehr als ausreichend ist, hat Helbig festgestellt, dass TS beim Wochenproblem noch nicht konvergiert ist [178]. Aufgrund der guten Erfahrungen bei den einzelnen Tagen wählt er ein Abbruchkriterium, welches das Siebenfache der (sieben Tage der Woche) 400.000 Berechnungen beträgt. Die jeweilige TS-Konfiguration mit 2,8 Mio. statt 400.000 Berechnungen wird in Tabelle 7.1 mit TS* kenntlich gemacht. Der Mittelwert ist bei TS* VNS I3 mit einem erweiterten Abbruchkriterium signifikant besser (51.767,2) als beim Pendant TS VNS I3 ($\alpha = 0,05$; $T = -5,371$; $p < 0,001$). Es wird aber keine absoluter Bestwert im Vergleich zu anderen Algorithmen erzeugt.

Zusammenfassend wird klar, dass TS recht viele Iterationen und demzufolge viel Rechenzeit für gute Ergebnisse benötigt. Die Anzahl der erforderlichen Iterationen hängt u.a. stark von der Größe des Lösungsraums ab. Z.B. beim Wochenproblem im Vergleich zu den einzelnen Tagen. In Experimenten zeigt sich, dass TS beim Wochenproblem selbst nach 2,8 Mio. Berechnungen der Fitnessfunktion nicht vollends konvergiert ist. Eine geringe Wahrscheinlichkeit auf weitere Verbesserungen ist immer noch vorhanden. Auf noch umfangreichere Berechnungen hat Helbig jedoch verzichtet, da er mit 2,8 Mio. Berechnungen und jeweils 30 Replikationen bereits an praktikable Zeitgrenzen mit modernen Rechnern gestoßen ist. CPU-Zeit ist an dieser Stelle zwar nicht der begrenzende Faktor. Mehrtägige Berechnungen einer Replikation sollen jedoch vermieden werden.

Zum Implementierungs- und Parametrierungsaufwand macht Helbig [178] keine Aussagen. Es wird jedoch deutlich, dass sein TS-Ansatz über eine Vielzahl an Parametern verfügt, die einzustellen sind.

6.1.6 Evolutionsstrategien

Alle Ergebnisse sind im Anhang in den Tabellen 7.1 bis 7.32 dargestellt. In Klammern ist die Anzahl der Eltern, der Selektionstyp (Plus- oder Kommaselektion) sowie die Anzahl der Nachkommen angegeben. TypG und TypE bezeichnen die Mutationsart und R steht für die Reparatur. Die Ergebnisse mit ES sind bei allen Varianten der Problemstellung sehr gut. Es treten keine Qualifikationsfehler auf und auch die Besetzungsabweichungen werden stark reduziert.

Vergleicht man die besten Durchschnittswerte (mit und ohne Reparatur), die mit der angepassten Gauß-Mutation (TypG) und mit der Mutation mit dem Konzept der Maximalen Entropie (TypE) erzielt werden, so schneidet TypG bei den Tagesproblemen ohne Stammarbeitsplatzregelung immer signifikant besser ab als TypE. Am komplexeren Wochenproblem ohne Stammarbeitsplatzregelung ist jedoch TypE signifikant besser. Alle t-Tests wurden mit einem Signifikanzniveau von 95% durchgeführt. Die p -Werte waren immer kleiner als 0,001. Bei den Problemstellungen mit Stammarbeitsplatzregelung ist TypG nur am Freitag, Samstag und Sonntag signifikant besser als TypE ($\alpha = 0,05$ und p jeweils $< 0,001$). Und TypE war nur am Dienstag und Mittwoch signifikant besser. An den anderen Tagen und am Wochenproblem sind die Mittelwerte der jeweils besten Konfigurationen nicht signifikant verschieden. Mit der Mutation vom TypE soll die Wahrscheinlichkeitsverteilung für die Mutationsschrittweite besser an die Charakteristik des Lösungsraums angepasst werden. Dies scheint hier jedoch nicht sonderlich gut zu gelingen.

ES mit TypG zeigt die Tendenz, dass bessere Ergebnisse bei einer kleinen Population zu erwarten sind. Mit Ausnahme der Problemvarianten mit Stammarbeitsplatzregelung inkl. Reparatur, wo keine Tendenz zu erkennen ist, gilt dies für alle Problemvarianten des Logistikproblems. In t-Tests wurde jeweils die kleinste Population mit der größten Population mit und ohne Reparatur für die Plus- und Kommaselektion an allen Problemen verglichen (außer die acht Problemvarianten mit Stammarbeitsplatzregelung und Reparatur). Insgesamt wurden 48 t-Tests mit einem Signifikanzniveau von 95% durchgeführt, wobei die Mittelwerte von ES (1+/,5) TypG (jeweils mit und ohne R) im Vergleich zu den Mittelwerten ES (30+/,200) TypG (jeweils mit und ohne R) in 37 der 48 Tests signifikant besser waren. In fünf Tests gab es keinen signifikanten unterschied und in sechs Tests war der Mittelwert der größeren Population signifikant besser. Der Grund für diesen Effekt liegt im einheitlichen Abbruchkriterium von 400.000 bewerteten Fitnessfunktionen. Eine kleinere Population kann bei diesem Abbruchkriterium mehr Generationen durchlaufen, was entscheidend ist. Denn vom initialen Plan bis zu einem guten Ergebnis müssen z.B. bei den 36.000 Dimensionen des Wochenproblems eine Vielzahl an Änderungen vorgenommen werden. Da jedoch nur wenige zufällige Änderungen pro Generation zu guten Ergebnissen führen, sind entsprechend viele Generationen notwendig. Der Vorteil einer großen Population wiegt bei diesem Abbruchkriterium also nicht so stark. Dies würde sich ändern, wenn die Anzahl an Generationen das alleinige Abbruch-

kriterium wäre. Allgemein gilt, dass es hilfreich ist, nur wenige zufällige Änderungen pro Generation zu machen. Bei zu vielen Änderungen steigt die Wahrscheinlichkeit, mehr Fehler als Verbesserungen herbeizuführen. Eine Ergebnisverbesserung würde dann nicht zustande kommen und die Individuen würden sich ziellos ändern.

Die Aussage bei TypG bzgl. der Vorteilhaftigkeit kleiner Populationen gilt nicht bei TypE. In 64 t-Tests schneiden bei fast allen Problemvarianten (1+/,5)-Konfigurationen im Vergleich zu (30+/,200)-Konfigurationen am signifikant schlechtesten ab (jeweils mit und ohne Reparatur sowie $\alpha = 0,05$). Zwar gilt auch bei TypE, dass kleine Populationen mehr Generationen zulassen, was durchaus vorteilhaft ist. Stärker wiegt dort jedoch der Vorteil einer großen Population, die die vorzeitige Konvergenz in einem lokalen Optimum verhindern hilft. Für TypE scheint eine größere Gefahr zu bestehen, dass die Population in einem lokalen Optimum festsitzt und nicht entkommen kann. Beim Logistikproblem kann dem mit einer größeren Population begegnet werden.

Die Plusselektion hat bei Logistik_Complete einen kleinen Vorteil gegenüber der Kommaselektion. In zwölf t-Test ($\alpha = 0,05$; mit und ohne R, TypG und TypE sowie alle drei Populationsgrößen) waren die Mittelwerte mit der Plusselektion in sechs Fällen signifikant besser als mit der Kommaselektion. Und in den sechs restlichen Fällen gab es keinen signifikanten Unterschied. Es ist also durchaus vorteilhaft, Eltern nicht aus der Selektion auszuschließen. „Gute“ Eltern haben dann eine hohe Wahrscheinlichkeit zu überleben und die Suche zu leiten, da „gute“ Pläne nicht verloren gehen. Bei den einzelnen Tagen ohne Stammarbeitsplatzregelung ist dieser Effekt weniger stark ausgeprägt bzw. nicht vorhanden, sodass die zuvor gemachte Aussage nicht generalisiert werden soll. Bei den entsprechenden t-Tests mit einem Signifikanzniveau von 95% gab es meist keine signifikanten Unterschiede zwischen den Mittelwerten. Und nur gelegentlich war die Plusselektion signifikant besser. Genau gegenteilig verhält es sich bei Logistik_S_Complete. Dort gab es ebenfalls keinen signifikanten Unterschied zwischen den Mittelwerten von sechs Fällen. Bei den anderen sechs Fällen war jedoch die Kommaselektion signifikant besser. Ähnliche Ergebnisse zeigen sich in den t-Tests bei den einzelnen Wochentagen mit Stammarbeitsplatzregelung. Hier ist die Kommaselektion signifikant besser – besonders dort, wo die Reparatur zum Einsatz kommt. Anscheinend hat sie einen sehr großen Einfluss auf die Individuen. Da die Reparatur auch Stammarbeitsplatzfehler korrigiert, beschränkt sie das Verhalten der Individuen recht stark. Eltern, können derart repariert werden, dass ihre Mutation immer wieder rückgängig gemacht wird. Die Kommaselektion ist bei den Problemen mit Stammarbeitsplatzregelung daher vorteilhaft. Auf den Einfluss der Reparatur wird etwas später erneut eingegangen.

Die dargestellten Zusammenhänge für Logistik_Complete sind in Abbildung 6.1 zu erkennen. Die Abbildung zeigt den Verlauf der besten ES-Konfigurationen mit einem Elter und fünf Nachkommen für die Plus- und Kommaselektion bei TypG und TypE mit Reparatur. Die ersten 2.000 berechneten Fitnesswerte sind nicht dargestellt, da die Skalierung dann wenig Details erkennen lässt. Nach der Initialisierung weisen die Verfahren etwa 110.000 Fehlerpunkte auf und reduzieren den Wert innerhalb von 2.000 berechneten Fit-

nessfunktionen sehr stark. Deutlich erkennt man die Überlegenheit von TypG, wo Plus- und Kommaselektion sehr nahe beieinander liegen. Bei TypE hingegen hat die Plusselektion sehr lange recht schlechte Werte im Vergleich zur Kommaselektion. Gute Eltern überleben hier anscheinend sehr lange und beeinflussen die Suche zunächst negativ.

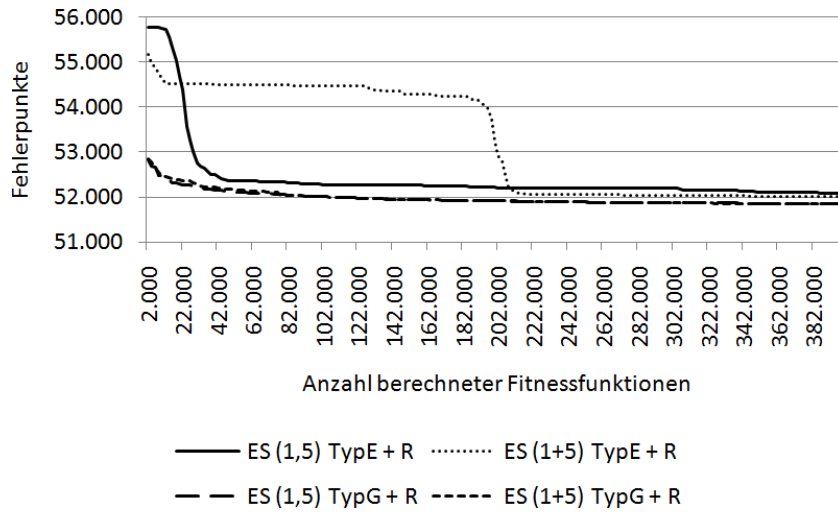


Abbildung 6.1: Verlauf der Fehlerpunkte für ES bei Logistik_Complete

Analog zu Abbildung 6.1 wurde auch für Logistik_S_Complete der Verlauf dargestellt. Bei den Kommaselektionen sind kleinere Ausschläge nach oben zu beobachten. Hier ist die Kommaselektion der Plusselektion überlegen.

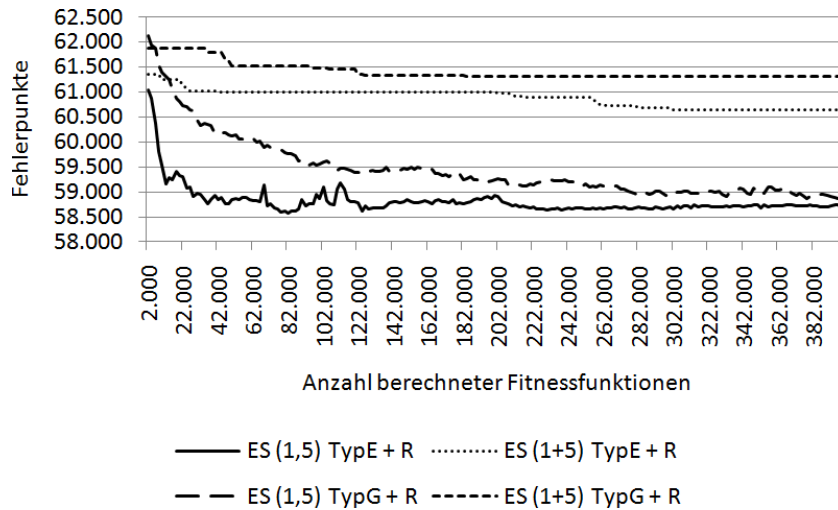


Abbildung 6.2: Verlauf der Fehlerpunkte für ES bei Logistik_S_Complete

Für alle Logistikprobleme wurden Experimente mit und ohne Reparatur bei ES durchgeführt. Die Reparatur sorgt fast immer bei den Problemen ohne Stammarbeitsplatz-

regelung für eine signifikante Verbesserung der Mittelwerte ($\alpha = 0,05$), die auch ohne Reparatur schon recht gut sind. Durch die Reparatur werden alle Fehler für Über- und Unterdeckungen sowie Qualifikationsfehler auf das Optimum reduziert. Somit treten dort nur noch vermeidbare Fehler bei der Anzahl an Arbeitsplatzwechseln auf. Anders verhält es sich bei der Problemvariante mit Stammarbeitsplatzregelung. Dort sind alle Ergebnisse ohne Reparatur besser als mit Reparatur. Der Einsatz der Reparatur ist dort nicht zielführend. Denn der Verzicht auf die Reparatur führt zu weniger Stammarbeitsplatzfehlern, obwohl es in der Reparatur Mechanismen zur Korrektur dieser Fehler gibt, die intensiv auf ihre korrekte Funktionsweise hin überprüft wurden. Die Ursache für diesen Effekt scheint darin begründet zu sein, dass die Reparatur die Bewegung der Individuen einschränkt. Zwar werden einige Stammarbeitsplatzfehler beseitigt. Bestimmte Regionen im Lösungsraum sind aber nicht mehr erreichbar. Bei den Problemen ohne Stammarbeitsplatzregelung war diese Einschränkung der Individuen völlig unkritisch. Mitarbeiter werden dort nicht immer auf ihren Stammarbeitsplatz zurückgesetzt, wenn dies möglich ist. Mit Stammarbeitsplatzregelung ist die Reparatur jedoch völlig unbrauchbar, da sie die Bewegung der Individuen im nun stärker zerklüfteten Lösungsraum zusätzlich einschränkt – jeder Mitarbeiter hat eben nur einen Stammarbeitsplatz. Dennoch ist eins zu erkennen: Auch wenn die Reparatur höhere Gesamtfehlerpunkte durch mehr Stammarbeitsplatzfehler aufweist, so sind doch die Besetzungsabweichungen geringer als beim Verzicht auf die Reparatur. Insgesamt ist dies ein sehr gutes Beispiel für die Auswirkungen eines zu starken Eingriffs in die Suche.

Die Reparatur erfordert zusätzliche Berechnungen und sorgt für eine Verdoppelung der CPU-Zeit. ES benötigt ohne Reparatur bei beiden Wochenproblemen Logistik_Complete sowie Logistik_S_Complete 25 Minuten und mit Reparatur etwa 50 Minuten. Bei den einzelnen Tagen entspricht die CPU-Zeit ein Siebtel der genannten Werte.

An dieser Stelle sei angemerkt, dass im Rahmen dieser Arbeit Tests in Zusammenhang mit veränderlichen Populationsgrößen und veränderlichem Selektionsdruck durchgeführt wurden. Dies geschah vor dem Hintergrund, dass die Lösungsevaluierungen so verteilt sein sollten, dass eine möglichst hohe Chance auf Verbesserungen besteht. In Phasen, in denen kleine Populationen für Verbesserungen genügen, sollten auch nur kleine Populationen verwendet werden. In Phasen mit einer langsamen Fortschrittsgeschwindigkeit sollte die Population vergrößert werden. Ebenso wurden Tests mit dem Selektionsdruck durchgeführt. Kommt es nach einer bestimmten Anzahl an Generationen zu keine Verbesserung, wird der Selektionsdruck herabgesetzt. Bei vielen Verbesserungen – also einer raschen Fortschrittsgeschwindigkeit – wird er heraufgesetzt. Bei den Experimenten wurden auch verschiedene Schrittweiten für Anpassungen erprobt. Es zeigt sich jedoch, dass trotz dieser Anpassungen Verbesserungen bereits nach wenigen Generationen immer seltener sind. Die Populationsgröße und der Selektionsdruck laufen daher recht schnell gegen vorgegebene Grenzen, von denen sie sich nur äußerst selten wieder kurzzeitig entfernen. Die Tests, bei denen Mechanismen zur Änderung der Populationsgröße und des Selektionsdrucks implementiert waren, liefern daher sehr schlechte Ergebnisse nahe der Startlösung, weshalb hierzu keine weiterführenden Experimente angestellt wurden.

Der Implementierungs- und Anpassungsaufwand für ES ist nicht ganz unerheblich, da die verschiedenen Suchoperatoren Rekombination, Mutation und Selektion implementiert werden müssen. Die Parametereinstellung hält sich bei ES TypG in Grenzen. Bei TypE ist sie ein wenig aufwändiger. Durch die zusätzliche Einführung der Reparatur wird der Implementierungsaufwand wesentlich erhöht, da die Reparatur sehr viel problemspezifisches Wissen beinhaltet, welches durch die Analyse von erzeugten Einsatzplänen gewonnen werden muss. Daher sind im Maximum bis zu fünf Manntage je Mutationstyp als Implementierungs- und Anpassungsaufwand für ES zu veranschlagen.

6.1.7 Particle Swarm Optimization

Die Ergebnisse zu PSO sind in den Tabellen 7.1 bis 7.31 im Anhang zu finden. In Klammern ist jeweils die Schwarmgröße angegeben. R steht für die Reparatur und gBest, lBest, Circle sowie Wheel bezeichnen die Nachbarschaftstopologie des Schwarms.

Generell liefert PSO sehr gute Ergebnisse. Dennoch gibt es Konfigurationen, in denen Qualifikationsfehler und sehr starke Besetzungsabweichungen auftreten. An den Problemen ohne Stammarbeitsplatzregelung verbessert die Reparatur bei PSO alle Ergebnisse. Für Probleme mit Stammarbeitsplatzregelung ist die Reparatur nur bei der kompletten Woche vorteilhaft. Bei den Tagen sind die Ergebnisse schlechter als ohne Reparatur. Dies ist nicht verwunderlich, da bei PSO die gleiche Reparatur zum Einsatz kommt wie bei ES. Die CPU-Zeit wird auch bei PSO durch die Reparatur verdoppelt, sodass sie dann etwa 50 Minuten statt 25 Minuten ohne Reparatur entspricht. Bei den einzelnen Tagen reduzieren sich die Zeiten auf ein Siebtel.

Wie bei ES mit der Mutation vom TypG haben auch bei PSO kleine Schwarmgrößen häufig signifikant besserer Mittelwerte als große Schwärme. Die Ursache ist das Abbruchkriterium, welches auf der Anzahl an berechneten Fitnessfunktionen beruht und bei kleinen Schwarmgrößen mehr Iterationen zulässt. Da viele kleine Änderungen erforderlich sind, ist dies vorteilhaft. Für eine tiefer gehende Begründung sei auf die zuvor gemachten Aussagen in Kapitel 6.1.6 verwiesen. Der Effekt hinsichtlich der Populationsgröße gilt für fast alle Probleme mit und ohne Stammarbeitsplatzregelung. Bei den Tagesproblemen mit Stammarbeitsplatzregelung und ohne Reparatur sind jedoch z.T. Tendenzen in Richtung größerer Schwärme zu erkennen. Der Vorteil eines großen Schwarms, der sich weit im Lösungsraum verteilen kann, wiegt in diesen Fällen stärker als der Vorteil, aufgrund vieler Iterationen deutlich mehr Änderungen im Plan vollziehen zu können. In 128 t-Tests mit einem Signifikanzniveau von 95%, in denen an allen Problemvarianten und für alle vier Nachbarschaftstopologien PSO (20) mit PSO (200) verglichen wurde (mit und ohne Reparatur), schnitten die Konfigurationen mit 20 Partikeln 95 mal signifikant besser ab als PSO (200). Und dies vor allem bei den Varianten ohne Stammarbeitsplatzregelung. Dort waren die Mittelwerte mit 200 Partikeln nie signifikant besser als die mit 20 Partikeln. Lediglich in zwei Fällen gab es keinen signifikanten Unterschied. Wie bereits

erläutert, ist der kleinere Schwarm bei den Problemvarianten mit Stamarbeitsplatzregelung nicht mehr derart dominierend. Dort treten in Summe 17 Fälle auf, in denen kein signifikanter Unterschied zwischen den Mittelwerten besteht. Eine Ausnahme ist dort die Wheel-Topologie, wo ein kleiner Schwarm immer signifikant bessere Mittelwerte liefert. In dieser Topologie werden Informationen nur äußerst langsam im Schwarm verteilt. Er benötigt also eine Vielzahl an Iterationen, was kleine Schwarmgrößen bevorteilt.

Nachfolgend wird auf den Vergleich verschiedener Nachbarschaftstopologien eingegangen. Hierzu werden die Ergebnisse von gBest, Wheel, Circle ($k=2$) und lBest ($k=4$) beschrieben. Vergleicht man die jeweils besten Mittelwerte der Nachbarschaftstopologien am Problem Logistik_Complete (mit und ohne Reparatur), zeigt sich, dass gBest immer signifikant besser abschneidet als die anderen drei Topologien ($\alpha = 0,05$ und p -Wert jeweils $< 0,001$). Dies steht im Gegensatz zu den Arbeiten von Kennedy und Mendes [207] sowie von Czogalla und Fink [89], die nahe legen, dass die gBest-Topologie zu einer vorzeitigen Konvergenz in einem lokalen Optimum führt. Die Topologien lBest und Circle schneiden ein wenig schlechter als gBest ab und Wheel ist am schlechtesten. Dort treten in einigen Konfigurationen sogar Qualifikationsverletzungen auf. Bei Logistik_Complete werden die Ergebnisse umso besser, je größer die Nachbarschaft ist. Der Vorteil kleinerer Nachbarschaften, die Konvergenz in einem lokalen Optimum eher zu vermeiden, fällt nicht so stark ins Gewicht. Aufgrund der Komplexität der Problemstellung werden „gute“ Besetzungen einzelner Mitarbeiter nur schwer gefunden. Steht die Information einer derartigen Besetzung nicht sofort allen Partikeln zur Verfügung, besteht die Gefahr, dass sie verloren geht. Ein weiterer Grund, warum gBest bei Logistik_Complete so erfolgreich ist, liegt im Ablauf des für diese Problemstellung modifizierten PSO-Algorithmus. In der zweiten Aktion für die Positionsneuberechnung wird eine Arbeitsplatzzuweisung zufällig geändert. Daher bleiben immer alle Partikel in Bewegung und haben die Chance, aus einem lokalen Optimum zu entkommen.

Für PSO (20) + R ist der Verlauf der vier Nachbarschaftstopologien in Abbildung 6.3 dargestellt. Hierbei handelt es sich um die besten Durchläufe. Da die Fehlerpunkte sehr rasch reduziert werden (Fehlerpunkte bei Initialisierung = 411.330), mussten die ersten 2.000 Fitnesswerte weggelassen werden. In der Abbildung wären sonst lediglich rechte Winkel zu erkennen. Dies verdeutlicht, dass sich PSO sehr schnell an gute Lösungen annähert. Wie zuvor beschrieben, schneidet gBest am besten ab, dicht gefolgt von lBest und Circle. Wheel hat immer die schlechtesten Fehlerpunkte.

Bei Logistik_S_Complete zeigt sich ein anderes Bild, welches den Erkenntnissen von Kennedy und Mendes [207] sowie von Czogalla und Fink [89] entspricht. Auch hier wurden t-Tests mit einem Signifikanzniveau von 95% durchgeführt. Nun schneidet die lBest-Topologie, die eine vorzeitige Konvergenz in einem lokalen Optimum verhindert, mit und ohne Reparatur immer signifikant besser ab als die gBest-Topologie. Mit Stamarbeitsplatzregelung ist der Lösungsraum zerklüfteter als ohne. Der Schwarm kann schneller in einem lokalen Optimum konvergieren, was durch die lBest-Topologie eher verhindert wird. Dort werden Informationen nicht sofort allen Partikeln zugänglich gemacht. Zu

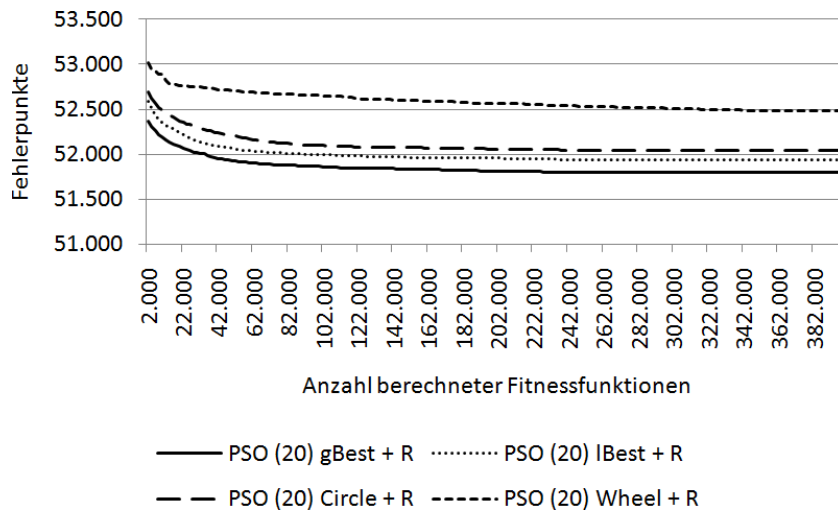


Abbildung 6.3: Verlauf der Fehlerpunkte verschiedener Nachbarschaftstopologien für PSO bei Logistik_Complete

stark darf die Informationsweitergabe jedoch nicht eingeschränkt werden, da die Wheel-Topologie wiederum am signifikant schlechtesten abschneidet.

Dieser Zusammenhang wird in Abbildung 6.4 verdeutlicht. Während sich gBest und Wheel nach 2.000 berechneten Fitnessfunktionen nicht merklich verbessern, dominiert vor allem lBest.

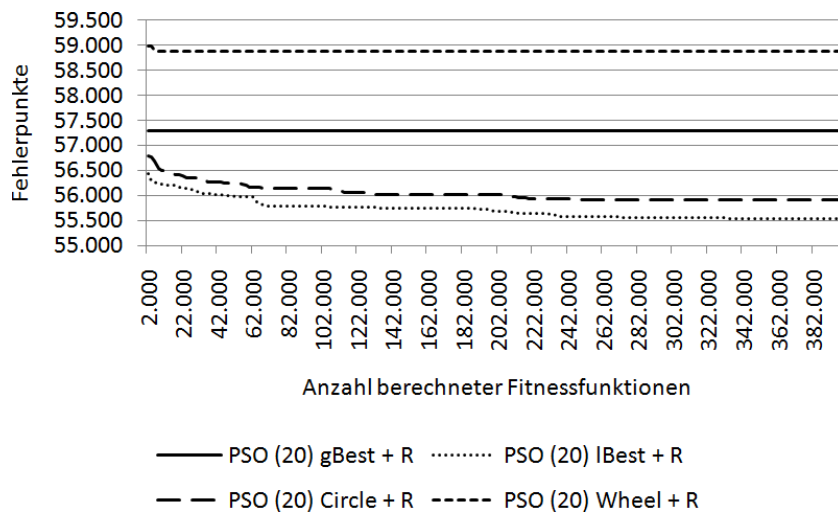


Abbildung 6.4: Verlauf der Fehlerpunkte verschiedener Nachbarschaftstopologien für PSO bei Logistik_S_Complete

Die vier Nachbarschaftstopologien werden auch an den einzelnen Tagen der beiden Problemvarianten und somit an deutlich kleineren Problemstellungen getestet. In Tabelle 6.5

sind für jeden Tag jeweils der durchschnittliche Fehler und der minimale Fehler für gBest, Wheel, Circle (k=2) und lBest (k=4) aufgeführt. Diese Tabelle zeigt die Ergebnisse der Tagesprobleme ohne Stammarbeitsplatzregelung und ohne Reparatur. Die vollständigen Daten können im Anhang den Tabellen 7.3 bis 7.31 entnommen werden. Vergleicht man die durchschnittlichen Fehler der Topologien an allen Tagen in Tabelle 6.5 (die besten Werte sind unterstrichen), so schneidet gBest nicht immer am besten ab. Nur an fünf von sieben Tagen schlägt gBest die anderen Topologien. Circle und lBest liefern meist ähnlich gute Werte. Wheel ist im Vergleich dazu immer recht schlecht. Auch in Tabelle 6.6, wo die Ergebnisse der einzelnen Tage mit Stammarbeitsplatzregelung und ohne Reparatur dargestellt sind, schneidet Wheel am schlechtesten ab. Dort sind nun aber gBest und Circle lBest und Wheel überlegen. Die Tatsache, dass gBest an den einzelnen Tagen nicht grundsätzlich zu bevorzugen ist, mag daran liegen, dass diese Problemstellungen weniger komplex als die jeweilige Variante des Wochenproblems sind. Es fällt dem Schwarm nun deutlich leichter, „gute“ Arbeitsplatzbesetzungen zu finden, wodurch der Einfluss maximalen Informationsaustausches innerhalb des Schwarms weniger ins Gewicht fällt. Der Verlust „guter“ Besetzungen kann deutlich besser kompensiert werden und der Vorteil der anderen Topologien bzgl. der Vermeidung einer vorzeitigen Konvergenz in einem lokalen Optimum wirkt nun z.T. stärker.

Tabelle 6.5: Ergebnisse verschiedener Nachbarschaftstopologien mit 20 Partikeln an den einzelnen Tagen des Logistikproblems ohne Stammarbeitsplatzregelung und ohne Reparatur (beste \circ und Min. unterstrichen)

Nachbarschafts- topologie	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag
gBest \circ	7.782	<u>5.918</u>	<u>8.173</u>	<u>8.246</u>	5.573	<u>8.825</u>	<u>7.301</u>
gBest Min.	7.706	<u>5.904</u>	<u>8.146</u>	<u>8.230</u>	5.504	<u>8.817</u>	<u>7.292</u>
Wheel \circ	7.968	6.205	8.485	8.423	5.754	8.958	7.496
Wheel Min.	7.896	6.079	8.348	8.360	5.667	8.910	7.424
Circle \circ	7.746	5.926	8.194	8.256	5.528	8.833	7.308
Circle Min.	<u>7.701</u>	5.908	8.154	8.240	5.505	8.824	7.299
lBest \circ	<u>7.744</u>	5.928	8.194	8.254	<u>5.527</u>	8.834	7.310
lBest Min.	7.705	5.913	8.157	8.236	<u>5.498</u>	8.823	7.302

Ähnlich wie bei ES sind auch bei PSO maximal fünf Manntage als Aufwand für Implementierung und Parametereinstellung zu veranschlagen. Bei der Parametereinstellung der vier Aktionen für die Positionsneuberechnung muss lediglich die Eintrittswahrscheinlichkeit für Aktion zwei exakt angepasst werden. Alle anderen drei Parameter sind recht unempfindlich gegenüber Änderungen. Es wird die gleiche Reparatur wie bei ES genutzt, deren Ablauf mit Hilfe von detaillierten Ergebnisanalysen hergeleitet wurde.

Tabelle 6.6: Ergebnisse verschiedener Nachbarschaftstopologien mit 20 Partikeln an den einzelnen Tagen des Logistikproblems mit Stammarbeitsplatzregelung und ohne Reparatur (beste \emptyset und Min. unterstrichen)

Nachbarschafts-topologie	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag
gBest \emptyset	7.263	<u>5.741</u>	7.901	<u>7.234</u>	5.354	7.234	<u>6.333</u>
gBest Min.	7.199	<u>5.685</u>	7.852	7.203	5.298	7.221	6.320
Wheel \emptyset	7.456	6.048	8.128	7.335	5.489	7.324	6.470
Wheel Min.	7.313	5.907	8.034	7.270	5.432	7.289	6.414
Circle \emptyset	<u>7.258</u>	5.745	<u>7.901</u>	7.237	<u>5.348</u>	<u>7.231</u>	6.334
Circle Min.	<u>7.177</u>	5.686	7.815	7.198	5.284	<u>7.215</u>	<u>6.311</u>
lBest \emptyset	7.278	5.750	7.905	7.241	5.350	7.232	6.334
lBest Min.	7.183	5.694	<u>7.814</u>	<u>7.196</u>	<u>5.274</u>	7.216	6.317

6.1.8 Multiagentensysteme

Das Ergebnis des Agentenansatzes liegt bei Logistik_Complete und den zugehörigen Tagesproblemen recht nahe an den Bestwerten, die mit anderen Verfahren erzielt wurden. Es kommt zu keinen Qualifikationsfehlern und die Besetzungsabweichungen sind auf das mögliche Minimum reduziert. Da der Agentenansatz in seinem Kern eine Lösung mit Hilfe von problemspezifischem Wissen konstruiert – ähnlich der Reparatur bei ES und PSO – können Qualifikations- und Besetzungsfehler auf das mögliche Minimum reduziert werden. In diesem Zusammenhang ist das Ergebnis optimal. Es treten jedoch unnötige Arbeitsplatzwechsel auf, was das Gesamtergebnis etwas schlechter ausfallen lässt. Bei deutlich kleineren Problemstellungen – nämlich den einzelnen Tagen des Logistikproblems – schneidet der Agentenansatz ebenfalls recht gut ab. Es werden jedoch auch hier nicht die Ergebnisse des jeweils besten Verfahrens erreicht. Sollte jedoch die Rechenzeit eine kritische Größe sein, so ist der Agentenansatz bei relativ einfachen Problemen die erste Wahl, wenn man sich mit ein wenig mehr Arbeitsplatzwechseln als nötig begnügt. Das Problem Logistik_Complete wird in unter zwei Sekunden und alle Tagesprobleme werden in unter einer Sekunde gelöst.

Vergleichbare CPU-Zeiten treten bei Logistik_S_Complete und den entsprechenden Tagesproblemen auf. Dort sind die Ergebnisse jedoch schlechter, was vor allem an Stammarbeitsplatzfehlern liegt. Obwohl hierzu explizit Verhandlungen vorgesehen sind, die derartige Fehler beheben sollen. Es gelingt jedoch, Qualifikationsfehler gänzlich zu vermeiden und auch die Fehler für Besetzungsabweichungen sind meist sehr gut – können jedoch nicht immer auf den Optimalwert reduziert werden. Auch bei den Problemstellungen mit Stammarbeitsplatzregelung treten unnötige Arbeitsplatzwechsel auf.

Trotz einer sehr individuellen Anpassung der Verhandlungen an die vorliegenden Pro-

blemvarianten können Arbeitsplatzwechsel oder Stammarbeitsplatzfehler nicht weiter reduziert werden. Grundsätzlich wären weitere Verbesserungen möglich, was durch die niedrigeren Fehlerpunktzahlen bei anderen Verfahren belegt wird. Dies würde jedoch nur dann erreicht werden, wenn deutlich umfangreichere Umplanungen erfolgen. Die Verhandlungen eines Mitarbeiters mit dem Service-Agenten oder mit anderen Mitarbeitern über eine Umbesetzung bzw. einen Tausch genügen nicht. Eine Verbesserung wäre dann möglich, wenn die Änderungen tiefer gestaffelt sind. Zusätzlich zu den bisherigen Verhandlungen müssten weitere Umbesetzungen anderer Mitarbeiter geprüft werden. Zudem müssten Ringtausche von drei oder mehr Mitarbeitern durchgeführt werden. Und das nicht nur zu einem Zeitpunkt, sondern u.U. für weite Perioden des Planungszeitraums. Derartig komplexe Umplanungen sind nur durch eine zentrale Planungsinstanz zu gewährleisten, die sich über die Agenten hinweg setzt und eigenständig Umplanungen vornimmt. Dies widerspricht jedoch dem dezentralen Konzept der Agenten, wie es von Puppe u.a. [315] sowie von de Causmaecker u.a. [93] gefordert wird. Ein Mitarbeiter kann auch nur schwerlich Umbesetzungen an mehreren Tagen bei z.B. 15 der 65 Mitarbeiter gegen deren Willen veranlassen. Zudem wäre ein derartiges Vorgehen nur mit erheblichem Aufwand zu implementieren. Mit der Forderung nach einer zentralen Planungsinstanz ist nicht das Konzept des zentralen Agenten (Central Agent) von Meisels und Kaplansky [259] gemeint. Der zentrale Agent hat dort lediglich die Aufgabe, das Verhalten der Scheduling-Agenten (in Abbildung 4.23 sind die Agenten (3) und (7) Scheduling-Agenten) zu beeinflussen. Die hier verwendeten Agenten (2) und (6) haben ähnliche Aufgaben wie der zentrale Agent von Meisels und Kaplansky. Sie erzeugen einen oder mehrere Agenten, sie vollziehen Entscheidungen und Agent (6) stößt Verhandlungen an oder verhandelt selbst mit einem Agenten, wenn Verstöße festgestellt werden.

Die Umsetzung des Agentenansatzes setzt sehr viel problemspezifisches Wissen voraus, was im Vorfeld recht aufwändige Überlegungen erforderlich macht. Ist dieses Wissen und somit ein Konzept für den Ablauf vorhanden, kann die Implementierung sehr zügig erfolgen. Es existieren zudem keine Parameter, sodass Tests gänzlich entfallen. In Summe werden vor allem wegen der Erstellung eines Konzeptes für den Ablauf maximal fünf Manntage für die Implementierung veranschlagt.

6.1.9 Hybride Lösungsverfahren

Wie bei TS (Kapitel 6.1.5) basieren auch diese Ausführungen auf der Diplomarbeit von Helbig [178] und den dort durchgeführten Experimenten. Daher beziehen sich alle nachfolgenden Aussagen ebenfalls ausschließlich auf die Problemvarianten ohne Stammarbeitsplatzregelung. Die Ergebnisse sind im Anhang in den Tabellen 7.1 bis 7.15 ersichtlich. Die hybride Variante trägt z.B. die Bezeichnung TS I3 + HC3. I3 steht für die Initialisierungsvariante, hier: 3. Und HC3 bezeichnet den HC, der Blöcke mit einer Länge von bis zu 3 Zeitintervallen tauscht. Beim Wochenproblem ist zudem F für First Accept (statt Best Accept), -A für Experimente ohne das Aspirationskriterium, /T für

eine Zerlegung des Wochenproblems in Tage und TS* für ein erhöhtes Abbruchkriterium von 2,8 Mio. Berechnungen der Fitnessfunktion (statt 400.000) zu finden.

Betrachtet man die Ergebnisse der hybriden Variante aus TS und HC, so wird deutlich, dass die Erweiterung von TS um HC fast immer zu signifikant besseren Ergebnissen führt. Sie ist sogar so gut, dass der hybride Ansatz bei allen Tagesproblemen neue Bestwerte erzielt. Und dies im Durchschnitt sowie im Minimum. Im Rahmen von 15 t-Tests mit $\alpha = 0,05$ wurden TS I2 mit TS I2 + HC3 sowie TS I3 mit TS I3 + HC3 verglichen. Lediglich in zwei Fällen – bei TS I3 mit TS I3 + HC3 am Mittwoch und am Wochenproblem – gibt es keinen signifikanten Unterschied der Mittelwerte. In allen anderen Tests sind die Mittelwerte des hybriden Ansatzes signifikant besser als die von TS. TS + HC ist demnach äußerst gut für das Logistikproblem ohne Stammarbeitsplatzregelung geeignet. Es fällt jedoch auf, dass wie bei TS (Kapitel 6.1.5) auch hier nicht immer die Minimalwerte für die Über- und Unterdeckung im Personaleinsatz gefunden werden. Die Abweichungen sind jedoch minimal. Sie zeigen aber, dass die systematische Reparatur bei PSO und ES in diesem Punkt überlegen ist. Eine hybride Variante aus TS + HC + Reparatur würde die Ergebnisse minimal verbessern können. Die Rechenzeit würde sich wie bei PSO und ES jedoch verdoppeln.

Analog zu TS (Kapitel 6.1.5) wird auch hier die Initialisierung mit der Variante 1 nicht weiter verfolgt, da sie zu sehr schlechten Ergebnissen führt. Selbst der zusätzliche HC ist mit derart vielen Arbeitsplatzwechseln überfordert. Es ist also ein Zusammenhang zwischen Startlösung und Endergebnis zu erkennen. Die Startlösung muss wie bei TS recht strukturiert sein. Sie sollte nach Möglichkeit keine Arbeitsplatzwechsel enthalten, da diese nur mit hohem Aufwand minimiert werden können. Im Vergleich der Startlösungen I2 und I3 zeigen sich bei sonst gleichen TS + HC-Varianten nur geringe Unterschiede in den Ergebnissen. Auch hier wurden t-Tests bei den einzelnen Tagen durchgeführt. Das Signifikanzniveau ist wiederum 95%. Die Mittelwerte der hybriden Ansätze sind im Vergleich von I2 mit I3 am Montag, Mittwoch, Donnerstag und Sonntag nicht signifikant verschieden. An den anderen Tagen sind die Mittelwerte von I3 signifikant besser. Die Initialisierungsvariante 3 ist nun also nicht mehr absolut überlegen. Die Ursache ist simpel. Die Ergebnisse beider Initialisierungsvarianten unterscheiden sich bei TS vor allem hinsichtlich der Anzahl an Arbeitsplatzwechseln. In der hybriden Form aus TS + HC hat HC die Aufgabe, Arbeitsplatzwechsel zu reduzieren, was auch recht gut gelingt. Daher sind die Ergebnisse beider Initialisierungsvarianten nun näher beieinander.

Im Vergleich von HC3 und HC8 bewirkt eine größere Anzahl an zu tauschenden Blöcken nicht zwangsläufig eine Ergebnisverbesserung. Zwar ist z.T. nur dann eine Reduzierung von Arbeitsplatzwechseln möglich, wenn längere Blöcke getauscht werden (HC8 statt HC3). Dies führt jedoch zu erheblich mehr Berechnungen der Fitnessfunktion innerhalb von HC. Diese fehlen dann bei TS, was der Ergebnisqualität schaden kann. Signifikanztests mit $\alpha = 0,05$ belegen, dass die Mittelwerte bei den Tagesproblemen mit TS I3 + HC8 nie besser als die mit TS I3 + HC3 sind.

Wie bei TS sind auch bei TS + HC die Ergebnisse am Wochenproblem nicht zufriedenstellend. TS I3 + HC3 erreicht im Durchschnitt nur 117.279,7 Fehlerpunkten. Bei TS I3 allein sind es immerhin 116.797,9 Fehlerpunkte (der Mittelwert von TS I3 ist jedoch nicht signifikant besser als der von TS I3 + HC3). Die Anzahl an Arbeitsplatzwechseln liegt bei TS I3 und TS I3 + HC3 im Durchschnitt sehr nahe beieinander. In der hybriden Variante kommt HC, welches die Reduzierung der Arbeitsplatzwechsel verfolgt, demnach nur sehr selten zum Einsatz. HC wird nämlich nur dann gestartet, wenn nach 50 Iterationen von TS keine Verbesserung eintritt. Dies scheint nicht sehr häufig zu geschehen. Wenn HC jedoch gestartet wird, werden nur Arbeitsplatzwechsel reduziert. Im Vergleich zu deutlich stärker gewichteten Qualifikationsverletzungen oder Abweichungen von den Besetzungsvorgaben reduziert HC die Gesamtfehlerpunktzahl nur minimal, verbraucht aber Berechnungen der Fitnessfunktion. Der Einsatz von HC nach zu wenigen Iterationen von TS ist daher nicht zielführend und erklärt das Ergebnis von TS I3 im Vergleich zu TS I3 + HC am Wochenproblem.

Mit weiteren Modifikationen will Helbig daher bessere Lösungen erzielen – ein späteres Starten von HC nach z.B. 200.000 berechneten Fitnessfunktionen prüft er jedoch nicht. Zunächst spart er Berechnungen, indem bei TS statt Best Accept, First Accept (F) als Akzeptanzkriterium gewählt wird. Experimente, in denen First Accept zum Einsatz kommt, sind mit (F) gekennzeichnet. Ist kein (F) angegeben, wird wie gewohnt Best Accept verwendet. Zudem wird das Aspirationskriterium Aspiration by Objective bei TS ausgeschlossen. Mit ihm war es möglich, einen tabu gesetzten Schritt dennoch auszuführen, wenn der Fitnesswert dadurch besser wurde. Experimente ohne das Aspirationskriterium sind mit (-A) gekennzeichnet. Ist nichts angegeben, ist Aspiration by Objective enthalten. Am Wochenproblem konnte mit TS I3 + HC3 (F) (-A) ein signifikant besserer Durchschnittswert von 68.902,1 Fehlerpunkten erzielt werden. Dies ist jedoch noch nicht zufrieden stellend.

Daher wird zusätzlich ein Umstand ausgenutzt, der nur beim Wochenproblem und nicht bei den einzelnen Tagen auftritt. Die Zuweisungen in den einzelnen Tagen beeinflussen keine anderen Tage. Daher wird der Lösungsraum auf die sieben Tage aufgeteilt. Die Suche springt nach 1/7 der maximal zulässigen Berechnungen der Fitnessfunktion zum nächsten Tag. Dieses Vorgehen ist mit /T gekennzeichnet. TS I3 + HC3 (F) (-A) /T liefert im Durchschnitt 67.320,7 Fehlerpunkte. Er ist jedoch nicht signifikant besser als der Mittelwert von TS I3 + HC3 (F) (-A) bei $\alpha = 0,05$ ($T = 0,697$; $p = 0,488$).

Zuvor wurden Modifikationen vorgestellt, um Berechnungen der Fitnessfunktion zu sparen. Da dies zu besseren Ergebnissen führt und da Helbig bei TS bereits gute Erfahrungen mit einem erweiterten Abbruchkriterium von 2,8 Mio. Berechnungen der Fitnessfunktion gemacht hat, werden auch hier entsprechende Experimente durchgeführt. Der hybride Ansatz mit dem erweiterten Abbruchkriterium ist ebenfalls mit einem TS* gekennzeichnet (z.B. TS* I3 + HC3). Alle so erweiterten hybriden Verfahren liefern signifikant bessere Ergebnisse als bei einem vorzeitigen Abbruch (mit $\alpha = 0,05$). Dennoch gelingt es am Wochenproblem nicht, die Bestwerte von PSO und ES zu unterbieten. Bei den

Experimenten mit 2,8 Mio. Berechnungen der Fitnessfunktion fällt auf, dass TS I3 + HC3 (F) (-A) recht schnell Verbesserungen der Gesamtfehlerpunkte erzeugt. TS I3 + HC3 benötigt deutlich länger. Ab etwa 1,2 Mio. Berechnungen der Fitnessfunktion ist dieses Verfahren jedoch besser. Denn die Modifikationen (F) und (-A) behindern die gründliche Suche im Lösungsraum langfristig.

Wie bei TS (Kapitel 6.1.5) lassen sich auch hier weitere Verbesserungen mit einer erheblichen Erhöhung der Abbruchschranke erzielen. Da dies nicht praktikable Rechenzeiten bedeutet, werden von Helbig hierzu keine weiteren Experimente durchgeführt. Es wird aber deutlich, dass TS und hybride TS-Varianten bei größer werdenden Problemstellungen eine erhebliche Anzahl an Iterationen für gute Ergebnisse benötigen.

Auch beim hybriden Ansatz macht Helbig [178] keine Aussagen zum Implementierungs- und Parametrierungsaufwand. Im Vergleich zu TS (Kapitel 6.1.5) dürfte der Aufwand für Implementierung und Parametrierung durch den zusätzlichen HC, bei dem die geeignete Blocklänge für einen Tausch zu wählen ist, ein wenig größer sein.

In Kapitel 4.16.2.1 wurden weitere hybride Ansätze vorgestellt. Hierbei handelt es sich um diverse Varianten in Kombination mit MAS sowie um ESO, einer Kombination aus ES und PSO. Da all diese hybriden Verfahren keine besseren Ergebnisse erzielen als die alleinige Anwendung eines der Verfahren, wurde auf weiterführende Experimente verzichtet.

6.1.10 Vergleich der Lösungsverfahren

In den vorangegangenen Unterkapiteln sind die Ergebnisse diverser Lösungsverfahren dargestellt worden, wobei auch auf Erkenntnisse zur geeigneten Parametrierung der Verfahren eingegangen wurde. Nachfolgend soll zunächst die zentrale Forschungsfrage anhand des Logistikproblems beantwortet werden.

Bereits in Kapitel 4.2 wurde gezeigt, dass sich eine vollständige Enumeration nicht für die Problemvarianten des Logistikdienstleisters eignet. Der benötigte Zeitaufwand für die Berechnungen sprengt alle praktikablen Grenzen. Dieses Problem ergibt sich auch, wenn klassische OR-Verfahren zum Einsatz kommen. In Kapitel 6.1.2 wurde deutlich, dass die GAMS-Solver die MINLP-Modelle nicht in akzeptabler Zeit lösen können. Diese MINLP-Modelle entsprechen exakt den jeweiligen Problemvarianten. Die Rechenzeit wird erst durch die Abbildung der Problemstellungen als MIP-Modell drastisch reduziert. Die ursprünglichen Probleme müssen hierzu jedoch stark geändert werden, was die Ergebnisse für die Ausgangsprobleme unbrauchbar macht. Zusammenfassend kann gesagt werden, dass sich klassische OR-Verfahren nicht für das Logistikproblem mit seinen Varianten eignet.

Tabelle 6.7: t-Tests (einseitig) bei Logistik_Complete

H_1	T	df	Signi- fikanz H_0	95% Konfidenzintervall der Differenz	
				untere	obere
PSO (20) I3 gBest + R < ES (10+50) I2 TypE + R	-26,399	58	< 0,001	-96,408	-84,926
PSO (20) I3 gBest + R < HC I1	-36,838	29,126	< 0,001	-1.386,140	-1.263,926
PSO (20) I3 gBest + R < TS VNS I3	-5,389	29,000	< 0,001	-6.231,129	-3.243,604
PSO (20) I3 gBest + R < TS I3 + HC3 (F) -(A) /T	-8,532	29,000	< 0,001	-18.669,112	-12.467,955
PSO (20) I3 gBest + R < MAS	-23,570	47,165	< 0,001	-82,767	-71,766

Tabelle 6.8: t-Tests (einseitig) bei Logistik_S_Complete

H_1	T	df	Signi- fikanz H_0	95% Konfidenzintervall der Differenz	
				untere	obere
HC I2 < PSO (20) I3 lBest + R	-154,021	58	< 0,001	-10.100,777	-9.883,889
HC I2 < ES (1+5) I2 TypG	-18,475	58	< 0,001	-7.491,978	-6.231,022
HC I2 < MAS	-99,347	29,816	< 0,001	-5.446,422	-5.266,031

In den Kapiteln 6.1.3 bis 6.1.7 wurden die Ergebnisse der Metaheuristiken HC, GLS, TS, ES und PSO interpretiert. Zudem wurde in Kapitel 6.1.9 auf eine hybride Metaheuristik aus TS und HC eingegangen. Bei den Problemvarianten ohne Stammarbeitsplatzregelung liefert der hybride Ansatz aus TS + HC die besten Ergebnisse an den einzelnen Tagen. Beim zugehörigen Wochenproblem erzeugt PSO die besten Ergebnisse (Abbruchkriterium mit 400.000 Berechnungen der Fitnessfunktion). Und mit einem erweiterten Abbruchkriterium von 2,8 Mio. Berechnungen der Fitnessfunktion liefert TS* I3 + HC3 neue Bestwerte am Wochenproblem. Für TS und den hybriden TS-Ansatz liegen keine Ergebnisse bei den Problemstellungen mit Stammarbeitsplatzregelung vor. Dort ist PSO an den einzelnen Tagen zumeist zu bevorzugen, in Ausnahmen auch ES. Interessant ist dort, dass der Einsatz der Reparatur, die bei den Varianten ohne Stammarbeitsplatzregelung zu besseren Ergebnissen führt, nun die Ergebnisse meist verschlechtert. Die Ursache liegt darin, dass die Reparatur die Suche bei ES und PSO zu stark einschränkt und zu viele Bereiche des Lösungsraums unerreichbar macht.

Erstaunlich gut schneidet HC bei allen Problemvarianten ab. Besonders, wenn man bedenkt, dass der Ablauf recht simpel ist. Bei Logistik_S_Complete liefert HC I2 sogar die Bestwerte. GLS ist für das Logistikproblem absolut ungeeignet, da die Problemstellung viel zu groß ist und GLS somit keinen Einfluss auf das Verhalten von HC hat.

Die entsprechenden Ergebnisse der t-Tests für die jeweils besten Konfigurationen der Lösungsverfahren am Wochenproblem ohne Stammarbeitsplatzregelung können Tabelle 6.7 entnommen werden. Für das Wochenproblem mit Stammarbeitsplatzregelung sind die entsprechenden Daten in Tabelle 6.8 dargestellt.

Für PSO und ES kann der Verlauf der Fehlerpunkte bei beiden Varianten des Wochenproblems grafisch veranschaulicht werden. In den Abbildungen 6.5 und 6.6 ist jeweils der beste Durchlauf von ES und PSO mit Reparatur dargestellt. Bei Logistik_Complete sind es ES (1+5) TypG + R sowie PSO (20) gBest + R und bei Logistik_S_Complete sind es ES (1,5) TypE + R sowie PSO (20) lBest + R. In beiden Fällen schneidet PSO am besten ab. Die Fehlerpunkte liegen beim Start je nach Initialisierungsvariante zwischen 500.000 und 300.000. Allgemein wird deutlich, dass ES wie auch PSO bereits innerhalb der ersten 2.000 berechneten Fitnessfunktionen die Fehlerpunkte stark reduzieren. Daher mussten auch die Werte der ersten 2.000 Berechnungen in den Abbildungen entfallen, da der Kurvenverlauf andernfalls einen rechten Winkel darstellen würde.

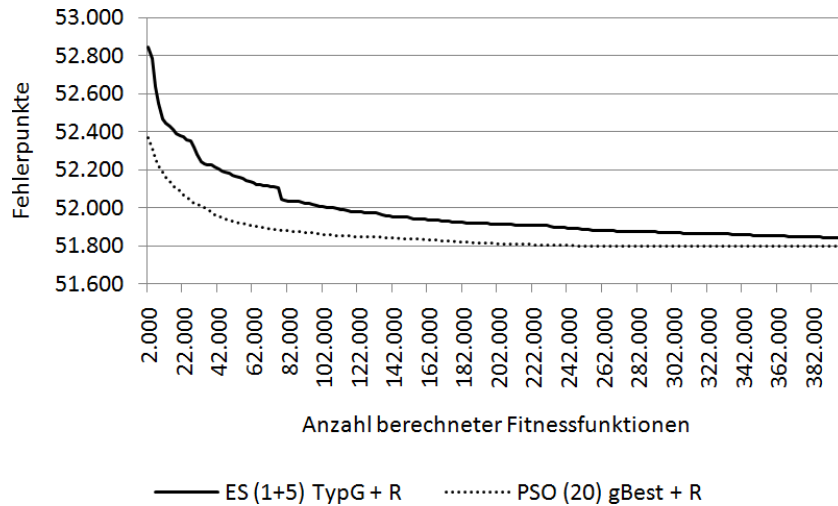


Abbildung 6.5: Verlauf der Fehlerpunkte für ES und PSO bei Logistik_Complete

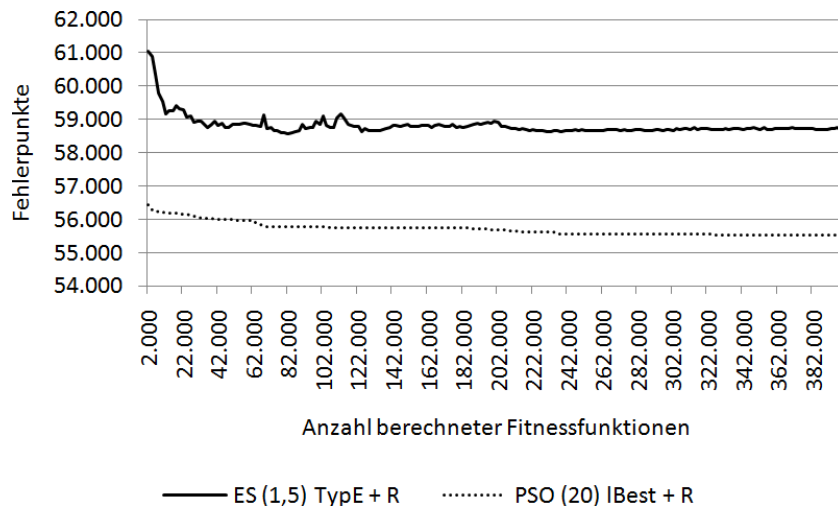


Abbildung 6.6: Verlauf der Fehlerpunkte für ES und PSO bei Logistik_S_Complete

Allgemein kann eine z.T. recht starke Abhängigkeit der Ergebnisqualität von der Initialisierungsvariante festgestellt werden. Sehr geordnete Startlösungen mit möglichst wenigen Arbeitsplatzwechsellern sind zwar nicht immer zu bevorzugen, liefern bei den meisten Verfahren jedoch bessere Ergebnisse als Startlösungen mit vielen Arbeitsplatzwechsellern.

Neben der geeigneten Wahl der Initialisierungsvariante kann das Ergebnis auch durch eine Zerlegung der Problemstellung verbessert werden. Beim Logistikproblem ist dies angeraten, da sich die einzelnen Tage nicht gegenseitig beeinflussen und daher problemlos eigenständig lösbar sind. Die Summe der besten Tagesergebnisse ist immer besser als die Ergebnisse an den Wochenproblemen. An den 16 Problemvarianten aus der Logistik wird deutlich, dass keine generelle Aussage gemacht werden kann, welche der getesteten Metaheuristiken zu bevorzugen ist.

Nachdem gezeigt wurde, dass klassische OR-Verfahren ungeeignet sind und einige Metaheuristiken sehr gute Ergebnisse liefern, bleibt zu klären, wie gut sich MAS eignen. Zwar bleiben sie hinsichtlich der Ergebnisqualität hinter den Bestwerten anderer Verfahren zurück, sind dafür jedoch extrem schnell. Ergebnisse werden immer innerhalb von zwei Sekunden geliefert. Auch wenn die Ergebnisse verbesserungswürdig sind, so halten sich die zusätzlichen Fehler doch in Grenzen. MAS sind also dann die erste Wahl, wenn die CPU-Zeit der begrenzende Faktor ist (nicht in dieser Arbeit). In den Tabellen 6.7 und 6.8 sind ebenfalls die t-Tests für MAS zu finden.

Einen Überblick über die Mittelwerte relevanter Verfahren zeigt Abbildung 6.7 für das Problem `Logistik_Complete`. Auf die Abbildung der sieben Tagesprobleme wurde aus Platzgründen verzichtet. Die senkrechten Linien markieren den Bereich von den Minimal- bis zu den Maximalwerten der 30 Replikationen. Während HC I1, HC I2, MAS sowie die abgebildeten PSO- und ES-Konfigurationen recht nah beieinander liegen, schneiden die dargestellten TS- und TS + HC-Varianten deutlich schlechter ab. Sie haben im Gegensatz zu den einfacheren Tagesproblemen nun auch eine sehr viel größere Streuung der Ergebnisse. Bei den PSO- und ES-Konfigurationen wird die Streuung durch den Einsatz der Reparatur reduziert.

Analog zu Abbildung 6.7 wird in Abbildung 6.8 für das Problem `Logistik_S_Complete` ein Überblick über Mittelwerte sowie über den Bereich der Minimal- und Maximalwerte relevanter Verfahren gegeben. Zu beachten ist hier die andere Skalierung der y-Achse im Vergleich zu Abbildung 6.7. Die Streuung aller Stichproben ist nun sehr gering. Man erkennt sehr deutlich das überaus gute Abschneiden von HC I1 und I2. Selbst das MAS ist beim Wochenproblem nun besser als PSO und ES.

Die CPU-Zeiten sind bei ES und PSO relativ ausgeglichen. Bei den beiden Varianten des Wochenproblems benötigen sie ohne Reparatur etwa 25 Minuten. Mit Reparatur sind es bereits 50 Minuten. HC hingegen führt intern weniger Berechnungen in jeder Iteration aus und verfügt über keine Reparatur, sodass das Ergebnis bereits nach etwa 15 Minuten geliefert wird. Für TS sowie TS + HC liegen keine CPU-Zeiten vor. Die

6.1 Untertägige Personaleinsatzplanung in der Logistik

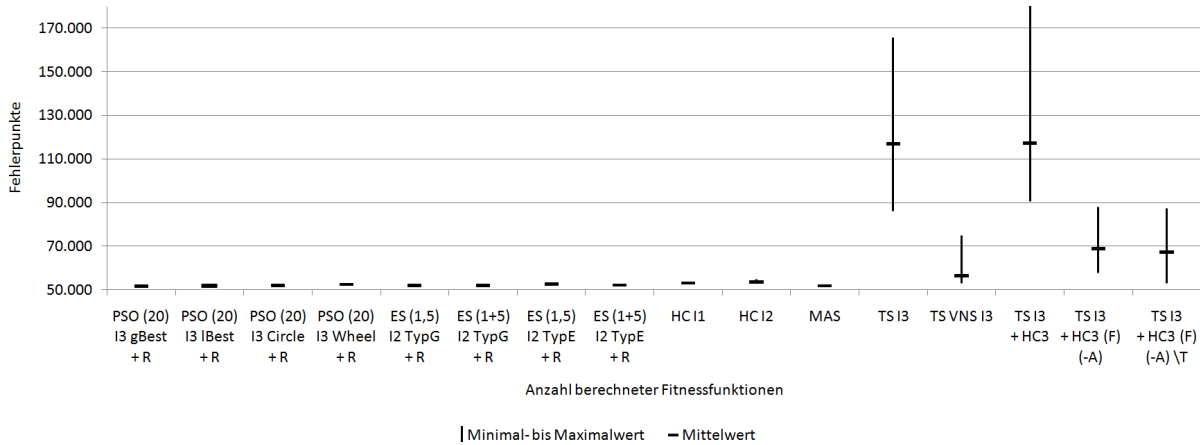


Abbildung 6.7: Vergleich verschiedener Lösungsverfahren bei Logistik_Complete

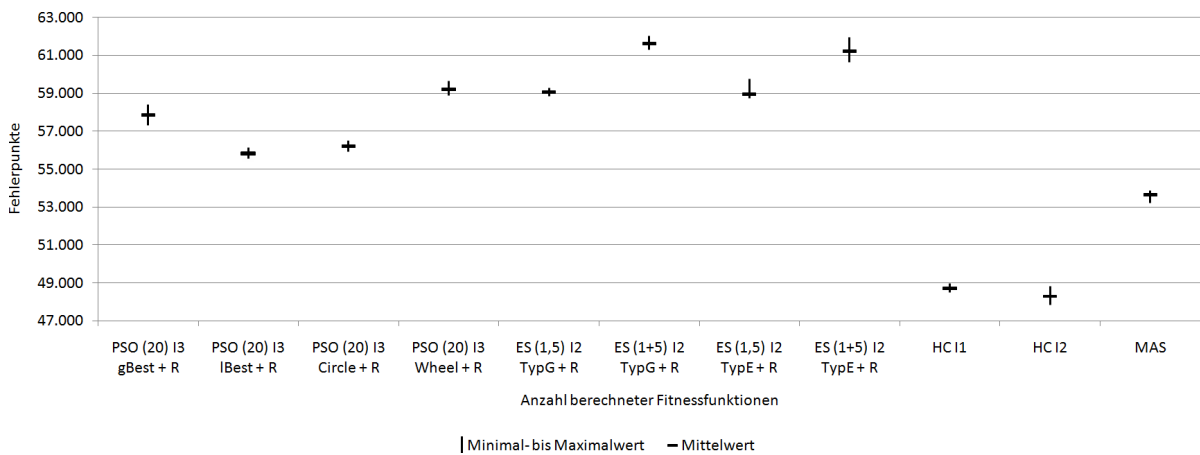


Abbildung 6.8: Vergleich verschiedener Lösungsverfahren bei Logistik_S_Complete

vollständige Enumeration und klassische OR-Verfahren sprengen hinsichtlich der CPU-Zeiten alle praktikablen Grenzen. Das mit Abstand schnellste Verfahren ist MAS mit weniger als zwei Sekunden am Wochenproblem.

Der Implementierungs- und Parametrierungsaufwand ist bei allen Verfahren, zu denen die Aufwände bekannt sind, in etwa gleich groß. Besonders schnell lässt sich HC mit der sehr systematischen Suche implementieren. Dort entfällt auch die Parametrierung gänzlich, da es keine Parameter gibt. Im Vergleich von ES und PSO ist PSO mit etwas weniger Aufwand parametrierbar, da lediglich die Eintrittswahrscheinlichkeit der zweiten Aktion im Rahmen der Positionsneuberechnung exakt angepasst werden muss. Alle anderen Parameter sind recht unempfindlich gegenüber Änderungen. Zudem sind viele Parameter wie Inertia Weight, Constriction Factor, etc. durch die Modifizierung von PSO für kombinatorische Probleme überflüssig geworden. Während bei HC kaum problemspezifisches Wissen für die Implementierung erforderlich ist, konnte das MAS ohne dieses Wissen nicht entwickelt werden. Ähnlich verhält es sich bei den Repara-

turverfahren von ES und PSO, die auf Basis eingehender Ergebnisanalysen entstanden sind.

Hinsichtlich der zentralen Forschungsfrage kann gesagt werden, dass sich am ehesten Metaheuristiken, gefolgt von MAS eignen. Klassische OR-Verfahren hingegen sind nicht praktikabel anwendbar.

Somit kann für das Logistikproblem auch die Forschungsfrage beantwortet werden, ob sich die Aussage von Puppe u.a. [315] auf dieses Problem übertragen lässt. Hiernach eignen sich zentrale Lösungsverfahren weniger gut für die Dienstplanung als MAS (dezentral). Es ist deutlich geworden, dass sich ausgewählte Metaheuristiken – also zentrale Lösungsverfahren – besonders gut für das Logistikproblem eignen. Zwar schneidet das dezentrale MAS recht gut ab. Wie zuvor dargestellt, leidet es aber gerade unter dem Fehlen einer zentralen Planungsinstanz, die tiefgreifende Änderungen vornimmt.

Die gefundenen Bestwerte von TS, HC, TS + HC oder PSO sind bereits recht nahe am Lower Bound. Und selbst nach sehr aufwändigen manuellen Prüfungen der Einsatzpläne lassen sich nur äußerst selten Verbesserungen finden. Die Ergebnisse können daher als überaus gut bezeichnet werden. Die Forschungsfrage, ob sich die getesteten Lösungsverfahren auch für Problemstellungen eignen, die direkt aus der Praxis stammen, kann für das Logistikproblem zwar nicht ausnahmslos bejaht werden – klassische OR-Verfahren oder GLS sind z.B. äußerst ungeeignet. Mit Blick auf die Bestwerte diverser Metaheuristiken kann aus dieser Gruppe jedoch eine geeignete Wahl getroffen werden, sodass sich ausgewählte Lösungsverfahren durchaus für das Logistikproblem – eine Problemstellungen aus der Praxis – eignen. Hier vor allem TS, TS + HC sowie PSO.

Eine weitere Forschungsfrage bezieht sich auf die betriebswirtschaftlichen Effekte, die mit hochflexiblem WFM realisiert werden können. Am Beispiel der untertägigen Planung des Logistikdienstleisters wird klar, dass die Effekte enorm sind. So werden je nach Lösungsverfahren (im Vergleich zur bisherigen manuellen Planung ohne Arbeitsplatzwechsel) die Überdeckung im Personaleinsatz um bis zu 26% und die Unterdeckung um bis zu 63% reduziert. Diese Ergebnisse machen die Bedeutung von Instrumenten des hochflexiblen WFM deutlich. Hier: Planung untertägiger Arbeitsplatzwechsel für den passgenauen Personaleinsatz bei volatilem Personalbedarf.

6.2 Automatische Arbeitszeitmodellerstellung im Handel

In diesem Kapitel werden für das Handelsproblem die Ergebnisse verschiedener Lösungsverfahren erläutert. Die Daten der Experimente sind im Anhang in den Tabellen 7.33

bis 7.36 dargestellt. Da keines der erprobten Verfahren Fehlerpunkte für eine Überplanung bei einem Bedarf von null erzeugt, wurde diese Spalte in den Ergebnistabellen aus Platzgründen weggelassen. Ergebnisse, die nicht im Anhang stehen, sind nachfolgend im entsprechenden Kapitel zu finden.

Zunächst wird in Kapitel 6.2.1 auf den Lower Bound eingegangen. Anschließend folgen die Erläuterungen zu den einzelnen Lösungsverfahren. Das Ergebnis für das KV von Sauer und Schumann [334] wurde freundlicher Weise von Schumann für die Probleme `Handel_1AP` und `Handel_1AP_Jan` berechnet. Das zweite KV ist Bestandteil einer kommerziellen Lösung, auf die Zugriff bestand. Alle anderen Verfahren sind selbst implementiert worden.

6.2.1 Lower Bound

Ein analoges Vorgehen wie beim Logistikproblem zur Herleitung des Lower Bounds ist hier nicht möglich. Dies liegt an den Regeln für die zu erstellenden Arbeitszeitmodellen, die im MIP-Modell nicht exakt berücksichtigt werden können. Es macht also keinen Sinn, ein vereinfachtes MIP-Modell zu lösen. Ein Lower Bound wird daher hergeleitet – bzw. es werden Aussagen darüber gemacht, bei welchen Nebenbedingungen auf keinen Fall Fehler auftreten können. Die nachfolgenden Aussagen gelten für alle vier Varianten des Handelsproblems: `Handel_1AP`, `Handel_1AP_Jan`, `Handel_2AP` und `Handel_2AP_Jan`.

Bei den Arbeitsplatzwechseln ist klar, dass diese in Problemen mit nur einem Arbeitsplatz (`Handel_1AP` und `Handel_1AP_Jan`) nicht auftreten können. In den Problemen `Handel_2AP` und `Handel_2AP_Jan` sind sie sehr wohl erforderlich, um den Bedarf optimal zu decken. Eine Aussage über die minimale Anzahl kann dort leider nicht gemacht werden.

Hinsichtlich der zu erstellenden Arbeitszeitmodelle gibt es die Regel, dass sie nicht kürzer als drei und nicht länger als neun Stunden sein dürfen. Zudem dürfen sie am Tag nicht unterbrochen sein. Betrachtet man das Bedarfsgebirge der einzelnen Tage, so existieren keine losgelösten Bedarfe, die kleiner als drei Stunden sind. Das Bedarfsgebirge ist an jedem Tag immer zusammenhängend. Da die Regeln für die Arbeitszeitmodellerstellung immer die größten Fehlerpunkte im Vergleich zu allen anderen weichen Nebenbedingungen haben, ist die Einhaltung dieser Regeln immer vorteilhaft. Die Einhaltung ist auch immer möglich. Verstöße gegen die drei Regelungen für die Erstellung von Arbeitszeitmodellen dürfen im Optimum demnach nicht auftreten. Es ergeben sich dann zwangsläufig Über- oder Unterdeckungen sowie ggf. Verstöße gegen die maximale vertragliche Sollarbeitszeit je Woche.

Die minimalen Verstöße gegen die wöchentliche Sollarbeitszeit kann berechnet werden. Dem Stundenbedarf je Woche kann die verfügbare Sollarbeitszeit aller Mitarbeiter ge-

genübertestellt werden. Die Sollarbeitszeit kann wöchentllich schwanken, da Mitarbeiter nicht immer an allen Tagen anwesend sind (Urlaub, Krankheit, etc.) und sich die Sollarbeitszeit auch aufgrund von Feiertagen reduzieren kann. Ist der Bedarf größer als die verfügbaren Sollstunden, kann die Überschreitung der wöchentllichen Arbeitszeit in Minuten in Kauf genommen werden, um Unterdeckungen zu vermeiden. Möchte man die wöchentlliche Sollarbeitszeit nicht überschreiten, führt dies zwangsläufig zu Unterdeckungen, da einfach nicht genügend Mitarbeiter zur Verfügung stehen. Beide Fehlerarten sind gleich hoch bewertet. Sie werden nachfolgend für die vier Problemstellungen aufgeführt.

- Handel_1AP: 2.760 Fehlerpunkte
- Handel_1AP_Jan: 420 Fehlerpunkte
- Handel_2AP: 780 Fehlerpunkte
- Handel_2AP_Jan: 0 Fehlerpunkte

Blickt man auf die bisher gefundenen Bestwerte, so sind die 3.840 Fehlerpunkte für Handel_1AP und die 4.911 Fehlerpunkte für Handel_2AP beim kompletten Jahr noch relativ weit von den hergeleiteten Werten entfernt. Handel_2AP_Jan ist mit zehn Fehlerpunkten recht nahe am Lower Bound von null Punkten. Die z.T. starken Abweichungen vom Lower Bound liegen daran, dass Arbeitsplatzwechsel und Überdeckungen nicht berücksichtigt sind. Zudem dürften die optimalen Werte für Unterdeckungen z.T. deutlich höher sein. Regelungen für die Arbeitszeitmodellerstellung und Abwesenheiten in Kombination mit einem stark schwankenden Bedarf können zu zusätzlichen Verstößen gegen die wöchentlliche Sollarbeitszeit sowie zu Unterdeckungen führen. Derartige Teilaspekte des Lower Bounds sind nicht herleitbar und müssen folglich ausgelassen werden. Lediglich bei Handel_1AP_Jan werden von mehreren Lösungsverfahren exakt 420 Fehlerpunkte als Minimalwert erzielt. Dort wurde folglich das Optimum gefunden. Bei einer sehr zeitintensiven manuellen Überprüfung des jeweils besten Einsatzplans der drei übrigen Problemvarianten konnten nur in Ausnahmen weitere Verbesserungen gefunden werden. Es ist davon auszugehen, dass die Optima dieser drei Problemvarianten sehr nahe am jeweiligen Bestwert liegen.

6.2.2 Ganzzahlige Optimierung

Bereits beim Logistikproblem wurde in Kapitel 6.1.2 gezeigt, dass sich das originale Problem nicht als MIP-Modell abbilden lässt. Untertägige Arbeitsplatzwechsel und erhöhte Fehlerpunkte, wenn kein Personalbedarf auf einem Arbeitsplatz vorliegt, führen zu Nichtlinearitäten. Ein MINLP-Modell ist daher zwingend erforderlich. Das Logistikproblem lässt sich jedoch nicht in annehmbarer Zeit mit Hilfe von MINLP-Solvern lösen.

Im Gegensatz zum Logistikproblem haben die beiden Varianten des Handelsproblems für das komplette Jahr ungleich mehr Dimensionen. Bei den beiden Monatsvarianten sind es noch immer erheblich mehr Dimensionen als bei Logistik_Complete_20. Die Varianten des Handelsproblems wären bereits bei sonst gleichen Nebenbedingungen somit ebenfalls nicht in annehmbarer Zeit lösbar. Hinsichtlich der Nebenbedingungen gestaltet sich das Handelsproblem aber deutlich komplexer. Es müssen nun zusätzlich Regelungen für die Erstellung von Arbeitszeitmodellen berücksichtigt werden. Aber allein schon die untertägigen Arbeitsplatzwechsel (nur bei den Problemvarianten mit zwei Arbeitsplätzen) und erhöhte Fehlerpunkte, wenn kein Personalbedarf auf einem Arbeitsplatz vorliegt (bei allen Problemvarianten), führen zu Nichtlinearitäten. Der Lösungsraum ist aufgrund der zusätzlichen Nebenbedingungen weniger flach gestaltet als beim Logistikproblem und die Anzahl an Optima ist demzufolge geringer.

Auf die Abbildung des NP-schweren Handelsproblems als MINLP-Modell wurde daher verzichtet. Ergebnisse sind innerhalb einer praktikablen Rechenzeit nicht zu erwarten. Bzgl. des Implementierungsaufwands ist eine exakte Aussage nicht möglich, da die GAMS-Solver nicht selbst entwickelt wurden und in der Literatur keine Aussagen zu diesem Thema zu finden sind.

6.2.3 Konstruierende Verfahren

Im Rahmen dieser Arbeit wurden zwei verschiedenen KV erprobt. Deren Ergebnisse werden nachfolgend dargestellt und erläutert. Die entsprechenden Werte sind in den Tabellen 7.33 bis 7.36 im Angang zu finden. KV 1 steht dabei für das Verfahren von Sauer und Schumann, wogegen KV 2 für das KV einer weiteren kommerziellen Softwarelösung (Black Box) steht.

Beiden Verfahren ist gemein, dass ihre Ergebnisse im Vergleich zu anderen Verfahren recht schlecht sind. Und dies, obwohl sie jeweils Teil einer kommerziellen Software sind und explizit für derartige Problemstellungen entwickelt wurden. Zudem werden sie bei vielen namhaften Unternehmen des Einzelhandels eingesetzt. Sie können daher trotz ihres schlechten Abschneidens als Benchmark gesehen werden.

Verfahren von Sauer und Schumann Das KV von Sauer und Schumann kann nur auf Problemstellungen mit einem Arbeitsplatz angewendet werden. Daher existieren keine Ergebnisse für Handel_2AP und Handel_2AP_Jan. Obwohl es stochastische Elemente enthält – Entscheidung bei Gleichstand im Ranking von auszuwählenden Tagen – werden bei Handel_1AP und Handel_1AP_Jan immer die gleichen Ergebnisse erzielt. Bei der Problemstellung treten demnach keine oder wenige Situationen auf, in denen es zu einem Gleichstand beim Ranking der Tage kommt. Dies ist nicht ungewöhnlich, da die

verschiedenen Wochen und Tage beim Handelsproblem sehr unterschiedliche Gesamtbedarfe haben. Zudem hat die zufällige Auswahl bei einem Gleichstand in den seltenen Fällen ihres Auftretens anscheinend keinen Einfluss auf das Ergebnis. Die Berechnungen wurden auf einem Intel Core Duo 2 x 2,0 GHz und 3 GB RAM mit Windows Vista 32-Bit durchgeführt. Die CPU-Zeiten betragen im Durchschnitt 116 Sekunden bei Handel_1AP (27.660 Fehlerpunkte) und 10 Sekunden bei Handel_1AP_Jan (1.920 Fehlerpunkte).

Wie bereits erwähnt, sind die Ergebnisse nicht sonderlich gut. Deutlich ausgeprägt ist die Anzahl an Fehlerpunkten bei der Überdeckung im Personaleinsatz. Grundsätzlich hat das Verfahren beim Jahres- und Monatsproblem Schwierigkeiten, passgenaue Arbeitszeitmodelle zu generieren. Mitarbeiter werden oft zu lange eingeplant, obwohl sie nicht benötigt werden und es auch sonst keine Gründe wie etwa minimale Arbeitszeiten gibt. Dieser Umstand erzeugt die Überdeckungen. Hinsichtlich der Unterdeckung liefert das Verfahren beim kompletten Jahr recht gute Ergebnisse. Denn gerade auf die Reduzierung der Unterdeckung im Personaleinsatz wurde es ausgerichtet. Unter anderem wird dies jedoch durch eine Überschreitung der wöchentlichen Sollarbeitszeit in Kauf genommen. Beim Monatsproblem hält sich die Überschreitung der wöchentlichen Sollarbeitszeit in Grenzen. Dafür ist wiederum die Unterdeckung im Personaleinsatz sehr hoch.

Bei der eingehenden Betrachtung des Einsatzplans fallen sofort viele unnötige Fehler auf. Ein Großteil lässt sich durch die Verkürzung einiger Arbeitszeitmodelle korrigieren, was die Fehler bei der Überdeckung im Personaleinsatz und der Überschreitung der wöchentlichen Sollarbeitszeit deutlich reduziert. Der Einsatz der Reparaturmethode für PSO und ES (siehe Kapitel 4.12.4.2), die unabhängig von Optimierungsverfahren einsetzbar ist, würde sich nach Beendigung des KV anbieten. Aufgrund der bisher gemachten Erfahrungen und der beobachtenden Fehler ist eine Reduzierung der Gesamtfehlerpunktzahl um 30–40% zu erwarten.

Das Verfahren wurde nicht selbst implementiert, sodass keine Aussagen über den Implementierungsaufwand gemacht werden können. Die Spezifikation der Parameter legt jedoch den Schluss nahe, dass die Parametrierung sehr schnell erfolgen kann.

Black Box Über die Funktionsweise dieses KV dürfen keine Details bekannt gegeben werden. Es ist jedoch in der Lage, alle vier Varianten des Handelsproblems zu lösen, da es explizit für derartige Problemstellungen entwickelt wurde. Als Bestandteil einer kommerziellen Lösung wird es bei zahlreichen namhaften Unternehmen des Einzelhandels eingesetzt. Zum Aufwand, der bei der Implementierung des KV anfällt, ist ebenfalls nichts bekannt. Parameter können nicht eingestellt werden.

Das Verfahren enthält keine stochastischen Elemente und erzielt immer das gleiche Ergebnis. Daher genügt eine Replikation für die Experimente. Für einen Durchlauf werden bei Handel_2AP etwa zehn Minuten benötigt. Bei Handel_1AP, das 40% weniger Dimen-

sionen als Handel_2AP hat, werden etwa fünf Minuten benötigt. Die beiden Monatsprobleme beanspruchen ca. ein Zwölftel der entsprechenden CPU-Zeiten. Damit erfordert dieses KV etwa 2,5 mal mehr Zeit als das KV von Sauer und Schumann, wobei letzteres auf einem etwas langsamerem Rechner getestet wurde.

Die Ergebnisse dieses Verfahrens sind bei beiden Handelsproblemen recht schlecht. Es schneidet bei Handel_1AP (25.770 Fehlerpunkte) jedoch ein wenig besser ab als das Verfahren von Sauer und Schumann. Dort ist es bei den Fehlerpunkten für die Überschreitung der wöchentlichen Sollarbeitszeit und bei der Reduzierung der Überdeckung besser als KV 1. Dafür hat es mehr Probleme bei der Reduzierung der Unterdeckung. Ein völlig anderes Bild zeigt sich bei Handel_1AP_Jan (3.300 Fehlerpunkte). Dort ist KV 1 von Sauer und Schumann erheblich besser. Das Ergebnis der Black Box leidet unter einer massiven Überschreitung der wöchentlichen Sollarbeitszeit. Dafür sind die Abweichungen vom Personalbedarf geringer. Eine generelle Überlegenheit von KV 1 oder KV 2 ist demnach nicht festzustellen.

Obwohl KV 2 in der Lage ist, mit untertägigen Arbeitsplatzwechseln umzugehen und dies explizit in dessen Konfiguration eingestellt wurde, werden bei Handel_2AP (84.690 Fehlerpunkte) und Handel_2AP_Jan (9.480 Fehlerpunkte) keine Wechsel vollzogen. Die Möglichkeit, Unterdeckungen im Personaleinsatz durch geeignete Arbeitsplatzwechsel weiter zu reduzieren, wird nicht genutzt. In den Varianten Handel_1AP und Handel_1AP_Jan existiert nur ein Arbeitsplatz. Demzufolge können dort keine Arbeitsplatzwechsel auftreten.

Bezüglich der maximal zulässigen wöchentlichen Arbeitszeit fällt auf, dass das KV bei den Handelsproblemen keine Unterschiede zwischen den vertraglich fixierten wöchentlichen Sollarbeitszeiten der Mitarbeiter macht, obwohl auch dies in der Konfiguration hinterlegt wurde. Daher werden geringfügig beschäftigte Mitarbeiter wie Vollzeitmitarbeiter verplant, was den fertigen Einsatzplan unbrauchbar macht. Der Planer müsste derart viele manuelle Umplanungen vornehmen, dass er den Plan gleich selbst manuell hätte erstellen können.

Zum Aufwand für die Implementierung und Parametrierung liegen keine Informationen vor. In der Softwarelösung können vom Anwender lediglich Fehlerpunkte für Nebenbedingungen angegeben werden. Eine Parametrierung des Lösungsverfahrens ist in der Oberfläche nicht möglich und wäre nur direkt im Quellcode machbar.

6.2.4 Hill Climbing

Alle Ergebnisse für HC sind im Anhang in den Tabellen 7.33 bis 7.36 zu finden. I2 steht dabei für die Initialisierungsvariante 2. HC benötigt für Handel_2AP etwa zweieinhalb

Stunden. Bei Handel_1AP sind es etwa 75 Minuten. Die Zeiten bei den zugehörigen Monatsproblemen betragen ein Zwölftel der CPU-Zeiten für das Kalenderjahr.

Die deutlich schlechtesten Ergebnisse bei allen Handelsproblemen liefert HC mit Multistart im Vergleich aller getesteten Lösungsverfahren. HC ist das einzige Verfahren, bei dem Fehlerpunkte durch falsch erzeugte Arbeitszeitmodelle auftreten. Zudem ist die Höhe der Über- und Unterdeckungen im Vergleich zu allen anderen Verfahren ungleich größer. Bzgl. der Arbeitsplatzwechsel gibt es bei Handel_2AP_Jan eine Besonderheit. Dort treten keine Arbeitsplatzwechsel auf, obwohl HC dazu in der Lage wäre.

Da in diesem Verfahren nur wenig problemspezifisches Wissen integriert wurde, ist das schlechte Abschneiden von HC nicht verwunderlich. Zumal das Handelsproblem im Vergleich zum Logistikproblem erheblich komplexer ist. Der einfache Wechsel eines Arbeitsplatzes ist für die Erzeugung gültiger Arbeitszeitmodelle nicht zielführend. Die Regelungen der Arbeitszeitmodelle sind einfach zu komplex, sodass HC rasch in einem lokalen Optimum konvergiert.

Ähnlich wie beim Logistikproblem ist auch hier der Implementierungsaufwand trivial. Da keine Parameter einzustellen sind, wird der Aufwand mit maximal einem Manntag beziffert.

6.2.5 Guided Local Search

In Kapitel 6.1.3 wurde bereits dargestellt, warum sich GLS nicht für das Logistikproblem, wohl aber für das Problem der British Telecom eignet. Das entscheidende Kriterium ist die Anzahl an Features, die GLS nutzt, um die Suche zu lenken. GLS scheitert beim Logistikproblem, da dort mehrere tausend Features berücksichtigt werden müssen. Das Problem der British Telecom hingegen hat nur 250 Features, sodass GLS dort das Ergebnis von HC verbessert.

Das Handelsproblem unterscheidet sich von Logistikproblem durch deutlich mehr, sehr viel komplexere Nebenbedingungen für die Erstellung von Arbeitszeitmodellen und mit Blick auf GLS vor allem in der Anzahl an Dimensionen. Es ist bei Handel_2AP mit 131.400 Dimensionen um den Faktor 3,6 größer als Logistik_Complete. Lediglich das Problem Handel_1AP_Jan mit 6.696 Dimensionen ist kleiner. Jedoch überschreitet auch dort die Anzahl an Dimensionen bei Weitem die Zahl der 250 Features beim British Telecom-Problem. Zwar sind 250 Features nicht die obere Grenze für GLS. Eine zu große Zahl an Features erfordert jedoch eine sehr hohe Zahl an Iterationen, damit GLS überhaupt Wirkung zeigt. Wie beim Logistikproblem, wo GLS aufgrund der Vielzahl an Features wirkungslos ist, ist es auch beim Handelsproblem unbrauchbar. Der Einsatz von GLS führt lediglich zu einer Erhöhung der CPU-Zeit sowie zu Mehraufwand bei der Implementierung. Auf umfangreiche Experimente wurde daher verzichtet.

Wie beim Logistikproblem hält sich auch hier der Implementierungsaufwand in Grenzen. Lediglich die Fitnessfunktion muss erweitert werden. Dies ist wegen der höheren Anzahl an Nebenbedingungen ein wenig aufwändiger. Der Aufwand für die Parametrierung ist fast vernachlässigbar, da der einzige Parameter λ recht unempfindlich gegenüber Änderungen ist. In Summe wird nicht mehr als ein Manntag für Implementierung und Parametrierung benötigt.

6.2.6 Evolutionsstrategien

Wie in den vorangegangenen Kapiteln sind auch bei ES alle Ergebnisse im Anhang in den Tabellen 7.33 bis 7.36 zu finden. In Klammern ist die Anzahl der Eltern, der Selektionstyp (Plus- oder Kommaselektion) sowie die Anzahl der Nachkommen angegeben. TypG und TypE bezeichnen die Art der Mutation und R steht für die Reparatur.

Das beste minimale und signifikant beste durchschnittliche Ergebnis aller Lösungsverfahren (mit $\alpha = 0,05$) wird bei drei der vier Handelsproblemen mit Evolutionsstrategien erzielt. Dort werden alle Fehlerarten im Vergleich zu anderen Verfahren erheblich reduziert. ES(1,5) I1 TypG + R schneidet im Durchschnitt bei Handel_1AP, Handel_2AP und Handel_2AP_Jan am besten ab. Obwohl die Mutation vom TypE mit dem Konzept der maximalen Entropie nach Rudolph [329] speziell für ganzzahlige Lösungsräume entwickelt wurde, schneidet sie bei den drei Problemen im Durchschnitt nicht besser ab als TypG. TypE liefert jedoch den besten bisher gefundenen Plan bei Handel_2AP. Es zeigt sich hier also ein ähnliches Bild wie beim Logistikproblem (siehe Kapitel 6.1.6). In der vierten Problemstellung Handel_1AP_Jan, die im Vergleich zu den anderen drei Problemen am einfachsten ist, finden alle Verfahren, die eine Reparatur einsetzen, innerhalb der 30 Replikationen wenigstens einmal das Optimum – also alle Varianten von ES TypG, ES TypE und PSO. Den signifikant besten Durchschnittswert mit 420 Fehlerpunkten, der gleichzeitig das Optimum ist, erzeugt jedoch PSO (20) I1 Circle + R. ES (1,5) I1 TypG + R liegt im Durchschnitt mit 430 Fehlerpunkten recht nahe an diesem Wert.

ES benötigen beim Abbruchkriterium von 400.000 inspizierten Lösungen etwa sechs Stunden bei Handel_2AP und drei Stunden bei Handel_1AP. Dies liegt zum einen daran, dass eine Reparatur zwingend erforderlich ist, welche die CPU-Zeit verdoppelt. Experimente ohne Reparatur haben keine brauchbaren Ergebnisse geliefert. Massive Verstöße gegen die Regelungen der Arbeitszeitmodelle wurden permanent erzeugt und konnten von ES nur schwer beseitigt werden. Ohne Reparatur sind ES hinsichtlich der Ergebnisqualität nicht praktikabel anwendbar. Für die Monatsprobleme reduziert sich die CPU-Zeit auf ein Zwölftel der Zeit des entsprechenden Jahresproblems.

Betrachtet man die Selektionstypen für jeweils gleiche Konfigurationen von μ und λ , so schneidet die Plusselektion bei allen vier Handelsproblemen (24 t-Tests mit $\alpha = 0,05$) nie signifikant besser als die Kommaselektion ab. ES mit Plusselektion haben größere

Probleme aus lokalen Optima zu entkommen. Auf dem Weg von einem lokalen Optimum zu besseren Ergebnissen müssen meist deutlich schlechtere Ergebnisse über eine Vielzahl an Generationen in Kauf genommen werden. Bei der Plus-Strategie ist der Einfluss der Eltern mit relativ guten Ergebnissen einfach zu stark, um aus dem lokalen Optimum zu entkommen. Die Kommaselektion ist diesbezüglich deutlich besser. Dort werden in der Selektion nur die Kinder zu Eltern der neuen Generation. Dies lässt Ergebnisverschlechterungen zu, was das Entkommen aus einem lokalen Optimum erleichtert. Die Konvergenzgeschwindigkeiten der Komma- und Plusselektion ist in Abbildung 6.9 dargestellt. Dort ist der Verlauf von ES TypE und TypG mit einer (1,5)- und (1+5)-Selektion für die jeweils besten Durchläufe bei 30 Replikationen am Problem Handel_1AP_Jan abgebildet. Da es sich um die besten Durchläufe handelt, erreichen alle vier ES-Konfigurationen das Optimum von 420 Fehlerpunkten. Die ersten 2.000 Werte wurden weggelassen, da sich die Fehlerpunkte sehr rasch verbessern. Im Diagramm wäre nur ein rechter Winkel wie in Abbildung 4.2 zu erkennen. Es wird deutlich, dass die beiden ES-Konfigurationen mit Plusselektion sehr schnell konvergieren. Die Konfigurationen mit Kommaselektion konvergieren erst nach sehr viel mehr Generationen. Dieses Verhalten ist bei allen vier Problemvarianten zu beobachten.

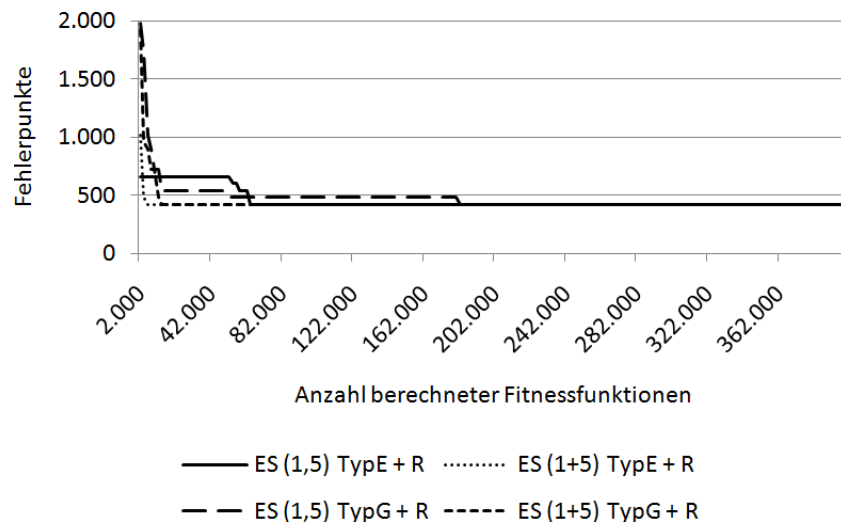


Abbildung 6.9: Verlauf der Fehlerpunkte für ES bei Handel_1AP_Jan

In der Kommaselektion kann hinsichtlich besserer Ergebnisse eine Tendenz zu kleineren Populationen beobachtet werden. Die Mittelwerte der Konfigurationen mit 30 Eltern und 200 Nachkommen sind nie signifikant besser ($\alpha = 0,05$) als die Mittelwerte der Konfigurationen mit einem Elter und fünf Kindern. Der Grund für dieses Verhalten ist wiederum das Abbruchkriterium von 400.000 inspizierten Lösungen. In jeder Generation werden nur wenige Änderungen vorgenommen. Um gute Lösungen zu erzeugen, sind jedoch in Summe sehr viele Änderungen nötig. Mit weniger Individuen können mehr Generationen pro Individuum durchlaufen werden. Es ist für bessere Ergebnisse also hilfreich, mehr Generationen zu durchlaufen und dafür auf eine größere Kenntnis über

den Lösungsraum (durch eine größere Population) in jeder Generation zu verzichten. Bei der Plusselektion ist dieses Verhalten nicht so stark wie bei der Kommaselektion zu beobachten. Zwar wirkt auch hier der positive Effekt möglichst vieler Generationen bei gleichzeitig kleinen Änderungen. Doch bei der Plusselektion besteht das oben beschriebene Problem im Entkommen aus lokalen Optima. Eine große Population hat eine bessere Chance aus einem lokalen Optimum zu entkommen, da sie eine größere Diversität hat und somit eine breitere Suche ermöglicht. Es treten also zwei Effekte auf, die zu den relativ schlechten Ergebnissen der Plusselektion führen. Am stärksten ist die Tendenz zu kleinen Populationen bei beiden Selektionstypen in den Monatsproblemen. Aufgrund der geringeren Problemgröße scheint dort die Gefahr, vorzeitig in einem lokalen Optimum zu konvergieren, weniger ausgeprägt zu sein.

Die besten erzeugten Einsatzpläne von ES mit TypG und TypE wurden eingehend analysiert. Manuelle Verbesserungen konnten nur sehr selten und vor allem nur mit sehr komplexen Änderungen erzielt werden. Zudem sind die Ergebnisse von ES signifikant besser als die der KV, die erfolgreich in kommerziellen Softwarelösungen eingesetzt werden. Aus diesen Gründen können die erzeugten Pläne als praktisch überaus brauchbar angesehen werden.

Bzgl. des Implementierungsaufwands fällt die Reparatur stark ins Gewicht. Diese enthält sehr viel problemspezifisches Wissen und setzt eingehende Analysen erzeugter Einsatzpläne voraus. Die Parametrierung ist bei TypE etwas aufwändiger als bei TypG. In Summe sind je Mutationstyp Implementierung und Parametrierung in maximal fünf Manntagen abgeschlossen.

6.2.7 Particle Swarm Optimization

Die Ergebnisse von PSO zeigen bei den Problemvarianten erhöhte Fehlerpunkte für die Unterdeckung im Personaleinsatz und für die Überschreitung der wöchentlichen Arbeitszeit. Alle Ergebnisse können im Anhang den Tabellen 7.33 bis 7.36 entnommen werden. Ein Durchlauf von PSO dauert bei Handel_2AP etwa sechs Stunden und bei Handel_1AP etwa drei Stunden. Die CPU-Zeiten reduzieren sich bei den Monatsproblemen auf ein Zwölftel. Damit sind PSO und ES etwa gleich schnell. Auch bei PSO ist eine Reparatur erforderlich, wobei die Reparatur von ES zum Einsatz kommt. Ohne Reparatur würden zu viele fehlerhafte Arbeitszeitmodelle erstellt, die nicht immer korrigiert werden können. Die Reparatur sorgt für eine Verdoppelung der CPU-Zeit.

Im Vergleich der Nachbarschaftstopologien sind die Mittelwerte von Wheel, lBest ($k=4$) und Circle bei Handel_1AP, Handel_2AP und Handel_2AP_Jan bei α von 0,05 nie signifikant besser als die von gBest. Dieses Ergebnis deckt sich mit den Erfahrungen, die beim Logistikproblem gemacht wurden. Die Aussage von Kennedy und Mendes [207] sowie von Czogalla und Fink [89], die der gBest-Topologie eine vorzeitige Konvergenz in einem

lokales Optimum zuschreiben, kann auch hier widerlegt werden. Die besondere Eignung der gBest-Topologie liegt in der Funktionsweise des für kombinatorische Problemstellungen modifizierten PSO-Algorithmus. Dieser ist recht gut geeignet aus lokalen Optima zu entkommen, da es durch die zweite Aktion in der Positionsneuberechnung in jeder Iteration zufällige Änderungen in den Partikeln gibt. Der Schwarm bleibt also immer in Bewegung. Dadurch verliert auch die Bildung von kleineren Nachbarschaften zur Verhinderung einer vorzeitigen Konvergenz in einem lokalen Optimum an Gewicht. Vielmehr zeigt sich, dass es lohnender ist, Informationen schneller im Schwarm zu verteilen, da sonst die Gefahr besteht, dass sie verloren gehen.

Eine Ausnahme bei den Nachbarschaftstopologien bildet das einfachste der vier Probleme Handel_1AP_Jan. Dort liefert Circle den signifikant besten Durchschnittswert (bei $\alpha = 0,05$), der sogar das Optimum darstellt. Die gBest-Topologie hingegen schneidet dort im Durchschnitt am schlechtesten ab. Bei Handel_1AP_Jan ist es nicht mehr lohnender, Informationen schnell im Schwarm zu verteilen. Individuen sind hier eher in der Lage, erneut gute Ergebnisse zu produzieren, auch wenn ein Wert verloren gehen sollte. Vielmehr macht es Sinn, den Effekt einer kleineren Nachbarschaftstopologie zu nutzen, um besser aus lokalen Optima zu entkommen. Dieser Zusammenhang ist in Abbildung 6.10 für die besten Durchläufe zu erkennen. Auf die Darstellung der Werte bei den ersten 2.000 berechneten Fitnessfunktionen wurde verzichtet, da andernfalls lediglich ein rechter Winkel wie in Abbildung 4.2 zu erkennen wäre. Deutlich sieht man, dass gBest zunächst sehr gut abschneidet, da Informationen schnell im Schwarm verteilt werden. Das Entkommen aus lokalen Optima ist bei gBest jedoch schwerer als in der Circle-Topologie, sodass gBest erst recht spät das Optimum von 420 Fehlerpunkten trifft. Circle hingegen nähert sich zunächst langsamer an die 420 Punkte an, findet den Wert insgesamt jedoch schneller als gBest.

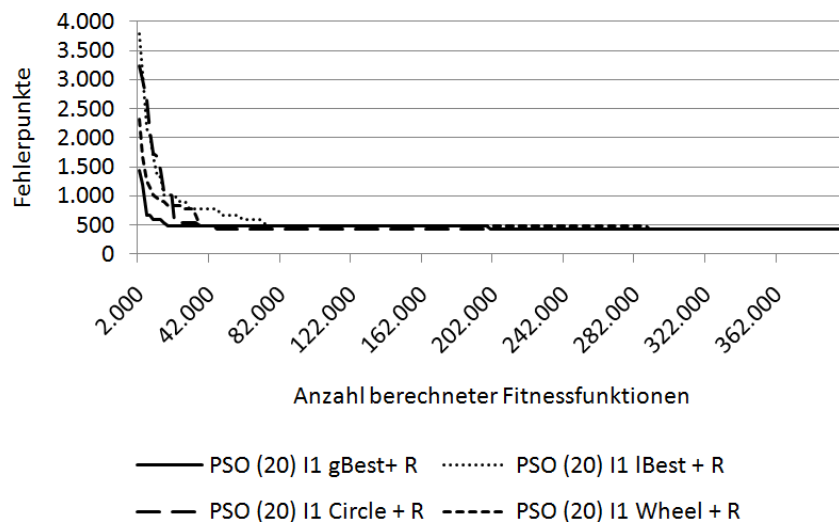


Abbildung 6.10: Verlauf der Fehlerpunkte verschiedener Nachbarschaftstopologien für PSO (20) bei Handel_1AP_Jan

Bzgl. einer Lösungsverbesserung ist eine Tendenz hin zu kleineren Schwarmgrößen zu erkennen. Der beste Einsatzplan und der niedrigste durchschnittliche Fehlerwert über 30 Replikationen werden bei einer gBest-Topologie nie mit 200 Partikeln erzeugt. Meist sind 20 Partikel vorteilhaft – bei Handel_1AP_Jan sind es 40 Partikel am besten. Mit einem t-Test, bei dem die Mittelwerte von PSO (20) I1 gBest + R und PSO (200) I1 gBest + R verglichen werden, ergibt sich bei homogenen Varianzen ein p -Wert kleiner als 0,001 ($T = -4,686$; $\alpha = 0,05$). Der Grund für dieses Verhalten ist wie bei ES (siehe Kapitel 6.2.6) das Abbruchkriterium von 400.000 inspizierten Lösungen. In jeder Iteration werden nur wenige Änderungen vorgenommen, da sonst die Wahrscheinlichkeit zu groß ist, Verschlechterungen hervorzurufen. Um ausgehend von der Startlösung gute Ergebnisse zu erzeugen, sind in Summe jedoch sehr viele Änderungen nötig. Mit weniger Partikeln können mehr Iterationen durchlaufen werden, was für bessere Ergebnisse sorgt. Auch hier sind also mehr Iterationen der größeren Kenntnis über den Lösungsraum in jeder Iteration vorzuziehen.

Wie bei ES ist auch der Implementierungsaufwand für PSO bei den Handelsproblemen stark durch die Reparatur geprägt, die recht anspruchsvoll ist. Die Parametrierung erweist sich ein wenig einfacher als bei ES, da lediglich die richtige Eintrittswahrscheinlichkeit für Aktion zwei gefunden werden muss. Die anderen drei Parameter zur Positionsberechnung sind relativ unempfindlich gegenüber Änderungen. In Summe werden bei PSO fünf Manntag als Aufwand veranschlagt.

6.2.8 Multiagentensysteme

Im Vergleich zum Logistikproblem müssen beim Handelsproblem mehr Restriktionen berücksichtigt werden. Außerdem umfasst es deutlich mehr Dimensionen. Es zeigt sich, dass der Agentenansatz bei einem nun sehr viel komplexeren Problem weiter von Bestwerten anderer Verfahren entfernt ist, als dies beim Logistikproblem der Fall war. Die Ergebnisse sind in den Tabellen 7.33 bis 7.36 im Anhang zu finden.

Der Agentenansatz hat bei allen vier Problemen erhebliche Schwierigkeiten, die wöchentliche Sollarbeitszeit der Mitarbeiter nicht zu überschreiten. Wie bereits beim Logistikproblem beschrieben, wäre dies nur über eine zentrale Planungsinstanz zu erreichen, die bei vielen Mitarbeitern Umplanungen in weiten Teilen des Planungszeitraums vornimmt. Änderungen in der wöchentlichen Arbeitszeit sind von einzelnen Mitarbeitern nur eingeschränkt durch Tausch und Umbesetzung zu variieren. Dieses Problem ist ebenfalls bei der Unterdeckung im Personaleinsatz festzustellen. Positiv ist jedoch, dass in keinem der vier Problemvarianten Fehler bei den erstellten Arbeitszeitmodellen und bei Überdeckungen im Personaleinsatz auftreten, was auf erfolgreiche Verhandlungen der Mitarbeiter zurückzuführen ist.

Mit durchschnittlich 0,6 Sekunden bei Handel_2AP, 0,3 Sekunden bei Handel_1AP und

0,1 Sekunden bei den beiden Monatsproblemen ist der Agentenansatz mit Abstand das schnellste getestete Lösungsverfahren für die Handelsprobleme. Die CPU-Zeiten sind trotz des meist größeren Lösungsraums geringer als beim Logistikproblem, da hier weniger Arbeitsplätze berücksichtigt werden müssen, was Zuweisungen und Verhandlungen reduziert.

Der Implementierungsaufwand kann analog zum MAS beim Logistikproblem mit maximal fünf Manntagen beziffert werden. Parameter werden nicht eingestellt.

6.2.9 Vergleich der Lösungsverfahren

Die Forschungsfrage nach der geeigneten Parametrierung von Lösungsverfahren wurde in den vorangegangenen Kapiteln ausführlich an den vier Varianten des Handelsproblems erörtert. Nachfolgend wird zunächst auf die zentrale Forschungsfrage dieser Arbeit eingegangen, woraufhin weitere Fragen beantwortet werden.

Analog zum Logistikproblem konnte auch beim Handelsproblem gezeigt werden, dass klassische OR-Verfahren ungeeignet sind. Entweder sprengt die benötigte Rechenzeit alle praktikablen Grenzen oder die Problemstellungen müssen derart vereinfacht werden, dass die Ergebnisse unbrauchbar sind.

Die Ergebnisse diverser Metaheuristiken wurden in den Kapiteln 6.2.4 bis 6.2.7 erläutert. Als Benchmark konnten zwei KV herangezogen werden, die Bestandteil kommerzieller Lösungen sind. Obwohl die KV von zahlreichen Unternehmen für ähnliche Problemstellungen eingesetzt werden, sind die Ergebnisse sehr schlecht. Bei fast allen Problemen schneidet ES mit einer sehr kleinen Population am besten ab, wobei die Mutation TypG der Mutation TypE signifikant überlegen ist. Nur an der kleinsten Problemstellung ist PSO signifikant besser. Dort ist PSO sogar in der Lage, im Durchschnitt das Optimum zu treffen. Die entsprechenden t-Tests können den Tabellen 6.9 bis 6.12 entnommen werden. Am Problem Handel_1AP_Jan ist kein t-Test zwischen den Mittelwerten von PSO (20) I1 Circle + R und KV 1 möglich, da die Standardabweichungen bei beiden Gruppen null sind. Wie zuvor erläutert, hat KV 1 jedoch den schlechteren Mittelwert. Beim Einsatz von ES und PSO ist immer eine Reparatur erforderlich, die besonders für korrekte Arbeitszeitmodelle sorgt. Ohne Reparatur wären ES und PSO nicht anwendbar. Bei allen vier Problemvarianten ist HC das schlechteste Verfahren. Im Gegensatz zum Logistikproblem, wo HC überaus gut ist, gibt es nun Probleme mit der Einhaltung sehr komplexer Nebenbedingungen. Dies ist jedoch nicht verwunderlich, wenn man das recht simple Vorgehen von HC berücksichtigt. Während am deutlich einfacheren Logistikproblem (weniger Dimensionen sowie weniger und einfachere Nebenbedingungen) punktbasierte Lösungsverfahren überaus gut abschnitten, sind hier die populationsbasierten Verfahren überlegen.

Als dritte Gruppe, neben klassischen OR-Verfahren und Metaheuristiken, werden MAS im Rahmen der zentralen Forschungsfrage beurteilt. Zwar sind MAS auch hier extrem schnell. Sie können jedoch nur die Ergebnisse der KV und von HC schlagen. Von den Bestwerten, die mit ES oder PSO erzielt werden, sind sie weit entfernt. Selbst wenn die CPU-Zeit ein begrenzender Faktor sein sollte, erweisen sich MAS als weniger geeignet. Ggf. ist die Erzeugung grober Planungsvorschläge denkbar, die dann vom Planer nachbearbeitet werden. Auch hier sind die entsprechenden t-Tests für alle vier Problemvarianten in den Tabellen 6.9 bis 6.12 aufgeführt.

Tabelle 6.9: t-Tests (einseitig) bei Handel_1AP

H_1	T	df	Signifikanz H_0	95% Konfidenzintervall der Differenz	
				untere	obere
ES (1,5) I1 TypG + R < PSO (20) I1 gBest + R	-10,675	31,476	< 0,001	-4.746,735	-3.446,065
ES (1,5) I1 TypG + R < HC I1	-60,264	29,001	< 0,001	-1.512.093,024	-1.429.164,576
ES (1,5) I1 TypG + R < KV 2	-268,421	29,000	< 0,001	-20.992,045	-20.727,955
ES (1,5) I1 TypG + R < MAS	-107,467	58	< 0,001	-11.967,289	-11.600,711

Tabelle 6.10: t-Tests (einseitig) bei Handel_1AP_Jan

H_1	T	df	Signifikanz H_0	95% Konfidenzintervall der Differenz	
				untere	obere
PSO (20) I1 Circle + R < ES (1,5) I1 TypG + R	-2,408	29,000	0,012	-17,055	-2,945
PSO (20) I1 Circle + R < HC I1	-17,680	29,000	< 0,001	-98.745,275	-81.429,925
PSO (20) I1 Circle + R < KV 1	-	-	-	-	-
PSO (20) I1 Circle + R < MAS	-11,958	29,000	< 0,001	-486,531	-365,469

Tabelle 6.11: t-Tests (einseitig) bei Handel_2AP

H_1	T	df	Signifikanz H_0	95% Konfidenzintervall der Differenz	
				untere	obere
ES (1,5) I1 TypG + R < PSO (20) I1 gBest + R	-13,305	29,666	< 0,001	-32.532,388	-25.169,212
ES (1,5) I1 TypG + R < HC I1	-121,492	29,003	< 0,001	-4.317.210,211	-4.198.119,255
ES (1,5) I1 TypG + R < KV 2	-330,701	29,000	< 0,001	-76.815,524	-76.030,210
ES (1,5) I1 TypG + R < MAS	-114,742	58	< 0,001	-38.627,067	-37.517,799

Die Abbildungen 6.11 bis 6.14 zeigen jeweils die Mittelwerte relevanter Lösungsverfahren für alle vier Varianten des Logistikproblems. Die senkrechten Linien definieren den Bereich, in dem sich die Minimal- und Maximalwerte der 30 Replikationen befinden. Die waagerechten Striche zeigen den Mittelwert der Stichproben. Auf die Darstellung von HC I1 musste verzichtet werden. HC I1 schneidet derart schlecht ab, dass die Skalierung des Diagramms unvorteilhaft geändert worden wäre, was kaum Details erkennen ließe. Bei den kleineren Monatsproblemen fällt deutlich die geringe Streuung der Ergebnisse für die PSO-Konfigurationen und ES (1,5) I1 + R auf. Dies ist bei den Jahresproblemen lediglich für einige wenige ES-Konfigurationen feststellbar, wo sie sehr zielsicher Bestwerte

Tabelle 6.12: t-Tests (einseitig) bei Handel_1AP_Jan

H_1	T	df	Signifikanz H_0	95% Konfidenzintervall der Differenz	
				untere	obere
ES (1,5) I1 TypG + R < PSO (20) I1 gBest + R	-8,250	35,570	< 0,001	-609,778	-402,556
ES (1,5) I1 TypG + R < HC I1	-7,942	29,003	< 0,001	-26.619,862	-17.237,471
ES (1,5) I1 TypG + R < KV 2	-473,553	29,000	< 0,001	-9.355,448	-9.288,552
ES (1,5) I1 TypG + R < MAS	-34,242	32,239	< 0,001	-3.071,200	-2.781,733

finden. KV 1 und KV 2 liefern auch bei 30 Replikationen immer die gleichen Ergebnisse. Bei KV 1 haben die stochastischen Elemente keinen Einfluss auf die Ergebnisse und der Ablauf von KV 2 ist deterministisch. Außer bei HC I1, KV 1 und 2, die allgemein recht schlecht abschneiden, sind die Ergebnisse aller anderen Verfahren umso dichter beieinander, je einfacher das zu lösende Problem wird. Beginnend bei Handel_2AP, Handel_1AP über Handel_2AP_Jan bis hin zu Handel_1AP_Jan. Die unterschiedlichen Suchoperatoren und Lösungskonzepte der Verfahren gewinnen demnach mit zunehmender Komplexität der Problemstellungen an Bedeutung, da es immer schwerer wird, aus lokalen Optima zu entkommen und die Suche bestmöglich zu steuern.

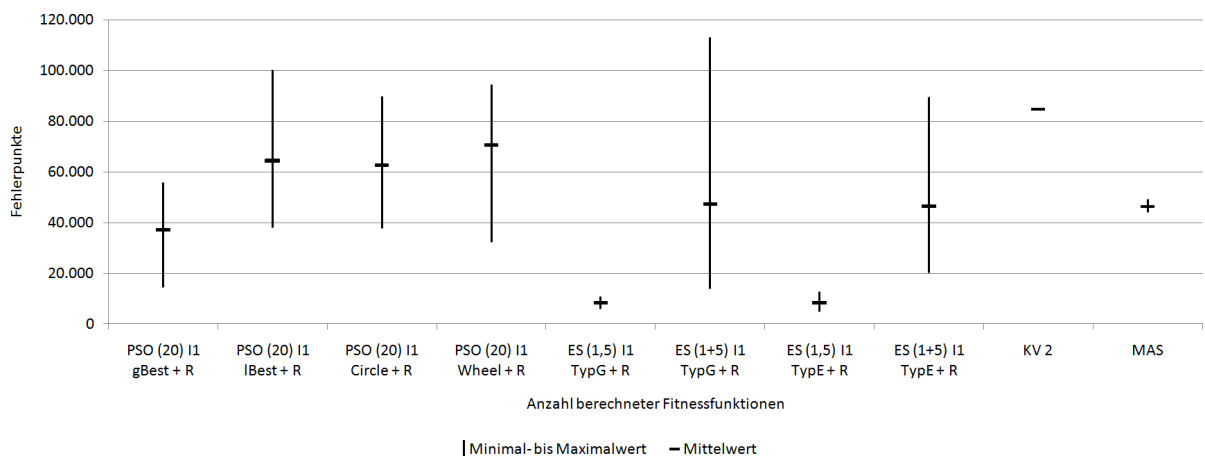


Abbildung 6.11: Vergleich verschiedener Lösungsverfahren bei Handel_2AP

Hinsichtlich der zu erwartenden CPU-Zeiten bei der vollständigen Enumeration und bei klassischen OR-Verfahren, wurde gezeigt, dass diese nicht praktikabel sind. Wie beim Logistikproblem benötigen auch hier ES und PSO etwa gleich viel Rechenzeit. Sie liegt bei bis zu sechs Stunden beim größten Problem. HC liefert bei Handel_2AP nach etwa zweieinhalb Stunden ein Ergebnis. HC ist schneller als ES und PSO, da er keine Reparatur enthält und ein Move recht einfach berechnet wird. Bei Handel_2AP benötigt KV 2 etwa zehn Minuten. Für KV 1 liegen keine Rechenzeiten für das größte Planungsproblem mit zwei Arbeitsplätzen vor, da KV 1 nur mit einem Arbeitsplatz zurechtkommt. Würde man mit KV 1 zunächst den ersten Arbeitsplatz von Handel_2AP und anschlie-

6.2 Automatische Arbeitszeitmodellerstellung im Handel

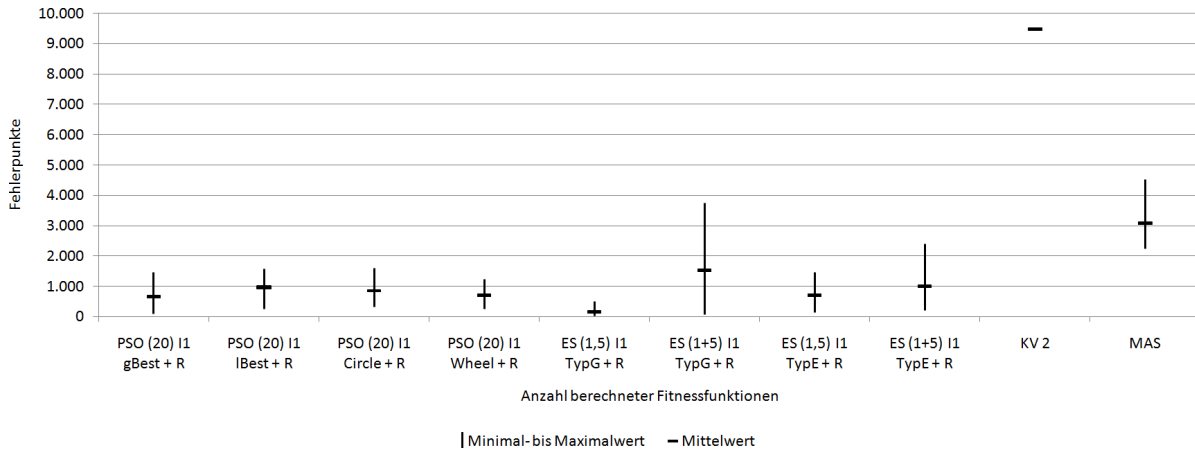


Abbildung 6.12: Vergleich verschiedener Lösungsverfahren bei Handel_2AP_Jan

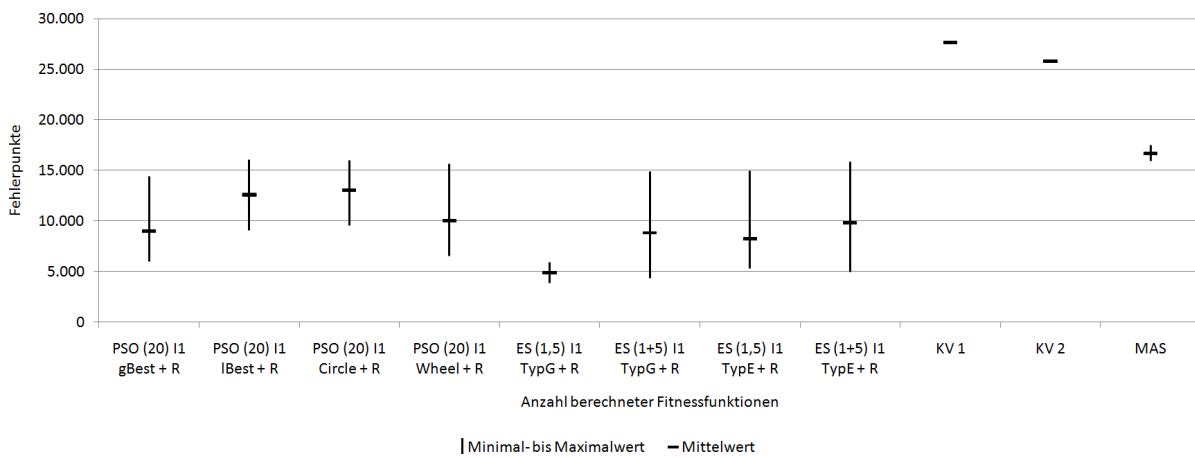


Abbildung 6.13: Vergleich verschiedener Lösungsverfahren bei Handel_1AP

ßend den zweiten Arbeitsplatz berechnen lassen, so würde KV 1 nach etwa fünf Minuten das Ergebnis liefern (Mitarbeitern ist je ein Arbeitsplatz in ihrem Personalstammsatz zugeordnet, nach denen die Gruppierung erfolgen würde. Im finalen Einsatzplan wären selbstverständlich keine Arbeitsplatzwechsel zu finden.). Das schnellste Verfahren ist auch hier das MAS mit weniger als eine Sekunde für die größte Problemstellung.

Der Aufwand für die Implementierung und Parametrierung ist bei allen Verfahren – außer bei HC, welches sehr schnell umzusetzen ist – etwa gleich groß. Im Vergleich zum Logistikproblem haben sich die Aufwände aufgrund der komplexeren Nebenbedingungen jedoch erhöht. Gerade beim Agentenansatz und bei der Reparatur für PSO und ES.

Bzgl. der zentralen Forschungsfrage ist für die Varianten des Handelsproblems festzustellen, dass Metaheuristiken am geeignetsten sind. Klassische OR-Verfahren sind unbrauchbar und MAS sollten nur dann zum Einsatz kommen, wenn schnell ein Planungsvorschlag

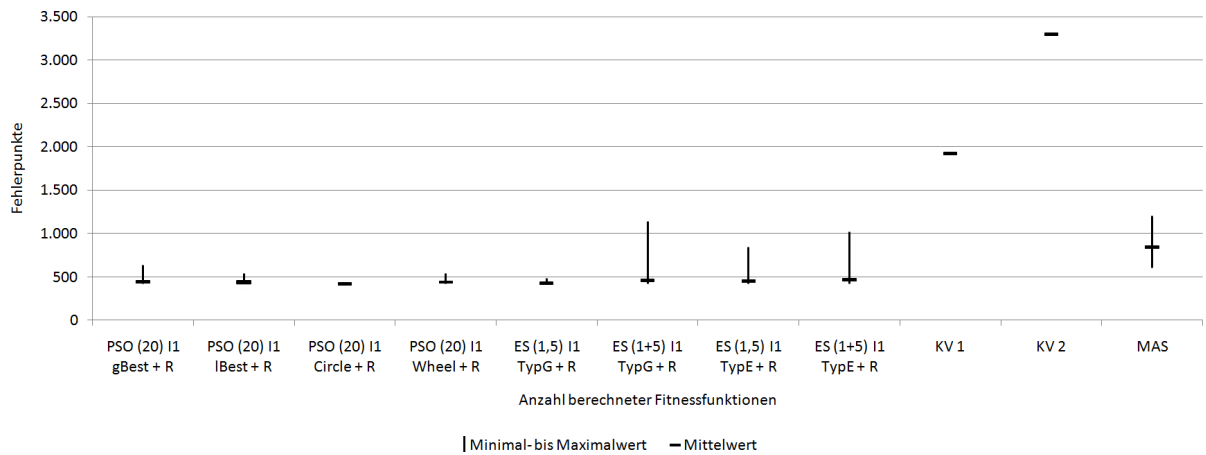


Abbildung 6.14: Vergleich verschiedener Lösungsverfahren bei Handel_1AP_Jan

erzeugt werden soll, der anschließend vom Planer manuell korrigiert wird.

Analog zum Logistikproblem kann auch beim Handelsproblem die Forschungsfrage beantwortet werden, ob die gemachte Aussage von Puppe u.a. [315] auch hier zutreffend ist. Puppe u.a. haben gesagt, dass sich zentrale Lösungsverfahren weniger gut für die Dienstplanung eignen als MAS (dezentral). Beim Handelsproblem ist deutlich geworden, dass Metaheuristiken als zentraler Ansatz den dezentralen MAS überlegen sind. Die Aussage von Puppe u.a. gilt für das Handelsproblem demnach nicht.

Zwar sind die Bestwerte bei den Jahresproblemen noch recht weit vom Lower Bound entfernt. Aber dort lassen sich manuelle Verbesserungen der Einsatzpläne in Ausnahmen nur mit äußerst hohem Aufwand finden. Bei den Monatsproblemen ist der Bestwert einmal maximal zehn Fehlerpunkte vom Optimum entfernt (das Optimum muss dort zwischen null und zehn Punkten liegen) und wird einmal exakt getroffen. Die Qualität der besten Pläne kann daher als überaus gut bezeichnet werden. Gerade im Vergleich zu den Ergebnissen kommerzieller Lösungen, die am Markt etabliert sind. Die Forschungsfrage, ob sich die getesteten Lösungsverfahren auch für Problemstellungen eignen, die direkt aus der Praxis stammen, kann in Abhängigkeit vom gewählten Lösungsverfahren für alle Varianten des Handelsproblems bejaht werden. Hier ist es vor allem ES – bei einem sehr kleinen Problem auch PSO.

Die letzte Forschungsfrage, die an dieser Stelle beantwortet wird, bezieht sich auf die betriebswirtschaftlichen Effekte, die mit hochflexiblem WFM realisiert werden können. Zum Vergleich werden die KV der kommerziellen Lösungen herangezogen, da sie bei vielen Einzelhändlern Deutschlands im Einsatz sind. Im Vergleich von KV 1 mit dem jeweils besten eigenen Lösungsverfahren ergeben sich im Durchschnitt über die beiden Problemvarianten mit einem Arbeitsplatz die folgenden Effekte:

- Reduktion der Überdeckung im Personaleinsatz (bezahlte – aber unproduktive

Leerzeit) um 100%.

- Reduktion der Unterdeckung im Personaleinsatz (wird meist durch Überstunden aufgefangen) um 50%.
- Reduktion der Überschreitung der wöchentlichen Sollarbeitszeit um 88%.

Im entsprechenden Vergleich von KV2 mit dem jeweils besten eigenen Verfahren ergeben sich im Durchschnitt über alle vier Problemvarianten die nachfolgend aufgeführten Effekte:

- Reduktion der Überdeckung im Personaleinsatz (bezahlte – aber unproduktive Leerzeit) um 100%.
- Reduktion der Unterdeckung im Personaleinsatz (wird meist durch Überstunden aufgefangen) um 65%.
- Reduktion der Überschreitung der wöchentlichen Sollarbeitszeit um 90%.

Es wird eindrucksvoll belegt, dass die betriebswirtschaftlichen Potenziale enorm sind. Hinzu kommen die Effekte, die mit der automatischen Arbeitszeitmodellerstellung direkt in der Planung erzielt werden. Leider liegen keine Vergleichswerte von einem Ein- oder Zweischichtbetrieb der Abteilung vor. Es ist jedoch einleuchtend, dass die automatische Erstellung von Arbeitszeitmodellen im Vergleich zu wenigen starren Schichten Vorteile hinsichtlich des passgenauen Personaleinsatzes hat (siehe Kapitel 3.3.1 sowie [162], [150]).

6.3 Kombinierte Personaleinsatzplanung und Tourenplanung bei Technikern

In diesem Kapitel werden die Ergebnisse verschiedener Lösungsverfahren vorgestellt. Die Verfahren SA, GA, CLP sowie HC, FLS und Fast GLS stammen von anderen Autoren. Von diesen Verfahren sind jeweils nur die Minimalwerte bekannt, was die Vergleichbarkeit sehr erschwert. Ein Hinweis auf die Durchschnittswerte, Abbruchkriterien sowie die Anzahl an Replikationen und Einzelwerte war nicht zu finden. Für GA und SA sind zudem keine CPU-Zeiten angegeben.

Dennoch liefern die Ergebnisse dieser Verfahren wertvolle Erkenntnisse, da die Minimalwerte als Benchmark angesehen werden können. Eine Vielzahl an Autoren hat sich z.T. in Dissertationen und geförderten Forschungsprojekten intensiv mit dem British

Telecom-Problem und möglichen Optimierungsalgorithmen beschäftigt. Zudem wurden die Erkenntnisse bei einschlägigen Magazinen und Konferenzen mit Begutachtungsprozess veröffentlicht.

Alle eigenen Lösungsverfahren werden am Problem BT_Mod-250-118 erprobt, da dort die bisher umfangreichsten Experimente von anderen Autoren vorgenommen wurden. Die Ergebnisse aller Varianten des British Telecom-Problems sind im Anhang in den Tabellen 7.37 bis 7.40 zu finden.

Im folgenden Kapitel wird zunächst auf den Lower Bound eingegangen. Anschließend werden die Ergebnisse diverser Lösungsverfahren in den Kapiteln 6.3.4 bis 6.3.11 erörtert. Mit einer Zusammenfassung der Erkenntnisse und der Beantwortung zahlreicher Forschungsfragen endet dieses Unterkapitel.

6.3.1 Lower Bound

Die Ermittlung einer Optimallösung sowie die exakte Berechnung oder Herleitung des Lower Bounds ist aufgrund der Komplexität des Problems leider nicht möglich. Dies ist auch nicht der Vielzahl an Autoren gelungen, die sich seit nunmehr 18 Jahren mit der Problemstellung der British Telecom befassen.

Das bisher beste Ergebnis vor dieser Arbeit wurde von Tsang und Voudouris 1997 [383] mit Fast GLS am Problem BT_Mod-250-118 (20.433 Fehlerpunkte) gefunden. Da bei Tsang und Voudouris die in Kapitel 4.1.1 geschilderten Probleme mit der Abbildung der Problemstellung auftreten, wird die Optimallösung höchstwahrscheinlich unter diesem bisherigen Bestwert liegen. Ein sehr viel besseres Ergebnis ist aufgrund der Vielzahl an sehr intensiven Arbeiten an diesem Problem jedoch nicht mehr zu erwarten.

6.3.2 Ganzzahlige Optimierung

Keinem der zahlreichen Autoren, die das British Telecom-Problem seit 18 Jahren bearbeiten, ist es gelungen, dieses Problem mit Hilfe der Ganzzahligen Optimierung zu lösen. Dies ist nicht verwunderlich, da das Problem NP-schwer ist.

In Kapitel 6.1.2 wurde umfassend dargestellt, dass sich MIP und MINLP nicht zur Lösung des Logistikproblems eignen. Dies wurde in Kapitel 6.2.2 ebenfalls für das Handelsproblem nachgewiesen. Ursache hierfür sind Nichtlinearitäten in der Zielfunktion und die Größe der Lösungsräume. Allein die Berücksichtigung von Überstunden führt beim British Telecom-Problem zu einem quadratischen Element in der Zielfunktion, was

zur Nichtlinearität der Zielfunktion führt. Auch die Bestimmung, ob ein Auftrag erfüllt oder unerfüllt ist, ist nur in nichtlinearer Form möglich. Zudem ist die Anzahl an binären Entscheidungsvariablen ungleich null bei 118 Mitarbeitern und 250 Aufträgen nicht unerheblich. Daher wurde auf eigene Experimente mit GAMS verzichtet.

Wie beim Logistik- und Handelsproblem kann auch hier keine Aussage zum Implementierungsaufwand gemacht werden, da die GAMS-Solver nicht selbst entwickelt wurden und in der Literatur keine Aussagen hierzu getätigt werden.

6.3.3 Konstruierende Verfahren

Für das Problem BT_Mod-250-118 wurde KV 3 erstellt, dessen Ergebnisse im Anhang in Tabelle 7.37 zu finden sind. Dieses Verfahren ist sehr schnell und liefert ein Ergebnis in weniger als einer Sekunde, da die Fitnessfunktion nur einmal am Ende zur Ermittlung der Fehlerpunkte berechnet wird.

Mit durchschnittlich 44.007,7 Fehlerpunkten ist KV 3 das schlechteste aller getesteten Lösungsverfahren. Zwar erreichen die Fahrtkosten (2.521,3 Fehlerpunkte) einen absoluten Minimalwert. Dies liegt jedoch daran, dass durchschnittlich nur 76 Aufträge zugewiesen werden. Als Folge ergeben sich sehr hohe Fehlerpunkte für nicht zugewiesene Aufträge (41.486,4). Das Konzept der regionalen Zuteilung von Aufträgen zu Technikern unter Berücksichtigung ihrer Kapazität mag zwar plausibel klingen, ist in der jetzigen Form jedoch nicht für das Problem der British-Telecom geeignet. Das KV müsste deutlich mehr problemspezifisches Wissen enthalten, um Aufträge besser verplanen zu können. In diesem Punkt stößt man mit KV jedoch an die Grenzen des mit akzeptablem Aufwand Machbaren.

KV 3 enthält bereits recht viel problemspezifisches Wissen, der Quellcode ist jedoch nicht sonderlich umfangreich, sodass die Implementierung innerhalb von einem Manntag abgeschlossen ist. Parameter sind nicht einzustellen.

6.3.4 Constraint Logic Programming

Nachfolgend wird auf die beiden CLP-Ansätze eingegangen. Zunächst auf den Ansatz von Azarmi und Abdul-Hameed [18], anschließend auf den Ansatz von Yang [428], der in Kapitel 6.3.11 von einem MAS genutzt wird. Die entsprechenden Ergebnisse befinden sich im Anhang in den Tabellen 7.37, 7.38 und 7.40.

Ansatz von Azarmi und Abdul-Hameed Azarmi und Abdul-Hameed vergleichen CLP TG + R und CLP CG + R mit SA [22] sowie GA und GA + R [277] am Problem BT_Mod-250-118 [18]. Ihre CLP-Varianten liefern zwar ein besseres Ergebnis als GA und GA + R [18]. Sie können SA jedoch nicht schlagen. Bei beiden CLP-Varianten konnten mehr Aufträge zugewiesen werden als bei SA sowie GA und GA + R, da die Reparatur der CLP-Varianten einen Fokus auf die Reduzierung nicht ausgeführter Aufträge hat. Bei GA + R hingegen liegt der Fokus der Reparatur auf der Reduzierung der Gesamtfehlerpunkte. SA verfügt über keine Reparatur und hat dennoch die geringsten Gesamtfehlerpunkte.

Am Problem BT_Mod-250-118 ist CLP CG + R (21.292) besser als CLP TG + R (22.241). Beide CLP-Varianten benötigen jeweils 600 Sekunden. Der Hauptunterschied der Varianten besteht darin, dass bei CLP CG + R die Tour eines Technikers immer in eine optimale Reihenfolge gebracht wird, sobald ein neuer Auftrag hinzu kommt. Dies führt dazu, dass CLP CG + R 4.902 und CLP TG + R 5.269 Fehlerpunkte bei den Fahrtkosten hat.

An einer weiteren Problemstellung der British Telecom mit 434 Aufträgen und 120 Technikern führen Azarmi und Abdul-Hameed ebenfalls Experimente durch. Diese Problemvariante wird von Azarmi und Abdul-Hameed nur kurz erwähnt und in dieser Arbeit daher nicht explizit vorgestellt. Zudem wenden sie CLP TG + R und CPT CG + R an diversen Ausprägungen eines Vehicle Routing Problems (VRP) der Universität Strathclyde an. Sie konnten die Erkenntnisse für ihre CLP-Varianten bei BT_Mod-250-118 jeweils bestätigen.

Da Workforce Scheduling ein Problem mit wenigen harten Nebenbedingungen ist, sind nach Azarmi und Abdul-Hameed stochastische Lösungsverfahren wie SA und GA besser geeignet. Bei einem deutlich beschränkteren Problem dürfte CLP jedoch zu bevorzugen sein. Ein tieferes Verständnis für die Problemstruktur und den daraus resultierenden Anpassungen von CLP sollte dessen Anwendung ebenfalls begünstigen.

Der Implementierungs- und Parametrierungsaufwand wird von Azarmi und Abdul-Hameed nicht beschrieben. Sie sagen lediglich, dass die Entwicklungszeit mit CLP relativ gering ist.

Ansatz von Yang Yang vergleicht seine CLP-Ansätze [428] mit SA [22], GA, GA + R [277], CLP TG + R, CLP CG + R [18] sowie Fast GLS [383] am Problem BT_Mod-250-118. Er konnte mit einigen CLP-Ansätzen alle für den Vergleich herangezogenen Verfahren außer Fast GLS schlagen. Einige seiner CLP-Ansätze sind demnach besser als die von Azarmi und Abdul-Hameed.

Es erprobt alle drei Heuristiken Closest First (CF), Same Direction First (SD) und Least

Busy First (LB) nicht nur am Problem BT_Mod-250-118, sondern auch an BT_Rel-250-118 und BT_Bristol-250-118. SD schnitt bei allen drei Problemen am besten ab – gefolgt von CF. LB liefert erwartungsgemäß die schlechtesten Fehlerpunkte. Dort liegt der Fokus auf der Reduzierung untätiger Techniker, was jedoch nicht in die Zielfunktion einfließt. Dafür ist dort jedoch nur einer der 118 Techniker untätig. Bei Experimenten an allen drei Heuristiken mit und ohne Forward Checking (FC) war die Anwendung von FC immer lohnend. Am Rel-250-118-Problem und am Bristol-Mod-250-118-Problem konnten die Fehlerpunkte damit um etwa 40% reduziert werden. Bei BT_Mod-250-118 jedoch nur um 4–5%. Dies liegt daran, dass das Problem BT_Mod-250-118 weniger beschränkt ist als die anderen beiden Problemstellungen, wodurch CLP ohne FC bereits recht gute Ergebnisse liefert.

Die CPU-Zeiten liegen bei allen drei getesteten Problemen je nach verwendeter Heuristik zwischen 9 und 100 Sekunden, was recht schnell ist.

Yang nutzt kein Reparaturverfahren. Es erfolgt demnach auch keine Neuordnung der Tour wie bei den CLP-Ansätzen von Azarmi und Abdul-Hameed, wenn einem Techniker ein neuer Auftrag zugewiesen wurde. Yang geht davon aus, dass eine Neuordnung den CLP-Algorithmus weiter verbessern dürfte. CLP TG + R und CLP CG + R sind sehr ähnlich zu CLP von Yang. Sie verfügen jedoch über kein Forward Checking.

Aussagen bzgl. des Implementierungs- und Parametrierungsaufwands werden von Yang nicht getätigt.

6.3.5 Hill Climbing

1997 wenden Tsang und Voudouris [383] HC auf die Probleme BT_Mod-250-118 und BT_Tsang-250-118 an. Obwohl sich der Ablauf von HC recht einfach gestaltet, sind die Ergebnisse erstaunlich gut. HC liefert bei BT_Mod-250-118 ein besseres Ergebnis (20.788) als in früheren Arbeiten mit SA, CLP und GA erzielt werden konnte. HC kann jedoch nicht den Bestwert von Fast GLS schlagen, der 1997 von Tsang und Voudouris in der gleichen Arbeit veröffentlicht wurde. Die entsprechenden Ergebnisse können im Anhang in den Tabellen 7.37 und 7.39 nachgeschlagen werden.

Für das Problem BT_Mod-250-118 benötigt HC fünfeinhalb Stunden und für BT_Tsang-250-118 fast sieben Stunden. Dabei werden alle Berechnungen Ende der 90er in C++ mit einem DEC Alpha durchgeführt. Die Rechenzeiten ist auf heutigen Rechnern deutlich kleiner.

HC von Tsang und Voudouris hat gegenüber den im Rahmen dieser Arbeit selbst entwickelten Lösungsverfahren wie PSO und ES den Nachteil, dass eine Permutation im Rahmen der Fitnessbewertung transformiert werden muss (siehe Kapitel 4.1.1). Durch

die Art der Transformation sind Bereiche des Lösungsraums nicht erreichbar. Der Vorteil ist jedoch, dass man sich bei HC keine Gedanken über die Einhaltung von Zeitfenstern der Aufträge machen muss. Dies wird bei der Transformation berücksichtigt. Alle selbst implementierten Lösungsverfahren verwenden eine andere Variante zur Abbildung der Problemstellung. Diese erfordert zwar aufwändige Plausibilitätsprüfungen, um falsche Zeitfenster zu vermeiden. Z.B. bei der Neubestimmung der Partikelposition sowie bei der Mutation und Rekombination. Dafür sind aber alle Positionen im Lösungsraum erreichbar.

Hinsichtlich des Implementierungsaufwands machen Tsang und Voudouris keine Aussagen. Parameter sind nicht einzustellen.

6.3.6 Fast Local Search und Guided Local Search

Tsang und Voudouris [383] führen an den Problemen BT_Mod-250-118 und BT_Tsang-250-118 Experimente mit FLS und Fast GLS durch. Mit Fast GLS haben sie den damaligen Bestwert für BT_Mod-250-118 gefunden (20.433), der in dieser Arbeit jedoch von PSO unterboten wird. FLS (20.732) hat etwas weniger Fehler für gefahrene Kilometer. Dafür sind bei Fast GLS, eine Kombination aus FLS und GLS, die Fehlerpunkte für nicht erledigte Aufträge geringer. Analog zu HC sind auch diese Ergebnisse im Anhang in den Tabellen 7.37 und 7.39 zu finden.

FLS ist mit einer CPU-Zeit von 1.242 Sekunden und ähnlichen Ergebnissen wie HC bei BT_Mod-250-118 um den Faktor 16 schneller und bei BT_Tsang-250-118 um den Faktor 20 schneller als HC. Für diesen Geschwindigkeitsvorteil muss jedoch in Kauf genommen werden, dass Lösungen unberücksichtigt bleiben, die u.U. vielversprechend sein können. Auch tendiert FLS analog zu anderen HC dazu, in einem lokalen Optimum festzusitzen. Die Ergebnisverschlechterungen im Vergleich zu HC treten jedoch nicht im erwarteten Umfang ein. Bei BT_Mod-250-118 ist FLS sogar um 0,3% besser als HC und bei BT_Tsang-250-118 lediglich um 0,5% schlechter. Die Berechnungen wurden wie bei HC 1997 in C++ mit einem DEC Alpha durchgeführt. Die CPU-Zeiten werden auf einem modernen Rechner deutlich geringer sein.

Mit Fast GLS werden der FLS- und der GLS-Ansatz kombiniert. Hierdurch werden die bisherigen Ergebnisse von FLS verbessert. Bei BT_Mod-250-118 um 1,4% und bei BT_Tsang-250-118 um 1,1%. Gegenüber HC können die Ergebnisse mit Fast GLS bei BT_Mod-250-118 um 1,7% und bei BT_Tsang-250-118 um 0,6% verbessert werden. Ergebnisse für GLS sind nicht angegeben. Es zeigt sich also, dass das zu Grunde liegende lokale Suchverfahren nun erfolgreicher aus lokalen Optima entkommen kann. Auch die Berechnungen zu Fast GLS wurden in C++ mit einem DEC Alpha durchgeführt. Fast GLS benötigt bei BT_Mod-250-118 9.183 Sekunden und bei BT_Tsang-250-118 4.102 Sekunden. Es ist aufgrund zusätzlicher Berechnungen, die der GLS-Ansatz erfordert,

langsamer als FLS.

Bei den Ausführungen zu HC von Tsang und Voudouris wurde bereits geschildert, dass dort die Art, wie die Problemstellung abgebildet ist, Nachteile hat. Auch bei FLS und Fast GLS wählen Tsang und Voudouris dieses Vorgehen, wodurch sie Bereiche des Lösungsraums unerreichbar machen, die womöglich zu besseren Ergebnissen führen. Trotz dieses Umstandes konnten sie mit Fast GLS 1997 einen neuen Bestwert liefern.

Über den Aufwand für Implementierung und Parametereinstellung machen Tsang und Voudouris keine Aussagen. Der einzige Parameter λ für GLS ist nach ihren Aussagen jedoch relativ unempfindlich gegenüber Änderungen.

6.3.7 Simulated Annealing

Die ersten Ergebnisse zum Problem BT_Mod-250-118 wurden von Baker 1993 [22] für SA veröffentlicht (siehe Tabelle 7.37). Er hat mit seinem Ansatz 21.050 Fehlerpunkte erzielt. Dabei sind die Fahrtkosten mit 4.390 Fehlerpunkten äußerst niedrig. Für die Erzeugung von Moves wird jeweils eine der Aktionen 2-opt, relocate, exchange und cross verwendet. Eine oder mehrere der vier Aktionen müssen demnach einen positiven Einfluss auf die Reduzierung der Fahrtkosten haben.

Zum Implementierungs- und Parametrierungsaufwand macht Baker keine Aussagen.

6.3.8 Evolutionsstrategien

Alle Ergebnisse zu ES am Problem BT_Mod-250-118 können Tabelle 7.37 im Anhang entnommen werden. Bei ES liefert ES(10+50) TypG + R das beste Ergebnis im Minimum (20.783 Fehlerpunkte). Im Durchschnitt schneidet ES (1+5) TypG + R mit 21.288,8 Fehlerpunkten am besten ab. Die Bestwerte von Fast GLS oder PSO können also nicht unterboten werden. ES benötigt – abhängig von der Populationsgröße – zehn bis dreizehn Stunden. Wie bei den Handelsproblemen ist auch hier eine Reparatur erforderlich. Diese ist jedoch weit weniger aufwändig, da lediglich Qualifikationsfehler korrigiert werden. Ohne die Reparatur wäre ES nicht anwendbar, da dann nicht garantiert werden kann, dass keine Qualifikationsfehler auftreten, was eine harte Nebenbedingung darstellt.

Im Vergleich der Mutationstypen ist die Mutation TypG mit Plusselektion und gleichen μ und λ immer signifikant besser als TypE. Und dies, obwohl TypG ursprünglich für kontinuierliche Lösungsräume und TypE für ganzzahlige Lösungsräume entwickelt wurde. Hierzu wurden Mittelwertvergleiche mit Hilfe von drei t-Tests mit einem Signifikanzniveau von 95% durchgeführt, bei denen die p -Werte immer kleiner als 0,001 waren. Bei der

Kommaselektion zeigt sich das gegenteilige Bild. Hier wurden ebenfalls t-Tests durchgeführt. Die Mutation TypE schneidet bei gleichen μ und λ immer signifikant besser ab als die Mutation TypG (jeweils mit p -Werten $< 0,001$).

Mit durchschnittlich 21.288,8 Fehlerpunkten bei ES (1+5) TypG + R ist Typ G jedoch signifikant besser als die beste Konfiguration von TypE. Die Plusselektion ist bei TypG der Kommaselektion überlegen. Es macht dort also Sinn, eine Selektion zu verwenden, bei der gute Individuen erhalten bleiben. In den Handelsproblemen war die Kommaselektion besser, da es mit der Plusselektion Probleme beim Entkommen aus einem lokalen Optimum gibt. Individuen können nicht in andere Bereiche vordringen und auf dem Weg dorthin über einen längeren Zeitraum schlechtere Ergebnisse hinnehmen. Der Einfluss von guten Eltern, die Eltern der neuen Generation werden können, war dort einfach zu groß. Beim Problem BT_Mod-250-118 schneidet die Plusselektion bei TypG besser ab als die Kommaselektion, da es für die Individuen beim British Telecom-Problem leichter ist, schnell in vielversprechende Regionen des Lösungsraums zu gelangen. Auf der anderen Seite ist es auch schnell möglich, in sehr schlechte Regionen zu gelangen. Kleine Änderungen genügen hierzu bereits. Wenn nun gute Individuen überleben, können sie die Suche positiv beeinflussen. Dieser Effekt ist bei TypE nicht feststellbar. Signifikante Unterschiede zwischen den Selektionsarten treten nur zwischen (1,5) und (1+5) auf, wo die Mittelwerte der Kommaselektion signifikant besser sind ($\alpha = 0,05$; $T = -2,296$; $p = 0,013$).

Die geschilderten Zusammenhänge sind in Abbildung 6.15 dargestellt. Der Verlauf der jeweils besten Experimente für ES TypG und TypE mit jeweils einer (1,5)- und einer (1+5)-Konfiguration sind dort zu sehen. Man erkennt, dass ES (1+5) TypG + R im Vergleich der vier Kurven am besten abschneidet. Dahinter liegen ES (1,5) TypE + R sowie ES (1+5) TypE + R sehr nahe beieinander. Am schlechtesten schneidet ES (1,5) TypG + R ab. Da dort keine Eltern überleben, kann selbst das jeweils beste Individuum einer Generation u.U. äußerst schlechte Werte annehmen. Dies ist hier sehr stark ausgeprägt und ist bei TypE sowie bei den Experimenten an den Handels- und Logistikproblemen in dieser Stärke nicht aufgetreten. Ausschläge nach oben waren dort sehr viel kleiner und wurden mit Hilfe der Reparatur gedämpft. Beim British-Telecom-Problem ist es dagegen mit nur einer einzigen Umplanung möglich, einen sehr schlechten Fitnesswert zu erzeugen. Bei TypE scheint die Suche deutlich gerichteter abzulaufen, weshalb dort nur minimale Ausschläge nach oben auftreten, die aufgrund der Skalierung in Abbildung 6.15 nicht auffallen.

Analog zum Logistikproblem sind auch hier sehr kleine Populationen bei TypE nicht mehr zu bevorzugen. Die Anzahl der Iterationen, die aus der Populationsgröße und dem Abbruchkriterium resultiert, welches nach einer bestimmten Anzahl an Bewertungen der Fitnessfunktion greift, scheint nicht mehr so stark ins Gewicht zu fallen. Moderat dimensionierte Populationen mit 10 bis 30 Eltern sowie 50 bis 200 Nachkommen haben signifikant bessere Mittelwerte als kleine Populationen ($\alpha = 0,05$) und sind daher bei TypE zu bevorzugen. Dies führt zu einer größeren Kenntnis über den Lösungsraum in je-

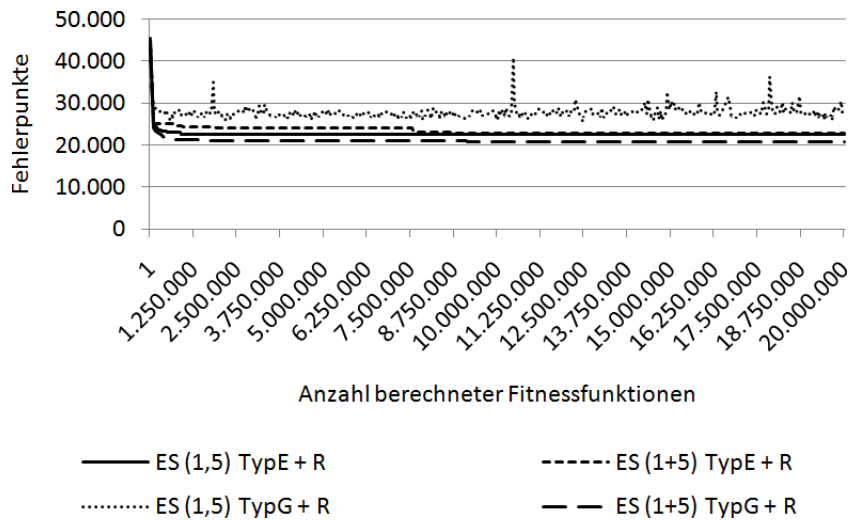


Abbildung 6.15: Verlauf der Fehlerpunkte für ES bei BT_Mod-250-118

der Generation und erleichtert das Entkommen aus einem lokalen Optimum. Der Effekt einer Vielzahl an Generationen tritt dahinter zurück. Bei TypG sind die Mittelwertunterschiede zwischen den Populationsgrößen eher uneinheitlich. Der beste Wert wird mit ES (1+5) TypG + R erzielt. Der nächsthöhere Mittelwert von ES (30+200) TypG + R weist jedoch keine signifikanten Mittelwertunterschiede auf ($\alpha = 0,05$; $T = -0,562$; $p = 0,288$). Bei TypG ist die Anzahl der Individuen anscheinend nicht sehr entscheidend für das Endergebnis. Vielmehr hat dort die Art der Selektion einen großen Einfluss auf die Ergebnisqualität.

Beim Aufwand für die Implementierung gibt es keine Unterschiede zwischen ES mit TypG und TypE. Hinsichtlich der Parametereinstellung ist TypE jedoch um ein Vielfaches aufwändiger als TypG, da dort umfangreichere Testläufe für die Einstellung der Mutation durchgeführt werden müssen. TypE reagiert beim British Telecom-Problem auch erheblich empfindlicher auf Parameteränderungen als beim Logistik- und Handelsproblem. Der Gesamtaufwand bei TypG ist daher innerhalb von fünf Manntagen und bei TypE erst innerhalb von zwanzig Manntagen zu bewältigen.

6.3.9 Genetische Algorithmen

Muller u.a. [277] nutzen einen verteilt arbeitenden GA mit und ohne Reparatur (siehe Tabelle 7.37 im Anhang). Der GA mit Reparatur erzielt 22.570 Fehlerpunkte. Ohne Reparatur sind es 23.790 Fehlerpunkte. Die Reparatur bewirkt also ein um 5,1% besseres Ergebnis beim GA. Sie setzt laut Muller u.a. den Fokus auf die Reduktion der Gesamtfehlerpunktzahl.

Vom GA sind leider nur wenige Details veröffentlicht worden. So sind keine Fahrtkosten oder Fehler für nicht erledigte Aufträge bekannt. Ebenso keine CPU-Zeiten oder Angaben zum Implementierungs- und Parametrierungsaufwand.

6.3.10 Particle Swarm Optimization

Mit PSO werden die bisher besten minimalen Fehlerpunkte am Problem BT_Mod-250-118 erzielt. Der Minimalwert von PSO (10) lBest + R beträgt 20.193 Fehlerpunkte und im Durchschnitt sind es 20.435,9 Fehlerpunkte. An dieser Problemvariante des British Telecom-Problems wurden in der Vergangenheit die meisten Lösungsverfahren erprobt. Die Ergebnisse können Tabelle 7.37 im Anhang entnommen werden. An dieser Stelle sei angemerkt, dass die minimale Schwarmgröße bei PSO in den Tabellen 10 Partikel sind. Beim Logistik- und Handelsproblem waren es noch 20. Während bei diesen beiden Problemen kleinere Schwärme mit z.B. 10 Partikeln schlechter abschnitten als ein Schwarm mit 20 Partikeln, sind beim British Telecom-Problem 10 Partikel optimal. Ein noch kleinerer Schwarm würde aber auch hier zu schlechteren Ergebnissen führen, da dann die Kenntnis über den Lösungsraum in jeder Iteration zu klein ist.

Hinsichtlich der CPU-Zeit ist PSO mit etwa drei Stunden im Vergleich zu ES mit zehn bis dreizehn Stunden bei 20 Mio. Berechnungen der Fitnessfunktion auf dem gleichen Rechner erheblich schneller. Bei PSO wird viel CPU-Zeit für die Berechnung der neuen Partikelpositionen verwendet. Bei ES werden zusätzlich zur Mutation auch eine Selektion und eine Rekombination eingesetzt, die zusätzliche CPU-Zeit verbrauchen. Entscheidend ist dabei auch, dass bei ES in der Reparatur aufgrund tiefer greifenderer Änderungen der Individuen in jeder Generation deutlich mehr zu tun ist als bei PSO. Während PSO und ES beim Logistik- und Handelsproblem in ARROW etwa gleich schnell waren, ist PSO nun bzgl. der Rechenzeit überlegen.

Im Vergleich verschiedener Schwarmgrößen erkennt man bei der gBest-Topologie, dass ein Schwarm mit zehn Partikeln im Durchschnitt signifikant besser abschneidet als ein Schwarm mit 200 Partikeln ($\alpha = 0,05$; $T = -16,444$; $p < 0,001$). Dieses Ergebnis deckt sich mit den gemachten Erfahrungen am Logistik- und Handelsproblem. Der Schwarm braucht auch hier eine Vielzahl an Iterationen, um von der Startlösung zu einer guten Lösung zu gelangen. Bei einem Abbruchkriterium, das auf der Anzahl an Berechnungen der Fitnessfunktion basiert, sind mit kleinen Schwärmen deutlich mehr Iterationen möglich als mit großen Schwärmen.

Aufgrund dieser Erkenntnisse wurden alle weiteren Berechnungen für die Nachbarschaftstopologien Wheel, lBest ($k=4$) und Circle mit 10 Partikeln durchgeführt. Hinsichtlich der minimalen Fehlerpunkte schneidet lBest, gefolgt von Circle, Wheel und gBest am besten ab. Ebenso hat lBest auch den signifikant besten Durchschnittswert ($\alpha = 0,05$; $T = -3,893$; $p < 0,001$) – gefolgt von Circle, gBest und Wheel. Es lohnt sich also, Infor-

mationen nicht sofort an alle Partikel weiterzugeben. Durch die Informationsweitergabe lediglich innerhalb kleinerer Nachbarschaften kann eine vorzeitige Konvergenz in einem lokalen Optimum vermieden werden. Dieses Ergebnis entspricht vielen Aussagen in der Literatur und dem Grundgedanken der Nachbarschaftstopologien [207], [89]. Es widerspricht jedoch den Erkenntnissen, die z.T. am Logistik- und Handelsproblem in dieser Arbeit gemacht wurden. Dort schnitt gBest häufig am besten ab. Denn gute Lösungen waren nur schwer zu finden. Wurden diese nicht unverzüglich allen Partikeln zugänglich gemacht, bestand die Gefahr, dass sie verloren gehen. Dieses Problem scheint hier nicht in diesem Maße zu bestehen.

Abbildung 6.16 zeigt den Verlauf der Fehlerpunkte für die besten Läufe der vier Nachbarschaften bei PSO (10). Alle vier Kurven liegen recht nahe beieinander. Lediglich die Wheel-Topologie benötigt am Anfang etwas länger zur Reduzierung der Fehlerpunkte.

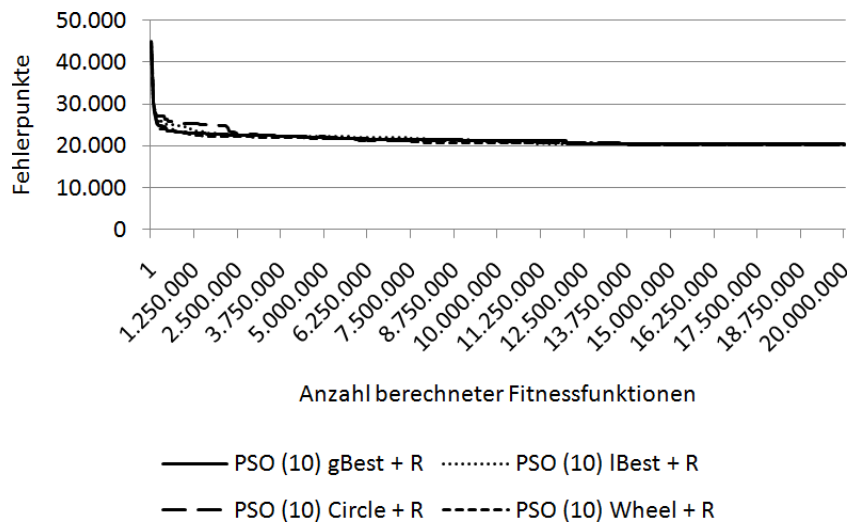


Abbildung 6.16: Verlauf der Fehlerpunkte verschiedener Nachbarschaftstopologien für PSO bei BT_Mod-250-118

Der Aufwand für die Implementierung und Parametereinstellung ist innerhalb von fünf Manntagen zu bewältigen. Beim Logistik- und Handelsproblem muss lediglich der Parameter für die zweite Aktion in der Positionsberechnung eingestellt werden. Für das British Telecom-Problem wurde PSO weiter modifiziert. Es existieren nun sechs statt vier Parameter, die jetzt deutlich empfindlicher reagieren. Dies macht die Parametereinstellung etwas aufwändiger.

6.3.11 Multiagentensysteme

Nachfolgend wird zunächst auf die Ergebnisse des Agentenansatzes von Kokkoras u.a. eingegangen, woraufhin Ausführungen zu den Ergebnissen des eigenen MAS folgen. Alle Ergebnisse sind in den Tabellen 7.37 und 7.38 im Anhang zu finden.

Ansatz von Kokkoras Kokkoras u.a. [217], [218], [330] führen lediglich Experimente am Problem BT_Rel-250-118 durch. Ein Vergleich ist daher nur mit CLP von Yang [428] möglich, der seine Verfahren dort ebenfalls erprobt hat.

Kokkoras u.a. implementieren diverse Agentenansätze, in denen sie CLP von Yang [428] verwenden. Alle nutzen Forward Checking (FC). Die zusätzliche Verwendung von FC hatte bereits bei Yang zu besseren Ergebnissen geführt. Kokkoras u.a. geben lediglich minimale Fehlerpunkte an. Zumeist fehlt auch die Angabe der Fehlerpunkte für Fahrtkosten und nicht erledigte Aufträge. Die CPU-Zeit ist recht gering und liegt je nach Ansatz zwischen 8 und 32 Sekunden. Für D-CLP3 (B) SD + FC wird keine CPU-Zeit genannt.

Das beste Ergebnis liefert D-CLP2 (B) SD + FC mit 22.078 Fehlerpunkten. Damit ist der Agentenansatz, der CLP von Yang verwendet, schlechter als CLP selbst. CLP SD + FC von Yang erzielt nämlich nur 21.426 Fehlerpunkte im Minimum. Besonders die Fehlerpunkte für nicht erledigte Aufträge sind dort geringer. Ebenso die Anzahl nicht erledigter Aufträge. Die Aufteilung des Problems auf Agenten (Servicecenter) behindert die Minimierung der Fehlerpunkte für unerledigte Aufträge. Dafür ist der Ansatz von Kokkoras u.a. deutlich schneller als CLP von Yang.

Insgesamt zeigt sich, dass beim Agentenansatz die Heuristik Same Direction First (SD) besser als Closest First (CL) ist. Dies gilt für Balloon (B) als auch für Mixed (M) bei der Problemzerlegung in Phase eins. Somit gewinnen Kokkoras u.a. die gleichen Erkenntnisse wie Yang bei CLP, der die Heuristiken SD und CL zuvor getestet hat. Kokkoras u.a. erwähnen zwar, dass sie auch Yang's Heuristik Least Busy First (LB) erproben, veröffentlichen hierzu jedoch keine Ergebnisse. LB schnitt bei Yang am schlechtesten ab.

Kokkoras u.a. testen in Phase eins verschiedene Varianten für die Zerlegung der Problemstellung auf einzelne Agenten (Servicecenter). Bei Mixed werden entfernte Aufträge von einem Servicecenter mit Balloon zugewiesen. Anschließend werden die nahen Aufträge mit Center of Gravity zugeteilt. Es zeigt sich, dass Balloon immer besser als Mixed bei sonst identischen Konfigurationen der Agentenansätze ist. Es macht also keinen Sinn, bei Mixed für die sehr nah am Servicecenter befindlichen Aufträge Center of Gravity. Für die alleinige Verwendung der Methode Center of Gravity werden von Kokkoras u.a. keine Ergebnisse veröffentlicht.

Es wird ebenfalls gezeigt, dass Phase drei beim Ansatz von Kokkoras die Ergebnisse weiter verbessern kann. So wurden 20 Aufträge zusätzlich zugewiesen und die Fehlerpunkte konnten um 10,5% reduziert werden.

Hinsichtlich des Implementierungs- und Parametrierungsaufwands machen Kokkoras u.a. keine Aussagen.

Eigener Ansatz Dem eigenen Agentenansatz, der in ähnlicher Form bereits beim Logistik- und Handelsproblem implementiert wurde, gelingt es beim Problem BT_Mod-250-118 zwar, die Fahrtkosten auf äußerst niedrige Werte zu reduzieren. Jedoch sind die Fehlerpunkte für unerledigte Aufträge noch verhältnismäßig hoch, was das Gesamtergebnis schmälert. So werden im Durchschnitt 23.494 und im Minimum 22.977 Fehlerpunkte erzielt.

Der eigene Agentenansatz unterscheidet sich vom Ansatz von Kokkoras u.a. durch die deutlich stärkere Zerlegung der Problemstellung auf Agenten. Hier repräsentieren auch einzelne Aufträge und Techniker einen Agenten, wogegen bei Kokkoras u.a. für jedes Servicecenter ein Agent erzeugt wird, dem eine Vielzahl an Aufträgen zugeordnet ist. Zwar ist es dort z.T. möglich, agenten-/servicecenterübergreifend Aufträge zuzuweisen, um Fahrtkosten zu minimieren. Dies gelingt bei einer stärkeren Zerlegung aber erheblich besser, bei der jeder Auftrag einen Agenten hat. Einschränkungen in der Zuweisung von Technikern, die einem Servicecenter zugeordnet sind, bestehen beim eigenen Ansatz nicht.

Bereits in Untersuchungen zum Logistik- und Handelsproblem wurde gezeigt, dass die starke Zerlegung auf Auftrags- oder Mitarbeiterebene einen Nachteil hat. Es fehlt eine übergeordnete Instanz, die Umplanungen vornimmt, welche von einzelnen Agenten niemals vorgenommen werden können. Die Ergebnisse legen nahe, dass eine weniger starke Zerlegung lediglich auf Servicecenterebene diesem Problem begegnet. Daher sind die Fehlerpunkte für nicht erledigte Aufträge beim Ansatz von Kokkoras auch erheblich niedriger.

Das Abbruchkriterium wurde bei allen selbst erstellten Lösungsverfahren für das British Telecom-Problem auf 20 Mio. Berechnungen der Fitnessfunktion gesetzt. Eine Ausnahme bildet (neben KV) der eigene Agentenansatz. Dort treten ablaufbedingt im Durchschnitt nach 142.962 Berechnungen der Fitnessfunktion garantiert keine Verbesserungen mehr auf. Weitere Berechnungen sind demnach nicht zielführend. Durch die geringe Anzahl an Berechnungen der Fitnessfunktion ist auch die CPU-Zeit mit durchschnittlich 39 Sekunden sehr gering. 20 Mio. Berechnungen würden ca. 5.456 Sekunden dauern, was im Vergleich zu PSO und ES um den Faktor zwei bis neun geringer ist. Der Agentenansatz ist im Vergleich zu PSO und ES schneller, da er recht einfache Berechnungen beinhaltet und keine aufwändige Positionsneubestimmung oder Mutation, Rekombination etc.

beinhaltet.

Für die Implementierung des Agentenansatzes werden maximal fünf Manntage benötigt. Hierbei ist sehr viel problemspezifisches Wissen erforderlich. Parameter sind nicht einzustellen.

6.3.12 Vergleich der Lösungsverfahren

In den Kapiteln 6.3.2 bis 6.3.11 wurden die Ergebnisse diverser Lösungsverfahren erörtert. Soweit bekannt, wurde dabei auch auf die geeignete Parametrierung eingegangen, was eine Forschungsfrage dieser Arbeit ist. Nachfolgend wird zunächst geklärt, ob sich eher klassische OR-Verfahren, Metaheuristiken oder MAS für das Problem der British Telecom eignen. Anschließend werden weitere Forschungsfragen beantwortet.

Die vollständige Enumeration (Kapitel 4.2) und klassische OR-Verfahren (Kapitel 6.3.2) sind nicht für das British Telecom-Problem geeignet. Die Rechenzeiten würde alle praktikablen Grenzen sprengen. Das Auftreten von Nichtlinearitäten erschwert zudem die Abbildung und macht den Einsatz von MIP-Verfahren unmöglich.

Die Ergebnisse punkt- und populationsbasierter Metaheuristiken wurden in den Kapiteln 6.3.5 bis 6.3.10 erläutert. Zusätzlich wurden CLP und ein selbst entwickeltes KV als Vergleichsverfahren herangezogen. Das bisher beste Verfahren vor dieser Arbeit war Fast GLS mit 20.433 Fehlerpunkten (dies ist ein Minimalwert – Durchschnittswerte wurden nicht angegeben). Im Durchschnitt können weder PSO noch ES diese Fehlerpunkte unterbieten. ES findet auch innerhalb von 30 Replikationen kein besseres Ergebnis als Fast GLS. Im Minimum gelingt es jedoch PSO mit einer kleinen Population und verschiedenen Nachbarschaftstopologien bessere Ergebnisse zu liefern. Die relevanten t-Tests können Tabelle 6.13 entnommen werden. Da von allen Lösungsverfahren anderer Autoren die Einzelwerte der Replikationen fehlen, können mit ihnen keine t-Tests angestellt werden.

Der Verlauf von ES (1+5) TypG + R sowie PSO (10) lBest + R ist in Abbildung 6.17 dargestellt. ES konvergiert anfangs etwas schneller. PSO gelingt es jedoch sehr rasch, geringere Fehlerpunkte als ES zu erzeugen.

Während sich klassische OR-Verfahren als ungeeignet erweisen und PSO als Metaheuristik sehr gute Ergebnisse liefert, bleibt zu klären, ob MAS beim British Telecom-Problem anwendbar sind. Beide Agentenansätze kommen nicht an die Bestwerte von PSO und Fast GLS heran und sind daher nicht gut für diese Problemstellung geeignet. Für das eigene MAS ist ebenfalls ein t-Test in Tabelle 6.13 dargestellt.

In Abbildung 6.18 sind die Mittelwerte relevanter Lösungsverfahren dargestellt. Die senk-

6.3 Kombinierte Personaleinsatzplanung und Tourenplanung bei Technikern

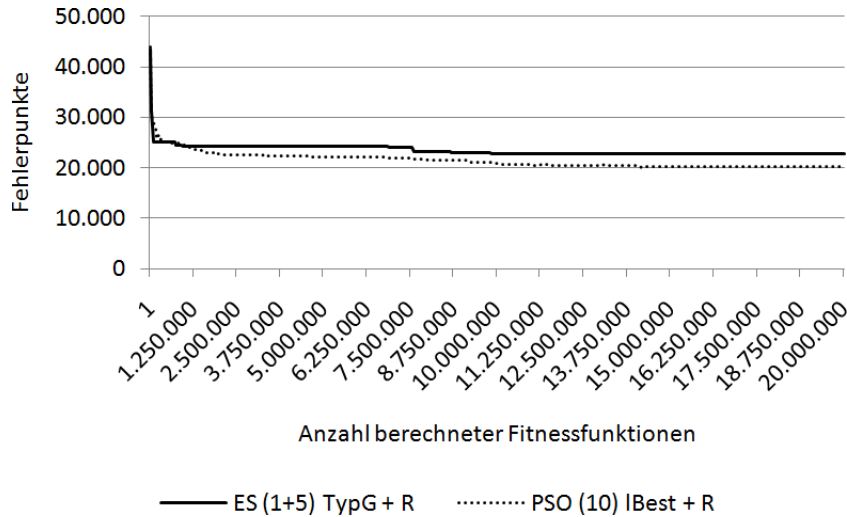


Abbildung 6.17: Verlauf der Fehlerpunkte für ES und PSO bei BT_Mod-250-118

Tabelle 6.13: t-Tests (einseitig) bei BT_Mod-250-118

H_1	T	df	Signifikanz H_0	95% Konfidenzintervall der Differenz	
				untere	obere
PSO (10) lBest + R < PSO (10) Circle + R	-2,205	58	0,016	-122,254	-16,812
PSO (10) lBest + R < PSO (10) gBest + R	-3,893	58	< 0,001	-213,832	-85,368
PSO (10) lBest + R < PSO (10) Wheel + R	-5,303	58	< 0,001	-265,415	-138,185
PSO (10) lBest + R < ES (1+5) TypG + R	-17,130	44,925	< 0,001	-936,594	-769,339
PSO (10) lBest + R < KV 3	-148,247	30,339	< 0,001	-23.841,575	-23.302,025
PSO (10) lBest + R < MAS	-47,815	38,166	< 0,001	-3.165,986	-2.950,347

rechten Striche definieren den Bereich, in dem sich jeweils die Minimal- und Maximalwerte der 30 Replikationen befinden. Da von den Verfahren anderer Autoren nur jeweils ein Wert bekannt ist, wird dieser analog zu den Mittelwerten mit einem Querstrich dargestellt. KV 3 ist nicht im Diagramm zu finden, da die Ergebnisse sehr groß sind. Die Skalierung des Diagramms würde mit KV 3 kaum Details erkennen lassen. Entsprechend der zuvor gemachten Aussagen ist in Abbildung 6.18 zu erkennen, dass die Ergebnisse von PSO recht gut sind. Die Mittelwerte können jedoch nicht den Bestwert von Fast GLS unterbieten. Da von Fast GLS und anderen Verfahren jedoch keine Einzelwerte vorliegen, kann ein Vergleich lediglich anhand der Minimalwerte vorgenommen werden. Diese sind bei einigen PSO-Konfigurationen besser als bei Fast GLS. Deutlich sind ebenfalls die große Varianz und die recht schlechten Ergebnisse bei ES und MAS zu erkennen.

Von allen nicht selbst erstellten Lösungsverfahren zum British Telecom-Problem ist nicht bekannt, wie häufig die Fitnessfunktion berechnet wurde. Nur bei den Quellen [18], [428] und [383] ist die CPU-Zeit angegeben. Da sich die verwendeten Rechner und die Implementierung der Verfahren stark unterscheiden, ist ein Vergleich der Verfahren anhand

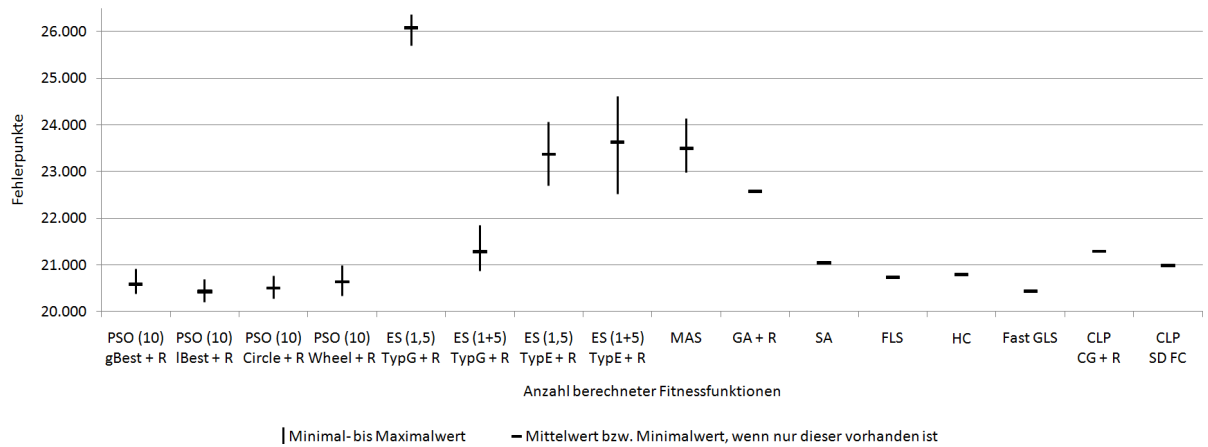


Abbildung 6.18: Vergleich verschiedener Lösungsverfahren bei BT_Mod-250-118

der CPU-Zeiten nur schwerlich möglich. Grundsätzlich kann festgehalten werden, dass Agenten, KV und CLP im Vergleich zu HC, PSO und ES erheblich schneller sind. Die Berechnungen mit FLS, HC und Fast GLS wurden Ende der 90er auf einem schnellen Rechner angestellt. Es ist davon auszugehen, dass die benötigten Rechenzeiten auf dem für Agenten, PSO und ES verwendeten Rechner deutlich kleiner sein dürften. Ohne einheitliches Abbruchkriterium können derartige Aussagen aber nur als Richtwerte gesehen werden. Bei der vollständigen Enumeration und bei klassischen OR-Verfahren wurde gezeigt, dass die zu erwartenden Rechenzeiten nicht praktikabel sind.

Zum Implementierungs- und Parametrierungsaufwand können nur Aussagen zu PSO, ES und dem eigenen Agentenansatz gemacht werden. Alle anderen Verfahren wurden nicht selbst implementiert und es sind diesbezüglich auch keine Aussagen der Autoren bekannt. PSO, ES TypG sowie der eigene Agentenansatz liegen mit maximal fünf Manntagen gleich auf. Lediglich ES TypE benötigt maximal 20 Manntage, was an den sehr zeitintensiven Parametertests liegt.

Zusammenfassend kann also festgehalten werden, dass klassische OR-Verfahren ungeeignet für das British Telecom-Problem sind. Auf der anderen Seite eignen sich Metaheuristiken sehr gut. MAS kommen nicht an die Bestwerte einiger Metaheuristiken heran und sollten daher nicht gewählt werden. Sie könnten dann interessant sein, wenn die CPU-Zeit ein begrenzender Faktor sein sollte.

Aus diesen Erkenntnissen folgt, dass auch beim British Telecom-Problem zentrale Lösungsansätze – hier: ausgewählte Metaheuristiken – den dezentralen MAS überlegen sind. Dies konnte nicht nur mit eigenen Experimenten belegt werden. Auch die Ergebnisse von Yang [428] mit CLP waren besser als die von Kokkoras u.a. [217], [218], [330] mit einem MAS, welches CLP nutzt. Die Aussage von Puppe u.a. [315], nach der sich zentrale Ansätze weniger gut für die Dienstplanung eignen als MAS, lässt sich also nicht auf das British Telecom-Problem übertragen.

Grundsätzlich kann der neue Bestwert von PSO als sehr gut erachtet werden. Schließlich wird seit vielen Jahren mit immer neuen Ansätzen und schnelleren Rechnern an dieser Problemstellung gearbeitet. Die Frage, ob sich die erprobten Lösungsverfahren für das Problem der British Telecom, was eine praktische Problemstellung ist, eignen, kann mit Einschränkungen bejaht werden. Während Verfahren wie die vollständige Enumeration oder klassische OR-Verfahren ungeeignet sind, kam in der Vergangenheit bereits der SA-Ansatz bei der British-Telecom zum Einsatz. Alle anderen getesteten Verfahren – außer GA und CLP von Kokkoras u.a. – sind in der Lage, bessere Ergebnisse zu liefern.

Die Forschungsfrage hinsichtlich der betriebswirtschaftlichen Effekte, die mit hochflexiblem WFM realisiert werden können, kann beim British Telecom-Problem nicht quantitativ beantwortet werden. Es fehlen Vergleichswerte eines zyklischen Planungsvorgehens aus Personaleinsatzplanung und Tourenplanung (siehe Kapitel 3.4.1). Durch die Erzielung eines neuen Minimalwertes, der den bisherigen Wert um 1,2% unterbietet, lassen sich jedoch Effekte ableiten. Diese Einsparung wurde bei lediglich 118 Technikern an einem Arbeitstag erzielt. Die British Telecom setzt mehrere zehntausend Techniker an jedem Tag des Jahres ein. Selbst minimale Ergebnisverbesserungen bewirken somit erhebliche Verbesserungen bei den Fahrtkosten, beim Servicelevel sowie bei der Kundenzufriedenheit.

6.4 Abschließende Beurteilung der Lösungsverfahren

In den vorangegangenen Kapiteln wurden die Lösungsverfahren separat an jeweils einer der drei Problemstellungen beurteilt. Vor allem mit Blick auf die zentrale Forschungsfrage, bei der geklärt werden soll, ob sich eher klassische OR-Verfahren, Metaheuristiken oder MAS für die drei beispielhaften Probleme des hochflexiblen WFM eignen. Alle drei Problemstellungen stammen aus der Praxis und beinhalten Instrumente zur Steigerung der WFM-Agilität. Dies sind die untertägige Personaleinsatzplanung mit einer Vielzahl an Arbeitsplatzwechseln, die automatische Arbeitszeitmodellerstellung direkt in der Einsatzplanung sowie die Kombination von Touren- und Personaleinsatzplanung. Es konnte belegt werden, dass sich einige der untersuchten Verfahren sehr gut für Problemstellungen eignen, die direkt aus der Praxis stammen. In diesem Zusammenhang wurde ausführlich auf Parametereinstellungen eingegangen. Auch die Effekte, die sich durch die Instrumente des hochflexiblen WFM erzielen lassen, wurden dargestellt. Nachfolgend soll eine Beurteilung der Verfahren entsprechend der Kernfrage dieser Arbeit über alle drei Problemstellungen hinweg erfolgen.

Bereits in Kapitel 4.2 wurde gezeigt, dass sich die vollständige Enumeration bei keiner der drei Problemstellungen mit ihren vielfältigen Varianten praktikabel anwenden lässt. Die Rechenzeiten wären derart groß, dass es keinen Sinn macht, auf ein Ergebnis zu warten.

Ähnlich verhält es sich bei den klassischen OR-Verfahren. Zur Erprobung diverser Solver mit B&B, B&C sowie OA wird GAMS verwendet. Es stellt sich heraus, dass die Probleme z.T. stark vereinfacht werden müssen, um schnell optimalen Ergebnissen zu erzielen. Diese Ergebnisse sind jedoch unbrauchbar, da sie ja nicht das eigentliche Problem lösen, sondern eine Problemstellung, bei der wesentliche Nebenbedingungen vernachlässigt werden. Verzichtet man auf eine Komplexitätsreduzierung, so treten CPU-Zeiten in einer nicht akzeptablen Größenordnung auf. Der Einsatz von klassischen OR-Verfahren macht bei den drei Problemstellungen demnach keinen Sinn.

Bei den Metaheuristiken wird aus der Gruppe der punktbasierten Verfahren zunächst auf HC eingegangen. Der eigene HC-Ansatz zeichnet sich durch seine Einfachheit aus. Er ist daher auch sehr schnell implementierbar. Während die Ergebnisqualität beim Handelsproblem sehr schlecht ist und beim British Telecom-Problem im Mittelfeld liegt, werden beim Logistikproblem recht gute Werte erzielt – im Einzelfall sogar Bestwerte. Da HC bei jeder Iteration eine Berechnung der Fitnessfunktion erfordert, ist die CPU-Zeit im Vergleich zu MAS extrem hoch. HC ist jedoch schneller als ES oder PSO, da intern weniger Operationen ausgeführt werden. HC eignet sich demnach vor allem für eher einfach strukturierte Problemstellungen mit weniger komplexen Nebenbedingungen, die mit Hilfe von einfachen Umplanungen eingehalten werden können.

Ein weiteres punktbasiertes Verfahren ist TS, welches von Helbig [178] im Rahmen einer Diplomarbeit an der TU Ilmenau erstellt wurde. Ergebnisse liegen daher nur für das Logistikproblem ohne Stammarbeitsplatzregelung vor. An diesem Problem erzielt TS sehr gute Ergebnisse. Vor allem bei den einzelnen Tagen. Noch bessere Ergebnisse werden immer mit einer hybriden TS-Variante erzielt, in der regelbasiert zwischen TS und HC gewechselt wird. Dieses hybride Verfahren liefert Bestwerte an den einzelnen Tagen. Hinsichtlich Laufzeit sowie Implementierungsaufwand sind keine Details von HC und TS + HC bekannt.

Mit GLS bzw. Fast GLS wird ein punktbasiertes Lösungsverfahren aufgegriffen, welches beim British Telecom-Problem sehr gut abschneidet [383]. GLS ist beim Logistik- und Handelsproblem jedoch unwirksam. Im Vergleich zum British Telecom-Problem, bei dem 250 Aufträge zu verplanen sind (Features bei GLS), müssen beim Logistik- und Handelsproblem mehrere tausend Matrixelemente als Feature berücksichtigt werden. Bevor GLS bei einer derart großen Zahl an Features seine Wirkung entfalten kann, hat das Abbruchkriterium längst gegriffen. In Tests war GLS am Logistik- und Handelsproblem daher nicht besser als HC. Nur die CPU-Zeit und der Implementierungsaufwand wurden durch GLS erhöht. Es zeigt sich also, dass der Auswahl und der Anzahl der Features eine große Bedeutung zukommt.

Das letzte punktbasierte Verfahren dieser Arbeit, das hier nur am Rande erwähnt werden soll, ist SA. Es wurde 1993 für die Technikereinsatzplanung bei der British Telecom von Baker [22] entwickelt und war Bestandteil einer Softwarelösung. Über dieses Verfahren sind wenig Details bekannt. Das Ergebnis beim British Telecom-Problem liegt

zwar vor GA und dem CLP-Ansatz von Kokkoras u.a. Dennoch kommt es nicht an die Ergebnisse von Fast GLS, ES oder PSO heran.

Blickt man in den vorangegangenen Kapiteln auf die Ausführungen zu vergleichbaren Arbeiten in der Literatur, so stehen dort punktbasierte Verfahren immer wieder im Fokus. Z.T. mit sehr guten Ergebnissen. Diese Stellung verteidigen sie auch hier. Zwar schneiden sie nicht bei allen Problemen dieser Arbeit sehr gut ab, liefern aber beim einfacheren Logistikproblem äußerst gute Ergebnisse.

Neben den punktbasierten Lösungsverfahren werden nun populationsbasierte Verfahren erörtert. Besonders umfassende Experimente wurden mit ES angestellt. Die Implementierung von ES und die Parametereinstellung sind nicht trivial. Beim British Telecom-Problem werden für ES mit TypE sogar bis zu 20 Tage benötigt – vor allem für die aufwändige Parametereinstellung. Zudem liegen die CPU-Zeiten je nach Problemstellung im mehrstündigen Bereich. ES liefert aber meist sehr gute Ergebnisse. Beim Handelsproblem ist es fast immer das beste Lösungsverfahren. Und auch im Logistik- und British Telecom-Problem sind die Ergebnisse recht gut. Im Rahmen dieser Arbeit wurden diverse Populationsgrößen getestet. Z.T. wurde dabei eine Tendenz hin zu kleinen Populationen ersichtlich. Der Grund für diesen Effekt liegt im Abbruchkriterium, dass nach einer bestimmten Anzahl an berechneten Fitnessfunktionen greift. Zudem macht es bei allen drei Problemen Sinn, nur wenige Änderungen pro Generation zu machen. Die Gefahr von Verschlechterungen ist sonst zu groß. Beide Zusammenhänge bewirken häufig, dass kleine Populationen besser abschneiden als große, da eine kleine Population mehr Generationen durchlaufen und ein Individuum somit mehr Änderungen vornehmen kann. Diese Aussage ist jedoch nicht allgemeingültig. So gibt es beim British Telecom-Problem z.B. Ausnahmen. Auch hinsichtlich eines überlegenen Selektionstyps kann keine allgemeingültige Aussage getroffen werden. So ist die Kommaselektion beim Handelsproblem hervorragend geeignet. Wogegen die Plusselektion beim British Telecom-Problem besser abschneidet. Beim Mutationstyp zeigt sich eine kleine Überlegenheit von TypG gegenüber TypE, was jedoch nicht generalisiert werden soll. Zusammenfassend wird festgestellt, dass die geeignete Parametereinstellung bei ES sehr problemspezifisch ist und im Rahmen von Tests ermittelt werden sollte.

Ein zweites Verfahren aus der Gruppe der EA ist GA, welches lediglich am British Telecom-Problem von Muller u.a. [277] erprobt wurde. Details sind zu diesem Verfahren kaum bekannt. Die Ergebnisse von GA sind recht schlecht. Er ist lediglich in der Lage, einige der schlechteren CLP- und ES-Varianten zu schlagen. Es ist leider nicht klar, ob sich die Ergebnisqualität durch Modifikationen des GA-Ansatzes oder durch eine Ausweitung des Abbruchkriteriums hätte verbessern lassen.

Ein weiteres populationsbasiertes Lösungsverfahren dieser Arbeit ist PSO. Es gehört der Gruppe der schwarmintelligenten Verfahren an und wird an allen drei Problemstellungen umfassend untersucht. Dabei wird es stark für kombinatorische Problemstellungen modifiziert. Derartige Anpassungen wurden nur in Grundzügen von anderen Autoren

angewendet. Gerade die zusätzliche Fähigkeit von PSO beim British-Telecom-Problem zu „sehen“ ist neuartig und zudem äußerst erfolgreich. Wie bei ES ist auch hier die CPU-Zeit mit z.T. mehreren Stunden recht hoch. Und auch der Implementierungs- und Parametrierungsaufwand ist ähnlich groß wie bei ES. Insgesamt ist PSO jedoch in der Lage, Bestwerte beim Logistikproblem (vor allem bei der Variante mit Stammarbeitsplatzregelung) sowie beim British Telecom-Problem zu liefern. Beim Handelsproblem findet PSO bei einem der vier Problemvarianten als einziges Verfahren im Durchschnitt das Optimum – ist bei den anderen drei Varianten aber immer hinter ES. Ähnlich wie bei ES ist auch bei PSO eine Tendenz hin zu kleineren Schwärmen (bei ES Populationen) festzustellen. Die Begründung hierfür kann analog zu den Ausführungen bei ES erfolgen. Es ist vorteilhaft in jeder Iteration nur wenige Änderungen in einem Partikel vorzunehmen. Für einen guten Einsatzplan sind jedoch viele Änderungen ausgehend von der Startlösung erforderlich. Da das Abbruchkriterium auf der Anzahl an bewerteten Fitnessfunktionen basiert, können mit kleinen Schwärmen deutlich mehr Änderungen je Partikel vorgenommen werden, was dann zu den besseren Ergebnissen im Vergleich zu großen Schwärmen führt. Die Tendenz hin zu kleineren Schwärmen ist jedoch nicht allgemeingültig. So kann es durchaus zu Ausnahmen kommen. Wie bei der Schwarmgröße kann auch bei den Nachbarschaftstopologien keine allgemeingültige Aussage getroffen werden, ob sich nun gBest, lBest, Circle oder Wheel am besten eignen. Dies deckt sich mit den Erkenntnissen in der Literatur (siehe Kapitel 4.14.2.1). Recht häufig werden jedoch mit gBest sehr gute Werte erzielt, wogegen Wheel meist am schlechtesten ist. Zusammenfassend kann gesagt werden, dass PSO sehr gut für die meisten getesteten Problemvarianten geeignet ist. Wie bei ES kann auch hier keine generelle Handlungsempfehlung bzgl. der optimalen Parameterwahl getroffen werden. Diese muss bei neuen Problemstellungen in Tests ermittelt werden. Diese Tests könnten jedoch mit kleinen Schwärmen und einer gBest-Topologie beginnen.

In dieser Arbeit wurde deutlich, dass Reparaturverfahren, die aufgrund hinterlegter Regeln Fehler korrigieren, bei zahlreichen Lösungsverfahren erfolgreich angewendet werden. Eine Ausnahme ist bei den Varianten des Logistikproblems mit Stammarbeitsplatzregelung zu finden. Dort korrigiert die Reparatur die Individuen bzw. Partikel derart stark, dass sie bestimmte Bereiche des Lösungsraums nie erreichen oder durchschreiten können. Bei allen anderen Problemstellungen war die Reparatur jedoch immer sehr erfolgreich und z.T. auch die Voraussetzung für die Anwendung von ES und PSO. Der Einsatz einer Reparatur ist also immer dann hilfreich, wenn sie zwar Korrekturen vornimmt, dies jedoch nicht zu intensiv tut und somit die Suche stark behindert.

KV wurden beim Handelsproblem als Heuristik zum Vergleich herangezogen. Besonders vor dem Hintergrund, dass beide KV jeweils Bestandteil kommerzieller Lösungen sind. Trotz dieses Umstands konnte keines der Verfahren überzeugen. Lediglich HC war bei den vier Varianten des Handelsproblems noch schlechter. Hier zeigt sich, dass KV nicht sehr flexibel einsetzbar sind. Sie können immer dann schlecht abschneiden, wenn die Problemstellung, die sie lösen sollen, nicht exakt auf sie zugeschnitten ist. So hat KV 2 z.B. Probleme mit geringfügig beschäftigten Mitarbeitern und KV 1 erzeugt häufig zu

lange Arbeitszeitmodelle. Die CPU-Zeiten sind bei den KV jedoch mit maximal zehn Minuten recht gut. Auch beim Problem der British Telecom kam ein KV zum Einsatz, welches ebenfalls sehr schlecht abschneidet. Dafür wird das Ergebnis von KV 3 bereits innerhalb einer Sekunde geliefert. Als eigenständiges Lösungsverfahren ist es beim British Telecom-Problem unbrauchbar. Für die Ermittlung einer Startlösung für ES und PSO aber sehr gut geeignet.

Ein weiteres Lösungsverfahren ist CLP. Es wurde nicht selbst implementiert und von anderen Autoren lediglich am British Telecom-Problem erprobt. Daher wird an dieser Stelle nur kurz auf CLP eingegangen. In dieser Arbeit wird zum einen der Ansatz von Azarmi und Abdul-Hameed [18] sowie der Ansatz von Yang [428] behandelt. Auffällig sind die zahlreichen CLP-Variationen, die von den Autoren erstellt werden. Alle Ansätze benötigen eine sehr geringe CPU-Zeit von meist weniger als einer Minute – in Ausnahmen von bis zu zehn Minuten. Im Vergleich schneidet der Ansatz von Yang besser ab als der Ansatz von Azarmi und Abdul-Hameed. CLP kommt bzgl. der Ergebnisqualität nicht an die Bestwerte von PSO, ES, Fast GLS, FLS und HC heran und ist daher weniger gut geeignet, wenn die CPU-Zeit nicht der begrenzende Faktor ist.

Zuvor wurde u.a. ausführlich auf klassische OR-Verfahren und auf Metaheuristiken eingegangen. Im Rahmen der zentralen Forschungsfrage soll auch geklärt werden, wie sich MAS für die drei beispielhaften Problemstellungen des hochflexiblen WFM eignen. In dieser Arbeit wurde daher ein selbst entwickeltes MAS an allen drei Problemstellungen erprobt. Zudem kann beim British Telecom-Problem auf die Ergebnisse eines zweiten MAS von Kokkoras u.a. [217], [218], [330] zurückgegriffen werden. Insgesamt wird deutlich, dass sich MAS (dezentral) hinsichtlich der Ergebnisqualität nicht mit ausgewählten Metaheuristiken (zentral) messen können. Gerade bei den komplexeren Problemen für Handel und Technikereinsatz sind die Ergebnisse nicht sonderlich gut. MAS gehören aber auch dort nicht zu den schlechtesten Verfahren. Besonders vielversprechend sind die Ergebnisse beim Logistikproblem, wo das MAS sehr nahe an die Bestwerte heran kommt. Der große Vorteil der hier erprobten MAS ist die geringe CPU-Zeit, die meist kleiner als eine Sekunde ist. Und selbst beim British-Telecom-Problem, wo der eigene Agentenansatz modifiziert werden musste, wird das Ergebnis in 39 Sekunden geliefert. Die geringe CPU-Zeit ist darin begründet, dass der eigene MAS-Ansatz eine Lösung konstruiert und beim Logistik- und Handelsproblem nur einmal am Ende die Fitnessfunktion berechnen muss. Da die CPU-Zeit in dieser Arbeit aber nicht begrenzt ist, sollten MAS nicht die erste Wahl sein. Geeigneten Metaheuristiken ist der Vorzug zu geben. Bei veränderten Rahmenbedingungen, in denen der Planer ein Ergebnis auf Knopfdruck bekommen möchte, sind MAS, die eine Lösung konstruieren, neben CLP jedoch allen anderen, in dieser Arbeit getesteten Verfahren, überlegen.

In dieser Arbeit wurden für alle drei Problemstellungen auch diverse hybride Ansätze diskutiert. Besonders zu beachten ist die Reparatur, die als eine Lokale Suche fast immer zu deutlichen Ergebnisverbesserungen beiträgt. Zudem liefern PSO und ES beim Handels- und British-Telecom Problem ohne eine Reparatur auch keine brauchbaren Ergebnisse.

Die Kombination einer generischen Metaheuristik mit einer problemspezifischen Reparatur scheint bei den untersuchten Problemstellungen demnach sehr vielversprechend zu sein. Außer dem bereits erwähnten Ansatz aus TS und HC und dem Einsatz der Reparatur konnte kein hybrider Ansatz zu einer Ergebnisverbesserung beitragen oder überhaupt umgesetzt werden. Eine Ausnahme besteht beim British Telecom-Problem, wo die Startlösungen für ES und PSO von einem KV erzeugt wird. Der Übersichtlichkeit halber wurden ES und PSO bei der Planung des Technikereinsatzes nicht als hybrides Verfahren deklariert. Die Gliederung dieser Arbeit hätte sich zu stark verschoben, zumal beide Verfahren ein Fokus dieser Arbeit sind.

An dieser Stelle soll auch die Robustheit der erprobten Lösungsverfahren mit Blick auf deren Anwendung bei verschiedenen Problemstellungen beurteilt werden. Die Probleme zeichnen sich durch eine hohe Komplexität und sehr individuelle Nebenbedingungen aus. PSO und ES nutzen daher eine Reparatur, die speziell auf die Anforderungen des jeweiligen Planungsproblems zugeschnitten ist. Aber auch der Ablauf von MAS, TS, TS + HC, CLP, KV, etc. ist sehr problemspezifisch gestaltet. Dies führt dazu, dass sich einige Verfahren gar nicht bei anderen Problemstellungen anwenden lassen bzw. äußerst schlechte Ergebnisse produzieren. Der beste Beleg ist HC, welches am Logistikproblem recht gut und beim Handelsproblem äußerst schlecht abschneidet. Erst nach einer individuellen Anpassung gelingt es z.B. bei PSO, ES oder MAS die Ergebnisqualität zu verbessern. Die hier erprobten Lösungsverfahren mit ihren entsprechenden Modifikationen müssen daher als wenig robust erachtet werden, wenn sie bei verschiedenen Problemstellungen zur Anwendung kommen. Innerhalb einer Problemstellung verhalten sie die Lösungsverfahren sehr viel robuster. Unterschiede treten meist nur aufgrund veränderter Lösungsraumgrößen zwischen Problemvarianten auf (z.B. Wochen- und Tagesprobleme beim Logistikdienstleister).

In Summe wurden 21 Problemvarianten, die aus den drei Bereichen Logistik, Handel und Technikereinsatz stammen, in dieser Arbeit untersucht. Mit Blick auf die zentrale Forschungsfrage wird deutlich, dass sich klassische OR-Verfahren nicht für die hier untersuchten Problemstellungen eignen. Ebenso verhält es sich mit MAS, die jedoch bzgl. der geringen CPU-Zeit positiv in Erscheinung treten. Am besten schneiden Metaheuristiken ab, wobei keine generelle Aussage darüber getroffen werden kann, welche Metaheuristik am geeignetsten ist. Deren Eignung hängt neben anderen Faktoren vor allem von der Problemgröße und der Komplexität der Nebenbedingungen ab.

Mit Blick auf die Einteilung in starke und schwache Heuristiken, werden alle in dieser Arbeit untersuchten Heuristiken als starke Heuristiken klassifiziert. Denn sie sind immer nur speziell für eine Problemstellung angepasst worden und liefern dort z.T. recht gute Ergebnisse. Ihre Anwendung bei anderen Problemstellungen – ohne Anpassung – ist jedoch nicht zielführend. Nur nach einer intensiven Parametrierung und ggf. Anpassung liefern sie auch bei anderen Problemstellungen z.T. recht gute Ergebnisse.

Es wurde gezeigt, dass ausgewählte Metaheuristiken besser für die drei Problemstel-

lungen des hochflexiblen WFM geeignet sind als MAS. Für Dienstplanungsprobleme, die durch starre Schichten gekennzeichnet sind, sagen Puppe u.a. [315], dass sich dort zentrale Lösungsverfahren weniger gut eignen als dezentrale. In einer Forschungsfrage dieser Arbeit sollte geklärt werden, ob sich diese Aussage auch auf die drei beispielhaften Probleme des hochflexiblen WFM übertragen lässt. Auf Basis verschiedenster Problemvarianten kann die Antwort auf diese Frage verneint werden. Hier eignen sich zentrale Lösungsansätze besser als dezentrale. Vielmehr ist aufgefallen, dass MAS unter dem Fehlen einer zentralen Planungsinstanz leiden, die tiefgreifende Umplanungen vornimmt, zu denen ein einzelner Agent nicht in der Lage ist. Fraglich ist zudem, ob die Aussage von Puppe u.a. [315] allgemeingültig für die Dienstplanung ist. Die Beantwortung dieser Frage ist jedoch nicht Gegenstand dieser Arbeit.

Die erzeugten Einsatzpläne wurden z.T. äußerst ausführlich untersucht. Planungsfehler konnten nur im Ausnahmefall und mit erheblichem Aufwand manuell identifiziert und beseitigt werden. Zusätzlich konnte beim Handelsproblem auf Vergleichswerte zweier kommerzieller Lösungen zurückgegriffen werden. Und auch beim British Telecom-Problem standen zahlreiche Vergleichsverfahren anderer Autoren als Benchmark zur Verfügung. Die Bestwerte, die mit einigen Lösungsverfahren erzielt wurden, können daher als überaus gut angesehen werden. Die Frage, ob sich die erprobten Lösungsverfahren für die drei Problemstellungen eignen, die direkt aus der Praxis stammen, kann mit Einschränkungen bejaht werden. Einschränkungen ergeben sich mit Blick auf die zentrale Forschungsfrage. Klassischen OR-Verfahren sind ungeeignet. MAS sollten nur gewählt werden, wenn die CPU-Zeit begrenzt ist, was hier jedoch nicht der Fall ist. Metaheuristiken sind überaus gut geeignet, wenn es gelingt, das richtige Verfahren mit der geeigneten Parametrierung zu wählen.

Im Rahmen der Experimente wurden die betriebswirtschaftlichen Effekte durch die Instrumente des hochflexiblen WFM z.T. eindrucksvoll dargestellt. So lassen sich beim Logistikdienstleister die Überdeckung im Personaleinsatz um 26% und die Unterdeckung um 63% reduzieren. Im Handel können die Ergebnisse kommerzieller Lösungen ebenfalls erheblich verbessert werden. Eine Reduktion der Überdeckung um 100% und eine Reduktion der Unterdeckung um 65% war mit geeigneten Lösungsverfahren an einigen Problemstellungen möglich. Dies berücksichtigt jedoch nicht die Effekte der automatischen Arbeitszeitmodellerstellung direkt in der Planung. Wie bereits in Kapitel 3.3.1 gezeigt wurde, wären die Effekte im Vergleich zu einer starren Schichtplanung noch weitaus größer. Und auch beim Problem der British-Telecom konnten die Ergebnisse etablierter Verfahren weiter verbessert werden. Eine Ergebnisverbesserung um 1,2% würde bei mehreren zehntausend Technikern der British Telecom eine beachtliche Fahrtkostensparnis und Erhöhung des Servicelevels ausmachen. Doch auch hier fehlen Vergleichswerte eines zyklischen Planungsvorgehens aus Personaleinsatzplanung und Tourenplanung, wie es in Kapitel 3.4.1 geschildert wurde. Daher kann die Forschungsfrage nur für das Logistikproblem quantitativ beantwortet werden. Beim Handels- und British Telecom-Problem fehlen die Vergleichswerte von Ansätzen, die keine Instrumente des hochflexiblen WFM enthalten. Lediglich Verbesserungspotenziale im Vergleich zu etablierten Verfahren konn-

ten dort quantitativ aufgezeigt werden.

Abschließend sei angemerkt, dass die Wahl der geeigneten Initialisierung in den Experimenten fast immer einen entscheidenden Einfluss auf die Ergebnisqualität hat. In vielen Fällen ist eine Initialisierung erfolgreich, die eine sehr einheitliche Startlösung erzeugt, z.B. ohne zahlreiche untertägige Arbeitsplatzwechsel. Wobei dies nicht generalisiert werden soll, da es auch Ausnahmen gibt. Beim British Telecom-Problem kann die Startlösung z.B. nicht ohne untertägige Auftragswechsel erfolgen, da jeder Auftrag nur einmal existiert. Hier hat sich KV 3 für die Erzeugung der Startlösungen von ES und PSO bewährt.

Kapitel 7

Zusammenfassung

Dieses Kapitel besteht aus zwei Teilen. Zunächst werden die wichtigsten Erkenntnisse dieser Arbeit kurz zusammengefasst. Abschließend werden zukünftige Forschungsgegenstände erörtert.

7.1 Fazit

Problemstellungen der starren Schicht- oder Dienstplanung wurden in der Vergangenheit bereits vielfältig untersucht. Dort werden klassische OR-Verfahren sowie KV am häufigsten genutzt [114]. Dies ist jedoch kein Indikator dafür, ob diese Lösungsverfahren generell am besten für derartige Planungsprobleme geeignet sind. Ein Ergebnisvergleich ist aufgrund der starken Unterschiede zwischen den Problemen kaum möglich. Zudem stammen die Problemstellungen meist nicht aus der Praxis bzw. wurden derart vereinfacht, dass sie z.B. von klassischen OR-Verfahren wie Simplex, B&B oder B&C gelöst werden können, dann aber keinen praktischen Nutzen haben [279].

Zunehmend ist bei Unternehmen ein Trend weg von der starren Schicht- oder Dienstplanung hin zu einer auf den Personalbedarf ausgerichteten Planung festzustellen [252], [298]. In diesem Zusammenhang wird der Begriff hochflexibles WFM geprägt. Mit Instrumenten wie der Planung untertägiger Arbeitsplatzwechsel, der Kombination aus Arbeitszeitmodellerstellung und Einsatzplanung sowie der kombinierten Personaleinsatz- und Tourenplanung kann der Personaleinsatz sehr gut an den Personalbedarf angepasst werden. Derartige Problemstellungen sind neuartig und fanden in der Forschung bisher wenig Beachtung. Daher werden in dieser Arbeit ausschließlich Problemstellungen untersucht, die Instrumente des hochflexiblen WFM einsetzen.

In diesem Zusammenhang soll die zentrale Forschungsfrage beispielhaft an drei Problemstellungen beantwortet werden, ob sich eher klassische OR-Verfahren, Metaheuristiken

oder MAS für Probleme des hochflexiblen WFM eignen. Im Rahmen der Ergebnisauswertungen in Kapitel 6 zeigt sich, dass die Ergebnisqualität bei ausgewählten Metaheuristiken am besten ist. Je nach Problemvariante waren TS, HC, TS + HC, PSO oder ES besser geeignet. Dabei hat sich gezeigt, dass die Kombination mit einer problemspezifischen Reparatur z.T. erst für den Erfolg einiger Metaheuristiken verantwortlich ist. Ohne Reparatur würden einige Verfahren keine brauchbaren Ergebnisse liefern. Die klassischen OR-Verfahren erwiesen sich aufgrund der zu langen Rechenzeit als untauglich. MAS konnten nicht an die Bestwerte der Metaheuristiken heranreichen und sind somit ebenfalls nicht optimal für die gewählten Problemvarianten des hochflexiblen WFM geeignet. Da sie jedoch äußerst schnell ein Ergebnis liefern, wären sie eine interessante Option für ein Rescheduling. Oder für Szenarien, in denen der Anwender schnell einen guten Planungsvorschlag benötigt, den er dann manuell anpasst. Eine begrenzte CPU-Zeit (nicht in dieser Arbeit) würde ihren Einsatz deutlich attraktiver machen. Metaheuristiken benötigten im Gegensatz zu den hier erprobten MAS mehrere Minuten oder Stunden je nach Problemstellung und Lösungsverfahren. Hinsichtlich des Implementierungs- und Parametrierungsaufwands gibt es zwischen den Verfahren kaum nennenswerte Unterschiede. Ein neues Verfahren ist meist innerhalb von fünf Manntagen komplett fertig gestellt. Lediglich bei ES ist der Parametrierungsaufwand gelegentlich höher. Und HC sowie KV ließen sich in wenigen Stunden implementieren. Voraussetzungen hierfür sind sehr gute Programmierkenntnisse sowie ein tiefes Wissen über die Lösungsverfahren, die Problemstellungen und das in dieser Arbeit erstellte Planungstool ARROW.

Aufgrund der in Kapitel 6 gewonnenen Erkenntnisse kann auch die Forschungsfrage beantwortet werden, ob sich eher zentrale oder dezentrale Lösungsansätze für die drei beispielhaft ausgewählten Probleme des hochflexiblen WFM eignen. Metaheuristiken – also zentrale Ansätze – sind deutlich besser geeignet. MAS fehlt eine zentrale Planungskomponente, die tiefgreifende Änderungen vornimmt, zu denen Agenten nicht in der Lage sind. Die Aussage von Puppe u.a. [315], dass dezentrale Lösungsansätze für die Dienstplanung besser geeignet sind als zentrale Ansätze, kann demnach nicht auf die hier untersuchten Probleme des hochflexiblen WFM übertragen werden.

Bei der Auswahl der Probleme wurden bewusst solche gewählt, die direkt aus der Praxis stammen (Echtdaten sind auf der Homepage der TU Ilmenau zu finden [153]). Zudem wurden sie nicht vereinfacht, um eine weitere Forschungsfrage zu beantworten. Es soll geklärt werden, ob diese praktischen Probleme des hochflexiblen WFM überhaupt mit den verwendeten Verfahren gelöst werden können. Zum einen sind diese Probleme sehr komplex und sehr groß. Zum anderen werden sehr hohe Anforderungen an die Ergebnisqualität gestellt. In Kapitel 6 wurde gezeigt, dass die besten Ergebnisse sehr nahe an den Lower Bounds liegen und im Einzelfall optimal sind. Auch konnten weitere manuelle Verbesserungen der erzeugten Pläne – wenn überhaupt – nur mit äußerst hohem Aufwand gefunden werden. Ausgewählte Metaheuristiken sind daher überaus gut für die praktischen Probleme geeignet.

Zahlreiche Lösungsverfahren, die in dieser Arbeit untersucht wurden, sind ursprüng-

lich für kontinuierliche Problemstellungen entwickelt worden. So z.B. PSO oder ES. Vor diesem Hintergrund ergibt sich die Forschungsfrage, wie derartige Lösungsverfahren an kombinatorische Problemstellungen angepasst werden sollten. Die Beantwortung wurde beispielhaft an drei Problemstellungen vorgenommen (Kapitel 4). Hierbei wurden z.T. neue Wege gegangen. Z.B. bei der Anpassung von PSO, wo Partikel beim Problem der British Telecom nun auch ihre Umwelt „sehen“ und sich dementsprechend verhalten können. Aber auch einige Anpassungen bei ES sind äußerst vielversprechend und liefern z.T. Bestwerte im Vergleich zu anderen Verfahren. Es zeigt sich, dass diese Anpassungen auf Basis von problemspezifischem Wissen meist derart individuell gestaltet sind, dass sich die Lösungsverfahren dann kaum für andere Probleme mit veränderten Rahmenbedingungen eignen. Dies schadet der Robustheit und erfordert jeweils problemspezifische Modifikationen.

Grundsätzlich stellt sich bei der Anwendung von Lösungsverfahren – besonders bei Verfahren, die für kombinatorische Problemstellungen modifiziert sind – die Frage nach der optimalen Parameterwahl. Im Rahmen dieser Arbeit wurden mehrere tausend Stunden an CPU-Zeit für Parametertests aufgewendet. Die Ergebnisse dieser sehr umfassenden Experimente und einige Handlungsempfehlungen sind in Kapitel 6 dargestellt.

Mit den Instrumenten des hochflexiblen WFM können erhebliche betriebswirtschaftliche Effekte erzielt werden. Wie diese konkret aussehen, war eine Forschungsfrage dieser Arbeit. In den Kapiteln 2.2, 3.2.1, 3.3.1 und 3.4 wurde zunächst allgemein auf die Potenziale der hier verwendeten Instrumente eingegangen. Exakte Berechnungen auf Basis der drei praktischen Problemstellungen wurden in den Kapiteln 6.1.10, 6.2.9 und 6.3.12 ausgewertet. Hier zeigen sich besonders Effekte in Richtung verbesserter Servicelevel (ggf. mit einhergehender Umsatzsteigerung [291]) sowie eine Kostenreduktion, die durch weniger Überstunden oder geringere Fahrtkosten realisiert wird.

7.2 Zukünftige Forschungsgegenstände

In dieser Arbeit wurden zahlreiche Forschungsfragen beantwortet. In diesem Zusammenhang sind auch völlig neue Fragen. Zudem mussten diverse Forschungsfragen von vornherein ausgeklammert werden, da sie den Rahmen dieser Arbeit sprengen würden.

Die Anzahl untätiger Techniker beim British Telecom-Problem ist bei allen Lösungsverfahren sehr hoch. Hierauf kann jedoch kein Einfluss genommen werden, da die Techniker in einer vorgelagerten Stufe für den zu planenden Tag eingeplant werden. Sie werden also zwangsläufig bezahlt – leisten aber nicht immer einen wertschöpfenden Beitrag. Hier liegt neben der Reduzierung der Fahrtkosten und der Anzahl unerledigter Aufträge das größte betriebswirtschaftliche Potenzial. Zur Reduktion der Anzahl unerledigter Aufträge können die untätigen Techniker nicht eingesetzt werden, da sie entweder nicht

qualifiziert sind, die Einsatzorte zu weit entfernt sind oder ihr Einsatz aufgrund der verwendeten Zielfunktion nicht lohnt. Interessant wäre daher eine Problemstellung, in der das British Telecom-Problem um die automatische Arbeitszeitmodellerstellung des Handelsproblems erweitert wird.

Im Rahmen dieser Arbeit wurden Lösungsverfahren an zahlreichen Varianten von drei Problemstellungen untersucht. Weitere Problemstellungen mit veränderten Rahmenbedingungen können zukünftig herangezogen werden, um die hier gemachten Aussagen auf ein breiteres Fundament zu stellen.

Obwohl bereits zahlreiche Lösungsverfahren mit diversen Ausprägungen untersucht wurden, konnte bei weitem nicht auf alle bekannten Verfahren eingegangen werden. Zukünftige Untersuchungen können sich daher weiteren Lösungsverfahren widmen und diese beurteilen.

In dieser Arbeit ist CPU-Zeit nicht der begrenzende Faktor. Die Berechnungen können problemlos über Nacht laufen. Interessant wären Betrachtungen, in denen die CPU-Zeit begrenzt ist. Hier stellt sich die Frage, wie die Lösungsverfahren dann an die Problemstellungen angepasst werden sollten, welche Lösungsverfahren am geeignetsten sind und wie sie parametrisiert werden sollten. Zudem kann ein Vergleich mit den hier gemachten Erkenntnissen erfolgen.

Analog zu einem veränderten Abbruchkriterium können die dort gestellten Fragen auch in Zusammenhang mit einer Mehrzieloptimierung geklärt werden. Die hier untersuchten Problemstellungen sollten recht leicht in diese Richtung anpassbar sein.

Alle untersuchten Problemstellungen unterliegen keinen dynamischen Änderungen. Ein Rescheduling findet also nicht statt. Besonders beim British Telecom-Problem untersuchen Tsang u.a. [379], [380], [382] dynamische Änderungen. Dies wäre mit Blick auf weitere Modifikationen der Lösungsverfahren und ihrer optimalen Parametrisierung sowie bzgl. eines Vergleichs der Lösungsverfahren ebenfalls ein lohnender Forschungsgegenstand für das Logistik- und Handelsproblem. Interessant wäre zudem eine dynamische Mehrzieloptimierung im Rahmen des hochflexiblen WFM.

Bei zwei Varianten des Handelsproblems wurde ein Planungshorizont von einem Jahr bearbeitet. Prinzipiell wäre es möglich, noch längere Planungszeiträume zu wählen. In Kombination mit einer Simulation verschiedener Belegschaftsstrukturen (Vollzeit-, Teilzeitkräfte, geringfügig Beschäftigte) oder einer Simulation von konjunkturellen Veränderungen mit einhergehenden Bedarfsänderungen ließe sich ein strategisches WFM betreiben. Dieses basiert dann auf tatsächlich berechneten Einsatzplänen und nicht bloß auf Stellenplänen und Stellenbeschreibungen. Selbst die Frage, was eine teure Fortbildung eines Mitarbeiters zur Verbesserung des Servicelevels beiträgt, kann so beantwortet werden. Aufgrund des sehr langen Planungshorizonts und der Notwendigkeit vieler Simulationsläufe, müssten die Lösungsverfahren nicht nur sehr exakt, sondern auch sehr

7.2 *Zukünftige Forschungsgegenstände*

schnell sein. Auf Basis der hier gemachten Erfahrungen dürften sich KV und vor allem MAS, die eine Lösung konstruieren, besonders eignen.

Literaturverzeichnis

- [1] ABDUL-HAMEED, W.: *Workforce scheduling in constraint logic programming CHIP*. ISR Technical Report, BT Laboratories, September, 1993.
- [2] ABDUL-HAMEED, W.: *Experimentation with parallel constraint logic programming*. MEng Thesis, Department of Computing, Imperial College, London, Juni, 1994.
- [3] ABDULLAH, S., E. K. BURKE und B. MC COLLUM: *An investigation of variable neighbourhood search for university course timetabling*. In: *Proceedings of the 2nd Multidisciplinary International Conference on Scheduling: Theory & Applications (MISTA 2005)*, Seiten 413–427, New York, USA, 18.–21. Juli, 2005.
- [4] ABHISHEK, K., S. LEYFFER und J. T. LINDEROTH: *FilMINT: An Outer-Approximation-Based Solver for Nonlinear Mixed Integer Programs*. Technischer Bericht Preprint ANL/MCS-P1374-0906, Argonne National Laboratory, Mathematics and Computer Science Division, September, 2006.
- [5] ABLAY, P.: *Optimieren mit Evolutionsstrategien*. Doktorarbeit, Universität Heidelberg, Fachbereich Wirtschafts- und Sozialwissenschaften, 1997.
- [6] AICKELIN, U. und K. A. DOWSLAND: *Exploiting Problem Structure in a Genetic Algorithm Approach to a Nurse Rostering Problem*. *Journal of Scheduling*, 3(3):139–153, 2000.
- [7] AICKELIN, U. und J. LI: *An Estimation of Distribution Algorithm for Nurse Scheduling*. *Annals of Operations Research*, 155(1):289–309, 2007.
- [8] AL-KAZEMI, B. und C. K. MOHAN: *Multi-phase Discrete Particle Swarm Optimization*. In: *Proceedings of the 4th International Workshop on Frontiers in Evolutionary Algorithms*, Seiten 622–625, Triangle Park, North Carolina, USA, 2002.
- [9] ALFARES, H. K.: *Optimum Workforce Scheduling Under the (14, 21) Days-Off Timetable*. *Journal of Applied Mathematics and Decision Sciences*, 6(3):191–199, 2002.

- [10] APPELBLAD, A. S. und S. LÖNN: *A Study of Workforce Management*. Diplomarbeit, Department of Informatics, Göteborg University, 2004.
- [11] ARTHUR, J. L. und A. RAVINDRAN: *A Multiple Objective Nurse Scheduling Model*. AIIE Transactions, 13(1):55–60, 1988.
- [12] ATLASON, J., M. A. EPELMAN und S. G. HENDERSON: *Call center staffing with simulation and cutting plane methods*. Annals of Operations Research, 127:333–358, 2004.
- [13] ATOSS SOFTWARE AG (Herausgeber): *ATOSS Staff Efficiency Suite 5 & ATOSS Startup Edition. Referenzhandbuch*. München, 2009.
- [14] ATOSS SOFTWARE AG und FH HEIDELBERG (Herausgeber): *Standort Deutschland 2006. Zukunftssicherung durch intelligentes Personalmanagement*. München: Maisberger, 2006.
- [15] AUF’M HOFE, H. M.: *Optimierung mit Constraintverfahren*. Technischer Bericht SIEDA-T-2000/1, SIEDA GmbH, 2000.
- [16] AUF’M HOFE, H. M. und E. TOLZMANN: *Global Constraints with Fuzzy Extensions*. KI - Künstliche Intelligenz, Themenheft Constraints, 11(1):37–40, 1997.
- [17] AWAD, R. und J. CHINNECK: *Proctor Assignment at Carleton University*. Interfaces, 28(2):58–71, 1998.
- [18] AZARMI, N. und W. ABDUL-HAMEED: *Workforce scheduling with constraint logic programming*. BT Technology Journal, 13(1):81–94, Januar, 1995.
- [19] AZARMI, N. und R. SMITH: *Intelligent scheduling and planning systems for telecommunications resource management*. BT Technology Journal, 25(3–4):241–248, 2007.
- [20] BAILEY, R. N., K. M. GARNER und M. F. HOBBS: *Using simulated annealing and genetic algorithms to solve staff-scheduling problems*. Asia-Pacific Journal of Operational Research, 14(2):27–43, 1997.
- [21] BAKER, E., L. BODIN, W. FINNEGAN und R. PONDER: *Efficient Heuristic Solutions to an Airline Crew Scheduling Problem*. IIE Transactions, 11(2):79–85, 1979.
- [22] BAKER, S.: *Applying simulated annealing to the workforce management problem*. ISR, British Telecom Laboratories, Martlesham Heath, Ipswich, 1993.

- [23] BALEV, S. und O. GACI: *A greedy MIP-based heuristic approach to the problem of scheduling technicians and interventions for telecommunications*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [24] BARTHOLDI, J. J.: *A Guaranteed-Accuracy Round-off Algorithm for Cyclic Scheduling and Set Covering*. *Operations Research*, 29(3):501–510, 1981.
- [25] BATTITI, R. und G. TECCHIOLLI: *The Reactive Tabu Search*. *ORSA Journal on Computing*, 6(2):126–140, 1994.
- [26] BÄCK, T.: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.
- [27] BÄCK, T. und H. P. SCHWEFEL: *An Overview of Evolutionary Algorithms for Parameter Optimization*. *Evolutionary Computation*, 1(1):1–23, 1993.
- [28] BECK, J. C., P. PROSSER und E. SELENSKY: *Vehicle Routing and Job Shop Scheduling: What's the Difference?* In: *Proceedings of the 13th International Conference on Automated Planning and Scheduling (ICAPS 2003)*, Seiten 267–276, Trenton, Italien, 9.–13. Juni, 2003.
- [29] BENI, G.: *From Swarm Intelligence to Swarm Robotics*. In: ŞAHIN, E. und W. M. SPEARS (Herausgeber): *Swarm Robotics. SAB 2004 International Workshop, Revised Selected Papers*, Band 3342 der Reihe *LNCS*, Seiten 1–9. Berlin, Heidelberg: Springer, Santa Monica, USA, Juli 2004, 2005.
- [30] BEYER, H.G. und H. P. SCHWEFEL: *Evolution strategies: a comprehensive introduction*. *Natural Computing*, 1:3–52, 2002.
- [31] BHULAI, S., G. M. KOOLE und S. A. POT: *Appendix: Simple Methods for Shift Scheduling in Multi-Skill Call Centers*, 2007. <http://www.math.vu.nl/~sapot/call-centers/ShiftAppendix.pdf>, letzter Abruf: 09. April 2010.
- [32] BHULAI, S., G. M. KOOLE und S. A. POT: *Simple Methods for Shift Scheduling in Multi-Skill Call Centers*. *Manufacturing & Service Operations Management*, 10(3):411–420, Summer, 2008.
- [33] BIETHAHN, J., A. HÖHNERLOH, J. KUHL und V. NISSEN: *Methoden der praktischen Entscheidungsfindung*. Göttingen: Georg-August-Universität Göttingen Institut für Wirtschaftsinformatik, 4. Auflage, 2000.
- [34] BIETHAHN, J., A. LACKNER, M. RANGE und O. B. BRODERSEN: *Optimierung und Simulation*. München, Wien: Oldenbourg Verlag, 2004.

- [35] BIRNSTIEL, R.: *Optimierung der untertägigen Personaleinsatzplanung mit Evolutionsstrategien und Vergleich mit der Particle Swarm Optimization*. Diplomarbeit, Technischen Universität Ilmenau, Fakultät für Wirtschaftswissenschaften, Fachgebiet Wirtschaftsinformatik II, 2008.
- [36] BLACKWELL, T. M. und P. BENTLEY: *Don't push me! Collision-avoiding swarms*. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, Band 2, Seiten 1691–1696, Honolulu, USA, 12.–17. Mai, 2002.
- [37] BLÖCHLIGER, I.: *Modelling Staff Scheduling Problems. A Tutorial*. *European Journal of Operational Research*, 158(3):533–542, 2004.
- [38] BOIVIN, S., A. ZANARINI und G. PESANT: *Heuristique constructive guidée par relaxations de problèmes d'ordonnancement*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [39] BONABEAU, E., M. DORIGO und G. THERAULAZ: *Swarm Intelligence. From Natural to Artificial Systems*. Santa Fe Institute. Studies in the Science of Complexity. New York: Oxford University Press, 1999.
- [40] BONAMIA, P., L. T. BIEGLER, A. R. CONNA, G. CORNUÉJOLSC, I. E. GROSSMANN, C. D. LAIRDE, J. LEEA, A. LODIF, F. MARGOTC, N. SAWAYAB und A. WÄCHTER: *An algorithmic framework for convex mixed integer nonlinear programs*. *Discrete Optimization*, 5(2):186–204, Mai, 2008.
- [41] BORTZ, J. und N. DÖRING: *Forschungsmethoden und Evaluation. für Human- und Sozialwissenschaftler*. Berlin: Springer, 4. überarb. Auflage, 2006.
- [42] BORTZ, J. und R. WEBER: *Statistik. für Human- und Sozialwissenschaftler*. Berlin: Springer, 6. Auflage, 2005.
- [43] BORYCZKA, M.-A.: *Génération d'emploi du temps de façon évolutive dans le cadre du challenge FRANCORO V / ROADEF 2007*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [44] BRITISH TELECOM LABORATORIES: *250-118 data set of WMS problem*, 2002. <http://dces.essex.ac.uk/CSP/WFS>, letzter Abruf: 09. April 2010.
- [45] BRODER, S.: *Final examination scheduling*. *Communications of the ACM*, 7:494–498, 1964.

- [46] BRODERSEN, O. B.: *Eignung schwarmintelligenter Verfahren für die betriebliche Entscheidungsunterstützung*. Cuvillier, 2008.
- [47] BRODERSEN, O. B. und M. SCHUMANN: *Einsatz der Particle Swarm Optimization zur Optimierung universitärer Stundenpläne*. Technischer Bericht 05/2007, Georg-August-Universität Göttingen, 2007.
- [48] BRODERSEN, O. B. und M. SCHUMANN: *Optimizing a Stochastic Warehouse using Particle Swarm Optimization*. In: *Proceedings of the Second International Conference on Innovative Computing, Information and Control (ICICIC 2007)*, Seiten 449–449, Kumamoto, Japan, 5.–7. September, 2007.
- [49] BROWN, M. B. und A. B. FORSYTHE: *Robust tests for equality of variances*. Journal of the American Statistical Association, 69(346):364–367, 1974.
- [50] BRUCKER, P., E. K. BURKE, T. CURTOIS, R. QU und G. VANDEN BERGHE: *Adaptive Construction of Nurse Schedules: A Shift Sequence Based Approach*. Technical Report NOTTCS-TR-2007-1, School of Computer Science and Information Technology, University of Nottingham, Jubilee Campus, Juni, 2007.
- [51] BRUCKER, P., E. K. BURKE, T. CURTOIS, R. QU und G. VANDEN BERGHE: *A Scatter Search for the Nurse Rostering Problem*. Technical Report NOTTCS-TR-2007-7, School of Computer Science and Information Technology, University of Nottingham, Jubilee Campus, 2007.
- [52] BRUCKER, P., E. K. BURKE, T. CURTOIS, R. QU und G. VANDEN BERGHE: *A Time Pre-defined Variable Depth Search for Nurse Rostering*. Technical Report NOTTCS-TR-2007-6, School of Computer Science and Information Technology, University of Nottingham, Jubilee Campus, 2007.
- [53] BRUSCO, M. J. und L. W. JACOBS: *A Simulated Annealing Approach to the Cyclic Staff-Scheduling Problem*. Naval Research Logistics, 40(1):69–84, 1993.
- [54] BRUSCO, M. J. und L. W. JACOBS: *A Simulated Annealing Approach to the Solution of Flexible Labour Scheduling Problems*. Journal of the Operational Research Society, 44:1191–1200, 1993.
- [55] BÜTTCHER, S.: *Entwurf und Implementierung eines Genetischen Algorithmus zur Erstellung von Zeitplänen*. Studienarbeit, Institut für Informatik, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2002.
- [56] BURKE, E. K., T. CURTOIS, G. POST, R. QU und B. VELTMAN: *A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem*.

Technical Report NOTTCS-TR-2005-9, School of Computer Science and Information Technology, University of Nottingham, Jubilee Campus, 2005.

- [57] BURKE, E. K., P. DE CAUSMAECKER, S. PETROVIC und G. VANDEN BERGHE: *Variable neighborhood search for nurse rostering problems*. In: RESENDE, M. G. C. und J. P. DE SOUSA (Herausgeber): *Metaheuristics: Computer Decision-Making*, Band 86 der Reihe *Applied Optimization*, Seiten 153–172. Springer, 2004.
- [58] BURKE, E. K., P. DE CAUSMAECKER und G. VANDEN BERGHE: *A Hybrid Tabu Search Algorithm for the Nurse Rostering Problem*. In: *Selected papers from the Second Asia-Pacific Conference on Simulated Evolution and Learning on Simulated Evolution and Learning*, Seiten 187–194, 24.–27. November, 1998.
- [59] BURKE, E. K., J. H. KINGSTON und P. W. PEPPER: *A standard Data Format for Timetabling Instances*. In: BURKE, E. K. und M. CARTER (Herausgeber): *Practice and Theory of Automated Timetabling II, Second International Conference (PATAT 1997)*, Band 1408 der Reihe *LNCS*, Seiten 213–222. Berlin u.a.: Springer, Toronto, Kanada, 20.–22. August 1997, Selected Papers, 1998.
- [60] BURKE, E. K., J. LI und R. QU: *A Hybrid Model of Integer Programming and Variable Neighbourhood Search for Highly-constrained Nurses Rostering Problems*. *European Journal of Operational Research*, 203(2):484–493, 2010.
- [61] BURKE, E. K. und E. SOUBEIGA: *A Real-World Workforce Scheduling Problem in the Hospitality Industry*. In: *Theoretical Models and Algorithmic Methods. Workshop on Real-Life Applications of Metaheuristics*, 2003.
- [62] BUSSIECK, M. R., A. S. DRUD und A. MEERAUS: *MINLPLib - A Collection of Test Models for Mixed-Integer Nonlinear Programming*. *INFORMS Journal on Computing*, 15(1):114–119, 2003.
- [63] BUSSIECK, M. R. und A. PRUESSNER: *Mixed-Integer Nonlinear Programming*. *SIAG/OPT Newsletter: Views & News*, 14(1):19–22, 19. Februar, 2003.
- [64] CAO, B. und G. UEBE: *Solving Transportation Problems with Nonlinear Side Constraints with Tabu Search*. *Computers & Operations Research*, 22(6):593–603, 1995.
- [65] CAVIQUE, L., C. REGO und I. THEMIDO: *Subgraph ejection chains and tabu search for the crew scheduling problem*. *Journal of the Operational Research Society*, 50:608–616, 1999.

- [66] CERNY, V.: *A thermodynamical approach to the travelling salesman problem. An efficient simulated annealing algorithm.* Journal of Optimization Theory and Applications, 45:41–51, 1985.
- [67] CEZIK, M. T. und P. L'ECUYER: *Staffing multi-skill call centers via linear programming and simulation.* Management Science, 54(2):310–323, Februar, 2008.
- [68] CHIU, D. K. W., S. C. CHEUNG und H.-F. LEUNG: *A Multi-Agent Infrastructure for Mobile Workforce Management in a Service Oriented Enterprise.* In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005)*, Seite 85c, Big Island, USA, 3.–6. Januar, 2005.
- [69] CHU, S.-C., Y.-T. CHEN und J.-H. HO: *Timetable Scheduling Using Particle Swarm Optimization.* In: *Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC 2006)*, Band 3, Seiten 324–327, Beijing, 30. August – 1. September, 2006.
- [70] CLEARWATER, S. C., B. A. HUBERMAN und T. HOGG: *Cooperative Solution of Constraint Satisfaction Problems.* Science Magazine, 254(5035):1181–1183, 1991.
- [71] CLERC, M. und J. KENNEDY: *The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space.* In: *IEEE Transaction on Evolutionary Computation*, Band 6(1), Seiten 58–73. Piscataway: IEEE Neural Networks Council, 2002.
- [72] COHEN, J.: *Statistical Power Analysis for the Behavioral Sciences.* Lawrence Erlbaum, 2. Auflage, 1988.
- [73] COIN-OR FOUNDATION INC.: *COmputational INfrastructure for Operations Research Home Page*, 2009. <http://www.coin-or.org>, letzter Abruf: 09. April 2010.
- [74] COLE, A. J.: *The preparation of examination timetables using a small store computer.* Computer Journal, 7:117–121, 1964.
- [75] COLLINS, J. E. und E. M. SISLEY: *Automated Assignment and Scheduling of Service Personnel.* IEEE Expert: Intelligent Systems and Their Applications archive, 9(2):33–39, April, 1994.
- [76] CONSOLI, S., JOSE A. MORENO PÉREZ, K. DARBY-DOWMAN und N. MLADENOVIC: *Discrete Particle Swarm Optimization for the Minimum Labelling Steiner Tree Problem.* In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, Band 129 der Reihe *Studies in Computational Intelligence*, Seiten 313–322. Springer, Acireale, Italien, 8.–10. November, 2008.

- [77] CONTINENZA, B.: *Darwin. Ein Leben für die Evolutionstheorie*. Nummer 1 in *Spektrum der Wissenschaft Highlights*. Heidelberg: Spektrum der Wissenschaft Verlag, 2009.
- [78] COOK, S. A.: *The Complexity of Theorem Proving Procedures*. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, Seiten 151–158, Shaker Heights, USA, 3.–5. Mai, 1971.
- [79] COOPER, T. B. und J. H. KINGSTON: *The Complexity of Timetable Construction Problems*. Technischer Bericht 495, University of Sydney, Basser Department of Computer Science, Februar, 1995.
- [80] COOPER, T. B. und J. H. KINGSTON: *The Complexity of Timetable Construction Problems*. In: BURKE, E. K. und P. ROSS (Herausgeber): *Practice and Theory of Automated Timetabling, First International Conference (PATAT 1995)*, Band 1153 der Reihe *LNCS*, Seiten 283–295. Berlin u.a.: Springer, Edinburgh, U.K., 29. August – 1. September 1995, Selected Papers, 1996.
- [81] CORDEAU, J.-F., G. LAPORTE, F. PASIN und S. ROPKE: *Roadef 2007 Challenge*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [82] CORMEN, T. H., C. E. LEISERSON, R. L. RIVEST und C. STEIN: *Introduction to Algorithms*. Cambridge, London: MIT Press, 2. Auflage, 2001.
- [83] COWLING, P., G. KENDALL und E. SOUBEIGA: *A Parameter-Free Hyperheuristic for Scheduling a Sales Summit*. In: *Proceedings of the 4th Metaheuristics International Conference (MIC 2001)*, Seiten 127–131, Porto, Portugal, 16.–20. Juli, 2001.
- [84] CROWDER, H. P., R. S. DEMBO und J. M. MULVEY: *On Reporting Computational Experiments with Mathematical Software*. *ACM Transactions on Mathematical Software*, 5(2):193–203, 1979.
- [85] CROWDER, H. P., R. S. DEMBO und J. M. MULVEY: *Reporting computational experiments in mathematical programming*. *Mathematical Programming*, 15(1):316–329, 1987.
- [86] CUMMING, A. und B. PAECHTER: *Seminar: Standard Timetabling Data Format*. In: *Practice and Theory of Automated Timetabling, First International Conference (PATAT 1995)*, Edinburgh, U.K., 29. August – 1. September, 1995.

- [87] CUNG, V.-D. und O. BRIANT: *Challenge ROADEF 2007. Technicians and Interventions Scheduling for Telecommunications*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007>, letzter Abruf: 09. April 2010.
- [88] CURTOIS, T.: *Personnel Scheduling Data Sets and Benchmarks*, 2010. <http://www.cs.nott.ac.uk/tec/NRP>, letzter Abruf: 09. April 2010.
- [89] CZOGALLA, J. und A. FINK: *Particle Swarm Topologies for Resource Constrained Project Scheduling*. In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2008)*, Band 236/2009, Seiten 61–73. Berlin: Springer, 2009.
- [90] DAKIN, R. J.: *A tree-search algorithm for mixed integer programming problems*. The Computer Journal, 8(3):250–255, 1965.
- [91] DARWIN, C.: *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, Band 1. London: John Murray, 1859.
- [92] DE CAUSMAECKER, P., P. DEMEESTER, G. VANDEN BERGHE und B. VERBEKE: *An Agent Based Algorithm for personnel scheduling*. In: *7th Workshop on Models and Algorithms for Planning and Scheduling Problems (MAPSP 2005)*, Seiten 103–105., Sienna, Italien, 6.–10. Juni, 2005.
- [93] DE CAUSMAECKER, P., D. OUELHADJ und G. VANDEN BERGHE: *Agents in Timetabling Problems*. In: *Proceedings of 1st Multidisciplinary International Conference on Scheduling (MISTA 2003): Practice and Theory*, Seiten 67–71, Nottingham, U.K., 13.–16. August, 2003.
- [94] DE JONG, K.: *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Doktorarbeit, University of Michigan, Ann Arbor, 1975.
- [95] DECKER, K. und L. JINJIANG: *Coordinated hospital patient scheduling*. Artificial Intelligence, 49:104–111, 1998.
- [96] DI GASPERO, L.: *Local Search Techniques for Scheduling Problems: Algorithms and Software Tools*. Doktorarbeit, Università degli Studi di Udine, Dipartimento di Matematica e Informatica, Dottorato di Ricerca in Informatica, 2003.
- [97] DI GASPERO, L. und A. SCHAERF: *Tabu Search Techniques for Examination Timetabling*. In: BURKE, E. K. und W. ERBEN (Herausgeber): *Practice and Theory of Automated Timetabling III, Third International Conference (PATAT 2000)*, Band 2079 der Reihe LNCS, Seiten 104–117. Berlin u.a.: Springer, Konstanz, Deutschland, 16.–18. August 2000, Selected Papers, 2001.

- [98] DIJKSTRA, E. W.: *A note on two problems in connexion with graphs*. Numerische Mathematik, 1:269–271, 1959.
- [99] DINCIBAS, M., P. VAN HENTENRYCK, H. SIMONIS, A. AGGOUN und A. HEROLD: *The CHIP system: Constraint handling in Prolog*. In: *9th International Conference on Automated Deduction*, Band 310 der Reihe LNCS, Seiten 774–775. Berlin, Heidelberg: Springer, 1988.
- [100] DOTÚ, I., Á. DEL VAL und P. VAN HENTENRYCK: *Local search for the ROADEF 2007 challenge*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [101] DOWLING, D., M. KRISHNAMOORTHY, H. MAC KENZIE und D. SIER: *Staff Rostering at a Large International Airport*. Annals of Operations Research, 72(0):125–147, 1997.
- [102] DOWSLAND, K. A.: *Nurse Scheduling with Tabu Search and Strategic Oscillation*. European Journal of Operational Research, 106(2–3):393–407, 1998.
- [103] DOWSLAND, K. A. und J. M. THOMPSON: *Solving a Nurse Scheduling Problem with Knapsacks, Networks and Tabu Search*. Journal of the Operational Research Society, 51:825–833, 2000.
- [104] DRUMM, H. J.: *Personalplanung*. In: GAUGLER, E. und W. WEBER (Herausgeber): *Handwörterbuch des Personalwesens*, Band 2, Seiten 1758–1769. Stuttgart: Poeschel Verlag, 1992.
- [105] DRUMM, H. J.: *Personalwirtschaft*, Band 6. Berlin, Heidelberg.: Springer, 2008.
- [106] DRUMM, H. J. und C. SCHOLZ: *Personalplanung – Planungsmethoden und Methodenakzeptanz*. Bern: Paul Haupt Verlag, 1988.
- [107] DUBBEN, H.-H. und H.-P. BECK-BORNHOLDT: *Was ist Power und warum ausgerechnet 80%?* Medizinische Klinik, 94, Supplement II:5–7, 1999.
- [108] DURAN, M. und I. E. GROSSMANN: *An outer-approximation algorithm for a class of mixed-integer nonlinear programs*. Mathematical Programming, 36:307–339, 1986.
- [109] DUTOT, P.-F., A. LAUGIER und A.-M. BUSTOS: *Technicians and Interterventions scheduling for Telecommunications*, 2006. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/en/sujet/sujet2.pdf>, letzter Abruf: 09. April 2010.

- [110] EASTON, F. F. und N. MANSOUR: *A Distributed Genetic Algorithm for Employee Staffing and Scheduling Problems*. In: *Proceedings of the 5th International Conference on Genetic Algorithms (ICGA 1993)*, Seiten 360–367, Urbana-Champaign, USA, 17.–21. Juli, 1993.
- [111] EASTON, F. F. und D. F. ROSSIN: *Equivalent Alternate Solutions for the Tour Scheduling Problem*. *Decision Sciences*, 22:985–1007, 1991.
- [112] EBERHART, R. C., P. K. SIMPSON und R. C. DOBBINS: *Computational Intelligence PC Tools*. Boston: Academic Press Professional, 1996.
- [113] EMDEN-WEINERT, T. und M. PROKSCH: *Best Practice Simulated Annealing for the Airline Crew Scheduling Problem*. *Journal of Heuristics*, 5(4):419–436, 1999.
- [114] ERNST, A. T., H. JIANG, M. KRISHNAMOORTHY, B. OWENS und D. SIER: *An Annotated Bibliography of Personnel Scheduling and Rostering*. *Annals of Operations Research*, 127:21–144, 2004.
- [115] ERNST, A. T., H. JIANG, M. KRISHNAMOORTHY und D. SIER: *Staff Scheduling and Rostering: A Review of Applications, Methods, and Models*. *European Journal of Operational Research*, 153(1):3–27, Februar, 2004.
- [116] ESTELLON, B., F. GARDI und K. ET NOUIOUA: *Une heuristique à base de recherche locale pour la planification de tâches avec affectation de ressources*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [117] FAUL, F., E. ERDFELDER, A.-G. LANG und A. BUCHNER: *G*Power 3. A flexible statistical power analysis program for the social, behavioral, and biomedical sciences*. *Behavior Research Methods*, 39:175–191, 2007.
- [118] FEALKO, D. R.: *Evaluating Particle Swarm Intelligence Techniques for Solving University Examination Timetabling Problems*. Diplomarbeit, Nova Southeastern University, USA, 2005.
- [119] FELDMANN, H.-W.: *Workforce Productivity. Ganzheitliche Optimierungsstrategien für Human Resources*. Fürth: Hirschenverlag, 2006.
- [120] FLETCHER, R. und S. LEYFFER: *Solving mixed integer nonlinear programs by outer approximation*. *Mathematical Programming*, 66:327–349, 1994.
- [121] FLOUDAS, C. A.: *Nonlinear and mixed-integer optimization: fundamentals and applications*. New York: Oxford University Press, 1995.

- [122] FOGEL, L. J., A. J. OWENS und M. J. WALSH: *Artificial Intelligence through Simulated Evolution*. John Wiley, 1966.
- [123] FOX, M. S. und S. F. SMITH: *ISIS: A Knowledge-Based System for Factory Scheduling*. Expert Systems Journal, 1(1):25–49, 1984.
- [124] FOXLEY, E. und K. LOCKYER: *The construction of examination timetables by computer*. The Computer Journal, 11:264–268, 1968.
- [125] FREE SOFTWARE FOUNDATION, INC.: *GNU Linear Programming Kit. Reference Manual. Version 4.38 (DRAFT, May 2009)*, 2009. <http://ftp.gnu.org/gnu/glpk>, letzter Abruf: 09. April 2010.
- [126] FREESE, T.: *Ein generisches Ablaufplanungssystem auf Basis von Constraints*. Diplomarbeit, Universität Oldenburg, 1998.
- [127] FRENCH, S.: *Sequencing and Scheduling. An Introduction to the Mathematics of the Job-Shop*. New York: Ellis Horwood Ltd., 1982.
- [128] FRÜHWIRTH, T. und S. ABDENNADHER: *Constraint-programmierung: Grundlagen und Anwendungen*. Berlin, u.a.: Springer, 1997.
- [129] FUKUYAMA, Y. und FUJI ELECTRIC CO. R&D: *Fundamentals of Particle Swarm Optimization Techniques*. In: LEE, K. Y. und M. A. EL-SHARKAWI (Herausgeber): *Modern Heuristic Optimization Techniques with Applications to Power Systems*, Seiten 24–51. New York: John Wiley, 2003.
- [130] FUKUYAMA, Y. und H. YOSHIDA: *A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power Systems*. In: *Proceedings of the 2001 IEEE Congress on Evolutionary Computation (CEC 2001)*, Band 1, Seiten 87–93. IEEE Computer Society Press, Seoul, Südkorea, 27.–30. Mai, 2001.
- [131] GAMS DEVELOPMENT CORPORATION: *BDMLP*, 2009. <http://www.gams.com/dd/docs/solvers/bdmlp.pdf>.
- [132] GAMS DEVELOPMENT CORPORATION: *Gams Home Page*, 2009. <http://www.gams.com>, letzter Abruf: 09. April 2010.
- [133] GANS, N., G. KOOLE und A. MANDELBAUM: *Telephone Call Centers: Tutorial, Review, and Research Prospects*. Manufacturing & Service Operations Management, 5(2):79–14, 2003.

- [134] GARCÍA, F. J. M. und J. A. M. PÉREZ: *Jumping Frogs Optimization: a new swarm method for discrete optimization*. Technical Report DEIOC 03/2008, Universidad de La Laguna, 2008.
- [135] GAREY, M. R. und D. S. JOHNSON: *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Series of Books in the Mathematical Sciences. Freeman, 1979.
- [136] GARWOOD, G. J.: *Work Manager*. BT Technology Journal, 15(1):58–68, 1997.
- [137] GAVRANOVIČ, H.: *One compact and one column generation model for the problem of technicians and interventions in telecom*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [138] GEBHARDT, B.: *Branchenspezifische Aspekte des Handels*. In: FANK, M. und B. SCHERF (Herausgeber): *Handbuch Personaleinsatzplanung*, Seiten 225–261. Frechen: Datakontext, 1. Auflage, 2005.
- [139] GENDREAU, M.: *An Introduction to Tabu Search*. In: GLOVER, F. und G. A. KOCHENBERGER (Herausgeber): *Handbook of Metaheuristics*, Seiten 185–218. Boston: Kluwer Verlag, 2003.
- [140] GERSHMAN, A., A. GRUBSTEIN, A. MEISELS, L. ROKACH und R. ZIVAN: *Scheduling Meetings by Agents*. In: BURKE, E. K. und M. GENDREAU (Herausgeber): *Practice and Theory of Automated Timetabling VII, Seventh International Conference (PATAT 2008)*, Seite HC3, Montréal, Kanada, 18.–22. August, 2008.
- [141] GILAT, D., A. LANDAU, A. RIBAK, Y. SHILOACH und S. WASSERKRUG: *SWOPS (Shift Work Optimized Planning and Scheduling)*. In: BURKE, E. K. und H. RUDOVÁ (Herausgeber): *Practice and Theory of Automated Timetabling VI, Sixth International Conference (PATAT 2006)*, Band 3867 der Reihe LNCS, Seiten 518–523. Berlin u.a.: Springer, Brno, Tschechien, 30. August – 1. September 2006, Revised Selected Papers, 2006.
- [142] GLOVER, F.: *Future paths for integer programming and links to artificial intelligence*. Computers & Operations Research, 13:533–549, 1986.
- [143] GLOVER, F. und M. LAGUNA: *Tabu Search*. Boston: Kluwer Academic Publishers, 1979.
- [144] GÜNEŞ, E. D.: *Workforce Scheduling*. Technical Report TR-06533, Department of Industrial Engineering, Bilkent University, Ankara, Türkei, 1999.

- [145] GÜNTHER, M.: *Untersuchungen zur Anwendbarkeit der Particle Swarm Optimization auf Management-Applikationen am Beispiel eines Lagerhaltungsproblems*. Diplomarbeit, Georg-August-Universität Göttingen, 2005.
- [146] GÜNTHER, M.: *Ein XML-Format für die Personaleinsatzplanung inkl. untertägiger Personaleinsatzplanung*. Technischer Bericht, Technischen Universität Ilmenau, Fakultät für Wirtschaftswissenschaften, Fachgebiet Wirtschaftsinformatik II, 2008.
- [147] GÜNTHER, M.: *A Practical Application for Demand Driven Workforce Management in Retailing*. In: EDELKAMP, S., J. SAUER und B. SCHATTENBERG (Herausgeber): *Proceedings of the 22th Workshop Planen, Scheduling und Konfigurieren, Entwerfen (PuK 2008)*, Seiten 135–151, Kaiserslautern, Deutschland, 23. September, 2008.
- [148] GÜNTHER, M.: *Bedarfsorientierter Personaleinsatz. Gewappnet gegen die mageren Jahre*. IT & Production. Zeitschrift für erfolgreiche Produktion, 9/2009:40–42, 2009.
- [149] GÜNTHER, M.: *Hochflexibles Workforce Management in der Logistik*. In: *LOGISTIK inside. Online-Fachbeitrag, erschienen am 27.03.2009*. Springer Transport Media, 2009.
- [150] GÜNTHER, M.: *Workforce Management in der Logistik*. In: PRADEL, U.-H., W. SÜSSENGUTH, J. PIONTEK und A. F. SCHWOLGIN (Herausgeber): *Praxishandbuch Logistik. Erfolgreiche Logistik in Industrie, Handel und Dienstleistungsunternehmen*, Kapitel 14.6. Köln: Dt. Wirtschaftsdienst, 2009.
- [151] GÜNTHER, M.: *Mehr Flexibilität und Transparenz beim Personaleinsatz*. Unternehmen & Trends: Sonderausgabe zur CeBIT, 1/2010:12–13, 2010.
- [152] GÜNTHER, M.: *Mehr Flexibilität und Transparenz beim Personaleinsatz*. OEM & Lieferant, 1/2010:85–86, 2010.
- [153] GÜNTHER, M.: *Sub-Daily Staff Scheduling Data Sets and Benchmarks*, 2010. <http://www.tu-ilmenau.de/en/wid/research/test-data-sub-daily-staff-scheduling>, letzter Abruf: 26. Juni 2010.
- [154] GÜNTHER, M.: *Workforce Management - automatische Planerstellung auf Knopfdruck*. HR-Performance, 2/2010:13–17, April, 2010.
- [155] GÜNTHER, M.: *Zufriedene Kunden und mehr Umsatz*. rt. retail technology. EHI-Magazin für IT & Logistik im Handel, 2:34–35, 2010.

- [156] GÜNTHER, M.: *Workforce Management an einem Praxisbeispiel aus der Produktion. Einführung einer Softwarelösung bei der Allgaier Werke GmbH.* In: SCHUMANN, M., L. M. KOLBE, M. H. BREITNER und A. FRERICHS (Herausgeber): *Multikonferenz Wirtschaftsinformatik 2010 (MKWI 2010)*, Seiten 2321–2332. Göttingen: Universitätsverlag Göttingen, Göttingen (Deutschland), 23.–25. Februar, 2010 (CD).
- [157] GÜNTHER, M. und E. JÄGER: *Hochflexibles Workforce Management - weniger Kosten, mehr Produktivität. Mitarbeiterpotenzial richtig nutzen und optimieren.* ew das magazin für die energiewirtschaft, 108(5):50–53, März, 2009.
- [158] GÜNTHER, M. und E. JÄGER: *Workforce Management für Energieversorger. Effizienz steigern, Kosten senken.* Energiewirtschaftliche Tagesfragen. Zeitschrift für Energiewirtschaft, Recht und Umwelt, 10(59):49–51, 2009.
- [159] GÜNTHER, M. und V. NISSEN: *Automatic Generation of Optimized Working Time Models in Personnel Planning.* In: *Book of Abstracts of the XXIII European Conference on Operational Research (EURO XXIII)*, Seite 262, Bonn, Deutschland, 5.–8. Juli, 2009.
- [160] GÜNTHER, M. und V. NISSEN: *A Comparison of Metaheuristics on a Practical Staff Scheduling Problem.* In: ALTHOFF, K.-D., K. BACH und M. REICHLER (Herausgeber): *KI 2009. 32nd Annual Conference on Artificial Intelligence. Workshop Proceedings (online)*, Seiten 261–272, Paderborn, Deutschland, 15.–18. September, 2009.
- [161] GÜNTHER, M. und V. NISSEN: *A Comparison of Neighbourhood Topologies for Staff Scheduling with Particle Swarm Optimisation.* In: MERTSCHING, B., M. HUND und A. ZAHEER (Herausgeber): *KI 2009: Advances in Artificial Intelligence, 32nd Annual Conference on AI*, Band 5803 der Reihe *LNAI*, Seiten 185–192. Springer, Paderborn, Deutschland, 15.–18. September, 2009.
- [162] GÜNTHER, M. und V. NISSEN: *Combined Working Time Model Generation and Personnel Scheduling.* In: DANGELMAIER, W., A. BLECKEN, R. DELIUS und S. KLÖPFER (Herausgeber): *8th International Heinz Nixdorf Symposium. Changing Paradigms: Advanced Manufacturing and Sustainable Logistics*, Band 46 der Reihe *LNBIP*, Seiten 210–221. Berlin: Springer, Paderborn, Deutschland, 21.–22. April, 2010.
- [163] GÜNTHER, M. und V. NISSEN: *A Comparison of Three Heuristics on a Practical Case of Sub-Daily Staff Scheduling.* In: MC COLLUM, B., E. BURKE und B. WHITE (Herausgeber): *Practice and Theory of Automated Timetabling, Eighth International Conference (PATAT 2010)*, Seiten 224–240, Belfast, Irland, 10.–13. August, 2010.

- [164] GÜNTHER, M. und V. NISSEN: *Particle Swarm Optimization and an Agent-Based Algorithm for a Problem of Staff Scheduling*. In: DI CHIO, C. (Herausgeber): *Proceedings EvoApplications 2010*, Band 6025 der Reihe LNCS, Seiten 451–461. Berlin: Springer, Istanbul, Türkei, 7.–9. April, 2010.
- [165] GÜNTHER, M. und V. NISSEN: *Sub-Daily Staff Scheduling for a Logistics Service Provider*. KI - Künstliche Intelligenz, 25:105–113, 2010.
- [166] GÜNTHER, M. und T. ZIMMERMAN: *Workforce-Management: Durch bedarfsorientierten Personaleinsatz die Kosten senken und die Effizienz steigern*. In: ENERGY-RELATIONS GMBH (Herausgeber): *Tagungsband Personalforum Energie 2009. 4. Fachkongress für Personalmanagement und -entwicklung in der Energiewirtschaft*, Seiten 73–88, Köln, Deutschland, 9.–10. November, 2009.
- [167] GOMORY, R. E.: *An Algorithm for Integer Solutions to Linear Programs*. In: GRAVES, R. L. und P. WOLFE (Herausgeber): *Recent Advances in Mathematical Programming*, Seiten 269–302. New York, San Francisco: McGraw-Hill, 1963.
- [168] GREENBERG, H. J.: *Computational Testing: Why, How and How Much*. ORSA Journal on Computing, 2(1):94–97, 1990.
- [169] GROSSMAN, I. E.: *Mixed-Integer Nonlinear Programming Techniques for the Synthesis of Engineering Systems*. Research in Engineering Design, 1(3–4):205–228, 1990.
- [170] GÖTZ, L.-O.: *Rechnergestützte Integration von Schichtplanung und operativer Personaleinsatzsteuerung*. Aachen: Shaker Verlag, 1998.
- [171] GUPTA, O. K. und V. RAVINDRAN: *Branch and bound experiments in convex nonlinear integer programming*. Management Science, 31:1533–1546, 1985.
- [172] GUTMANN, J. und J. TERSCHÜREN: *Personalplanung. Wie Sie Mitarbeiter richtig einsetzen*. Planegg b. München : Haufe, 2004.
- [173] HAMDAN, S. A.: *Hybrid Particle Swarm Optimiser using multi-neighborhood topologies*. INFOCOMP Journal of Computer Science, 7(1):36–44, 2008.
- [174] HANSEN, P.: *The steepest ascent mildest descent heuristic for combinatorial programming*. In: *Congress on Numerical Methods in Combinatorial Optimization*, Capri, Italien, 1986.
- [175] HANSEN, P. und N. MLADENović: *Variable Neighborhood Search*. In: GLOVER, F. und G. A. KOCHENBERGER (Herausgeber): *Handbook of Metaheuristics*, Seiten 145–184. Boston: Kluwer Verlag, 2003.

- [176] HANSEN, P. und N. MLADENović: *Variable Neighborhood Search*. In: BURKE, E. K. und G. KENDALL (Herausgeber): *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, Seiten 211–238. Berlin: Springer, 2005.
- [177] HASHIMOTO, H., S. BOUSSIER und M. VASQUEZ: *An iterated local search algorithm for technicians and interventions scheduling for telecommunications*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [178] HELBIG, T.: *Optimierung der untertägigen Personaleinsatzplanung unter Anwendung einer Heuristik*. Diplomarbeit, Technischen Universität Ilmenau, Fakultät für Wirtschaftswissenschaften, Fachgebiet Wirtschaftsinformatik II, 2009.
- [179] HENDTLASS, T. und T. RODGERS: *Discrete Evaluation and the Particle Swarm Algorithm*. In: *Proceedings of the Swarm Fest 2004*, Michigan, USA, 9.–11. Mai, 2004.
- [180] HENS, T. und P. PAMINI: *Grundzüge der analytischen Mikroökonomie*. Berlin, Heidelberg: Springer, 2008.
- [181] HEPPNER, F. und U. GREANDER: *A Stochastic Nonlinear Model for Coordinated Bird Flocks*. In: KRASNER, S. (Herausgeber): *The Ubiquity of Chaos*, Seiten 233–238. Washington: AAAS Publications, 1990.
- [182] HERDY, M.: *Application of the 'Evolutionstrategie' to Discrete Optimization Problems*. In: SCHWEFEL, H. P. und R. MÄNNER (Herausgeber): *Parallel Problem Solving from Nature*, Seiten 188–192. Springer, 1990.
- [183] HESSELN, S. und G. ZANDER: *Wesentliche Funktionen der Personaleinsatzplanung*. In: FANK, M. und B. SCHERF (Herausgeber): *Handbuch Personaleinsatzplanung*, Seiten 119–148. Frechen: Datakontext, 1. Auflage, 2005.
- [184] HÜFTLE, M.: *Ganzzahlige und gemischt-ganzzahlige Optimierung*, 2006. <http://www.ivh.uni-hannover.de/optiv/download.htm>, letzter Abruf: 09. April 2010.
- [185] HOFSTEDT, P. und A. WOLF: *Einführung in die constraint-programmierung: Grundlagen, Methoden, Sprachen*. Berlin, Heidelberg, New York: Springer, 2007.
- [186] HOLLAND, J. H.: *A new kind of turnpike theorem*. Bulletin of the American Mathematical Society, 75(6):1311–1317, 1969.
- [187] HOLLAND, J. H.: *Genetic algorithms and the optimal allocation of trials*. SIAM Journal on Computing, 752(2):88–105, 1973.

- [188] HOLLAND, J. H.: *Adaptation in natural and artificial systems*. Cambridge: MIT Press, 1992.
- [189] HOMBERGER, J.: *Eine Evolutionsstrategie für das Standardproblem der Tourenplanung mit Zeitfensterrestriktionen*. In: *Operations Research Proceedings 1997*, Seite 545 ff. Berlin u.a.: Springer Verlag, 1998.
- [190] HOMBERGER, J.: *Zwei Evolutionsstrategien für das Standardproblem der Tourenplanung mit Zeitfensterrestriktionen*. In: BIETHAHN, J., A. HÖNERLOH, J. KUHL, M-C. LEISEWITZ, V. NISSEN und M. TIETZE (Herausgeber): *Betriebswirtschaftliche Anwendungen des Softcomputing. Neuronale Netze, Fuzzy Logic und Evolutionäre Algorithmen*. Wiesbaden: Vieweg, 1998.
- [191] HOOKER, J. N.: *Testing Heuristics: We Have It All Wrong*. Journal of Heuristics, 1:33–42, 1995.
- [192] HORSCH, J.: *Personalplanung*. Herne, Berlin: Verlag Neue Wirtschafts-Briefe, 2000.
- [193] HUHSE, J.: *Selbstorganisierende Evolutionsstrategien*. Berlin: TENA, 2003.
- [194] HUI, C. K.: *Knowledge-Based Approach to Roster Scheduling Problems*. Diplomarbeit, University of Hong Kong, 1988.
- [195] HURKENS, C. A. J.: *Incorporating the strength of MIP modeling in schedule construction*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [196] JAMPEL, M.: *Constraint (Logic) Programming: A Bibliography*. Technischer Bericht, Dept of Computer Science, City University, London, 1996.
- [197] JASKOWSKI, W. und S. WASIK: *Efficient Greedy Algorithm with Hill Climbing for Technicians and Interventions Scheduling Problem*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [198] JONES, R.: *Development of an Automated Airline Crew Bid Generation System*. Interface, 19(4):44–51, 1989.
- [199] KALLRATH, J.: *Gemischt-ganzzahlige Optimierung: Modellierung in der Praxis. Mit Fallstudien aus Chemie, Energiewirtschaft, Metallgewerbe, Produktion und Logistik*. Braunschweig, Wiesbaden: Vieweg, 1. Auflage, 2002.
- [200] KALVELAGEN, E.: *An Assignment Problem Based On The Mona Lisa*, 2002.

- [201] KAPLANSKY, E. und A. MEISELS: *Distributed Personnel Scheduling - Negotiation among Scheduling Agents*. Annals of Operations Research, 155(1):227–255, 2007.
- [202] KARP, R. M.: *Reducibility among Combinatorial Problems*. In: MILLER, R. E. und J. W. THATCHER (Herausgeber): *Complexity of Computer Computations*, Seiten 85–103. New York: Plenum Press, 1972.
- [203] KENNEDY, J.: *Optimization Based on Sociocognition*. In: DE CASTRO, L. N. und F. J. VON ZUBEN (Herausgeber): *Recent Developments in Biologically Inspired Computing*, Seiten 235–268. IDEA Group Publishing, 2005.
- [204] KENNEDY, J. und R. C. EBERHART: *Particle Swarm Optimization*. In: *Proceedings of the IEEE International Conference on Neural Networks*, Band 4, Seiten 1942–1948. IEEE Service Center, Perth, Australia, 27. November – 1. Dezember, 1995.
- [205] KENNEDY, J. und R. C. EBERHART: *A discrete binary version of the particle swarm algorithm*. In: *IEEE Conference on Systems, Man, and Cybernetics (SMC 1997)*, Band 5, Seiten 4104–4108, Orlando, USA, 12.–15. Oktober, 1997.
- [206] KENNEDY, J., R. C. EBERHART und Y. SHI: *Swarm Intelligence*. The Morgan Kaufmann Series in Evolutionary Computation. San Francisco: Kaufmann, 2001.
- [207] KENNEDY, J. und R. MENDES: *Population structure and particle swarm performance*. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, Band 2, Seiten 1671–1676, Honolulu, USA, 12.–17. Mai, 2002.
- [208] KENNEDY, J. und W. M. SPEARS: *Matching Algorithms to Problems. An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator*. In: *Proceedings of the 1998 IEEE Congress on Evolutionary Computation (CEC 1998)*, Seiten 78–83. IEEE Service Center, Anchorage, USA, 1998.
- [209] KERN, M., G. ANIM-ANSAH, G. OWUSU und C. VOUDOURIS: *FieldPlan: Tactical Field Force Planning in BT*. In: *Artificial Intelligence in Theory and Practice*, Nummer 217 in *IFIP International Federation for Information Processing*, Seiten 345–354. Boston: Springer, 2006.
- [210] KÄHLER, W.-M.: *Statistische Datenanalyse. Verfahren verstehen und mit SPSS gekonnt einsetzen*. Wiesbaden: Vieweg, 5. verb. und erw. Auflage, 2008.
- [211] KILBY, P.: *The Augmented Regret Heuristic for Staff Scheduling*. In: *Proceedings of the 16th Australian Society for Operations Research (ASOR 2001)*, McLaren Vale, Australien, 2001.

- [212] KINGSTON, J. H.: *Modelling timetabling problems with STTL*. In: BURKE, E. K. und W. ERBEN (Herausgeber): *Practice and Theory of Automated Timetabling III, Third International Conference (PATAT 2000)*, Band 2079 der Reihe LNCS, Seiten 309–321. Berlin u.a.: Springer, Konstanz, Deutschland, 16.–18. August 2000, Selected Papers, 2001.
- [213] KIRKPATRICK, S., C. D. GELATT und M. P. VECCHI: *Optimization by simulated annealing*. *Science*, 220:671–680, 1983.
- [214] KIRN, S., O. HERZOG und P. LOCKEMANN (Herausgeber): *Multiagent Engineering. Theory and Applications in Enterprises*. International Handbooks on Information Systems. Berlin: Springer, 2006.
- [215] KLIEM, R. L. und H. B. ANDERSON: *Styles and Their Impact on Project Management Results*. *Project Management Journal*, 27(1):41–50, 1996.
- [216] KLINKERT, A.: *Days-Off Scheduling in Large-Scale Multi-Skill Staff Rostering: An Integer Programming Solution*. In: BURKE, E. K. und M. GENDREAU (Herausgeber): *Practice and Theory of Automated Timetabling VII, Seventh International Conference (PATAT 2008)*, Seite HA3, Montréal, Kanada, 18.–22. August, 2008.
- [217] KOKKORAS, F.: *D-WMS: A Distributed Version of a CLP-based Workforce Management System*. Diplomarbeit, Computer Science Department, University of Bristol, UK, 1996.
- [218] KOKKORAS, F. und S. GREGORY: *D-WMS: Distributed Workforce Management Using CLP*. In: *Proceedings of the 4th International Conference on the Practical Application of Constraint Technology (PACT 1998)*, Seiten 129–146, London, UK, März, 1998.
- [219] KOOPMAN, B. O.: *The Theory of Search, Part III. The Optimum Distribution of Searching Effort*. *Operations Research*, 5:613–626, 1957.
- [220] KORTEWEG, P.: *When to hire the A-Team*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [221] KOST, B.: *Optimierung mit Evolutionsstrategien: eine Einführung in Methodik und Praxis mit Visualisierungsprogrammen*, Band 1. Frankfurt am Main: Deutsch, 2003.
- [222] KOSTREVA, M. und K. JENNINGS: *Nurse Scheduling on a Microcomputer*. *Computers & Operations Research*, 18(8):731–739, 1991.

- [223] KOSTREVA, M., M. LESZCZYNSKI und F. PASSINI: *The Nurse Scheduling Decision via Mixed-Integer Programming*. In: *Proceedings of the American Hospital Association Forum on Nurse Scheduling*, Seiten 281–305, 1978.
- [224] KOZA, J. R.: *Hierarchical genetic algorithms operating on populations of computer programs*. In: SRIDHARAN, N. S. (Herausgeber): *Proc. of the 11th Joint Conf. on Genetic Algorithm*, Seiten 786–774. San Francisco: Morgan Kaufmann Publishers, 1989.
- [225] KOZA, J. R.: *Genetic programming: on the programming of computers by means of natural selection*. Cambridge: MIT Press, 1992.
- [226] KPMG (Herausgeber): *Trends im Handel 2010*. Köln, 2006.
- [227] KRAGELUND, L. und T. KABEL: *Employee Timetabling. An Empirical Study of Solving Real Life Multi-Objective Combinatorial Optimisation Problems by means of Constraint-Based Heuristic Search Methods*. Master's Thesis in Computer Science, Department of Computer Science, University of Aarhus, Dänemark, 1998.
- [228] KREMPELS, K.-H.: *Lösen von Scheduling-Konflikten durch Verhandlungen zwischen Agenten*. In: SAUER, J. (Herausgeber): *Proceedings of the 16th Workshop Planen, Scheduling und Konfigurieren, Entwerfen (PuK 2002)*, Freiburg, Deutschland, 10.–11. Oktober, 2002.
- [229] KRINK, T., J. S. VESTERSTROM und J. RIGET: *Particle swarm optimization with spatial particle extension*. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, Band 2, Seiten 1474–1479, Honolulu, USA, 12.–17. Mai, 2002.
- [230] KROON, L., M. SALOMON und L. VAN WASSENHOVE: *Exact and Approximation Algorithms for the Tactical Fixed Interval Scheduling Problem*. *Operations Research*, 45(4):624–638, 1997.
- [231] KRUSKAL, J.: *On the shortest spanning subtree and the traveling salesman problem*. *Proceedings of the American Mathematical Society*, 7:48–50, 1956.
- [232] LASTUSILTA, T., M. R. BUSSIECK und T. WESTERLUND: *An Experimental Study of the GAMS/AlphaECP MINLP Solver*. *Industrial & Engineering Chemistry Research*, 48(15):7337–7345, 2009.
- [233] LAU, T. L.: *Guided Genetic Algorithm*. Doktorarbeit, Department of Computer Science, University of Essex, 1999.

- [234] LAU, T. L. und E. P. K. TSANG: *The guided genetic algorithm and its application to the general assignment problems*. In: *IEEE 10th International Conference on Tools with Artificial Intelligence (ICTAI'98)*, Seiten 336–343. IEEE Press, Taipei, Taiwan, 10.–12. November, 1998.
- [235] LAU, T. L. und E. P. K. TSANG: *Solving large processor configuration problems with the guided genetic algorithm*. In: *IEEE 10th International Conference on Tools with Artificial Intelligence (ICTAI'98)*, Seiten 320–327. IEEE Press, Taipei, Taiwan, 10.–12. November, 1998.
- [236] LAU, T. L. und E. P. K. TSANG: *Solving the radio link frequency assignment problem with the guided genetic algorithm*. In: *Proceedings of the NATO Symposium on Radio Length Frequency Assignment, Sharing and Conservation Systems (Aerospace)*, Seite 14b, Aalborg, Dänemark, Oktober, 1998.
- [237] LAU, T. L. und E. P. K. TSANG: *Guided genetic algorithm and its application to radio link frequency assignment problems*. *Constraints*, 6(4):373–398, 2001.
- [238] LEE, J.: *Mixed-integer nonlinear programming: Some modeling and solution issues*. *IBM Journal of Research and Development*, 51(3/4):489–497, Mai/Juni, 2007.
- [239] LESAIN, D., N. AZARMI, R. LAITHWAITE und P. WALKER: *Engineering Dynamic Scheduler for Work Manager*. *BT Technology Journal*, 16(3):16–29, 1998.
- [240] LESAIN, D., C. VOUDOURIS und N. AZARMI: *Dynamic Workforce Scheduling for British Telecommunications plc*. *Interfaces*, 30(1):45–56, 2000.
- [241] LESAIN, D., C. VOUDOURIS, N. AZARMI, I. ALLETSON und B. LAITHWAITE: *Field workforce scheduling*. *BT Technology Journal*, 21(4):23–26, 2003.
- [242] LESAIN, D., C. VOUDOURIS, N. AZARMI und B. LAITHWAITE: *Dynamic Workforce Management*. In: *Proceedings of the 1997 IEE Colloquium on AI for Network Management Systems*, Seiten 1/1–1/5. IEE Stevenage, London, UK, 14. April, 1997.
- [243] LEVENE, H.: *Robust Tests for Equality of Variances*. In: OLKIN, I. (Herausgeber): *Contributions to Probability and Statistics*, Seiten 278–292. Palo Alto: Stanford Univ. Press, 1960.
- [244] LEVER, J., M. WALLACE und B. RICHARDS: *Constraint logic programming for scheduling and planning*. *BT Technology Journal*, 13(1):73–80, 1995.
- [245] LI, I.-L.: *Particle Swarm Optimization for Solving Constraint Satisfaction Problems*. Diplomarbeit, Simon Fraser University, Kanada, 2005.

- [246] LIM, T.-S. und W.-Y. LOH: *A comparison of tests of equality of variances*. Computational Statistical & Data Analysis, 22(3):287–301, 1996.
- [247] LINDEROTH, J. T. und T. K. RALPHS: *Noncommercial Software for Mixed-Integer Linear Programming*. In: KARLOF, J. (Herausgeber): *Integer Programming: Theory and Practice*, CRC Press Operations Research Series, Seiten 253–303. CRC Press, 2005.
- [248] LIU, H. und A. ABRAHAM: *An Hybrid Fuzzy Variable Neighborhood Particle Swarm Optimization Algorithm for Solving Quadratic Assignment Problems*. Journal of Universal Computer Science, 13(9):1309–1331, 2007.
- [249] LIU, H., A. ABRAHAM und J. ZHANG: *A Particle Swarm Approach to Quadratic Assignment Problems*. In: SAAD, A., E. AVINERI, K. DAHAL, M. SARFRAZ und R. ROY (Herausgeber): *11th Online World Conference on Soft Computing in Industrial Applications (WSC11). Advances in Soft Computing*, Band 39, Seiten 213–222. Springer, 2007.
- [250] LOURENÇO, H. R., J. P. PAIX AO und R. PORTUGAL: *Metaheuristics for the Bus-Driver Scheduling Problem*. Technischer Bericht, Department of Economics and Management, Universitat Pompeu Fabra, Barcelona, 2001.
- [251] LØVBJERG, M. und T. KRINK: *Extending particle swarm optimisers with self-organized criticality*. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, Band 2, Seiten 1588–1593, Honolulu, USA, 12.–17. Mai, 2002.
- [252] LYKKEGAARD, B.: *Western Europe, HCM and Payroll Applications, Competitive Analysis, 2007*, IDC: Framingham, September, 2007.
- [253] MAENHOUT, B. und M. VANHOUCKE: *Branching Strategies in a Branch-and-Price Approach for a Multiple Objective Nurse Scheduling Problem*. In: *Journal of Scheduling*. Springer, März, 2009.
- [254] MAG, W.: *Einführung in die betriebliche Personalplanung*, Band 2 der Reihe *WiSo-Kurzlehrbücher*. München: Vahlen, 1998.
- [255] MAN, K.-F., K.-S. TANG und S. KWONG: *Genetic Algorithms. Concepts and Designs*, Band 2. London, Berlin, Heidelberg: Springer, 1999.
- [256] MARINAGI, C. C., C. D. SPYROPOULOS, C. PAPTHEODOROU und S. KOKKOTOS: *Continual planning and scheduling for managing patient tests in hospital laboratories*. In: *Artificial Intelligence in Medicine*, Band 20(2), Seiten 139–154. Elsevier, 2000.

- [257] MC CARL, B. A., A. MEERAUS, P. VAN DER EIJK, M. BUSSIECK, S. DIRKSE und P. STEACY: *McCarl GAMS User Guide. Version 23.3*, 2009. <http://www.gams.com/dd/docs/bigdocs/gams2002/mccarlgamsuserguide.pdf>, letzter Abruf: 09. April 2010.
- [258] MEISEL, A. und A. SCHAERF: *Modelling and Solving Employee Timetabling Problems*. *Annals of Mathematics and Artificial Intelligence*, 39(1–2):41–59, 2003.
- [259] MEISELS, A. und E. KAPLANSKY: *Scheduling Agents - Distributed Timetabling Problems (DisTTP)*. In: BURKE, E. K. und P. DE CAUSMAECKER (Herausgeber): *Practice and Theory of Automated Timetabling IV, Fourth International Conference (PATAT 2002)*, Band 2740 der Reihe LNCS, Seiten 166–177. Berlin u.a.: Springer, Gent, Belgien, 21.–23. August 2002, Selected Revisited Papers, 2003.
- [260] MEISELS, A. und N. LUSTERNIK: *Experiments on Networks of Employee Timetabling Problems*. In: BURKE, E. K. und M. CARTER (Herausgeber): *Practice and Theory of Automated Timetabling II, Second International Conference (PATAT 1997)*, Band 1408 der Reihe LNCS, Seiten 130–141. Berlin u.a.: Springer, Toronto, Kanada, 20.–22. August 1997, Selected Papers, 1998.
- [261] MENDES, R.: *Population Topologies and Their Influence in Particle Swarm Performance*. Doktorarbeit, Departamento de Informática, Escola de Engenharia, Universidade do Minho, 2004.
- [262] MENDES, R., J. KENNEDY und J. NEVES: *Avoiding the Pitfalls of Local Optima. How topologies can Save the Day*. In: *Proceedings of the 12th Conference Intelligent Systems Application to Power Systems (ISAP 2003)*. IEEE Computer Society, Lemnos, Griechenland, August, 2003.
- [263] MENDES, R., J. KENNEDY und J. NEVES: *Watch Thy Neighbor Or How The Swarm Can Learn From Its Environment*. In: *Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2003)*, Seiten 88–94. IEEE Computer Society, Indianapolis, USA, 24.–26. April, 2003.
- [264] METRO AG: *METRO Group European Consumption Report*, 2006.
- [265] MIEBACH CONSULTING GMBH (Herausgeber): *Personalbedarfsplanung. Personalplanungs- und Flexibilisierungsmethoden in der logistischen Praxis*. Frankfurt, 2008.
- [266] MILLER, H. E., W. P. PIERSKALLA und G. J. RATH: *Nurse Scheduling Using Mathematical Programming*. *Operations Research*, 24(5):857–870, 1976.

- [267] MILLONAS, M. M.: *Swarms, Phase Transitions and Collective Intelligence*. In: LANGTON, C. G. (Herausgeber): *Artificial Life III. Proceedings of the 3rd International Conference on A-Life*, Band 17 der Reihe *Studies in the Science of Complexity*, Seiten 417–445. Addison-Wesley, 1994.
- [268] MILLS, P. H.: *Extensions to Guided Local Search*. Doktorarbeit, Department of Computer Science, University of Essex, 2002.
- [269] MILLS, P. H. und E. P. K. TSANG: *Guided local search applied to the satisfiability (SAT) problem*. In: *Proceedings of the 15th National Conference of the Australian Society for Operations Research (ASOR'99)*, Seiten 872–883, Queensland, Australien, Juli, 1999.
- [270] MILLS, P. H. und E. P. K. TSANG: *Guided local search for solving SAT and weighted MAX-SAT problems*. *Journal of Automated Reasoning*, Special Issue on Satisfiability Problems, 24:205–223, 2000.
- [271] MILLS, P. H., E. P. K. TSANG und J. FORD: *Applying an Extended Guided Local Search on the Quadratic Assignment Problem*. *Annals of Operations Research*, 118:121–135, 2003.
- [272] MINTON, S., M. D. JOHNSTON, A. B. PHILIPS und P. LAIRD: *Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems*. *Artificial Intelligence*, 58:161–205, 1992.
- [273] MÜLLER, A.: *Evolutionsstrategien*, 2007. <http://www.evocomp.de/themen/evolutionsstrategien/evostrat.html>, letzter Abruf: 09. April 2010.
- [274] MÜLLER, J. P.: *The Design of Intelligent Agents: A Layered Approach*, Band 1177 der Reihe *LNAI*. Berlin u.a.: Springer, 1996.
- [275] MORRIS, P.: *Solutions Without Exhaustive Search: An Iterative Descent Method for Binary Constraint Satisfaction Problems*. In: *AAAI-90 Workshop on Constraint Directed Reasoning Working Notes*, Boston, USA, 1990.
- [276] MORRIS, P.: *The breakout method for escaping from local minima*. In: *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-93)*, Seiten 40–45. AAAI Press, MIT Press, 1993.
- [277] MULLER, C., E. H. MAGILL, P. PROSSER und D. G. SMITH: *Distributed genetic algorithms for resource allocation*. In: DORN, J. und K. FRÖSCHL (Herausgeber): *Scheduling of production processes*, Seiten 70–78. Ellis Horwood, 1993.

- [278] MUSLIU, N.: *Heuristic Methods for Automatic Rotating Workforce Scheduling*. International Journal of Computational Intelligence Research, 2(4):309–326, 2006.
- [279] NAIDU, K. D., K. M. SULLIVAN, P. P. WANG und Y. YI: *Managing Personnel through Staff Scheduling Algorithms*. In: WANG, P. P. (Herausgeber): *Proceedings of the Fifth Joint Conference on Information Sciences 2*, Seiten 829–835, Atlantic City, USA, 27. Februar – 3. März, 2000.
- [280] NANDA, R. und J. BROWNE: *Introduction to Employee Scheduling*. New York: Van Nostrand Reinhold, 1992.
- [281] NAVEH, Y., Y. RICHTER, Y. ALTSHULER, D. L. GRESH und D. P. CONNORS: *Workforce optimization: identification and assignment of professional workers using constraint programming*. IBM Journal of Research and Development, 51(3):263–279, 2007.
- [282] NEEF, A. und K. BURMEISTER: *Swarm Organization. a new paradigm for the Enterprise of the future*. In: KUHLLIN, B. und H. THIELMANN (Herausgeber): *The Practical Real-Time Enterprise. Facts and Perspectives*, Seiten 509–517. Berlin u. a.: Springer, 2005.
- [283] NEMHAUSER, G. L. und L. A. WOLSEY: *Integer and Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. New York: John Wiley & Sons Inc., 1988.
- [284] NISSEN, V.: *Evolutionäre Algorithmen. Darstellung, Beispiele, betriebswirtschaftliche Anwendungsmöglichkeiten*. Wiesbaden: Dt. Univ.-Verl., 1994.
- [285] NISSEN, V.: *Solving the Quadratic Assignment Problem with Clues from Nature*. IEEE Transactions on Neural Networks, 5(1):66–72, 1994.
- [286] NISSEN, V.: *Einführung in Evolutionäre Algorithmen. Optimierung nach dem Vorbild der Evolution*. Reihe Computational Intelligence. Wiesbaden: Vieweg, 1997.
- [287] NISSEN, V. und M. GÜNTHER: *Staff Scheduling With Particle Swarm Optimisation and Evolution Strategies*. In: COTTA, C. und P. COWLING (Herausgeber): *9th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2009)*, Band 5482 der Reihe LNCS, Seiten 228–239. Springer, Tübingen, Deutschland, 15.–17. April, 2009.
- [288] NISSEN, V. und M. GÜNTHER: *Automatic Generation of Optimised Working Time Models in Personnel Planning*. In: GROSS, R. und M. DORIGO (Herausgeber): *Proceedings of the Seventh International Conference on Swarm Intelligence (ANTS*

2010), Band 6234 der Reihe LNCS, Seiten 384–391. Berlin: Springer, Brüssel, Belgien, 8.–10. September, 2010.

- [289] NISSEN, V. und M. GÜNTHER: *Steigerung der Mitarbeiterproduktivität durch automatische Erstellung von Arbeitszeitmodellen in Verbindung mit der Personaleinsatzplanung am Beispiel Handel*. In: BRUHN, M., D. GEORGI und K. HADWICH (Herausgeber): *Forum Dienstleistungsmanagement 2011 - Dienstleistungsproduktivität*, (zur Veröffentlichung angenommen).
- [290] NISSEN, V. und S. GOLD: *Survivable Network Design with an Evolution Strategy*. In: YANG, A., Y. SHAN und L. T. BUI (Herausgeber): *Success in Evolutionary Computation*, Seiten 263–283. Berlin u.a.: Springer, 2008.
- [291] OC&C STRATEGY CONSULTANTS: *Personalkosten im Verkauf. Optimierung ohne Leistungsverzicht*, 2006.
- [292] OLFERT, K.: *Personalwirtschaft*. Leipzig: Friedrich Kiehl Verlag, 2006.
- [293] OR, I.: *Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking*. Doktorarbeit, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, USA, 1976.
- [294] ORTEC LOGIPLAN GMBH: *ORTEC. Resource planning and optimization*, 2009. <http://de.ortec.com>, letzter Abruf: 09. April 2010.
- [295] OSMAN, I. H. und J. P. KELLY: *Meta-Heuristics: An Overview*. In: OSMAN, I. H. und J. P. KELLY (Herausgeber): *Meta-Heuristics: Theory & Applications*, Seiten 1–22. Boston: Kluwer Academic Publishers, 1996.
- [296] OTTO, S.: *Ein agentenbasierter evolutionärer Adaptions- und Optimierungsansatz für verteilte Systeme*. Doktorarbeit, Universität Erlangen-Nürnberg, Technische Fakultät, 2009.
- [297] OWUSU, G., C. VOUDOURIS, R. DORNE, C. LADDE, G. ANIM-ANSAH, K. GASSON und G. CONNOLLY: *ARMS – Application of AI and OR Methods to Resource Management*. BT Technology Journal, 21(4):27–32, 2003.
- [298] PANG, A.: *Competitive Analysis. Analysis of 25 Vendors in the Core HR; eRecruiting; Incentive, Performance, and Workforce Management; and SaaS Segments*, IDC: Framingham, Dezember, 2007.
- [299] PANG, W., K. WANG, C. ZHOU und L. DONG: *Fuzzy discrete particle swarm optimization for solving traveling salesman problem*. In: *Proceedings of the Fourth*

- International Conference on Computer and Information Technology*, Seiten 796–800. IEEE CS Press, 2004.
- [300] PARSOPOULOS, K. E. und M. N. VRAHATIS: *Recent approaches to global optimization problems through Particle Swarm Optimization*. *Natural Computing*, 1(2–3):235–306, 2002.
- [301] PAWLAK, G., S. BAK, P. LICHOCKI und W. MRUCZKIEWICZ: *Heuristic algorithm for problem of Technicians and Interventions Scheduling for Telecommunication*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [302] PFOHL, E.: *Dienstplanung im Krankenhaus durch agentenbasierte elektronische Märkte*. Diplomarbeit, TU Ilmenau, 2000.
- [303] PLANT BIOMECHANICS GROUP FREIBURG: *Evolutionsstrategie*, 2009. http://www.bionik-online.de/mediapool/71/718569/data/Evolutionsstrategie_01_Info.pdf, letzter Abruf: 09. April 2010.
- [304] POHLHEIM, H.: *Evolutionäre Algorithmen – Verfahren, Operatoren, Hinweise aus der Praxis*. Berlin u.a.: Springer-Verlag, 2000.
- [305] POKUTTA, S. und G. STAUFFER: *France Telecom Workforce Scheduling Problem*, 2007. <http://www.g-scop.inpg.fr/ChallengeROADEF2007/index.php?page=6&lang=1>, letzter Abruf: 09. April 2010.
- [306] POLI, R.: *An Analysis of Publications on Particle Swarm Optimization*. Technical Report CSM-469, Department of Computer Science University of Essex, 2007.
- [307] POST, G. und B. VELTMAN: *Harmonious Personnel Scheduling*. In: BURKE, E. K. und M. TRICK (Herausgeber): *Practice and Theory of Automated Timetabling V, Fifth International Conference (PATAT 2004)*, Band 3616 der Reihe LNCS, Seiten 557–559. Berlin u.a.: Springer, Pittsburgh, USA, 18.–20. August 2004, Revisited Selected Papers, 2004.
- [308] POT, S. A., S. BHULAI und G. M. KOOLE: *A Simple Staffing Method for Multi-Skill Call Centers*. *Manufacturing & Service Operations Management*, 10(3):421–428, Summer, 2008.
- [309] PRESTWICH, S.: *ElipSys Programming Tutorial*. Technischer Bericht ECRC-93-2, European Computer-Industry Research Centra (ECRC), 1993.
- [310] PÉREZ, J. A. M., J. P. C. GUTIÉRREZ, F. J. M. GARCÍA, B. MELIÁN, J. M. MORENO-VEGA und J. RAMOS: *Discrete Particle Swarm Optimization for the p-*

median problem. In: *Proceedings of the 7th Metaheuristics International Conference (MIC 2007)*, Montréal, Kanada, 2007.

- [311] PRIM, R. C.: *Shortest connection networks and some generalisations*. Bell System Technical Journal, 36:1389–1401, 1957.
- [312] PRÜM, H.: *Entwicklung von Algorithmen zur Personaleinsatzplanung mittels ganzzahliger linearer Optimierung. Untersuchung eines bedarfsorientierten Modells mit flexibler Schichtkonfiguration*. Diplomarbeit, FH Trier, FB Informatik, April, 2006.
- [313] PROKSCH, M.: *Simulated Annealing und seine Anwendung auf das Crew-Scheduling-Problem*. Diplomarbeit, Humboldt-Universität zu Berlin, Institut für Informatik, Lehrstuhl für Algorithmen und Komplexität, 1997.
- [314] PROUDFOOT CONSULTING: *Produktivitätsbericht 2007 - Die Suche nach versteckten Werten in großen Unternehmen*. Technical report, Oktober, 2007.
- [315] PUPPE, F., F. KLÜGL, R. HERRLER, S. KIRN und C. HEINE: *Konzeption einer flexiblen Agentenkomponente für Schedulingaufgaben im Krankenhausumfeld*. In: *Proceedings des 2. Kolloquiums des Schwerpunktprogrammes "Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien"*, 2000.
- [316] QUESADA, I. und I. E. GROSSMANN: *An LP/NLP based branch and bound algorithm for convex MINLP optimization problems*. Computers & Chemical Engineering, 16(10–11):937–947, Oktober–November, 1992.
- [317] RANSON, D. und S. AHMADI: *An Extensible Modelling Framework for Timetabling Problems*. In: BURKE, E. K. und H. RUDOVÁ (Herausgeber): *Practice and Theory of Automated Timetabling VI, Sixth International Conference (PATAT 2006)*, Band 3867 der Reihe LNCS, Seiten 383–393. Berlin u.a.: Springer, Brno, Tschechien, 30. August – 1. September 2006, Revised Selected Papers, 2006.
- [318] RECHENBERG, I.: *Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog, 1973.
- [319] RECHENBERG, I.: *Evolutionsstrategie '94*. Stuttgart: Frommann-Holzboog, 1994.
- [320] RECHENBERG, I.: *Auf dem Weg zur nichtlinearen Theorie. Korridormodell, Kugelmodell und die 1/5-Erfolgsregel*. In: *Vorlesung Evolutionsstrategie I*, Kapitel 4. TU Berlin, WS 09/10, 2009.
- [321] RECHENBERG, I.: *Das Evolutionsfenster*, 2009. <http://www.bionik.tu-berlin.de/institut/s2evfen.html>, letzter Abruf: 09. April 2010.

- [322] RECHENBERG, I.: *Multimodale Optimierung*, 2009. <http://www.bionik.tu-berlin.de/institut/s2mulmo.html>, letzter Abruf: 09. April 2010.
- [323] REINELT, G.: *TSPLIB - A Traveling Salesman Problem Library*. *ORSA Journal on Computing*, 3:376–384, 1991.
- [324] REIS, L. P. und E. OLIVEIRA: *A Language for Specifying Complete Timetabling Problems*. In: BURKE, E. K. und W. ERBEN (Herausgeber): *Practice and Theory of Automated Timetabling III, Third International Conference (PATAT 2000)*, Band 2079 der Reihe *LNCS*, Seiten 322–341. Berlin u.a.: Springer, Konstanz, Deutschland, 16.–18. August 2000, Selected Papers, 2001.
- [325] REYNOLDS, C. W.: *Flocks, Herds and Schools. A Distributed Behavioral Model*. In: STONE, M. C. (Herausgeber): *ACM SIGGRAPH '87 Conference Proceedings*, Band 21(4), Seiten 25–34. New York: ACM Press, Anaheim, USA, Juli, 1987.
- [326] REYNOLDS, C. W.: *Boids. Background and Update*, 2001. <http://www.red3d.com/cwr/boids>, letzter Abruf: 09. April 2010.
- [327] RIDLEY, M.: *Evolution*, Band 3. Malden: Blackwell, 2004.
- [328] ROSENTHAL, R. E.: *GAMS – A User's Guide*, 2008. <http://www.gams.com/dd/docs/gams/Tutorial.pdf>, letzter Abruf: 09. April 2010.
- [329] RUDOLPH, G.: *An Evolutionary Algorithm for Integer Programming*. In: DAVIDOR, Y., H.-P. SCHWEFEL und R. MÄNNER (Herausgeber): *Parallel Problem Solving from Nature - PPSN III. International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature*, Band 866 der Reihe *LNCS*, Seiten 139–148. Berlin: Springer, Jerusalem, Israel, 9.–14. Oktober, 1994.
- [330] SAKELLARIOU, I., F. KOKKORAS und I. VLAHAVAS: *Applying a Distributed CLP Platform to a Workforce Management Problem*. In: *Proceedings of the 12th Conference on Intelligent Systems Application to Power Systems (ISAP 2003)*, Lemnos, Griechenland, August, 2003.
- [331] SANDHU, P.: *The MathML Handbook*. Hingham: Charles River Media, 1. Auflage, 2002.
- [332] SANTOS, C. VITOR, D. H. D. WARREN und R. YANG: *Andorra-I: A Parallel Prolog System that Transparently Exploits both And- and Or-Parallelism*. In: *Proceedings of the 3rd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, Seiten 83–93. ACM Press, April, 1991.

- [333] SAUER, J.: *Intelligente Ablaufplanung in lokalen und verteilten Anwendungsszenarien*, Band 1. Stuttgart, Leipzig, Wiesbaden: Teubner, 2004.
- [334] SAUER, J. und R. SCHUMANN: *Modelling and Solving Workforce Scheduling Problems*. In: SAUER, J., S. EDELKAMP und B. SCHATTENBERG (Herausgeber): *Proceedings of the 21th Workshop Planen, Scheduling und Konfigurieren, Entwerfen (PuK 2007)*, Seiten 93–101, Osnabrück, Deutschland, 10. September, 2007.
- [335] SCHAERF, A. und A. MEISELS: *Solving Employee Timetabling Problems by Generalized Local Search*. In: *The 6th Congress of the Italian Association for Artificial Intelligence (AI*AI 99), Advances in Artificial Intelligence*, Band 1792 der Reihe LNAI, Seiten 380–390, Bologna, Italien, 1999.
- [336] SCHERF, B.: *Personaleinsatzplanung und Arbeitszeitmanagement in der Logistik*. In: PFOHL, H.-C. (Herausgeber): *Personalführung in der Logistik. Innovative Ansätze und praktische Lösungen*, Seiten 216–244. Deutscher Verkehrs-Verlag, 1. Auflage, 2004.
- [337] SCHERF, B.: *Ein Vorgehensmodell zur Personaleinsatzplanung*. In: FANK, M. und B. SCHERF (Herausgeber): *Handbuch Personaleinsatzplanung*, Seiten 15–53. Frechen: Datakontext, 1. Auflage, 2005.
- [338] SCHERF, B.: *Wirtschaftliche Nutzenaspekte der Personaleinsatzplanung*. In: FANK, M. und B. SCHERF (Herausgeber): *Handbuch Personaleinsatzplanung*, Seiten 55–83. Frechen: Datakontext, 1. Auflage, 2005.
- [339] SCHÄFER, J.: *Evolutionsstrategien*. Seminararbeit, Universität Paderborn, Fakultät für Elektrotechnik, Informatik und Mathematik, Institut für Informatik, WS 04/05, 2004.
- [340] SCHINDLER, B., F. ROTHLAUF und E. PESCH: *Evolution strategies, Network Random Keys, and the One-Max Tree Problem*. In: *Applications of Evolutionary Computing: EvoWorkshops 2002*, Band 2279 der Reihe LNCS, Seiten 29–40. Springer, 2002.
- [341] SCHMITT, R.: *Evolutionsstrategien nach Rechenberg und Schwefel. Seminar Evolutionäre Algorithmen/Genetische Programmierung*. Seminararbeit, Fachhochschule Trier, Fachbereich Angewandte Informatik, WS97/98, 1997.
- [342] SCHWEFEL, H.-P.: *Numerical optimization of computer models*. Chichester: John Wiley & Sons, 1981.
- [343] SCHWEFEL, H.-P.: *Collective Phenomena in Evolutionary Systems*. In: CHECKLAND, P. und I. KISS (Herausgeber): *Problems of Constancy and Change – The*

- Complementarity of Systems Approaches to Complexity. Proceedings of the 31st annual Meeting of the International Society for General System Research*, Seiten 1025–1033, Budapest, Juni, 1987.
- [344] SCHWEFEL, H.-P.: *Evolution and Optimum Seeking*. Chichester: Wiley & Sons, 1995.
- [345] SECREST, B. R.: *Traveling Salesman Problem for Surveillance Mission Using Particle Swarm Optimization*. Diplomarbeit, School of Engineering and Management of the Air Force Institute of Technology, Air University, Ohio, USA, 2001.
- [346] SHI, Y.: *Particle Swarm Optimization*. coNNectionS. The newsletter of the IEEE Neural Networks Society, 2(1):8–13, 2004.
- [347] SHI, Y. und R. C. EBERHART: *Parameter Selection in Particle Swarm Optimization*. In: PORTO, V. W., N. SARAVANAN, D. WAAGEN und A. E. EIBEN (Herausgeber): *Evolutionary Programming VII. Proceedings of the Seventh Annual Conference on Evolutionary Programming (EP 1998)*, Band 1447 der Reihe LNCS, Seiten 591–600. Berlin u. a.: Springer, New York, USA, 1998.
- [348] SHI, Y. und R. C. EBERHART: *Empirical Study of Particle Swarm Optimization*. In: *Proceedings of the 1999 IEEE Congress on Evolutionary Computation (CEC 1999)*, Band 3, Seiten 1945–1950, Washington DC, USA, 6.–9. Juli, 1999.
- [349] SIVANANDAM, S. N., P. VISALAKSHI und A. BHUVANESWARI: *Multiprocessor Scheduling Using Hybrid Particle Swarm Optimization with Dynamically Varying Inertia*. International Journal of Computer Science & Applications, 4(3):95–106, 2007.
- [350] SMITH, S. F., P. S. OW, N. MUSCETTOLA, J. Y. POTVIN und D. C. MATHYS: *OPIS: an Opportunistic Factory Scheduling System*. In: *Proceedings of the 3rd International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-90)*, Band 1, Seiten 268–274, Charlston, USA, Mai, 1990.
- [351] SOLC, T.: *Tablix2 file format – looking forward*, 2008. <http://www.tablix.org/~avian/tablix%20format%20-%20looking%20forward.pdf>, letzter Abruf: 09. April 2010.
- [352] SOLC, T.: *Tablix. Free software for solving timetabling problems*, 2009. <http://www.tablix.org>, letzter Abruf: 09. April 2010.
- [353] SOLOMON, M. M.: *Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints*. Operations Research, 35(2):54–265, 1987.

- [354] SONDERGELD, L.: *Performance analysis methods for heuristic search optimization with an application to cooperative agent algorithms*. Dissertation, TU Braunschweig, Braunschweig: Shaker, 2001.
- [355] STATISTISCHES BUNDESAMT (Herausgeber): *Statistisches Jahrbuch 2007. Für die Bundesrepublik Deutschland*. Reutlingen: SFG, 2007.
- [356] STATISTISCHES BUNDESAMT (Herausgeber): *Verdienste und Arbeitskosten 2008. Begleitmaterial zur Pressekonferenz am 13. Mai 2009 in Berlin*. Wiesbaden, 2009.
- [357] STELZER, D.: *Forschungsmethodik in der Wirtschaftsinformatik*. Vortrag, TU Ilmenau, Fakultät für Wirtschaftswissenschaften, Fachgebiet Informations- und Wissensmanagement, 2008.
- [358] STOLLETZ, R.: *operational workforce planning for check-in counters at airports*. Transportation Research Part E: Logistics and Transportation Review, 46(3):414–425, Mai, 2010.
- [359] STONE, L. D.: *The Process of Search Planning: Current Approaches and Continuing Problems*. Operations Research, 31:207–233, 1983.
- [360] STÜTZLE, T. G.: *Lokale Suchverfahren für Constraint Satisfaction Probleme: Die min-conflicts Heuristik und Tabu Search*. Künstliche Intelligenz, 1/97:14–20, 1997.
- [361] STÜTZLE, T. G.: *Local Search in Algorithms for Combinatorial Problems - Analysis, Algorithms, and New Applications*, Band 220 der Reihe *Dissertations in Artificial Intelligence*. Amsterdam: IOS Press, 1999.
- [362] SUGANTHAN, P. N.: *Particle Swarm Optimiser with Neighbourhood Operator*. In: *Proceedings of the 1999 IEEE Congress on Evolutionary Computation (CEC 1999)*, Band 3, Seiten 1958–1962, Washington DC, USA, 6.–9. Juli, 1999.
- [363] SUN, M., J. E. ARONSON, P. G. MC KEOWN und D. DRINKA: *A Tabu Search Heuristic Procedure for the Fixed Charge Transportation Problem*. European Journal of Operational Research, 106(2–3):441–456, 1998.
- [364] TAILLARD, É. D.: *Robust taboo search for the quadratic assignment problem*. Parallel Computing, 17:443–455, 1991.
- [365] TAILLARD, É. D.: *Robust taboo search for the quadratic assignment problem*. Parallel Computing, 17:443–455, 1991.
- [366] TASGETIREN, M. F., Y.-C. LIANG, M. SEVKLI und G. GENCYILMAZ: *A particle swarm optimization algorithm for makespan and total flowtime minimization in*

- the permutation flowshop sequencing problem*. European Journal of Operational Research, 177(3):1930–1947, 2007.
- [367] TASGETIREN, M. F., M. SEVKLI, Y.-C. LIANG und G. GENÇYILMAZ: *Particle swarm optimization algorithm for single machine total weighted tardiness problem*. In: *Proceedings of the Congress on Evolutionary Computation (CEC 2004)*, Band 2, Seiten 1412–1419. IEEE Service Center, Portland, 19.–23. Juni, 2004.
- [368] TASGETIREN, M. F., M. SEVKLI und Y.-C. LIANG und M. M. YENISEY: *Particle Swarm Optimization and Differential Evolution Algorithms for Job Shop Scheduling Problem*. International Journal of Operations Research, 3(2):120–135, 2006.
- [369] TASGETIREN, M. F., P. N. SUGANTHAN und Q.-K. PANG: *A Discrete Particle Swarm Optimization Algorithm for the Generalized Traveling Salesman Problem*. In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation (GECCO 2007)*, Seiten 158–167. ACM Press, London, UK, 7.–11. Juli, 2007.
- [370] THOMPSON, G. M.: *A Simulated-Annealing Heuristic for Shift Scheduling Using Non-Continuously Available Employees*. Computers & Operations Research, 23(3):275–288, 1996.
- [371] THORNTON, J. R.: *An Enhanced Cyclic Descent Algorithm for Nurse Rostering*. Honours Thesis, Faculty of Engineering and Applied Science, Griffith University, 1995.
- [372] THORNTON, J. R. und A. SATTAR: *Nurse Rostering and Integer Programming Revisited*. In: VERMA, B. und X. YAO (Herausgeber): *Proceedings of 1st International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 1997)*, Seiten 49–58, Gold Coast, Australien, 10.–12. Februar, 1997.
- [373] THRUN, S. B.: *Efficient exploration in reinforcement learning*. Technischer Bericht CMU-CS-92-102, School of Computer Science, Carnegie-Mellon University, 1992.
- [374] TIEN, J. M. und A. KAMIYAMA: *Manpower Scheduling Algorithms*. SIAM Review, 24(3):275–287, 1982.
- [375] TOLZMANN, E.: *OC:Planner Automatische Dienstplanung mittels Constraintpropagierung*. Künstliche Intelligenz, 1/2007:53–54, 2007.
- [376] TOREX RETAIL HOLDINGS LIMITED: *Torex – POS & Enterprise Lösungen für Einzelhandel, Gastronomie, Lebensmittel, Tankstellen*, 2010. <http://www.torex.com/de>, letzter Abruf: 09. April 2010.

- [377] TSANG, E. P. K.: *An Annotated Bibliography of Personnel Scheduling and Rostering*. BT Technology Journal, 13(1):16–28, 1995.
- [378] TSANG, E. P. K., J. FORD, P. MILLS, R. WILLIAMS und P. SCOTT: *A Personnel Scheduling System Based on Constraint Programming*. Technical Report CSM-406, University of Essex, Department of Computer Science, Essex, U.K., 2004.
- [379] TSANG, E. P. K., T. GOSLING, B. VIRGINAS, C. VOUDOURIS und G. OWUSU: *Retractable Contract Network for Distributed Scheduling*. In: *Proceedings of the 2nd Multidisciplinary International Conference on Scheduling: Theory & Applications (MISTA 2005)*, Seiten 485–500, New York, USA, 18.–21. Juli, 2005.
- [380] TSANG, E. P. K., T. GOSLING, B. VIRGINAS, C. VOUDOURIS, G. OWUSU und W. LIU: *Retractable contract network for empowerment in workforce scheduling*. Special Issue on Negotiation Mechanisms, Multiagent and Grid Systems, 4(1):25–44, 2008.
- [381] TSANG, E. P. K., P. H. MILLS und J. FORD: *Extending Guided Local Search - Towards a Metaheuristic Algorithm With No Parameters To Tune*. Technischer Bericht CSM-371, University of Essex, Colchester, UK, April, 2002.
- [382] TSANG, E. P. K., B. VIRGINAS, T. GOSLING und W. LIU: *Multi-Agent Systems for Staff Empowerment*. In: VOUDOURIS, C., G. OWUSU, R. DORNE und D. LE-SAINT (Herausgeber): *Service Chain Management. Technology Innovation for the Service Business*, Seiten 236–274. Springer, 2008.
- [383] TSANG, E. P. K. und C. VOUDOURIS: *Fast Local Search and Guided Local Search and Their Application to British Telecom's Workforce Scheduling Problem*. Operations Research Letters, 29(3):119–127, 1997.
- [384] TSANG, E. P. K. und C. J. WANG: *A generic neural network approach for constraint satisfaction problems*. In: TAYLOR, J. G. (Herausgeber): *Neural Network Applications*, Seiten 12–22. Springer, 1992.
- [385] UCAR, H. und M. F. TASGETIREN: *A particle swarm optimization algorithm for permutation flow shop sequencing problem with the number of tardy jobs criterion*. In: *Proceedings of the 5th International Symposium on Intelligent Manufacturing Systems (IMS 06)*, Seiten 1110–1120, Sakarya, Türkei, 29.–31. Mai, 2006.
- [386] VALOUXIS, C. und E. HOUSOS: *Hybrid Optimization Techniques for the Workshift and Rest Assignment of Nursing Personnel*. Artificial Intelligence in Medicine, 20(2):155–175, 2000.

- [387] VAN DEN BERGH, F.: *An Analysis of Particle Swarm Optimizers*. Diplomarbeit, Universität Pretoria, 2002.
- [388] VAN HENTENRYCK, P.: *Constraint satisfaction in logic programming*. Cambridge: MIT Press, 1989.
- [389] VAN HENTENRYCK, P.: *The CLP language CHIP: constraint solving and applications*. In: *Comcon Spring '91. Digest of Papers*, Seiten 382–387. IEEE Computer Society, San Francisco, USA, 25. Februar – 1. März, 1991.
- [390] VAN LAARHOVEN, P. J. M. und E. H. L. AARTS: *Simulated Annealing: Theory and Applications*. Norwell: Kluwer Academic Publishers, 1987.
- [391] VANDEN BERGHE, G.: *An Advanced Model and Novel Meta-heuristic Solution Methods to Personnel Scheduling in Healthcare*. Ph.D. Thesis, University of Gent, Belgien, 2002.
- [392] VANHOUCKE, M. und B. MAENHOUT: *NSPLib - A Nurse Scheduling Problem Library: A tool to evaluate (meta-)heuristic procedures*. In: *Proceedings of the 31st Meeting of the European Working Group on Operational Research Applied to Health Services (ORAHS 2005)*, Southampton, U.K., 31. Juli – 5. August, 2005.
- [393] VEERAMACHANENI, K., L. OSADCIW und G. KAMATH: *Probabilistically driven particle swarms for optimization of multi valued discrete problems: Design and analysis*. In: *Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2007)*, Seiten 141–149, Honolulu, 1.–5. April, 2007.
- [394] VEERAMACHANENI, K., T. PERAM, C. MOHAN und L. A. OSADCIW: *Optimization Using Particle Swarm with Near Neighbor Interactions*. In: GOOS, G., J. HARTMANIS und J. VAN LEEUWEN (Herausgeber): *Genetic and Evolutionary Computation Conference (GECCO-2003)*, Band 2723 der Reihe LNCS, Seiten 110–121. Berlin u. a.: Springer, Chicago, USA, 12.–16. Juli, 2003.
- [395] VENTER, G. und J. SOBIESZCZANSKI-SOBIESKI: *Particle Swarm Optimization*. AIAA Journal 2003, 41(8):1583–1589, 2003.
- [396] VESTERSTRØM, J. S. und J. RIGET: *Particle Swarms. Extensions for improved local, multi-modal, and dynamic search in numerical optimization*. Diplomarbeit, Universität Aarhus, 2002.
- [397] VLASSIS, N.: *A Concise Introduction to Multiagent Systems and Distributed Artificial*. San Rafael: Morgan and Claypool Publishers, 2007.

- [398] VOUDOURIS, C.: *Guided local search for combinatorial optimisation problems*. Doktorarbeit, Department of Computer Science, University of Essex, Colchester, UK, Juli, 1997.
- [399] VOUDOURIS, C.: *Guided Local Search - An illustrative example in function optimisation*. BT Technology Journal, 16(3):46–50, 1998.
- [400] VOUDOURIS, C., G. OWUSU, R. DORNE, C. LADDE und B. VIRGINAS: *ARMS: An automated resource management system for British Telecommunications plc*. European Journal of Operational Research, 171(3):951–961, 2006.
- [401] VOUDOURIS, C. und E. P. K. TSANG: *Function Optimization using Guided Local Search*. Technischer Bericht CSM-249, University of Essex, Colchester, UK, September, 1995.
- [402] VOUDOURIS, C. und E. P. K. TSANG: *Partial Constraint Satisfaction Problems and Guided Local Search*. In: *Proceedings of Practical Application of Constraint Technology (PACT'96)*, Seiten 337–356, London, UK, April, 1996.
- [403] VOUDOURIS, C. und E. P. K. TSANG: *Solving the Radio Link Frequency Assignment Problem using Guided Local Search*. In: *Proceedings of the NATO Symposium on Radio Length Frequency Assignment, Sharing and Conservation Systems (Aerospace)*, Seite 14a, Aalborg, Dänemark, Oktober, 1998.
- [404] VOUDOURIS, C. und E. P. K. TSANG: *Guided Local Search and its application to the Travelling Salesman Problem*. European Journal of Operational Research, 113(2):469–499, 1999.
- [405] VOUDOURIS, C. und E. P. K. TSANG: *Guided local search joins the elite in discrete optimisation*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 57:29–39, 2001.
- [406] VOUDOURIS, C. und E. P. K. TSANG: *Guided local search*. In: GLOVER, F. und G. A. KOCHENBERGER (Herausgeber): *Handbook of Metaheuristics*, Seiten 185–218. Boston: Kluwer Verlag, 2003.
- [407] WALLACE, R. und E. C. FREUDER: *Stable Solutions for Dynamic Constraint Satisfaction Problems*. In: *Fourth Conference on Principles and Practice of Constraint Programming (CP'98)*, Band 1520 der Reihe LNCS, Seiten 447–461. Berlin: Springer, 1998.
- [408] WANG, C. J. und E. P. K. TSANG: *Solving constraint satisfaction problems using neural-networks*. In: *Proceedings of IEE Second International Conference on Artificial Neural Networks*, Seiten 295–299, 1991.

- [409] WARNER, D. M.: *Scheduling Nursing Personnel According to Nursing Preference: A Mathematical Programming Approach*. Operations Research, 24(5):842–856, 1976.
- [410] WATTS, D. J.: *The Dynamics of Networks Between Order and Randomness*. Princeton University Press, 1999.
- [411] WATTS, D. J. und S. H. STROGATZ: *Collective dynamics of small-world networks*. Nature, 393:440–442, Juni, 1998.
- [412] WAUE, V.: *Entwicklung von Software zur Lösung von gemischt-ganzzahligen Optimierungsmodellen mit einem Branch and Cut Ansatz*. Dissertation, FU Berlin, FB Wirtschaftswissenschaft, 2007.
- [413] WEGENER, I.: *Theoretische Informatik – eine algorithmenorientierte Einführung*, Band 3. Wiesbaden: Teubner Verlag, 2005.
- [414] WEICKER, K.: *Evolutionäre Algorithmen*. In: WEICKER, K. (Herausgeber): *Soft-computing – Tagungsband zum ersten Softcomputing-Treffen*, Seiten 27–39, Stuttgart, Deutschland, 1999.
- [415] WEICKER, K.: *Evolutionäre Algorithmen*, Band 2. Wiesbaden: Teubner, 2007.
- [416] WEISS, G.: *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge: MIT Press, 1999.
- [417] WIKIPEDIA: *NP-complete*, 2010. <http://en.wikipedia.org/wiki/NP-complete>, letzter Abruf: 10. Dezember 2010.
- [418] WILDE, T. und T. HESS: *Forschungsmethoden der Wirtschaftsinformatik. Eine empirische Untersuchung*. Wirtschaftsinformatik, 49(4):280–287, 2007.
- [419] WILLIAMS, J. P. und G. DOZIER: *A Comparison of Hill-Climbing Methods for Solving Static and Recurrent Dynamic Constraint Satisfaction Problems*. Technischer Bericht CSSE01-01, Department Computer Science and Software Engineering, Auburn University, Januar, 2001.
- [420] WINTZEN, J. und E. JÄGER: *Branchenspezifische Aspekte Logistik*. In: FANK, M. und B. SCHERF (Herausgeber): *Handbuch Personaleinsatzplanung*, Seiten 343–361. Frechen: Datakontext, 1. Auflage, 2005.
- [421] WOOD, D. C.: *A system for computing university examination timetables*. The Computer Journal, 11:41–47, 1968.

- [422] WOOLDRIDGE, M. J.: *An Introduction to MultiAgent Systems*. Chichester: John Wiley & Sons, 2002.
- [423] WOOLDRIDGE, M. J. und N. R. JENNINGS: *Intelligent agents: Theory and practice*. The Knowledge Engineering Review, 10(2):115–152, 1995.
- [424] WREN, A. und D. O. WREN: *A Genetic Algorithm for Public Transport Driver Scheduling*. Computers & Operations Research, 22(1):101–110, 1995.
- [425] WRIGHT, M.: *Scheduling English Cricket Umpires*. Journal of the Operational Research Society, 42(6):447–452, 1991.
- [426] WÜRTZ, J.: *Lösen kombinatorischer Probleme mit Constraintprogrammierung in Oz*. Doktorarbeit, Universität des Saarlandes, Technische Fakultät, 1998.
- [427] XIE, X. F., W. J. ZHANG und Z. L. YANG: *Dissipative particle swarm optimization*. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, Band 2, Seiten 1456–1461, Honolulu, USA, 12.–17. Mai, 2002.
- [428] YANG, R.: *Solving a Workforce Management Problem with Constraint Programming*. In: *Proceedings of the 2nd International Conference on the Practical Application of Constraint Technology (PACT 1996)*, Seiten 373–387, London, UK, 22.–26. April, 1996.
- [429] YANG, R., T. BEAUMONT und D. MUNAF: *Programming Style Guide for Andorra-I CHIP Extension*. Technischer Bericht, Department of Computer Science, University of Bristol, August, 1993.
- [430] YANG, R., D. INÊS und C. V. SANTOS: *Design of the Andorra-I System*. Technischer Bericht CSTR-91-PEPMA, ESPRIT Project 2471, Department of Computer Science, University of Bristol, Januar, 1991.
- [431] ÖZCAN, E.: *Towards an XML based standard for Timetabling Problems: TTML*. In: KENDALL, G., E. K. BURKE, S. PETROVIC und M. GENDREAU (Herausgeber): *Multidisciplinary Scheduling. Theory and Applications. 1st International Conference (MISTA 2003)*. Berlin u.a.: Springer, Nottingham, U.K., 13.–15. August 2003, Selected Papers, 2005.
- [432] ZHANG, L.-P., H.-J. YU und S.-X. HU: *Optimal choice of parameters for particle swarm optimization*. Journal of Zhejiang University SCIENCE, 6(6):528–534, 2005.

Anhang

7.3 Beispielproblem im XML-Format

```
<?xml version="1.0" encoding="UTF-8"?>
<SchedulingPeriod ID="1" OrganisationID="A">
  <DayPeriods>
    <Date ID="2008-03-03" ShortDesc="D1">
      <Start >08:00:00</Start >
      <End>16:00:00</End>
      <TimeIntervalCount >8</TimeIntervalCount>
      <TimeIntervalLength Unit="min">60</TimeIntervalLength>
    </Date>
  </DayPeriods>

  <Contracts>
    <Contract ID="1" ShortDesc="C1">
      <Label>Standardvertrag</Label>
    </Contract>
  </Contracts>

  <Employees>
    <Employee ID="1" ShortDesc="MA1">
      <Name>Patrick Trampnau</Name>
      <ContractID >1</ContractID>
    </Employee>
  </Employees>

  <Workstations>
    <Workstation ID="1" ShortDesc="AP1">
      <Name>Verkauf</Name>
    </Workstation>
  </Workstations>

  <ShiftTypes>
    <Shift ID="1" ShortDesc="S1">
```

```

    <Colour>#66FF66</Colour>
    <Description>Schicht 1</Description>
    <StartTime>12:00:00</StartTime>
    <EndTime>16:00:00</EndTime>
  </Shift>
</ShiftTypes>

<Assignments>
  <Assignment ID="1">
    <Date>2008-03-03</Date>
    <EmployeeID>1</EmployeeID>
    <ShiftID>1</ShiftID>
    <WorkstationID>1</WorkstationID>
  </Assignment>
</Assignments>

<MasterWeights>
  <PrefOverStaffing Unit="min">1</PrefOverStaffing>
  <PrefUnderStaffing Unit="min">1</PrefUnderStaffing>
</MasterWeights>

<CoverRequirements>
  <DateSpecificCover>
    <Date>2008-03-03</Date>
    <Cover>
      <WorkstationID>1</WorkstationID>
      <Preferred>2,2,2,2,1,1,1,1</Preferred>
    </Cover>
  </DateSpecificCover>
</CoverRequirements>
</SchedulingPeriod>

```


7.4 Ergebnistabellen

Tabelle 7.1: Ergebnisse bei Logistik_Complete (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer und TS* mit 2.800.000)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	∅	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
Manueller Plan	411.330,0	411.330	-	0,0	1.545,0	20.130,0	14.610,0	33.795,0
MAS	51.829,4	51.801	15,4	1.579,0	0,0	7.365,0	28.395,0	7.245,0
HC I1	53.077,2	52.625	196,8	2.061,1	0,0	7.746,0	28.776,0	7.245,0
HC I2	53.500,9	52.909	515,9	1.477,0	0,0	8.236,5	29.266,5	7.245,0
TS I3	116.797,9	85.973	18.923,3	700,4	214,5	10.900,0	16.052,5	23.122,5
TS VNS I3	56.489,5	52.902	4.815,2	1.430,0	14,0	7.727,5	27.473,0	8.529,5
TS* VNS I3	51.767,2	51.704	40,5	1.354,2	0,0	7.422,5	28.414,5	7.283,0
TS I3 + HC3	117.279,7	90.613	21.328,5	689,2	215,0	11.143,0	16.088,5	23.229,5
TS* I3 + HC3	51.649,5	51.555	62,0	1.286,0	0,0	7.415,5	28.433,0	7.257,5
TS I3 + HC3 (F) (-A)	68.902,1	57.765	7.371,9	1.344,6	54,0	8.293,5	24.673,0	11.895,5
TS I3 + HC3 (F) (-A) /T	67.320,7	53.015	9.994,9	1.420,2	60,0	8.288,5	27.515,0	9.048,5
TS* I3 + HC3 (F) (-A) /T	52.463,0	51.591	1.890,3	1.283,0	3,5	7.447,5	28.412,5	7.310,0
PSO (20) I3 gBest	52.162,0	51.967	92,2	1.666,8	0,0	7.478,5	28.488,0	7.265,5
PSO (40) I3 gBest	52.222,3	52.085	79,9	1.730,2	0,0	7.568,6	28.246,5	7.339,4
PSO (100) I3 gBest	52.591,2	52.400	107,7	1.778,5	0,0	7.576,3	28.152,1	7.542,0
PSO (200) I3 gBest	53.726,5	53.467	138,9	2.220,3	0,0	7.658,5	28.017,0	7.916,5
PSO (20) I3 Wheel	71.104,0	70.161	534,0	2.872,8	0,0	10.987,5	21.281,5	17.981,0
PSO (40) I3 Wheel	90.854,0	80.708	7.105,4	2.497,6	48,0	13.520,0	18.353,5	23.441,5
PSO (100) I3 Wheel	173.848,0	137.768	17.943,1	1.648,0	411,0	16.518,5	16.105,5	28.688,0
PSO (200) I3 Wheel	248.082,0	21.9251	15.436,3	1.025,0	757,5	18.083,5	15.243,5	31.115,0
PSO (20) I3 Circle	52.614,0	52.402	105,2	1.925,4	0,0	7.539,5	28.479,5	7.335,0
PSO (40) I3 Circle	52.995,0	52.719	149,4	2.161,0	0,0	7.566,0	28.414,0	7.427,0
PSO (100) I3 Circle	57.841,0	56.714	822,1	2.416,8	0,0	7.831,0	24.618,5	11.487,5
PSO (200) I3 Circle	70.099,0	69.152	539,8	1.949,7	0,0	9.854,0	17.962,5	20.166,5
PSO (20) I3 lBest	52.498,0	52.300	117,2	1.863,0	0,0	7.523,5	28.485,5	7.313,0
PSO (40) I3 lBest	52.776,0	52.627	87,8	2.029,8	0,0	7.551,5	28.458,5	7.368,0
PSO (100) I3 lBest	54.133,0	53.868	138,8	2.353,1	0,0	7.637,5	27.682,5	8.230,0
PSO (200) I3 lBest	63.640,0	62.528	665,4	2.118,6	0,0	8.516,5	20.578,5	16.213,0

Tabelle 7.2: Ergebnisse bei Logistik_Complete (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluierungen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	∅	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
PSO (20) I3 gBest + R	51.781,3	51.763	9,3	1.531,4	0,0	7.365,0	28.395,0	7.245,0
PSO (40) I3 gBest + R	51.805,2	51.783	9,5	1.555,2	0,0	7.365,0	28.395,0	7.245,0
PSO (100) I3 gBest + R	51.825,8	51.811	9,0	1.575,8	0,0	7.365,0	28.395,0	7.245,0
PSO (200) I3 gBest + R	51.841,2	51.817	13,1	1.591,2	0,0	7.365,0	28.395,0	7.245,0
PSO (20) I3 Wheel + R	52.430,4	52.377	26,4	2.172,9	0,0	7.365,0	28.395,0	7.245,0
PSO (40) I3 Wheel + R	52.468,1	52.427	26,7	2.210,2	0,0	7.365,0	28.395,0	7.245,0
PSO (100) I3 Wheel + R	52.511,0	52.438	28,5	2.254,2	0,0	7.365,0	28.395,0	7.245,0
PSO (200) I3 Wheel + R	52.520,5	52.446	35,7	2.261,9	0,0	7.365,0	28.395,0	7.245,0
PSO (20) I3 Circle + R	51.938,8	51.882	34,4	1.682,4	0,0	7.365,0	28.395,0	7.245,0
PSO (40) I3 Circle + R	52.033,4	51.988	26,6	1.778,1	0,0	7.365,0	28.395,0	7.245,0
PSO (100) I3 Circle + R	52.157,0	52.117	20,3	1.901,3	0,0	7.365,0	28.395,0	7.245,0
PSO (200) I3 Circle + R	52.251,1	52.213	17,7	1.996,1	0,0	7.365,0	28.395,0	7.245,0
PSO (20) I3 lBest + R	51.910,9	51.884	14,9	1.655,4	0,0	7.365,0	28.395,0	7.245,0
PSO (40) I3 lBest + R	51.939,9	51.884	29,0	1.684,4	0,0	7.365,0	28.395,0	7.245,0
PSO (100) I3 lBest + R	52.067,1	52.016	20,0	1.812,5	0,0	7.365,0	28.395,0	7.245,0
PSO (200) I3 lBest + R	52.171,2	52.136	14,4	1.916,2	0,0	7.365,0	28.395,0	7.245,0
ES (1,5) I2 TypG	55.987,4	55.545	187,5	1.616,8	0,0	7.994,5	26.163,0	10.106,5
ES (1+5) I2 TypG	55.893,1	55.575	218,4	1.604,2	0,0	7.978,5	26.181,0	10.064,5
ES (10,50) I2 TypG	56.947,6	55.744	1.391,3	1.677,3	0,0	8.093,0	25.560,0	10.808,0
ES (10+50) I2 TypG	56.483,8	55.701	676,5	1.664,8	0,0	8.029,5	25.819,5	10.485,0
ES (30,200) I2 TypG	63.953,3	58.587	7.252,6	1.536,8	0,0	8.999,5	21.132,5	16.142,0
ES (30+200) I2 TypG	63.634,2	58.449	6.744,2	1.531,7	0,0	8.906,0	21.165,5	16.015,5
ES (1,5) I2 TypG + R	51.867,1	51.851	14,1	1.615,1	0,0	7.366,0	28.396,0	7.245,0
ES (1+5) I2 TypG + R	51.863,6	51.839	17,3	1.611,6	0,0	7.366,0	28.396,0	7.245,0
ES (10,50) I2 TypG + R	51.917,8	51.877	21,0	1.665,8	0,0	7.366,0	28.396,0	7.245,0
ES (10+50) I2 TypG + R	51.906,4	51.886	15,4	1.656,4	0,0	7.365,0	28.395,0	7.245,0
ES (30,200) I2 TypG + R	51.924,9	51.904	13,5	1.674,9	0,0	7.365,0	28.395,0	7.245,0
ES (30+200) I2 TypG + R	51.926,1	51.904	16,3	1.674,1	0,0	7.366,0	28.396,0	7.245,0
ES (1,5) I2 TypE	69.521,6	61.052	6.098,7	3.583,9	0,0	11.706,0	24.397,7	14.917,0
ES (1+5) I2 TypE	70.144,8	62.158	3.924,2	3.063,8	0,0	11.899,0	21.980,0	16.601,0
ES (10,50) I2 TypE	53.647,8	53.048	388,2	2.149,4	0,0	7.674,5	28.075,2	7.873,5
ES (10+50) I2 TypE	52.863,7	52.493	343,9	1.919,2	0,0	8.401,5	27.422,3	7.560,5
ES (30,200) I2 TypE	54.471,1	53.954	233,3	2.451,1	0,0	7.725,5	27.706,5	8.294,0
ES (30+200) I2 TypE	53.824,1	53.428	167,0	2.351,1	0,0	7.678,0	28.111,0	7.842,0
ES (1,5) I2 TypE + R	52.486,5	52.090	205,0	2.236,5	0,0	7.365,0	28.395,0	7.245,0
ES (1+5) I2 TypE + R	52.128,9	52.018	55,1	1.878,9	0,0	7.365,0	28.395,0	7.245,0
ES (10,50) I2 TypE + R	51.870,3	51.842	18,1	1.620,3	0,0	7.365,0	28.395,0	7.245,0
ES (10+50) I2 TypE + R	51.842,8	51.816	16,4	1.592,5	0,0	7.365,0	28.395,0	7.245,0
ES (30,200) I2 TypE + R	51.854,5	51.835	13,6	1.604,5	0,0	7.365,0	28.395,0	7.245,0
ES (30+200) I2 TypE + R	51.846,3	51.820	11,7	1.596,3	0,0	7.365,0	28.395,0	7.245,0

Tabelle 7.3: Ergebnisse bei Logistik_Mo (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	∅	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
Manueller Plan	85.185,0	85.185	-	0,0	345,0	2.955,0	2.160,0	5.535,0
MAS	7.727,3	7.722	3,5	227,3	0,0	900,0	4.680,0	960,0
HC I1	7.824,8	7.767	38,5	276,8	0,0	924,0	4.704,0	960,0
HC I2	7.905,5	7.771	93,5	204,5	0,0	1.000,5	4.780,5	960,0
TS I2	7.721,0	7.700	14,1	214,0	0,0	903,5	4.683,5	960,0
TS I3	7.694,2	7.679	14,9	182,2	0,0	906,0	4.686,0	960,0
TS I2 + HC3	7.678,6	7.665	16,0	171,6	0,0	903,5	4.683,5	960,0
TS I3 + HC3	7.681,7	7.665	16,5	168,7	0,0	906,5	4.686,5	960,0
TS I3 + HC8	7.686,2	7.668	19,1	171,2	0,0	907,5	4.687,5	960,0
PSO (20) I2 gBest	7.972,9	7.850	54,1	307,4	0,0	958,5	4.690,0	1.008,5
PSO (40) I2 gBest	8.513,4	8.372	85,5	367,9	0,0	1.066,5	4.534,0	1.272,5
PSO (200) I2 gBest	9.949,0	9.626	184,5	413,5	0,0	1.322,5	3.911,5	2.151,0
PSO (20) I3 gBest	7.782,4	7.706	43,7	212,8	0,0	934,7	4.714,7	960,0
PSO (200) I3 gBest	7.807,0	7.739	45,6	226,0	0,0	940,5	4.720,5	960,0
PSO (20) I3 Wheel	7.968,0	7.896	53,4	319,2	0,0	941,5	4.656,0	1.025,5
PSO (200) I3 Wheel	11.914,0	11.269	714,5	376,0	0,0	1.721,0	3.104,5	3.356,5
PSO (20) I3 Circle	7.746,0	7.701	32,8	215,9	0,0	915,0	4.695,0	960,0
PSO (200) I3 Circle	7.821,0	7.792	18,3	289,7	0,0	913,5	4.689,5	964,0
PSO (20) I3 lBest	7.744,0	7.705	28,2	215,0	0,0	914,5	4.694,5	960,0
PSO (200) I3 lBest	7.765,0	7.732	22,1	244,3	0,0	910,0	4.689,5	960,5
PSO (20) I3 gBest + R	7.712,2	7.704	3,6	212,2	0,0	900,0	4.680,0	960,0
PSO (40) I3 gBest + R	7.717,0	7.711	3,7	217,0	0,0	900,0	4.680,0	960,0
PSO (200) I3 gBest + R	7.727,3	7.720	3,4	227,3	0,0	900,0	4.680,0	960,0
PSO (20) I3 Wheel + R	7.728,5	7.720	3,7	227,4	0,0	900,0	4.680,0	960,0
PSO (200) I3 Wheel + R	7.783,3	7.772	5,3	281,4	0,0	900,0	4.680,0	960,0
PSO (20) I3 Circle + R	7.712,2	7.708	2,7	212,2	0,0	900,0	4.680,0	960,0
PSO (200) I3 Circle + R	7.722,8	7.716	3,0	222,8	0,0	900,0	4.680,0	960,0
PSO (20) I3 lBest + R	7.712,5	7.706	2,9	212,5	0,0	900,0	4.680,0	960,0
PSO (200) I3 lBest + R	7.721,1	7.715	2,3	221,0	0,0	900,0	4.680,0	960,0

Tabelle 7.4: Ergebnisse bei Logistik_Mo (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	⊙	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
ES (1,5) I2 TypG	7.729,5	7.698	24,3	212,6	0,0	908,8	4.688,4	960,4
ES (1+5) I2 TypG	7.728,7	7.690	22,3	194,3	0,0	918,9	4.698,9	960,0
ES (10,50) I2 TypG	7.780,3	7.713	43,1	215,3	0,0	932,5	4.712,5	960,0
ES (10+50) I2 TypG	7.782,0	7.711	44,2	215,9	0,0	933,0	4.713,0	960,0
ES (30,200) I2 TypG	7.808,1	7.752	40,2	225,0	0,0	941,5	4.721,5	960,0
ES (30+200) I2 TypG	7.812,2	7.723	41,6	225,2	0,0	943,5	4.723,5	960,0
ES (1,5) I2 TypG + R	7.707,0	7.702	3,2	207,0	0,0	900,0	4.680,0	960,0
ES (1+5) I2 TypG + R	7.705,7	7.698	2,7	205,7	0,0	900,0	4.680,0	960,0
ES (10,50) I2 TypG + R	7.725,6	7.717	4,5	225,6	0,0	900,0	4.680,0	960,0
ES (10+50) I2 TypG + R	7.724,7	7.716	3,9	224,7	0,0	900,0	4.680,0	960,0
ES (30,200) I2 TypG + R	7.735,6	7.725	5,4	235,6	0,0	900,0	4.680,0	960,0
ES (30+200) I2 TypG + R	7.737,1	7.729	3,7	237,1	0,0	900,0	4.680,0	960,0
ES (1,5) I2 TypE	8.067,5	7.797	374,9	343,0	0,0	985,0	4.710,5	1.014,5
ES (1+5) I2 TypE	8.003,2	7.745	436,5	320,7	0,0	983,5	4.748,0	975,5
ES (10,50) I2 TypE	7.876,1	7.767	126,4	305,6	0,0	934,0	4.711,5	962,5
ES (10+50) I2 TypE	7.861,2	7.762	88,5	297,2	0,0	931,5	4.710,5	961,0
ES (30,200) I2 TypE	7.855,4	7.783	57,9	302,4	0,0	926,0	4.705,0	961,0
ES (30+200) I2 TypE	7.857,3	7.782	38,8	306,8	0,0	925,0	4.704,5	960,5
ES (1,5) I2 TypE + R	7.770,9	7.732	49,2	270,9	0,0	900,0	4.680,0	960,0
ES (1+5) I2 TypE + R	7.750,3	7.726	18,1	250,3	0,0	900,0	4.680,0	960,0
ES (10,50) I2 TypE + R	7.764,6	7.732	18,0	264,6	0,0	900,0	4.680,0	960,0
ES (10+50) I2 TypE + R	7.762,6	7.734	18,1	262,6	0,0	900,0	4.680,0	960,0
ES (30,200) I2 TypE + R	7.775,4	7.749	14,0	275,4	0,0	900,0	4.680,0	960,0
ES (30+200) I2 TypE + R	7.775,1	7.758	13,4	275,1	0,0	900,0	4.680,0	960,0

Tabelle 7.5: Ergebnisse bei Logistik_Di (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	∅	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
Manueller Plan	178.260,0	178.260	-	0,0	825,0	3.420,0	1.680,0	4.080,0
MAS	5.917,3	5.910	4,4	217,3	0,0	1.320,0	2.940,0	720,0
HC I1	6.149,4	5.998	69,9	303,4	0,0	1.393,0	3.013,0	720,0
HC I2	6.175,6	6.039	79,6	217,6	0,0	1.449,0	3.069,0	720,0
TS I2	5.959,8	5.921	31,8	236,8	0,0	1.331,5	2.951,5	720,0
TS I3	5.895,4	5.888	5,9	194,4	0,0	1.320,5	2.940,5	720,0
TS I2 + HC3	5.897,9	5.873	32,3	179,9	0,0	1.329,0	2.949,0	720,0
TS I3 + HC3	5.873,0	5.868	2,1	173,0	0,0	1.320,0	2.940,0	720,0
TS I3 + HC8	5.877,7	5.869	3,8	177,7	0,0	1.320,0	2.940,0	720,0
PSO (20) I2 gBest	6.180,0	6.073	53,0	322,5	0,0	1.361,5	2.907,0	794,5
PSO (40) I2 gBest	6.679,5	6.490	121,2	381,0	0,0	1.475,5	2.808,0	1.007,5
PSO (200) I2 gBest	8.276,2	7.826	566,3	423,7	0,5	1.877,0	2.558,5	1.658,5
PSO (20) I3 gBest	5.917,6	5.904	11,5	213,6	0,0	1.322,0	2.942,0	720,0
PSO (200) I3 gBest	5.944,1	5.917	23,9	231,1	0,0	1.326,5	2.946,5	720,0
PSO (20) I3 Wheel	6.205,0	6.079	65,9	336,1	0,0	1.362,0	2.897,5	804,5
PSO (200) I3 Wheel	12.576,0	10.091	2.646,5	374,6	10,0	2.600,0	2.278,5	2.661,5
PSO (20) I3 Circle	5.926,0	5.908	14,7	224,0	0,0	1.321,0	2.941,0	720,0
PSO (200) I3 Circle	6.013,0	5.976	22,2	293,6	0,0	1.325,5	2.937,5	728,0
PSO (20) I3 lBest	5.928,0	5.913	12,2	224,8	0,0	1.321,5	2.941,5	720,0
PSO (200) I3 lBest	5.953,0	5.932	12,4	248,7	0,0	1.321,5	2.940,5	721,0
PSO (20) I3 gBest + R	5.900,3	5.892	2,8	200,3	0,0	1.320,0	2.940,0	720,0
PSO (40) I3 gBest + R	5.905,2	5.901	2,3	205,2	0,0	1.320,0	2.940,0	720,0
PSO (200) I3 gBest + R	5.918,9	5.914	3,1	218,9	0,0	1.320,0	2.940,0	720,0
PSO (20) I3 Wheel + R	5.915,5	5.907	3,9	214,2	0,0	1.320,0	2.940,0	720,0
PSO (200) I3 Wheel + R	5.973,3	5.962	5,9	271,6	0,0	1.320,0	2.940,0	720,0
PSO (20) I3 Circle + R	5.900,5	5.894	3,1	200,5	0,0	1.320,0	2.940,0	720,0
PSO (200) I3 Circle + R	5.909,2	5.904	3,0	209,2	0,0	1.320,0	2.940,0	720,0
PSO (20) I3 lBest + R	5.900,8	5.894	3,0	200,8	0,0	1.320,0	2.940,0	720,0
PSO (200) I3 lBest + R	5.909,1	5.903	2,8	209,1	0,0	1.320,0	2.940,0	720,0

Tabelle 7.6: Ergebnisse bei Logistik_Di (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	⊙	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
ES (1,5) I2 TypG	5.954,3	5.918	30,3	235,4	0,0	1.330,0	2.946,2	720,4
ES (1+5) I2 TypG	5.900,6	5.891	4,3	200,8	0,0	1.320,0	2.940,0	720,0
ES (10,50) I2 TypG	5.927,9	5.900	21,6	216,9	0,0	1.325,5	2.945,5	720,0
ES (10+50) I2 TypG	5.926,6	5.902	21,3	217,6	0,0	1.324,5	2.944,5	720,0
ES (30,200) I2 TypG	5.939,9	5.915	24,0	224,9	0,0	1.327,5	2.947,5	720,0
ES (30+200) I2 TypG	5.947,4	5.916	22,1	223,4	0,0	1.332,0	2.952,0	720,0
ES (1,5) I2 TypG + R	5.895,5	5.890	2,6	195,5	0,0	1.320,0	2.940,0	720,0
ES (1+5) I2 TypG + R	5.894,3	5.890	2,9	194,3	0,0	1.320,0	2.940,0	720,0
ES (10,50) I2 TypG + R	5.910,5	5.901	3,4	210,5	0,0	1.320,0	2.940,0	720,0
ES (10+50) I2 TypG + R	5.909,0	5.903	3,1	209,0	0,0	1.320,0	2.940,0	720,0
ES (30,200) I2 TypG + R	5.918,4	5.912	3,3	218,4	0,0	1.320,0	2.940,0	720,0
ES (30+200) I2 TypG + R	5.918,7	5.910	3,8	218,7	0,0	1.320,0	2.940,0	720,0
ES (1,5) I2 TypE	6.433,0	6.036	526,6	374,0	0,0	1.467,0	3.022,0	785,0
ES (1+5) I2 TypE	6.303,7	5.953	462,4	336,7	0,0	1.447,5	3.055,5	732,0
ES (10,50) I2 TypE	6.159,5	5.983	120,4	328,5	0,0	1.385,0	3.004,0	721,0
ES (10+50) I2 TypE	6.153,6	6.019	118,7	322,1	0,0	1.384,5	3.002,0	722,5
ES (30,200) I2 TypE	6.141,4	6.032	77,7	325,9	0,0	1.377,0	2.995,5	721,5
ES (30+200) I2 TypE	6.121,7	6.032	57,0	322,2	0,0	1.369,5	2.989,0	720,5
ES (1,5) I2 TypE + R	5.956,2	5.915	38,8	256,2	0,0	1.320,0	2.940,0	720,0
ES (1+5) I2 TypE + R	5.944,8	5.905	29,5	244,8	0,0	1.320,0	2.940,0	720,0
ES (10,50) I2 TypE + R	5.956,5	5.921	21,0	256,5	0,0	1.320,0	2.940,0	720,0
ES (10+50) I2 TypE + R	5.950,7	5.919	16,8	250,7	0,0	1.320,0	2.940,0	720,0
ES (30,200) I2 TypE + R	5.967,8	5.941	12,1	267,8	0,0	1.320,0	2.940,0	720,0
ES (30+200) I2 TypE + R	5.968,7	5.931	15,4	268,7	0,0	1.320,0	2.940,0	720,0

Tabelle 7.7: Ergebnisse bei Logistik_Mi (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	∅	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
Manueller Plan	91.140,0	91.140	-	0,0	375,0	3.810,0	1.890,0	5.220,0
MAS	8.183,2	8.180	1,8	248,2	0,0	1.965,0	4.560,0	705,0
HC I1	8.350,9	8.229	65,5	292,9	0,0	2.026,5	4.621,5	705,0
HC I2	8.395,4	8.305	60,1	225,4	0,0	2.082,5	4.677,5	705,0
TS I2	8.188,2	8.155	33,8	228,2	0,0	1.977,5	4.572,5	705,0
TS I3	8.150,6	8.133	15,2	203,6	0,0	1.971,0	4.566,0	705,0
TS I2 + HC3	8.146,6	8.122	34,5	192,6	0,0	1.974,5	4.569,5	705,0
TS I3 + HC3	8.145,0	8.121	15,3	191,0	0,0	1.974,5	4.569,5	705,0
TS I3 + HC8	8.154,4	8.123	21,4	192,4	0,0	1.978,5	4.573,5	705,0
PSO (20) I2 gBest	8.451,7	8.349	63,4	333,3	0,0	2.017,0	4.532,5	784,5
PSO (40) I2 gBest	9.034,9	8.749	118,7	403,0	0,0	2.105,5	4.284,5	1.121,0
PSO (200) I2 gBest	10.559,4	10.254	192,8	437,5	0,0	2.331,5	3.472,0	2159,5
PSO (20) I3 gBest	8.173,4	8.146	21,5	217,4	0,0	1.975,5	4.570,5	705,0
PSO (200) I3 gBest	8.217,7	8.162	32,7	244,7	0,0	1.984,0	4.579,0	705,0
PSO (20) I3 Wheel	8.485,0	8.348	77,8	344,6	0,0	2.012,0	4.495,5	816,5
PSO (200) I3 Wheel	12.833,0	12.066	961,3	381,9	1,5	2.732,0	2.645,0	3.387,0
PSO (20) I3 Circle	8.194,0	8.154	37,8	227,5	0,0	1.981,0	4.576,0	705,0
PSO (200) I3 Circle	8.279,0	8.245	24,7	315,1	0,0	1.977,5	4.568,5	709,0
PSO (20) I3 lBest	8.194,0	8.157	27,5	230,1	0,0	1.979,5	4.574,5	705,0
PSO (200) I3 lBest	8.215,0	8.186	21,1	263,4	0,0	1.973,5	4.568,5	705,0
PSO (20) I3 gBest + R	8.161,2	8.155	2,7	226,2	0,0	1.965,0	4.560,0	705,0
PSO (40) I3 gBest + R	8.166,0	8.161	2,8	231,0	0,0	1.965,0	4.560,0	705,0
PSO (200) I3 gBest + R	8.177,6	8.172	3,1	242,6	0,0	1.965,0	4.560,0	705,0
PSO (20) I3 Wheel + R	8.180,4	8.167	5,2	244,1	0,0	1.965,0	4.560,0	705,0
PSO (200) I3 Wheel + R	8.243,6	8.231	6,9	307,5	0,0	1.965,0	4.560,0	705,0
PSO (20) I3 Circle + R	8.161,3	8.157	2,6	226,3	0,0	1.965,0	4.560,0	705,0
PSO (200) I3 Circle + R	8.173,2	8.164	2,5	238,1	0,0	1.965,0	4.560,0	705,0
PSO (20) I3 lBest + R	8.162,1	8.157	3,1	227,1	0,0	1.965,0	4.560,0	705,0
PSO (200) I3 lBest + R	8.171,7	8.166	2,9	236,6	0,0	1.965,0	4.560,0	705,0

Tabelle 7.8: Ergebnisse bei Logistik_Mi (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	⊙	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
ES (1,5) I2 TypG	8.185,3	8.155	23,0	218,0	0,0	1.981,2	4.575,9	705,4
ES (1+5) I2 TypG	8.156,6	8.132	20,7	204,2	0,0	1.973,0	4.568,0	705,0
ES (10,50) I2 TypG	8.208,7	8.160	35,0	228,7	0,0	1.987,5	4.582,5	705,0
ES (10+50) I2 TypG	8.203,4	8.158	29,9	229,4	0,0	1.984,5	4.579,5	705,0
ES (30,200) I2 TypG	8.251,6	8.176	43,1	238,9	0,0	2.003,0	4.598,0	705,0
ES (30+200) I2 TypG	8.225,1	8.172	31,3	238,1	0,0	1.991,0	4.586,0	705,0
ES (1,5) I2 TypG + R	8.155,8	8.151	3,0	220,8	0,0	1.965,0	4.560,0	705,0
ES (1+5) I2 TypG + R	8.154,3	8.150	2,2	219,3	0,0	1.965,0	4.560,0	705,0
ES (10,50) I2 TypG + R	8.171,2	8.163	4,2	236,2	0,0	1.965,0	4.560,0	705,0
ES (10+50) I2 TypG + R	8.169,9	8.165	3,1	234,9	0,0	1.965,0	4.560,0	705,0
ES (30,200) I2 TypG + R	8.183,9	8.174	4,5	248,9	0,0	1.965,0	4.560,0	705,0
ES (30+200) I2 TypG + R	8.183,0	8.174	3,8	248,0	0,0	1.965,0	4.560,0	705,0
ES (1,5) I2 TypE	8.548,0	8.275	356,5	369,5	0,0	2.058,5	4.597,0	761,5
ES (1+5) I2 TypE	8.379,9	8.197	199,6	319,4	0,0	2.022,0	4.605,5	716,5
ES (10,50) I2 TypE	8.363,1	8.224	108,4	319,1	0,0	2.018,5	4.611,5	707,0
ES (10+50) I2 TypE	8.326,4	8.216	86,2	314,4	0,0	2.002,5	4.595,5	707,0
ES (30,200) I2 TypE	8.345,0	8.235	62,2	324,0	0,0	2.008,0	4.603,0	705,0
ES (30+200) I2 TypE	8.371,8	8.261	94,5	330,3	0,0	2.017,5	4.611,0	706,5
ES (1,5) I2 TypE + R	8.223,3	8.193	27,3	288,3	0,0	1.965,0	4.560,0	705,0
ES (1+5) I2 TypE + R	8.221,8	8.181	32,8	286,8	0,0	1.965,0	4.560,0	705,0
ES (10,50) I2 TypE + R	8.223,8	8.187	22,8	288,8	0,0	1.965,0	4.560,0	705,0
ES (10+50) I2 TypE + R	8.225,9	8.190	21,9	290,9	0,0	1.965,0	4.560,0	705,0
ES (30,200) I2 TypE + R	8.233,8	8.202	16,8	298,8	0,0	1.965,0	4.560,0	705,0
ES (30+200) I2 TypE + R	8.229,9	8.203	14,6	294,9	0,0	1.965,0	4.560,0	705,0

Tabelle 7.9: Ergebnisse bei Logistik_Do (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	∅	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
Manueller Plan	15.465,0	15.465	-	0,0	0,0	2.760,0	1.635,0	5.535,0
MAS	8.272,4	8.269	2,1	232,4	0,0	1.185,0	4.335,0	1.260,0
HC I1	8.396,1	8.322	58,4	289,1	0,0	1.218,5	4.368,5	1.260,0
HC I2	8.487,4	8.354	131,9	219,4	0,0	1.299,0	4.449,0	1.260,0
TS I2	8.268,3	8.255	9,8	226,3	0,0	1.186,0	4.336,0	1.260,0
TS I3	8.226,8	8.222	5,7	185,8	0,0	1.185,5	4.335,5	1.260,0
TS I2 + HC3	8.224,1	8.215	5,8	183,1	0,0	1.185,5	4.335,5	1.260,0
TS I3 + HC3	8.222,6	8.215	8,0	180,6	0,0	1.186,0	4.336,0	1.260,0
TS I3 + HC8	8.220,5	8.214	3,6	180,5	0,0	1.185,0	4.335,0	1.260,0
PSO (20) I2 gBest	8.428,7	8.364	45,3	298,2	0,0	1.199,5	4.288,0	1.321,5
PSO (40) I2 gBest	8.777,2	8.636	81,3	352,8	0,0	1.232,0	4.091,5	1.550,5
PSO (200) I2 gBest	10.049,3	9.769	148,1	371,3	0,0	1.412,5	3.379,5	2.443,0
PSO (20) I3 gBest	8.245,6	8.230	14,4	196,6	0,0	1.189,5	4.339,5	1.260,0
PSO (200) I3 gBest	8.269,2	8.245	16,1	214,2	0,0	1.192,5	4.342,5	1.260,0
PSO (20) I3 Wheel	8.423,0	8.360	34,3	306,0	0,0	1.194,0	4.285,5	1.318,5
PSO (200) I3 Wheel	11.497,0	11.103	150,0	320,8	0,0	1.644,0	2.575,5	3.478,5
PSO (20) I3 Circle	8.256,0	8.240	13,8	206,4	0,0	1.190,0	4.340,0	1.260,0
PSO (200) I3 Circle	8.289,0	8.274	8,6	246,9	0,0	1.186,0	4.336,0	1.260,0
PSO (20) I3 lBest	8.254,0	8.236	16,4	203,4	0,0	1.190,5	4.340,5	1.260,0
PSO (200) I3 lBest	8.259,0	8.247	10,0	217,4	0,0	1.186,0	4.336,0	1.260,0
PSO (20) I3 gBest + R	8.247,6	8.244	1,9	207,6	0,0	1.185,0	4.335,0	1.260,0
PSO (40) I3 gBest + R	8.249,8	8.245	2,6	209,8	0,0	1.185,0	4.335,0	1.260,0
PSO (200) I3 gBest + R	8.256,0	8.250	3,5	216,0	0,0	1.185,0	4.335,0	1.260,0
PSO (20) I3 Wheel + R	8.262,2	8.256	3,2	220,9	0,0	1.185,0	4.335,0	1.260,0
PSO (200) I3 Wheel + R	8.308,9	8.297	6,6	267,4	0,0	1.185,0	4.335,0	1.260,0
PSO (20) I3 Circle + R	8.247,1	8.241	2,2	207,1	0,0	1.185,0	4.335,0	1.260,0
PSO (200) I3 Circle + R	8.251,8	8.248	1,9	211,8	0,0	1.185,0	4.335,0	1.260,0
PSO (20) I3 lBest + R	8.247,8	8.242	3,0	207,8	0,0	1.185,0	4.335,0	1.260,0
PSO (200) I3 lBest + R	8.250,7	8.246	1,9	210,7	0,0	1.185,0	4.335,0	1.260,0

Tabelle 7.10: Ergebnisse bei Logistik_Do (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	⊙	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
ES (1,5) I2 TypG	8.256,8	8.237	14,7	202,9	0,0	1.186,5	4.345,4	1.260,0
ES (1+5) I2 TypG	8.236,5	8.226	12,1	191,0	0,0	1.188,7	4.338,7	1.260,0
ES (10,50) I2 TypG	8.264,4	8.239	17,0	206,4	0,0	1.194,0	4.344,0	1.260,0
ES (10+50) I2 TypG	8.262,4	8.239	18,6	206,4	0,0	1.193,0	4.343,0	1.260,0
ES (30,200) I2 TypG	8.273,3	8.247	18,2	210,3	0,0	1.196,5	4.346,5	1.260,0
ES (30+200) I2 TypG	8.269,8	8.243	18,6	209,8	0,0	1.195,0	4.345,0	1.260,0
ES (1,5) I2 TypG + R	8.246,5	8.242	2,4	206,5	0,0	1.185,0	4.335,0	1.260,0
ES (1+5) I2 TypG + R	8.245,8	8.241	2,3	205,8	0,0	1.185,0	4.335,0	1.260,0
ES (10,50) I2 TypG + R	8.256,9	8.249	3,2	216,9	0,0	1.185,0	4.335,0	1.260,0
ES (10+50) I2 TypG + R	8.254,9	8.248	3,1	214,9	0,0	1.185,0	4.335,0	1.260,0
ES (30,200) I2 TypG + R	8.262,4	8.255	4,4	222,4	0,0	1.185,0	4.335,0	1.260,0
ES (30+200) I2 TypG + R	8.263,0	8.258	3,2	223,0	0,0	1.185,0	4.335,0	1.260,0
ES (1,5) I2 TypE	8.797,5	8.351	559,0	386,0	0,0	1.311,5	4.343,0	1.378,5
ES (1+5) I2 TypE	8.578,8	8.293	461,2	335,3	0,0	1.280,5	4.418,0	1.272,5
ES (10,50) I2 TypE	8.426,6	8.323	123,9	317,1	0,0	1.219,0	4.367,5	1.261,5
ES (10+50) I2 TypE	8.413,1	8.313	97,6	315,1	0,0	1.213,0	4.361,0	1.262,0
ES (30,200) I2 TypE	8.430,8	8.332	83,2	320,3	0,0	1.220,0	4.369,5	1.260,5
ES (30+200) I2 TypE	8.409,9	8.326	59,7	315,9	0,0	1.212,0	4.362,0	1.260,0
ES (1,5) I2 TypE + R	8.319,8	8.282	32,6	279,8	0,0	1.185,0	4.335,0	1.260,0
ES (1+5) I2 TypE + R	8.291,4	8.264	21,9	251,4	0,0	1.185,0	4.335,0	1.260,0
ES (10,50) I2 TypE + R	8.308,8	8.272	17,2	268,8	0,0	1.185,0	4.335,0	1.260,0
ES (10+50) I2 TypE + R	8.306,1	8.282	17,9	266,1	0,0	1.185,0	4.335,0	1.260,0
ES (30,200) I2 TypE + R	8.314,5	8.290	15,2	274,5	0,0	1.185,0	4.335,0	1.260,0
ES (30+200) I2 TypE + R	8.314,7	8.292	12,7	274,7	0,0	1.185,0	4.335,0	1.260,0

Tabelle 7.11: Ergebnisse bei Logistik_Fr (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	∅	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
Manueller Plan	11.850,0	11.850	-	0,0	0,0	2.400,0	1.950,0	3.750,0
MAS	5.528,2	5.523	3,2	218,2	0,0	750,0	3.540,0	510,0
HC I1	5.916,9	5.701	129,8	322,9	0,0	892,0	3.682,0	510,0
HC I2	6.005,5	5.722	223,1	216,5	0,0	989,5	3.779,5	510,0
TS I2	5.566,0	5.529	23,4	248,0	0,0	753,5	3.542,5	511,0
TS I3	5.490,7	5.482	4,0	180,7	0,0	750,0	3.540,0	510,0
TS I2 + HC3	5.487,9	5.476	11,4	174,9	0,0	751,5	3.541,5	510,0
TS I3 + HC3	5.477,9	5.472	3,7	167,4	0,0	750,0	3.539,5	510,5
TS I3 + HC8	5.481,2	5.474	7,9	169,7	0,0	750,5	3.540,0	510,5
PSO (20) I2 gBest	5.755,9	5.678	59,3	287,9	0,0	801,0	3.535,0	566,0
PSO (40) I2 gBest	6.186,8	5.980	124,8	320,8	0,0	924,5	3.507,5	717,0
PSO (200) I2 gBest	7.390,3	7.119	183,3	319,8	0,0	1.187,5	3.092,0	1.395,5
PSO (20) I3 gBest	5.573,2	5.504	53,2	200,2	0,0	781,5	3.571,5	510,0
PSO (200) I3 gBest	5.620,6	5.508	59,6	208,6	0,0	801,0	3.591,0	510,0
PSO (20) I3 Wheel	5.754,0	5.667	64,5	294,5	0,0	795,0	3.525,5	569,5
PSO (200) I3 Wheel	8.550,0	8.075	224,5	266,2	0,0	1.450,0	2.666,5	2.083,5
PSO (20) I3 Circle	5.528,0	5.505	33,2	203,0	0,0	757,5	3.547,5	510,0
PSO (200) I3 Circle	5.572,0	5.545	29,4	240,7	0,0	758,0	3.543,0	515,0
PSO (20) I3 lBest	5.527,0	5.498	38,3	202,4	0,0	757,5	3.547,5	510,0
PSO (200) I3 lBest	5.538,0	5.516	28,4	213,9	0,0	757,0	3.546,5	510,5
PSO (20) I3 gBest + R	5.500,2	5.494	3,1	190,2	0,0	750,0	3.540,0	510,0
PSO (40) I3 gBest + R	5.502,9	5.496	3,9	192,9	0,0	750,0	3.540,0	510,0
PSO (200) I3 gBest + R	5.508,3	5.501	3,5	198,3	0,0	750,0	3.540,0	510,0
PSO (20) I3 Wheel + R	5.516,4	5.508	4,2	205,4	0,0	750,0	3.540,0	510,0
PSO (200) I3 Wheel + R	5.556,3	5.544	5,9	244,7	0,0	750,0	3.540,0	510,0
PSO (20) I3 Circle + R	5.501,1	5.496	3,2	191,1	0,0	750,0	3.540,0	510,0
PSO (200) I3 Circle + R	5.504,0	5.500	2,0	194,0	0,0	750,0	3.540,0	510,0
PSO (20) I3 lBest + R	5.500,1	5.495	2,6	190,1	0,0	750,0	3.540,0	510,0
PSO (200) I3 lBest + R	5.503,1	5.498	2,0	193,1	0,0	750,0	3.540,0	510,0

Tabelle 7.12: Ergebnisse bei Logistik_Fr (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	⊙	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
ES (1,5) I2 TypG	5.560,3	5.521	26,1	215,3	0,0	772,0	3.550,0	510,0
ES (1+5) I2 TypG	5.495,8	5.487	3,8	185,8	0,0	750,0	3.540,0	510,0
ES (10,50) I2 TypG	5.570,7	5.499	57,3	199,7	0,0	780,5	3.570,5	510,0
ES (10+50) I2 TypG	5.577,2	5.503	67,6	201,2	0,0	783,0	3.573,0	510,0
ES (30,200) I2 TypG	5.593,7	5.511	49,8	205,7	0,0	789,0	3.579,0	510,0
ES (30+200) I2 TypG	5.593,8	5.510	48,2	204,8	0,0	789,5	3.579,5	510,0
ES (1,5) I2 TypG + R	5.500,4	5.496	2,4	190,5	0,0	750,0	3.540,0	510,0
ES (1+5) I2 TypG + R	5.500,3	5.493	2,8	190,3	0,0	750,0	3.540,0	510,0
ES (10,50) I2 TypG + R	5.508,1	5.500	3,6	198,1	0,0	750,0	3.540,0	510,0
ES (10+50) I2 TypG + R	5.507,5	5.500	4,0	197,5	0,0	750,0	3.540,0	510,0
ES (30,200) I2 TypG + R	5.514,6	5.507	4,3	204,3	0,0	750,0	3.540,0	510,0
ES (30+200) I2 TypG + R	5.514,2	5.508	2,4	204,2	0,0	750,0	3.540,0	510,0
ES (1,5) I2 TypE	6.350,2	5.720	616,4	435,7	0,0	1.016,0	3.733,5	582,5
ES (1+5) I2 TypE	6.318,0	5.625	1.216,5	403,0	0,0	1.043,0	3.814,0	529,0
ES (10,50) I2 TypE	5.930,6	5.647	299,4	367,1	0,0	876,0	3.664,5	511,5
ES (10+50) I2 TypE	5.835,6	5.667	129,9	358,6	0,0	832,0	3.619,0	513,0
ES (30,200) I2 TypE	5.819,7	5.641	165,9	363,2	0,0	822,0	3.609,5	512,5
ES (30+200) I2 TypE	5.833,3	5.648	120,5	364,3	0,0	828,0	3.615,0	513,0
ES (1,5) I2 TypE + R	5.583,6	5.544	31,7	273,6	0,0	750,0	3.540,0	510,0
ES (1+5) I2 TypE + R	5.578,7	5.535	38,9	268,7	0,0	750,0	3.540,0	510,0
ES (10,50) I2 TypE + R	5.595,8	5.556	24,6	285,8	0,0	750,0	3.540,0	510,0
ES (10+50) I2 TypE + R	5.592,7	5.556	24,1	282,7	0,0	750,0	3.540,0	510,0
ES (30,200) I2 TypE + R	5.602,3	5.574	19,5	292,3	0,0	750,0	3.540,0	510,0
ES (30+200) I2 TypE + R	5.603,5	5.574	19,8	293,5	0,0	750,0	3.540,0	510,0

Tabelle 7.13: Ergebnisse bei Logistik_Sa (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	∅	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
Manueller Plan	14.850,0	14.850	-	0,0	0,0	2.280,0	2.910,0	4.830,0
MAS	8.861,0	8.858	1,7	206,0	0,0	675,0	4.290,0	1.845,0
HC I1	8.985,6	8.907	42,0	274,6	0,0	703,0	4.318,0	1.845,0
HC I2	8.915,6	8.851	47,2	190,6	0,0	710,0	4.325,0	1.845,0
TS I2	8.871,9	8.837	41,3	195,9	0,0	685,5	4.300,5	1.845,0
TS I3	8.815,1	8.812	2,2	160,1	0,0	675,0	4.290,0	1.845,0
TS I2 + HC3	8.820,7	8.808	24,1	158,7	0,0	678,5	4.293,5	1.845,0
TS I3 + HC3	8.807,3	8.802	2,4	152,3	0,0	675,0	4.290,0	1.845,0
TS I3 + HC8	8.809,3	8.803	2,1	154,3	0,0	675,0	4.290,0	1.845,0
PSO (20) I2 gBest	8.960,6	8.907	27,7	252,6	0,0	680,5	4.253,5	1.887,0
PSO (40) I2 gBest	9.209,0	9.043	75,4	294,5	0,0	697,5	4.098,0	2.059,5
PSO (200) I2 gBest	10.197,9	9.889	143,7	322,4	0,0	840,5	3.566,0	2.734,5
PSO (20) I3 gBest	8.824,5	8.817	3,9	169,5	0,0	675,0	4.290,0	1.845,0
PSO (200) I3 gBest	8.841,8	8.831	5,1	186,8	0,0	675,0	4.290,0	1.845,0
PSO (20) I3 Wheel	8.958,0	8.910	35,6	262,8	0,0	677,5	4.257,0	1.880,5
PSO (200) I3 Wheel	11.409,0	11.014	202,3	277,5	0,0	1.091,5	3062,5	3.489,0
PSO (20) I3 Circle	8.833,0	8.824	4,8	177,5	0,0	675,0	4.290,0	1.845,0
PSO (200) I3 Circle	8.856,0	8.844	5,5	201,4	0,0	675,0	4.290,0	1.845,0
PSO (20) I3 lBest	8.834,0	8.823	8,3	177,5	0,0	675,5	4.290,5	1.845,0
PSO (200) I3 lBest	8.842,0	8.836	3,7	186,5	0,0	675,0	4.290,0	1.845,0
PSO (20) I3 gBest + R	8.837,6	8.833	2,3	182,6	0,0	675,0	4.290,0	1.845,0
PSO (40) I3 gBest + R	8.841,4	8.832	3,8	186,4	0,0	675,0	4.290,0	1.845,0
PSO (200) I3 gBest + R	8.854,1	8.847	3,3	199,1	0,0	675,0	4.290,0	1.845,0
PSO (20) I3 Wheel + R	8.855,8	8.849	4,3	199,3	0,0	675,0	4.290,0	1.845,0
PSO (200) I3 Wheel + R	8.899,7	8.884	5,6	243,0	0,0	675,0	4.290,0	1.845,0
PSO (20) I3 Circle + R	8.838,4	8.833	3,2	183,4	0,0	675,0	4.290,0	1.845,0
PSO (200) I3 Circle + R	8.847,2	8.843	1,9	192,2	0,0	675,0	4.290,0	1.845,0
PSO (20) I3 lBest + R	8.839,3	8.832	3,5	184,3	0,0	675,0	4.290,0	1.845,0
PSO (200) I3 lBest + R	8.845,6	8.839	2,9	190,6	0,0	675,0	4.290,0	1.845,0

Tabelle 7.14: Ergebnisse bei Logistik_Sa (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	⊙	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
ES (1,5) I2 TypG	8.849,9	8.835	16,9	192,9	0,0	675,0	4.288,0	1.847,0
ES (1+5) I2 TypG	8.815,0	8.809	2,9	160,0	0,0	675,0	4.290,0	1.845,0
ES (10,50) I2 TypG	8.831,2	8.823	4,2	176,2	0,0	675,0	4.290,0	1.845,0
ES (10+50) I2 TypG	8.830,9	8.825	4,2	175,9	0,0	675,0	4.290,0	1.845,0
ES (30,200) I2 TypG	8.840,1	8.833	4,4	185,1	0,0	675,0	4.290,0	1.845,0
ES (30+200) I2 TypG	8.839,2	8.827	6,0	184,2	0,0	675,0	4.290,0	1.845,0
ES (1,5) I2 TypG + R	8.835,9	8.830	2,7	180,9	0,0	675,0	4.290,0	1.845,0
ES (1+5) I2 TypG + R	8.832,0	8.827	2,3	177,0	0,0	675,0	4.290,0	1.845,0
ES (10,50) I2 TypG + R	8.845,7	8.836	4,2	190,7	0,0	675,0	4.290,0	1.845,0
ES (10+50) I2 TypG + R	8.843,6	8.832	4,7	188,6	0,0	675,0	4.290,0	1.845,0
ES (30,200) I2 TypG + R	8.856,6	8.850	3,7	201,6	0,0	675,0	4.290,0	1.845,0
ES (30+200) I2 TypG + R	8.856,7	8.846	5,6	201,7	0,0	675,0	4.290,0	1.845,0
ES (1,5) I2 TypE	9.422,4	8.938	732,2	356,9	0,0	853,5	4.415,0	1.898,5
ES (1+5) I2 TypE	9.115,0	8.917	217,9	319,5	0,0	738,0	4.338,5	1.859,5
ES (10,50) I2 TypE	8.983,2	8.885	76,0	280,7	0,0	697,0	4.308,5	1.848,5
ES (10+50) I2 TypE	9.000,1	8.905	79,1	279,6	0,0	705,0	4.314,5	1.850,5
ES (30,200) I2 TypE	8.989,5	8.919	63,7	289,5	0,0	697,0	4.311,0	1.846,0
ES (30+200) I2 TypE	8.983,8	8.915	57,3	288,8	0,0	694,0	4.307,0	1.847,0
ES (1,5) I2 TypE + R	8.917,7	8.887	35,6	262,7	0,0	675,0	4.290,0	1.845,0
ES (1+5) I2 TypE + R	8.906,4	8.863	37,0	251,4	0,0	675,0	4.290,0	1.845,0
ES (10,50) I2 TypE + R	8.902,0	8.869	20,7	247,0	0,0	675,0	4.290,0	1.845,0
ES (10+50) I2 TypE + R	8.911,4	8.879	20,4	256,4	0,0	675,0	4.290,0	1.845,0
ES (30,200) I2 TypE + R	8.918,7	8.888	14,3	263,7	0,0	675,0	4.290,0	1.845,0
ES (30+200) I2 TypE + R	8.916,0	8.891	16,6	261,0	0,0	675,0	4.290,0	1.845,0

Tabelle 7.15: Ergebnisse bei Logistik_So (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	∅	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
Manueller Plan	14.580,0	14.580	-	0,0	0,0	2.505,0	2.385,0	4.845,0
MAS	7.337,1	7.333	3,1	227,1	0,0	570,0	4.050,0	1.245,0
HC I1	7.471,0	7.397	56,7	295,0	0,0	603,0	4.083,0	1.245,0
HC I2	7.580,0	7.312	205,6	212,0	0,0	699,0	4.179,0	1.245,0
TS I2	7.321,9	7.307	17,1	208,9	0,0	571,5	4.051,5	1.245,0
TS I3	7.289,7	7.285	2,5	179,7	0,0	570,0	4.050,0	1.245,0
TS I2 + HC3	7.283,1	7.277	3,8	173,1	0,0	570,0	4.050,0	1.245,0
TS I3 + HC3	7.282,0	7.278	2,0	172,0	0,0	570,0	4.050,0	1.245,0
TS I3 + HC8	7.283,2	7.278	4,0	172,7	0,0	570,0	4.049,5	1.245,5
PSO (20) I2 gBest	7.487,9	7.433	33,7	295,3	0,0	576,0	3.986,0	1.315,0
PSO (40) I2 gBest	7.872,3	7.670	80,3	348,3	0,0	614,0	3.768,0	1.571,0
PSO (200) I2 gBest	9.059,6	8.633	203,0	373,1	0,0	810,5	3.195,0	2.340,5
PSO (20) I3 gBest	7.300,5	7.292	5,8	190,5	0,0	570,0	4.050,0	1.245,0
PSO (200) I3 gBest	7.324,2	7.312	6,3	214,2	0,0	570,0	4.050,0	1.245,0
PSO (20) I3 Wheel	7.496,0	7.424	46,4	306,0	0,0	573,0	3.978,5	1.319,5
PSO (200) I3 Wheel	10.441,0	10.131	182,2	322,9	0,0	1.112,0	2.667,5	3.169,5
PSO (20) I3 Circle	7.308,0	7.299	5,2	197,6	0,0	570,0	4.050,0	1.245,0
PSO (200) I3 Circle	7.362,0	7.343	7,9	251,0	0,0	570,0	4.049,0	1.246,0
PSO (20) I3 lBest	7.310,0	7.302	5,8	200,4	0,0	570,0	4.050,0	1.245,0
PSO (200) I3 lBest	7.328,0	7.318	5,3	218,0	0,0	570,0	4.050,0	1.245,0
PSO (20) I3 gBest + R	7.330,3	7.313	2,8	220,3	0,0	570,0	4.050,0	1.245,0
PSO (40) I3 gBest + R	7.318,3	7.318	2,6	208,3	0,0	570,0	4.050,0	1.245,0
PSO (200) I3 gBest + R	7.322,3	7.324	3,2	212,3	0,0	570,0	4.050,0	1.245,0
PSO (20) I3 Wheel + R	7.337,5	7.331	3,7	226,1	0,0	570,0	4.050,0	1.245,0
PSO (200) I3 Wheel + R	7.387,5	7.377	5,0	275,8	0,0	570,0	4.050,0	1.245,0
PSO (20) I3 Circle + R	7.320,1	7.314	2,1	210,1	0,0	570,0	4.050,0	1.245,0
PSO (200) I3 Circle + R	7.326,6	7.321	2,5	216,6	0,0	570,0	4.050,0	1.245,0
PSO (20) I3 lBest + R	7.319,5	7.313	2,9	209,5	0,0	570,0	4.050,0	1.245,0
PSO (200) I3 lBest + R	7.325,2	7.322	2,0	215,2	0,0	570,0	4.050,0	1.245,0

Tabelle 7.16: Ergebnisse bei Logistik_So (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	
	⊙	Minimum	Std.-abw.				Bedarf>0	Bedarf=0
ES (1,5) I2 TypG	7.329,6	7.317	6,6	219,6	0,0	570,0	4.050,0	1.245,0
ES (1+5) I2 TypG	7.290,2	7.287	2,2	180,2	0,0	570,0	4.050,0	1.245,0
ES (10,50) I2 TypG	7.308,7	7.301	5,3	198,7	0,0	570,0	4.050,0	1.245,0
ES (10+50) I2 TypG	7.310,4	7.299	4,6	200,4	0,0	570,0	4.050,0	1.245,0
ES (30,200) I2 TypG	7.319,9	7.311	4,4	209,9	0,0	570,0	4.050,0	1.245,0
ES (30+200) I2 TypG	7.319,5	7.309	4,8	209,5	0,0	570,0	4.050,0	1.245,0
ES (1,5) I2 TypG + R	7.316,8	7.311	2,7	206,8	0,0	570,0	4.050,0	1.245,0
ES (1+5) I2 TypG + R	7.312,9	7.309	1,9	202,9	0,0	570,0	4.050,0	1.245,0
ES (10,50) I2 TypG + R	7.324,7	7.317	3,3	214,7	0,0	570,0	4.050,0	1.245,0
ES (10+50) I2 TypG + R	7.325,0	7.318	3,6	215,0	0,0	570,0	4.050,0	1.245,0
ES (30,200) I2 TypG + R	7.336,4	7.330	4,1	226,4	0,0	570,0	4.050,0	1.245,0
ES (30+200) I2 TypG + R	7.334,9	7.326	4,0	224,9	0,0	570,0	4.050,0	1.245,0
ES (1,5) I2 TypE	7.639,4	7.416	267,1	357,9	0,0	624,5	4.042,0	1.307,5
ES (1+5) I2 TypE	7.720,9	7.342	560,8	357,4	0,0	683,0	4.135,5	1.272,5
ES (10,50) I2 TypE	7.505,1	7.370	108,5	326,1	0,0	604,0	4.083,0	1.246,0
ES (10+50) I2 TypE	7.510,4	7.361	158,3	326,4	0,0	606,0	4.084,0	1.247,0
ES (30,200) I2 TypE	7.480,7	7.396	56,2	321,2	0,0	594,5	4.074,0	1.245,5
ES (30+200) I2 TypE	7.468,1	7.397	49,9	317,1	0,0	590,5	4.070,5	1.245,0
ES (1,5) I2 TypE + R	7.383,7	7.345	33,0	273,7	0,0	570,0	4.050,0	1.245,0
ES (1+5) I2 TypE + R	7.365,4	7.322	24,3	255,4	0,0	570,0	4.050,0	1.245,0
ES (10,50) I2 TypE + R	7.376,3	7.344	21,1	266,3	0,0	570,0	4.050,0	1.245,0
ES (10+50) I2 TypE + R	7.387,2	7.346	23,2	277,2	0,0	570,0	4.050,0	1.245,0
ES (30,200) I2 TypE + R	7.393,5	7.353	18,5	283,5	0,0	570,0	4.050,0	1.245,0
ES (30+200) I2 TypE + R	7.390,8	7.367	12,3	280,8	0,0	570,0	4.050,0	1.245,0

Tabelle 7.17: Ergebnisse bei Logistik_S_Complete (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	∅	Minimum	Std.-abw.					
Manueller Plan	377.535,0	377.535	-	0,0	1.545,0	20.130,0	14.610,0	33.795,0
MAS	53.650,8	53.197	172,7	1.732,7	0,0	7.420,0	35.695,0	8.803,1
HC I1	48.726,4	48.487	110,4	1.658,3	0,0	7.503,5	35.778,5	3.786,1
HC I2	48.294,5	47.842	239,6	1.155,4	0,0	7.605,5	35.880,5	3.653,1
PSO (20) I3 gBest	59.161,0	58.736	230,2	3.462,1	0,0	8.343,5	36.618,5	10.736,9
PSO (200) I3 gBest	59.674,4	59.120	216,3	2.912,1	0,0	9.062,0	37.337,0	10.363,3
PSO (20) I3 Wheel	73.874,4	72.774	448,9	3.786,3	0,0	15.036,5	43.311,5	11.740,1
PSO (200) I3 Wheel	82.731,3	81.945	962,7	2.400,9	32,0	20.889,5	49.164,5	10.244,5
PSO (20) I3 Circle	58.589,1	58.111	233,7	3.226,0	0,0	8.261,0	36.536,0	10.566,1
PSO (200) I3 Circle	66.041,6	65.121	359,6	2.840,4	0,0	12.238,0	40.513,0	10.450,2
PSO (20) I3 lBest	58.286,9	57.695	262,4	3.170,3	0,0	8.190,5	36.465,5	10.460,6
PSO (200) I3 lBest	62.126,7	61.374	357,1	2.713,2	0,0	10.445,0	38.720,0	10.248,5
PSO (20) I3 gBest + R	57.852,4	57.291	274,7	3.439,9	0,0	7.374,0	35.649,0	11.389,5
PSO (200) I3 gBest + R	57.561,6	56.979	352,4	3.333,4	0,0	7.372,5	35.647,5	11.208,2
PSO (20) I3 Wheel + R	59.188,8	58.887	173,4	3.669,7	0,0	7.413,0	35.688,0	12.418,1
PSO (200) I3 Wheel + R	60.149,3	59.828	207,3	3.832,4	0,0	7.552,0	35.827,0	12.937,9
PSO (20) I3 Circle + R	56.205,1	55.909	168,9	3.076,9	0,0	7.365,5	35.640,5	10.122,2
PSO (200) I3 Circle + R	57.134,4	56.960	88,4	3.250,2	0,0	7.372,0	35.647,0	10.865,2
PSO (20) I3 lBest + R	55.824,3	55.540	149,4	2.984,0	0,0	7.366,0	35.641,0	9.833,3
PSO (200) I3 lBest + R	56.651,7	56.443	125,4	3.134,7	0,0	7.367,5	35.642,5	10.507,0
ES (1,5) I2 TypG	55.376,9	50.632	1.842,6	2.616,4	0,0	8.862,5	37.137,5	6.760,5
ES (1+5) I2 TypG	55.156,0	50.488	2.020,1	2.556,8	0,0	8.821,0	37.096,0	6.682,2
ES (10,50) I2 TypG	58.612,6	56.796	1.350,9	3.154,5	0,0	9.336,0	37.611,0	8.511,1
ES (10+50) I2 TypG	58.270,8	57.255	703,8	3.133,4	0,0	9.248,5	37.523,5	8.365,4
ES (30,200) I2 TypG	65.550,4	53.645	4.707,3	3.705,6	0,0	10.800,5	39.075,5	11.968,8
ES (30+200) I2 TypG	66.283,5	53.582	3.477,1	3.884,2	0,0	10.897,5	39.172,5	12.329,3

Tabelle 7.18: Ergebnisse bei Logistik_S_Complete (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	∅	Minimum	Std.-abw.					
ES (1,5) I2 TypG + R	59.063,7	58.832	106,0	3.623,0	0,0	7.430,0	35.705,0	12.305,7
ES (1+5) I2 TypG + R	61.624,5	61.303	181,3	3.928,8	0,0	7.922,5	36.197,5	13.575,7
ES (10,50) I2 TypG + R	58.524,1	58.278	134,6	3.471,2	0,0	7.409,5	35.684,5	11.958,9
ES (10+50) I2 TypG + R	60.782,7	60.430	205,9	3.874,2	0,0	7.630,0	35.905,0	13.373,5
ES (30,200) I2 TypG + R	58.662,0	58.349	122,5	3.486,0	0,0	7.401,0	35.676,0	12.099,0
ES (30+200) I2 TypG + R	60.202,2	59.836	170,4	3.801,3	0,0	7.506,0	35.781,0	13.113,9
ES (1,5) I2 TypE	62.217,6	55.233	4.760,3	3.252,5	0,0	10.573,0	38.848,0	9.544,1
ES (1+5) I2 TypE	60.933,6	52.301	7.545,6	3.140,2	0,0	10.106,5	38.381,5	9.305,4
ES (10,50) I2 TypE	58.790,2	51.916	5.766,4	3.149,7	0,0	9.059,5	37.334,5	9.246,5
ES (10+50) I2 TypE	56.673,9	49.663	4.667,2	2.916,4	0,0	8.451,0	36.726,0	8.580,5
ES (30,200) I2 TypE	60.126,9	50.921	9.366,9	3.050,5	0,0	9.418,0	37.693,0	9.965,4
ES (30+200) I2 TypE	60.039,8	50.879	7.533,3	3.228,8	0,0	9.040,0	37.315,0	10.456,0
ES (1,5) I2 TypE + R	58.939,8	58.735	183,0	3.605,6	0,0	7.400,5	35.675,5	12.258,2
ES (1+5) I2 TypE + R	61.222,5	60.631	316,6	3.920,6	0,0	7.765,0	36.040,0	13.496,9
ES (10,50) I2 TypE + R	58.455,2	56.748	1.191,6	3.467,0	0,0	7.425,0	35.700,0	11.863,2
ES (10+50) I2 TypE + R	60.721,7	59.891	573,5	3.848,7	0,0	7.625,5	35.900,5	13.347,0
ES (30,200) I2 TypE + R	58.263,6	57.268	664,3	3.411,2	0,0	7.382,5	35.657,5	11.812,4
ES (30+200) I2 TypE + R	60.351,5	58.815	890,4	3.809,2	0,0	7.572,5	35.847,5	13.122,3

Tabelle 7.19: Ergebnisse bei Logistik_S_Mo (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	∅	Minimum	Std.-abw.					
Manueller Plan	79.650,0	79.650	-	0,0	345,0	2.955,0	2.160,0	5.535,0
MAS	8.089,6	8.036	39,8	247,0	0,0	900,0	5.639,5	1.303,1
HC I1	7.337,0	7.337	0,0	137,0	0,0	990,0	5.730,0	480,0
HC I2	7.335,9	7.329	5,2	136,3	0,0	990,0	5.730,0	479,6
PSO (20) I3 gBest	7.262,9	7.199	42,6	154,0	0,0	994,0	5.734,0	380,9
PSO (200) I3 gBest	7.293,9	7.200	53,1	158,7	0,0	1.008,0	5.748,0	379,2
PSO (20) I3 Wheel	7.456,3	7.313	74,3	207,9	0,0	1.014,5	5.754,5	478,9
PSO (200) I3 Wheel	8.884,7	8.573	169,3	246,7	0,0	1.708,5	6.448,5	481,1
PSO (20) I3 Circle	7.257,9	7.177	49,0	156,3	0,0	989,5	5.729,5	382,6
PSO (200) I3 Circle	7.255,1	7.199	28,2	162,6	0,0	981,0	5.721,0	390,5
PSO (20) I3 lBest	7.278,3	7.183	48,6	155,4	0,0	1.001,5	5.741,5	379,8
PSO (200) I3 lBest	7.236,9	7.169	31,2	158,9	0,0	978,0	5.718,0	381,9
PSO (20) I3 gBest + R	8.498,6	8.383	89,3	413,4	0,0	900,0	5.640,0	1.545,2
PSO (200) I3 gBest + R	8.521,1	8.357	61,9	414,4	0,0	900,0	5.640,0	1.566,7
PSO (20) I3 Wheel + R	8.844,4	8.773	39,5	485,5	0,0	901,5	5.641,5	1.815,9
PSO (200) I3 Wheel + R	8.959,0	8.851	41,8	512,6	0,0	902,5	5.642,5	1.901,4
PSO (20) I3 Circle + R	8.260,1	8.183	43,4	365,8	0,0	900,0	5.640,0	1.354,3
PSO (200) I3 Circle + R	8.331,4	8.266	28,1	375,2	0,0	900,0	5.640,0	1.416,2
PSO (20) I3 lBest + R	8.247,4	8.144	53,9	361,9	0,0	900,0	5.640,0	1.345,5
PSO (200) I3 lBest + R	8.206,8	8.151	28,1	356,8	0,0	900,0	5.640,0	1.310,0
ES (1,5) I2 TypG	7.330,7	7.229	51,1	203,6	0,0	935,5	5.675,5	516,1
ES (1+5) I2 TypG	7.363,6	7.283	53,9	213,5	0,0	943,5	5.683,5	523,1
ES (10,50) I2 TypG	7.412,8	7.319	53,2	230,2	0,0	942,5	5.682,5	557,6
ES (10+50) I2 TypG	7.424,1	7.316	67,4	228,8	0,0	950,5	5.690,5	554,3
ES (30,200) I2 TypG	7.382,7	7.298	58,8	226,9	0,0	926,0	5.666,0	563,8
ES (30+200) I2 TypG	7.398,8	7.290	54,7	232,1	0,0	927,5	5.667,5	571,7

Tabelle 7.20: Ergebnisse bei Logistik_S_Mo (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	∅	Minimum	Std.-abw.					
ES (1,5) I2 TypG + R	8.691,0	8.631	20,0	452,9	0,0	900,0	5.640,0	1.698,1
ES (1+5) I2 TypG + R	8.999,7	8.937	38,1	514,8	0,0	905,5	5.645,5	1.933,9
ES (10,50) I2 TypG + R	8.564,5	8.201	103,0	421,6	0,0	900,0	5.640,0	1.602,9
ES (10+50) I2 TypG + R	8.980,8	8.813	61,7	515,3	0,0	904,0	5.644,0	1.917,5
ES (30,200) I2 TypG + R	8.664,6	8.195	138,5	440,3	0,0	900,0	5.640,0	1.684,3
ES (30+200) I2 TypG + R	8.972,1	8.726	86,3	512,9	0,0	904,0	5.644,0	1.911,2
ES (1,5) I2 TypE	7.552,6	7.259	230,3	206,9	0,0	1.057,5	5.797,5	490,7
ES (1+5) I2 TypE	7.490,8	7.231	263,3	168,0	0,0	1.095,5	5.835,5	391,8
ES (10,50) I2 TypE	7.346,5	7.210	82,5	153,1	0,0	1.038,0	5.778,0	377,4
ES (10+50) I2 TypE	7.346,8	7.261	70,0	152,8	0,0	1.039,5	5.779,5	375,0
ES (30,200) I2 TypE	7.325,4	7.215	63,1	141,3	0,0	1.038,5	5.778,5	367,1
ES (30+200) I2 TypE	7.303,8	7.204	59,1	146,3	0,0	1.023,0	5.763,0	371,5
ES (1,5) I2 TypE + R	8.743,2	8.672	77,4	454,9	0,0	901,5	5.641,5	1.745,3
ES (1+5) I2 TypE + R	9.008,5	8.940	38,4	524,4	0,0	903,5	5.643,5	1.937,1
ES (10,50) I2 TypE + R	8.533,7	8.158	223,4	411,1	0,0	901,5	5.641,5	1.579,6
ES (10+50) I2 TypE + R	8.951,9	8.838	72,2	505,9	0,0	904,5	5.644,5	1.897,0
ES (30,200) I2 TypE + R	8.494,4	8.168	137,6	400,9	0,0	900,0	5.640,0	1.553,5
ES (30+200) I2 TypE + R	8.907,4	8.761	104,9	492,2	0,0	903,0	5.643,0	1.869,2

Tabelle 7.21: Ergebnisse bei Logistik_S_Di (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	∅	Minimum	Std.-abw.					
Manueller Plan	174.180,0	174.180	-	0,0	825,0	3.420,0	1.680,0	4.080,0
MAS	6.425,0	6.264	60,7	241,7	0,0	1.327,5	3.653,0	1.202,8
HC I1	5.894,0	5.890	6,3	193,2	0,0	1.335,0	3.675,0	690,8
HC I2	5.897,6	5.890	4,4	193,6	0,0	1.335,0	3.675,0	694,0
PSO (20) I3 gBest	5.740,5	5.685	30,1	176,7	0,0	1.346,0	3.686,0	531,8
PSO (200) I3 gBest	5.755,7	5.682	36,5	177,6	0,0	1.355,5	3.695,5	527,1
PSO (20) I3 Wheel	6.047,5	5.907	56,3	255,5	0,0	1.390,0	3.730,0	672,0
PSO (200) I3 Wheel	8.429,1	8.134	181,7	292,9	11,5	2.587,5	4.927,5	609,8
PSO (20) I3 Circle	5.745,3	5.686	34,8	182,1	0,0	1.344,0	3.684,0	535,2
PSO (200) I3 Circle	5.783,1	5.708	30,4	193,3	0,0	1.352,5	3.692,5	543,7
PSO (20) I3 lBest	5.749,5	5.694	36,1	181,9	0,0	1.344,5	3.684,5	538,6
PSO (200) I3 lBest	5.731,8	5.684	26,5	177,6	0,0	1.344,0	3.684,0	526,4
PSO (20) I3 gBest + R	6.481,0	6.295	114,7	345,2	0,0	1.320,0	3.660,0	1.155,8
PSO (200) I3 gBest + R	6.553,6	6.451	78,5	346,0	0,0	1.320,0	3.660,0	1.227,6
PSO (20) I3 Wheel + R	6.963,2	6.918	42,2	454,6	0,0	1.320,0	3.660,0	1.528,6
PSO (200) I3 Wheel + R	7.127,1	7.044	40,5	491,0	0,0	1.320,0	3.660,0	1.656,1
PSO (20) I3 Circle + R	6.260,7	6.193	34,2	286,6	0,0	1.320,0	3.660,0	994,1
PSO (200) I3 Circle + R	6.393,1	6.345	26,6	310,0	0,0	1.320,0	3.660,0	1.103,1
PSO (20) I3 lBest + R	6.268,4	6.199	31,9	287,1	0,0	1.320,0	3.660,0	1.001,3
PSO (200) I3 lBest + R	6.265,9	6.196	38,8	281,0	0,0	1.320,0	3.660,0	1.004,9
ES (1,5) I2 TypG	5.906,9	5.855	34,5	236,1	0,0	1.323,0	3.663,0	684,8
ES (1+5) I2 TypG	6.000,3	5.919	37,0	265,9	0,0	1.325,5	3.665,5	743,4
ES (10,50) I2 TypG	6.055,1	5.974	40,3	279,3	0,0	1.331,0	3.671,0	773,8
ES (10+50) I2 TypG	6.058,4	5.949	65,4	283,4	0,0	1.327,0	3.667,0	781,0
ES (30,200) I2 TypG	6.051,4	5.972	37,4	280,1	0,0	1.325,5	3.665,5	780,3
ES (30+200) I2 TypG	6.064,2	5.984	42,9	285,2	0,0	1.327,0	3.667,0	785,0

Tabelle 7.22: Ergebnisse bei Logistik_S_Di (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	∅	Minimum	Std.-abw.					
ES (1,5) I2 TypG + R	6.812,5	6.764	21,5	421,3	0,0	1.320,0	3.660,0	1.411,2
ES (1+5) I2 TypG + R	7.234,3	7.131	48,2	519,1	0,0	1.324,5	3.664,5	1.726,2
ES (10,50) I2 TypG + R	6.653,3	6.275	92,2	391,9	0,0	1.320,0	3.660,0	1.281,4
ES (10+50) I2 TypG + R	7.210,9	6.965	88,9	506,2	0,0	1.325,5	3.665,5	1.713,7
ES (30,200) I2 TypG + R	6.758,5	6.314	157,3	406,6	0,0	1.320,0	3.660,0	1.371,9
ES (30+200) I2 TypG + R	7.194,0	6.765	137,3	503,9	0,0	1.327,0	3.667,0	1.696,1
ES (1,5) I2 TypE	6.303,3	5.847	717,5	276,8	0,0	1.470,0	3.810,0	746,5
ES (1+5) I2 TypE	5.988,9	5.709	369,4	206,2	0,0	1.436,0	3.776,0	570,7
ES (10,50) I2 TypE	5.814,4	5.714	152,7	190,1	0,0	1.364,0	3.704,0	556,3
ES (10+50) I2 TypE	5.815,0	5.713	146,7	183,2	0,0	1.371,5	3.711,5	548,8
ES (30,200) I2 TypE	5.778,6	5.692	123,6	173,6	0,0	1.363,0	3.703,0	539,0
ES (30+200) I2 TypE	5.780,4	5.686	131,7	176,5	0,0	1.361,0	3.701,0	541,9
ES (1,5) I2 TypE + R	6.904,8	6.788	160,2	437,4	0,0	1.326,0	3.666,0	1.475,4
ES (1+5) I2 TypE + R	7.305,3	7.206	53,0	527,3	0,0	1.332,5	3.672,5	1.773,0
ES (10,50) I2 TypE + R	6.751,9	6.594	155,9	410,6	0,0	1.320,5	3.660,5	1.360,3
ES (10+50) I2 TypE + R	7.282,4	7.083	61,5	530,9	0,0	1.327,0	3.667,0	1.757,5
ES (30,200) I2 TypE + R	7.004,5	6.691	162,9	459,4	0,0	1.322,5	3.662,5	1.560,1
ES (30+200) I2 TypE + R	7.284,8	7.005	65,0	530,9	0,0	1.328,0	3.668,0	1.757,9

Tabelle 7.23: Ergebnisse bei Logistik_S_Mi (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	∅	Minimum	Std.-abw.					
Manueller Plan	85.920,0	85.920	-	0,0	375,0	3.810,0	1.890,0	5.220,0
MAS	8.863,5	8.769	55,6	275,9	0,0	1.965,0	5.262,5	1.360,1
HC I1	8.060,0	8.060	0,0	170,0	0,0	2.055,0	5.355,0	480,0
HC I2	8.060,0	8.060	0,0	170,0	0,0	2.055,0	5.355,0	480,0
PSO (20) I3 gBest	7.901,4	7.852	30,9	153,5	0,0	2.045,5	5.345,5	356,9
PSO (200) I3 gBest	7.914,6	7.834	42,2	155,6	0,0	2.051,5	5.351,5	356,0
PSO (20) I3 Wheel	8.127,5	8.034	64,0	220,3	0,0	2.066,0	5.366,0	475,2
PSO (200) I3 Wheel	9.598,3	9.369	144,7	256,9	0,5	2.775,0	6.075,0	491,0
PSO (20) I3 Circle	7.901,1	7.815	42,9	157,4	0,0	2.040,5	5.340,5	362,7
PSO (200) I3 Circle	7.893,6	7.834	30,8	163,1	0,0	2.030,0	5.330,0	370,0
PSO (20) I3 lBest	7.905,4	7.814	40,1	158,5	0,0	2.042,0	5.342,0	362,9
PSO (200) I3 lBest	7.876,6	7.810	33,0	155,0	0,0	2.031,5	5.331,5	358,6
PSO (20) I3 gBest + R	9.177,3	9.042	94,7	471,0	0,0	1.965,5	5.265,5	1.475,3
PSO (200) I3 gBest + R	9.164,4	9.001	116,5	457,6	0,0	1.965,0	5.265,0	1.476,8
PSO (20) I3 Wheel + R	9.532,4	9.466	36,7	543,7	0,0	1.965,5	5.265,5	1.757,7
PSO (200) I3 Wheel + R	9.654,2	9.560	52,7	561,5	0,0	1.969,0	5.269,0	1.854,7
PSO (20) I3 Circle + R	8.810,8	8.728	63,3	378,4	0,0	1.965,0	5.265,0	1.202,4
PSO (200) I3 Circle + R	8.923,1	8.847	42,7	403,1	0,0	1.965,0	5.265,0	1.290,0
PSO (20) I3 lBest + R	8.791,5	8.683	51,9	373,9	0,0	1.965,0	5.265,0	1.187,6
PSO (200) I3 lBest + R	8.765,9	8.678	50,2	365,9	0,0	1.965,0	5.265,0	1.170,0
ES (1,5) I2 TypG	7.999,3	7.923	46,8	218,4	0,0	1.978,5	5.278,5	523,9
ES (1+5) I2 TypG	8.068,7	7.967	54,5	236,9	0,0	1.992,5	5.292,5	546,8
ES (10,50) I2 TypG	8.115,8	7.998	61,3	245,6	0,0	2.000,0	5.300,0	570,2
ES (10+50) I2 TypG	8.081,6	8.006	39,0	240,3	0,0	1.994,5	5.294,5	552,3
ES (30,200) I2 TypG	8.076,8	7.944	65,7	234,1	0,0	1.994,5	5.294,5	553,7
ES (30+200) I2 TypG	8.071,9	7.992	51,3	235,4	0,0	1.989,0	5.289,0	558,5

Tabelle 7.24: Ergebnisse bei Logistik_S_Mi (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	∅	Minimum	Std.-abw.					
ES (1,5) I2 TypG + R	9.380,2	9.336	24,7	518,0	0,0	1.965,0	5.265,0	1.632,2
ES (1+5) I2 TypG + R	9.744,0	9.624	57,2	559,8	0,0	1.971,5	5.271,5	1.941,2
ES (10,50) I2 TypG + R	9.219,0	8.703	167,6	479,9	0,0	1.965,5	5.265,5	1.508,1
ES (10+50) I2 TypG + R	9.718,1	9.576	59,4	556,4	0,0	1.969,5	5.269,5	1.922,7
ES (30,200) I2 TypG + R	9.330,2	8.649	214,0	489,7	0,0	1.965,0	5.265,0	1.610,5
ES (30+200) I2 TypG + R	9.686,6	9.407	86,7	559,6	0,0	1.968,5	5.268,5	1.890,0
ES (1,5) I2 TypE	8.290,8	7.973	244,2	230,8	0,0	2.127,0	5.427,0	506,0
ES (1+5) I2 TypE	8.087,0	7.896	322,7	170,3	0,0	2.121,0	5.421,0	374,7
ES (10,50) I2 TypE	7.954,2	7.876	59,1	155,8	0,0	2.072,0	5.372,0	354,4
ES (10+50) I2 TypE	7.969,8	7.891	82,2	155,5	0,0	2.080,0	5.380,0	354,3
ES (30,200) I2 TypE	7.919,7	7.877	32,0	144,6	0,0	2.064,5	5.364,5	346,1
ES (30+200) I2 TypE	7.918,5	7.840	40,0	144,7	0,0	2.063,5	5.363,5	346,8
ES (1, 5) I2 TypE + R	9.458,1	9.375	117,2	527,2	0,0	1.969,5	5.269,5	1.691,9
ES (1+ 5) I2 TypE + R	9.771,2	9.689	48,8	566,2	0,0	1.973,0	5.273,0	1.959,0
ES (10,50) I2 TypE + R	9.393,9	8.883	193,7	503,7	0,0	1.968,5	5.268,5	1.653,2
ES (10+50) I2 TypE + R	9.727,7	9.611	70,6	555,8	0,0	1.971,5	5.271,5	1.928,9
ES (30,200) I2 TypE + R	9.462,0	8.939	222,7	507,0	0,0	1.967,0	5.267,0	1.721,0
ES (30+200) I2 TypE + R	9.671,0	9.555	82,5	551,2	0,0	1.968,0	5.268,0	1.883,8

Tabelle 7.25: Ergebnisse bei Logistik_S_Do (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	⊙	Minimum	Std.-abw.					
Manueller Plan	9.930,0	9.930	-	0,0	0,0	2.760,0	1.635,0	5.535,0
MAS	8.521,9	8.467	32,4	240,3	0,0	1.245,0	5.655,0	1.381,6
HC I1	7.404,4	7.390	10,4	122,7	0,0	1.245,0	5.655,0	381,7
HC I2	7.401,6	7.390	11,1	121,7	0,0	1.245,0	5.655,0	379,9
PSO (20) I3 gBest	7.234,5	7.203	27,2	129,3	0,0	1.216,0	5.626,0	263,2
PSO (200) I3 gBest	7.245,4	7.208	34,5	129,1	0,0	1.221,5	5.631,5	263,3
PSO (20) I3 Wheel	7.335,2	7.270	33,6	163,8	0,0	1.218,5	5.628,5	324,4
PSO (200) I3 Wheel	8.147,4	7.997	95,9	167,8	0,0	1.634,5	6.044,5	300,6
PSO (20) I3 Circle	7.237,0	7.198	17,8	130,5	0,0	1.215,5	5.625,5	265,5
PSO (200) I3 Circle	7.224,6	7.203	9,8	128,3	0,0	1.210,0	5.620,0	266,3
PSO (20) I3 lBest	7.240,7	7.196	28,1	129,7	0,0	1.217,5	5.627,5	266,0
PSO (200) I3 lBest	7.222,7	7.204	7,0	125,1	0,0	1.211,5	5.621,5	264,6
PSO (20) I3 gBest + R	8.363,1	8.225	85,2	378,7	0,0	1.185,0	5.595,0	1.204,4
PSO (200) I3 gBest + R	8.372,1	8.264	82,9	366,7	0,0	1.185,0	5.595,0	1.225,4
PSO (20) I3 Wheel + R	8.776,5	8.642	43,0	468,1	0,0	1.185,0	5.595,0	1.528,4
PSO (200) I3 Wheel + R	8.984,0	8.905	46,3	507,0	0,0	1.185,0	5.595,0	1.697,0
PSO (20) I3 Circle + R	8.107,6	8.019	41,9	306,3	0,0	1.185,0	5.595,0	1.021,3
PSO (200) I3 Circle + R	8.105,2	8.052	27,9	310,9	0,0	1.185,0	5.595,0	1.014,3
PSO (20) I3 lBest + R	8.122,6	8.058	39,6	305,0	0,0	1.185,0	5.595,0	1.037,6
PSO (200) I3 lBest + R	8.028,3	7.991	24,9	291,8	0,0	1.185,0	5.595,0	956,5
ES (1,5) I2 TypG	7.229,2	7.192	15,9	130,5	0,0	1.206,5	5.616,5	275,7
ES (1+5) I2 TypG	7.232,3	7.223	14,8	128,4	0,0	1.216,5	5.626,5	260,9
ES (10,50) I2 TypG	7.244,2	7.220	33,0	128,7	0,0	1.222,5	5.632,5	260,5
ES (10+50) I2 TypG	7.240,9	7.219	30,4	128,3	0,0	1.221,0	5.631,0	260,6
ES (30,200) I2 TypG	7.239,6	7.215	32,9	124,1	0,0	1.222,5	5.632,5	260,5
ES (30+200) I2 TypG	7.239,3	7.210	35,3	120,9	0,0	1.224,0	5.634,0	260,4

Tabelle 7.26: Ergebnisse bei Logistik_S_Do (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	\emptyset	Minimum	Std.-abw.					
ES (1,5) I2 TypG + R	8.610,9	8.583	17,7	430,1	0,0	1.185,0	5.595,0	1.400,8
ES (1+5) I2 TypG + R	9.094,3	8.985	60,6	519,3	0,0	1.185,0	5.595,0	1.795,0
ES (10,50) I2 TypG + R	8.482,2	8.149	70,1	400,1	0,0	1.185,0	5.595,0	1.302,1
ES (10+50) I2 TypG + R	9.108,6	8.997	48,8	521,3	0,0	1.185,0	5.595,0	1.807,3
ES (30,200) I2 TypG + R	8.639,9	8.208	89,1	420,9	0,0	1.185,0	5.595,0	1.439,0
ES (30+200) I2 TypG + R	9.082,1	8.781	71,7	514,6	0,0	1.185,0	5.595,0	1.787,5
ES (1,5) I2 TypE	7.471,8	7.297	125,2	187,2	0,0	1.247,5	5.657,5	379,6
ES (1+5) I2 TypE	7.303,2	7.226	92,0	140,3	0,0	1.239,0	5.649,0	274,9
ES (10, 50) I2 TypE	7.261,4	7.216	45,1	131,4	0,0	1.227,5	5.637,5	265,0
ES (10+ 50) I2 TypE	7.251,3	7.203	33,4	130,7	0,0	1.222,0	5.632,0	266,6
ES (30, 200) I2 TypE	7.249,1	7.214	36,0	127,2	0,0	1.224,5	5.634,5	262,9
ES (30+ 200) I2 TypE	7.239,2	7.192	32,7	124,5	0,0	1.221,0	5.631,0	262,7
ES (1,5) I2 TypE + R	8.688,9	8.616	73,8	447,7	0,0	1.185,0	5.595,0	1.461,2
ES (1+5) I2 TypE + R	9.146,4	9.054	38,6	531,0	0,0	1.185,5	5.595,5	1.834,4
ES (10,50) I2 TypE + R	8.679,3	8.474	209,9	434,0	0,0	1.185,5	5.595,5	1.464,3
ES (10+50) I2 TypE + R	9.121,2	8.997	52,3	519,9	0,0	1.185,0	5.595,0	1.821,3
ES (30,200) I2 TypE + R	8.902,4	8.525	150,7	472,8	0,0	1.185,0	5.595,0	1.649,6
ES (30+200) I2 TypE + R	9.128,0	8.910	58,6	524,4	0,0	1.185,0	5.595,0	1.823,6

Tabelle 7.27: Ergebnisse bei Logistik_S_Fr (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	⊙	Minimum	Std.-abw.					
Manueller Plan	8.100,0	8.100	-	0,0	0,0	2.400,0	1.950,0	3.750,0
MAS	6.162,0	6.076	55,8	231,9	0,0	750,0	4.050,0	1.130,1
HC I1	5.370,0	5.370	0,0	129,0	0,0	780,0	4.080,0	381,0
HC I2	5.369,8	5.364	1,1	129,0	0,0	780,0	4.080,0	380,8
PSO (20) I3 gBest	5.354,2	5.298	41,4	106,1	0,0	854,0	4.154,0	240,1
PSO (200) I3 gBest	5.366,4	5.323	34,0	102,3	0,0	863,5	4.163,5	237,1
PSO (20) I3 Wheel	5.489,1	5.432	42,7	145,0	0,0	865,0	4.165,0	313,8
PSO (200) I3 Wheel	6.455,7	6.232	121,2	136,4	0,0	1.391,5	4.691,5	236,8
PSO (20) I3 Circle	5.347,8	5.284	39,0	107,5	0,0	848,5	4.148,5	243,3
PSO (200) I3 Circle	5.332,9	5.238	29,9	109,2	0,0	838,5	4.138,5	246,7
PSO (20) I3 lBest	5.350,2	5.274	39,2	104,0	0,0	853,0	4.153,0	240,2
PSO (200) I3 lBest	5.321,4	5.287	20,0	105,5	0,0	836,5	4.136,5	242,9
PSO (20) I3 gBest + R	6.361,5	6.090	133,8	373,8	0,0	750,0	4.050,0	1.187,7
PSO (200) I3 gBest + R	6.472,9	6.295	104,9	383,8	0,0	750,0	4.050,0	1.289,1
PSO (20) I3 Wheel + R	6.993,4	6.891	85,8	490,4	0,0	750,0	4.050,0	1.703,0
PSO (200) I3 Wheel + R	7.237,5	7.137	67,2	540,0	0,0	756,0	4.056,0	1.885,5
PSO (20) I3 Circle + R	6.059,1	5.936	52,5	302,7	0,0	750,0	4.050,0	956,4
PSO (200) I3 Circle + R	6.181,2	6.095	40,5	326,5	0,0	750,0	4.050,0	1.054,7
PSO (20) I3 lBest + R	6.064,9	5.963	46,8	302,5	0,0	750,0	4.050,0	962,4
PSO (200) I3 lBest + R	6.054,8	5.988	38,6	298,5	0,0	750,0	4.050,0	956,3
ES (1,5) I2 TypG	5.504,6	5.424	40,9	205,4	0,0	762,5	4.062,5	474,2
ES (1+5) I2 TypG	5.561,5	5.456	51,4	225,8	0,0	764,0	4.064,0	507,7
ES (10,50) I2 TypG	5.622,6	5.527	42,6	239,1	0,0	768,0	4.068,0	547,5
ES (10+50) I2 TypG	5.615,8	5.543	54,4	240,3	0,0	764,5	4.064,5	546,5
ES (30,200) I2 TypG	5.628,6	5.539	55,0	244,3	0,0	757,5	4.057,5	569,3
ES (30+200) I2 TypG	5.636,9	5.530	48,6	243,7	0,0	764,0	4.064,0	565,2

Tabelle 7.28: Ergebnisse bei Logistik_S_Fr (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammapbeitsplatz
	\emptyset	Minimum	Std.-abw.					
ES (1,5) I2 TypG + R	6.722,4	6.669	33,1	449,8	0,0	750,0	4.050,0	1.472,6
ES (1+5) I2 TypG + R	7.360,1	7.190	86,8	556,4	0,0	755,5	4.055,5	1.992,7
ES (10,50) I2 TypG + R	6.051,3	5.961	42,7	302,9	0,0	750,0	4.050,0	948,4
ES (10+50) I2 TypG + R	7.068,4	6.898	111,6	494,1	0,0	753,0	4.053,0	1.768,3
ES (30,200) I2 TypG + R	6.111,5	6.005	48,9	309,3	0,0	750,0	4.050,0	1.002,2
ES (30+200) I2 TypG + R	6.839,0	6.653	106,7	451,9	0,0	750,0	4.050,0	1.587,1
ES (1,5) I2 TypE	6.796,7	5.741	1.525,0	419,0	0,0	1.028,5	4.328,5	1.020,7
ES (1+5) I2 TypE	6.278,4	5.687	782,9	357,8	0,0	879,0	4.179,0	862,6
ES (10,50) I2 TypE	5.785,2	5.613	213,8	285,7	0,0	767,5	4.067,5	664,5
ES (10+50) I2 TypE	5.770,1	5.634	154,0	282,2	0,0	762,5	4.062,5	662,9
ES (30,200) I2 TypE	5.754,7	5.593	70,9	282,6	0,0	755,0	4.055,0	662,1
ES (30+200) I2 TypE	5.743,2	5.577	112,7	275,1	0,0	759,0	4.059,0	650,1
ES (1,5) I2 TypE + R	6.942,2	6.742	199,2	482,1	0,0	761,0	4.061,0	1.638,1
ES (1+5) I2 TypE + R	7.448,2	7.286	103,5	572,0	0,0	770,5	4.070,5	2.035,2
ES (10,50) I2 TypE + R	6.581,4	6.075	356,8	410,2	0,0	752,5	4.052,5	1.366,2
ES (10+50) I2 TypE + R	7.304,5	7.181	63,7	546,8	0,0	755,5	4.055,5	1.946,7
ES (30,200) I2 TypE + R	6.619,2	6.300	215,0	408,1	0,0	750,0	4.050,0	1.411,1
ES (30+200) I2 TypE + R	7.121,3	6.950	79,4	517,8	0,0	751,5	4.051,5	1.800,5

Tabelle 7.29: Ergebnisse bei Logistik_S_Sa (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	∅	Minimum	Std.-abw.					
Manueller Plan	10.020,0	10.020	-	0,0	0,0	2.280,0	2.910,0	4.830,0
MAS	8.097,3	8.027	59,7	217,9	0,0	675,0	6.135,0	1.069,4
HC I1	7.280,0	7.280	0,0	131,0	0,0	675,0	6.135,0	339,0
HC I2	7.280,2	7.280	1,1	131,1	0,0	675,0	6.135,0	339,1
PSO (20) I3 gBest	7.234,5	7.221	7,3	141,6	0,0	675,0	6.135,0	282,9
PSO (200) I3 gBest	7.228,7	7.217	7,4	138,0	0,0	675,0	6.135,0	280,7
PSO (20) I3 Wheel	7.324,2	7.289	20,0	168,0	0,0	677,5	6.137,5	340,9
PSO (200) I3 Wheel	8.106,4	7.963	96,1	184,6	0,0	1.065,5	6.525,5	331,2
PSO (20) I3 Circle	7.231,4	7.215	8,7	139,1	0,0	675,0	6.135,0	282,3
PSO (200) I3 Circle	7.227,3	7.219	4,8	135,1	0,0	675,0	6.135,0	282,2
PSO (20) I3 lBest	7.232,5	7.216	5,9	140,6	0,0	675,0	6.135,0	281,9
PSO (200) I3 lBest	7.225,5	7.213	6,1	134,8	0,0	675,0	6.135,0	280,7
PSO (20) I3 gBest + R	8.261,0	8.042	152,8	362,9	0,0	675,0	6.135,0	1.088,1
PSO (200) I3 gBest + R	8.339,1	8.156	119,7	363,3	0,0	675,0	6.135,0	1.165,8
PSO (20) I3 Wheel + R	8.811,5	8.710	58,2	467,5	0,0	675,0	6.135,0	1.534,0
PSO (200) I3 Wheel + R	8.974,8	8.906	36,1	486,4	0,0	675,0	6.135,0	1.678,4
PSO (20) I3 Circle + R	7.883,9	7.805	40,7	276,3	0,0	675,0	6.135,0	797,6
PSO (200) I3 Circle + R	8.007,3	7.954	33,3	304,4	0,0	675,0	6.135,0	892,9
PSO (20) I3 lBest + R	7.876,0	7.815	41,3	272,9	0,0	675,0	6.135,0	793,1
PSO (200) I3 lBest + R	7.861,8	7.809	35,2	268,7	0,0	675,0	6.135,0	783,1
ES (1,5) I2 TypG	7.259,8	7.235	9,9	135,3	0,0	675,0	6.135,0	314,5
ES (1+5) I2 TypG	7.228,9	7.219	4,9	139,9	0,0	675,0	6.135,0	279,0
ES (10,50) I2 TypG	7.227,4	7.209	7,3	138,4	0,0	675,0	6.135,0	279,0
ES (10+50) I2 TypG	7.227,3	7.218	5,5	138,3	0,0	675,0	6.135,0	279,0
ES (30,200) I2 TypG	7.216,9	7.204	6,0	127,9	0,0	675,0	6.135,0	279,0
ES (30+200) I2 TypG	7.218,6	7.206	7,7	129,4	0,0	675,0	6.135,0	279,2

Tabelle 7.30: Ergebnisse bei Logistik_S_Sa (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	\emptyset	Minimum	Std.-abw.					
ES (1,5) I2 TypG + R	8.621,3	8.542	28,4	434,0	0,0	675,0	6.135,0	1.377,3
ES (1+5) I2 TypG + R	9.054,6	8.986	32,8	498,1	0,0	675,0	6.135,0	1.746,5
ES (10,50) I2 TypG + R	8.440,4	7.852	117,6	404,0	0,0	675,0	6.135,0	1.226,4
ES (10+50) I2 TypG + R	9.042,3	8.757	78,0	500,6	0,0	675,0	6.135,0	1.731,7
ES (30,200) I2 TypG + R	8.578,6	7.919	134,5	418,5	0,0	675,0	6.135,0	1.350,1
ES (30+200) I2 TypG + R	9.033,3	8.473	129,7	495,6	0,0	675,0	6.135,0	1.727,7
ES (1,5) I2 TypE	7.473,8	7.248	195,6	172,0	0,0	745,0	6.205,0	351,8
ES (1+5) I2 TypE	7.320,8	7.223	127,0	148,4	0,0	707,0	6.167,0	298,4
ES (10,50) I2 TypE	7.239,6	7.218	22,4	140,9	0,0	677,0	6.137,0	284,7
ES (10+50) I2 TypE	7.233,3	7.219	15,0	138,1	0,0	676,0	6.136,0	283,2
ES (30,200) I2 TypE	7.224,3	7.209	11,2	131,3	0,0	676,0	6.136,0	281,0
ES (30+200) I2 TypE	7.226,8	7.211	15,9	132,6	0,0	676,5	6.136,5	281,2
ES (1,5) I2 TypE + R	8.693,4	8.618	92,9	446,4	0,0	676,5	6.136,5	1.434,0
ES (1+5) I2 TypE + R	9.103,3	9.032	37,7	514,8	0,0	675,0	6.135,0	1.778,5
ES (10,50) I2 TypE + R	8.496,0	7.987	296,4	392,1	0,0	678,0	6.138,0	1.287,9
ES (10+50) I2 TypE + R	9.099,9	8.963	57,5	509,9	0,0	675,5	6.135,5	1.779,0
ES (30,200) I2 TypE + R	8.492,4	8.092	239,4	391,5	0,0	675,0	6.135,0	1.290,9
ES (30+200) I2 TypE + R	9.063,3	8.827	93,3	505,1	0,0	675,5	6.135,5	1.747,2

Tabelle 7.31: Ergebnisse bei Logistik_S_So (Teil 1) (Abbruchkriterium = 400.000 Lösungsevaluierungen – außer MAS mit nur einer)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	⊙	Minimum	Std.-abw.					
Manueller Plan	9.735,0	9.735	-	0,0	0,0	2.505,0	2.385,0	4.845,0
MAS	7.275,2	7.238	27,8	254,2	0,0	570,0	5.295,0	1.156,0
HC I1	6.452,0	6.452	0,0	152,0	0,0	570,0	5.295,0	435,0
HC I2	6.452,0	6.452	0,0	152,0	0,0	570,0	5.295,0	435,0
PSO (20) I3 gBest	6.333,3	6.320	9,5	140,2	0,0	570,0	5.295,0	328,1
PSO (200) I3 gBest	6.328,3	6.309	9,1	138,5	0,0	570,0	5.295,0	324,8
PSO (20) I3 Wheel	6.469,8	6.414	29,9	185,2	0,0	572,5	5.297,5	415,0
PSO (200) I3 Wheel	7.481,8	7.218	146,8	199,1	0,0	1.094,0	5.819,0	369,4
PSO (20) I3 Circle	6.333,8	6.311	8,1	140,9	0,0	570,0	5.295,0	327,9
PSO (200) I3 Circle	6.327,6	6.311	7,0	136,8	0,0	570,0	5.295,0	325,8
PSO (20) I3 lBest	6.334,0	6.317	11,3	140,0	0,0	570,0	5.295,0	329,0
PSO (200) I3 lBest	6.322,6	6.313	5,6	133,7	0,0	570,0	5.295,0	323,9
PSO (20) I3 gBest + R	7.540,6	7.368	148,0	396,6	0,0	570,0	5.295,0	1.279,0
PSO (200) I3 gBest + R	7.588,1	7.458	79,9	393,2	0,0	570,0	5.295,0	1.329,9
PSO (20) I3 Wheel + R	8.019,1	7.905	45,6	492,0	0,0	570,0	5.295,0	1.662,1
PSO (200) I3 Wheel + R	8.212,6	8.067	57,2	524,7	0,0	570,0	5.295,0	1.822,9
PSO (20) I3 Circle + R	7.212,5	7.113	55,3	320,7	0,0	570,0	5.295,0	1.026,8
PSO (200) I3 Circle + R	7.331,7	7.256	36,2	344,3	0,0	570,0	5.295,0	1.122,4
PSO (20) I3 lBest + R	7.205,3	7.122	42,4	318,8	0,0	570,0	5.295,0	1.021,5
PSO (200) I3 lBest + R	7.198,2	7.141	38,0	318,6	0,0	570,0	5.295,0	1.014,6
ES (1,5) I2 TypG	6.375,4	6.350	12,4	142,9	0,0	570,0	5.295,0	367,5
ES (1+5) I2 TypG	6.323,6	6.312	6,2	137,6	0,0	570,0	5.295,0	321,0
ES (10,50) I2 TypG	6.320,0	6.300	6,9	134,0	0,0	570,0	5.295,0	321,0
ES (10+50) I2 TypG	6.320,3	6.311	5,1	134,3	0,0	570,0	5.295,0	321,0
ES (30,200) I2 TypG	6.310,6	6.300	6,6	124,6	0,0	570,0	5.295,0	321,0
ES (30+200) I2 TypG	6.309,3	6.299	6,6	123,3	0,0	570,0	5.295,0	321,0

Tabelle 7.32: Ergebnisse bei Logistik_S_So (Teil 2) (Abbruchkriterium = 400.000 Lösungsevaluationen)

Verfahren	Fehler			Arbeitsplatzwechsel	Qualifikationsfehler in Min.	Unterdeckung in Min.	Überdeckung in Min.	Falscher Stammarbeitsplatz
	\emptyset	Minimum	Std.-abw.					
ES (1,5) I2 TypG + R	7.831,8	7.756	32,0	458,9	0,0	570,0	5.295,0	1.507,9
ES (1+5) I2 TypG + R	8.285,6	8.164	39,6	538,4	0,0	571,0	5.296,0	1.880,2
ES (10,50) I2 TypG + R	7.625,2	7.145	137,4	418,9	0,0	570,0	5.295,0	1.341,3
ES (10+50) I2 TypG + R	8.241,7	7.994	88,6	529,1	0,0	570,0	5.295,0	1.847,6
ES (30,200) I2 TypG + R	7.775,0	7.212	173,1	440,7	0,0	570,0	5.295,0	1.469,3
ES (30+200) I2 TypG + R	8.232,2	7.764	130,7	524,2	0,0	570,5	5.295,5	1.842,0
ES (1,5) I2 TypE	6.700,3	6.373	323,0	222,1	0,0	625,0	5.350,0	503,2
ES (1+5) I2 TypE	6.448,8	6.324	232,1	156,4	0,0	612,0	5.337,0	343,4
ES (10,50) I2 TypE	6.343,6	6.323	30,2	143,6	0,0	573,5	5.298,5	328,0
ES (10+50) I2 TypE	6.342,7	6.322	13,3	145,5	0,0	571,0	5.296,0	330,2
ES (30,200) I2 TypE	6.322,1	6.306	9,0	132,2	0,0	570,0	5.295,0	324,9
ES (30+200) I2 TypE	6.323,9	6.310	8,6	132,9	0,0	570,0	5.295,0	326,0
ES (1,5) I2 TypE + R	7.913,9	7.814	103,9	473,8	0,0	572,5	5.297,5	1.570,1
ES (1+5) I2 TypE + R	8.323,6	8.208	49,6	541,2	0,0	573,0	5.298,0	1.911,4
ES (10,50) I2 TypE + R	7.669,8	7.282	242,5	412,9	0,0	570,0	5.295,0	1.391,9
ES (10+50) I2 TypE + R	8.228,4	8.150	52,7	527,4	0,0	570,5	5.295,5	1.835,0
ES (30,200) I2 TypE + R	7.685,4	7.269	204,8	406,1	0,0	570,0	5.295,0	1.414,3
ES (30+200) I2 TypE + R	8.120,4	7.939	80,6	497,6	0,0	570,5	5.295,5	1.756,8

Tabelle 7.33: Ergebnisse bei Handel_1AP (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS und KV 1 mit nur einer; bei KV 2 ist die Anzahl unbekannt)

Verfahren	Fehler			Unterd. in Min.	Überd. in Min. (Bedarf>0)	Überschr. Sollzeit in Min.	AZM < 3 h	AZM > 9 h	> 1 AZM am Tag
	∅	Minimum	Std.-abw.						
MAS	16.694,0	15.912	423,7	6.048,0	0,0	10.646,0	0,0	0,0	0,0
KV 1	27.660,0	27.660	0,0	2.280,0	2.160,0	23.220,0	0,0	0,0	0,0
KV 2	25.770,0	25.770	0,0	3.420,0	240,0	22.110,0	0,0	0,0	0,0
HC I1	1.475.538,8	1.268.852	133.661,5	21.332,0	26.948,0	9.858,8	359,2	66,7	141,7
PSO (20) I1 gBest + R	9.006,4	5.976	2.058,3	3.326,0	0,0	5.680,4	0,0	0,0	0,0
PSO (40) I1 gBest + R	9.740,8	6.144	2.104,9	3.578,0	4,0	6.158,8	0,0	0,0	0,0
PSO (100) I1 gBest + R	9.928,4	7.536	1.767,5	3.578,0	4,0	6.346,4	0,0	0,0	0,0
PSO (200) I1 gBest + R	10.037,6	7.548	1.784,4	3.560,0	2,0	6.475,6	0,0	0,0	0,0
PSO (20) I1 Wheel + R	10.014,0	6.504	1.956,8	3.606,0	0,0	6.404,0	0,0	0,0	0,0
PSO (20) I1 Circle + R	13.035,6	9.552	1.746,1	4.034,0	0,0	9.001,6	0,0	0,0	0,0
PSO (20) I1 lBest + R	12.612,8	9.036	2.056,0	4.108,0	0,0	8.502,8	0,0	0,0	0,0
ES (1,5) I1 TypG + R	4.910,0	3.840	425,7	2.178,0	0,0	2.881,2	0,0	0,0	0,0
ES (1+5) I1 TypG + R	8.828,8	4.320	2.786,0	3.496,0	2,0	5.330,8	0,0	0,0	0,0
ES (10,50) I1 TypG + R	5.996,4	4.836	656,2	2.616,0	0,0	3.380,4	0,0	0,0	0,0
ES (10+50) I1 TypG + R	17.440,4	6.132	11.405,3	4.328,0	4,0	13.108,4	0,0	0,0	0,0
ES (1,5) I1 TypE + R	8257,6	5316	2416,9	3272,0	6,0	4979,6	0,0	0,0	0,0
ES (1+5) I1 TypE + R	9809,6	4980	2827,0	3600,0	0,0	6209,6	0,0	0,0	0,0
ES (10,50) I1 TypE + R	8174,8	5136	1464,1	3278,0	0,0	4896,8	0,0	0,0	0,0
ES (10+50) I1 TypE + R	15316,8	7296	6638,5	4244,0	2,0	11070,8	0,0	0,0	0,0
ES (30,200) I1 TypE + R	7278,0	5040	1209,3	3118,0	2,0	4158,0	0,0	0,0	0,0
ES (30+200) I1 TypE + R	11503,6	6000	3608,0	3666,0	4,0	7833,6	0,0	0,0	0,0

Tabelle 7.34: Ergebnisse bei Handel_1AP_Jan (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS und KV 1 mit nur einer; bei KV 2 ist die Anzahl unbekannt)

Verfahren	Fehler			Unterd. in Min.	Überd. in Min. (Bedarf>0)	Überschr. Sollzeit in Min.	AZM < 3 h	AZM > 9 h	> 1 AZM am Tag
	∅	Minimum	Std.-abw.						
MAS	846,0	600	195,1	252,0	0,0	594,0	0,0	0,0	0,0
KV 1	1.920,0	1.920	0,0	1.200,0	240,0	480,0	0,0	0,0	0,0
KV 2	3.300,0	3.300	0,0	900,0	180,0	2.220,0	0,0	0,0	0,0
HC I1	90.507,6	23.780	27.908,5	2.120,0	748,0	639,6	0,0	0,0	8,7
PSO (20) I1 gBest + R	443,2	420	45,2	82,0	0,0	361,2	0,0	0,0	0,0
PSO (40) I1 gBest + R	442,0	420	36,9	80,0	0,0	362,0	0,0	0,0	0,0
PSO (100) I1 gBest + R	450,0	420	49,2	82,0	0,0	368,0	0,0	0,0	0,0
PSO (200) I1 gBest + R	450,0	420	43,9	74,0	0,0	376,0	0,0	0,0	0,0
PSO (20) I1 Wheel + R	440,0	420	36,4	72,0	0,0	368,0	0,0	0,0	0,0
PSO (20) I1 Circle + R	420,0	420	0,0	52,0	0,0	368,0	0,0	0,0	0,0
PSO (20) I1 lBest + R	435,6	420	29,6	100,0	0,0	335,6	0,0	0,0	0,0
ES (1,5) I1 TypG + R	430,0	420	22,7	74,0	0,0	356,0	0,0	0,0	0,0
ES (1+5) I1 TypG + R	460,0	420	144,7	92,0	0,0	368,0	0,0	0,0	0,0
ES (10,50) I1 TypG + R	442,0	420	43,1	80,0	0,0	362,0	0,0	0,0	0,0
ES (10+50) I1 TypG + R	463,2	420	76,1	104,0	0,0	359,2	0,0	0,0	0,0
ES (30,200) I1 TypG + R	478,0	420	112,0	86,0	0,0	392,0	0,0	0,0	0,0
ES (30+200) I1 TypG + R	485,2	420	108,6	80,0	0,0	405,2	0,0	0,0	0,0
ES (1,5) I1 TypE + R	449,2	420	81,3	80,0	0,0	369,2	0,0	0,0	0,0
ES (1+5) I1 TypE + R	468,0	420	117,3	100,0	0,0	368,0	0,0	0,0	0,0
ES (10,50) I1 TypE + R	462,0	420	59,3	36,0	0,0	426,0	0,0	0,0	0,0
ES (10+50) I1 TypE + R	448,0	420	43,8	82,0	0,0	366,0	0,0	0,0	0,0
ES (30,200) I1 TypE + R	512,0	420	162,0	102,0	0,0	410,0	0,0	0,0	0,0
ES (30+200) I1 TypE + R	514,0	420	142,4	84,0	0,0	430,0	0,0	0,0	0,0

Tabelle 7.35: Ergebnisse bei Handel_2AP (Abbruchkriterium = 400.000 Lösungs-evaluationen – außer MAS mit nur einer; bei KV 2 ist die Anzahl unbekannt)

Verfahren	Fehler			Arbeitsplatzwechsel	Unterd. in Min.	Überd. in Min. (Bedarf>0)	Überschr. Sollzeit in Min.	> 1 AZM am Tag
	∅	Minimum	Std.-abw.					
MAS	46.339,6	43.866	1.304,1	262,8	7.316,0	0,0	38.761,2	0,0
KV 2	84.690,0	84.690	0,0	0,0	1.500,0	0,0	83.190,0	0,0
HC I1	4.265.931,9	3.807.375	191.944,4	2.434,5	12.568,0	85.246,0	57.350,0	410,8
PSO (20) I1 gBest + R	37.117,9	14.385	11.808,9	389,9	834,0	20,0	35.874,0	0,0
PSO (40) I1 gBest + R	40.583,7	14.446	12.832,3	402,5	966,0	26,0	39.189,2	0,0
PSO (100) I1 gBest + R	40.710,8	20.998	14.738,0	403,2	874,0	22,0	39.411,6	0,0
PSO (200) I1 gBest + R	51.879,5	27.967	12.578,4	463,9	1.142,0	12,0	50.261,6	0,0
PSO (20) I1 Wheel + R	70.646,9	32.251	14.193,9	521,3	1.312,0	28,0	68.785,6	0,0
PSO (20) I1 Circle + R	62.748,6	37.685	14.942,1	407,0	960,0	24,0	61.357,6	0,0
PSO (20) I1 lBest + R	64.351,6	38.068	19.503,8	402,4	1.094,0	28,0	62.827,2	0,0
ES (1,5) I1 TypG + R	8.267,1	5.924	1.265,8	214,3	834,0	8,0	7.210,8	0,0
ES (1+5) I1 TypG + R	47.198,1	13.720	20.486,4	444,5	1.870,0	28,0	44.855,6	0,0
ES (10,50) I1 TypG + R	17.528,5	8.459	4.094,2	247,3	1.170,0	12,0	16.099,2	0,0
ES (10+50) I1 TypG + R	49.794,5	24.609	26.010,7	451,7	2.022,0	24,0	47.296,8	0,0
ES (30,200) I1 TypG + R	22.222,7	13.579	5.780,6	283,9	1.130,0	10,0	20.798,8	0,0
ES (30+200) I1 TypG + R	39.491,5	25.706	14.801,5	460,7	1.720,0	12,0	37.298,8	0,0
ES (1,5) I1 TypE + R	8.464,8	4.911	2.010,1	248,0	842,0	8,0	7.366,8	0,0
ES (1+5) I1 TypE + R	46.444,5	20.019	16.144,0	463,6	1.676,0	4,0	29.545,0	0,0
ES (10,50) I1 TypE + R	19.206,3	6.336	5.491,9	263,9	1.276,0	6,0	17.660,4	0,0
ES (10+50) I1 TypE + R	36.116,8	19.072	11.601,9	447,6	1.706,0	10,0	23.020,7	0,0
ES (30,200) I1 TypE + R	20.897,6	15.043	4.659,3	302,4	1.306,0	16,0	19.273,2	0,0
ES (30+200) I1 TypE + R	30.866,3	17.848	8.711,1	469,3	1.656,0	8,0	20.843,5	0,0

Tabelle 7.36: Ergebnisse bei Handel_2AP_Jan (Abbruchkriterium = 400.000 Lösungsevaluationen – außer MAS mit nur einer; bei KV 2 ist die Anzahl unbekannt)

Verfahren	Fehler			Arbeitsplatzwechsel	Unterd. in Min.	Überd. in Min. (Bedarf>0)	Überschr. Sollzeit in Min.	> 1 AZM am Tag
	∅	Minimum	Std.-abw.					
MAS	3084,5	2237	455,5	11,3	300,0	0,0	2773,2	0,0
KV 2	9480,0	9480	0,0	0,0	300,0	0,0	9180,0	0,0
HC I1	22086,7	2940	15122,0	0,0	1206,0	28,0	2186,0	1,9
PSO (20) I1 gBest + R	664,2	86	318,3	15,0	52,0	2,0	595,2	0,0
PSO (40) I1 gBest + R	774,5	198	415,5	16,9	88,0	0,0	669,6	0,0
PSO (100) I1 gBest + R	971,8	263	381,5	20,6	70,0	2,0	879,2	0,0
PSO (200) I1 gBest + R	1071,6	260	571,6	23,2	78,0	0,0	970,4	0,0
PSO (20) I1 Wheel + R	698,2	257	268,0	15,4	34,0	0,0	648,8	0,0
PSO (20) I1 Circle + R	852,3	319	242,5	21,9	58,0	2,0	770,4	0,0
PSO (20) I1 lBest + R	964,3	256	335,7	20,3	84,0	6,0	854,0	0,0
ES (1,5) I1 TypG + R	158,0	10	107,8	12,8	50,0	0,0	95,2	0,0
ES (1+5) I1 TypG + R	1526,6	68	1077,7	19,0	82,0	4,0	1421,6	0,0
ES (10,50) I1 TypG + R	760,0	250	351,4	13,2	80,0	2,0	664,8	0,0
ES (10+50) I1 TypG + R	1462,7	309	936,0	17,1	96,0	0,0	1349,6	0,0
ES (30,200) I1 TypG + R	904,0	195	337,5	15,6	84,0	4,0	800,4	0,0
ES (30+200) I1 TypG + R	1705,1	311	1130,6	21,1	96,0	4,0	1584,0	0,0
ES (1,5) I1 TypE + R	704,9	136	354,1	19,3	60,0	0,0	625,6	0,0
ES (1+5) I1 TypE + R	996,4	198	578,2	18,0	64,0	4,0	910,4	0,0
ES (10,50) I1 TypE + R	1077,6	351	640,5	18,0	78,0	0,0	981,6	0,0
ES (10+50) I1 TypE + R	1137,8	442	481,1	20,2	74,0	0,0	1043,6	0,0
ES (30,200) I1 TypE + R	927,6	191	466,4	18,0	54,0	0,0	855,6	0,0
ES (30+200) I1 TypE + R	1003,7	252	654,5	18,5	72,0	0,0	913,2	0,0

Tabelle 7.37: Ergebnisse bei BT_Mod-250-118 (Abbruchkriterium bei PSO und ES = 20.000.000 Lösungs Evaluationen; bei MAS im Durchschnitt = 142.962 Lösungs Evaluationen; bei KV 3 = 1 Lösungs Evaluation; bei den anderen Verfahren ist die Anzahl unbekannt)

Verfahren	Fehler			Fahrtkosten	Fehler offene Aufträge	Anzahl offene Aufträge	Anzahl untätige Techniker	CPU-Zeit in Sek.	Quellen
	∅	Minimum	Std.-abw.						
GA	-	23.790	-	-	-	67,0	-	-	[277]
GA + R	-	22.570	-	-	-	54,0	-	-	[277]
SA	-	21.050	-	4.390,0	16.660,0	56,0	-	-	[20]
HC	-	20.788	-	4.604,0	16.184,0	50,0	-	20.056,0	[383]
FLS	-	20.732	-	4.608,0	16.124,0	49,0	-	1.242,0	[383]
Fast GLS	-	20.433	-	4.707,0	15.726,0	48,0	-	9.183,0	[383]
CLP TG + R	-	22.241	-	5.269,0	16.972,0	48,0	-	600,0	[18]
CLP CG + R	-	21.292	-	4.902,0	16.390,0	53,0	-	600,0	[18]
CLP SD	-	22.107	-	4.648,0	17.459,0	57,0	-	86,0	[428]
CLP CL	-	23.576	-	6.029,0	17.547,0	56,0	-	14,0	[428]
CLP LB	-	26.096	-	8.439,0	17.657,0	57,0	-	23,0	[428]
CLP SD + FC	-	20.981	-	4.716,0	16.220,0	54,0	-	97,0	[428]
CLP CL + FC	-	23.094	-	6.211,0	16.883,0	57,0	-	20,0	[428]
CLP LB + FC	-	24.995	-	8.371,0	16.624,0	55,0	-	35,0	[428]
MAS	23.494,0	22.977	325,0	4.133,9	19.360,1	70,3	51,0	39,0	-
KV 3	44.007,7	42.256	861,0	2.521,3	41.486,4	173,8	63,4	0,1	-
PSO (10) gBest + R	20.585,5	20.371	164,8	4.221,5	16.363,9	55,9	52,1	10.306,0	-
PSO (200) gBest + R	21.184,1	20.958	112,2	4.374,4	16.809,7	60,2	50,4	10.607,3	-
PSO (10) Wheel + R	20.637,7	20.340	162,2	4.169,5	16.468,2	56,6	53,4	10.668,0	-
PSO (10) Circle + R	20.505,4	20.273	112,8	4.185,0	16.320,4	55,0	53,4	10.572,0	-
PSO (10) lBest + R	20.435,9	20.193	130,9	4.168,0	16.267,8	54,5	53,4	10.614,7	-
ES (1,5) TypG + R	26.078,7	25.687	167,2	5.085,3	20.993,4	99,6	38,0	44.225,9	-
ES (1+5) TypG + R	21.288,8	20.874	239,3	4.393,5	16.895,3	59,5	49,2	43.533,1	-
ES (10,50) TypG + R	24.086,6	23.848	121,4	4.908,1	19.178,5	89,6	45,1	40.525,0	-
ES (10+50) TypG + R	21.384,6	20.783	277,3	4.334,5	17.050,1	60,6	51,2	41.935,0	-
ES (30,200) TypG + R	23.898,6	23.277	282,5	4.735,6	19.163,1	88,2	48,5	36.670,0	-
ES (30+200) TypG + R	21.327,8	20.912	294,9	4.357,2	16.970,6	61,8	51,3	38.600,0	-
ES (1,5) TypE + R	23.371,5	22.689	304,9	5.228,2	18.143,3	71,6	39,8	47.280,0	-
ES (1+5) TypE + R	23.624,5	22.517	520,9	5.131,2	18.493,2	70,7	42,0	49.850,0	-
ES (10,50) TypE + R	23.177,6	21.794	755,3	4.815,4	18.362,2	70,5	45,3	45.250,0	-
ES (10+50) TypE + R	22.828,4	21.336	805,2	4.791,1	18.037,3	68,2	44,8	46.085,0	-
ES (30,200) TypE + R	22.834,2	21.953	537,9	4.776,7	18.057,6	70,0	45,8	41.925,0	-
ES (30+200) TypE + R	22.723,7	22.078	374,9	4.712,0	18.011,7	69,2	45,9	42.635,0	-

Tabelle 7.38: Ergebnisse bei BT_Rel-250-118 (Anzahl Lösungsevaluationen unbekannt)

Verfahren	Fehler			Fahrtkosten	Fehler offene Aufträge	Anzahl offene Aufträge	Anzahl untätige Techniker	CPU-Zeit in Sek.	Quellen
	∅	Minimum	Std.-abw.						
CLP SD	-	27.622	-	3.783,0	23.839,0	82,0	-	70,0	[428]
CLP CL	-	28.816	-	4.768,0	24.048,0	83,0	-	12,0	[428]
CLP LB	-	31.514	-	7.200,0	24.314,0	83,0	-	20,0	[428]
CLP SD + FC	-	21.426	-	4.800,0	16.626,0	54,0	50,0	95,0	[428]
CLP CL + FC	-	23.563	-	5.846,0	14.177,0	59,0	51,0	20,0	[428]
CLP LB + FC	-	24.910	-	7.983,0	13.627,0	55,0	6,0	35,0	[428]
D-CLP1 (B) SD + FC	-	23.666	-	5.585,0	14.181,0	65,0	51,0	31,6	[217]
D-CLP2 (B) CL + FC	-	23.775	-	-	-	65,0	51,0	7,6	[218]
D-CLP2 (B) SD + FC	-	22.078	-	4.731,0	17.347,0	63,0	51,0	15,2	[218]
D-CLP2 (M) CL + FC	-	25.099	-	-	-	72,0	53,0	8,1	[218]
D-CLP2 (M) SD + FC	-	23.202	-	-	-	69,0	54,0	14,0	[218]
D-CLP3 (B) SD + FC	-	21.948	-	-	-	60,0	51,0	-	[330]

Tabelle 7.39: Ergebnisse bei BT_Tsang-250-118 (Anzahl Lösungsevaluationen unbekannt)

Verfahren	Fehler			Fahrtkosten	Fehler offene Aufträge	Fehler für Überstunden	Anzahl offene Aufträge	Anzahl untätige Techniker	CPU-Zeit in Sek.	Quellen
	∅	Minimum	Std.-abw.							
HC	-	20.124	-	4.595,0	15.358,0	171,0	48,0	-	25.195,0	[383]
FLS	-	20.224	-	4.651,0	15.448,0	125,0	51,0	-	1.244,0	[383]
Fast GLS	-	19.997	-	4.648,0	15.209,0	140,0	49,0	-	4.102,0	[383]

Tabelle 7.40: Ergebnisse bei BT_Bristol-250-118 (Anzahl Lösungsevaluationen unbekannt)

Verfahren	Fehler			Fahrtkosten	Fehler offene Aufträge	Anzahl offene Aufträge	Anzahl untätige Techniker	CPU-Zeit in Sek.	Quellen
	∅	Minimum	Std.-abw.						
CLP SD	-	29.546	-	26.077,0	3.469,0	89,0	-	64,0	[428]
CLP CL	-	30.615	-	25.969,0	4.646,0	89,0	-	9,0	[428]
CLP LB	-	33.044	-	26.136,0	6.908,0	89,0	-	18,0	[428]
CLP SD + FC	-	19.405	-	14.016,0	5.389,0	48,0	-	100,0	[428]
CLP CL + FC	-	20.620	-	13.799,0	6.821,0	48,0	-	19,0	[428]
CLP LB + FC	-	22.884	-	14.593,0	8.291,0	50,0	-	30,0	[428]