

Forschungsberichte aus dem
wbk Institut für Produktionstechnik
Karlsruher Institut für Technologie (KIT)

Constantin Hofmann

**Vorausschauende und reaktive
Mehrzieloptimierung für
die Produktionssteuerung
einer Matrixproduktion**

Band 250

Forschungsberichte aus dem
wbk Institut für Produktionstechnik
Karlsruher Institut für Technologie (KIT)

Hrsg.: Prof. Dr.-Ing. Jürgen Fleischer
Prof. Dr.-Ing. Gisela Lanza
Prof. Dr.-Ing. habil. Volker Schulze

Constantin Carl Hofmann

**Vorausschauende und reaktive Mehrzieloptimierung für die
Produktionssteuerung einer Matrixproduktion**

Band 250

Vorausschauende und reaktive Mehrzieloptimierung für die Produktionssteuerung einer Matrixproduktion

Zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)
von der KIT-Fakultät für Maschinenbau des
Karlsruher Institut für Technologie (KIT)

angenommene

DISSERTATION

von

Constantin Carl Hofmann
aus Frankfurt am Main

Tag der mündlichen Prüfung: 07.10.2021

Hauptreferentin: Prof. Dr.-Ing. Gisela Lanza

Korreferent: Prof. Dr. techn. Christian Ramsauer

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Karlsruhe, Karlsruher Institut für Technologie, Diss., 2021

Copyright Shaker Verlag 2021

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-8316-3

ISSN 0724-4967

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren
Telefon: 02421 / 99 0 11 - 0 • Telefax: 02421 / 99 0 11 - 9
Internet: www.shaker.de • E-Mail: info@shaker.de

Vorwort der Herausgeber

Die schnelle und effiziente Umsetzung innovativer Technologien wird vor dem Hintergrund der Globalisierung der Wirtschaft der entscheidende Wirtschaftsfaktor für produzierende Unternehmen. Universitäten können als “Wertschöpfungspartner” einen wesentlichen Beitrag zur Wettbewerbsfähigkeit der Industrie leisten, indem sie wissenschaftliche Grundlagen sowie neue Methoden und Technologien erarbeiten und aktiv den Umsetzungsprozess in die praktische Anwendung unterstützen.

Vor diesem Hintergrund wird im Rahmen dieser Schriftenreihe über aktuelle Forschungsergebnisse des Instituts für Produktionstechnik (wbk) am Karlsruher Institut für Technologie (KIT) berichtet. Unsere Forschungsarbeiten beschäftigen sich sowohl mit der Leistungssteigerung von additiven und subtraktiven Fertigungsverfahren, den Produktionsanlagen und der Prozessautomatisierung sowie mit der ganzheitlichen Betrachtung und Optimierung der Produktionssysteme und -netzwerke. Hierbei werden jeweils technologische wie auch organisatorische Aspekte betrachtet.

Prof. Dr.-Ing. Jürgen Fleischer

Prof. Dr.-Ing. Gisela Lanza

Prof. Dr.-Ing. habil. Volker Schulze

Vorwort des Verfassers

Diese Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am wbk Institut für Produktionstechnik des Karlsruher Instituts für Technologie (KIT).

Ich möchte mich besonders bei Prof. Dr.-Ing. Gisela Lanza für die Betreuung und Begleitung dieser Arbeit in den letzten Jahren bedanken. Mein Dank gilt insbesondere auch für ihr entgegengebrachtes Vertrauen und den großen Freiraum, den ich in dieser Zeit am Institut genießen durfte. Die familiäre, inspirierende und durch persönliches Engagement geprägte Atmosphäre des wbks habe ich sehr geschätzt. Mein herzliches Dankeschön gilt Prof. Dr. techn. Christian Ramsauer für die Übernahme des Korreferats. Der Projektgruppe der SAP möchte ich für den Rahmen und die wertvollen Diskussionen danken, die zum Gelingen dieser Arbeit beigetragen haben.

Bei allen Kolleginnen und Kollegen am wbk möchte ich mich für den außergewöhnlichen Zusammenhalt und das freundschaftliche Arbeitsumfeld ganz herzlich bedanken. Insbesondere möchte ich Tom Stähr, Andreas Kuhnle, Dietrich Berger, Robin Kopf, Bastian Verhaelen, Niclas Eschner, Daniel Gauder, Raphael Wagner, Sina Peukert und Carmen Krahe nennen. Besonderen Respekt zolle ich den Studierenden, die durch ihr Engagement, ihren Erfindungsgeist und ihr Durchhaltevermögen diese Arbeit geprägt haben.

Abschließend möchte ich mich bei meinem Bruder Valentin, meinen Eltern, Großeltern und vor allem bei meiner Freundin Priscilla für den Rückhalt und die tatkräftige Unterstützung bedanken.

Karlsruhe, 23. Juli 2021

Constantin Carl Hofmann

Abstract

An increasingly diverse production program with uncertain quantities makes it difficult to operate production systems economically. If product individualization causes different processing times at the production stations, takt time losses occur. Fluctuations in the proportions of product variants can also lead to dynamic bottlenecks. The concept of matrix production pursues a flexibilization of the production structure by dissolving the rigid linkage, the takt time dependency and by using redundant multi-purpose stations. These measures allow production control to vary the sequence of operations within the limits of the precedence graph and to adjust the route of each job. Reactive multi-objective control is needed to take advantage of these degrees of freedom and to meet the varying objectives of production systems. By using domain knowledge in the optimization, efficiency can be increased for the specific problem. However, due to the diversity of production systems and target variables, production control should be able to adapt autonomously to the use case at hand. As the durations for cycle, transport and setup times are important input variables for scheduling, a method to determine realistic values is needed.

Due to the complexity of the control decision, heuristics are most suitable. In particular, Monte Carlo Tree Search, as an iterative search tree method, has good properties for use as a reactive production control. However, so far, approaches to meet the requirements of controlling a matrix production have been lacking.

In this thesis, a reactive multi-objective control based on MO-MCTS is developed for the production control of a matrix production considering setup and transport operations. In addition, a post optimization based on local search is integrated into the MO-MCTS flow. In order to quickly achieve high solution quality for different objectives and production systems, two methods are developed to autonomously adapt the production control. To ensure the accuracy of the durations used in scheduling, a method to derive and update the underlying distributions is presented. The detailed evaluations based on different use cases show that the production control is able to successfully optimize different objectives. The methods for autonomous adaptation also lead to a faster increase in solution quality. The comparison to optimal reference solutions and to benchmark problems from literature also prove the high solution quality. The application to a real-world scheduling scenario demonstrates the behavior of the production control regarding breakdowns and deviations. This work presents a comprehensive analysis of the production control with a thorough analysis of the influence factors on the attainability of objectives and the increase of solution quality.

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abkürzungsverzeichnis	V
Variablenverzeichnis	VII
1 Einleitung	1
1.1 Motivation und Problemstellung	1
1.2 Zielsetzung	3
1.3 Aufbau der Arbeit	3
2 Grundlagen	4
2.1 Industrielle Produktion und Produktionssysteme	4
2.1.1 Fertigungstypen und -prinzipien	5
2.1.2 Matrixproduktion	7
2.2 Produktionsplanung und -steuerung	9
2.2.1 Aufgaben und Ziele der Produktionsplanung und -steuerung	10
2.2.2 Klassifikation von Produktionssteuerungsproblemen	11
2.2.3 Ansätze zur dynamischen Produktionsplanung und -steuerung	15
2.3 Monte Carlo Tree Search	18
2.3.1 Ablauf der Monte Carlo Tree Search	19
2.3.2 Abgrenzung Monte Carlo Tree Search und Reinforcement Learning	22
2.4 Lokale Suche als Verbesserungsverfahren	23
2.4.1 Kritischer Pfad	24
2.4.2 Gerichtete Graphen als Modellierung des Job Shop Scheduling Problems für die lokale Suche	25
2.4.3 Nachbarschaftstrukturen für die Produktionsplanung und -steuerung	25
3 Stand der Forschung	28
3.1 Kriterien zur Bewertung des Stands der Forschung	28
3.2 Ansätze zur Lösung des Job Shop Scheduling Problems	29
3.2.1 Exakte Verfahren	29

3.2.2	Heuristiken	30
3.2.3	Maschinelles Lernen	35
3.2.4	Ansätze der robusten Produktionssteuerung und Verteilungsänderungsdetektion	37
3.3	Monte Carlo Tree Search in der Produktionssteuerung	41
3.4	Forschungsdefizit	47
4	Methode zur Steuerung einer Matrixproduktion	52
4.1	Problemeingrenzung, Steuerungsaufgabe und Zieldefinition	53
4.1.1	Problemeingrenzung	53
4.1.2	Steuerungsaufgabe	54
4.1.3	Steuerungsziele	54
4.1.4	Normierung der Zielgrößen	55
4.2	Ableitung realistischer Planzeiten aus Beobachtungen	57
4.2.1	Concept-Drift-Erkennung zur Detektion von Verteilungsänderungen	58
4.2.2	Verteilungsschätzung	60
4.2.3	Korrektur saisonaler Effekte	64
4.3	Multi-Objective Monte Carlo Tree Search	64
4.3.1	Modellierung als Markov-Entscheidungsproblem	64
4.3.2	Modifikationen der Selektionsphase	66
4.3.3	Modifikationen der Expansionsphase	71
4.3.4	Modifikationen der Rollout-Phase	79
4.3.5	Modifikationen der Backpropagation-Phase	80
4.4	Parallelisierung des Suchbaumaufbaus	81
4.5	Hybridisierung durch Postoptimierung mit lokaler Suche	83
4.5.1	Auswahl der Ausgangslösungen für die lokale Suche	84
4.5.2	Modellierung der Ausgangslösung als gerichteter Graph	85
4.5.3	Anpassung der Nachbarschaftsstruktur LSONE	86
4.5.4	Algorithmen der lokalen Suche	92
4.5.5	Reallokation eines Vorgangs	96
4.5.6	Abbruchkriterium der lokalen Suche	97
4.6	Prototypische Umsetzung als Software-Demonstrator	98
5	Untersuchung der Produktionssteuerung	100
5.1	Anwendungsfälle	102
5.1.1	Generisches Anwendungsproblem	102
5.1.2	Anwendungsfall der Getriebemontage	103
5.1.3	Benchmark-Problem aus der Literatur	104

5.2	Anpassung der Planzeiten basierend auf Realdaten	105
5.2.1	Ermittlung der theoretischen Verteilung	105
5.2.2	Erkennung von Verteilungsänderungen	106
5.2.3	Erkennung von Trends und Saisonalitäten	109
5.3	Lenkung des Suchfokus im mehrdimensionalen Lösungsraum	110
5.3.1	Auswirkungen der Aktionsgenerierung auf den Suchfokus	110
5.3.2	Parallelisierung zur zeitgleichen Exploration des Suchraums	122
5.3.3	Alternierende Zielgewichtung zur sequentiellen Exploration des Suchraums	123
5.3.4	Auswirkung der lokalen Suche auf den Suchfokus	125
5.3.5	Zusammenfassung	135
5.4	Anstieg der Lösungsgüte	136
5.4.1	Einfluss der Selektionsfunktion	136
5.4.2	Einfluss der intelligenten Einschränkung der Aktionen auf den Anstieg der Lösungsgüte	141
5.4.3	Einfluss der Parallelisierung auf den Anstieg der Lösungsgüte	142
5.4.4	Einfluss lokaler Suche auf den Anstieg der Lösungsgüte	144
5.4.5	Zusammenfassung	146
5.5	Bestimmung der Lösungsgüte	147
5.5.1	Vergleich mit exakten Verfahren	147
5.5.2	Vergleich anhand von Benchmark-Problemen aus der Literatur	150
5.5.3	Erprobung am Anwendungsfall der Getriebemontage	151
6	Diskussion und Ausblick	155
6.1	Diskussion	155
6.2	Ausblick	157
7	Zusammenfassung	159
	Eigene Veröffentlichungen	161
	Literaturverzeichnis	163
	Abbildungsverzeichnis	179
	Tabellenverzeichnis	184
	Anhang	XIV
	A1 Klassifikation von Produktionssteuerungsproblemen für die einstufige Fertigung	XIV
	A2 Generische Anwendungsprobleme	XVI

A3 Anwendungsfall aus der Getriebemontage	XVIII
A4 Gemischt-ganzzahliges Optimierungsproblem	XX
A5 Belegungspläne zur Demonstration des Einflusses der Aktionsgenerierung auf den Suchfokus	XXVIII
A6 Hypervolumen für das 12x5 Problem	XXX
A7 Optimale Belegungspläne für das exakte Referenzproblem	XXXI

Abkürzungsverzeichnis

ADWIN	Adaptive Windowing
AL	Approach by Localization
AMAF	All Moves As First
BPI	Menge der besten Pareto-Lösungen am Ende einer Iteration
BPA	Menge der besten Pareto-Lösungen am Ende einer Ausführung
CP	Constraint Programming
DFS	Depth First Search
DUE	Due Date Method
EPH	Evolutionary Planning Heuristic
EWMA	Exponentially Weighted Moving Average
FIFO	First In – First Out
FJSP	Flexible Job Shop Problem
FJSSP	Flexible Job Shop Scheduling Problem
FPH	Fixed Planning Heuristic
FTF	Fahrerloses Transportfahrzeug
FTS	Fahrerloses Transportsystem
GA	Genetischer Algorithmus
GLS	Guided Local Search
LS	Local Search / lokale Suche
LSONE	Local Search Of Moving One Operation

MCTS	Monte Carlo Tree Search
MES	Manufacturing Execution System
MILP	Mixed Integer Linear Programming
MTTR	Mean Time To Repair
MO-MCTS	Multi Objective Monte Carlo Tree Search
MTBF	Mean Time Between Failure
ML-Schätzer	Maximum-Likelihood-Schätzer
NSUCT	Non-dominated Sorting UCT
PPS	Produktionsplanung- und steuerung
PRS	Path Random Search
QLE	Queue Length Estimator
RAVE	Rapid Action Value Estimate
RL	Reinforcement Learning
SA	Simulated Annealing
SOMA	Sequential Operation Maschine Assignment
TS	Tabu Suche / Tabu Search
UCB-1	Upper Confidence Bound 1
UCT	Upper Confidence Bound 1 applied to trees
VND	Variable Neighborhood Descend
VNS	Variable Neighborhood Search
ZE	Zeiteinheit

Variablenverzeichnis

Symbol	Beschreibung
Abschnitt 2.2.2	Klassifikation von Produktionsproblemen
α	Maschinenumgebung
α_1	Maschinenart und Anordnung
$\alpha_1 = P$	Einstufige Fertigung: Parallele Maschinen
$\alpha_1 = Q$	Einstufige Fertigung: Uniforme, parallele Maschinen
$\alpha_1 = R$	Einstufige Fertigung: Heterogene, parallele Maschinen
$\alpha_1 = J$	Mehrstufige Fertigung: Job Shop
$\alpha_1 = F$	Mehrstufige Fertigung: Flow Shop
$\alpha_1 = O$	Mehrstufige Fertigung: Open Shop
$\alpha_1 = FJ$	Mehrstufige Fertigung: Flexible Job Shop
α_2	Fertigungsstufen
$\alpha_2 = \emptyset$	Beliebige, planungsabhängige Vorgangszahl
$\alpha_2 = MF$	Feste Vorgangszahl
β	Auftragscharakteristika
$\beta_1 = \emptyset$	Beliebige Anzahl Produktionsaufträge
$\beta_1 = K$	Konstante Anzahl Produktionsaufträge
$\beta_2 = \emptyset$	Alle Produktionsaufträge verfügbar
$\beta_2 = A_k^*$	Unterschiedliche Freigabetermine pro Auftrag k
$\beta_2 = nz_{k,m}$	Maschinen- und auftragsabhängige Nachlaufzeit nz_{km}
$\beta_3 = \emptyset$	Keine Fertigstellungstermine k
$\beta_3 = F_k^*$	Individuelle Fertigstellungstermine pro Auftrag k
$\beta_3 = \tilde{F}_k^*$	Überschreitung der Fertigstellungstermine unzulässig
$\beta_3 = T$	Fertigstellung aller Aufträge im Planungshorizont T
$\beta_4 = \emptyset$	Maximale Anzahl Prozessschritte unbegrenzt
$\beta_4 = O_k \leq O_{max}$	Maximale Anzahl Prozessschritte begrenzt durch O_{max}
$\beta_5 = \emptyset$	Produktionsauftrag nicht unterbrechbar
$\beta_5 = pmnt$	Produktionsauftrag ist unterbrechbar (<i>preemption</i>)
$\beta_6 = p$	Identische Bearbeitungszeit für alle Aufträge

$\beta_6 = p_u \leq p_{k,m} \leq$	Maschinen- und auftragsabhängige, beschränkte Bearbeitungszeit
p_o	
$\beta_6 = p_{km}$	Beliebige maschinen- und auftragsabhängige Bearbeitungszeit
$\beta_7 = \emptyset$	Keine Reihenfolgerestriktionen
$\beta_7 = prec$	Reihenfolgerestriktionen zwischen Aufträgen
$\beta_8 = \emptyset$	Keine Rüstzeiten
$\beta_8 = rz_k$	Auftragsbezogene Rüstzeiten
$\beta_8 = rz_m$	Maschinenbezogene Rüstzeiten
$\beta_8 = rz_{m,k}$	Maschinen- und auftragsbezogene Rüstzeiten
$\beta_8 = rz_{k_1,k_2}$	Rüstzeiten nur abhängig von der Auftragsreihenfolge
$\beta_8 = rz_{k_1,k_2,m}$	Rüstzeiten abhängig von der Auftragsreihenfolge und Maschine
$\beta_9 = \emptyset$	Keine Maschinenausfälle
$\beta_9 = breakdown$	Maschinenausfälle sind möglich
$\beta_9 = az_m$	Maschinenabhängige Anlaufzeit nach der Instandsetzung
$\beta_{10} = \emptyset$	Keine zusätzlichen Ressourcen
$\beta_{10} = res, \lambda\sigma\rho$	λ verschiedene, endliche Ressourcentypen mit Anzahl σ und einer Maximalbeanspruchung von ρ
γ	Zielfunktion

Abschnitt 2.3.1	Ablauf der Monte Carlo Tree Search
$UCT(s, a)$	Upper confidence bound of trees für den Folgezustand erreicht durch Aktion a in Zustand s
$Q_{UCT}(s, a)$	Belohnung des Folgezustands erreicht durch Aktion a in Zustand s
$N(s, a)$	Anzahl der Besuche des Folgezustands erreicht durch Aktion a in Zustand s
$N(s)$	Anzahl der Besuche des Zustands s
τ	Aktueller Suchbaum
$A(s)$	Aktionen ausgehend vom Zustand s

Abschnitt 2.4	Lokale Suche
V	Menge der Vorgänge eines Produktionsmittels
A	Menge der Vorgänge eines Produktionsauftrags
E	Menge der nicht-zusammenhängenden Knoten
$G = (V, A, E)$	Gerichteter Graph aus den Knoten V, A, E
$d_{i,j}$	Kante (Dauer) zwischen den Knoten i, j
$PJ(v)$	Vorgänger des Vorgangs v
$SJ(v)$	Nachfolger des Vorgangs v
$PM(v)$	Vorgänger-Produktionsmittel des Vorgangs v
$SM(v)$	Nachfolger-Produktionsmittel des Vorgangs v

s^E	Frühster Startzeitpunkt des Vorgangs v
s^L	Spätester Startzeitpunkt des Vorgangs v
G^-	Reduzierter Graph im Rahmen von LSONE
G'	Resultierender Graph im Rahmen von LSONE
Abschnitt 3	Stand der Technik
$\beta_6 = p_{km}$	Beliebige maschinen- und auftragsabhängige Bearbeitungszeit
$\beta_9 = \text{breakdown}$	Maschinenausfälle sind möglich
$\beta_8 = rz_k$	Auftragsbezogene Rüstzeiten
X	Zufallsvariable X
$p_{t_0}(X, y)$	Gemeinsame Verteilung zum Zeitpunkt t_0 aus der Menge an Eingangsgrößen X und der Zielvariablen y
Q_{UR}	Selektionsfunktion als Linearkombination der UCT und RAVE-Funktion
$Q(s, a)$	Wert der Aktion a ausgehend vom Zustand s
$Q_{RAVE}(s, a)$	RAVE-Wert der Aktion a für den Teilbaum unter dem Zustand s
$\beta(s, a)$	Gewichtungsfaktor zwischen der UCT-Bewertung und der RAVE-Bewertung abhängig von der Anzahl der Besuche des Zustands s und der Aktion a
Abschnitt 4.1	Problemeingrenzung, Steuerungsaufgabe und Zieldefinition
$\beta_1 = \emptyset$	Beliebige Anzahl Produktionsaufträge
$\beta_4 = \emptyset$	Maximale Anzahl Prozessschritte unbegrenzt
$\beta_2 = A_k^*$	Unterschiedliche Freigabeterminale pro Auftrag k
$\beta_3 = F_k^*$	Individuelle Fertigstellungstermine pro Auftrag k
$\beta_7 = \emptyset$	Keine Reihenfolgerestriktionen
$\alpha_1 = FJ$	Mehrstufige Fertigung: Flexible Job Shop
$\beta_6 = p_{km}$	Beliebige maschinen- und auftragsabhängige Bearbeitungszeit
$\beta_8 = rz_{k_1, k_2}$	Rüstzeiten nur abhängig von der Auftragsreihenfolge
$\beta_9 = \text{breakdown}$	Maschinenausfälle sind möglich
$\beta_5 = \emptyset$	Produktionsauftrag nicht unterbrechbar
$\beta_{10} = \emptyset$	Keine zusätzlichen Ressourcen
$t_{ft,p}$	Spätmöglicher Fertigstellungszeitpunkt des Auftrags $p \in P$
t_{mks}	Makespan
t_w	Gesamtarbeitsmenge
t_{mrw}	Maximale Arbeitsbelastung
t_u	Auslastung
t_{ct}	Gesamtdurchlaufzeit aller Produktionsaufträge
t_d	Gesamtverspätung aller Produktionsaufträge

v_{t_s}	Tatsächlicher Beginn des Vorgangs v
v_{t_e}	Tatsächliches Ende des Vorgangs v
V	Menge aller eingeplanten Produktionsvorgänge
V_m	Menge aller eingeplanten Vorgänge auf Produktionsmittel m
\hat{V}	Menge aller Produktionsvorgänge
V_p	Menge der Produktionsvorgänge des Auftrags $p \in P$
M	Menge aller Produktionsmittel
P	Menge aller Produktionsaufträge
<hr/>	
Abschnitt 4.2.1	Concept Drift Erkennung
D_i	Unbekannte i -te Verteilung
W	Datenfenster der Länge w
W_i	Teildatenfenster der Länge w_1 mit $w_0 + w_1 = w$
w_i	Länge des Datenfesters W_i
$\hat{\mu}_0$	Mittelwert des Datenfensters w_0
$\hat{\mu}_1$	Mittelwert des Datenfensters w_1
ϵ_{cut}	Grenzwert zur Bewertung der Mittelwerte der Teilfenster
m_{ham}	Harmonisches Mittel
δ'	Anteiliges Konfidenzintervall entsprechend Datenfenstergröße
EG	Eingriffsgrenze der Qualitätsregelkarte
Z_t	Wert der EWMA-Qualitätsregelkarte der Periode t
L	Parameter der EWMA-Qualitätsregelkarte
λ	Gewichtungsfaktor des aktuellen Werts der EWMA-Qualitätsregelkarte
σ	Standardabweichung
<hr/>	
Abschnitt 4.2.2	Verteilungsschätzung
g	Anzahl Klassen
n	Stichprobenumfang
$H(x)$	Kummulierte Häufigkeitsverteilung
$h(a_i)$	Anzahl der Beobachtungen der Ausprägung a_i
$F(x)$	Empirische Verteilungsfunktion
L	Likelihood-Funktion des Maximum-Likelihood-Schätzers
$\hat{\Theta}_{ML}$	Beste Parameterschätzer der ML-Schätzung
Θ_{ML}	Parameterraum der ML-Schätzung
p_i	Diskrete Dichte der Klasse i
v	Anzahl der Freiheitsgrade des χ^2 -Anpassungstests
$1 - \alpha$	Signifikanzniveau des χ^2 -Anpassungstests
<hr/>	
Abschnitt 4.3.1	Problemmodellierung als Markov-Entscheidungs-problem

S	Menge aller Zustände S
s_t	Zustand zum Zeitpunkt t
A	Menge aller Aktionen
$A_s = A(s)$	Menge der Aktionen, die im Zustand s zur Verfügung stehen
a_t	Aktion zum Zeitpunkt t
$r(\cdot)$	Belohnungsfunktion
x	Einzelnes Zielkriterium
R^s	Menge der Belohnungen, die zu einem Zustand s gehören
Abschnitt 4.3.2	Selektionsphase
X	Menge der Zielkriterien
x_i	Zielkriterium i
γ_i	Gewichtung des Zielkriteriums i
r_s	Gewichtetes Mittel der Zielerreichung des Zustands s
$UCT_{HV}(s, a)$	Upper confidence bound of trees basierend auf dem Hypervolumen für den Folgezustand erreicht durch Aktion a in Zustand s
$Q_{HV}(s, a)$	Belohnung des Folgezustands basierend auf dem Hypervolumen erreicht durch Aktion a in Zustand s
τ	Suchbaum als Untermenge von $A \times S$
K	Anzahl geplanter Iterationen (Abbruchkriterium)
k	Aktuelle Iteration
c_s	Startwert des Explorationsfaktors C bei dynamischer Anpassung
c_e	Endwert des Explorationsfaktors C bei dynamischer Anpassung
$w_{i,s}$	Startwert der Zielgewichtung des Ziels x_i
$w_{i,e}$	Endwert der Zielgewichtung des Ziels x_i
$w_i(k)$	Zielgewichtung des Ziels x_i zur Iteration k
Abschnitt 4.3.3.2	Generierung zulässiger Aktionen
$a_{t_s}(m, v, p)$	Frühster Startzeitpunkt der Aktion a
$a_{t_e}(m, v, p)$	Frühster Endzeitpunkt der Aktion a
$t_e^{m, v_{prev}}$	Endzeitpunkt bis zu dem das Produktionsmittel m durch den Vorgang v_{prev} gelegt ist
$t_e^{p, v_{prev}}$	Endzeitpunkt an dem der Produktionsauftrag v den Vorgänger v_{prev} abgeschlossen hat
v	Bearbeitungsvorgang v
m	Produktionsmittel m
$v_{m, prev}$	Vorgängerprozess auf Maschine m
$t_t(l_{prev}, l_m)$	Transportdauer von dem Vorgängerstandort l_{prev} zum Nachfolgestandort l_m

$t_{su}(v_{prev}, v)$	Rüstdauer von dem Vorgänger v_{prev} zum aktuellen Vorgang v
$t_{m,v}$	Bearbeitungsdauer des Vorgangs v am Produktionsmittel m
t_i	Aktuell betrachteter Produktionszeitpunkt
Abschnitt 4.3.3.4	Methode zur intelligenten Einschränkung zulässiger Lösungen
n	Anzahl n besten Pareto-Punkte nach dem gleichgewichteten Zielfunktionswert
Θ	Menge der ausgewählten Filterkombinationen
Θ^P	Menge der ausgewählten Filterkombinationen nach dem Pareto-Kriterium
Θ^{pn}	Menge der ausgewählten Filterkombinationen nach dem Pareto-Nähe-Kriterium
Θ^ϵ	Menge der ausgewählten Filterkombinationen nach dem ϵ -Kriterium
$\bar{\Theta}$	Menge der angewendeten Filterkombinationen während der Selektionsphase
Θ_γ^{pm}	Menge der γ ausgewählten Filterkombinationen
Θ^*	Gesamtmenge der ausgewählten Filterkombinationen
ϵ	Anteil der zufällig gewählten Filterkombinationen an der Gesamtzahl der gewählten Filterkombinationen
ϵ_{prod}	Wahrscheinlichkeit in der Produktivphase eine zufällige Filterkombination auszuwählen
γ	Anzahl der zu wählenden Filterkombinationen
ξ	Relativer Abstand zur Pareto-Front
θ	Angewendete Filterkombination
θ_x	Durch die Filterkombination erreichter normalisierter Zielfunktionswert
$\Psi(\bar{\Theta}, i)$	Menge der Filterkombinationen, die zu den i besten Pareto-Punkten geführt haben
D_ξ	Menge der Hyperebenen mit Abstand ξ zur Pareto-Front
n_{min}	Mindestanzahl, die eine Filterkombination zu einer ausreichend guten Lösungen geführt haben muss
σ	Mindestwahrscheinlichkeit mit der die Filterkombination zu einem Ergebnis in Pareto-Nähe geführt haben muss
$\Lambda(\bar{\Theta}, \xi)$	Menge der Filterkombinationen, die zu Ergebnissen geführt haben, die zwischen den Hyperebenen und der Pareto-Front liegen.
$\Phi(\Theta^{pm}, i)$	Absteigend geordnete Menge der Filterkombinationen nach dem durchschnittlichen Hypervolumen
Abschnitt 4.3.5	Modifikationen der Backpropagation-Phase

$Q(s)$	Bewertung des Zustands s
R_s	Menge der normalisierten Zielfunktionswerte besuchter Zustände
s	Zustand s
Abschnitt 4.5	
Hybridisierung durch Kombination mit lokaler Suche	
r_{init}	Mindestbelohnung zur Ausführung einer lokalen Suche
$r_{min,i}$	Mindestbelohnung zur Ausführung einer lokalen Suche in Iteration i
r_{init}^{UB}	Obere Schranke der minimalen Belohnung
k	Faktor, um den der minimale geforderte Belohnung pro Iteration erhöht wird
$c^E(v)$	Frühster Endzeitpunkt des Transport-, Rüst- oder Bearbeitungsvorgangs v
$c^L(v)$	Spätester Endzeitpunkt des Transport-, Rüst- oder Bearbeitungsvorgangs v
$PJ(v)$	Vorgänger des Vorgangs v
$SJ(v)$	Nachfolger des Vorgangs v
$PM(v)$	Vorgänger-Produktionsmittel des Vorgangs v
$SM(v)$	Nachfolger-Produktionsmittel des Vorgangs v
$s^E(v)$	Frühster Startzeitpunkt des Transport-, Rüst- oder Bearbeitungsvorgangs v
$s^L(v)$	Spätester Startzeitpunkt des Transport-, Rüst- oder Bearbeitungsvorgangs v
t_v	Dauer des Transport-, Rüst- oder Bearbeitungsvorgangs v
$t_t(m_v, m_u)$	Dauer des Transports vom Produktionsmittel des Vorgangs v zum Produktionsmittel des Vorgangs u
$t_{m_u,v}$	Dauer des Bearbeitungsvorgangs v auf dem Produktionsmittel m_u
s^{E-}	Frühster Startzeitpunkt des Vorgangs v im reduzierten Graph G^-
s^{L-}	Spätester Startzeitpunkt des Vorgangs v im reduzierten Graph G^-
t_c	Konstante Toleranz für die Mindestverbesserung zwischen Iterationen der lokalen Suche
t_d	Abnehmende Toleranz für Verschlechterungen zwischen Iterationen der lokalen Suche
G^{ref}	Referenzlösung
G^i	Lösung i
$Q(G^i)$	Zielerreichung der Lösung i
Δ	Verschlechterungen zwischen Iterationen der lokalen Suche
T	“Temperatur“ des Simulated Annealing Verfahrens

1 Einleitung

Die industrielle Produktion spielt in Deutschland eine wichtige Rolle. 2019 betrug der Anteil des produzierenden Gewerbes ohne das Baugewerbe 24,2% an der Bruttowertschöpfung (Statistisches Bundesamt 2020). Im selben Jahr wurden knapp 8 Millionen Erwerbstätige in diesem Wirtschaftszweig gemeldet, dies entspricht 18,6% der Gesamterwerbstätigen¹. Der Vergleich dieser Anteile zeigt, dass das produzierende Gewerbe überdurchschnittlich zur Wertschöpfung beiträgt und gleichzeitig ein bedeutender Arbeitgeber ist. Auch wenn die Zahl der Erwerbstätigen im produzierenden Gewerbe verglichen zum Basisjahr 1991 gesunken ist, hat sich deren Zahl seit 2013 stabilisiert und ist sogar gegenüber 2013 leicht gewachsen¹.

1.1 Motivation und Problemstellung

Produktindividualisierung ist ein zentrales Motiv des produzierenden Gewerbes (Hu & Zhu u. a. 2008). Trotz der hohen Variantenzahl und der damit einhergehenden Komplexität werden Produktionssysteme und -netzwerke angestrebt, die ein ähnliches Kosten- und Qualitätsniveau erreichen wie eine Massenproduktion auf Basis einer gut ausgetakteten Linienfertigung. Das Streben nach hoher Effizienz trotz geringer Stückzahlen pro Variante führt zur Abkehr von der Einproduktlinie hin zu rekonfigurierbaren, modularen Systemen und Mehrproduktlinien (Koren & Gu u. a. 2018). Diese verteilen die hohen Investitionskosten auf eine größere Stückzahl und sind in der Lage, zu einem gewissen Grad Stückzahlschwankungen einzelner Varianten auszugleichen (Hu & Zhu u. a. 2008; ElMaraghy & Schuh u. a. 2013).

Im globalen Wettbewerb spielt die Effizienz der Produktionssysteme eine entscheidende Rolle. Eine hohe Auslastung, insbesondere bei einem kapitalintensiven Produktionssystem, ist ausschlaggebend, um Wertschöpfung am Wirtschaftsstandort Deutschland zu halten. Neben einer hohen Auslastung stellen kurze Lieferzeiten in vielen Kundensegmenten einen Wettbewerbsvorteil dar. Diese konkurrierenden Ziele bedeuten für die Produktion jedoch, dass eine auslastungsoptimale Reihenfolge nicht zu Lasten der Durchlaufzeiten gehen kann.

¹ Statistisches Bundesamt (2020), Anzahl der Erwerbstätigen im produzierenden Gewerbe(ohne Baugewerbe) in Deutschland von 1991 bis 2020 (in 1.000)., <https://de.statista.com/statistik/daten/studie/2190/umfrage/anzahl-der-erwerbstaetigen-im-produzierenden-gewerbe/>, aufgerufen am 20.07.2021

Eine weitere Herausforderung für die Planung und den Betrieb von Produktionssystemen ist die Volatilität der Kundennachfrage, die zunehmend unvorhersehbarer ist (Lanza & Nyhuis u. a. 2018). Diese Unsicherheit hat vielfältige Ursachen, zu denen neben konjunkturellen und saisonalen Einflüssen auch der technologische Wandel zählt (Grinninger 2012).

Durch die hohe Variantenzahl mit schwer prognostizierbaren Stückzahlen und den hohen Kostendruck gewinnen Produktionskonzepte an Bedeutung, die durch die Flexibilisierung der Produktionsstruktur einen wirtschaftlichen Betrieb selbst bei einem breiten Produktionsprogramm anstreben (Bochmann 2018; Greschke & Schönemann u. a. 2014). Auch wenn diese Konzepte unter verschiedenen Bezeichnungen in Forschung und Praxis bekannt sind, teilen sie einige grundlegende Eigenschaften. Zentral ist die Auflösung der Taktzeitbindung der einzelnen Arbeitsstationen durch eine Entkopplung des Materialflusses. Häufig wird der Materialfluss über flexible Transportmittel wie fahrerlose Transportsysteme (FTS), Hängebahnen oder durch Mitarbeiter gelöst. Übereinstimmend findet man auch die Verwendung von Mehrzweckstationen und redundanten Arbeitsplätzen. Der freie Materialfluss ermöglicht es, alternative Prozessreihenfolgen zur Herstellung der Erzeugnisse zu nutzen. Diese Konzepte firmieren unter den Begriffen **Matrixproduktion**² (Greschke 2016; Bochmann 2018), **modulare Montage**³ oder **cubeTEC**⁴. Das Konzept der linienlosen, mobilen Montagesysteme basiert ebenfalls auf den genannten Prinzipien (Hüttemann 2021). Die Flexibilisierung, die diese Systeme gemeinsam haben, kann genutzt werden, um dynamisch auf Abweichungen zu reagieren und durch eine intelligente Steuerung die Auslastung des Produktionssystems gleichmäßig zu verteilen.

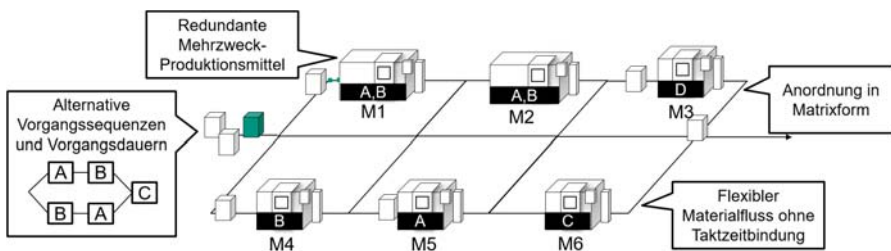


Abbildung 1.1: Matrixproduktion mit den Produktionsmitteln M1-M6 und den Prozessen A-C

²KUKA Systems GmbH (2016), "Matrix Produktion: ein Beispiel für Industrie 4.0", <https://www.kuka.com/de/branchen/loesungsdatenbank/2016/10/solution-systems-matrix-produktion>, aufgerufen am 09.11.2020

³Poll, D. (2016), "Das Ende des Fließbands", <https://www.produktion.de/trends-innovationen/das-ende-des-fließbands-306.html>, aufgerufen am 9.11.2020

⁴Krugsbeger & Krust (2017), "Der Rohbau der Zukunft ist flexibel", <https://media.daimler.com/marsMediaSite/de/instance/ko/Mercedes-Benz-Cars-Der-Rohbau-der-Zukunft-ist-flexibel.xhtml?oid=30023981>, aufgerufen am 09.11.2020

Der Produktionssteuerung kommt hierbei eine entscheidende Rolle zu. Sie beeinflusst maßgeblich die logistischen Kennzahlen (Bauernhansl & Ten Hompel u. a. 2014) und muss auch bei Störungen den Produktionsablauf sicherstellen können. Die Zielgrößen der Produktionssteuerung sind jedoch fallabhängig und teils konkurrierend. Im Gegensatz zur starr verketteten Linie kann die Produktionssteuerung den Weg jedes Produktionsauftrags innerhalb der durch den Montagevorranggraph gesetzten Grenzen anpassen. Um diese neu dazugewonnene Flexibilität trotz der damit einhergehenden Komplexität auch kurzfristig bei Störungen nutzen zu können, sind neue Ansätze erforderlich, die in kurzer Zeit, zulässige und möglichst gute und vorausschauende Steuerungsentscheidungen treffen. Vorausschauende Verfahren zeichnen sich dadurch aus, dass die Auswirkungen auf die nachfolgenden Schritte abgeschätzt werden (Domschke & Drexl u. a. 2015).

1.2 Zielsetzung

Ziel dieser Arbeit ist es, eine vorausschauende Produktionssteuerung unter Berücksichtigung alternativer Vorgänge sowie Rüst- und Transportzeiten für die Matrixproduktion zu erforschen, die in der Lage ist, nach unterschiedlichen Zielen zu steuern. Für eine Übertragbarkeit und Anwendbarkeit der Ergebnisse soll die angestrebte Produktionssteuerung zudem in der Lage sein, Steuerungsaufgaben für Produktionssysteme unterschiedlicher Struktur bezüglich der vorhandenen Produktionsmittel, -prozesse und des Produktionsprogramms zu übernehmen. Von großer Bedeutung ist die Fähigkeit, gezielt unterschiedliche Zielgrößen zu optimieren. Um auch kurzfristig auf Störungen reagieren zu können, soll erforscht werden, in wie weit sich eine hohe Reaktionsfähigkeit und eine hohe Lösungsgüte vereinbaren lassen. Zentrale Aufgaben der Produktionssteuerung sind die Allokation und Terminierung der zur Herstellung der Produktionsaufträge notwendigen Produktionsprozesse auf Produktionsmittel.

1.3 Aufbau der Arbeit

Kapitel 2 legt die Grundlagen für das Verständnis der Arbeit. Die Anforderungen an die Produktionssteuerung werden in Kapitel 3 hergeleitet und genutzt, um den Stand der Forschung zu bewerten und ein Forschungsdefizit abzuleiten. In Kapitel 4 wird die entwickelte Steuerung vorgestellt. Zur Beantwortung der Forschungsfragen wird in Kapitel 5 untersucht, wie das Verhalten der Steuerung beeinflusst werden kann, wie schnell die Lösungsgüte ansteigt und welches Lösungsniveau erreichbar ist. Zudem wird die Steuerung auf ein Praxisbeispiel und ein Benchmark-Problem angewendet. Die Arbeit schließt mit Diskussion und Ausblick.

2 Grundlagen

Im Folgenden wird zunächst das Konzept der Matrixproduktion eingeführt und eingeordnet. Zweiter Schwerpunkt des Kapitels bilden die Produktionsplanung und -steuerung. Als drittes wird das Verfahren Monte Carlo Tree Search vorgestellt. Der vierte Punkt behandelt die Grundlagen der lokalen Suche als Verbesserungsverfahren. Beide Verfahren bilden die Grundlage der Produktionssteuerung, die in Kapitel 4 entwickelt wird.

2.1 Industrielle Produktion und Produktionssysteme

Der Begriff der **industriellen Produktion** bezeichnet den zielgerichteten Transformationsprozess von materiellen und nicht materiellen Inputfaktoren zum Erzeugen von Produkten, die einen höheren Wert besitzen (Westkämper & Löffler 2016). Historisch wurde zwischen Teilefertigung und Montage unterschieden, wobei in der Teilefertigung urformende, umformende und trennende Fertigungsverfahren zum Einsatz kommen, während in der Montage die fügenden Fertigungsverfahren dominieren (Eversheim 1989). Wenn im Folgenden die historischen Begriffe Fertigungstypen und Fertigungsprinzipien verwendet werden, sind sie auf den Begriff der Produktion übertragbar (Eversheim 1989).

Ein **Produktionssystem** umfasst alle organisatorischen und technischen Elemente zur Planung, Entwicklung, Steuerung und zum Betrieb der Produktionsprozesse (Eversheim 1992). Der **Produktionsprozess** bezeichnet alle Bearbeitungs-, Montage- und Transportvorgänge, die zur Herstellung benötigt werden (Wiendahl & Reichardt u. a. 2014). Hierbei können unterschiedliche Produktionstechnologien zum Einsatz kommen.

Im Folgenden wird mit **Vorgangssequenz oder Vorgangsreihenfolge** die lineare Abfolge der Bearbeitungsvorgänge bezeichnet. Die Begriffe Vorgang, Produktionsvorgang und Prozessschritt werden synonym verwendet. Der Vorranggraph beinhaltet alle möglichen Vorgangssequenzen. Der Begriff **Produktionsmittel** steht für Betriebsmittel, Produktionsinseln (Bauernhansl & Ten Hompel u. a. 2014) oder Arbeitsstationen. Diese können einen beliebigen Automatisierungsgrad aufweisen und mehrere Bearbeitungsvorgänge realisieren. Ist der

Begriff der Maschine gebräuchlicher, beispielsweise bei der Klassifizierung von Produktionssteuerungsproblemen in Kapitel 2.2.2, wird auf den Begriff Maschine zurückgegriffen. Dieser wird synonym verwendet.

Die **Verkettungsarten** unterscheiden starre, lose und elastisch verkettete Systeme. Bei der **starrten Verkettung** befinden sich zwischen den Produktionsmitteln keine Puffer, sodass die Produktionsaufträge synchron weitergegeben werden müssen. Eine Störung eines Produktionsmittels betrifft somit das gesamte Produktionssystem. Die **elastische Verkettung** erlaubt die Weitergabe des Produktionsauftrags, sobald ein freier Pufferplatz verfügbar ist. In diesem System gibt es eine begrenzte Anzahl Pufferplätze zwischen den Produktionsmitteln, sodass eine lokale Entkopplung erzielt wird. Bei der **losen Verkettung** sind unbegrenzte Pufferkapazitäten zwischen den Produktionsmitteln vorhanden, sodass diese unabhängig voneinander sind (Wiendahl & Hegenscheidt 2006; Arnold & Isermann u. a. 2008).

2.1.1 Fertigungstypen und -prinzipien

Der **Wiederholcharakter** einer Produktion bezeichnet die Stückzahlcharakteristik vergleichbarer Erzeugnisse eines Produktionssystems. Wird ein Produkt unter überwiegender Verwendung der gleichen Produktionsmittel, Arbeitspläne und Zeichnungen wiederholt gefertigt, spricht man von **Wiederholfertigung**. Die **Mehrfachfertigung** zeichnet sich dadurch aus, dass das gleiche Produkt mehrfach, unmittelbar hintereinander gefertigt wird. Sofern die aufeinander folgenden Erzeugnisse Abwandlungen eines Grundtyps sind, spricht man von einer **Sorten- oder Variantenfertigung**. Werden die einzelnen Varianten des Grundtyps in großen Stückzahlen gefertigt, ist der Begriff der **Serienfertigung** gebräuchlich. Im Falle von großen Stückzahlen ohne Varianten bezeichnet man die Fertigung als **Massenfertigung**. Diese Differenzierungen sind als **Fertigungstypen** bekannt (Wiendahl 2019).

Die drei Faktoren, räumliche Anordnung der Betriebsmittel, die Art des Durchlaufs der Werkstücke sowie die Einbindung des Menschen, definieren das **Fertigungsprinzip** (Wiendahl 2019). Die Wahl des Fertigungsprinzips hängt mit dem Fertigungstyp zusammen und hat einen großen Einfluss auf das Niveau der logistischen und produktionstechnischen Kennzahlen wie beispielsweise Durchlaufzeit, Termintreue, Auslastung oder Bestand (Wiendahl 2019). Die Aufteilung der Fertigungsprinzipien nach ihrer räumlichen Struktur ist weit verbreitet. Im Folgenden sollen die für diese Arbeit relevanten Fertigungsprinzipien charakterisiert werden, das Verrichtungsprinzip, das Inselprinzip und das Fließprinzip.

Im **Verrichtungsprinzip** werden gleichartige Produktionsmittel räumlich zusammen angeordnet. Die Produktionsaufträge werden in Losen oder im Einstückfluss zwischen den Produkti-

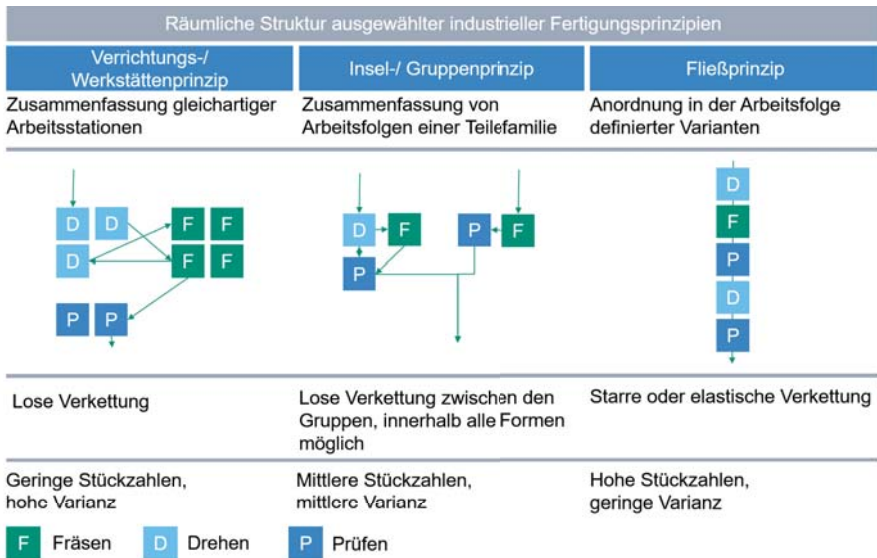


Abbildung 2.1: Räumliche Struktur ausgewählter Fertigungsprinzipien, angelehnt an (Wiendahl 2019)

onsmitteln gemäß ihrer individuellen Vorgangsreihenfolge bewegt. Dieses Prinzip zeichnet sich durch eine hohe Flexibilität bezüglich unterschiedlicher Bearbeitungszeiten und Reihenfolgen aus. Die Verkettung der einzelnen Produktionsmittel erfolgt lose. Für größere Stückzahlen findet dieses Prinzip auf Grund der langen Durchlaufzeiten und häufig hohen Bestände keine Anwendung (Wiendahl 2019), siehe Abbildung 2.1.

Das **Insel- oder Gruppenprinzip** verfolgt eine räumliche Gruppierung von Produktionsmitteln, die für die Erstellung eines Teilerzeugnisses benötigt werden. Innerhalb der Insel sind die Produktionsmittel meist flussorientiert angeordnet, damit die Durchlaufzeit gegenüber dem Verrichtungsprinzip sinkt. Es ist dennoch weiterhin möglich, einzelne Produktionsmittel auszulassen, sodass unterschiedliche Vorgangsreihenfolgen einfach realisierbar sind. Dieses Prinzip kann daher bei einem diversen Produktionsprogramm vorteilhaft sein. Zwischen den Gruppen gilt weiterhin das Verrichtungsprinzip (Wannenwetsch 2014). Innerhalb der Produktionsinseln kann die Verkettung ebenfalls elastisch oder starr realisiert werden. Das Prinzip wird oft mit einer eigenverantwortlichen, dezentralen Feinplanung innerhalb der Produktionsinseln verknüpft (Wiendahl & Reichardt u. a. 2014).

Beim **Fließprinzip** werden die Produktionsmittel in der Reihenfolge der Produktionsvorgänge

für eine definierte Produktvariante angeordnet. Die Bearbeitung erfolgt im Einstückfluss. Neben der starren Verkettung ist eine elastische Verkettung der Produktionsmittel möglich, um durch die Zwischenpuffer die Arbeitsstationen zu entkoppeln. Für den wirtschaftlichen Betrieb einer Fließfertigung ist eine hohe Auslastung nötig (Wannenwetsch 2014), Umbaumaßnahmen sind auf Grund der spezialisierten Anordnung oft teuer. Das Fließprinzip wird meist mit einer Taktung verknüpft. Bei einer ausgetakteten Produktionslinie werden somit Wartezeiten vermieden. Das Fließprinzip zeichnet sich vor allem bei der Produktion gleichbleibender, hoher Stückzahlen durch kurze Durchlaufzeiten und eine besonders hohe Wirtschaftlichkeit aus (Wiendahl & Reichardt u. a. 2014). Voraussetzung ist allerdings ein stabiler Produktionsprozess. Mehrproduktlinien sind Produktionslinien, die mehrere Erzeugnisse herstellen können. Unterscheiden sich die Bearbeitungsdauern der unterschiedlichen Erzeugnisse an den einzelnen Produktionsstationen, kommt es zu Taktzeitverlusten und damit Wartezeiten. Die positive Differenz aus Taktzeit und Bearbeitungszeit wird als Taktzeitverlust bezeichnet (Greschke & Schönemann u. a. 2014).

2.1.2 Matrixproduktion

Die Matrixproduktion ist eine Kombination aus Fließprinzip und Verrichtungsprinzip. Das Konzept der Matrixproduktion geht aus den Ansätzen der rekonfigurierbaren Produktionssysteme hervor, beispielsweise Koren & Shpitalni (2010) und den Überlegungen zu gekapselten und modularen, holonischen Produktionsinseln (Leitão & Restivo 2006). Auf Grund der hohen Flexibilität und dem gleichzeitigen Fokus auf Wirtschaftlichkeit eignet sich die Matrixproduktion für den Fertigungstyp der Sorten- oder Variantenfertigung (Schönemann & Herrmann u. a. 2015; Echsler Minguillon & Lanza 2017).

Abbildung 2.2 stellt die grundlegenden Eigenschaften der Matrixproduktion, die offenen Forschungsfragestellungen und die Einflussfaktoren auf die Steuerung dar.

Der Aufbau der Matrixproduktion ist gekennzeichnet durch **in Matrixform angeordnete Produktionsmittel** (Kern & Lämmermann u. a. 2017; Greschke 2016) im Raum. Wie beim Verrichtungsprinzip sind die einzelnen Produktionsstationen lose verkettet, allerdings werden sie gemäß des Materialflusses angeordnet (Schönemann & Herrmann u. a. 2015; Greschke & Schönemann u. a. 2014). Ein **flexibler Materialfluss ohne Taktzeitbindung** wird durch Transportmittel wie fahrerlose Transportsysteme (FTS), Hängebahnen oder den manuellen Transport realisiert. Durch die einhergehende Auflösung der Taktabhängigkeit der einzelnen Produktionsmittel können Wartezeiten aufgrund von blockierten oder leergelaufenen Produktionsmitteln als Konsequenz unterschiedlicher Bearbeitungszeiten vermieden werden

(Greschke & Schönemann u. a. 2014). Durch die fehlende Taktzeitbindung kann ein diversifiziertes Produktionsprogramm gemäß der Kundennachfrage produziert werden (Schönemann & Herrmann u. a. 2015; Kern & Lämmermann u. a. 2017; Keckl & Kern u. a. 2016). Das Konzept der Matrixproduktion schließt die Verwendung von **redundanten Mehrzweckmaschinen** ein. Nur wenn Produktionsmittel mehr als einen Produktionsprozess anbieten, kann eine hohe Auslastung erreicht werden, da sonst einzelnen Produktionsmitteln auf Grund des diversen Produktionsprogramms phasenweise zu wenige passende Aufträge gegenüberstehen könnten (Schönemann & Herrmann u. a. 2015). Redundante Produktionsmittel können ebenfalls eingesetzt werden, um die Anzahl alternativer Produktionsmittel zu erhöhen und somit weiter Wartezeiten auf Grund von belegten Produktionsmitteln zu verringern (Schönemann & Herrmann u. a. 2015). Durch den freien Materialfluss und die damit verbundene individuelle Route der Aufträge durch das Produktionssystem kann die Matrixproduktion **unterschiedliche Vorgangssequenzen** für die einzelnen Produktionsaufträge realisieren. Dies kann auch genutzt werden, um unterschiedliche Zielgrößen je Produktionsauftrag zu optimieren (Schönemann & Herrmann u. a. 2015; Greschke 2016).

Der modulare Aufbau der Matrixproduktion aus Produktionsinseln erhöht die Wandlungsfähigkeit, da einzelne Produktionsinseln durch die lose Verkettung einfacher hinzugefügt oder entfernt werden können. Diese Wandlungsfähigkeit kann gezielt genutzt werden, um das Produktionssystem an den aktuellen Produktionsmix anzupassen. Das Konzept der linienlosen, mobilen Arbeitssysteme nach Hüttemann nutzt mobile Produktionsmittel, um die Produktionsinseln bedarfsgerecht zu rekonfigurieren (Hüttemann 2021).

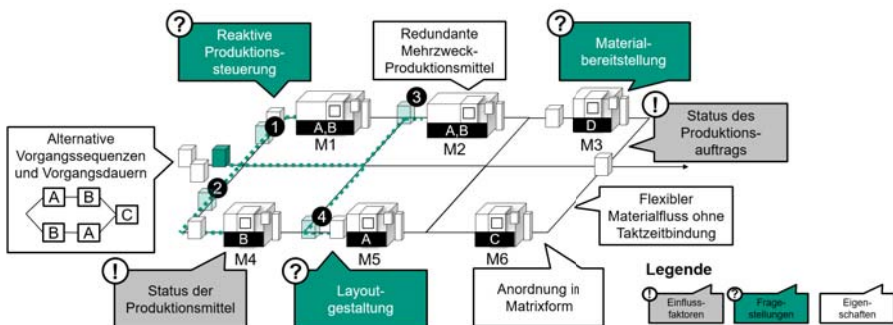


Abbildung 2.2: Eigenschaften, Forschungsfragen und Einflussfaktoren auf die Steuerung einer Matrixproduktion anhand eines Beispiels aus den Produktionsmitteln M1-M6, den Vorgängen A-D und einem Auftrag mit vier alternativen Routen 1-4

Im Kontext der Matrixproduktion gelten drei Forschungsfelder als besondere Herausforderung. Erstens muss die **Materialbereitstellung** sichergestellt werden. Durch die hohe Anzahl

an alternativen Vorgangssequenzen sinkt die Vorhersehbarkeit des Materialbedarfs in der Logistik. Soll steuerungseitig auf kurzfristige Ereignisse reagiert werden, nimmt zudem die Vorlaufzeit ab (Schönemann & Herrmann u. a. 2015; Greschke 2016). Die Fragen der Materialbereitstellung sind Gegenstand der Forschung von Kern & Lämmermann u. a. (2017). Behältersysteme, dynamische Supermärkte und Teilesätze sind mögliche Logistikkonzepte zur Umsetzung einer modularen Montage (Kern & Lämmermann u. a. 2017). Zweitens steigt die Komplexität der **Layoutplanung**. Durch die alternativen Vorgangssequenzen ist es schwierig, Transportintensitäten zu bestimmen, um Produktionsmittel, zwischen denen häufig Transporte stattfinden, räumlich zusammenzustellen. Gleichzeitig ist die empirische Bestimmung der Transportintensitäten verzerrt, da die Produktionssteuerung wegen der räumlichen Nähe Transporte zwischen benachbarten Stationen vorzieht. Wird das Layout geändert, verändern sich auch die Transportintensitäten zwischen den Produktionsstationen (Greschke 2016). Drittens gilt die **Produktionssteuerung** als besondere Herausforderung. An ihr ist es, die gestiegene Komplexität und die damit verbundenen Freiheitsgrade zu nutzen, sodass das Gesamtsystem Schwankungen dämpft und ein hohes Kennzahlenniveau erreicht wird. Aufgabe der Produktionssteuerung ist die Entscheidung, zu welchem Zeitpunkt welcher Vorgang an welcher Produktionsinsel bearbeitet werden soll (Schönemann & Herrmann u. a. 2015). Die Auftragsfreigabe, die sich aus dieser Entscheidung ergibt, bestimmt den Umlaufbestand der Produktion und beeinflusst somit Flächenbedarf und Wartezeiten. Wichtige Einflussfaktoren auf die Steuerungsentscheidung sind neben dem Aufbau des Produktionssystems der **Status der Produktionsmittel**, beispielsweise eine Störung, und der **Status der Produktionsaufträge**, beispielsweise aktuelle Durchlaufzeiten oder Verspätungen. Eine besondere Herausforderung für die Produktionssteuerung ist zudem die Verfügbarkeit genauer Daten zu den Dauern der Vorgänge sowie alternativer Vorgangssequenzen. Die Dauern der Vorgänge basieren oft auf Schätzung aus der Planungsphase (Schuh 2006). Eine operative Steuerung der Matrixproduktion benötigt jedoch exakte Informationen zu den Dauern, dem Status der Produktionsmittel und den Produktionsaufträgen (Greschke 2016).

2.2 Produktionsplanung und -steuerung

Die Produktionsplanung und -steuerung (PPS) schließt alle Geschäftsprozesse von der Auftragserstellung bis zum Versand ein (Luczak & Eversheim u. a. 1999). Aufgabe der operativen Produktionsplanung ist es, die Kundennachfrage einer bestimmten Periode zu decken (Wiendahl 1997). Die Produktionssteuerung befasst sich mit der Umsetzung der Produktionsplanung bezüglich der mengen- und zeitbezogenen Anforderungen unter Berücksichtigung der auftretenden Störungen (Schuh 2006).

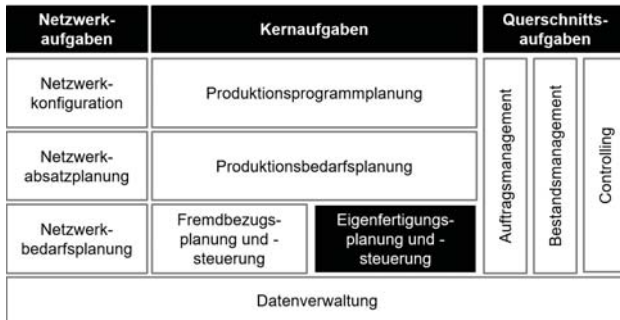


Abbildung 2.3: Aachner PPS-Modell nach Schuh (2006)

Nach dem Aachener PPS-Modell, siehe Abbildung 2.3, lassen sich die Aufgaben der Produktionsplanung in die drei Bereiche Netzwerkaufgaben, Kernaufgaben und Querschnittsaufgaben unterteilen. Innerhalb der **Kernaufgaben** befasst sich der Bereich der Eigenfertigungsplanung und -steuerung mit den Prozessen zur Herstellung der durch das Unternehmen selbst produzierten Erzeugnissen. Die Aufgaben der **Eigenfertigungsplanung und -steuerung** lassen sich weiter untergliedern in die Bereiche Belegungsplanung, Auftragsfreigabe, Auftragsüberwachung und Ressourcenüberwachung. (Schuh 2006)

Im Rahmen der Eigenfertigungsplanung und -steuerung müssen Planzeiten für die benötigten Vorgänge erhoben werden (Schuh 2006). In der Planungsphase liegen jedoch oft nur ungenaue, auf Schätzungen basierende Werte für die Planzeiten vor (Schuh 2006). Die Diskrepanz zwischen den Vorgabezeiten und den realen Zeiten führen zu nicht durchführbaren Plänen und häufigen Änderungen (Bauernhansl & Ten Hompel u. a. 2014). Durch Lerneffekte verkürzen sich zudem die Dauern der Vorgänge mit wachsender Erfahrung im Produktionsalltag, ohne dass jedoch die Planzeiten angepasst werden (Westkämper & Löffler 2016). Für die Produktionssteuerung ist das Vorliegen realistischer Planzeiten und deren ständige Aktualisierung jedoch eine wichtige Voraussetzung, da die resultierenden Pläne sonst kaum umsetzbar sind.

2.2.1 Aufgaben und Ziele der Produktionsplanung und -steuerung

Die **Aufgabe** der Eigenfertigungsplanung und -steuerung ist demnach die Entscheidung, welcher Produktionsvorgang welchen Produktes zu welchem Zeitpunkt auf welchem Produktionsmittel durchgeführt werden sollte unter Berücksichtigung der Auftragsdaten, wie

beispielsweise des Liefertermins, und der Ressourcendaten, wie die Verfügbarkeit der Produktionsmittel. Der Maschinenbelegungsplan, auf Englisch *Schedule*, enthält die zeitliche Allokation von Produktionsvorgängen auf Produktionsmittel (Schuh 2006).

Ziel des *Schedulings* ist es, eine Maschinenbelegung zu finden, die möglichst gute Zielkriterien erreicht und die Restriktionen der Produktionsaufträge und Produktionsmittel berücksichtigt (Pinedo 2016; Bochmann 2018). Die Zielgrößen des *Schedulings* lassen sich in kosten-, durchlaufzeit-, termin- und kapazitätsbezogenen Größen einteilen (Acker 2011). Typische Zielkriterien sind die Minimierung des Bestands, die Einhaltung von Terminen, eine gleichmäßige und hohe Auslastung und kurze Durchlaufzeiten (Wiendahl 1997) sowie eine hohe Flexibilität (Schuh 2006). Diese oben genannten Zielgrößen stehen in Konkurrenz zueinander. Eine hohe Auslastung lässt sich beispielsweise durch hohe Bestände erreichen, gleichzeitig führt dies jedoch meist zu einer schlechteren Termintreue und langen Durchlaufzeiten. Diese Zielkonflikte sind als **Dilemma der Produktionsplanung** bekannt (Domschke & Scholl u. a. 1997). Für die Wirtschaftlichkeit eines Produktionssystems ist vor allem die Auslastung entscheidend. In einem kundenzentrierten Marktumfeld, wo Wettbewerbsvorteile in Termintreue und kurzen Lieferzeiten liegen, steigt jedoch die Bedeutung zeitbezogener Zielgrößen. Auf Grund dieses Dilemmas ist die Gewichtung und Ausbalancierung dieser Zielgrößen eine fallbezogene und nicht verallgemeinerbar Aufgabe der PPS.

2.2.2 Klassifikation von Produktionssteuerungsproblemen

Produktionssteuerungsprobleme lassen sich nach Graham & Lawler u. a. (1979) anhand des Tripels $\alpha|\beta|\gamma$ der drei Merkmale α Maschinenumgebung, β Auftragscharakteristika und der Zielfunktion γ einteilen. Diese Einteilung hilft, die Anwendbarkeit von Steuerungsalgorithmen zu beschreiben und einzugrenzen.

2.2.2.1 Maschinenumgebung α

Die Maschinenumgebung α wird durch zwei Parameter α_1 und α_2 beschrieben. α_1 repräsentiert die Maschinenart und Anordnung. α_2 beschreibt die Anzahl der Maschinen oder der Fertigungsstufen. Es wird zwischen einstufiger und mehrstufiger Fertigung unterschieden. Da für die Matrixproduktion die mehrstufige Fertigung relevant ist, wird der Fokus auf diese Formen gelegt. Die Klassifikation für die einstufige Produktion findet sich im Anhang A1.

Maschinenart und Anordnung α_1

Für die **Maschinenart und die Anordnung** α_1 werden für die mehrstufige Fertigung folgende Ausprägungen unterschieden:

- *J*: Jeder Produktionsauftrag benötigt eine individuelle Vorgangssequenz. Diese Sequenz ist auftragsabhängig. Zwei aufeinanderfolgende Vorgänge müssen an unterschiedlichen Maschinen durchgeführt werden (Job Shop).
- *F*: Jeder Produktionsauftrag benötigt dieselbe Vorgangssequenz und besucht jede Maschine genau einmal (Flow Shop).
- *O*: Jeder Produktionsauftrag benötigt eine individuelle Vorgangssequenz und besucht jede Maschine genau einmal (Open Shop).
- *FJ*: Für mindestens einen Prozess stehen mehrere alternative Maschinen zur Verfügung (Flexible Job Shop).

Die Tabelle 2.1 fasst die Werte für die Maschinenart und -anordnung zusammen. Eine Matrixproduktion kann als Flexible Job Shop Problem (FJSP) eingeordnet werden.

Tabelle 2.1: Maschinenart und Anordnung der mehrstufigen Produktion α_1

	Mehrstufige Produktion			
	<i>J</i>	<i>F</i>	<i>O</i>	<i>FJ</i>
Maschinenzahl	> 1	> 1	> 1	> 1
Maschinenart	verschieden	verschieden	verschieden	verschieden
Bearbeitungszeit	maschinen- & auftragsabh.	maschinen- & auftragsabh.	maschinen- & auftragsabh.	maschinen- & auftragsabh.
Prozessschritte	>1	> 1	> 1	> 1
Prozesssequenz	verschieden	gleich	verschieden	verschieden
Durchläufe pro Maschine	beliebig	1	1	beliebig
Alternative Maschinen	nein	nein	nein	ja

Im Fall der mehrstufigen Fertigung gibt α_2 an, wie viele Vorgänge (Fertigungsstufen) benötigt werden. Folgende Werte sind für α_2 möglich (Acker 2011):

- \emptyset : Die Anzahl an Vorgängen ist beliebig und abhängig von der Planung. Dies ist dann der Fall, wenn durch eine bestimmte Sequenz ein Vorgang entfällt oder hinzukommt.
- *MF*: Die Anzahl benötigter Vorgänge ist fest.

2.2.2.2 Auftragscharakteristika β

Bei der Auftragscharakteristika werden zehn Untergruppen unterschieden. Jede Untergruppe kann verschiedene Ausprägungen annehmen. Im Folgenden werden die möglichen Ausprägungen der **Auftragscharakteristika** β zusammengefasst (Acker 2011).

Auftragsanzahl β_1

- \emptyset : Beliebige Anzahl Produktionsaufträge
- K : Konstante Anzahl Produktionsaufträge

Freigabetermin und Nachlaufzeit β_2

- \emptyset : Alle Produktionsaufträge stehen zu Anfang der Periode zur Verfügung.
- A_k^* : Es gibt unterschiedliche Freigabetermine.
- $nz_{k,m}$: Es müssen auftragsabhängige k und maschinenabhängige m Nachlaufzeiten, d.h. Wartezeiten nach Abschluss eines Vorgangs, berücksichtigt werden.

Fertigstellungstermin β_3

- \emptyset : Keine Fertigstellungstermine
- F_k^* : Es gibt einen Fertigstellungstermin pro Auftrag k .
- \tilde{F}_k^* : Überschreitung des Fertigstellungstermins F_k^* ist unzulässig.
- T : Es müssen alle Aufträge innerhalb des Planungshorizonts T fertiggestellt werden.

Maximale Arbeitsgangzahl β_4

- \emptyset : Es gibt keine Einschränkungen für die maximale Anzahl Vorgänge pro Produktionsauftrag.
- $O_k \leq O_{max}$: Die Anzahl der Vorgänge pro Auftrag k muss unter der Maximalzahl O_{max} liegen.

Unterbrechbarkeit von Prozessen β_5

- \emptyset : Die Bearbeitung eines Produktionsauftrag darf nicht unterbrochen werden, auch nicht ausfallbedingt.

- *pmtn* (*preemption*): Die Bearbeitung eines Produktionsauftrags darf unterbrochen werden.

Bearbeitungszeiten β_6

- p : Alle Produktionsaufträge haben dieselbe Bearbeitungszeit p .
- $p_u \leq p_{k,m} \leq p_o$: Die Bearbeitung des Produktionsauftrags k auf Maschine m ist zwischen den beiden Schranken p_u und p_o .
- $p_{k,m}$: Beliebige maschinen- und auftragsabhängige Bearbeitungszeiten sind möglich.

Reihenfolgebeziehungen β_7

- \emptyset : Es gibt keine Reihenfolgerestriktionen zwischen den einzelnen Produktionsaufträgen.
- *prec*: Es gibt Einschränkungen bezüglich der Auftragsreihenfolge.

Rüstzeiten β_8

- \emptyset : Es gibt keine Rüstzeiten.
- rz_k : Die Rüstzeit hängt nur vom Produktionsauftrag ab.
- rz_m : Die Rüstzeit hängt nur von der Maschine ab.
- $rz_{m,k}$: Die Rüstzeit hängt vom Produktionsauftrag und der Maschine ab.
- rz_{k_1,k_2} : Die Rüstzeit hängt nur von der Reihenfolge der Aufträge ab.
- $rz_{k_1,k_2,m}$: Die Rüstzeit hängt von der Reihenfolge der Aufträge und der betrachteten Maschine ab.

Unterbrechbarkeit von Maschinen β_9

- \emptyset : Es gibt keine Maschinenausfälle.
- *breakdown*: Maschinen können ausfallen.
- az_m : Maschinen können ausfallen und nach der Instandsetzung ist eine maschinenabhängige Anlaufzeit az_m nötig.

Ressourcen β_{10}

- \emptyset : Es gibt keine zusätzlichen Ressourcen.

- $res, \lambda \sigma \rho$: Es gibt λ verschiedene Ressourcentypen mit Anzahl σ , wobei ρ die maximale Ressourcenbeanspruchung abbildet.

2.2.2.3 Zielfunktion

Das dritte Element des Trippels $\alpha|\beta|\gamma$ ist die Zielfunktion. Es werden drei Kategorien unterschieden: kostenbezogene, zeit- bzw. terminbezogene und kapazitätsbezogene Kennzahlen. Je nach Kennzahl sind Aggregationsformen gebräuchlich beispielsweise die Summe, gewichtete Summe oder ein Maximal- bzw. Minimalwert (Acker 2011). Da die Zielfunktionen vom betrachteten Produktionssystem abhängen, wird an dieser Stelle darauf verzichtet beispielhafte Zielfunktionen aufzuführen.

2.2.3 Ansätze zur dynamischen Produktionsplanung und -steuerung

Die dynamische Produktionsplanung und -steuerung beziehen, anders als die statische PPS, Echtzeitereignisse ein (Ouelhadj & Petrovic 2008). Die Unterscheidung von kapazitäts- und auftragsbezogenen Ereignissen ist verbreitet (Cowling & Johansson 2002; Stoop & Wiers 1996). Je nach Autor unterscheiden sich die Unsicherheitsfaktoren innerhalb dieser Kategorien geringfügig. Übereinstimmend werden Maschinenausfälle, Nichtverfügbarkeit von Material und Werkzeug, Nichtverfügbarkeit von Transportmitteln und die Abwesenheit von Mitarbeitern als **kapazitätsbezogene Ereignisse** angesehen (Vieira & Herrmann u. a. 2003). Als **auftragsbezogene Ereignisse** werden das Eintreffen eines Eilauftrags, die Änderung eines spätesten Fertigstellungszeitpunkts, die Auftragsstornierung und die Änderung der Auftragspriorität zusammengefasst (Vieira & Herrmann u. a. 2003).

Im **dynamischen Scheduling** sind drei Strategien bekannt, um auf diese unvorhergesehenen Ereignisse zu reagieren. Der reaktive und der prädikativ-reaktive Ansatz greifen auf Basis von Echtzeitereignissen in die Steuerung ein. Der dritte Ansatz, die proaktive, robuste Herangehensweise, versucht durch geeignete Puffer die Auswirkungen erwartbarer Ereignisse zu begrenzen (Ouelhadj & Petrovic 2008). Dieser Ansatz wird im Weiteren nicht betrachtet.

Im **reaktiven Scheduling** wird kein Produktionsplan vor Produktionsstart erstellt. Das *Scheduling* erfolgt in der Regel echtzeitnah und lokal unter Nutzung von Prioritätsregeln (Ouelhadj & Petrovic 2008).

Im **prädikativ-reaktiven Scheduling** wird ein initialer Produktionsplan erstellt, der ausgelöst durch Echtzeitereignisse angepasst wird. Die Anpassung des initialen Plans aufgrund von exogenen Einflussfaktoren wird als **Rescheduling** bezeichnet (Vieira & Herrmann u. a. 2003).

Rescheduling-Umgebung				
Statische Auftragsmenge		Dynamische Auftragsmenge		
Deterministisch	Stochastisch	Zyklische Produktion	Auftragseingangsvariabilität	Prozessesequenzvariabilität

Rescheduling-Strategien				
Dynamisch		Prädikativ-reaktiv		
Prioritätsregeln	Steuerungstheoretisch	Periodisches Rescheduling	Ereignisbasiertes Rescheduling	Hybrides Rescheduling

Rescheduling-Methoden				
Schedule-Generierung		Schedule-Reparatur		
Robuster Schedule		Right-Shift-Methode	Teilweises Rescheduling	Vollständiges Rescheduling

Abbildung 2.4: Einordnung unterschiedlicher Ansätze der dynamischen Produktionsplanung und -steuerung in Anlehnung an Vieira & Herrmann u. a. (2003)

Rescheduling läuft in drei Phasen ab. In der ersten Phase wird die Auswirkung der Störung auf die Zielgrößen evaluiert. Sind diese klein genug, ist keine Plananpassung nötig. Der zweite Schritt befasst sich mit der Generierung möglicher Plananpassungen. Im dritten Schritt wird der initiale Plan angepasst und übernommen, sofern die Zielerreichung den Anforderungen genügt (Vieira & Herrmann u. a. 2003). Beim *Rescheduling* können unterschiedliche Ansätze nach der *Rescheduling*-Umgebung, der Strategie und der angewendeten Methode unterschieden werden (Vieira & Herrmann u. a. 2003), siehe Abbildung 2.4. Im Folgenden wird ein Überblick über die für die Matrixproduktion relevanten Kategorien gegeben.

Die **Rescheduling-Umgebung** unterscheidet statische und dynamische Auftragseingänge. Im Gegensatz zum statischen Auftragseingang sind im dynamischen Szenario nicht alle Produktionsaufträge bekannt. Für die Matrixproduktion ist in der Kategorie des dynamischen Auftragseingangs die Untergruppe der Prozessequenzvariabilität relevant, da die lose Verkettung der Matrixproduktion eine situationsabhängige Wahl der Prozessreihenfolge ermöglicht, siehe Abbildung 2.4.

Im Bereich der **Rescheduling-Strategien** werden dynamische Ansätze, zu denen Prioritätsregeln und steuerungstheoretische Ansätze zur Generierung von Prioritätsregeln zählen, sowie prädikativ-reaktive Ansätze unterschieden. Die prädikativ-reaktiven Ansätze können anhand des Zeitpunkts der Plananpassung untergliedert werden. Periodische Ansätze überprüfen und passen den Plan ausschließlich in festen Zeitabständen an. Periodische Ansätze sind auch unter dem Begriff der rollierenden Planung bekannt. Häufige Zeitintervalle sind zu Schichtbeginn. Die Bestimmung einer geeigneten *Rescheduling*-Periode ist jedoch eine offene Forschungsfrage. Die resultierenden relativ stabilen Produktionspläne liefern in dynamischen Umgebungen jedoch oftmals eine schlechte Zielerreichung (Vieira & Herrmann u. a. 2003). In ereignisbasierten Ansätzen lösen besondere Vorkommnisse eine Plananpassung

aus. Diese können sowohl kapazitäts- als auch auftragsbezogen sein. Oft werden Maschinenausfälle oder das Eintreffen eines neuen Produktionsauftrags als Auslöser genutzt. Hybride Ansätze kombinieren das periodische und ereignisbasierte Vorgehen. Den Basistakt des *Reschedulings* bildet eine periodische Überprüfung. Zudem erfolgt eine Neuplanung beim Auftreten bestimmter Ereignisse, beispielsweise bei Maschinenausfall oder beim Eintreffen oder bei Stornierung eines Produktionsauftrags (Church & Uzsoy 1992). Dynamisches *Scheduling* zeigt bei einfachen Systemen gute Ergebnisse. Bei sehr komplexen Systemen werden jedoch sehr viele Informationen benötigt, sodass die Reaktionszeit und der Rechenaufwand stark ansteigen (Vieira & Herrmann u. a. 2003). Da mit der Matrixproduktion auch das Ziel verfolgt wird, auf Abweichungen zu reagieren (Kern & Lämmermann u. a. 2017), sind vor allem das ereignisbasierte *Rescheduling* und das hybride *Rescheduling* relevant.

Der dritte Bereich beschäftigt sich mit den ***Rescheduling-Methoden***. In diesem Bereich wird zwischen Ansätzen unterschieden, die Abweichungen in der Planung bereits berücksichtigen und Ansätzen, die den ursprünglichen Plan aufgrund von Abweichungen ändern. Werden erwartbare Abweichungen bereits im ursprünglichen Plan berücksichtigt, spricht man von einem robusten Plan. Robustheit bezeichnet die Fähigkeit, trotz auftretender Störungen ein vergleichbares Leistungsniveau zu halten (Pinedo 2016). Treten Störungen auf, die zur Unzulässigkeit oder zu einer niedrigen Zielerreichung führen, kann der Plan repariert werden. Ein einfaches Verfahren ist die *Right-Shift-Methode*, bei der alle betroffenen Operationen soweit in die Zukunft verschoben werden, bis die Zulässigkeit wiederhergestellt ist. Dieser Ansatz führt jedoch häufig zu schlechten Lösungen, da das Potential der Reallokation von Produktionsschritten auf andere Produktionsmittel nicht berücksichtigt wird (Vieira & Herrmann u. a. 2003). Das teilweise *Rescheduling* verfolgt den Ansatz, nur die von der Verzögerung betroffenen Bereiche des Produktionsplans unter Nutzung der zur Verfügung stehenden planerischen Freiheitsgrade anzupassen. Ist keine ausreichend gute Lösung durch Umplanung des betrachteten Bereichs möglich, kann der Bereich erweitert werden, um durch die Vergrößerung des Lösungsraums weitere Lösungen zu erschließen. Beim vollständigen *Rescheduling* wird die Gesamtheit der offenen Produktionsaufträge neu eingeplant. Während dieser Ansatz den größtmöglichen Freiheitsgrad besitzt, führt die lange Berechnungszeit einer neuen Lösung zu einer niedrigen Flexibilität (Vieira & Herrmann u. a. 2003). Im Kontext einer echtzeitnahen Steuerung der Matrixproduktion sind insbesondere das teilweise und vollständige *Rescheduling* wichtig.

Ein **robuster *Schedule*** bezeichnet in der PPS einen Plan, der die Leistungsfähigkeit auch bei Störungen beibehält (Pinedo 2016). Neben der Erreichung der Leistungsfähigkeit spielt auch die realistische Terminierung der Vorgänge eine wichtige Rolle (Wiendahl 2019). Der Begriff der Robustheit wird auch in anderen Zusammenhängen verwendet, beispielsweise im

Projektmanagement. Im Bereich Robustheit werden vier Dimensionen unterschieden. Die Ergebnisrobustheit bezeichnet das Erreichen eines vergleichbaren Ergebnisniveaus. Die Zulässigkeitsrobustheit erfordert lediglich die Zulässigkeit der Lösung trotz Störeinflüsse. Die Optimalitätsrobustheit hingegen verlangt, dass die Lösung weiterhin die Optimallösung darstellt. Die Informationsrobustheit befasst sich mit der Robustheit gegenüber fehlenden Informationen. (Scholl 2000).

2.3 Monte Carlo Tree Search

Um bei Abweichungen steuernd in den Ablauf der Matrixproduktion eingreifen zu können, sind Verfahren nötig, die ausreichend schnell gute Lösungen erzielen können, um ein teilweises oder vollständiges *Rescheduling* zu ermöglichen. Aufgrund der Komplexität der Steuerungsaufgabe und der zahlreichen Abhängigkeiten sind Verfahren erforderlich, die solche Zusammenhänge berücksichtigen können. Monte Carlo Tree Search (MCTS) verfügt über vier Eigenschaften, die im Kontext der PPS wichtig sind.

MCTS ist ein iteratives Suchbaumverfahren, das in vier sich wiederholenden Phasen abläuft, bis der Suchbaum vollständig aufgebaut ist oder ein zeit- oder iterationsbezogenes Terminierungskriterium erreicht ist (Browne & Powley u. a. 2012). MCTS kombiniert die Repräsentation von Markov-Problemen als Baumstruktur und Monte Carlo Simulation. Zustände bilden die Knoten des Baums und die Aktionen werden als Kanten repräsentiert. Die Grundidee des Verfahrens besteht darin, iterativ eine weitere zulässige Aktion ausgehend vom jeweils vielversprechendsten Zustand durchzuführen und im Anschluss durch zufällige weitere Aktionen (Monte Carlo Simulation) einen möglichen Terminalzustand zu ermitteln. Die Güte der Aktion wird abschließend auf Basis des erreichten Terminalzustands ermittelt. MCTS führt hierzu iterativ vier Phasen aus, die in Abschnitt 2.3.1 eingeführt werden.

Für die Produktionssteuerung sind vier Eigenschaften von MCTS hervorzuheben:

Domänenunabhängigkeit: MCTS ist eine domänenunabhängige Heuristik, die zur Lösung unterschiedlicher Probleme ohne Hinzunahme von Domänenwissen angewendet werden kann. Domänenwissen bezeichnet in diesem Zusammenhang problemspezifisches Erfahrungswissen, beispielsweise in Form von problemspezifischen Entscheidungsregeln. Dies zeigt die Anwendung sowohl im Bereich der Spiele als auch beispielsweise der Produktionstechnik. Durch die Berücksichtigung von Domänenwissen kann jedoch die Suchgeschwindigkeit drastisch reduziert werden (Browne & Powley u. a. 2012). Auf der einen Seite hilft Domänenwissen spezifische Probleme zu lösen, auf der anderen Seite sinkt damit die Generalisierbarkeit (Drake & Uurtamo 2007).

Anytime-Algorithmus: MCTS gehört zur Klasse der *anytime*-Algorithmen, deren Eigenschaft es ist, zu jeder Zeit eine zulässige Lösung zurückzugeben. Mit steigender Ausführungszeit und damit höherer Iterationszahl verbessert sich in der Regel das Ergebnis. Dies stellt sicher, dass auch bei Abweichungen, die kurzfristig eine Anpassung erfordern, wie beispielsweise einem Ausfall, unmittelbar eine zulässige Lösung ermittelt werden kann.

Asymmetrisches Baumwachstum: Gesteuert durch den Selektionsmechanismus kann der Lösungsraum asymmetrisch erkundet werden, sodass der Suchfokus auf den vielversprechendsten Lösungen liegt (Browne & Powley u. a. 2012). Diese Eigenschaft ermöglicht, die Produktionssteuerungsalternativen priorisiert zu erschließen, die den individuellen Zielsetzungen entsprechen.

Bewertung auf Basis von Terminalzuständen: Die Bewertung einer Aktion basiert auf der Bewertung der erreichten Terminalzustände anhand der gewählten Zielfunktion. Für die Produktionssteuerung bedeutet dies, das vollständige Belegungspläne anhand der relevanten produktionstechnischen Kennzahlen evaluiert werden. Die schwierige Bewertung unvollständiger Pläne entfällt somit. Die Bewertung der Terminalzustände im domänenspezifischen Zielsystem, hier anhand von produktionstechnischen Kennzahlen, ermöglicht es zudem, mit diesem Ansatz verschiedene Ziele zu optimieren (Browne & Powley u. a. 2012).

2.3.1 Ablauf der Monte Carlo Tree Search

MCTS ist im Kontext von Spielen verbreitet (Silver & Huang u. a. 2016; Silver & Schrittwieser u. a. 2017). Das Verfahren geht auf die Arbeiten von Coulom (2006) zurück, der allerdings noch andere Begriffe für die einzelnen Phasen des Algorithmus verwendet.

MCTS läuft in vier iterativen Phasen ab (Chaslot & De Jong u. a. 2006), siehe Abbildung 2.5. Sobald der Suchbaum vollständig aufgebaut oder ein zeit- oder iterationsbezogenes Abbruchkriterium erreicht ist, wird der Zustand der obersten Baumebene mit der besten Bewertung oder die beste Entscheidungssequenz als Ergebnis zurückgegeben.

Selektion: In der ersten Phase des Verfahrens wird ein Blattknoten ausgewählt, der als Ausgangspunkt für die Aktionsgenerierung dient. Hierbei soll der vielversprechendste Ausgangszustand durch die Selektionsfunktion ermittelt werden. Die Selektionsfunktion beinhaltet unabhängig von ihrer konkreten Ausgestaltung zwei Bestandteile, zum einen die Güte des Zustands und zum anderen die Anzahl der Besuche des Zustands. Der erste Wert wird in der Selektionsfunktion genutzt, um Zustände mit gutem Zielwert weiter auszubauen. Der zweite Wert dient der Erkundung des unbekanntenen Lösungsraums (Browne & Powley u. a.

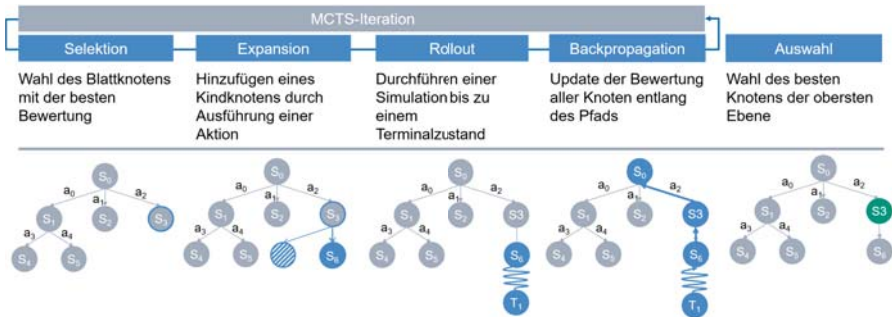


Abbildung 2.5: Ablauf des Monte Carlo Tree Search Verfahrens

2012). Diese Selektionsfunktion wird auch als *Tree Policy* bezeichnet. Die am weitest verbreitete Selektionsfunktion ist die *Upper Confidence Bound 1 applied to trees*-Funktion, kurz UCT (Browne & Powley u. a. 2012). Diese *Selection Policy* geht auf die Arbeit von Kocsis & Szepesvári (2006) zurück. Die Bewertung des Folgezustands, der sich ergibt, wenn man aus dem Zustand s die Aktion a ausführt, setzt sich aus der Bewertung dieses Folgezustands $Q_{UCT}(s, a)$ (*Exploitation*) und der mit der Explorationsfaktor c gewichteten Anzahl der Besuche dieses Folgezustands $N(s, a)$ im Verhältnis zu der Anzahl der Besuche des Elternknotens $N(s)$ zusammen, siehe Formel 2.3.1.

$$UCT(s, a) = Q_{UCT}(s, a) + c \sqrt{\frac{\ln(N(s))}{N(s, a)}} \quad 2.3.1$$

Wurde der Folgezustand noch nicht besucht, $N(s, a) = 0$ ergibt sich ein Wert von ∞ . Dies führt dazu, dass zum einen jeder Zustand mindestens einmal besucht wird und zum anderen zuerst alle Folgezustände einer Baumebene expandiert werden, bevor eine weitere Ebene hinzugefügt werden kann.

Ist der Folgezustand bereits Teil des Suchbaums τ , wird der Zustand mit der höchsten Bewertung selektiert, siehe Formel 2.3.2.

$$\operatorname{argmax}_a UCT(s, a), \quad \forall (s, a) \in \tau \quad 2.3.2$$

Ist der Folgezustand nicht Teil des Suchbaums τ , wird eine zulässige Aktion $A(s)$ zufällig

gewählt, sodass ein neuer Folgezustand erreicht wird, siehe Formel 2.3.3.

$$\underset{a}{\text{random}} A(s), \quad \forall (s, a) \notin \tau \quad 2.3.3$$

Kocsis & Szepesvári (2006) konnten zeigen, dass MCTS mit UCT als *Selection-Policy* für Zwei-Personen-Nullsummen-Spiele gegen den Minimax-Suchbaum konvergiert und damit gegen das Optimum des Spiels (Kocsis & Szepesvári 2006; Kocsis & Szepesvari u. a. 2006).

Expansion: In der zweiten Phase werden eine oder mehrere zulässige Aktionen an den selektierten Zustand im Suchbaum angefügt.

Rollout oder **Playout:** In der dritte Phase des Algorithmus werden weitere Aktionen ausgeführt bis ein Terminalzustand erreicht wird. Hier wird zwischen *heavy rollouts* und *light rollouts* unterschieden. Bei *light rollouts* werden die zulässigen Aktionen zufällig bestimmt. *Heavy rollouts* berücksichtigen zusätzliches Domänenwissen bei der Bestimmung der Aktionen. Die Funktion, die während der *light rollouts* zum Einsatz kommt, wird als *Default-Policy* bezeichnet. Hierbei werden zufällig Aktionen $a \in A(s)$ ausgewählt, bis ein Terminalzustand erreicht wird (Browne & Powley u. a. 2012).

Backpropagation: In der vierten Phase wird der durch das *Rollout* erreichte Terminalzustand anhand des Zielsystems bewertet. Neben dem Zielfunktionswert wird die Anzahl der Besuche pro Knoten auf dem Pfad zum Terminalzustand aktualisiert.

In allen vier Phasen des Algorithmus können zahlreiche Modifikationen vorgenommen werden. Einen breiten Überblick gibt Browne & Powley u. a. (2012).

Als **Abbruchkriterium** sind entweder ein fixes Zeitbudget oder eine feste Anzahl Iterationen verbreitet. Sobald das Abbruchkriterium erreicht ist, wird ein Kindknoten des Wurzelknotes gemäß einer Funktion zurückgegeben. Nach UCT wird der am häufigsten besuchte Knoten ausgewählt. Aufgrund der Eigenschaften der UCT-Funktion fällt das ebenfalls mit dem vielversprechendsten Knoten zusammen. Alternativen sind Kindknoten mit dem höchsten Zielfunktionswert, mit dem höchsten durchschnittlichen Zielfunktionswert oder einer Kombination aus Zielfunktionswert und Besuchsanzahl (Browne & Powley u. a. 2012). Als Ergebnis des Verfahrens kann ebenfalls die beste Entscheidungssequenz verwertet werden. Im Kontext der PPS entspricht dies einem vollständigen Belegungsplan.

2.3.2 Abgrenzung Monte Carlo Tree Search und Reinforcement Learning

MCTS weist einige Parallelen zum *Reinforcement Learning* (RL) auf. Für eine detaillierte Betrachtung sei auf Vodopivec & Samothrakis u. a. (2017) verwiesen. MCTS selbst gehört zu den Suchbaumverfahren. Diese Verfahren nutzen eine Baumstruktur, um den Lösungsraum effizient darzustellen.

RL und MCTS ermitteln beide effizient Aktionen, aktualisieren Bewertungen von Zuständen und balancieren Exploration und *Exploitation* aus (Vodopivec & Samothrakis u. a. 2017). MCTS-*Rollouts* entsprechen episodens-basiertem Lernen im RL. Episodens-basiertes Lernen liegt dann vor, wenn die Konsequenzen der Einzelhandlungen erst am Ende der Episode evaluiert werden, beispielsweise am Ende der Planungsperiode. Die Lernepisode im RL entspricht einem *Rollout* in MCTS, welches ebenfalls alle Aktionen und Zustände ausgehend vom Wurzelknoten bis zum Erreichen eines Terminalzustands beinhaltet. Eine weitere Gemeinsamkeit beider Verfahren ist, dass die Bewertung, die einzelnen Zuständen zugeordnet werden, aus der Erfahrung gewonnen werden. Beide Verfahren speichern das Wissen über die Güte von Zuständen. Im MCTS werden die Bewertungen im Rahmen der *Backpropagation*-Phase angepasst. Die *Backpropagation* entspricht sogenannten *offline*-Algorithmen, die erst am Ende der Episode die Bewertung der Aktionen anpassen (Vodopivec & Samothrakis u. a. 2017; Sutton & Barto 2015). MCTS speichert das gewonnene Wissen jedoch nur bis das Abbruchkriterium des MCTS-Laufs erreicht ist. Die Balance zwischen Exploration und *Exploitation* wird im MCTS mit Hilfe der UCT-Funktion nach Kocsis & Szepesvári (2006) kontrolliert. Im RL verfolgt die ϵ -*greedy*-*Policy* einen vergleichbaren Weg. Bei der ϵ -*greedy*-*Policy* werden zufällige Aktionen mit abnehmender Wahrscheinlichkeit ausgewählt.

Ein entscheidender Unterschied zwischen beiden Verfahren liegt darin, dass MCTS nur einen Teil des Suchraums in Form des Suchbaums speichert, während RL-Verfahren in der Regel den gesamten Suchraum durch eine Funktion approximieren. MCTS basiert nicht auf einer festen Codierung der Zustände und Aktionen, sondern ändert die Repräsentation des Lösungsraums *online*, also während der Optimierung, in Form des Baumwachstums. Ein weiterer wichtiger Unterschied zwischen den Verfahren stellen die *Rollouts* dar. Im Gegensatz zu RL kann in MCTS eine andere Strategie zur Bestimmung der Aktionen in der *Rollout*-Phase angewendet werden. Neben *light rollouts*, die eine hohe Geschwindigkeit ermöglichen, kann auch Domänenwissen eingebracht werden, um den Lösungsraum gerichteter zu erkunden (Vodopivec & Samothrakis u. a. 2017).

2.4 Lokale Suche als Verbesserungsverfahren

Das Produktionssteuerungsproblem ist NP-vollständig¹ (Garey & Johnson u. a. 1976). Aufgrund der Komplexität werden daher meist Heuristiken zur Lösung verwendet. Eine Herausforderung besteht in der Balance zwischen *Exploitation*, also der Suche in der Nähe guter Lösungen, und der *Exploration*, der Erkundung des noch unbekanntes Lösungsraums. Ein Ansatz, um die Vorteile beider Herangehensweisen zu vereinen, ist die Kombination eines Verfahrens mit guten Eigenschaften zur globalen Suche, beispielsweise genetische Algorithmen (Li & Gao 2016) mit lokaler Suche als Verbesserungsverfahren. Im Fall von MCTS kann die Exploration über die Selektionsfunktion gesteuert werden. Dennoch ist die Hybridisierung des Verfahrens mit lokaler Suche ein vielversprechender Ansatz (Browne & Powley u. a. 2012). In diesem Fall wird MCTS genutzt, um die Ausgangslösung für die lokale Suche zu erzeugen. Die lokale Suche kann dabei in den Ablauf des MCTS eingebettet werden, beispielsweise am Ende des *Rollouts* oder nachgelagert durchgeführt werden.

Grundsätzlich bezeichnet lokale Suche eine Familie von Optimierungsansätzen für kombinatorische Optimierungsprobleme, die ausgehend von einer Startlösung den benachbarten Lösungsraum durchsuchen (Domschke & Drexl u. a. 2015). Dieser wird durch eine Nachbarschaftsstruktur definiert. Lokale Suche wird oft in Verbindung mit einem anderen Optimierungsverfahren angewendet, welches die Startlösungen generiert. In diesem Fall spricht man von einem Verbesserungsverfahren. Sobald eine ausreichend gute Lösung in der Nachbarschaft der Startlösung gefunden wurde, wird diese als neue Ausgangslösung verwendet und die Suche startet erneut. Der Prozess endet, wenn keine Änderungen mehr möglich sind oder ein zeit-, güte- oder iterationsbezogenes Abbruchkriterium erreicht wurde.

Eine große Herausforderung in Verbindung mit lokaler Suche ist das Überbrücken von lokalen Optima. Vor allem zwei alternative Ansätze werden hierfür verfolgt: Erstens der ϵ -greedy-Ansatz, bei dem mit einer gewissen Wahrscheinlichkeit schlechtere Lösungen akzeptiert werden und zweitens die Tabu-Suche.

Tabu-Suche (Glover 1989) führt eine Liste mit bereits angewendeten Modifikationen beziehungsweise besuchten Lösungen ein. Diese Modifikationen dürfen nicht erneut durchgeführt werden. Üblicherweise werden die vorgenommenen Tauschoperationen bzw. die Rücktauschoperationen in der Tabu-Liste gespeichert. Die Elemente der Tabu-Liste werden nach

¹Kann für eine potentielle Lösung eines Problems in polynomieller Zeit geprüft werden, ob es sich um eine Lösung handelt, liegt das Problem in der Komplexitätsklasse NP. Diese Probleme werden als nichtdeterministische polynomielle Probleme bezeichnet. Darüber hinaus existieren Probleme, die so komplex sind, dass nicht in polynomieller Zeit geprüft werden kann, ob eine Lösung vorliegt. Diese werden als NP-vollständig bezeichnet. Für eine genaue Betrachtung sei auf Garey & Johnson u. a. (1976) verwiesen.

einer zu definierenden Regel nach dem FIFO-Prinzip entfernt. Tabu-Suche wendet die beste, nicht verbotene Operation an und verhindert somit eine Oszillation zwischen Lösungen (Glover 1989; Glover 1990). Tabu-Suche wird häufig auf *Job Shop* Probleme angewandt und modifiziert (Dell'Amico & Trubian 1993; Taillard 1994; Nowicki & Smutnicki 1996; Jain & Ranganaswamy u. a. 2000; Nowicki & Smutnicki 2005; Zhang & Li u. a. 2007).

Alternativ können Verfahren angewendet werden, die mit einer positiven Wahrscheinlichkeit einen nicht-optimalen Zug annehmen. Ein Vertreter ist *Simulated Annealing* (SA). Dieses Verfahren ahmt das Abkühlverhalten von Metallen nach. SA nimmt eine schlechtere Lösung als die Referenzlösung mit einer Wahrscheinlichkeit zwischen 0 und 1 an. Bessere Lösungen werden immer angenommen. Die Annahmewahrscheinlichkeit für schlechtere Lösungen reduziert sich analog zur Temperatur beim Abkühlprozess (Kirkpatrick & Gelatt u. a. 1983).

Ein alternativer Weg zur Vermeidung von lokalen Optima ist die Anpassung der Evaluationsfunktion, sobald ein lokales Optimum erreicht ist. *Guided Local Search* (GLS) fügt der Evaluationsfunktion Strafterme hinzu, sodass sich die Bewertung dieser Lösung des Minimierungsproblems verschlechtert (Balas & Vazacopoulos 1998).

Variable Neighborhood Search (VNS) nutzt unterschiedliche Nachbarschaftsstrukturen, um so lokale Optima zu vermeiden. Da die Lösungsmenge durch die Nachbarschaftsstruktur definiert wird, werden auch die lokalen Optima durch die Nachbarschaftsstruktur bestimmt. Durch die Verwendung verschiedener Nachbarschaftsstrukturen wird somit die Lösungsmenge erweitert und die Möglichkeiten gesteigert, lokale Optima zu verlassen (Mladenović & Hansen 1997; Hansen & Mladenović u. a. 2010).

Im Zusammenhang mit der PPS wird lokale Suche vorrangig zur Verbesserung des kritischen Pfads eingesetzt. Daher wird zuerst der kritische Pfad definiert, dann werden gerichtete Graphen als Modellierung des Steuerungsproblems für die lokale Suche eingeführt. Anschließend werden wichtige Nachbarschaftsstrukturen vorgestellt.

2.4.1 Kritischer Pfad

Ein kritischer Pfad bezeichnet eine Folge direkt aufeinanderfolgender Produktionsvorgänge, beginnend mit der Startzeit des ersten Vorgangs und endend mit der Fertigstellung des letzten Vorgangs. Der kritische Pfad ist damit der längste zusammenhängende Pfad des Belegungsplans. Seine Länge bestimmt den Makespan und jede Verzögerung auf dem kritischen Pfad wirkt sich auf den Gesamtplan aus. Der Makespan gibt dabei die Dauer vom Start des ersten Vorgangs bis zum Abschluss des letzten Vorgangs einer Menge an betrachteten Aufträgen an (Scholz-Reiter & Rekersbrink u. a. 2010b; Acker 2011; Blazewicz

& Domschke u. a. 1996). Ein Belegungsplan kann mehrere kritische Pfade aufweisen. Die Vorgänge auf dem kritischen Pfad werden als kritische Vorgänge bezeichnet. Die aufeinander folgenden kritischen Vorgänge auf einem Produktionsmittel stellen einen kritischen Block dar (Nowicki & Smutnicki 1996).

2.4.2 Gerichtete Graphen als Modellierung des Job Shop Scheduling Problems für die lokale Suche

Für die Modellierung von *Job Shop Scheduling* Problemen im Rahmen von Lokalen-Such-Ansätzen sind gerichtete Graphen verbreitet. Ein gerichteter Graph $G = (V, A, E)$ besteht aus einer Menge V an Knoten, die jeweils einen Vorgang darstellen sowie einem fiktiven Start- und Endknoten, einer Menge an zusammenhängenden Knoten A und einer Menge an nicht-zusammenhängenden Knoten E . Die Menge A bildet die Folge der Vorgänge eines Produktionsauftrags ab. Die Menge V stellt die Vorgänge dar, die auf einem Produktionsmittel durchgeführt werden. Die nicht-negative Länge d_{ij} der Kante $(i, j) \in A$ gibt die Dauer zwischen dem Start des Vorgangs i und dem Start des Vorgangs j an. Für einen Vorgang v gibt $PJ(v)$ die Vorgängeroperation an. $SJ(v)$ stellt den nachfolgenden Vorgang dar. Analog bezeichnet $PM(v)$ das Vorgänger-Produktionsmittel und $SM(v)$ das Nachfolger-Produktionsmittel des Vorgangs v (Balas 1969). Ein azyklischer Graph stellt eine zulässige Lösung des *Job Shop Scheduling Problems* dar. Abbildung 2.6 stellt einen beispielhaften gerichteten Graphen dar. Die unidirektionalen Kanten repräsentieren die Reihenfolgebeziehungen der Vorgänge der jeweiligen Aufträge. Die bidirektionalen Kanten verbinden die Vorgänge, die auf einem Produktionsmittel durchgeführt werden.

2.4.3 Nachbarschaftstrukturen für die Produktionsplanung und -steuerung

Lokale Suche bestimmt im ersten Schritt die benachbarten Lösungen einer Startlösung, die als Nachbarschaft bezeichnet werden, um diese vollständig oder teilweise zu durchsuchen. Alle Lösungen, die ausgehend von einer Startlösung erreichbar sind, werden durch die Nachbarschaftsfunktion bestimmt. Die Menge der erreichbaren Lösungen bilden die Nachbarschaft. Diese Funktion wird auch als Nachbarschaftsstruktur bezeichnet. (Acker 2011; Domschke & Drexel u. a. 2015) Sei S die Menge aller zulässigen Lösungen eines kombinatorischen Optimierungsproblems und $C : S \rightarrow \mathbb{R}$ die Zielfunktion, so lassen sich alle benachbarten Lösungen, die durch eine einzige Veränderung erreichbar sind, durch die Nachbarschaftsfunktion $N : S \rightarrow 2^S$ beschreiben. Diese Veränderungen werden als Zug bezeichnet (Domschke & Drexel u. a. 2015). Die unterschiedlichen lokalen Suchverfahren

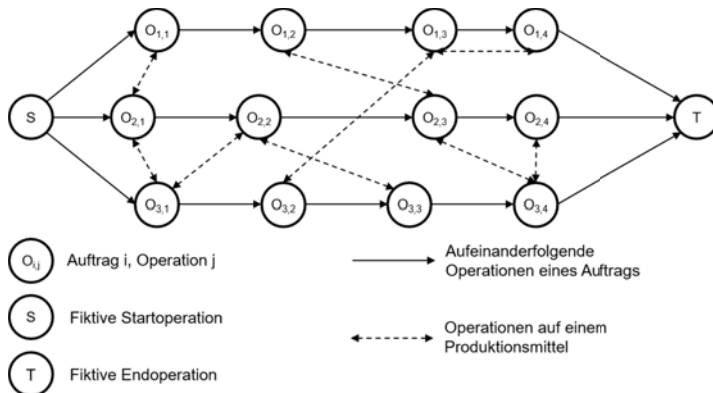


Abbildung 2.6: Gerichteter Graph für drei Produktionsaufträge und 3 Maschinen

nutzen verschiedene Nachbarschaftsfunktionen, sogenannte Nachbarschaftsstrukturen, um benachbarte Lösungen zu identifizieren. Je nach Optimierungsziel und Problemcharakteristik sind andere Nachbarschaftsfunktionen zielführend. Der Ansatz der *Variable Neighborhood Search* nutzt verschiedene Nachbarschaftsstrukturen, um den Lösungsraum abzudecken und die Gefahr der Stagnation in lokalen Optima zu verringern (Mladenović & Hansen 1997).

Im Folgenden werden grundlegende Nachbarschaftsfunktionen eingeführt, die oftmals als Ausgangspunkt für Anpassungen dienen. Die gleichen Nachbarschaftsfunktionen werden in Publikationen unterschiedlich bezeichnet, daher wird explizit auf den jeweiligen Autor verwiesen.

N1-, N2-Nachbarschaftsstruktur: Die N1-Nachbarschaftsstruktur betrachtet die Tauschoperationen aus direkt benachbarten Vorgängen auf demselben Produktionsmittel auf einem kritischen Pfad (Laarhoven & Aarts u. a. 1992). Die N2-Nachbarschaftsstruktur bezieht sich auf Tupel aus direkt nachfolgenden Vorgängen, wobei gilt, dass das vorangehende Produktionsmittel oder das nachfolgende Produktionsmittel nicht Teil des kritischen Pfads sein darf. Durch den Tausch der Vorgänge kann N2 zu einer Verbesserung des Makespan führen (Matsuo & Suh u. a. 1989). Beide Nachbarschaftsstrukturen nutzen einen gerichteten Graphen als Modellierung und garantieren, dass der Tausch zu einer zulässigen Lösung führt.

N4-Nachbarschaftsstruktur: Die Nachbarschaftsstruktur N4 reduziert die Größe der Nachbarschaft im Vergleich zu N1 und erhöht somit die Suchgeschwindigkeit. Betrachtet werden kritische Blöcke. Im ersten kritischen Block werden die letzten beiden Vorgänge und im letzten kritischen Block werden die ersten beiden Vorgänge getauscht. Für die anderen Blöcke

werden die ersten beiden Vorgänge mit den letzten beiden Vorgängen getauscht. Diese Nachbarschaftsstruktur führt dazu, dass alle kritischen Operationen einem Produktionsauftrag angehören oder auf einem Produktionsmittel liegen (Nowicki & Smutnicki 1996).

N5-Nachbarschaftsstruktur: Die Nachbarschaftsstruktur N5 wird bei *Guided Local Search* eingesetzt. Diese Nachbarschaftsstruktur ist nicht auf direkt benachbarte Vorgänge beschränkt. Daher werden Verschiebungen beliebiger kritischer Vorgänge eines Produktionsmittels betrachtet. Eine Vorwärtsverschiebung der Vorgänge v und w platziert v unmittelbar nach w . Eine Rückwärtsverschiebung ist analog definiert. (Balas & Vazacopoulos 1998)

N6-Nachbarschaftsstruktur: Die Nachbarschaftsstruktur N6 ähnelt N4, allerdings wird nur ein kritischer Pfad betrachtet. Um nur einen kritischen Pfad zu erhalten, wird die Einschränkung eingeführt, dass für einen kritischen Vorgang, der sowohl einen kritischen Vorgänger hat als auch auf einem kritischen Produktionsmittel liegt, nur der Vorgänger relevant ist. N6 führt im Vergleich zu N4 weitere einschränkende Regeln bezüglich der Tauschoperationen ein, sodass die Nachbarschaft kleiner ist (Geyik & Cedimoglu 2004).

LSONE-Nachbarschaftsstruktur: *Local Search by Moving One Operation* (LSONE) definiert einen Vorgang als kritisch, wenn der früheste Startzeitpunkt s^E und der späteste Startzeitpunkt s^L gleich sind. Ein kritischer Pfad besteht nur aus kritischen Vorgängen und durchläuft den gesamten Belegungsplan. LSONE beginnt mit der Reallokation des ersten kritischen Vorgangs des ersten kritischen Pfads. Hierfür wird der Belegungsplan als gerichteter Graph G dargestellt. Im ersten Schritt wird der kritische Vorgang aus dem Graphen gelöscht. Dies resultiert in einem reduzierten Graphen G^- . In diesem Graphen G^- wird nach einem geeigneten Intervall für den gelöschten, kritischen Vorgang gesucht. Ein geeignetes Intervall muss nach Ende des Vorgängers beginnen bzw. nach Beendigung der Vorgängeroperation auf dem Produktionsmittel, auf den der kritische Vorgang zugewiesen werden soll. Zudem muss die Bearbeitung vor dem Start des Nachfolgers beendet sein und auch keinen anderen Vorgang auf dem Produktionsmittel verzögern. Sofern ein solches Intervall existiert, wird der kritische Vorgang hinzugefügt, sodass der neue Plan G' entsteht. Dieses Verfahren endet, wenn keine Intervalle mehr existieren (Gao & Sun u. a. 2008). Eine Weiterentwicklung des LSONE schränkt die Wahl des Intervalls weiter ein (Wu & Wu u. a. 2013).

3 Stand der Forschung

3.1 Kriterien zur Bewertung des Stands der Forschung

Im folgenden Kapitel werden ausgesuchte, relevante Forschungsansätze vorgestellt. Diese werden anhand folgender Kriterien, abgeleitet aus der Zielsetzung, bewertet.

Abbilden einer realitätsnahen Matrixproduktion: Damit die Matrixproduktion als wirtschaftliche Fertigungsform realisiert werden kann, muss die Produktionssteuerung die Freiheitsgrade der Matrixproduktion und ihre Restriktionen berücksichtigen, siehe Abschnitt 2.1.2. Dies schließt sowohl die freie Wahl des nächsten Vorgangs, eingeschränkt durch den Vorranggraph, als auch die Wahl des Produktionsmittels ein ($\alpha_1 = FJ$). Auf Seiten der Restriktionen sollten reihenfolgeabhängige Rüstzeiten ($\beta_8 = rz_{k_1, k_2}$), auftragsbezogene Fertigstellungstermine ($\beta_3 = \tilde{F}_k^*$), maschinenbezogene Störungen ($\beta_9 = \text{breakdown}$) und durch das inselartige Layout anfallende Transportzeiten berücksichtigt werden.

Vorausschauende Mehrzieloptimierung: Die Produktionssteuerung kann sowohl termin-, durchlaufzeit- als auch kapazitätsbezogene Ziele verfolgen, siehe Abschnitt 2.2.2.3. Die Produktionssteuerung sollte in der Lage sein, die unterschiedlichen, teils konkurrierenden Zielgrößen gezielt zu optimieren. Daher sollte der **Suchfokus** des Verfahrens, das heißt der durchsuchte Bereich des Lösungsraums, steuerbar sein. Ebenfalls sollten die Auswirkungen der Steuerungsentscheidungen über mehrere Entscheidungsstufen betrachtet werden.

Kurzfristige Reaktionsfähigkeit bei Abweichungen: Die Produktionsumwelt ist durch Unsicherheit geprägt. Die Steuerung sollte fähig sein, auf kurzfristig auftretende Störungen ($\beta_9 = \text{breakdown}$) zu reagieren. Ein variables Zeitbudget für das Optimierungsverfahren ist daher wichtig. Um auch bei Störungen schnelle, gute Lösungen bieten zu können, ist ein **schneller Anstieg der Lösungsgüte** erforderlich.

Umgang mit ungenauen Vorgabezeiten: Die Planzeiten der Vorgänge sind eine wichtige Eingangsgröße für die Produktionssteuerung. Diese Planzeiten entsprechen jedoch oft Soll-Vorgaben und spiegeln nicht die realen Dauern wider, siehe Abschnitt 2.2. Zudem unterliegen die stochastischen Prozesse in der Produktion Änderungen, beispielsweise die Verteilung

der Rüst-, Transport-, und Bearbeitungszeiten. Die Produktionssteuerung sollte daher die zugrundeliegenden Verteilungen erheben und Verteilungsänderungen erkennen können.

Erreichbarkeit eines hohen Lösungsniveaus: Auch wenn das Problem der Produktionssteuerung nicht in kurzer Zeit exakt lösbar ist, ist dennoch eine möglichst hohe Lösungsgüte anzustreben. Die Lösungen des Verfahrens sollten sich mit steigender Rechenzeit dem globalen Optimum annähern. Die **erreichbare Lösungsgüte** ist daher ein weiteres Kriterium zur Evaluation der Steuerung.

Generalisierbarkeit des Lösungsverfahrens: Das Verfahren sollte in der Lage sein, unterschiedliche Steuerungsprobleme hinsichtlich der Produktionsmittel, Produktionsprozesse und des Produktionsprogramms zu übernehmen und dabei ein hohes Lösungsniveau zu erreichen. Eine manuelle Abstimmung der Parameter des Verfahrens auf das vorliegende Steuerungsproblem sollte daher gemieden werden.

Im weiteren Verlauf werden mögliche Herangehensweisen zur Lösung des Steuerungsproblems vorgestellt. Daran schließt sich die Darstellung relevanter Forschungsarbeiten auf Basis von Monte Carlo Tree Search an. Abschließend werden Vorgehen zum Umgang mit Unsicherheit in der Produktion betrachtet.

3.2 Ansätze zur Lösung des Job Shop Scheduling Problems

Aufgrund der Komplexität des Steuerungsproblems als NP-vollständiges Problem (Garey & Johnson u. a. 1976) dominieren heuristische Verfahren. Diese können das Erreichen des globalen Optimums nicht garantieren, sie verringern jedoch gegenüber exakten Verfahren die benötigte Rechenzeit stark. Exakte Verfahren finden auch beim FJSP Anwendung. Ihre Rolle liegt jedoch vor allem in der Generierung von Referenzlösungen, beispielsweise zur Bewertung heuristischer Verfahren. Es folgt ein Überblick verschiedener *Scheduling*-Ansätze.

3.2.1 Exakte Verfahren

Scheduling-Probleme sind kombinatorische Optimierungsprobleme, somit ist der Lösungsraum beschränkt (Müller-Merbach 1981). Durch Enumeration kann folglich das globale Optimum in endlich vielen Schritten gefunden werden. Die Enumerationsverfahren können in die Kategorien der vollständigen und der begrenzten Enumerationsverfahren eingeteilt werden (Acker 2011; Domschke & Drexl u. a. 2015). Die begrenzten Enumerationsverfahren schließen stückweise Bereiche des Lösungsraums aus, innerhalb derer das Optimum nicht

liegen kann. Ein bekannter Stellvertreter dieser Enumerationsverfahren ist das *Branch-and-Bound*-Verfahren (Domschke & Drexl u. a. 2015). Verfahren der begrenzten Enumeration kombinieren das exakte Verfahren mit einem heuristischen Eröffnungsverfahren, das in einem vorgelagerten Optimierungsschritt den Bereich des Lösungsraums identifiziert, innerhalb dessen die optimale Lösung liegen muss (Zimmermann & Stache 2010).

Bei ganzzahligen linearen Optimierungsproblemen können durch Relaxation, beispielsweise durch das Schnittebenenverfahren (Marchand & Martin u. a. 2002) oder die Lagrange-Relaxation (Lemaréchal 2001), die Ganzzahligkeitsbedingungen durch Ungleichungen ersetzt werden. Als Kombination des Schnittebenenverfahrens und des *Branch-and-Bound*-Algorithmus ist das *Branch-and-Cut*-Verfahren für das *Scheduling* von Bedeutung.

Forschung im Bereich der exakten Verfahren zur Lösung von *Scheduling*-Problemen beschäftigt sich mit effizienten Modellierungen. Dabei werden kleine Probleme zum Vergleich genutzt, wie beispielsweise das gemischt ganzzahlige Optimierungsproblem eines FJSP von Özgüven & Özbakır u. a. (2010) zur Minimierung der Gesamtkosten eines Job Shop Problems mit alternativen Produktionsmitteln ohne Berücksichtigung von Rüst- und Transportzeiten. Ein realitätsnäheres Modell stammt von Maravelias & Grossmann (2003). Sie modellieren eine variantenreiche Losgrößenproduktion als zeitkontinuierliches, gemischt-ganzzahliges Optimierungsproblem unter Berücksichtigung von Losgrößenaufteilung und -zusammenführung und der Beachtung von frühesten Start- und spätesten Fertigstellungszeitpunkten. Das Modell ist auf die unikriterielle Optimierung beschränkt (Maravelias & Grossmann 2003).

Aufgrund der Schwierigkeit, effiziente mathematische Problemmodellierungen zu finden, sind exakte Verfahren nur schwer erweiterbar und somit kaum an ein sich wandelndes Produktionssystem anpassbar. Durch die inhärente Komplexität des *Scheduling*-Problems spielen exakte Verfahren eine untergeordnete Rolle im reaktiven *Scheduling* (Acker 2011).

3.2.2 Heuristiken

Heuristiken sind Verfahren, die auf nicht-willkürliche Art potentielle Lösungen vom Suchprozess ausschließen und aufgrund des fehlenden Konvergenzbeweises keine Optimalität der Lösung garantieren (Streim 1975). Heuristiken können auf unterschiedliche Arten systematisiert werden, beispielsweise in Eröffnungs- und Iterationsverfahren (Müller-Merbach 1981) oder in problemspezifische Heuristiken und Meta-Heuristiken. Problemspezifische Heuristiken nutzen domänenbezogenes Wissen, um den Lösungsraum einzuschränken und somit kurze Berechnungszeiten und hohe Lösungsgüte zu realisieren. Meta-Heuristiken sind domänenunabhängige Lösungsansätze, oft angelehnt an Vorgänge der Natur, beispielsweise

Abkühlverhalten (*Simulated Annealing*), Schwarmverhalten (Pheromon-basierte Suche) oder genetische Evolution (genetische Algorithmen). Im folgenden Abschnitt werden anhand der letztgenannten Klassifikation stellvertretende Forschungsarbeiten für die jeweiligen Klassen vorgestellt und bewertet.

3.2.2.1 Problemspezifische Heuristiken

Eröffnungsverfahren sind problemspezifische Heuristiken, bei denen in einem Durchlauf eine zulässige Lösung generiert wird. Eröffnungsverfahren können sowohl vorrausschauend als auch *greedy* sein. Ansätze werden als *greedy* oder myopisch bezeichnet, wenn sie im jeweiligen Optimierungsschritt die Alternative mit der größtmöglichen Verbesserung wählen (Domschke & Drexl u. a. 2015). Im Gegensatz zum *greedy*-Ansatz berücksichtigt die vorrausschauende Heuristik nicht nur den Zielfunktionswert im aktuellen Schritt, sondern über eine Entscheidungssequenz hinweg (Zäpfel & Braune 2005). Als Teil der Eröffnungsverfahren sind die Prioritätsregeln die in der *Scheduling*-Praxis verbreitetsten Algorithmen (Acker 2011). Prioritätsregeln können zur Erzeugung eines Belegungsplans sowohl einzeln als auch kombiniert zum Einsatz kommen.

Scholz-Reiter & Beer u. a. (2004) untersuchten im Rahmen des Sonderforschungsbereichs 637 die Auswirkungen von autonomen Steuerungsansätzen auf logistische Kennzahlen anhand einer Matrixproduktion, allerdings ohne die Vorgangserienfolge der Produkte zu variieren. Untersucht wurden verschiedene Prioritätsregeln teils unter Nutzung lokaler Informationen. Es konnte gezeigt werden, dass dezentrale Prioritätsregeln bei starken Schwankungen der Bearbeitungszeiten und gleichzeitig hoher Redundanz der Produktionsmittel die logistischen Leistungskennzahlen verbessern (Scholz-Reiter & Beer u. a. 2004). Weitere Untersuchungen der Prioritätsregeln *Queue Length Estimator* (QLE) und *Due Date Method* (DUE) legen den Zusammenhang des Komplexitätslevels der Produktion, dem Grad der Selbststeuerung und der Erreichung logistischer Zielgrößen offen. Selbststeuerung mit Hilfe von Prioritätsregeln ist am effektivsten in einer Umgebung mittlerer Komplexität. Ist die Komplexität zu groß, kommt es aufgrund des fehlenden übergeordneten Steuerungsorgans zu suboptimalen und blockierenden Zuständen. Komplexität wird in diesem Fall über die Ausfallverteilung der Maschinen modelliert (Scholz-Reiter & Görge u. a. 2009a). Der Vergleich zwischen zentralen Heuristiken mit globaler Information und dezentralen Prioritätsregeln mit lokaler Information im Kontext eines flexiblen Flow Shops ähnlich einer Matrixproduktion zeigt, dass die dezentralen Prioritätsregeln bei hoher Dynamik den zentralen Heuristiken überlegen sind, jedoch bei geringer Dynamik aufgrund der unvollständigen Information zu schlechteren

Ergebnissen führen (Scholz-Reiter & Rekersbrink u. a. 2010a). Die Matrixproduktion ist aufgrund der Freiheitsgrade als komplex zu bewerten. Zudem soll die Produktionssteuerung auch kurzfristig bei Abweichungen gute Entscheidungen treffen. Die Forschungsergebnisse legen daher nahe, dass für diesen Zweck eine rein dezentrale Steuerung ungeeignet ist.

Als weitere Quelle von Unsicherheit wurden die Effekte von Nachfrageschwankungen im Zusammenhang mit verschiedenen Prioritätsregeln betrachtet. Verglichen wurden drei Produktlinien und eine über Prioritätsregeln und lokale Entscheidungen gesteuerte, vorwärtsgerichtete Matrixproduktion ohne Nutzung der Variabilität der Vorgangsreihenfolge. Es zeigt sich, dass die Matrixproduktion mit Hilfe der dezentralen Entscheidungsregeln besser auf die Nachfrageschwankungen reagiert als drei separate Linien (Scholz-Reiter & Freitag u. a. 2005). Die Auswirkungen von Schwankungen in der Kundennachfrage und Veränderungen von Transportdauern auf ein Produktionsnetzwerk verhalten sich ähnlich wie innerhalb eines Produktionssystems. Auch hier konnte gezeigt werden, dass eine reaktive, dezentrale Steuerung ein robusteres Leistungsniveau erreicht als eine zentrale Vorabplanung ohne Reaktionsfähigkeit. Die Untersuchungen auf Produktionssystem- und Produktionsnetzwerkebene zeigen, dass eine hohe Reaktionsfähigkeit notwendig ist, um trotz einer hohen Dynamik ein stabiles, hohes Kennzahlenniveau beizubehalten (Scholz-Reiter & Görge u. a. 2009b).

Die Untersuchungen zu Prioritätsregeln zeigen übereinstimmend, dass keine Prioritätsregel streng dominant ist (Blackstone & Phillips u. a. 1982). Darauf aufbauend existieren Ansätze, die sich mit der situationsabhängigen Auswahl von Prioritätsregeln beschäftigen, beispielsweise Mouelhi-Chibani und Pierreval oder Aydin und Öztemel (Mouelhi-Chibani & Pierreval 2010; Aydin & Öztemel 2000). Mouelhi-Chibani & Pierreval (2010) trainieren ein neuronales Netz, das abhängig vom Status des Produktionssystems und den zu diesem Zeitpunkt bekannten Produktionsaufträgen die beste Prioritätsregel identifiziert. Im Produktionskontext ermittelt das neuronale Netz für die bereitstehenden Produktionsaufträge, abhängig vom Systemzustand, nach welcher Prioritätsregel der nächste zu bearbeitende Auftrag bestimmt wird. Die Autoren nutzen eine Simulation, um das neuronale Netz zu trainieren und ermitteln die Güte der Entscheidung anhand einer unikriteriellen Zielfunktion. Das Vorgehen wird anhand eines einfachen Flow Shops erprobt und mit einem *Simulated Annealing*-Ansatz verglichen.

Ein weiterer Ansatz von Freitag & Hildebrandt (2016) besteht in der Generierung neuer Prioritätsregeln als Kombination elementarer Prioritätsregeln mit Hilfe genetischer Algorithmen im Kontext der Pareto-Optimierung (Freitag & Hildebrandt 2016). In diesem Fall wird vorab die beste Regel bestimmt, die im anschließenden Produktionsbetrieb Anwendung findet. Es existiert jedoch kein Mechanismus, der die Auswahl der Prioritätsregeln *online* anpasst. Ein situationsbedingter Wechsel der Regeln findet nicht statt.

3.2.2.2 Meta-Heuristiken

Im Bereich des *Schedulings* kommen vielfältige Meta-Heuristiken und Kombinationen aus Meta-Heuristiken zum Einsatz (Zäpfel & Braune 2005). Meist werden Verfahren zur Lösungssuche im umliegenden Lösungsraum (lokale Suche), beispielsweise Tabusuche oder *Simulated Annealing*, mit Ansätzen zur breiteren Suche (globale Suche), z.B. genetische Algorithmen oder Schwarmintelligenz-Verfahren, kombiniert (Li & Gao 2016). Insbesondere genetische Algorithmen und ihre Abwandlungen weisen ein gutes globales Suchverhalten auf, jedoch keine guten Leistungen bei der Umgebungssuche.

Xia & Wu (2005) teilen das Problem in die beiden Teilprobleme der Vorgangszuordnung auf Produktionsmittel und das *Schedulings* der Vorgänge auf den Produktionsmitteln auf. Das erste Problem wird mit Hilfe eines Partikel-Schwarm-Ansatzes gelöst, während auf das nachgelagerte Teilproblem *Simulated Annealing* als lokales Suchverfahren angewandt wird. Zur Evaluation des Steuerungsansatzes wird das Benchmark-Problem von Kacem (Kacem & Hammadi u. a. 2002) herangezogen. Hierbei handelt es sich um eine FJSP ohne Rüst- und Transportzeiten und ohne Flexibilität im Vorranggraph. Der hybride Ansatz zeigt gute Ergebnisse für komplexe Probleme mit zehn Maschinen und 15 Aufträgen. Die Modellierung des *Scheduling*-Problems, sodass die Partikel-Schwarm-Algorithmen effizient Lösungen finden, stellt eine große Herausforderung dar. (Xia & Wu 2005).

Kacem & Hammadi u. a. (2002) stellen ein kombiniertes Verfahren zur Optimierung beider Teilprobleme vor, bestehend aus einem auf Prioritätsregeln basierenden Algorithmus genannt *Approach by Localization* (AL) und einem genetischen Algorithmus (GA) zur breiteren Suche (Kacem & Hammadi u. a. 2002). Der AL-Ansatz basiert auf einer Menge an Prioritätsregeln zur Allokation von Vorgängen auf Produktionsmittel. Aufbauend auf den Ergebnissen des AL findet eine weitere Optimierung durch einen GA statt. Eine Erweiterung kann ebenfalls zur Generierung von Pareto-Lösungen für das FJSP angewendet werden (Kacem & Hammadi u. a. 2002). Insgesamt gibt es fünf als Kacem-Probleme bekannte Steuerungsprobleme. Die Steuerungsaufgabe entspricht einer vereinfachten Matrixproduktion ohne Variabilität in der Vorgangsreihenfolge und ohne Rüst- und Transportzeiten. Als Zielgrößen dienen die Gesamtarbeitsmenge, der Makespan und die maximal Arbeitsbelastung (Kacem & Hammadi u. a. 2002). Die Kacem-Probleme zählen nach der Klassifikation aus Abschnitt 2.2.2 zu den $\beta_6 = p_{km}$ -Problemen, bei der die Bearbeitungszeit sowohl vom Vorgang als auch vom Produktionsmittel abhängt.

Li & Gao (2016) nutzen ein hybrides Verfahren zur Optimierung der Gesamtfertigungsdauer, bestehend aus einem GA als Algorithmus der breiten Suche und Tabusuche für die lokale

Suche. Der vorgestellte Algorithmus kann sehr gute Ergebnisse für bekannte Benchmark-Probleme erzielen, unter anderem die Kacem-Probleme (Li & Gao 2016).

Wang & Gao u. a. (2010) nutzen ebenfalls einen genetischen Algorithmus, um das FJSP zu lösen. Der vorgestellte Mehrzieloptimierer verbessert die drei kapazitätsbezogenen Zielgrößen des Kacem-Problems ohne Berücksichtigung von Transportzeiten oder Rüstzeiten. Da die Population genetischer Algorithmen die Tendenz hat, gegen eine Lösung zu konvergieren, was bei der Pareto-Optimierung nachteilig ist, entwickeln die Forscher ein Konzept zur Steigerung der Entropie. Um die Entropie der Lösungsmenge zu erhöhen, wird die Reproduktionswahrscheinlichkeit ähnlicher Lösungen gesenkt. Der vorgestellte Ansatz kann einen Großteil der Pareto-Punkte des Kacem-Problems finden (Wang & Gao u. a. 2010).

Gao & Sun u. a. (2008) wenden einen hybriden genetischen Algorithmus auf das FJSP an. Die Startlösung für die lokale Suche wird mit einem genetischen Algorithmus erzeugt. Dieser basiert auf einem Maschinenallokationsvektor, der angibt, auf welchem Produktionsmittel ein Vorgang ausgeführt wird, und einem Vorgangssequenzvektor, der die Reihenfolge der Vorgänge auf einem Produktionsmittel abbildet. Bei der Terminierung wird nur der frühestmögliche Starttermin betrachtet. Die Selektion der Population der Folgegeneration wird durch den genetischen Algorithmus auf der Menge der Kinder und Eltern durchgeführt. Bevor jedoch die Selektion stattfindet, werden die einzelnen Lösungen durch eine lokale Suche verbessert. Die lokale Suche *Variable Neighborhood Descend* (VND) basiert auf den Nachbarschaftstrukturen *Local Search by Moving One Operation* (LSONE), siehe Kapitel 2.4.3 und *Local Search by Moving Two*. Den ersten Schritt der lokalen Suche bildet die Transformation des Problems in einen gerichteten Graphen, vgl. Abschnitt 2.4.2. Die simultane Löschung und Neuzuweisung von zwei Vorgängen im Rahmen von *Local Search by Moving Two* erhöht die Komplexität jedoch exponentiell, sodass im gleichen Zeitraum ein geringerer Teil des benachbarten Lösungsraums abgesucht werden kann (Gao & Sun u. a. 2008). Die Forscher erzielen sehr gute Lösungen auf dem Kacem-Benchmark-Problem. Allerdings ist die gewählte Einkodierung des FJSP nicht auf Probleme mit alternativen Vorgangssequenzen übertragbar, da die Reihenfolgebeziehung der Vorgänge nur über die Position im Vorgangssequenzvektor abgebildet wird. Alternative Reihenfolgen sind somit nicht abbildbar.

Der evolutionäre Algorithmus von Chiang & Lin (2012) wird ebenfalls am Kacem-Problem ausgerichtet. Um die Fähigkeiten zur Pareto-Optimierung zu stärken, setzt die Forschergruppe auf einen Mechanismus, der gleiche Chromosomen mit Hilfe von fünf Heuristiken so lange verändert, bis sich die Zielfunktionswerte von der ursprünglichen Lösung unterscheiden. Eine Verschlechterung wird akzeptiert, da der Fokus auf der Erhöhung der Entropie der Lösungsmenge liegt. Zur Generierung einer möglichst guten Startlösung werden mehrere an das Kacem-Problem angepasste Heuristiken angewendet. Der vorgestellte Algorithmus

kann fast alle Kacem-Pareto-Punkte erreichen (Chiang & Lin 2013). Die Autoren stellen einen verwandten hybriden genetischen Algorithmus kombiniert mit lokaler Suche vor. Die Erweiterung genetischer Algorithmen durch eine lokale Suche zur Vermeidung vorzeitiger Konvergenz wird als Memetischer Algorithmus bezeichnet. Die lokale Suche verwendet eine Linearkombination aus den drei Kacem-Zielgrößen mit einem starken Übergewicht für den Makespan. Auch dieser kombinierte Ansatz erzielt gute Ergebnisse auf dem Kacem-Benchmark (Chiang & Lin 2012).

Der Klasse der Schwarmintelligenz-Verfahren sind die Arbeiten von Scholz-Reiter & Hamann (2008) zuzurechnen, in denen das Verhalten von Ameisen imitiert wird, um die Durchlaufzeit für eine stark vereinfachte Matrixproduktion zu minimieren. Übereinstimmend mit anderen Publikationen konnte gezeigt werden, dass diese Meta-Heuristik gute Ergebnisse bei geringer Dynamik und großer Prozesszeitvarianz liefert und anderen Prioritätsregeln, beispielsweise *Queue Length Estimator* QLE überlegen ist (Scholz-Reiter & Hamann 2008; Armbruster & De Beer u. a. 2006). Aufgrund der inhärenten Latenz der Pheromon-basierten Verfahren sind diese nicht in der Lage, schnell auf Änderungen zu reagieren und somit ungeeignet für Systeme mit hoher Dynamik. Neuere Arbeiten, beispielsweise von Huang & Yang u. a. (2013) zur Optimierung der Termintreue, zeigen, dass mit Hilfe von mehreren Pheromonen das Verfahren angepasst werden kann, um die komplexe Produktionsrealität eines Job Shops mit Rüstzeiten abzubilden. Trotz Reduktion der Rechenzeit bleiben der exponentielle Charakter und die Trägheit des Verfahrens weiter bestehen (Huang & Yang u. a. 2013).

3.2.3 Maschinelles Lernen

Die situationsabhängige Maschinenbelegung mit Vorgängen kann mit Hilfe von Ansätzen des Maschinellen Lernens realisiert werden. Einige Ansätze behandeln die Wahl der richtigen Heuristik, die im Abschnitt 3.2.2.1 vorgestellt wurden. Im Folgenden soll der Fokus auf Ansätzen liegen, die Maschinelles Lernen direkt zur Steuerung der Produktion einsetzen.

Stegherr (2000) stellt ein Verfahren zur Steuerung eines variantenreichen Flow Shops mit Hilfe von *Reinforcement Learning* zur Maximierung mehrerer Zielgrößen vor (Stegherr 2000). Der betrachtete Anwendungsfall der Halbleiterteilfertigung unterscheidet sich jedoch stark von einer Matrixproduktion.

Das RL-Verfahren von Zhang & Zheng u. a. (2012) wird genutzt, um stochastisch eingesteuerte Aufträge auf eine Reihe paralleler Maschinen in einem fiktiven Produktionsszenario zu

verteilen. Ziel der Steuerung ist die Optimierung der Termintreue. Es konnte gezeigt werden, dass sich nach entsprechender Einlernphase für spätere Perioden bessere Ergebnisse erzielen lassen als mit prioritätsregelbasierten Heuristiken (Zhang & Zheng u. a. 2012).

Kuhnle (2020) zeigt, dass mit Hilfe von *Reinforcement Learning* auch Steuerungsaufgaben in einer komplexen Werkstattfertigung mit stochastischen Bearbeitungszeiten und Maschinenausfällen übernommen werden können. Die Simulationsergebnisse zeigen, dass bessere Ergebnisse hinsichtlich Durchlaufzeit und Auslastung erzielt werden als bei einem prioritätsregelbasierten Vorgehen (Stricker & Kuhnle u. a. 2018). Das vorgestellte Verfahren nutzt einen *Reinforcement Learning* Agenten, um zustandsabhängig zu entscheiden, welcher Produktionsauftrag an welcher Maschine als nächstes bearbeitet werden soll. Das Training erfolgt anhand einer ereignisdiskreten Simulation. Optimiert wird eine Linearkombination aus Auslastung als auch Durchlaufzeit (Kuhnle 2020).

Waschneck & Reichstaller u. a. (2018) stellen ein *Deep-Q-Learning*-Verfahren für die Produktionssteuerung einer Job Shop Produktion am Beispiel der Frontend Fertigung in der Halbleiterteilerei vor. Das betrachtete Produktionsszenario beinhaltet feste Produktionssequenzen und Ressourcenpools aus parallelen Produktionsmitteln, die ausfallen können ($\beta_9 = \text{breakdown}$) und sequenzabhängige Rüstzeiten ($\beta_8 = rz_k$). Beim Eintreffen eines neuen Auftrags oder bei Fertigstellung eines Produktionsschritts durch ein Produktionsmittel wird eine neue Steuerungsentscheidung durch das System getroffen. Dieser reaktive Charakter ermöglicht ein Umsteuern bei Abweichungen. Jede Gruppe aus Produktionsmitteln wird durch einen RL-Agenten gesteuert. Das Training erfolgt in zwei Phasen anhand einer ereignisdiskreten Simulation. In der ersten Phase werden die Agenten abwechselnd trainiert, die übrigen Agenten werden in dieser Phase durch problemspezifische Heuristiken gesteuert. An dieser Stelle wird auf Domänenwissen zurückgegriffen. In der zweiten Trainingsphase wird weiterhin jeweils ein Agent trainiert, die anderen Agenten jedoch verhalten sich gemäß ihrer erlernten Erfahrung. Die Agenten erhalten ihre Belohnung bezogen auf die kumulierte Verspätung des Gesamtsystems (kooperatives Lernen), sodass die Agenten vorausschauendes Verhalten erlernen. Die Untersuchungen bestätigen die Eignung von *Reinforcement Learning* zur Produktionssteuerung und übertreffen die Zielerreichung von prioritätsregelbasierten Heuristiken (Waschneck & Reichstaller u. a. 2018).

Alle Ansätze des *Reinforcement Learnings* zur Produktionssteuerung erfordern den Aufbau eines realitätsnahen Simulationsmodells zum Training des Algorithmus. Trotz der Möglichkeit zum Generalisieren erfordern größere Systemänderungen eine erneute Trainingsphase. Dies würde sowohl im Vorfeld der Nutzung als auch bei Änderungen die tatsächliche Flexibilität und Wandlungsfähigkeit der Matrixproduktion einschränken.

3.2.4 Ansätze der robusten Produktionssteuerung und Verteilungsänderungsdetektion

Die Produktionssteuerung einer Matrixproduktion operiert in einem von Unsicherheit geprägten Umfeld. Die Bearbeitungs-, Rüst-, Transportzeiten sind nicht deterministisch, sondern unterliegen einer Verteilung. Zusätzlich können Produktionsmittel durch Ausfälle temporär nicht verfügbar sein. Im Rahmen der robusten PPS geht es sowohl um die Generierung robuster Belegungspläne als auch um die robuste Terminierung der Vorgänge. Da das Vorliegen realistischer Vorgabezeiten Grundlage für die Erstellung robuster Pläne ist, sollte die Steuerung die Vorgabezeiten ermitteln und aktualisieren können. Der Umgang mit Unsicherheit im Zusammenhang mit der Produktionssteuerung ist bereits Gegenstand vieler Forschungsarbeiten. Im Folgenden werden relevante Ansätze aus dem Bereich der robusten Optimierung vorgestellt sowie Arbeiten, die sich mit der Schätzung und Erkennung von Verteilungen der Vorgabezeiten beschäftigen.

Liegen keine Informationen zur Verteilung vor, bieten Fuzzy-Sets eine gute Möglichkeit, die Unsicherheit zu berücksichtigen (Verderame & Elia u. a. 2010). Für die Steuerung einer Matrixproduktion können anhand der Rückmeldedaten aus der Produktion jedoch Verteilungsinformationen erhoben werden, Daher werden Fuzzy-Sets nicht weiter betrachtet.

Für Probleme mit bekannten Verteilungsfunktionen kann mittels *Parametric Programming* die Lösungsmenge abhängig von Variablen ausgedrückt werden. Der Ansatz von Li & Ierapetritou (2008) ermöglicht es, ein Steuerungsproblem, das als MILP formuliert werden kann, in Abhängigkeit der Unsicherheitsparameter, hier Eilaufträge und Maschinenausfälle, exakt zu lösen, sodass der Gewinn maximiert wird. Tritt eines dieser Ereignisse ein, ist sofort ein Alternativplan vorhanden. Dies ermöglicht schnelle Reaktionszeiten, ist allerdings auf bereits bekannte Unsicherheiten limitiert und erfordert einen sehr hohen Berechnungsaufwand, um die parametrisierte Lösungsmenge zu erzeugen (Li & Ierapetritou 2008). *Parametric Programming* ist daher ungeeignet für komplexe, sich wandelnde Probleme mit vielen Unsicherheitsfaktoren.

Die Ansätze des *Chance Constraint Programmings* begrenzen die Wahrscheinlichkeit, mit der Verschiebungen eintreten können und ersetzen entsprechend die Zufallsvariablen durch deterministische Ungleichungen. Petkov & Maranas (1997) stellen ein Modell zur Maximierung des Gewinns einer Mehrlinienproduktion über mehrere Planungsperioden bei unsicherer Nachfrage vor. Die Planung bleibt auf die Produktionsmengen der einzelnen Produkte in den Perioden beschränkt (Petkov & Maranas 1997). Dennoch zeigt das Vorgehen, wie die Unsicherheit eines Parameters auf die Zielfunktion übertragen und limitiert werden kann. Dieser Ansatz führt bei großen Problem ebenfalls zu hohem Rechenaufwand und ist somit nicht anwendbar für das Problem der Matrixproduktion.

Ein alternatives Vorgehen, geeignet für komplexere Probleme, ist die Aufteilung des Gesamtproblems in ein deterministisches Teilproblem und ein nachgelagertes stochastisches Problem. Die Stochastik des nachgelagerten Problems kann über Szenarien oder Verteilungen abgebildet werden. Bonfill & Espuña u. a. (2005) nutzen Szenarien, um Schwankungen der Bearbeitungszeiten in einer Produktionslinie der Prozessindustrie abzubilden. Optimierungsziele sind ein minimaler Makespan und minimale Wartezeiten. Das *Scheduling*-Problem wird als MILP formuliert. Je nach Risikoneigung wird am Ende die Lösung gewählt, die in den relevanten Szenarien die beste gewichtete Zielerreichung erzielt (Bonfill & Espuña u. a. 2005). Das Vorgehen ist geeignet, bei bekannten und enumerierbaren Szenarien die Wahrscheinlichkeit einer geringen Zielerreichung zu begrenzen. Die Anzahl der Szenarien wächst jedoch exponentiell mit der Anzahl unsicherheitsbehafteter Parameter.

Policella & Cesta u. a. (2008) lösen ein generisches *Scheduling*-Problem einer Linienfertigung im ersten Schritt unter Annahme einer deterministischen Umwelt. Optimierungsziele sind Makespan und Robustheit. Im zweiten Schritt werden die Vorgänge verschoben oder neu allokiert. Somit verringern sich die Abhängigkeiten zwischen den Produktionsmitteln, sodass die Robustheit gesteigert wird, ohne den Makespan zu verschlechtern. Kerngedanke ist dabei, nach Möglichkeit alle Vorgänge eines Produktionsauftrags auf dem selben Produktionsmittel durchzuführen, da ohnehin durch die Vorgangsfolge eine Abhängigkeit besteht. Würden die Vorgänge zwischen Produktionsmitteln aufgeteilt, kämen weitere Abhängigkeiten durch die Produktionsmittel hinzu. Benötigt ein Produktionsauftrag mehrere Produktionsmittel, so sollten die Wechsel zwischen den Produktionsmitteln möglichst minimiert werden (Policella & Cesta u. a. 2008). Ein Vorteil dieses Ansatzes liegt darin, dass der initiale Plan auch durch ein beliebiges Optimierungsverfahren erstellt werden kann. Nachteilig ist, dass das Verfahren nicht anwendbar ist bei anderen Zielgrößen wie beispielsweise der Verspätung.

Der Ansatz von Jiménez (2012) beschäftigt sich mit der robusten Optimierung eines flexiblen Job Shops. Der Autor teilt das Problem in zwei Teilprobleme. Im ersten Schritt wird ein möglichst optimaler initialer Belegungsplan im Hinblick auf den Makespan erstellt. Im zweiten Schritt werden kritische Produktionsmittel identifiziert, für die zusätzliche Schlupfzeiten eingebracht werden. Kritisch gelten Produktionsmittel bezogen auf ein Zeitintervall, wenn die Anzahl konsekutiver Operationen einen zu definierenden Schwellenwert überschreitet. Verteilungsinformationen werden nicht berücksichtigt. Auf den kritischen Produktionsmitteln werden zusätzliche Schlupfzeiten hinzugefügt. Die Höhe muss manuell definiert werden. Die Ergebnisse zeigen, dass das zielgerichtete Einbringen von Schlupfzeiten auf kritischen Produktionsmitteln bei Abweichungen zu einer geringeren Steigerung des Makespans führt als Methoden, die weniger zielgerichtet vorgehen (Jiménez 2012).

Jamili (2016) stellt zwei Robustheitsmaße zur zielgerichteten Robustheitssteigerung vor und

vergleicht die Effizienz unterschiedlicher Lösungsverfahren. Berücksichtigt werden sowohl Verzögerungen, die durch vorgelagerte Vorgänge auf demselben Produktionsmittel entstehen, als auch Verzögerungen, die durch Vorgänge des selben Produktionsauftrags auf anderen Produktionsmitteln hervorgerufen werden. Anders als Jiménez (2012) oder Policella & Cesta u. a. (2008) wird die Robustheitsanforderung als zusätzliche Nebenbedingung des Optimierungsproblems formuliert. Das eine Robustheitsmaß beschränkt die maximale erwartete Verspätung, das andere beschränkt die Wahrscheinlichkeit für das Eintreten einer bestimmten Verspätung. Der Vergleich zwischen *Branch and Bound* als exaktes Lösungsverfahren und der Meta-Heuristik Partikel-Schwarm-Verfahren zeigt, dass nur der heuristische Ansatz in der Lage ist, das Optimierungsproblem effizient zu lösen. Untersucht wurde allerdings nur die Makespan-Optimierung eines JSSP ohne Transport- und Rüstzeiten (Jamili 2016).

Echsler (2020) entwickelt einen prädikativ-reaktiven Ansatz zur Steuerung einer Matrixproduktion, angelehnt an den automobilen Karosserierohbau. Im ersten Schritt wird das Steuerungsproblem als lineares Problem, gemischt mit Elementen eines *Constraint*-Problems, formuliert und gelöst. Optimierungsziel ist die Minimierung des maximalen Abstands zwischen dem geforderten und geplanten Fertigstellungstermin über alle Aufträge. Danach werden gezielt weitere Schlupfzeiten eingebracht, um die Robustheit zu erhöhen. Kommt es im Betrieb dennoch zu Abweichungen, wird innerhalb eines Zeitkorridors umgeplant. Für die reaktive Umplanung wird ein RL-Agent trainiert, der stationsabhängig geeignete Prioritätsregeln erlernt, nach denen Aufträge an Stationen bearbeitet werden. Der integrierte Ansatz erlaubt es, einen robusten Ablaufplan für eine Matrixproduktion zu erstellen und bei Abweichungen über Prioritätsregeln reaktiv zu steuern (Echsler 2020). Für die Auswahl der Prioritätsregeln muss ein RL-Agent trainiert werden. Die Modellierung des Steuerungsproblems als lineares Problem führt zudem zu einem hohen Rechenaufwand zur Erstellung des initialen Plans.

Eine zentrale Eingangsgröße für die PPS sind die Vorgabezeiten der einzelnen Vorgänge. Viele Ansätze der robusten Optimierung nutzen zudem Informationen über die zugrundeliegenden Verteilungen, um gezielt Schlupfzeiten einzubringen. Im Folgenden werden daher Forschungsarbeiten vorgestellt, die sich mit der Ermittlung und Aktualisierung der Verteilungsinformationen beschäftigen. Für einen Überblick sei auf Kadwe & Suryawanshi (2015) und Gama & Žliobaitundefined u. a. (2014) verwiesen. Ein *Concept-Drift* liegt vor, wenn sich die Verteilung einer Zufallsvariable nach einer Phase der Stabilität ändert (Schlimmer & Granger 1986). Dies ist der Fall, wenn sich zu zwei Zeitpunkten t_0 und t_1 die gemeinsame Verteilung zwischen den Zufallsvariablen X und der Zielvariable y geändert hat, siehe Formel 3.2.1 (Gama & Žliobaitundefined u. a. 2014).

$$\exists X : p_{t_0}(X, y) \neq p_{t_1}(X, y)$$

Bei der *Concept-Drift*-Erkennung geht es darum, diese Änderungen der Verteilung zu detektieren. Im Zusammenhang mit dem *Scheduling* ist das Ziel, die Grundgesamtheit der Beobachtungen so anzupassen, dass nur die Beobachtungen aus der aktuell gültigen Verteilung für die Parameterschätzung genutzt werden. Im Folgenden werden Methoden zur *Concept Drift*-Erkennung vorgestellt. Sie basieren auf dem Verteilungsvergleich unterschiedlicher Datenfenster sowie Ansätzen zur Detektion der Änderung anhand von Kenngrößen. Ein Datenfenster beinhaltet zeitlich aufeinanderfolgende Beobachtungen.

Die Grundidee des Verteilungsvergleichs besteht darin, die Verteilung von zwei Datenfenstern zu vergleichen. Das Verfahren ist sowohl auf univariate als auch auf multivariate Verteilungen anwendbar. Die Nullhypothese besagt, dass die Verteilungen beider Datenfenster gleich sind. Es existieren verschiedene Ansätze zur Ausgestaltung der Datenfenster, deren Verteilungen miteinander verglichen werden (Gama & Žliobaitundefined u. a. 2014). Gemeinsam ist, dass eines der Datenfenster ältere Datenpunkte enthält und für die stabile Verteilung steht. Ein weiteres Datenfenster beinhaltet die neusten Datenpunkte und dient der Änderungsdetektion. Im Folgenden soll ein Ansatz vorgestellt werden, der dynamische Fenstergrößen nutzt. Dieses Verfahren ermittelt das größte Datenfenster, für das der Verteilungsvergleich auf eine Übereinstimmung schließen lässt. Somit wird die größtmögliche Datenbasis genutzt.

Das Verfahren ADWIN (**AD**aptive **WIN**dowing) nach Bifet & Gavaldà (2006) und Bifet & Gavaldà (2007) teilt die Grunddaten in zwei benachbarte Teilfenster auf. Eingangsgrößen des Verfahrens sind das gewünschte Konfidenzniveau der Entscheidung sowie eine Reihe unabhängiger Beobachtungen. Das Verfahren vergleicht die Mittelwerte der Teilfenster. Legt der Verteilungsvergleich nahe, dass die Beobachtungen nicht aus derselben Verteilung stammen, wird das stabile Fenster verkleinert, das die älteren Daten enthält. Auf diese Weise wird iterativ die größtmögliche Datenbasis ermittelt, für welche die Hypothese, dass die Daten derselben Verteilung entstammen, nicht abgelehnt werden kann (Bifet & Gavaldà 2007). ADWIN erreicht eine Komplexität von $O(\log w)^1$ und ist damit für die Integration in ein echtzeitnahes Steuerungssystem geeignet (Gama & Žliobaitundefined u. a. 2014).

Einen alternativen Ansatz bieten Regelkarten zur *Concept Drift*-Detektion. Analog zur statistischen Prozesskontrolle werden Kennwerte, wie der Mittelwert oder die Standardabweichung, in einem Diagramm abgetragen. Wird die Warn- oder Alarmgrenze über- oder unterschritten, sollte die Verteilung neu parametrisiert werden. Bei asymmetrischen Verteilungen können die Grenzen ebenfalls asymmetrisch gewählt werden (Gama & Žliobaitundefined u. a. 2014).

Die EWMA-Qualitätsregelkarte (**E**xponentially **W**eighted **M**oving **A**verage) gewichtet die

¹Die sogenannte O-Notation beschreibt die Veränderung der Ausführungszeit eines Algorithmus abhängig von der Änderung der Größe der Eingabedaten.

neusten Beobachtungen stärker. Die EWMA-Karte kann zur Erkennung eines *Concept Drift* eingesetzt werden (Ross & Adams u. a. 2012). Das Verhalten der *Concept-Drift*-Erkennung wird wie die ADWIN-Methode anhand von Marktdaten aus dem Elektrizitätsmarkt bewertet. Grundsätzlich ist die EWMA-Qualitätsregelkarte für normalverteilte Prozesse geeignet. Untersuchungen dieses Verfahrens bei nicht-normalverteilten Prozessen haben jedoch gezeigt, dass das Verfahren relativ robust gegen nicht-normalverteilte Prozesse reagiert. Die Leistungsfähigkeit hängt von der Schiefe, Symmetrie und der grundlegenden Form der Verteilung ab (Maravelakis & Panaretos u. a. 2005).

3.3 Monte Carlo Tree Search in der Produktionssteuerung

Erste Ansätze zur Nutzung von MCTS für die Produktionsplanung stammen von Chaslot & De Jong u. a. (2006). Gegenstand ihrer Forschung ist die Erzeugung einer Aktionssequenz zur Maximierung der Ausbringung unter Einhaltung von zeitlichen und monetären Restriktionen in einer deterministischen Umwelt. Der Vergleich von MCTS mit *Fixed Planning Heuristic* (FPH) und *Evolutionary Planning Heuristic* (EPH) bei Problemen unterschiedlicher Größe zeigt, dass MCTS insbesondere bei sehr komplexen Problemen überlegen ist und eine Alternative zu evolutionären Verfahren bietet (Chaslot & De Jong u. a. 2006). Die zugrundeliegende Produktion ist allerdings stark abstrahiert und MCTS wird nahezu in Reinform ohne Modifikationen umgesetzt.

Im Bereich des *Schedulings* wird MCTS von Runarsson & Schoenauer u. a. (2012) zur unikriteriellen Optimierung (Makespan) eines vereinfachten JSSP eingesetzt und mit der stark explorativen Pilot-Methode verglichen, die auf unterschiedlichen Prioritätsregeln basiert. Der für die Richtung des Baumwachstums entscheidende Schritt der Selektion wird bei MCTS mittels UCT-Funktion kombiniert mit dem ϵ -greedy-Verfahren durchgeführt, wodurch eine sinkende Exploration bei steigender Iterationszahl erzielt wird. Für das *Rollout* wird die *default policy* angewandt, die auf der zufälligen Wahl möglicher Aktionen basiert. Die Ergebnisse zeigen, dass MCTS auf kleinen und mittleren Probleminstanzen bessere Ergebnisse erzielt als die Pilotmethode. MCTS ist auf großen Problemen zwar erheblich schneller, erreicht jedoch eine niedrigere Lösungsgüte. Die schlechte Lösungsgüte liegt in der zufälligen Auswahl der Aktionen im *Rollout* (*light rollout*) begründet, während die Pilot-Methode hier durch die eingesetzten Prioritätsregeln Domänenwissen nutzen kann (Runarsson & Schoenauer u. a. 2012). Dies macht deutlich, dass es sinnvoll ist, zur Sicherstellung einer hohen Lösungsgüte bei gleichzeitig kurzer Lösungszeit, Anpassungen in der *Rollout*-Phase vorzunehmen.

Loth & Sebarg u. a. (2013) greifen die Arbeiten von Runarsson & Schoenauer u. a. (2012) auf und stellen einen hybriden Ansatz basierend auf *Constraint Programming* (CP) und MCTS für die Makespan-Optimierung eines JSSP vor (Loth & Sebarg u. a. 2013). Der MCTS-Algorithmus wird in der Selektions- und *Rollout*-Phase angepasst, um das Zusammenspiel mit dem CP zu verbessern. CP baut einen Suchbaum auf, bei dem jeder Zustand einem bestimmten Wert einer Entscheidungsvariable entspricht. Der linke Kindknoten eines Zustands entspricht dabei der wiederholten Zuweisung des bislang besten Wertes. Die Autoren untersuchen vier unterschiedliche Selektionsfunktionen, zum einen die zufällige Wahl des linken oder rechten Knotens mit einer Gleichverteilung sowie mit unterschiedlichen Gewichten und zum anderen zwei UCB-basierte Selektionsfunktionen. Im Gegensatz zu MCTS bricht der Baumaufbau ab, wenn eine unzulässige Lösung gefunden wird. Daher wird in der Regel kein Terminalzustand erreicht, der anhand des Zielkriteriums bewertet werden könnte. Als Ersatzkriterium wird deshalb für die UCB-basierten Selektionsfunktionen die Baumtiefe, bei der die Unzulässigkeit erkennbar wird, genutzt. In der *Rollout*-Phase kommt das *Depth-First-Search* (DFS) zur Anwendung. Dieses Verfahren fokussiert die Suche auf den benachbarten Lösungsraum um die bislang beste Lösung. Im Gegensatz zu den Arbeiten von Runarsson & Schoenauer u. a. (2012) wird der Lösungsraum zielgerichteter abgesucht. Die Untersuchung der unterschiedlichen Selektionsfunktionen in Verbindung mit dem *Rollout*-Verfahren zeigt, dass die UCB-basierten Selektionsfunktionen eine höhere Lösungsgüte und einen steileren Anstieg der Lösungsgüte erzielen als die Vergleichsfunktionen. Die Lösungsgüte dieses hybriden Verfahrens für das untersuchte Benchmark-Problem ist zudem vergleichbar mit anderen Lösungsverfahren (Loth & Sebarg u. a. 2013). Durch die Kombination mit CP kann jedoch der Vorteil von MCTS, die eigentliche Zielfunktion zur Bewertung von Zuständen zu nutzen, nicht angewandt werden.

Weitere Arbeiten zu MCTS im Zusammenhang mit FJSP belegen das hohe Potenzial von MCTS im Bereich des *Schedulings* und die gute Erweiterbarkeit des Basisalgorithmus. Der Fokus der Arbeiten liegt in der Weiterentwicklung von MCTS als effizientes Lösungsverfahren für FJSP. Daher wird die Güte des Verfahrens anhand der Benchmark-Probleme von Kacem bemessen (Kacem & Hammadi u. a. 2002). Im Folgenden werden die wichtigsten Anpassungen vorgestellt.

Wu & Wu u. a. (2013) erweitern MCTS, sodass eine Pareto-Menge nicht-dominierter Lösungen für die Mehrzieloptimierung generiert wird (MO-MCTS). Anhand des Kacem-Benchmarks kann gezeigt werden, dass der Algorithmus die meisten der bekannten Pareto-Punkte für die Kacem-Probleme in 4 bis 112 Sekunden finden kann (Wu & Wu u. a. 2013). Der Algorithmus erhält ein festes Iterations-Budget pro Ausführung. Das Ergebnis des MO-MCTS-Laufs wird anschließend als Startlösung für die nächste Optimierung genutzt, bis alle Aufträge

eingepflegt sind. Die Ergebnisqualität kann durch Modifikationen in den MCTS-Phasen sowie die Anwendung von lokaler Suche erreicht werden. Die Autoren empfehlen insbesondere die Anpassung des Algorithmus unter Zuhilfenahme von Domänenwissen (Wu & Wu u. a. 2013). Da der Makespan das am schwierigsten zu erreichende Zielkriterium der Kacem-Probleme ist, wird in der Selektionsphase ausschließlich nach dem Makespan entschieden. Um die Anzahl möglicher Aktionen zu verringern, werden in der Expansionsphase nur die Vorgang-Produktionsmittel-Kombinationen mit der geringsten Bearbeitungsdauer berücksichtigt. Um die Größe des Suchbaums zu limitieren, kommt *Subtree-Pruning* zum Einsatz. Hierbei werden Knoten aus dem Baum entfernt, wenn bereits eine Pareto-Lösung existiert, die den betrachteten Knoten dominiert (Wu & Wu u. a. 2013).

Bei den Kacem-Problemen hängt die Bearbeitungsdauer sowohl vom Vorgang als auch vom Produktionsmittel ab. Um diese problemspezifischen Unterschiede gezielt zu nutzen, werden in der *Rollout*-Phase daher *heavy rollouts* mit einer ϵ -greedy-Strategie kombiniert. Statt in der *Rollout*-Phase zufällig Aktionen auszuwählen, werden mit Wahrscheinlichkeit ϵ die Aktionen mit der niedrigsten Prozesszeit für die Kombination aus Produktionsmittel und Produktionsprozess gewählt. Um die Diversität der *Rollouts* zu erhöhen, wird mit Wahrscheinlichkeit $(1 - \epsilon)$ eine zufällige Aktion selektiert (Wu & Wu u. a. 2013).

Im Anschluss an jedes *Rollout* wird eine lokale Suche als zusätzliche Phase angewendet. Der verwendete LSONE-Algorithmus stammt aus verwandten Forschungsarbeiten zu hybriden genetischen Algorithmen im Kontext des *Scheduling*s (Gao & Sun u. a. 2008).

In der *Backpropagation*-Phase wird der von Gelly & Silver (2007) vorgestellte *Rapid Action Value Estimate*, kurz RAVE-Algorithmus angewendet. Ein Nachteil des UCT-Algorithmus nach Kocsis & Szepesvári (2006) ist, dass keine Generalisierung zwischen Zuständen erfolgt. Jede Aktion a wird ausschließlich im Kontext des Zustands s bewertet $Q(s, a)$. Ist ein Zustandsaktionspaar unbekannt, liegt somit keine Bewertung vor. Bei wenigen *Rollouts* ausgehend von einem Zustand ist die Varianz der Bewertungen aufgrund der geringen Anzahl erreichter Terminalzustände zudem hoch (Gelly & Silver 2007). RAVE betrachtet den gesamten Teilbaum unterhalb des expandierten Knotens und mittelt die Bewertung aller Terminalzustände, die in ihrem Pfad die Aktion a enthalten. Der RAVE-Wert der Aktion ergibt sich dann als gemittelter Zielfunktionswert der Terminalzustände. RAVE bewertet demnach die Aktion, losgelöst vom Zustand, in dem sie gewählt wurde. Diese Grundidee ist bereits als *All Moves as First*, kurz AMAF, bekannt (Brügmann 1993). Abbildung 3.1 illustriert das Verfahren. Die Selektionsfunktion wird dahingehend angepasst, dass sowohl der RAVE-Wert als auch der UCT-Wert genutzt werden, siehe Formel 3.3.1 (Gelly & Silver 2011). Insbesondere wenn ein Knoten selten besucht wurde, ist der UCT-Wert ungenau, da

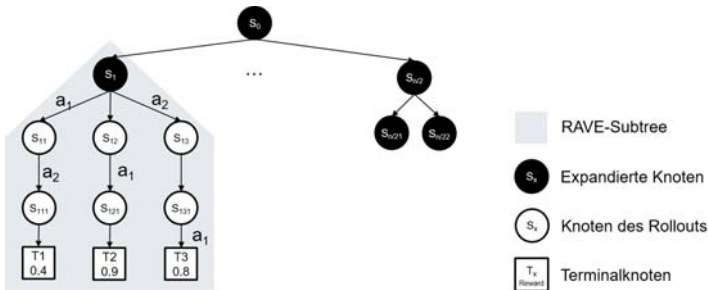


Abbildung 3.1: Ausgehend vom Knoten S_1 werden drei Rollouts durchgeführt. Die Bewertung der Aktion a_1 ergibt sich als Mittelwert der Zielfunktionen der Terminalzustände T_1, T_2, T_3 , da bei allen Zuständen die Aktion a_1 gewählt wurde. Die Bewertung für a_2 basiert nur auf T_1, T_3

er auf wenigen Terminalzuständen beruht. Der Faktor $\beta(s, a)$ gibt an, ab welcher Besuchszahl der UCT-Wert gleich hoch oder höher als der RAVE-Wert gewichtet werden soll.

$$Q_{UR} = \beta(s, a)Q_{RAVE}(s, a) + (1 - \beta(s, a))Q_{UCT}(s, a) \quad 3.3.1$$

Um die Bewertung unbekannter Zustände weiter zu verbessern, wird Vorwissen eingesetzt. Dieses rührt aus den Ergebnisse der lokalen Suche, die nach dem *Rollout* durchgeführt wird. Hierbei wird das Tripel aus Vorgang, Allokation auf ein Produktionsmittel und Position des Vorgangs in der Vorgangssequenz des Produktionsauftrags anhand des durchschnittlich erreichten Zielfunktionswerts in der lokalen Suche bewertet. (Wu & Wu u. a. 2013).

Chou & Liang u. a. (2015) nehmen eine Weiterentwicklung des vorgestellten Verfahrens von Wu & Wu u. a. (2013) vor. Das neue Verfahren nutzt weiterhin *Subtree Pruning*, RAVE, LSONE und Vorwissen. Der Unterschied besteht darin, dass nicht nur ein Knoten als Ergebnis eines Durchlaufs behalten wird, sondern die Menge der besten Knoten. Anders als beim ursprünglichen Algorithmus ist es somit möglich, die erste Entscheidung später zu korrigieren. Die Ergebnisse erhöhen vor allem die Ergebnisstabilität. Es kann gezeigt werden, dass die durchschnittliche Anzahl der gefundenen Pareto-Punkte steigt (Chou & Liang u. a. 2015). Die Auswertung zeigt allerdings auch, dass das Verfahren stark auf die Kacem-Probleme zugeschnitten ist. Wird der Parameter C , verantwortlich für die Exploration, größer als 0 gewählt, verschlechtern sich die Ergebnisse. Das zeigt, dass der Algorithmus nahezu perfekt auf das Problem zugeschnitten ist und somit nur schlecht generalisieren kann.

Lu & Chiu u. a. (2016) untersuchen das Verhalten eines Mehrziel-MCTS-Optimierers im

Vergleich zu hybriden genetischen Algorithmen. Sie zeigen, dass der MO-MCTS auf unterschiedlichen Vergleichsproblemen die Anzahl der Pareto-Lösungen erweitert (Lu & Chiu u. a. 2016). Dies belegt die gute Generalisierungsfähigkeit des Ansatzes und die Effizienz des Verfahrens auf komplexen Problemen. Deshalb kommen vor allem drei Modifikationen zum Einsatz, welche die Expansions-, *Rollout*-, und Selektionsphase betreffen. Einschränkend ist festzustellen, dass die verwendeten Vergleichsprobleme in Bezug auf die untersuchten Zielfunktionen und Restriktionen identisch sind (Lu & Chiu u. a. 2016). Die Modellierung des *Scheduling*-Problems nutzt den *Sequential Operation Machine Assignment*-Ansatz, kurz SOMA. Jede Aktion besteht aus dem Tupel (auftragsspezifischer Vorgang / Produktionsmittel). Das Verfahren bestimmt iterativ eine Menge zulässiger Aktionen für jeden Zustand unter Verwendung einer problemspezifischen Heuristik. Die erste Menge zulässiger Aktionen besteht aus den ersten Vorgängen aller Aufträge. Aus dieser Menge wird die Vorgang-Produktionsmittel-Kombination gewählt, die die kürzeste Prozesszeit aufweist (problemspezifische Heuristik). Der zweite Vorgang des Auftrags, dessen erster Vorgang in der ersten Iteration gewählt wurde, wird in der folgenden Iteration zur Menge der zulässigen Lösungen hinzugefügt. Dieses Verfahren wiederholt sich bis alle Aufträge eingeplant sind (Lu & Chiu u. a. 2016). Die resultierenden Mengen berücksichtigen die Reihenfolgeabhängigkeiten und limitieren die Anzahl möglicher Aktionen stark. SOMA ist nur anwendbar bei strikten Reihenfolgebeziehungen. In der *Rollout*-Phase wird *Path Random Search* (PRS) angewendet. Die Grundidee ist, jeden Vorgang entsprechend der Dauer zu gewichten, um Vorgänge mit kurzen Dauern mit einer höheren Wahrscheinlichkeit auszuwählen. In diesem Schritt kommt demnach dieselbe Heuristik wie bei SOMA zum Einsatz. In der Selektionsphase wird eine abgewandelte Form der UCT-Selektionsfunktion genutzt. Die Autoren bewerten die Güte eines Zustands anhand der Anzahl dominierter Zustände und der Anzahl nicht-dominierter Punkte in der bereits bekannten Pareto-Menge. Diese Selektionsfunktion ist als *Non-dominated Sorting UCT* (NSUCT) bekannt (Lu & Chiu u. a. 2016). Um die Komplexität des Problems beherrschbar zu machen, wenden die Autoren *Tree Pruning* auf allen Ebenen des Suchbaums an, indem die maximale Anzahl der Zustände limitiert wird. Die besten Zustände, bewertet anhand des NSUCT-Wertes, werden behalten (Lu & Chiu u. a. 2016).

Furuoka & Matsumoto (2017) nutzen einen MCTS-Ansatz ohne Modifikationen zur Evaluation von Prioritätsregeln im Kontext einer Produktion von Druckerzeugnissen. Die Druckaufträge durchlaufen mehrfach dieselben Druckmaschinen. Je nach Druckfarbe sind Rüstvorgänge nötig, die minimiert werden sollen. Übergeordnetes Ziel des vorgestellten Ansatzes ist es, neues Wissen über die Eignung von Prioritätsregeln zur Steuerung der betrachteten Produktion zu gewinnen. Hierfür werden drei Prioritätsregeln als *Rollout-Policy* (*heavy rollout*) implementiert. Zur Auswertung werden die drei MCTS-Ansätze sowie drei Steuerungsansätze, die nur jeweils eine der drei betrachteten Prioritätsregeln nutzen, anhand der Anzahl

Rüstvorgänge verglichen. Die Ergebnisse zeigen, dass alle drei untersuchten MCTS-Ansätze bessere Ergebnisse liefern als die Vergleichssteuerung. Hervorzuheben ist, dass die besten Ergebnisse nicht mit der am besten abschneidenden reinen Prioritätsregel erzielt wurden (Furuoka & Matsumoto 2017). Somit ist es nicht möglich, die MCTS-Steuerung einzusetzen, um gute Prioritätsregeln für die ausschließliche Nutzung abzuleiten. Die Ergebnisse verdeutlichen die Stärke von MCTS gegenüber reinen Prioritätsregeln, unter anderem begründet in der Evaluierung der Terminalzustände und dem zielgerichteten Baufortschritt.

Lubosch & Kunath u. a. (2018) stellen einen agnostischen, unikriteriellen (Makespan) MCTS-basierten Optimierer vor, den sie anhand eines Benchmark Flow-Shops aus der Halbleiterindustrie erproben. Die Autoren nutzen *Gradient Boosted Decision Trees* zur Wahl des besten Gewichtungsfaktors zwischen Exploration und *Exploitation* der UCT-Selektionsfunktion nach Kocsis & Szepesvári (2006). Die Simulationsexperimente unterschiedlich parametrisierter Selektionsfunktionen zeigen die Abhängigkeit der Ergebnisgüte von der Wahl der Selektionsfunktion und der zur Verfügung stehenden Rechenzeit. Einfache Prioritätsregeln schnitten bei großen Problemen und kurzer Rechenzeit besser ab als die aufwendigeren MCTS-basierten Ansätze (Lubosch & Kunath u. a. 2018). Ein möglicher Grund für das schlechtere Abschneiden von MCTS im Vergleich zu den Prioritätsregeln ist der Verzicht auf die Nutzung von Modifikationen zur zielgerichteten Auswahl von Aktionen und Zuständen sowie die fehlende Nutzung von Domänenwissen. Dies steht im Gegensatz zu den Empfehlungen anderer Autoren, beispielsweise Wu & Wu u. a. (2013).

Shang & Wu u. a. (2020) stellen einen übertragbaren unikriteriellen Ansatz aus einer anderen Domäne zur Berücksichtigung von Unsicherheit vor. Die Forschergruppe vergleicht die Leistungsfähigkeit eines genetischen Algorithmus mit einem MCTS ohne Modifikationen und einem MCTS, der die Stochastik der zugrunde liegenden Prozesse berücksichtigt anhand des Problems der Minimierung der Instandhaltungskosten bei der Wartung eines Stromnetzes. Um die Stochastik zu berücksichtigen, wird die *Rollout*-Phase angepasst. Der stochastische MCTS bewertet die anfallenden Kosten für Arbeitszeit, Material und Schäden durch Stromausfälle, abhängig von der normalverteilten elektrischen Leistung des jeweiligen Zustands. Hierfür wird eine zufällige Stichprobe der normalverteilten Zufallsvariable gezogen. Diese Stichprobe dient als Grundlage für die Szenarien zur Bewertung des Zustands (Shang & Wu u. a. 2020). Um das Optimierungsproblem weiter zu vereinfachen, werden die Restriktionen durch Strafterme in der Zielfunktion ersetzt. Die Höhe der Strafterme wird auf Basis von vereinfachten Berechnungen der Restriktionen gebildet. Die Ergebnisse zeigen, dass der stochastische MCTS bessere Ergebnisse erzielt als die beiden Alternativen. Zudem liefert das Verfahren durch die Berücksichtigung der zugrunde liegenden Wahrscheinlichkeitsverteilung ein stabileres und wiederholbareres Optimierungsergebnis Shang & Wu u. a. (2020). Das

Verfahren zeigt einen möglichen Ansatz, die Stochastik in die Bewertung einfließen zu lassen. In dem vorgestellten Verfahren werden jedoch die Eigenschaften der Normalverteilung als reproduzierende Verteilung genutzt, sodass das Vorgehen nicht auf andere Verteilungen uneingeschränkt übertragbar ist.

3.4 Forschungsdefizit

Exakte Verfahren garantieren Optimalität. Um effiziente Lösungsverfahren nutzen zu können, müssen Probleme jedoch in eine für das Verfahren geeignete Form gebracht werden. Eine entsprechende Modellierung ist schwer zu finden, sodass die Optimierung realitätsnaher Probleme und die Erweiterbarkeit nur schwierig möglich sind. Wegen langer Rechenzeiten spielen exakte Verfahren nur eine untergeordnete Rolle im reaktiven *Scheduling*.

Heuristiken sind weit verbreitet zur Lösung von *Scheduling*-Problemen. Problemspezifische Heuristiken werden aufgrund ihrer Einfachheit bei Steuerungsaufgaben mittlerer Komplexität angewandt. Die Auswahl der passenden Heuristik ist jedoch problem- und situationsabhängig, sodass eine hohe Lösungsgüte nur nach vorheriger eingehender Untersuchung möglich ist. Die Forschung hat außerdem gezeigt, dass einfache Heuristiken, die nur lokale Informationen nutzen, bei hoher Komplexität zunehmend schlechtere Ergebnisse erzielen. Der myopische Charakter der einfachen Entscheidungsregeln führt zu einer geringen Vorausschau, sodass Zielgrößen in komplexen Umgebungen nur schwer über mehrere Entscheidungsschritte optimiert werden können.

Im Bereich der Meta-Heuristiken gibt es zahlreiche Beispiele leistungsfähiger Steuerungsansätze auch für realitätsnahe Probleme. Einige Heuristiken wie *Simulated Annealing*, genetische Algorithmen oder der Partikel-Schwarm-Ansätze erfordern jedoch eine entsprechende Modellierung des Produktionsproblems. Die Modellierung analog zum zugrunde liegenden natürlichen Prozess, beispielsweise in Form von Chromosomen, erschwert die Generalisierung und Erweiterbarkeit des Verfahrens. Forschung im Bereich der hybriden genetischen Algorithmen belegt die gute Leistungsfähigkeit zur multikriteriellen Optimierung komplexer Probleme bei entsprechender Modellierung.

Im Bereich des Maschinellen Lernens existieren einige Ansätze zur Produktionssteuerung. Um gute Ergebnisse zu erzielen, ist jedoch vielfach eine entsprechende Trainingsphase nötig, die ein ausreichend genaues Simulationsmodell oder Realdaten erfordert. Die Forschung zeigt zwar eine gute Generalisierbarkeit der Ansätze bei kleineren Systemveränderungen, beim Übertrag auf neue Produktionssysteme ist jedoch ein erneutes Training nötig. Durch die realitätsnahe Modellierung eines Produktionssystems steigt der Zustandsraum deutlich

Tabelle 3.1: Bewertung der Scheduling-Ansätze aus dem Stand der Forschung anhand der Anforderungen der Matrixproduktion
 nicht erfüllt: ○, teils erfüllt: ◐, erfüllt: ●

	Abilden einer realitätsnahen Matrixproduktion	Vorausschauende Mehrzieloptimierung	Kurzfristige Reaktionsfähigkeit bei Abweichungen	Umgang mit ungenauen Vorgabezeiten	Erreichbarkeit eines hohen Lösungsniveaus	Generalisierbarkeit des Lösungsverfahrens
Exakte Verfahren						
(Özgüven & Özbakır u. a. 2010)	◐	○	○	○	●	◐
(Maravelias & Grossmann 2003)	◐	◐	○	○	●	◐
Problemspezifische Heuristiken						
(Scholz-Reiter & Görges u. a. 2009a)	◐	◐	●	○	◐	◐
(Scholz-Reiter & Freitag u. a. 2005)	◐	◐	●	○	◐	◐
(Mouelhi-Chibani & Pierreval 2010)	○	○	●	○	◐	◐
(Freitag & Hildebrandt 2016)	●	◐	●	○	◐	◐
Meta-Heuristiken						
(Xia & Wu 2005)	◐	●	○	○	●	◐
(Kacem & Hammadi u. a. 2002)	◐	●	◐	○	●	◐
(Li & Gao 2016)	◐	●	◐	○	●	◐
(Wang & Gao u. a. 2010)	◐	●	◐	○	●	◐
(Gao & Sun u. a. 2008)	◐	●	◐	○	●	◐
(Chiang & Lin 2012)	◐	●	◐	○	●	◐
(Chiang & Lin 2013)	◐	●	◐	○	●	◐
(Scholz-Reiter & Hamann 2008)	◐	○	◐	○	◐	◐
(Huang & Yang u. a. 2013)	◐	○	◐	○	◐	◐
Maschinelles Lernen						
(Stegherr 2000)	○	●	●	○	◐	◐
(Zhang & Zheng u. a. 2012)	○	●	●	○	◐	◐
(Kuhnle 2020)	◐	◐	●	○	◐	◐
(Waschneck & Reichstaller u. a. 2018)	◐	◐	●	○	◐	◐

Robustes Scheduling						
(Li & Ierapetritou 2008)	◐	◐	◐	◐	●	○
(Petkov & Maranas 1997)	○	◐	●	◐	●	○
(Bonfill & Espuña u. a. 2005)	◐	◐	◐	◐	●	◐
(Policella & Cesta u. a. 2008)	●	●	◐	◐	◐	●
(Jiménez 2012)	●	◐	◐	◐	◐	●
(Jamili 2016)	◐	◐	◐	◐	◐	●
(Echsler 2020)	●	◐	●	◐	●	◐
(Bifet & Gavalda 2007)	○	○	○	●	○	●
(Ross & Adams u. a. 2012)	○	○	○	●	○	◐
Monte Carlo Tree Search						
(Chaslot & De Jong u. a. 2006)	○	◐	◐	○	◐	●
(Runarsson & Schoenauer u. a. 2012)	◐	◐	◐	○	◐	●
(Loth & Sebag u. a. 2013)	◐	◐	●	○	◐	◐
(Wu & Wu u. a. 2013)	◐	●	●	○	●	◐
(Chou & Liang u. a. 2015)	◐	●	●	○	●	◐
(Lu & Chiu u. a. 2016)	◐	●	●	○	●	●
(Furuoka & Matsumoto 2017)	◐	◐	◐	○	●	●
(Lubosch & Kunath u. a. 2018)	◐	◐	◐	○	◐	●
(Shang & Wu u. a. 2020)	○	◐	◐	◐	●	◐

an, wodurch die Trainingsphase stark verlängert werden muss. Verbreitete Ansätze, wie das *Reinforcement Learning* erfordern zudem, dass der Zustandsraum eine feste Größe besitzt. Die untersuchten Ansätze zeigen nur eingeschränkte Fähigkeiten im Bereich der Mehrzieloptimierung. Das trainierte Modell kann jedoch schnell eine Entscheidung generieren, sodass dieser Ansatz Stärken im Bereich der echtzeitnahen Steuerung aufweist.

Die vorgestellten Forschungsarbeiten, die MCTS nutzen, zeigen vielversprechende Ergebnisse für FJSP. Da MCTS den Zustandsraum nicht vollständig aufbaut und dieser nicht vollständig modelliert werden muss, ist die Erweiterbarkeit gut möglich. Die Mehrzieloptimierung auch mit Hilfe von Pareto-Mengen ist effizient abbildbar. Einige hybride Ansätze erzielen herausragende Ergebnisse auf Benchmark-Problemen, allerdings unter Nutzung von Domänenwissen oft in Form spezifischer Prioritätsregeln. Ist dies der Fall, verringert sich die Generalisierbarkeit durch die Spezialisierung auf die Eigenschaften des betrachteten Problems. Die Forschungsergebnisse zeigen übereinstimmend, dass Modifikationen des Grundalgorithmus nötig sind, um sehr gute Resultate zu erzielen. Die Effizienz des Verfahrens ermöglicht ein reaktives *Scheduling*, um kurzfristig auf Abweichungen zu reagieren.

Die vorgestellten Ansätze benötigen Informationen über die Dauern der Vorgänge. Diese Informationen werden bei den betrachteten Steuerungsaufgaben als gegeben angenommen. Die Forschungsarbeiten der robusten Optimierung nutzen teilweise zusätzlich Verteilungsinformationen, um durch gezieltes Einbringen von Schlupfzeiten die Robustheit zu erhöhen. Dieses Vorgehen erfordert zudem Informationen über die zugrunde liegenden Verteilungen. Die Betrachtung der Arbeiten aus dem Bereich der *Concept Drift*-Erkennung zeigt, dass durch Analyse der Rückmeldedaten Verteilungsänderungen detektiert und Mittelwerte zutreffend geschätzt werden können.

Die Betrachtung des Stands der Forschung hat gezeigt, dass sich MCTS-basierte Ansätze zur Modellierung einer Matrixproduktion besonders eignen. Für das MCTS-Verfahren muss das Steuerungsproblem lediglich als sequentielles Entscheidungsproblem modelliert werden. Eine aufwendige Modellierung angelehnt an natürliche Prozesse, wie sie bei vielen Meta-Heuristiken gefordert ist, wird nicht benötigt. Dies erleichtert das Abbilden einer realitätsnahen Matrixproduktion. Aufgrund der Evaluation von vollständigen Belegungsplänen, repräsentiert durch den erreichten Terminalzustand am Ende der *Rollout*-Phase, ermittelt MCTS die Auswirkungen der einzelnen Entscheidungen über die Entscheidungssequenz und steuert somit vorausschauend. Dennoch fokussieren sich die betrachteten Forschungsansätze auf die Lösung von Benchmark-Problemen, die reale *Scheduling*-Aufgaben nur abstrahiert abbilden. Rüst- und Transportzeiten werden meist nicht berücksichtigt. Auch finden alternative Vorgangsreihenfolgen keine Berücksichtigung. Ein weiteres Defizit liegt in der Reduktion der Betrachtungsweise auf den initialen Plan. Daraus ergibt sich die erste Forschungsfrage:

Forschungsfrage F1 Wie kann eine reaktive und vorausschauende Mehrziel-Steuerung für die Matrixproduktion unter Berücksichtigung realer Restriktionen gestaltet werden?

Für eine reaktive Steuerung ist vor allem das schnelle Erreichen eines hohen Lösungsniveaus nötig, um im Falle von Abweichungen schnell Plananpassungen vornehmen zu können.

Die betrachteten Ansätze erzielen sehr gute Ergebnisse auf Benchmark-Problemen unter Nutzung von problemspezifischem Domänenwissen. Dieses Wissen wird meist in Form von angepassten Prioritätsregeln in der Expansions- und in der *Rollout*-Phase genutzt. Durch diese Spezialisierung kann das entsprechende Steuerungsproblem effizienter gelöst werden. Gleichzeitig sinkt die Leistungsfähigkeit der Steuerung bei anderen Anwendungsfällen und anderen Optimierungszielen, da die Auswahl geeigneter Prioritätsregeln problem-, ziel- und situationsspezifisch ist. Zwar benennen viele Forschende die Abhängigkeit der Lösungsgüte von der Parametrisierung und Spezialisierung der Verfahren, allerdings fehlen Ansätze, um diese Abhängigkeit effektiv zu reduzieren. Es ist offen, wie ein System ausgestaltet werden kann, das die Brücke zwischen hoher Lösungsgüte durch Nutzung von Domänenwissen

und guter Generalisierbarkeit auf andere Steuerungsprobleme schlägt. Daraus resultiert die zweite Forschungsfrage:

Forschungsfrage F2 Kann durch automatisierte Anpassung an das vorliegende Steuerungsproblem eine hohe Lösungsgüte bei unterschiedlichen Steuerungsproblemen und Optimierungszielen erreicht werden?

Zentrale Eingangsgrößen der Produktionssteuerung, die zudem Änderungen unterliegen, sind die Dauern der Vorgänge. Die Betrachtung des Stands der Forschung hat gezeigt, dass die Vorgabezeiten der Vorgänge beziehungsweise die zugrunde liegenden Verteilungen meist als gegeben angenommen werden. Da das Vorliegen von aktuellen und realitätsnahen Daten eine Grundvoraussetzung einer echtzeitnahen, realisierbaren Produktionssteuerung ist, widmet sich die dritte Forschungsfrage der Anpassung der Vorgabezeiten:

Forschungsfrage F3 Wie können die Dauern der Vorgänge in der Produktion fortlaufend überwacht, Verteilungsänderungen detektiert und realistische Dauern abgeschätzt werden?

Um die Forschungslücke zu schließen und die drei Forschungsfragen zu beantworten, wird im folgenden Kapitel eine hybride Produktionssteuerung für die Mehrzieloptimierung einer Matrixproduktion entwickelt. Diese basiert auf MCTS und lokaler Suche als Post-Optimierung. Zudem wird eine Methodik zur Verteilungsschätzung und Detektion von Verteilungsänderungen der zugrunde liegenden Prozesszeiten vorgestellt. Daraus ergibt sich eine reaktive Steuerung, die bei Abweichungen fortlaufend steuernd in die Produktion eingreift.

4 Methode zur Steuerung einer Matrixproduktion

Das nachfolgende Kapitel stellt den entwickelten Mehrzieloptimierer basierend auf Monte Carlo Tree Search vor. Zusätzlich wird ein Ansatz zur Hybridisierung des MO-MCTS mit lokaler Suche erläutert, der zur Post-Optimierung vielversprechender Belegungspläne angewendet werden kann. Ausgangspunkt bildet eine Methode zur Ableitung realistischer Planzeiten als Eingangsgröße für die Produktionssteuerung. Die entwickelte Steuerung ist in der Lage, selbstständig realistische Zeiten zu ermitteln, unter Berücksichtigung der Restriktionen einer Matrixproduktion, schnell Steuerungsentscheidungen hoher Güte zu treffen und reaktiv auf Änderungen in der Produktion zu reagieren.

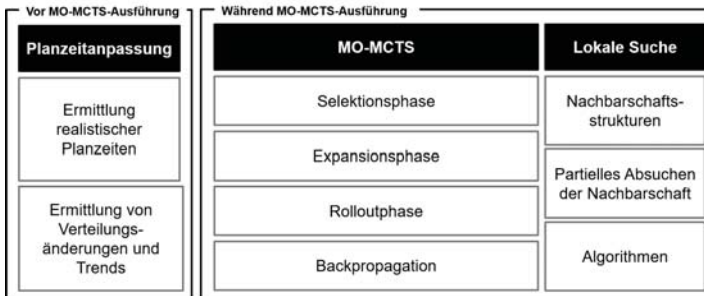


Abbildung 4.1: Gesamtsystem aus Planzeitanpassung, Basisoptimierungsverfahren MCTS und lokaler Suche als Post-Optimierungsschritt

Abbildung 4.1 stellt das Gesamtsystem bestehend aus der Methode zur Planzeitanpassung, dem Basisoptimierungsverfahren MCTS und dem Post-Optimierungsverfahren mit lokaler Suche vor. Zu Beginn dieses Kapitels werden das Steuerungsproblem, die Steuerungsaufgabe und die Optimierungsziele definiert. Anschließend wird die entwickelte Steuerungsmethode anhand der Bereiche aus Abbildung 4.1 vorgestellt.

4.1 Problemeingrenzung, Steuerungsaufgabe und Zieldefinition

Im Folgenden wird das Steuerungsproblem anhand der Systematik aus Kapitel 2.2.2 eingegrenzt. Anschließend werden die Steuerungsaufgabe benannt und die Zielgrößen definiert.

4.1.1 Problemeingrenzung

Nachfolgend wird die Matrixproduktion anhand des Produktionsprogramms ($\beta_1 = \emptyset$), den Produktionsmitteln, den Transportmitteln, der Logistik und den Mitarbeitern charakterisiert.

Forschungsgegenstand ist die Matrixproduktion eines Auftragsfertigers mit Losgröße 1. Für jedes zu fertigende Produkt wird die mögliche Abfolge der einzelnen **Bearbeitungsvorgänge** durch den Vorranggraph definiert. Bearbeitungsvorgänge laufen sequentiell ab. Durch den Vorranggraph wird die Abfolge der Vorgänge ohne Zuordnung zu Produktionsmitteln abgebildet. Die Gesamtzahl der Vorgänge pro Produktionsauftrag unterliegt keinen Einschränkungen ($\beta_4 = \emptyset$). Jeder **Produktionsauftrag** umfasst genau ein Produkt und ist mit einem frühestmöglichen Freigabezeitpunkt ($\beta_2 = A_k^*$) und spätestmöglichem Endzeitpunkt versehen. Jeder Produktionsauftrag hat einen Fertigstellungstermin ($\beta_3 = F_k^*$). Verspätungen sind möglich. Zwischen den einzelnen Produktionsaufträgen bestehen keine Reihenfolgebeziehungen ($\beta_7 = \emptyset$).

Die Matrixproduktion besteht aus lose verketteten **Produktionsmitteln**. Jedes Produktionsmittel bietet mindestens einen Bearbeitungsvorgang an ($\alpha = FJ$). Die Pufferplätze vor und nach den Produktionsmitteln sind unbeschränkt. Die Bearbeitungszeiten zur Ausführung eines Vorgangs sind produkt- und produktionsmittelabhängig ($\beta_6 = p_{km}$). Zwischen der Ausführung unterschiedlicher Vorgänge kann ein Rüstvorgang benötigt werden. Die Dauern der Rüstvorgänge zwischen den Vorgängen werden durch eine Rüstzeitmatrix festgelegt ($\beta_8 = rz_{k_1k_2}$). Die Verfügbarkeit der Produktionsmittel wird durch stochastische Ausfälle gemindert ($\beta_9 = breakdown$). Die gängige Vereinfachung, dass Ausfälle nur zwischen den Bearbeitungen auftreten ($\beta_5 = \emptyset$), wird übernommen.

Der **Transport** zwischen den Produktionsmitteln wird durch ein Transportsystem mit unbeschränkter Kapazität sichergestellt. Diese Annahme wirkt nicht einschränkend, sofern das Transportsystem eine ausreichende Kapazität vorhält. Die Transportzeiten zwischen allen Produktionsmitteln werden durch eine Transportmatrix definiert. **Mitarbeiter** werden bei der Planung nicht berücksichtigt ($\beta_{10} = \emptyset$).

4.1.2 Steuerungsaufgabe

Die Produktionssteuerung legt fest, welcher Vorgang in welchem Zeitintervall auf welchem Produktionsmittel durchgeführt werden soll. Ebenso werden für die Transportvorgänge Ausgangsstandort, Ziel, Transportbeginn und -ende bestimmt. Sind Rüstvorgänge nötig, werden der Rüstzielzustand sowie Beginn und Ende geplant. Die Produktionssteuerung definiert somit implizit die Vorgangsreihenfolge innerhalb der technisch möglichen Grenzen, gegeben durch den Vorranggraph.

Die Bereitstellung von Komponenten, Werkzeugen und Verbrauchsmaterialien ist nicht Teil der Steuerungsaufgabe. Zur Sicherstellung der Material- und Werkzeugverfügbarkeit wird jedoch ein zeitlicher Vorlauf bei der Umplanung eingehalten, innerhalb dessen nur Verschiebungen bereits eingeplanter Vorgänge möglich sind.

4.1.3 Steuerungsziele

Ziel der Steuerung ist die Optimierung kapazitätsbezogener und auftragsbezogener Zielgrößen. Der **Makespan** beschreibt die Dauer vom Beginn des ersten Bearbeitungsvorgangs des frühesten Produktionsauftrags $\min(v_{ts})$ bis zur Fertigstellung des letzten Bearbeitungsvorgangs des letzten Produktionsauftrags $\max(v_{te})$ aller betrachteten Vorgänge, siehe Formel 4.1.1. Damit sind alle Rüst- und Transportvorgänge inkludiert.

$$t_{mks} = \max(v_{te}) - \min(v_{ts}), \forall v \in V \quad 4.1.1$$

Die **Gesamtarbeitsmenge** bezeichnet die Gesamtdauer aller Bearbeitungsvorgänge über alle Produktionsmittel, siehe Formel 4.1.2. Durch die Wahl von Produktionsmitteln mit kürzeren Bearbeitungsdauern für bestimmte Vorgänge lässt sich diese Zielgröße minimieren.

$$t_w = \sum_{v \in V} (v_{te} - v_{ts}) \quad 4.1.2$$

Die **maximale Arbeitsbelastung**, siehe Formel 4.1.3, repräsentiert die Gesamtsumme der Bearbeitungszeiten des am stärksten belasteten Produktionsmittels. Die Minimierung dieser Zielgröße führt zu einer gleichmäßigen Auslastung der Produktionsmittel.

$$t_{mrw} = \max_{m \in M} \left(\sum_{v \in V} (v_{te} - v_{ts}) \right) \quad 4.1.3$$

Die Zielgröße der **durchschnittlichen Auslastung** errechnet sich als Quotient der Summe der Bearbeitungs- und Rüstzeiten eines Produktionsmittels und des Makespans, siehe Formel 4.1.4. Die Auslastung des Produktionssystems soll maximiert werden.

$$t_u = \frac{1}{|M|} \sum_{m \in M} \frac{\sum_{v \in V_m} (v_{t_e} - v_{t_s})}{t_{mks}} \quad 4.1.4$$

Die **Durchlaufzeit** bemisst sich als Zeitspanne des Beginns des ersten Vorgangs bis zum Abschluss des letzten Vorgangs eines Produktionsauftrags $p \in P$. Zur Optimierung der Durchlaufzeiten wird die Summe aller Einzeldurchlaufzeiten genutzt. Diese ergibt sich nach Formel 4.1.5. Kurze Durchlaufzeiten ermöglichen schnelle Lieferungen.

$$t_{ct} = \sum_{p \in P} (\max_{v \in V_p} (v_{t_e}) - \min_{v \in V_p} (v_{t_s})) \quad 4.1.5$$

Die **Verspätung** ist eine weitere auftragsbezogene Zielgröße und umfasst die Zeitspanne ausgehend vom letztmöglichen Fertigstellungszeitpunkt eines Produktionsauftrags $t_{lft,p}$ bis zur tatsächlichen Fertigstellung des letzten Vorgangs v_{t_e} . Zur Optimierung der Gesamtverspätung wird die Summe aller Verspätungen herangezogen. Verfrühte Lieferungen können Verspätungen nicht kompensieren, siehe Formel 4.1.6.

$$t_{ct} = \sum_{p \in P} (\max(\max_{v \in V_p} (v_{t_e} - t_{lft,p}), 0)) \quad 4.1.6$$

4.1.4 Normierung der Zielgrößen

Um die unterschiedlichen Zielgrößen vergleichbar zu machen, werden sie auf den Wertebereich von $[0,1]$ normiert. Zwei Teilaspekte der Normierung wirken sich auf das Optimierungsergebnis aus. Zum einen kann die Zuordnung der Zielfunktionswerte auf den Wertebereich durch unterschiedliche Funktionen erfolgen. Zum anderen muss festgelegt werden, welcher Zielfunktionswert dem höchsten Wert $r \in [0, 1]$ zugeordnet wird. Die Festlegung der oberen und unteren Schranke der Zielgrößen kann nicht immer exakt bestimmt werden. Da es teils a priori nicht möglich ist, das bestmögliche Ergebnis zu bestimmen. Im letzteren Fall besteht die Möglichkeit, die besten jemals beobachteten Ergebnisse als obere Schranke zu nutzen. Nachteil dieses Verfahrens ist jedoch, dass implizit die Vergleichbarkeit der Steuerungsentscheidungen vorausgesetzt wird. Daher wird von diesem Ansatz Abstand genommen und

stattdessen die obere und untere Schranke rechnerisch oder auf Basis von Domänenwissen bestimmt. Im Folgenden wird auf die jeweiligen Zielgrößen eingegangen.

Die untere Schranke des **Makespans** unter Berücksichtigung der Restriktionen ist rechnerisch nicht in kurzer Zeit exakt bestimmbar. Die theoretische untere Schranke ergibt sich für eine Gleichverteilung der minimalen Gesamtarbeitsmenge auf alle Stationen. Diesem Wert wird im Folgenden der Wert 1 zugeordnet. Als schlechtester Wert wird die Allokation aller Vorgänge auf einem Produktionsmittel angenommen.

Die optimale **maximale Arbeitsbelastung** verhält sich analog. Der Bestwert ist dann erreicht, wenn alle Produktionsmittel gleich beansprucht werden.

Die bestmögliche Lösung für die **Gesamtarbeitsmenge** ergibt sich aus der Summe aller erforderlichen Produktionsvorgänge zur Fertigstellung der Produktionsaufträge jeweils ausgeführt durch die Produktionsmittel, die für diese Vorgänge über die kürzesten Bearbeitungszeiten verfügen. Die schlechteste Gesamtarbeitsmenge ist gegeben durch die Wahl des langsamsten Produktionsmittels.

Der beste erreichbare Wert für die **Gesamtdurchlaufzeit** errechnet sich als Summe der minimalen Bearbeitungszeiten bei der optimalen Wahl der Produktionsmittel für jeden Vorgang. Dieser Wert impliziert, dass alle Schritte ohne Rüst- und Transportvorgänge auskommen. Auch dieser Wert ist unter Umständen nicht erreichbar. Als schlechteste Lösung wird die Allokation aller Vorgänge auf einem Produktionsmittel gewertet.

Die untere Schranke für die **minimale Verspätung** beträgt null Minuten. Da theoretisch die Verspätung unbeschränkt ist, wird eine obere Schranke angenommen, die sich aus der maximal akzeptablen Verspätung ergibt. Dieser Wert ist problemspezifisch. Akzeptiert ein Kunde maximal Lieferungen mit zehn Tagen Verspätung, würde dieser Wert als obere Schranke verwendet werden.

Um unterschiedliche Bewertungen der Zielgrößen abzubilden, werden verschiedene Funktionen für die Zuordnung der absoluten Zielwerte auf das Intervall $[0, 1]$ verwendet. Bei einer linearen Funktionen führen relative Verbesserungen stets zum gleichen Zuwachs. Wenn Verbesserungen im Grenzbereich ein besonders hoher Wert zugesprochen werden soll, empfiehlt sich die Verwendung einer exponentiellen Funktion. S-Kurven, beispielsweise die Sigmoid-Funktion oder die inverse Sigmoid-Funktion, können komplexe Präferenzen abbilden. Als S-Funktion modelliert, führt eine leichte Verspätung nur zu einem geringen Abfall des normierten Zielfunktionswerts. Ist die Verspätung bereits sehr groß, führt eine weitere Verspätung nur noch zu einer geringen Reduktion des normierten Zielfunktionswerts. Dadurch ist der Optimierungsdruck in den Randbereichen gering. Binäre Funktionen können genutzt

werden, um harte Zielgrößen abzubilden, beispielsweise strikte Liefertermine. Sofern nicht anders angegeben, wird eine lineare Funktion verwendet.

4.2 Ableitung realistischer Planzeiten aus Beobachtungen

Die Qualität der Steuerungsentscheidung hängt nicht zuletzt von der Güte der Eingangsgrößen ab. Die Dauern der Rüst-, Bearbeitungs- und Transportvorgänge sind dafür entscheidend. Daher werden Verteilungsinformationen zu den Rüst-, Transport- und Bearbeitungszeiten benötigt. Zudem sollten Veränderungen in den Verteilungen erkannt werden. Diese können sowohl graduell, periodisch als auch sprunghaft erfolgen. Mögliche Gründe hierfür sind beispielsweise Optimierungsmaßnahmen (sprunghaft), Verschleiß (graduell) oder Unterschiede zwischen Wochentagen (periodisch) (Gama & Žilobaitundė u. a. 2014).

Die Verteilungsinformationen können einerseits bei der Planung genutzt werden, um robuste Pläne zu erzeugen, andererseits helfen sie bei der Entscheidung, ob bei Abweichungen von der Planzeit eine Neuplanung erforderlich ist. Die vorgestellte Methode basiert auf der vom Autor angeleiteten Abschlussarbeit von (A_Wolf 2019). Abbildung 4.2 stellt das Gesamtsystem zur Ermittlung und Anpassung der Planzeiten dar.

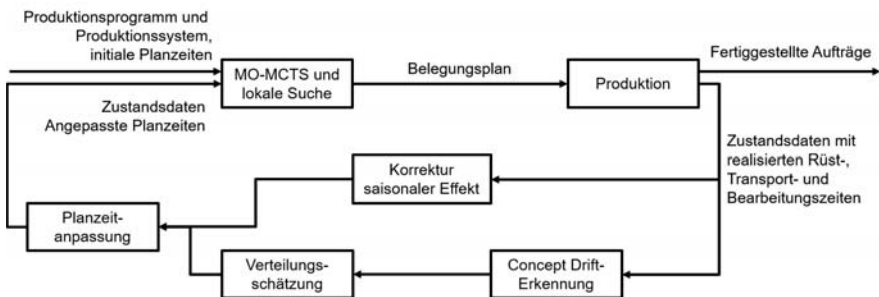


Abbildung 4.2: Schematische Darstellung des Aufbaus und der Anpassung der Planzeiten, angelehnt an einen Regelkreis nach (A_Wolf 2019)

Eingangsgröße des Systems sind Initialisierungsdaten mit allen relevanten Informationen zu den Produktionsmitteln und deren angebotenen Produktionsprozessen, dem Produktionsprogramm, inklusive der Vorranggraphen sowie zu den vorliegenden Produktionsaufträgen. Zudem werden initiale Planzeiten für die Rüst-, Transport- und Bearbeitungszeiten bereitgestellt. Diese können den Vorgabezeiten aus der Planungsphase entsprechen. Auf Basis der Eingangsinformationen ermittelt die Steuerung einen Belegungsplan, der zur Ausführung an

die Produktion übergeben wird. Die Produktion liefert fortwährend Zustandsdaten mit den realisierten Transport-, Rüst- und Bearbeitungszeiten an die Steuerung zurück. Die realisierten Zeiten werden einerseits genutzt, um eine Korrektur saisonaler Effekte anhand einer Zeitreihenanalyse vorzunehmen. Andererseits wird in einer Verteilungsanalyse die zugrundeliegende theoretische oder empirische Verteilung ermittelt. Vorgelagert findet mit Hilfe einer *Concept Drift*-Erkennung eine Untersuchung statt, ob die hinzugekommenen Beobachtungen weiterhin zur derzeit angenommenen Verteilung passen oder ob eine Verteilungsänderung vorliegt. Wird eine Verteilungsänderung detektiert, wird die Datenbasis so gewählt, dass die nachfolgende Verteilungsanalyse auf den Beobachtungen erfolgt, die zu der neuen stationären Verteilung gehören. Das Ergebnis dieser Schritte führt zu einer Planzeitanpassung aufgrund der Verteilungsinformationen. Die von der Produktion übermittelten Zustandsdaten und die angepassten Planzeiten stellen die Eingangsgrößen für die Produktionssteuerung dar. Im Folgenden werden die einzelnen Schritte des Gesamtsystems vorgestellt.

4.2.1 Concept-Drift-Erkennung zur Detektion von Verteilungsänderungen

Bei der *Concept-Drift*-Erkennung geht es darum, Verteilungsänderungen zu detektieren. Im Zusammenhang mit dem *Scheduling* ist das Ziel, die Grundgesamtheit der Beobachtungen so anzupassen, dass nur die Beobachtungen aus der aktuell gültigen Verteilung für die Parameterschätzung genutzt werden, siehe Abbildung 4.3.



Abbildung 4.3: *Concept-Drift-Erkennung*

Im Folgenden werden Methoden zur *Concept Drift*-Erkennung basierend auf dem Verteilungsvergleich unterschiedlicher Datenfenster sowie Ansätzen zur Detektion der Änderung anhand von Kenngrößen vorgestellt.

4.2.1.1 Änderungsdetektion durch Verteilungsvergleich

Als besonders vielversprechender Vertreter der Ansätze, die auf rollierenden Fenstern beruhen, wird das ADWIN-Verfahren nach Bifet & Gavaldà (2006) und Bifet & Gavaldà (2007) betrachtet, siehe Abschnitt 3.2.4.

ADWIN betrachtet alle möglichen benachbarten Teilfenster und verkleinert das Datenfenster, sobald eine Änderung detektiert wurde. Eingangsgrößen des Algorithmus sind das Konfidenzniveau $\delta \in (0, 1)$ und die Reihe an Beobachtungen x_1, x_2, \dots, x_t aus unabhängigen Verteilungen D_1, D_2, \dots, D_t . μ_t bezeichnet den unbekanntem Erwartungswert der ebenfalls unbekanntem Verteilung D_t . ADWIN baut ein Datenfenster W der letzten w Datenpunkte x_i auf. Der beobachtete Mittelwert des Datenfensters W wird als $\hat{\mu}_w$ bezeichnet. Das Datenfenster W wird in zwei Teilfenster W_0 und W_1 , der Länge w_0 und w_1 mit den Mittelwerten $\hat{\mu}_0$ und $\hat{\mu}_1$ geteilt, sodass $w_0 + w_1 = w$ gilt. Kommt ein neuer Datenpunkt x_t hinzu, wird er dem Fenster W angefügt. Anschließend wird für die Mittelwerte $\hat{\mu}_0$ und $\hat{\mu}_1$ der Teilfenster aus $\{x_1\}$ und $\{x_2, \dots, x_t\}$ geprüft, ob $|\hat{\mu}_0 - \hat{\mu}_1| \geq \epsilon_{cut}$ gilt. Ist dies nicht der Fall, wird das nachfolgende Teilfenster aus $\{x_1, x_2\}$ und $\{x_3, \dots, x_t\}$ geprüft. Sobald die Bedingung erfüllt ist, werden die Datenpunkte des Fensters w_0 gelöscht.

$$m_{ham} = \frac{1}{\frac{1}{w_0} + \frac{1}{w_1}} \quad 4.2.1$$

$$\delta' = \frac{\delta}{w} \quad 4.2.2$$

Zur Berechnung des Grenzwertes ϵ_{cut} wird das harmonische Mittel m_{ham} , siehe Formel 4.2.1, und das relative Konfidenzniveau, siehe Formel 4.2.2, genutzt.

$$\epsilon_{cut} = \sqrt{\frac{2}{m_{ham}} \sigma_w^2 \ln \frac{2}{\delta'} + \frac{2}{3m_{ham}} \ln \frac{2}{\delta'}} \quad 4.2.3$$

Der statistische Test überprüft, ob die Differenz der beobachteten Mittelwerte größer ist als ϵ_{cut} , siehe Formel 4.2.3.

4.2.1.2 Änderungsdetektion durch Kennzahlenvergleich

Analog zur statistischen Prozesskontrolle werden Kennwerte, wie der Mittelwert oder die Standardabweichung, in einem Diagramm abgetragen. Wird die vorab definierte Warn- oder Alarmgrenze über- oder unterschritten, sollte die Verteilung angepasst werden. Als Vertreter dieser Ansätze wird die EWMA-Qualitätsregelkarte untersucht.

Die EWMA-Qualitätsregelkarte gewichtet die neuesten Beobachtungen stärker. Zur Berechnung der EWMA-Karte wird ein Schätzer abhängig vom Gewichtungsfaktor λ für den Mittelwert errechnet, siehe Formel 4.2.4.

$$Z_t = (1 - \lambda)Z_{t-1} + \lambda X_t, \quad Z_0 = \mu_0 \quad 4.2.4$$

Die oberen und unteren Eingriffsgrenzen EG ergeben sich abhängig vom Parameter L , siehe Formel 4.2.5.

$$EG = \mu_0 \pm L\sigma \sqrt{\frac{\lambda}{2 - \lambda}(1 - (1 - \lambda)^{2t})} \quad 4.2.5$$

4.2.2 Verteilungsschätzung

Abbildung 4.4 fasst die einzelnen Schritte der Verteilungsanalyse zusammen, die beim Eintreffen eines neuen, realisierten Wertes durchgeführt wird. Der erste Schritt ist die Überprüfung der Anwendbarkeit der Verteilungsanalyse. Hier wird geprüft, ob die Stichprobe für eine Verteilungsschätzung ausreicht. Ist dies nicht erfüllt, werden die Planzeiten verwendet. Sind die Voraussetzungen erfüllt, werden in der Ausreißererkenkung singuläre Ausreißerwerte aus der Stichprobe entfernt. Die bereinigte Stichprobe dient als Grundlage der Histogrammerstellung. Das Histogramm wird genutzt, um die empirische Verteilung abzuleiten sowie die theoretische Verteilung in drei Schritten zu schätzen. Erster Teilschritt der Verteilungsschätzung ist die Auswahl der Verteilungsfamilie. Anschließend werden die Parameter der Verteilung geschätzt. Den letzten Teilschritt bilden Hypothesentests zur Überprüfung, ob die geschätzte Verteilung ausreichend gut zu den Beobachtungen passt. Trifft dies nicht zu, wird auf die empirische Verteilung zurückgegriffen. Genügt die Schätzung den Kriterien, werden die Planzeiten entsprechend der Verteilung aktualisiert.

Überprüfung der Anwendbarkeit der Verteilungsanalyse Die Größe der Stichprobe beeinflusst das erreichbare Signifikanzniveau der Hypothesentests und die Ausreißererkenkung über Quantile. Es existieren unterschiedliche Empfehlungen für die Stichprobengröße. Im Weiteren wird $n = 30$ als Richtwert verwendet (Kar & Ramalingam 2013).

Ausreißererkenkung Ausreißer sind Beobachtungen, die stark von den anderen Beobachtungen der Stichprobe abweichen (Cramer & Kamps u. a. 2008). Gründe hierfür können

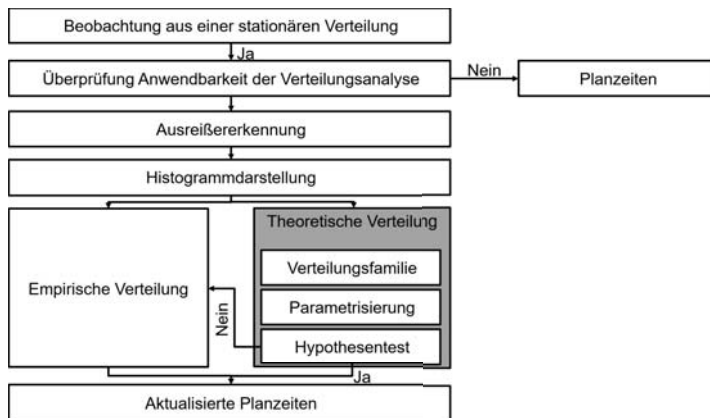


Abbildung 4.4: Schematische Darstellung des Aufbaus der Verteilungsanalyse

beispielsweise Fehler bei der Dokumentation von Vorgängen sein oder fehlerhafte Erfassungsgaräte. Die Bereinigung der Stichprobe erfolgt durch Abschneiden der Verteilungsränder. Hierfür wird nur das 99%-Quantil betrachtet.

Histogrammerstellung Aus den verbliebenen Daten der Stichprobe wird ein Histogramm erstellt. Bei der Einteilung in Klassen gibt es vielfältige Ansätze. Im Weiteren wird die Regel von Sturges (1926), siehe Formel 4.2.6, als Anhaltspunkt für die Klassenzahl g genutzt.

$$g = 1 + 3,3 \cdot \log(n) \quad 4.2.6$$

Empirische Verteilung Die empirische Verteilungsfunktion wird als kumulierte Häufigkeitsverteilung gemäß Formel 4.2.7 berechnet, wobei $h(a_i)$ die Anzahl der Ausprägung a_i darstellt. Die empirische Verteilungsfunktion, Formel 4.2.8, setzt die kumulierte Häufigkeitsverteilung mit der Gesamtzahl n ins Verhältnis (Fahrmeir & Heumann u. a. 2016).

$$H(x) = h(a_1) + \dots + h(a_j) = \sum_{i: a_i \leq x} h_i \quad 4.2.7$$

$$F(x) = \frac{H(x)}{n} \quad 4.2.8$$

Theoretische Verteilung Der erste Schritt der Ermittlung der theoretischen Verteilung ist die Auswahl der **Verteilungsfamilie**. Hierfür sind die Normalverteilung, die Gammaverteil-

lung, die logarithmische Normalverteilung, die Weibullverteilung, die Betaverteilung und die Gleichverteilung denkbar. In die Auswahl der Verteilung kann Domänenwissen über den zugrundeliegenden Prozess einfließen. Liegt kein Wissen vor, kann die Verteilung mit der besten Passung gewählt werden.

Die **Parameterschätzung** erfolgt mit Hilfe des Maximum-Likelihood-Schätzers, kurz ML-Schätzer. Das Vorgehen zur Ermittlung des ML-Schätzers ist für unterschiedliche Verteilungen gleich. ML-Schätzer sind mindestens asymptotisch erwartungstreu ¹, konsistent ² sowie mindestens asymptotisch effizient ³. Sie gelten daher als bester asymptotisch normaler Schätzer. (Hedderich & Sachs 2016)

Das Konzept des ML-Schätzers basiert auf der Idee, den Wert des Verteilungsparameters zu wählen, für den es am wahrscheinlichsten ist, die tatsächlichen Beobachtungen x_1, \dots, x_n zu machen. Der beste Schätzparameter $\hat{\Theta}$ der Verteilung ist der Wert mit der höchsten Wahrscheinlichkeit, diese Beobachtungen zu machen, siehe Formel 4.2.9 und 4.2.10.

$$L(\hat{\Theta}_{ML}) = \max_{\Theta_{ML}} L(\Theta_{ML}) \quad 4.2.9$$

$$f(x_1, \dots, x_n | \Theta_{ML}) = \max_{\Theta_{ML}} f(x_1, \dots, x_n | \Theta_{ML}) \quad 4.2.10$$

Zur einfacheren Bestimmung des Maximums der Funktion wird häufig die streng monotone Transformation der Logarithmierung angewandt. Danach ergibt sich die Darstellung gemäß Formel 4.2.11. (Hedderich & Sachs 2016)

$$\ln L(\Theta_{ML}) = \sum_{i=1}^n \ln f(x_i | \Theta_{ML}) \quad 4.2.11$$

Hypothesentest Der letzte Schritt in der Bestimmung der theoretischen Verteilung ist die Durchführung eines Hypothesentests zur Bewertung, wie gut die geschätzte, theoretische Verteilung zu den Beobachtungen passt. Es werden zwei Hypothesentests verwendet, die über unterschiedliche Eigenschaften verfügen.

¹Erwartungstreu bedeutet, dass der Erwartungswert des Schätzers dem wahren Wert entspricht. Bei der asymptotischen Erwartungstreue gilt dies für größer werdende Stichprobenumfänge (Fahrmeir & Heumann u. a. 2016)

²Konsistenz bedeutet, dass die mittlere quadratische Abweichung des Schätzers vom wahren Wert gegen null konvergiert. Dies ist durch die asymptotische Erwartungstreue gegeben. (Fahrmeir & Heumann u. a. 2016)

³Ein erwartungstreuer Schätzer wird als effizient bezeichnet, wenn die Varianz für alle zugelassenen Verteilungen den kleinstmöglichen Wert annimmt (Fahrmeir & Heumann u. a. 2016).

Die Nullhypothese H_0 lautet, dass die Stichprobe x_1, \dots, x_n einer Grundgesamtheit entspringt, deren Verteilung $F_0(x)$ mit der geschätzten Verteilung $F(x)$ übereinstimmt, siehe Formel 4.2.12.

$$H_0 : F(x) = F_0(x) \quad 4.2.12$$

χ^2 -Anpassungstest Der χ^2 -Test kommt ohne eine Verteilungsannahme aus und überprüft, ob die empirische Verteilung der Stichprobe zu der angenommenen Verteilungsfunktion passt. Es können unterschiedliche Signifikanzniveaus zur Ablehnung der Nullhypothese angenommen werden. Für den χ^2 -Anpassungstest müssen die n unabhängigen und identisch verteilten Beobachtungen in g Klassen eingeteilt werden. Die Testprüfgröße ergibt sich als quadratische Abweichung der tatsächlichen Häufigkeit n_i und der erwarteten Häufigkeit $p_i n$, normiert mit der erwarteten Häufigkeit der Klasse, wobei p_i die diskrete Dichte darstellt. Die Testprüfgröße ist χ^2 verteilt.

$$\hat{\chi}^2 = \sum_{i=1}^g \frac{(n_i - p_i n)^2}{p_i n} \quad 4.2.13$$

Die Wahl der Klassenanzahl sollte so erfolgen, dass pro Klasse mindestens eins bis fünf Beobachtungen vorliegen. Für diesen Wert gibt es in der Literatur unterschiedliche Richtwerte (Hedderich & Sachs 2016). Liegen zu wenige Werte vor, müssen Klassen zusammengelegt werden. Dies betrifft vor allem den Rand der Verteilung. Die Nullhypothese wird abgelehnt, wenn $\hat{\chi}^2 \geq \chi_{v, 1-\alpha}^2$ gilt, wobei $v = g - 1$ die Anzahl der Freiheitsgrade angibt und über $1 - \alpha$ das Signifikanzniveau festgelegt wird.

Kolmogorov-Smirnov-Anpassungstest Der Kolmogorov-Smirnov-Anpassungstest dient der Prüfung, ob die empirische Verteilung zur geschätzten Verteilung passt. Die Nullhypothese H_0 besagt, dass die Daten der Verteilung F_0 folgen. Im ersten Schritt wird die empirische Verteilungsfunktion bestimmt, siehe Formel 4.2.8. Die Testprüfgröße \hat{D} wird als maximale absolute Abweichung der empirischen Verteilung zur geschätzten Verteilung berechnet. Die Grenzwerte lassen sich Tabellen entnehmen, siehe (Miller 1956).

$$\hat{D} = \max_{1 \leq i \leq n} |F_0(x_{(i)}) - \frac{i}{n}| \quad 4.2.14$$

Wenn die Hypothesentests eine Ablehnung der Nullhypothese nahelegen, wird auf die empirische Verteilung zurückgegriffen. Kann die Nullhypothese nicht verworfen werden, wird im Folgenden die theoretische Verteilung genutzt.

4.2.3 Korrektur saisonaler Effekte

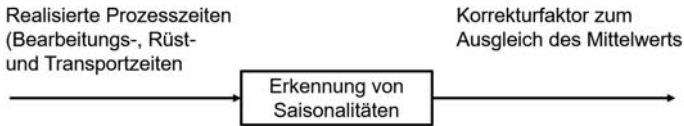


Abbildung 4.5: Erkennung von Saisonalitäten

Neben den graduellen oder sprunghaften Änderungen können Trends und Saisonalitäten eine Anpassung der Planzeiten nötig machen. Ergebnis der *Concept-Drift*-Erkennung ist eine Menge an Daten, die zu einer Verteilung gehören. Ist die Verteilung nicht stabil, enthält dieses Datenfenster nur einen kleinen Teil der Daten, sodass saisonale Effekte nicht mehr zu ermitteln sind. Daher wird für die Zeitreihenanalyse die gesamte Datenbasis genutzt. Ziel ist es, Änderungen des Mittelwerts der Bearbeitungs-, Transport- und Rüstzeiten über Schichten, Wochentage und Monate zu detektieren. Insbesondere bei manuellen Prozessen können diese Änderungen von der tageszeitabhängigen Leistungsfähigkeit der Menschen herrühren (Graf 1960). Ist der Unterschied zwischen dem Mittelwert des betrachteten Zeitraums und dem Mittelwert über den Gesamtzeitraum signifikant, wird ein Korrekturfaktor in Höhe der relativen Änderung eingeführt. Dieser wird genutzt, um den Planwert für den entsprechenden Zeitraum anzupassen, siehe Abbildung 4.5.

4.3 Multi-Objective Monte Carlo Tree Search

Im folgenden Abschnitt wird der auf Monte Carlo Tree Search basierende Optimierungsansatz vorgestellt. Zunächst wird das Steuerungsproblem als sequentielles Entscheidungsproblem in Form eines Markov-Entscheidungsproblems dargestellt. Anschließend werden die Weiterentwicklungen anhand der vier Phasen einer MCTS-Iteration erläutert.

4.3.1 Modellierung als Markov-Entscheidungsproblem

MCTS ist auf sequentielle Entscheidungsprobleme anwendbar. Das Produktionssteuerungsproblem muss daher als endliche Menge an Zuständen S , einer endliche Menge an Aktionen A , einem Aktionsmodell T , welches den Zustandsübergang von einem Ausgangszustand s_t durch eine Aktion a_t in einen Folgezustand s_{t+1} beschreibt sowie einer Belohnungsfunktion

$r : S \times A \times S \rightarrow \mathbb{R}$, die jedem Übergang eine Belohnung zuordnet, modelliert werden. Zudem muss der Ausgangszustand definiert sein. (Sutton & Barto 2015)

Der **Zustandsraum** S wird im Folgenden durch die Menge aller unterschiedlichen Belegungspläne für Produktionsmittel sowie für Produktionsaufträge definiert. Die Zustandsmenge enthält dabei nicht nur vollständige Belegungspläne, sondern alle möglichen Belegungspläne, die auch nur einen Teil der einzuplanenden Vorgänge enthalten können. Jeder Knoten des MCTS-Suchbaums entspricht einem Zustand des Markov-Entscheidungsproblems. Je weiter unten sich der Knoten im Suchbaum befindet, umso mehr Vorgänge sind im Belegungsplan enthalten. Sind alle Vorgänge eingeplant, handelt es sich um einen Blattknoten, auch Terminalknoten genannt. In einem Belegungsplan für Produktionsmittel werden die Produktionsmittel auf der Ordinate abgetragen und die Zeit auf der Abszisse. Ist ein Produktionsmittel mit einem Bearbeitungs- oder Rüstvorgang belegt, wird dieser an der entsprechenden Stelle eingetragen. Der Belegungsplan aus Sicht der Produktionsaufträge enthält alle Produktionsmittel auf der Ordinate und die Zeit auf der Abszisse. Neben den Bearbeitungsvorgängen werden hier die Transportvorgänge eingetragen, siehe Abbildung 4.6.

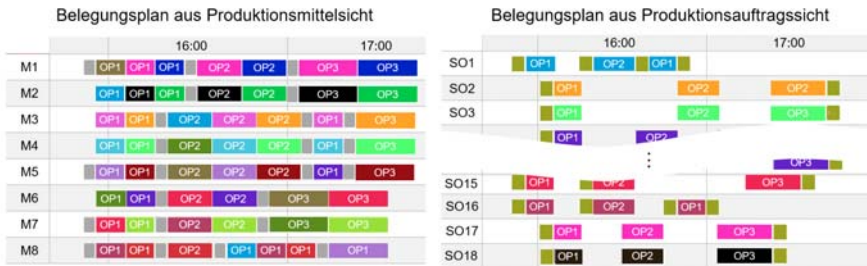


Abbildung 4.6: (links) Belegungsplan für Produktionsmittel mit Bearbeitungs- und Rüstvorgängen (grau). Die Farben fassen Produktionsaufträge zusammen. (rechts) Belegungsplan für Produktionsaufträge mit Bearbeitungs- und Transportvorgängen (grün)

Die **Menge der Aktionen** A ergibt sich aus der Vereinigungsmenge möglicher Aktionen ausgehend von einem Zustand, $A = \cup A(s), \forall s \in S$. Eine Aktion entspricht der Einplanung eines Transport-, Rüst-, oder Bearbeitungsvorgangs. Die Bestimmung der möglichen Aktionen wird durch die Produktionssteuerung übernommen, abhängig vom aktuellen Zustand $A_s = A(s)$ und dem damit verbundenen **Zustandsübergang** von s_t zu s_{t+1} . Es wird stets der frühestmögliche Einplanungszeitpunkt eines Vorgangs auf einem Produktionsmittel betrachtet. Die Restriktionen aus Rüst- und Transportvorgängen werden dabei berücksichtigt. Aktionen werden im MCTS-Suchbaum als Kanten dargestellt, siehe Abbildung 4.7.

Die **Bewertung der Zustände** erfolgt anhand der in Abschnitt 4.1 beschriebenen Bewertungsfunktionen. MCTS evaluiert keine Zwischenzustände, sondern nimmt die Evaluation nur für Terminalzustände, Blattknoten, vor. Die Bewertung wird anschließend über alle Zustände auf dem Weg des zum Blattknoten gehörigen Pfads zurückpropagiert.

Der **Ausgangszustand** des Markov-Entscheidungsproblems ist im Wurzelknoten des Suchbaums gespeichert. Dieser Wurzelknoten bildet den aktuellen Zustand zum betrachteten Zeitpunkt t_0 ab. Am Anfang einer Schicht entspricht dies einem leeren Belegungsplan. Im Fall einer abweichungsbedingten Umplanung beinhaltet der Ausgangszustand die bereits eingeplanten sowie die nicht verschiebbaren Vorgänge. Abbildung 4.7 zeigt den Zusammenhang zwischen dem Suchbaum, den Zuständen und Aktionen im Falle einer Neuplanung beispielsweise zu Schichtbeginn.

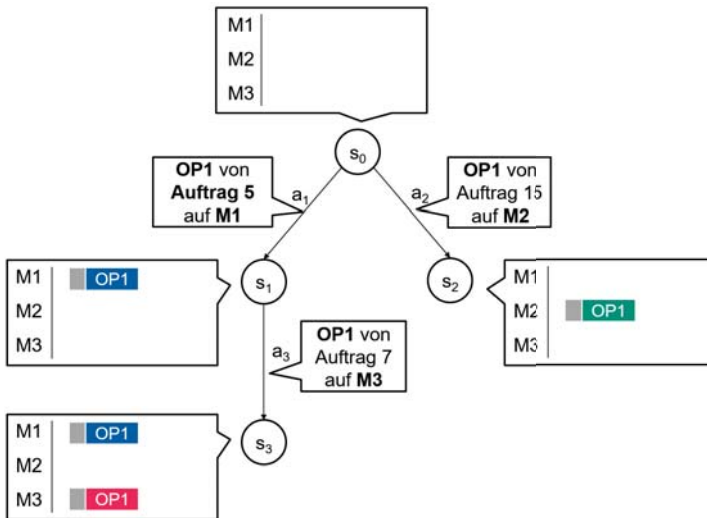


Abbildung 4.7: Schematische Darstellung des Suchbaums, der Zustände $s_0 - s_3$ und der Aktionen $a_1 - a_3$ für die Produktionsmittel M1 bis M3, den Vorgang OP1 und die Aufträge 5 (blau), 7 (rot) und 15 (grün)

4.3.2 Modifikationen der Selektionsphase

Ziel der Selektionsphase ist die Auswahl des vielversprechendsten Zustands für die weitere Expansion. Hierbei gilt es, eine Balance zwischen der verstärkten Suche in Bereichen des Lösungsraums mit guten Ergebnissen (*Exploitation*) und der Erkundung noch unbekannter

Bereiche (Exploration) zu finden. Auf die Bewertung der Terminalzustände wird im Unterkapitel 4.3.5 eingegangen. Im folgenden Abschnitt werden zwei Aspekte thematisiert: Erstens die Modellierung der Bewertung für die unikriterielle und multikriterielle Optimierung und zweitens das Ausbalancieren von *Exploitation* und Exploration.

Die Struktur der grundlegenden Selektionsfunktion ist durch einen *Exploitation*-Term gegeben, der auf der Güte der Zielkriterien basiert und einen Explorationsterm, der hohe Werte annimmt, wenn der Knoten noch wenig besucht wurde. Der Explorationsterm wird durch einen Explorationsfaktor gewichtet. Für die Bewertung eines Zustands wird gemäß des Stands der Technik die UCT-Funktion herangezogen (Kocsis & Szepesvári 2006; Kocsis & Szepesvári u. a. 2006), siehe Formel 2.3.1 und Abschnitt 2.3.1.

4.3.2.1 Modellierung der multikriteriellen Zielfunktionen

Dieser Abschnitt beschäftigt sich mit zwei Ansätzen, mehrere Zielgrößen im *Exploitation*-Term abzubilden. Zum einen besteht die Möglichkeit, die einzelnen Zielgrößen x_i mit a priori bekannten Gewichtungen γ_i zu einem Zielwert zu aggregieren. Das gewichtete Mittel verfolgt diesen Ansatz, siehe Gleichung 4.3.1. In diesem Fall kann die UCT-Funktion, siehe 2.3.1, zur Bewertung des Zustands r_s verwendet werden.

$$r_s = \sum_{x_i \in X} \gamma_i x_i \tag{4.3.1}$$

$$\sum_{\gamma_i \in \Gamma} \gamma_i = 1$$

Zum anderen kann ein Pareto-Ansatz verfolgt werden. In diesem Fall stellt die Ausdehnung der Pareto-Front dar, wie gut der Lösungsraum abgedeckt wird. Eine Lösung gilt als Pareto-optimal (effizient), wenn jede andere Lösung hinsichtlich mindestens eines der Zielkriterien schlechter oder gleich gut ist. Diese Menge wird als Pareto-Menge oder Pareto-Front bezeichnet (Domschke & Drexl u. a. 2015). Die Breite der Pareto-Front lässt sich mit Hilfe des Hypervolumens bemessen, definiert als der durch die Pareto-Punkte dominierte Raum (Zitzler & Thiele 1999). Um das Hypervolumen zu bestimmen, muss durch jeden Punkt aus der Pareto-Menge jeweils eine Ebene senkrecht zu der jeweiligen Achsen gelegt werden. Das Hypervolumen ergibt sich als Vereinigung der Polytope, gebildet aus den Schnittgeraden der Ebenen (Zitzler & Thiele 1999). Abbildung 4.8 illustriert das Hypervolumen, das sich für drei Pareto-Punkte im dreidimensionalen Lösungsraum ergibt.

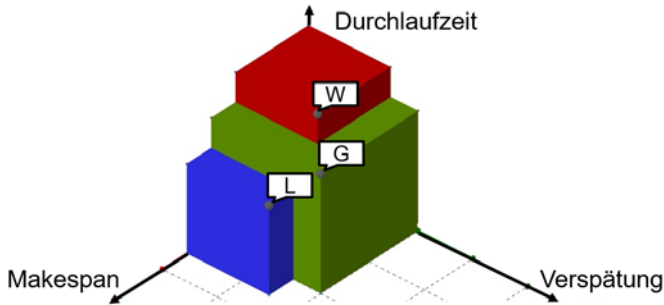


Abbildung 4.8: Hypervolumen der Pareto-Punkte L, G, W im dreidimensionalen Raum

Die Bewertung der Zustände erfolgt gemäß Formel 4.3.2, analog zur Formel 2.3.1.

$$UCT_{HV}(s, a) = Q_{HV}(s, a) + c \sqrt{\frac{\ln(N(s))}{N(s, a)}} \quad 4.3.2$$

Die Wahl des Zustands erfolgt nach Formel 4.3.3 und Formel 2.3.3 analog zum unikriteriellen Fall für den Suchbaum τ .

$$\operatorname{argmax}_a UCT_{HV}(s, a), \quad \forall (s, a) \in \tau \quad 4.3.3$$

4.3.2.2 Balance zwischen Exploration und Exploitation

Nachdem im vorangegangenen Abschnitt der *Exploitation*-Term der UCT-Funktion behandelt wurde, liegt jetzt der Schwerpunkt auf der Gewichtung zwischen *Exploitation* und Exploration. Das Suchverhalten lässt sich durch zwei Maßnahmen beeinflussen, zum einen durch die Anpassung der Gewichtungen der einzelnen Zielgrößen und zum anderen durch die Wahl des Explorationsfaktors c . Im Weiteren werden ein statischer, ein abnehmender und ein dynamischer Ansatz zur Bestimmung dieses Wertes vorgestellt. Anschließend werden zwei Ansätze der Steuerung des Suchverhaltens durch Variation der Zielgewichtungen aufgezeigt.

Statischer Explorationsfaktor

Die Wahl des Explorationsfaktors beeinflusst das Wachstum des Suchbaums. Eine geringe Exploration führt zu tiefen, aber schmalen Bäumen und so zur Gefahr, frühzeitig in einem

lokalen Optimum zu verharren. Eine hohe Exploration begünstigt breite Suchbäume mit geringer Tiefe, aber auch einen langsamen Anstieg der Lösungsgüte, siehe Abbildung 4.9.

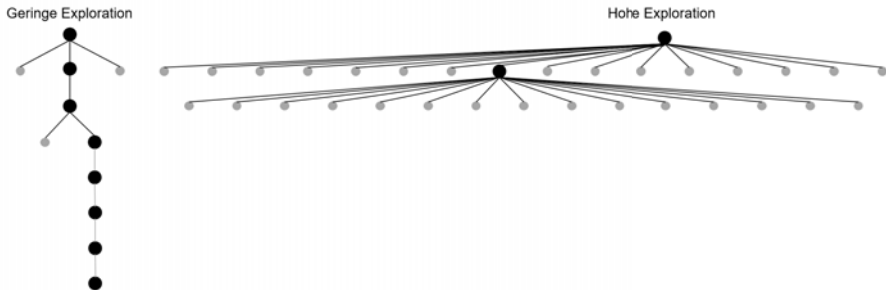


Abbildung 4.9: Suchbaum mit geringer Exploration (links) und hoher Exploration (rechts)

Ein Ansatz ist es, einen statischen Explorationsfaktor zu bestimmen, der über die gesamten MCTS-Iterationen unverändert bleibt.

Abnehmender Explorationsfaktor

Die Grundidee des abnehmenden Explorationsfaktors ist es, dass die Exploration am Anfang der Optimierung hoch sein sollte, da zu diesem Zeitpunkt wenig über den Lösungsraum bekannt ist. Somit wird am Anfang ein Wachstum des Suchbaums in die Breite gefördert. Der Explorationsfaktor nimmt mit zunehmender Anzahl Iterationen oder über die bereitgestellte Rechenzeit ab, sodass der Suchfokus zunehmend auf die Lösungsbereiche mit den besten bekannten Lösungen beschränkt ist. Dieser Ansatz hat Parallelen zu den abnehmenden Raten des *Simulated Annealings*. Die Gleichung 4.3.4 zeigt die Berechnung eines linear abnehmende C -Werts abhängig von der aktuellen Iteration k im Verhältnis zur Gesamtzahl der Iterationen K , wobei c_e der Endwert des Parameters c ist und c_s dessen Startwert markiert.

$$c(k) = c_s + \frac{k}{K}(c_e - c_s), c_s > c_e \quad 4.3.4$$

Dynamischer Explorationsfaktor

Der Ansatz des dynamischen Explorationsfaktors folgt der Idee der linearen Anpassung des Explorationsfaktors und kombiniert diesen mit einem Mechanismus, der bei Stagnation der Lösungsgüte den Suchfokus erneut zu Gunsten der Exploration einstellt. Der Ansatz besteht aus zwei Phasen. Ausgehend von einem initialen Wert wird in der ersten Phase der C -Wert

linear über die Anzahl der Iterationen verringert. In dieser Phase steigt somit die *Exploitation* guter Lösungen.

Sobald der MCTS in einem lokalen Minimum gefangen ist, geht das Verfahren in die zweite Phase über. Ob der Algorithmus in einem lokalen Minimum stagniert, ist nicht eindeutig bestimmbar. Ein Indiz ist die Wahl eines Terminalzustands in der Selektionsphase, da dann der Suchbaum bis zu seiner maximalen Tiefe aufgebaut sein muss. Dies ist nur möglich, wenn der Bereich bereits sehr umfangreich erkundet wurde. In der zweiten Phase wird der C -Wert sprunghaft um eine feste Größe erhöht, sodass erneut Exploration betrieben wird. Danach wird der C -Wert wieder linear verringert, siehe Algorithmus 1. Der Algorithmus baut auf dem Verfahren der vom Autor angeleiteten Abschlussarbeit (A_Deining 2020) auf.

Algorithmus 1 Dynamische Adaption des Explorationsfaktors

```

1: procedure DYNAMIC C
2:    $c_0 \leftarrow$  Startwert des Explorationsfaktors
3:    $\delta \leftarrow$  festes Dekrement
4:    $\Delta \leftarrow$  festes Inkrement
5:    $S \leftarrow$  aktueller Zustand
6:    $k \leftarrow$  aktuelle Iteration
7:    $c_i \leftarrow c_0$  ▷ Initialisierung des aktuellen Werts  $c_i$  mit dem Startwert

8:   if  $\text{Kindknoten}(S) = \emptyset$  then ▷ Blattknoten ohne Kindknoten erreicht
9:      $k \leftarrow 0$  ▷ Zurücksetzen des Iterationszählers  $k$ 
10:     $c_0 \leftarrow c_i + \Delta$  ▷ Neuer Startwert
11:    return  $c_0$ 
12:   else
13:      $k \leftarrow k + 1$  ▷ Iterationszähler erhöhen
14:      $c_i \leftarrow c_0 - k\delta$  ▷ C-Wert um Dekrement verringern
15:     return  $c_i$ 
16:   end if

17: end procedure

```

Lineare Variation der Zielgewichtung

Dieser Ansatz gilt nur für zwei Zielgrößen. Grundidee ist die lineare Anpassung der Gewichtung zwischen den beiden Zielgrößen ausgehend von einem Startwert $w_{i,s}$, $i \in \{1, 2\}$ bis zu einem Endwert $w_{i,e}$, $i \in 1, 2$ abhängig von der aktuellen Iteration. Formel 4.3.5 zeigt, wie sich die Zielgewichtung abhängig von der aktuellen Iteration k berechnet. Durch diesen Ansatz

schwenkt der Suchfokus linear über die Iterationen von einem Zielkriterium zum anderen.

$$w_i(k) = w_{i,s} + \frac{k}{K}(w_{i,e} - w_{i,s}), i \in \{1, 2\} \quad 4.3.5$$

Alternierende Zielgewichtung

Bei der rotierenden Zielgewichtung wird nur ein Ziel bei der Selektion berücksichtigt, siehe Formel 4.3.6. Die Zielgewichtung rotiert jedoch iterationsweise zwischen den Zielen. Der Suchbaum wächst somit abwechselnd nach jeweils einem Zielkriterium (A_Wurster 2019).

$$w_i(k) = \begin{cases} 0, & \text{wenn Ziel } i \text{ in Iteration } k \text{ gewählt} \\ 1, & \text{sonst} \end{cases} \quad 4.3.6$$

Parallelisierung von Selektionsfunktionen

Eine Parallelisierung des MO-MCTS-Verfahrens, siehe Abschnitt 4.4, kann genutzt werden, um zeitgleich unterschiedliche Selektionsfunktionen anzuwenden. Bei der *Root*-Parallelisierung werden parallel mehrere Suchbäume mit jeweils einer anderen Selektionsfunktion aufgebaut. Nach Ende des Optimierungslaufs wird die Vereinigungsmenge aller Pareto-Mengen aus den unterschiedlichen Pareto-Mengen der parallel aufgebauten Suchbäume gebildet.

4.3.3 Modifikationen der Expansionsphase

In der Expansionsphase wird eine zulässige Aktion basierend auf dem aktuellen Zustand gewählt. Zum einen kann die Menge der zulässigen Aktionen sehr groß sein, sodass deren Ermittlung zeitaufwendig ist. Zum anderen ist die Wahl einer vielversprechenden Aktion aus dieser Menge nicht trivial. Die Arbeiten aus dem Stand der Technik, siehe Abschnitt 3.3, haben gezeigt, dass die erzielte Lösungsgüte deutlich gesteigert werden kann, wenn auf Domänenwissen in Form von problemspezifischen Heuristiken in der Aktionsgenerierung zurückgegriffen wird. Nachteil dieses Vorgehens ist jedoch, dass die Auswahl der passenden problemspezifischen Heuristiken abhängig von der Struktur des Steuerungsproblems und der jeweiligen Zielgrößen ist. Diese Spezialisierung führt zu einer geringeren Leistungsfähigkeit bei generischen Problemen.

Im Folgenden wird ein Vorgehen zur Aktionsbestimmung in drei Schritten vorgestellt, siehe Abbildung 4.10. Der erste Schritt des Vorgehens beschäftigt sich mit der Festlegung der

zu betrachtenden Produktionsaufträge. Im zweiten Schritt werden für diese Produktionsaufträge alle zulässigen Aktionen generiert. Im dritten Schritt wird diese Menge zulässiger Aktionen eingeschränkt. Die verbleibenden Aktionen werden durch den MO-MCTS in der Expansionsphase verarbeitet.

Dieses Verfahren zur Aktionsgenerierung schränkt lediglich die gleichzeitig betrachteten Aktionen ein. In der nächsten Iteration des MO-MCTS werden die zur Verfügung stehenden Aktionen erneut ermittelt.

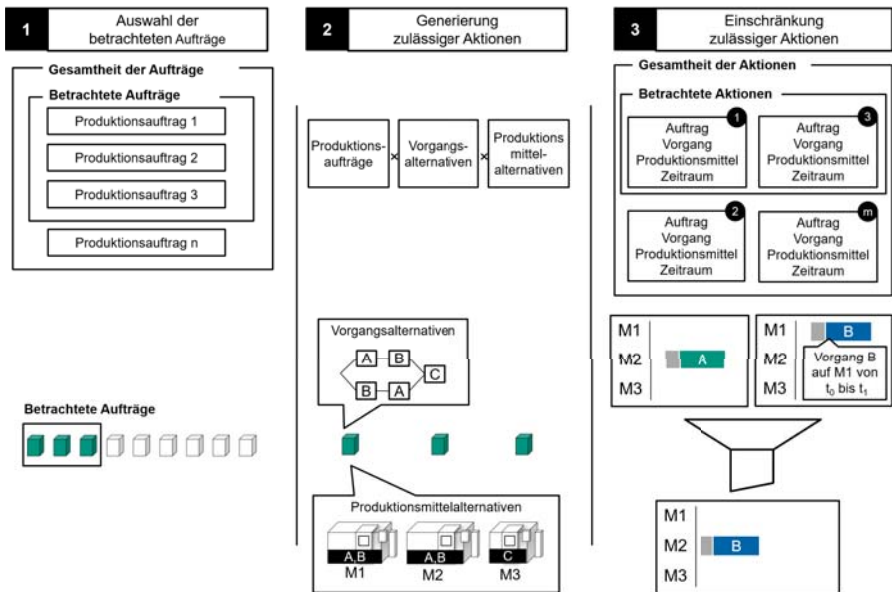


Abbildung 4.10: Generierung und Einschränkung von Aktionen in der Expansionsphase

Die Auswahl der betrachteten Produktionsaufträge in Schritt 1 und die Bestimmung der zulässigen Aktionen in Schritt 3 kann durch problemspezifische Heuristiken erfolgen. In den nächsten drei Abschnitten werden die Schritte 1 bis 3 vorgestellt. Die Auswahl der passenden Heuristiken kann analog zu den Arbeiten im Stand der Forschung (Wu & Wu u. a. 2013; Chou & Liang u. a. 2015; Lu & Chiu u. a. 2016) durch Domänenexperten erfolgen. Im vierten Abschnitt dieses Kapitels wird jedoch eine neue Methode eingeführt, die selbstständig bestimmt, welche problemspezifischen Heuristiken in den Schritten 1 und 3 genutzt werden sollten und diese Auswahl fortwährend prüft und anpasst. Diese Methode soll im Sinne der Forschungsfrage F2 ermöglichen, Domänenwissen automatisiert einzusetzen, um die

Steuerung an das jeweils vorliegende Steuerungsproblem anzupassen.

4.3.3.1 Schritt 1: Auswahl der betrachteten Produktionsaufträge

Die möglichen Aktionen in einem Zustand werden ausgehend von den Produktionsaufträgen ermittelt. Die Anzahl der parallel betrachteten Produktionsaufträge hat einen großen Einfluss auf die resultierende Menge der zulässigen Lösungen. Durch die Limitierung der parallel betrachteten Produktionsaufträge kann die Komplexität gesenkt werden, gleichzeitig werden bestimmte Lösungen durch diese Einschränkungen unerreichbar. Wird die Anzahl der Produktionsaufträge limitiert, muss eine Auswahl getroffen werden, welche Aufträge in die Betrachtung einbezogen werden können. Hierfür werden die Produktionsaufträge anhand von Kriterien sortiert. Herrscht nach einem Kriterium Indifferenz, wird ein weiteres Kriterium angewendet. Folgende Kriterien werden betrachtet:

- **Priorität:** Durch den Produktionsplaner werden Prioritäten vergeben, beispielsweise eine erhöhte Priorität für Eilaufträge.
- **Status:** Bereits begonnene Produktionsaufträge werden gegenüber noch nicht angefangenen Produktionsaufträgen präferiert.
- **Frühster Startzeitpunkt:** Die Produktionsaufträge mit den frühesten möglichen Startzeitpunkten werden priorisiert.
- **Frühster spätester Fertigstellungszeitpunkt:** Die zeitkritischsten Produktionsaufträge werden priorisiert.
- **Kürzeste Bearbeitungszeit:** Die Produktionsaufträge mit den kürzesten Gesamtbearbeitungszeit werden priorisiert.
- **Längste Bearbeitungszeit:** Die Produktionsaufträge mit den längsten Gesamtbearbeitungszeit werden priorisiert.
- **Name:** Die Produktionsaufträge werden aufsteigend alphabetisch sortiert. Dieses Kriterium stellt sicher, dass bei Indifferenz eine Entscheidung getroffen werden kann.

Die Auswahlkriterien müssen für die Sortierung in eine Rangfolge gebracht werden. Ergebnis dieser Phase ist eine geordnete Liste zu betrachtender Produktionsaufträge. In der nächsten Phase werden alle zulässigen Aktionen für eine Teilmenge fester Größe der Produktionsaufträge ermittelt. Diese Anzahl betrachteter Produktionsaufträge beeinflusst die Komplexität und muss durch einen Experten festgelegt werden.

4.3.3.2 Schritt 2: Generierung zulässiger Aktionen

Auf Basis der verbleibenden Produktionsaufträge werden in diesem Schritt alle zulässigen Aktionen ermittelt. Jedes einzigartige Tupel aus Produktionsauftrag, Vorgang und Produktionsmittel stellt eine Aktion dar. Für jedes Tupel wird der früheste Einplanungszeitraum berücksichtigt.

Der erste Teilschritt ist die Identifikation aller möglichen Vorgänge anhand des Vorranggraphs. Danach werden die frühesten Einplanungszeiträume auf den Produktionsmitteln bestimmt, die den geforderten Produktionsprozess anbieten. Im letzten Schritt werden für die entstandenen Aktionen die frühesten Start- und Endzeitpunkte ermittelt. Bei der Identifikation dieser Korridore müssen Rüst- und Transportvorgänge berücksichtigt werden.

$$a_{ts}(m, v, p) = \max(t_i, t_e^{m, v_{prev}} + t_{su}(v_{prev}, v), t_e^{p, v_{prev}} + t_t(l_{prev}, l_m)) \quad 4.3.7$$

$$a_{te}(m, v, p) = a_{ts}(m, v, p) + t_{m, v} \quad 4.3.8$$

Formel 4.3.7 zeigt die Berechnung des frühesten Startzeitpunkts abhängig vom Produktionsmittel m , Bearbeitungsvorgang v und Produktionsauftrag p . Der früheste Startzeitpunkt wird limitiert durch den frühesten Zeitpunkt, ab dem das Produktionsmittel frei ist oder ab dem der Produktionsauftrag am Produktionsmittel zur Verfügung steht. Handelt es sich um den ersten Produktionsauftrag, ist die früheste Startzeit durch den aktuellen Zeitpunkt gegeben. Der früheste Belegungszeitpunkt des Produktionsmittels errechnet sich als Endzeitpunkt des vorangegangenen Produktionsprozesses auf dem Produktionsmittel $t_e^{m, v_{prev}}$ zuzüglich der Rüstzeit vom vorangegangenen Produktionsprozess auf den benötigten Produktionsprozess $t_{su}(v_{prev}, v)$. Der früheste Ankunftszeitpunkt des Produktionsauftrags ergibt sich aus dem Ende des vorangegangenen Produktionsprozesses $t_e^{p, v_{prev}}$ plus der Transportzeit zum Produktionsmittel $t_t(l_{prev}, l_m)$. Handelt es sich um den ersten Produktionsschritt oder ist das Produktionsmittel noch nicht belegt worden, ergibt sich der früheste Anfangszeitpunkt als aktueller Zeitpunkt t_i . Der Endzeitpunkt wird errechnet als frühester Startzeitpunkt zuzüglich der Bearbeitungsdauer des Produktionsvorgangs v auf dem Produktionsmittel m , siehe Formel 4.3.8.

Als Ergebnis dieses Teilschritts liegen alle zulässigen Aktionen für die betrachteten Produktionsaufträge vor, bestehend aus den frühestmöglichen Einplanungszeiträumen für alle Kombinationen aus Vorgang und Produktionsmittel.

4.3.3.3 Schritt 3: Einschränkung zulässiger Aktionen

In dieser Phase werden die zulässigen Aktionen durch sequentielle Anwendung problemspezifischer Heuristiken in Form von Prioritätsregeln reduziert. Diese Prioritätsregeln werden daher im Weiteren als Filter bezeichnet. Dieses Vorgehen kann die Suchgeschwindigkeit durch die Reduktion des Lösungsraums stark erhöhen. Jedoch besteht auch die Gefahr durch die Anwendung der Filter gute Lösungen aus dem Lösungsraum auszuschließen. Im Folgenden werden einige mögliche Filter vorgestellt:

- **Dringlichste Produktionsaufträge:** Nur Produktionsvorgänge, die zu den Produktionsaufträgen mit dem frühesten Fertigstellungszeitpunkt gehören
- **Schnellste Maschinen-Prozess-Kombination:** Nur die Alternativen aus Produktionsmittel und Vorgang mit der kürzesten Bearbeitungszeit für jeden Vorgang.
- **Keine Rüst- und Transportvorgänge:** Nur Alternativen ohne Rüst- und Transportvorgänge
- **Keine Rüstvorgänge:** Nur Alternativen ohne Rüstvorgänge
- **Keine Transportvorgänge:** Nur Alternativen ohne Transportvorgänge
- **Nach Produktionsstatus:** Nur Produktionsvorgänge, die zu Produktionsaufträgen mit einem bestimmten Status gehören, beispielsweise begonnen oder verspätet
- **Nur frühestmögliche Vorgänge:** Nur die frühestmöglichen Vorgänge der Produktionsaufträge
- **Nur sich ausschließende Vorgänge:** Der erste mögliche Vorgang auf einem Produktionsmittel bildet die Basis. Anschließend werden zwei Mengen gebildet. Die erste Menge enthält alle Vorgänge, die zueinander Überschneidungen haben. Die zweite Menge enthält alle Vorgänge, die keine Überschneidungen zu Vorgängen aus der ersten Menge aufweisen. Die Entscheidungen über die Vorgänge aus der zweiten Menge können demnach ohne Verlust später getroffen werden.
- **Makespan-Grenze:** Nur Alternativen, die nicht zu einer Verlängerung des besten gefundenen Makespan führen. Dieser Filter berücksichtigt jedoch nur Verletzungen des Makespan durch die in diesem Schritt betrachtete Aktion.
- **Früherster Startzeitpunkt:** Es werden aus Sicht der Produktionsmittel nur die Vorgänge ausgewählt, die am frühesten starten. Bei Rüstvorgängen wird der Anfangszeitpunkt des Rüstvorgangs herangezogen.

Aus der Menge an Filtern wird eine Teilmenge ausgewählt. Diese Teilmenge wird als Filterkombination bezeichnet. Bleibt nach Anwendung eines Filters keine Aktion mehr übrig, wird der entsprechende Filter ausgelassen, sodass Aktionen für die folgende MCTS-Phase verbleiben. Die Gesamtmenge möglicher Filter kann erweitert werden. Die hier vorgestellte Auswahl hat sich im Laufe der Arbeiten als günstig erwiesen. Da der Fokus auf der Methodik liegt, siehe Abschnitt 4.3.3.4, wird die Auswahl der einzelnen Filter nicht näher erläutert.

4.3.3.4 Methode zur intelligenten Einschränkung der zulässigen Aktionen

Die Auswahl der geeigneten Kriterien zur Sortierung der Produktionsaufträge im Schritt 1 und der Filterkombinationen im Schritt 3 ist nur experimentell bestimmbar und problemabhängig. Zum einen macht diese Eigenschaft eine breite Anwendbarkeit der Methodik schwierig. Zum anderen kann sich die Struktur des Steuerungsproblems, hinsichtlich der zur Verfügung stehenden Produktionsmittel und angebotenen Prozesse sowie bezüglich des Produktionsprogramms aufgrund der hohen Flexibilität und Wandlungsfähigkeit der Matrixproduktion ändern, sodass die Auswahl der anzuwendenden Sortierkriterien und Filterkombinationen regelmäßig überprüft und angepasst werden sollte.

Ziel der Methode zur intelligenten Einschränkung der zulässigen Aktionen ist die Entwicklung eines automatisierten Mechanismus, der die Kriterien zur Sortierung der Produktionsaufträge und Filterkombinationen so ermittelt, dass die Lösungsgüte nicht sinkt, obwohl die Lösungsgeschwindigkeit ansteigt. Dies ist genau dann der Fall, wenn dominierte Lösungen aus dem Lösungsraum ausgeschlossen werden.

Das Verfahren hat Parallelen mit genetischen Algorithmen. Das Vorgehen gliedert sich in zwei Phasen. In der ersten Phase, dem **Experimentierbetrieb**, werden Sortierkriterien und Filterkombinationen zufällig angewendet. Basierend auf der erzielten Lösungsgüte in der *Rollout*-Phase des MO-MCTS werden erfolgreiche Kombinationen beibehalten. Dieser Prozess wird über mehrere MO-MCTS-Durchläufe wiederholt. In der zweiten Phase, dem **Produktivbetrieb**, werden diese erfolgreichen Sortierkriterien und Filterkombinationen verwendet. Auch im Produktivbetrieb wird zu einem kleinen Prozentsatz Exploration betrieben, sodass weiterhin neue Kombination ausprobiert werden. Sind diese neuen Kombinationen erfolgreich, werden sie in die Menge der erfolgreichen Kombinationen aufgenommen. Dieser Mechanismus bewirkt auch in der Produktivphase eine Anpassungsfähigkeit. Im Folgenden werden die beiden Phasen detaillierter vorgestellt (Stricker & Kuhle u. a. 2021).

Im Experimentierbetrieb, wird die Grundmenge aller zur Verfügung stehenden Sortierkriterien und Filterkombinationen und deren Parameter initialisiert. Ein Parameter der Filterheuris-

tik "Anzahl betrachteter Produktionsaufträge" ist beispielsweise die Anzahl betrachteter Produktionsaufträge selbst. Jedem Sortierkriterium und jeder Filterkombination wird eine Wahrscheinlichkeit zugeordnet, mit der dieser angewendet wird. Ist kein a priori-Wissen bekannt, wird eine Gleichverteilung angenommen. Sobald der Experimentierbetrieb startet, werden die Sortierkriterien und Filterkombinationen entsprechend ihrer Wahrscheinlichkeit ausgewählt und in der *Rollout*-Phase des MO-MCTS angewendet. Sobald das Terminierungskriterium der MO-MCTS-Optimierung erreicht ist, werden die Ergebnisse ausgewertet und daraufhin die Wahrscheinlichkeiten für die Wahl der Sortierkriterien und Filterkombinationen angepasst. Dieser Experimentierbetrieb kann mehrfach durchlaufen werden. Im Weiteren wird aus Gründen der Lesbarkeit nur von Filterkombinationen gesprochen. Hiermit werden sowohl die Sortierkriterien als auch die Filterkombinationen gemeint. Die Auswahl der Filterkombinationen erfolgt nach drei Kriterien:

1. **Pareto-Punkte:** Alle Filterkombinationen, die zum Entstehen der besten n Pareto-Lösung über alle Durchläufe geführt haben, werden ausgewählt.
2. **Pareto-Nähe:** Im Abstand ξ zur Pareto-Front werden Hyperebenen aufgespannt. Die γ besten Lösungen, die im Raum zwischen der Hyperebene und der Pareto-Front liegen, werden nach folgenden Kriterien ausgewählt. Erstens, die zu der Lösungen gehörende Filterkombination muss mindestens mit der Wahrscheinlichkeit σ zu Lösungen in der Menge geführt haben. Zweitens, es müssen mehr als die notwendige Mindestanzahl an erzielten Lösungen n_{min} in dem relevanten Bereich liegen.
3. **ϵ -Kriterium:** Dieses Kriterium wählt zufällig Filterkombinationen aus dem Bereich zwischen der Pareto-Front und der unteren Schranke, aufgespannt durch die Hyperebenen aus, die noch nicht selektiert wurden. Dieses Vorgehen erhöht die Entropie.

Formal besteht die Gesamtmenge der Filterkombinationen aus drei Teilmengen. Erstens, die Menge der Filterkombinationen, die über das Pareto-Kriterium gewählt werden Θ^p , zweitens aus der Menge der Filterkombinationen, die über die Nähe zur Pareto-Front gewählt werden Θ^m und drittens aus der Menge der Filterkombinationen, die über das zufallsbasierte ϵ -Kriterium gewählt werden, siehe Formel 4.3.9.

$$\Theta^* = \Theta^p \cup \Theta_\gamma^m \cup \Theta^\epsilon \quad 4.3.9$$

Die Menge der Pareto-Filterkombinationen Θ^p errechnet sich aus allen Filterkombinationen θ aus der Gesamtmenge der untersuchten Filterkombinationen Θ für die gilt, dass der

Zielfunktionswert θ_x unter den n besten Pareto-Punkten $\Psi(\bar{\Theta}, i)$ ist, siehe Formel 4.3.10.

$$\Theta^p = \{\forall \theta \in \bar{\Theta} : \theta_x \in \bigcup_{i=1}^n \Psi(\bar{\Theta}, i)\} \quad 4.3.10$$

Die zweite Menge ist gegeben durch die Punkte, die das Kriterium der Pareto-Nähe erfüllen. Formel 4.3.11 beschreibt die Menge der Hyperebenen, die die Pareto-Nähe definieren. Alle Vektoren \vec{x} des z -dimensionalen Lösungsraums sind Punkte der Hyperebene, wenn sie die Gleichung in Normalenform erfüllen. Die Stützvektoren \vec{p} werden durch die Pareto-Punkte definiert.

$$D_\xi = \bigcup_{\vec{p} \in \Psi(\bar{\Theta}, 1)} \{\forall \vec{x} \in \mathbb{R}^z : (\xi \vec{p} - \vec{x}) \vec{n} = 0\} \quad 4.3.11$$

Für die Menge der Filterkombinationen, die das Pareto-Nähe-Kriterium erfüllen, gilt, dass ein Punkt aus der Hyperebene existiert der kleiner oder gleich dem erreichten Zielwert des Filters θ_x ist, siehe Formel 4.3.12.

$$\Lambda(\bar{\Theta}, \xi) = \{\forall \theta \in \bar{\Theta} : (\exists \vec{x} \in D_\xi : \vec{x} \leq \theta_x)\} \quad 4.3.12$$

Für die Teilmenge der ausgewählten Filterkombinationen auf Basis des Pareto-Nähe-Kriteriums werden nur die Filterkombinationen ausgewählt, die mindestens n_{min} -mal gewählt wurden und mit einer Wahrscheinlichkeit von mindestens σ zu Lösungen in der Menge der Pareto-nahen Lösungen führen, siehe Formel 4.3.13.

$$\Theta^{pm} = \{\forall \theta \in \bar{\Theta} : (P(\theta | \theta \in \Lambda(\bar{\Theta}, \xi)) \geq \sigma \wedge |\bar{\Theta}(\theta)| \geq n_{min})\} \quad 4.3.13$$

Sollte die Menge der Pareto-nahen-Filterkombinationen die Gesamtmenge der zu wählenden Filterkombinationen γ überschreiten, werden die besten γ Filterkombinationen nach dem durchschnittlichen Hypervolumen gewählt, siehe Formel 4.3.14. $\Phi(\Theta^{pm}, i)$ beschreibt hierbei die absteigend geordnete Menge nach dem durchschnittlichen Hypervolumen der Filterkombination.

$$\Theta_\gamma^{pm} = \bigcup_{i=1}^{\gamma} \Phi(\Theta^{pm}, i) \quad 4.3.14$$

Die letzte Menge Θ^ϵ besteht aus zufällig ausgewählten Filterkombinationen aus der Menge der Pareto-nahen Filterkombinationen Θ^{pm} ohne die bereits ausgewählten Filterkombinationen, siehe Formel 4.3.15. Die Kardinalität der Menge ergibt sich aus dem Anteil ϵ zufälliger Filterkombinationen an der Gesamtzahl γ der zu wählenden Filterkombinationen.

$$\Theta^\epsilon = \bigcup_{i=1}^{\lfloor \epsilon \gamma \rfloor} \Theta^{pm} \setminus \Theta_\gamma^{pm} \quad 4.3.15$$

Die Gesamtmenge der ausgewählten Filterkombinationen ergibt sich aus der Vereinigungsmenge Θ^* der drei vorgestellten Teilmengen, siehe Formel 4.3.9.

In der zweiten Phase, der Produktivphase, wird während der Expansions- und *Rollout*-Phase des MO-MCTS einer der Filterkombinationen aus der Menge der ausgewählten Filterkombinationen gewählt. Mit einer Wahrscheinlichkeit ϵ_{prod} wird eine neue Filterkombination erzeugt. Auch in der Produktivphase werden die angewendeten Filterkombinationen und die erzielten Lösungen nach jedem MO-MCTS-Durchlauf gespeichert. Vor der nächsten Steuerungsentscheidung werden erneut die besten Filterkombinationen nach den genannten Kriterien ausgewählt. Ändert sich das Produktionssystem, beispielsweise durch Entfernen oder Hinzufügen einzelner Produktionsmittel oder kommt es zu einer Änderung im Produktionsprogramm, ist zu erwarten, dass andere Filterkombinationen erfolgreich sind. Durch das zufällige Erzeugen neuer Filterkombinationen und die erneute Bestimmung der besten Kombinationen passt sich das System an sich ändernde Rahmenbedingungen an. Somit leistet dieser Mechanismus einen Beitrag dazu, die Anforderungen einer automatischen Anpassung an verschiedene Steuerungsprobleme und Zielgrößen zu erfüllen.

4.3.4 Modifikationen der Rollout-Phase

Ausgehend von einem Zustand werden in der *Rollout*-Phase weitere zulässige Aktionen durchgeführt bis ein Terminalzustand erreicht wird. Die *Rollout*-Phase ähnelt einer wiederholt ausgeführten Expansion ausgehend vom jeweils entstandenen Zustand.

Die Auswahl der auszuführenden, zulässigen Aktionen kann zufällig erfolgen (*light rollout*) oder gemäß des Vorgehens zur intelligenten Einschränkung der Aktionen (*heavy rollout*), vorgestellt in Abschnitt 4.3.3.4.

4.3.5 Modifikationen der Backpropagation-Phase

Die *Backpropagation*-Phase schließt sich an das *Rollout* an. In dieser Phase werden die Bewertungen der Knoten auf dem Pfad, ausgehend vom im *Rollout* erreichten Terminalknoten bis zum Wurzelknoten angepasst. Die Fragestellung in dieser Phase des MO-MCTS-Verfahrens ist nicht die Bewertung des erreichten Terminalzustands, sondern wie das neue Ergebnisse des Terminalzustands mit den bereits aus früheren Iterationen erzielten Ergebnissen entlang des Pfads verrechnet werden. Im Folgenden werden Möglichkeiten der Aggregation der Bewertungen aus den einzelnen Iterationen für eine Zielgröße und anschließend für die Pareto-Optimierung vorgestellt.

4.3.5.1 Aggregation einer Zielgröße

Die folgenden Ansätze zur Aggregation einer Zielgröße können immer dann verwendet werden, wenn eine Kennzahl die Güte eines Zustands bemisst. Dies trifft auch zu, wenn das gewichtete Mittel aus mehreren Zielgrößen in der Zielfunktion genutzt wird.

Beim **arithmetischem Mittel** wird der Mittelwert aus den erreichten Zielfunktionswerten für jeden Knoten entlang des Pfads gebildet und als Bewertung des jeweiligen Knoten gespeichert. Der Mittelwert ist anfällig gegenüber Ausreißern und beinhaltet kein Streuungsmaß, siehe Formel 4.3.16. R_s bezeichnet die Menge der normierten Zielfunktionswerte der besuchten Zustände s .

$$Q(s) = \frac{\sum_{r \in R_s} r}{N(s)} \quad 4.3.16$$

Wird eine höhere Robustheit gegenüber Ausreißern benötigt, kann auf den **Median** ausgewichen werden, siehe Formel 4.3.17.

$$Q(s) = \begin{cases} x_{\frac{n+1}{2}}, & \text{wenn } n \text{ ungerade.} \\ 0,5x_{\frac{n}{2}} + 0,5x_{\frac{n}{2}+1}, & \text{wenn } n \text{ gerade.} \end{cases} \quad 4.3.17$$

Beim **maximalen Zielfunktionswert** erfolgt die Bewertung jedes Zustands anhand des besten bisher erreichten Zielfunktionswerts. Unter der Voraussetzung der Realisierung des

errechneten Belegungsplans ermöglicht dieses Maß, effizient die besten erreichten Lösungen zu wählen, siehe Formel 4.3.18.

$$Q(s) = \max_{r \in R^s} r \quad 4.3.18$$

4.3.5.2 Pareto-Optimierung

Bei einer multikriteriellen Optimierung mit Hilfe der Pareto-Lösungen kann in jedem Knoten die Menge der Pareto-Lösungen für die Kindknoten gespeichert werden. Im Rahmen der *Backpropagation* werden die Pareto-Mengen der Knoten ausgehend vom Blattknoten angepasst. Sobald ein Knoten erreicht wurde, dessen Pareto-Menge unverändert bleibt, kann der Vorgang abgebrochen werden, da die Pareto-Mengen der Elternknoten somit ebenfalls unverändert bleiben. Der Pareto-Ansatz ermöglicht es, nach der Optimierung dem Produktionsplaner die Menge aller nicht-dominierten Lösungen aufzuzeigen.

4.4 Parallelisierung des Suchbaumaufbaus

Parallelisierung im Zusammenhang mit Suchbäumen bezeichnet den parallelen Aufbau des Suchbaums. Einerseits wirkt sich die Parallelisierung auf die Suchgeschwindigkeit aus, gemessen in Iterationen pro Zeiteinheit, andererseits kann der Lösungsraum auch gleichzeitig in unterschiedlichen Bereichen aufgebaut werden.

Parallelisierung kann die Suchgeschwindigkeit nur linear beeinflussen. Dabei steigt die Suchgeschwindigkeit meist nur um weniger als den Faktor zwei bei einer Verdopplung der gleichzeitigen Operationen. Dies liegt an der Notwendigkeit zusätzlicher Rechenschritte zur Zusammenführung der Ergebnisse. Im Zusammenhang mit exponentiell wachsenden Lösungsräumen kann die Zunahme der Lösungsgeschwindigkeit nur einen untergeordneten Effekt haben. Neben der Beschleunigung der Suchgeschwindigkeit kann die Parallelisierung jedoch auch genutzt werden, um gleichzeitig mehrere Suchparameter zu evaluieren und somit schneller eine geeignete Wahl zu treffen. Dieser Effekt wirkt sich auf die Lösungsqualität aus. Im Folgenden werden drei verwendete Parallelisierungsansätze vorgestellt.

Subtree-Parallelisierung bezeichnet den parallelen Aufbau zweier voneinander getrennter Suchbäume ausgehend von einem gemeinsamen Wurzelknoten. In der Initialisierung des Verfahrens werden vom Wurzelknoten aus die zulässigen Aktionen auf mehrere parallele, unabhängige Prozesse aufgeteilt. In der Parallelisierungsphase bauen diese Prozesse bis

zum Erreichen eines Zeitlimits unabhängige und überschneidungsfreie Suchbäume auf. Dies wird durch die Teilung der Aktionsmengen ausgehend vom Wurzelknoten sichergestellt. Im Anschluss an den getrennten Baumaufbau findet die Konsolidierungsphase statt. Diese hat zum Ziel, die Bewertungen der Terminalzustände und Statistiken aus den Subbäumen auf Höhe des Wurzelknotens zu konsolidieren, um zu einem Gesamtergebnis zu gelangen. Das Vorgehen der *Subtree*-Parallelisierung hebt zu einem gewissen Grad die steuernde Wirkung der Selektionsfunktion aus. Durch die Trennung der Teilbäume werden alle Suchbäume parallel aufgebaut, obwohl es aus Sicht des Gesamtbaums sein könnte, dass bestimmte Teilbäume aufgrund der schlechten Lösungsgüte nicht besucht werden sollten.

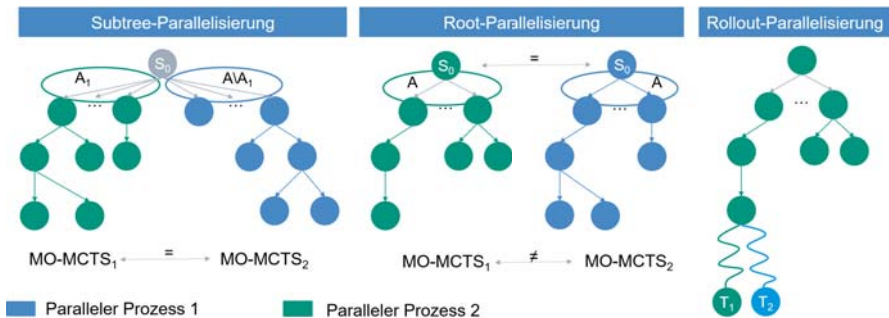


Abbildung 4.11: (links) *Subtree-Parallelisierung* mit identischen $MO - MCTS$ -Konfigurationen und geteilten Aktionsmengen. (Mitte) *Root-Parallelisierung* mit unterschiedlichen $MO - MCTS$ -Konfigurationen und identischen Aktionsmengen. (rechts) *Rollout-Parallelisierung*

Abbildung 4.11 zeigt links zwei identische $MO - MCTS$ -Konfigurationen $MO - MCTS_1$ und $MO - MCTS_2$, die jeweils auf einer überschneidungsfreien Teilmenge der Aktionen A_1 und $A \setminus A_1$ ausgehend vom Wurzelknoten zwei Teilbäume aufbauen.

Root-Parallelisierung steht für den parallelen Aufbau des gesamten Suchbaums ausgehend vom Wurzelknoten. Im Gegensatz zur *Subtree*-Parallelisierung wird die Aktionsmenge des Wurzelknotens nicht aufgeteilt. Allen unabhängigen Prozessen steht somit der gleiche Lösungsraum zur Verfügung. Die *Root*-Parallelisierung kann sinnvoll genutzt werden, um das gleiche Problem parallel mit unterschiedlichen Parametern zu optimieren. Bei der *Root*-Parallelisierung kann es dennoch sein, dass große Teile des Suchbaums mehrfach aufgebaut werden. Abbildung 4.11 zeigt in der Mitte zwei unterschiedliche $MO - MCTS$ -Konfigurationen $MO - MCTS_1$ und $MO - MCTS_2$, die jeweils ausgehend vom Wurzelknoten zwei Suchbäume aufbauen. Beide $MO - MCTS$ -Konfigurationen haben dabei die gesamte Aktionsmenge A ausgehend vom Wurzelknoten zur Verfügung.

Rollout-Parallelisierung bezeichnet die Durchführung paralleler *Rollouts* ausgehend vom erreichten Zustand nach der Expansion eines Knotens. Durch dieses Verfahren wird die Umgebung von einem Zustand aus schneller erkundet.

4.5 Hybridisierung durch Postoptimierung mit lokaler Suche

Abbildung 4.12 zeigt schematisch den Aufbau des hybriden Systems aus dem **Basisoptimierungsverfahren MO-MCTS** und der **Postoptimierung** basierend auf lokaler Suche.

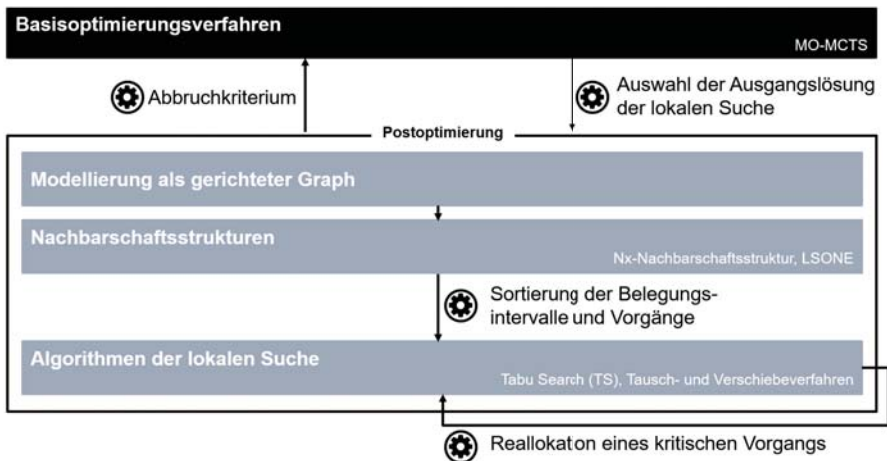


Abbildung 4.12: Schematischer Aufbau des hybriden Systems aus MCTS und lokaler Suche

MO-MCTS erzeugt pro Iteration einen neuen vollständigen Belegungsplan. Erste Gestaltungsmöglichkeit des hybriden Systems ist die **Auswahl der Ausgangslösung der lokalen Suche**. Die lokale Suche kann in den Ablauf des MO-MCTS-Optimierers integriert werden und Terminalzustände aus der *Rollout*-Phase optimieren oder am Ende des MO-MCTS-Laufs auf Pareto-Lösungen angewendet werden. Ist eine Ausgangslösung zur Postoptimierung ausgewählt, muss diese als **gerichteter Graph** modelliert werden. Anschließend wird anhand einer **Nachbarschaftsstruktur** die Nachbarschaft um die Ausgangslösung ermittelt. Hierfür stehen die Nachbarschaftsstrukturen aus Abschnitt 2.4.3 zur Verfügung. Um den Anwendungsfall der Matrixproduktion mit Rüst- und Transportzeiten abzubilden, muss die Nachbarschaftsstruktur LSONE angepasst werden. Diese benachbarten Lösungen bilden den Lösungsraum für die **Algorithmen der lokalen Suche**. Eine Gestaltungsmöglichkeit

ist die **Sortierung der Belegungsintervalle und kritischen Vorgänge**, die durch die Nachbarschaftsstruktur ermittelt werden. Diese Maßnahme steuert die Reihenfolge, in der die möglichen Optimierungsalternativen betrachtet werden. Die Algorithmen der lokalen Suche führen meist Tausch- und Verschiebeoperationen aus, bis keine Veränderungen mehr möglich sind oder bis ein anderes Kriterium erreicht wurde, beispielsweise eine Verbesserung der Ausgangslösung. Gestaltungsmöglichkeit in diesem Ablauf ist die **Reallokation der kritischen Vorgänge**. Bei der Reallokation werden für einen kritischen Vorgang mehrere Belegungsintervalle ausprobiert, bevor der Vorgang platziert wird und eine neue Iteration der lokalen Suche auf Basis des veränderten Belegungsplans gestartet wird. Als letzte Gestaltungsmöglichkeit bleibt die Wahl des **Abbruchkriteriums**. Dieses Kriterium legt fest, wann die lokale Suche terminiert und den möglicherweise verbesserten Belegungsplan an das Basisoptimierungsverfahren zurückgibt. Die Arbeiten basieren auf der vom Autor angeleiteten Abschlussarbeit (A_Liu 2021). Im Folgenden werden die einzelnen Bestandteile des hybriden Systems vorgestellt.

4.5.1 Auswahl der Ausgangslösungen für die lokale Suche

Das iterative Basisoptimierungsverfahren MO-MCTS erzeugt pro Iteration einen Terminalzustand. Dieser entspricht einem vollständigen Belegungsplan, der als Ausgangspunkt für eine lokale Suche genutzt werden könnte. Die Auswahl der Ausgangslösungen beschäftigt sich mit der Frage, nach welchen Kriterien die Ausgangslösungen identifiziert werden sollen. Prinzipiell ist die Durchführung einer lokalen Suche auf jedem erreichten Terminalzustand im Anschluss an die *Rollout*-Phase möglich. Dies führt jedoch dazu, dass Optimierungszeit auf schlechte Ausgangszustände verwendet wird. Eine Einschränkung der Ausgangslösungen ist daher sinnvoll. Folgende Möglichkeiten sollen untersucht werden:

1. **Sampling nach Zielfunktionswert:** Dieses Verfahren überprüft nach jeder Iteration, ob der Terminalzustand einen vorab definierten erforderlichen Mindestzielfunktionswert erreicht hat und führt dann eine lokale Suche aus. Für die i -te Iteration ist der Mindestwert durch $r_{init,i}$ gegeben. Eine Abwandlung dieser Methode sieht vor, dass diese untere Schranke mit der Anzahl an Iterationen um den Faktor k erhöht wird und durch eine obere Schranke r_{init}^{UB} beschränkt ist. Damit ergibt sich die Mindestanforderung an den Zielfunktionswert bei der Iteration i als $r_{min,i} = \min\{r_{init,0} + i \cdot k, r_{init}^{UB}\}$.
2. **Rollout-Sampling:** Nach jeder Iteration wird für den erreichten Terminalzustand mit einer bestimmten Wahrscheinlichkeit eine lokale Suche durchgeführt.

3. **Menge der besten Pareto-Lösungen je Iteration/Zeitintervall (BPI):** Dieses Verfahren unterbricht das MO-MCTS-Basisverfahren am Ende einer Iteration nach einer festen Ausführungszeit oder einer festen Anzahl Iterationen und führt eine lokale Suche für die besten Pareto-Lösungen aus.
4. **Menge der besten Pareto-Lösungen am Ende der Ausführung (BPA):** Dieses Verfahren führt eine lokale Suche am Ende der MO-MCTS-Exekution für die besten Pareto-Zustände aus.

4.5.2 Modellierung der Ausgangslösung als gerichteter Graph

Viele Verfahren der lokalen Suche für Produktionssteuerungsaufgaben nutzen gerichtete Graphen als Repräsentation der Lösungen, siehe Abschnitt 2.4.2. Für die Abbildung von Transport- und Rüstzeiten muss diese Modellierung angepasst werden. Zusätzlich zu den Reihenfolgebeziehungen innerhalb eines Produktionsauftrags beziehungsweise auf einem Produktionsmittel, siehe Abschnitt 2.4.2, werden Rüst- und Transportvorgänge aufgenommen. Die Rüstvorgänge sind ein weiterer Vorgangstyp und stehen in einer Reihenfolgebeziehung mit den Bearbeitungsvorgängen auf einem Produktionsmittel. Die Transportvorgänge aber stehen in einer Reihenfolgebeziehung mit den Bearbeitungsvorgängen eines Produktionsauftrags. Ist der Graph azyklisch, ist die Lösung zulässig.

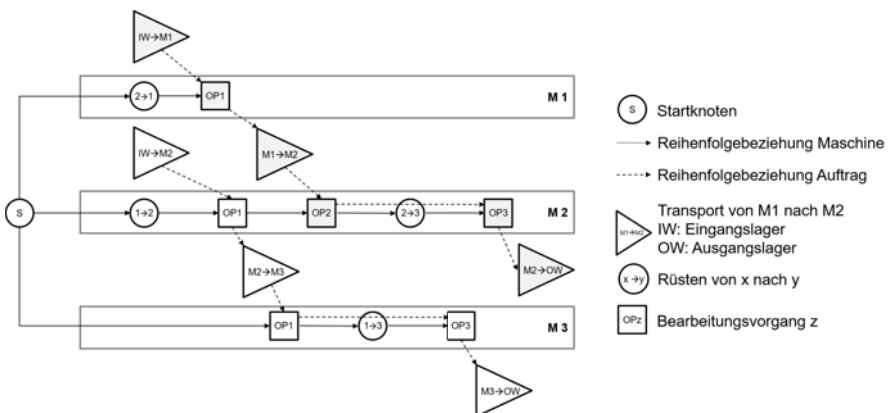


Abbildung 4.13: Darstellung des gerichteten Graphen für ein fiktives Beispiel aus drei Maschinen und zwei Produktionsaufträgen mit Transport- und Rüstvorgängen

Abbildung 4.13 zeigt die Datenstruktur anhand eines Beispiels bestehend aus drei Produktionsmitteln, drei Vorgängen und zwei Produktionsaufträgen. Die gestrichelten Linien verbinden die Vorgänge aus Sicht der Produktionsaufträge, also die Transport- und Bearbeitungsvorgänge. Die durchgezogenen Linien verbinden die Vorgänge aus Perspektive der Produktionsmittel, also die Rüst- und Bearbeitungsvorgänge.

4.5.3 Anpassung der Nachbarschaftsstruktur LSONE

Das betrachtete Steuerungsproblem der Matrixproduktion erfordert eine Anpassung der LSONE-Nachbarschaftsstruktur, die in Abschnitt 2.4.3 vorgestellt wurde. Um Rüst- und Transportzeiten zu berücksichtigen, können diese analog zu Bearbeitungsvorgängen als Vorgang modelliert werden. Abschnitt 4.5.2 zeigt den gerichteten Graph, in den Rüst- und Transportvorgänge integriert sind.

Zur Bestimmung der Nachbarschaft einer Ausgangslösung muss zuerst der kritische Pfad ermittelt werden. Die Identifikation des kritischen Pfads muss aufgrund der Rüst- und Transportzeiten angepasst werden. Anschließend müssen für die Vorgänge des kritischen Pfads mögliche Belegungsintervalle identifiziert werden. Die Nachbarschaftsstruktur erzeugt somit eine Menge kritischer Vorgänge und eine Menge möglicher Belegungsintervalle. Im Folgenden wird zuerst auf die Identifikation der kritischen Vorgänge und anschließend auf die Belegungsintervalle eingegangen.

4.5.3.1 Bestimmung des kritischen Pfads

LSONE berechnet den kritischen Pfad anhand der Pufferzeiten. Da die Rüst- und Transportvorgänge durch die Integration als Vorgang im gerichteten Graph erfasst werden, kann die Definition zur Bestimmung des kritischen Pfads nach Gao & Sun u. a. (2008) beibehalten werden. Erster Schritt ist die Bestimmung des frühesten und der spätesten Startzeitpunktes für jeden Vorgang (Rüst-, Transport- oder Bearbeitungsvorgang). Ein Vorgang gilt als kritisch, wenn beide Zeitpunkte identisch sind.

Der frühe Startzeitpunkt $s^E(v)$ bezeichnet den frühesten Zeitpunkt, zu dem der Vorgang v begonnen werden darf. Der spätest mögliche Startzeitpunkt $s^L(v)$ ist der letzte Zeitpunkt zu dem der Vorgang v begonnen werden kann, ohne dass es zu einer Verspätung der nachfolgenden Vorgänge kommt.

$$c^E(v) = s^E(v) + t_v$$

4.5.1

Formel 4.5.1 gibt den frühesten Fertigstellungszeitpunkt $c^E(v)$ für die Dauer t_v des Vorgangs an. Diese Dauer kann eine Bearbeitungs-, Transport- oder Rüstdauer bezeichnen. Formel 4.5.2 zeigt die analoge Berechnung für den spätesten Fertigstellungszeitpunkt $c^L(v)$. Der Ausgangspunkt für die Berechnung der spätesten Fertigstellungszeitpunkte und damit auch der spätesten Startzeitpunkte bildet das Ende des letzten Vorgangs im gerichteten Graphen. Dieser determiniert den Makespan.

$$c^L(v) = s^L(v) + t_v \quad 4.5.2$$

Der früheste Startzeitpunkt eines Vorgangs v ergibt sich aus dem Maximum des frühesten Fertigstellungszeitpunkts des vorangehenden Vorgangs dieses Auftrags $c^E(PJ(v))$ und des frühesten Fertigstellungszeitpunkts des vorangegangenen Vorgangs des Produktionsmittels $c^E(PM(v))$, siehe Formel 4.5.3. Der vorangegangene Vorgang des Produktionsauftrags kann auch ein Transportvorgang sein, der Vorgänger auf dem Produktionsmittel ein Rüstvorgang.

$$s^E(v) = \max\{c^E(PJ(v)), c^E(PM(v))\} \quad 4.5.3$$

4.5.3.2 Bestimmung möglicher Belegungsintervalle

Nach der Bestimmung des kritischen Pfads wird ein kritischer Vorgang aus dem Belegungsplan gelöscht, sodass der reduzierte Graph G^- entsteht. Im Gegensatz zum Grundalgorithmus müssen unter Umständen Rüst- und Transportvorgänge entfernt werden. Die Identifikation möglicher neuer Intervalle muss angepasst werden, um benötigte Transport- und Rüstvorgänge zu berücksichtigen. Im Folgenden werden die Regeln zur Identifikation eines Intervalls vorgestellt.

Der Vorgang v kann in dem Intervall zwischen dem Vorgang u und dem Vorgänger von u platziert werden, wenn das Intervall zwischen der Vorgängeroperation von u und dem spätesten Startzeitpunkt von u größer als die Bearbeitungsdauer von v ist (Gao & Sun u. a. 2008). Diese Anforderung stellt die Formel 4.5.4 dar. Zum einen ist der früheste Startzeitpunkt von v beschränkt durch den frühesten Endzeitpunkt der Vorgängeroperation von u auf dem Produktionsmittel, gegeben durch $c^{E-}(PM(u))$. Dies kann auch ein Rüstvorgang sein. Zum anderen ist der früheste Startzeitpunkt von v beschränkt durch den frühesten Endzeitpunkt der Vorgängeroperation des Produktionsauftrags $c^{E-}(PJ(v))$. Dies kann auch ein Transportvorgang sein. Der früheste Endzeitpunkt des Vorgangs v ergibt sich aus dem frühesten Startzeitpunkt zuzüglich der Bearbeitungszeit $t_{m_u,v}$ des Vorgangs v auf Produktionsmittel m_u .

Dieser Zeitpunkt muss vor dem spätesten Startzeitpunkt des Vorgangs u liegen, gegeben durch $s^{L^-}(u)$, und vor dem spätesten Start des Nachfolgevorgangs des Vorgangs v , gegeben durch $s^{L^-}(SJ(v))$.

$$\max\{c^{E^-}(PM(u)), c^{E^-}(PJ(v))\} + t_{m_u, v} < \min\{s^{L^-}(u), s^{L^-}(SJ(v))\} \quad 4.5.4$$

Die Formel 4.5.4 ist anzupassen, wenn durch Hinzufügen des Vorgangs v ein Transport- oder Rüstvorgang nötig oder hinfällig wird. Wenn $PJ(v)$ ein Bearbeitungsvorgang auf einem Produktionsmittel m_v ungleich m_u ist, muss ein Transportvorgang von m_v nach m_u hinzugefügt werden. Um dies abzubilden, wird $PJ(v)$ um die Dauer des Transports $t_t(m_v, m_u)$ erhöht.

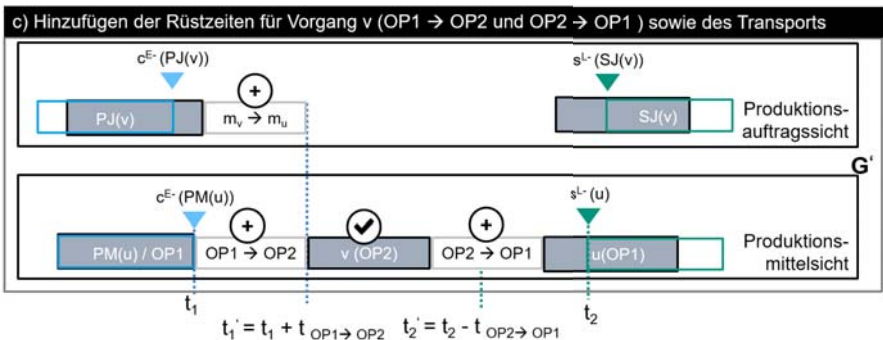
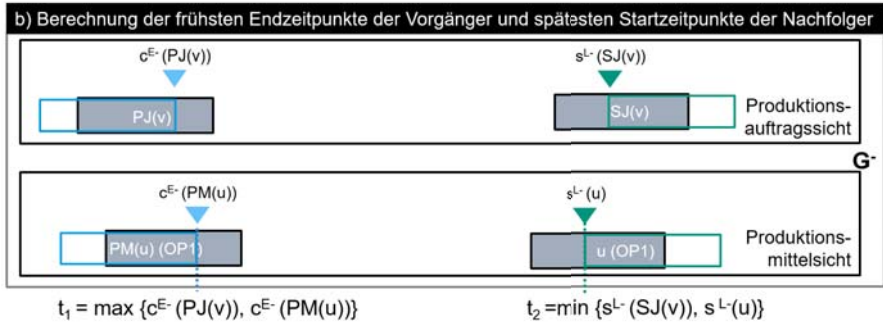
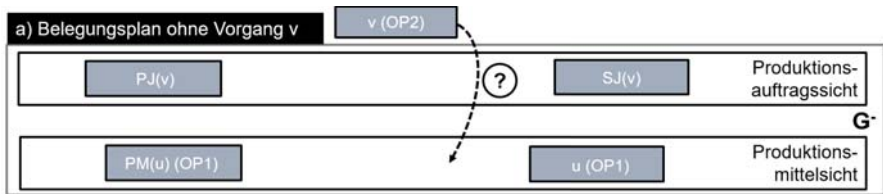
Analog dazu wird $SJ(v)$ angepasst, wenn ein Transportvorgang zum Produktionsmittel erforderlich ist, das den nachfolgenden Vorgang $SJ(v)$ durchführt. In diesem Fall muss der späteste Startzeitpunkt um die Transportdauer vorgezogen werden.

Wenn der vorangehende Vorgang $PJ(v)$ ein Transportvorgang von m_x nach m_v ist und m_x und m_u dasselbe Produktionsmittel darstellen, dann kann dieser Transportvorgang gelöscht werden, da sich der Auftrag bereits an dem Produktionsmittel m_u befindet. Da der vorangehende Transportvorgang gelöscht wurde, ist der früheste Endzeitpunkt durch den Vorvorgänger gegeben $c^{E^-}(PJ(PJ(v)))$.

Analog dazu müssen nachfolgende Transportvorgänge gelöscht werden, wenn das Transportziel Produktionsmittel m_u ist, da sich auch in diesem Fall der Produktionsauftrag bereits an dem richtigen Produktionsmittel befindet. Damit ergibt sich $s^{L^-}(SJ(SJ(v)))$.

Die Rüstvorgänge verhalten sich analog zu den Transportvorgängen. Ist ein Umrüsten zwischen u und v nötig, muss der früheste Fertigstellungszeitpunkt des vorangehenden Vorgangs $c^{E^-}(PM(u))$ um die Rüstzeit verlängert werden. Ist durch das Hinzufügen von v kein Umrüsten von $PM(u)$ nötig, kann der Rüstvorgang entfernt werden und der früheste Fertigstellungszeitpunkt wird durch die Vorvorgängeroperation bestimmt. Genauso verhält es sich, wenn kein Umrüsten von v nach u benötigt wird, sodass dieser Vorgang gelöscht werden kann.

Abbildung 4.14 zeigt die Überprüfung des Intervalls exemplarisch für den Fall, bei dem zwei zusätzliche Rüstvorgänge erforderlich sind und ein weiterer Transportvorgang. Ausgangspunkt bildet der reduzierte Graph G^- ohne den Vorgang v . Es soll ermittelt werden, ob der Vorgang v zwischen den Vorgänger des Vorgangs u , gegeben durch $PM(u)$, und den Vorgang u platziert werden kann, siehe Teil a) der Abbildung. Als erster Schritt werden die frühesten Endzeitpunkte der Vorgänger, also $c^{E^-}(PJ(v))$ und $c^{E^-}(PM(u))$, sowie die



Legende:

- $m_v \rightarrow m_u$: Transport vom Produktionsmittel des Vorgangs v zum Produktionsmittel des Vorgangs u
- OP1 → OP2: Rüstvorgang von OP1 auf OP2

Abbildung 4.14: Schematische Darstellung der Ermittlung des geeigneten Intervalls für den Vorgang v

spätesten Startzeitpunkte der nachfolgenden Vorgänge, $s^{L-}(SJ(v))$ und $s^{L-}(u)$, ermittelt. Im dargestellten Fall wird der Anfang des möglichen Belegungsintervalls für Vorgang v durch t_1 , hier den frühesten Endzeitpunkt von $PM(u)$, limitiert. Das Ende des möglichen Belegungsintervalls wird durch t_2 beschränkt, bestimmt durch den spätesten Startzeitpunkt des Vorgangs

u . Dieser Schritt wird in Teil b) der Abbildung dargestellt. Durch das Einfügen des Vorgangs v werden jedoch weitere Rüst- und Transportvorgänge benötigt. Das Produktionsmittel muss für die Durchführung des Vorgangs v von $OP1$ auf $OP2$ umgerüstet werden und nach dem Bearbeitungsvorgang v wieder für $OP2$ vorbereitet werden, da der nachfolgende Vorgang u den Prozess $OP2$ benötigt. Zusätzlich wird ein Transportvorgang vom Produktionsmittel des Vorgangs $PJ(v)$ zum Produktionsmittel des Vorgangs u benötigt. Um die neuen Vorgänge in die Überprüfung des Belegungsintervalls einzubeziehen, müssen die Grenzen t_1 und t_2 angepasst werden. Die neue untere Grenze t'_1 berücksichtigt den Rüstvorgang von $OP1 \rightarrow OP2$ ⁴. Die neue obere Grenze t'_2 enthält den Rüstvorgang von $OP2 \rightarrow OP1$. Die Überprüfung des resultierenden Intervalls ergibt, dass der Vorgang v eingeplant werden kann. Der Vorgang $PM(u)$ wird gemäß des frühesten Endzeitpunkts eingeplant und die neuen Rüst- und Transportvorgänge werden passend zu dem Bearbeitungsvorgang v hinzugefügt. Teil c) der Abbildung stellt den neuen vollständigen Belegungsplan G' dar.

4.5.3.3 Sortierung kritischer Vorgänge und Belegungsintervalle

Abhängig von der Nachbarschaftsstruktur kann die Nachbarschaft einer Ausgangslösung sehr groß sein. Durch eine Sortierung sowohl der kritischen Vorgänge als auch der möglichen Belegungsintervalle kann die lokale Suche beeinflusst werden. Zum einen kann erreicht werden, dass vielversprechende Tausch- und Verschiebeoperationen zuerst ausprobiert werden, zum anderen können durch eine entsprechende Sortierung bestimmte Zielgrößen gefördert werden. Dieses Vorgehen ist angelehnt an die Bevorzugung kurzer Bearbeitungsdauern in der lokalen Suche von Ho & Tay (2008) sowie die Vorsortierung der Produktionsmittel entsprechend der Bearbeitungszeiten von Xia & Wu (2005). Die kritischen Vorgänge können nach folgenden Kriterien sortiert werden:

1. **Dauer:** Absteigend nach ihrer Dauer
2. **Früherer Startzeitpunkt:** Absteigend nach der Startzeit
3. **Spätester Endzeitpunkt des Vorgangs:** Absteigend nach dem derzeit prognostizierten Fertigstellungszeitpunkt
4. **Arbeitsbelastung:** Absteigend gemäß der Gesamtbearbeitungsdauer des Produktionsmittels

⁴Es handelt sich um eine vereinfachte Darstellung. Es muss zusätzlich geprüft werden, ob die zusätzliche Transportoperation den Anfang des Intervalls verschiebt.

5. **Rüstvorgänge:** Absteigend nach der Anzahl benachbarter Rüstvorgänge. Durch die Reallokation können Rüstvorgänge wegfallen.
6. **Transportvorgänge:** Absteigend gemäß der Anzahl benachbarter Transportvorgänge. Durch die Reallokation können Transporte entfallen.
7. **Verhältnis Rüst- zu Bearbeitungsvorgängen:** Absteigend gemäß des Verhältnisses der Rüstvorgängen zu den Bearbeitungsvorgängen auf dem Produktionsmittel
8. **Verspätung:** Absteigend nach der Verspätung des dazugehörigen Produktionsauftrags
9. **Geforderter Fertigstellungszeitpunkt des Auftrags:** Absteigend gemäß des geforderten spätesten Fertigstellungszeitpunkts des dazugehörigen Produktionsauftrags
10. **Prognostizierter Fertigstellungszeitpunkt des Auftrags:** Absteigend gemäß des prognostizierten Fertigstellungszeitpunkt des dazugehörigen Produktionsauftrags
11. **Referenzdauer:** Absteigend gemäß der Differenz zur kürzest möglichen Dauer dieser Vorgänge abhängig vom Produktionsmittel
12. **Wartezeit:** Absteigend gemäß der Differenz aus der aktuellen Startzeit und dem Fertigstellungszeitpunkt des vorangegangenen Vorgangs

Belegungsintervalle können nach folgenden Kriterien sortiert werden:

1. **Startzeitpunkt:** Aufsteigend nach der frühesten Startzeit
2. **Abstand zum frühesten Startzeitpunkt:** Aufsteigend nach der Differenz aus frühestem Startzeitpunkt des kritischen Vorgangs und dem Startzeitpunkt des Belegungsintervalles. Damit werden Belegungsintervalle bevorzugt, die nah am frühesten Startzeitpunkt des kritischen Vorgangs liegen.
3. **Verhältnis Rüst- zu Bearbeitungsvorgängen:** Aufsteigend nach dem Verhältnis aus Rüst- und Bearbeitungsvorgängen
4. **Arbeitsbelastung:** Aufsteigend nach der Arbeitsbelastung des Produktionsmittels
5. **Rüstvorgänge:** Vorrangig, wenn nach dem Einfügen kein Rüsten erforderlich ist
6. **Transportvorgänge:** Vorrangig, wenn nach dem Einfügen kein Transportvorgang erforderlich ist

4.5.4 Algorithmen der lokalen Suche

Die Nachbarschaftsstruktur gibt den Lösungsraum für die lokale Suche vor. Im Folgenden werden die auf Verschiebe-, Tausch- und der LSONE-Nachbarschaftsstruktur basierenden Verfahren vorgestellt, die untersucht werden sollen.

4.5.4.1 Lokale Suche mit Tabu-Suche und Tauschoperationen

Tabu-Suche (Glover 1989; Glover 1990) ist ein etabliertes Verfahren der lokalen Suche. Das Führen einer Liste verbotener Züge (Tabu-Liste) verhindert oszillierende Zustände, die beispielsweise durch Vorwärts- und anschließenden Rückwärtstausch auftreten würden. Die Tabu-Suche wählt den besten nicht verbotenen Zug aus der Nachbarschaft einer Ausgangslösung aus. Es kann mit den tauschbasierten Nachbarschaftsstrukturen N1, NA, N4 und N6 kombiniert werden.

Algorithmus 2 zeigt vereinfacht das Vorgehen der lokalen Suche in Kombination mit Tabu-Suche. Im ersten Schritt wird der Ausgangszustand von MO-MCTS in die Datenstruktur der lokalen Suche überführt (2), anschließend wird die zu verwendende Nachbarschaftsstruktur (N1, NA, N4, N6) ausgewählt (3) und die Tabu-Liste (4) sowie die Liste der Lösungen (5) werden initialisiert.

Solange das Abbruchkriterium nicht erreicht ist, werden der kritische Pfad bestimmt (7) und alle möglichen Tauschoperationen ermittelt, abhängig vom derzeitigen Zustand des kritischen Pfades und der gewählten Nachbarschaftsstruktur (8). Für alle möglichen Tauschoperationen werden die Zielfunktionswerte errechnet, die erreicht werden können, wenn man die Tauschoperation im derzeitigen Zustand anwendet (11) und das Tupel aus Zielfunktionswert oder Zielfunktionsvektor und Tauschoperation in einer Liste abgespeichert (12). Nachdem alle Züge evaluiert wurden, wird die Liste der Tupels absteigend nach dem Zielfunktionswert sortiert (14). Für jedes der Tupel wird geprüft, ob die Tauschoperation "tabu" ist (16). Wenn die Tauschoperation "tabu" ist, wird geprüft, ob das Akzeptanzkriterium dennoch erfüllt ist (18). Als Akzeptanzkriterium wird der Vergleich zur bislang besten bekannten Lösung verwendet. Sollte für das Tupel das Akzeptanzkriterium erfüllt oder die Tauschoperation nicht "tabu" sein, wird die Tauschoperation zur Tabuliste hinzugefügt. Die älteste Tauschoperation wird aus der Tabu-Liste entfernt (21) und die Tauschoperation wird als gewählter Zug verwendet (22). An dieser Stelle werden keine weiteren Tupel evaluiert (23). Da die Liste geordnet ist, kann keine größere Verbesserung erzielt werden. Der Zustand wird auf Basis des gewählten Zugs aktualisiert (26), die Lösung wird gespeichert (27) und das Abbruchkriterium geprüft (28). Das Abbruchkriterium kann iterations- oder zeitbasiert sein. Sind keine Tauschoperationen

mehr möglich, bricht das Verfahren ebenfalls ab. Die Schritte 7-28 werden wiederholt, bis das Abbruchkriterium erfüllt ist.

Algorithmus 2 Tabu-Suche

```

1: procedure TABU-SUCHE
2:    $Graph \leftarrow \text{erstelleGraph}(\text{Ausgangslösung})$  ▷ Abschnitt 4.5.2
3:    $ns \leftarrow \text{wähleNachbarschaftsstruktur}(Graph)$ 
4:    $Tabu\text{-}Liste \leftarrow \emptyset$ 
5:    $Ergebnis \leftarrow \emptyset$ 
6:   while not abgeschlossen do
7:      $cp \leftarrow \text{identifiziere kritischen Pfad}(Graph)$  ▷ Abschnitt 2.4.1
8:      $möglicheTauschoperationen \leftarrow \text{möglicheTauschoperationen}(Graph, cp, ns)$ 
9:      $ZielerreichungTauschTupel \leftarrow \emptyset$ 
10:    for all  $Tausch \in möglicheTauschoperationen$  do
11:       $(Zielerreichung, Tausch) \leftarrow \text{führeAusWennZulässig}(Graph, Tausch)$ 
12:       $ZielerreichungTauschTupel.add((Zielerreichung, Tausch))$ 
13:    end for
14:     $ZielerreichungTauschTupel.sortiereAbsteigendNachZielerreichung()$ 
15:    for all  $Tupel \in ZielerreichungTauschTupel$  do
16:       $tabu \leftarrow \text{prüfeTabuStatus}(tupel.Tausch)$ 
17:      if tabu then
18:         $Akzeptanz \leftarrow \text{prüfeAkzeptanzkriterium}(tupel.Zielerreichung)$ 
19:      end if
20:      if not tabu or Akzeptanz then
21:         $Tabu\text{-}Liste \leftarrow \text{updateTabuListe}(tupel.Tausch)$ 
22:         $gewählterTausch \leftarrow tupel.Tausch$ 
23:        break
24:      end if
25:    end for
26:     $Graph \leftarrow \text{führeTauschAus}(gewählterTausch)$ 
27:     $Ergebnis \leftarrow \text{updateParetoFront}(Graph)$ 
28:     $abgeschlossen \leftarrow \text{prüfeAbbruchkriterium}()$ 
29:  end while
30:  return Ergebnis
31: end procedure

```

4.5.4.2 Lokale Suche mit Tauschoperationen

Der nachfolgende Algorithmus beruht ausschließlich auf Tauschoperationen und kann mit den Nachbarschaftsstrukturen N4, NA und N6 kombiniert werden. Algorithmus 3 zeigt den Ablauf des generischen Verfahrens.

Im ersten Schritt wird der MO-MCTS-Zustand in die Datenstruktur der lokalen Suche gewandelt (1). Anschließend wird die zu verwendende Nachbarschaftsstruktur festgelegt (2) und die Lösungsmenge initialisiert (3).

Solange das Abbruchkriterium nicht erreicht ist, wird für die aktuelle Ausgangslösung der kritische Pfad bestimmt (6). Aufbauend auf dem Zustand und dem kritischen Pfad werden alle möglichen Tauschoperationen abhängig von der Nachbarschaftsstruktur ermittelt (7).

Anschließend wird für jede Tauschoperationen der Tausch durchgeführt (9), die Zielfunktionswerte bestimmt (10) und geprüft, ob die Lösungsgüte den Mindestanforderungen entspricht (11). Ist dies nicht der Fall, wird der Ausgangszustand über einen *Rollback* wiederhergestellt (13). Erfüllt die neue Lösung die Gütekriterien, wird sie gespeichert (15) und die Suche ausgehend von diesem Ausgangszustand endet (16).

Solange das Abbruchkriterium (19) nicht erfüllt ist, werden die Schritte 6-19 wiederholt.

Algorithmus 3 Generische lokale Suche mit Tauschoperationen

```

1: procedure LOKALE_SUCHE_MIT_TAUSCHOPERATIONEN
2:   Graph ← erstelleGraph(Ausgangslösung)                                ▷ Abschnitt 4.5.2
3:   ns ← wähleNachbarschaftsstruktur(Graph)
4:   Ergebnis ←  $\emptyset$ 
5:   while not abgeschlossen do
6:     cp ← identifiziereKritischenPfad(Graph)                            ▷ Abschnitt 2.4.1
7:     möglicheOperationen ← möglicheTauschoperationen(Graph, cp, ns)
8:     for all Tausch ∈ möglicheOperationen do
9:       Graph ← führeAusWennZulässig(Tausch)
10:      Zielfunktionswert ← bewerte(Graph)
11:      tolerierbar ← prüfeRollbackkriterium(Zielfunktionswert)        ▷ Abschnitt 4.5.5
12:      if not tolerierbar then
13:        rollback(Graph)
14:      else
15:        Ergebnis ← updateParetoFront(Graph)
16:        break
17:      end if
18:    end for
19:    abgeschlossen ← prüfeAbbruchkriterium()                            ▷ Abschnitt 4.5.6
20:  end while
21:  return Ergebnis
22: end procedure

```

4.5.4.3 Lokale Suche mit LSONE

Im Folgenden wird der entwickelte Suchalgorithmus vorgestellt, der die um Rüst- und Transportvorgänge erweiterte LSONE-Nachbarschaftsstruktur aus Abschnitt 4.5.3 nutzt. Zudem wird der Ansatz mit dem ursprünglichen Verfahren von Gao & Sun u. a. (2008) verglichen.

Die lokale Suche, basierend auf LSONE bei Gao & Sun u. a. (2008), optimiert die Chromosomen der Population des verwendeten genetischen Algorithmus. Zum Einsatz kommt das VND-Verfahren, das durch Nutzung mehrerer Nachbarschaftsstrukturen lokale Optima zu vermeiden versucht. Die lokale Suche bestimmt für jeden kritischen Vorgang genau ein mögliches Belegungsintervall und verschiebt den kritischen Vorgang in dieses Belegungsintervall. Sind keine weiteren Verschiebungen mehr möglich, kommt die zweite Nachbarschaftsstruktur zum Einsatz, die zwei Vorgänge simultan verschiebt.

Das Verfahren nach Gao & Sun u. a. (2008) wurde im Rahmen dieser Arbeit angepasst. Anstelle der Optimierung aller Lösungen werden entweder Terminalzustände am Ende der *Rollout*-Phase oder Lösungen aus der Pareto-Menge als Ausgangslösung genutzt, siehe Abschnitt 4.5.1. Ein weiterer Unterschied besteht darin, dass für jeden kritischen Vorgang alle Belegungsintervalle bestimmt werden, um eine gründlichere Suche zu ermöglichen. Ist das Ergebnis durch die Verschiebung eines kritischen Vorgangs nicht zufriedenstellend, wird ein weiteres Intervall evaluiert, siehe Abschnitt 4.5.5. Im Gegensatz zum Ausgangsverfahren werden zudem mehrere Verbesserungsiterationen durchgeführt. Die erzielte Lösung stellt somit die nächste Ausgangslösung für eine weitere Iteration der lokalen Suche dar. Durch diese Anpassungen des Algorithmus kann eine größere Nachbarschaft untersucht werden.

Algorithmus 4 zeigt schematisch den Ablauf der lokalen Suche. Im ersten Schritt wird der Zustand von MO-MCTS in die Repräsentation zur lokalen Suche überführt (2) und die Lösungsmenge initialisiert (3). Solange das Abbruchkriterium nicht erreicht ist, werden die kritischen Vorgänge bestimmt (5) und anschließend vorsortiert (6). Für alle kritischen Vorgänge wird jeweils der reduzierte Graph G^- gebildet (8). Danach werden mögliche Intervalle für die kritischen Vorgänge ermittelt (9) und vorsortiert (10). Für die vorsortierten Intervalle wird jeweils der vervollständigte Graph G' gebildet (12)(13) und evaluiert (14). Anschließend erfolgt die Prüfung, ob das Ergebnis die Güteanforderungen erfüllt (15). Ist dies nicht der Fall, wird die Ausgangslösung wieder hergestellt und der Vorgang erneut auf ein anderes Belegungsintervall allokiert. Andernfalls wird das Ergebnis in die Lösungsmenge aufgenommen und anschließend der nächste kritische Vorgang betrachtet. Die Schritte 5 bis 24 werden wiederholt, solange das Abbruchkriterium nicht erreicht ist (24).

Algorithmus 4 Generische lokale Suche mit LSONE

```

1: procedure LOKALE_SUCHE_MIT_LSONE
2:    $Graph \leftarrow \text{erstelleGraph}(\text{Ausgangslösung})$  ▷ Abschnitt 4.5.2
3:    $Ergebnis \leftarrow \emptyset$ 
4:   while not abgeschlossen do
5:      $kritischeVorgänge \leftarrow \text{identifiziereKritischeVorgänge}(Graph)$ 
6:      $kritischeVorgänge \leftarrow \text{sortiere}(Graph, kritischeVorgänge)$  ▷ Abschnitt 4.5.3.3
7:     for all  $kritischerVorgang \in kritischeVorgänge$  do
8:        $G^- \leftarrow \text{entferne}(kritischerVorgang)$ 
9:        $Belegungsintervalle \leftarrow \text{identifiziereIntervalle}(G^-)$  ▷ Abschnitt 4.5.3
10:       $Intervalle \leftarrow \text{sortiere}(G^-, Belegungsintervalle)$  ▷ Abschnitt 4.5.3.3
11:      for all  $Intervall \in Intervalle$  do ▷ Abschnitt 4.5.3.3
12:         $Tausch \leftarrow \text{identifiziereTausch}(kritischerVorgang, Intervall)$ 
13:         $G' \leftarrow \text{führeAus}(Tausch)$ 
14:         $Zielerreichung \leftarrow \text{bewerte}(G')$ 
15:         $tolerierbar \leftarrow \text{prüfeRollbackkriterium}(Zielerreichung)$  ▷ Abschnitt 4.5.5
16:        if not tolerierbar then
17:          rollback( $G'$ )
18:        else
19:           $Ergebnis \leftarrow \text{updateParetoFront}(G')$ 
20:          break
21:        end if
22:      end for
23:    end for
24:     $abgeschlossen \leftarrow \text{prüfeAbbruchkriterium}()$  ▷ Abschnitt 4.5.6
25:  end while
26:  return  $Ergebnis$ 
27: end procedure

```

4.5.5 Reallokation eines Vorgangs

Eine weitere Gestaltungsmöglichkeit des hybriden System ist die Reallokation von kritischen Vorgängen in ein alternatives Belegungsintervall innerhalb einer Iteration der lokalen Suche. Für einen kritischen Vorgang können unterschiedliche kritische Belegungsintervalle zur Verfügung stehen. Die lokale Suche muss ein Belegungsintervall für den zu verschiebenden Vorgang auswählen. Es besteht daher die Gefahr, dass die lokale Suche eine ungünstige Auswahl trifft, die zu einer Verschlechterung führt, die auch in den nächsten Iterationen nicht mehr ausgeglichen werden kann.

Der Ansatz der Reallokation adressiert dieses Problem. Die Reallokation eines Vorgangs ermöglicht es, unterschiedliche Belegungsintervalle für einen kritischen Vorgang zu erproben. Erst wenn das erzielte Ergebnis den Anforderungen entspricht, wird eine neue Iteration auf der

erzielten Lösung gestartet. Werden die Anforderungen nicht erfüllt, wird der Ausgangszustand über einen *Rollback* wiederhergestellt. Ist die maximale Anzahl an Reallokationen erreicht, fährt das Verfahren mit dem nächsten kritischen Vorgang fort.

Damit die lokale Suche auch lokale Optima überbrücken kann, sollten vorübergehende Verschlechterungen toleriert werden. Im Folgenden werden Definition tolerierbarer Verschlechterungen vorgestellt.

1. **Konstante Toleranz** t_c : Die Verbesserung der aktuellen Zielerreichung zur Referenzlösung G^{ref} ist gegeben durch $\Delta = Q(G^i) - Q(G^{ref})$. Eine Lösung ist toleriert, wenn $\Delta \geq t_c$.
2. **Abnehmende Toleranz** t_d : Die Toleranz nimmt mit einem konstanten Faktor mit der Anzahl an Durchläufen der lokalen Suche ab.
3. **Simulated Annealing**: Eine Lösung mit negativer Verbesserung Δ wird mit einer abnehmenden Wahrscheinlichkeit toleriert. Die Wahrscheinlichkeit nimmt nicht-linear ab, angelehnt an das *Simulated Annealing* Verfahren. Die Wahrscheinlichkeit ergibt sich als $\exp(\frac{\Delta}{T})$, wobei T die aktuelle Temperatur widerspiegelt, die linear mit den Iterationen der lokalen Suche fällt.
4. **Pareto-Kriterium**: Wenn eine neue Pareto-Lösung gefunden wird, gilt diese als toleriert, auch wenn der gewichtete Zielfunktionswert $Q(G^n)$ geringer ist als von der Referenzlösung vorgegeben.

4.5.6 Abbruchkriterium der lokalen Suche

Eine weitere Gestaltungsmöglichkeit des hybriden Systems bildet das Abbruchkriterium der lokalen Suche. Ist dieses Kriterium erreicht, wird die Lösung an das Basisoptimierungsverfahren zurückgegeben. Zwei Abbruchkriterien sind im Zusammenhang mit lokaler Suche verbreitet.

1. **First Improvement** bricht ab, sobald die erste Verbesserung in der Nachbarschaft einer Ausgangslösung erzielt wurde. Diese Lösung kann als neue Ausgangslösung für eine weitere lokale Suche genutzt werden. *First Improvement* beschleunigt den Suchprozess stark und wird unter anderem von Gao & Sun u. a. (2008) sowie Hansen & Mladenović u. a. (2010) eingesetzt.

2. **Best Improvement** bezeichnet die vollständige Suche der Nachbarschaft einer Ausgangslösung. Ergebnis ist die größte gefundene Verbesserung. *Best improvement* kann eingeschränkt werden, indem die Gesamtzahl an Tausch- bzw. Schiebeoperationen limitiert wird. Somit wird auch bei einer großen Nachbarschaft sichergestellt, dass die lokale Suche ausgehend von einem Terminalzustand beschränkt bleibt.

4.6 Prototypische Umsetzung als Software-Demonstrator

Abbildung 4.15 zeigt schematisch den Aufbau des Software-Demonstrators.

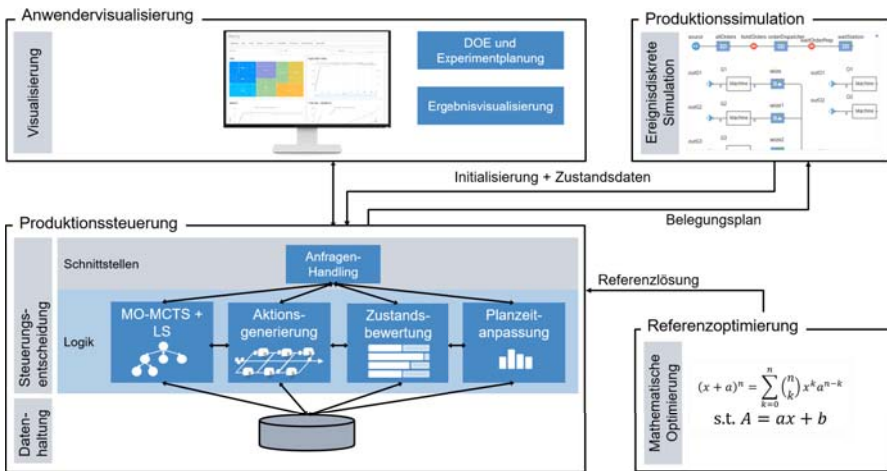


Abbildung 4.15: Gesamtarchitektur bestehend aus der Produktionssteuerung, einer Visualisierung, einer Produktionssimulation und einer Referenzoptimierung zur Generierung von Referenzlösungen

Das Gesamtsystem gliedert sich in vier Teilsysteme. Das zentrale Teilsystem bildet die Produktionssteuerung zur Berechnung der Steuerungsentscheidung mit angeschlossener Datenhaltung. Dieser Steuerungsserver in der Programmiersprache Python nimmt Steuerungsanfragen entgegen und leitet diese an den hybriden MO-MCTS-Optimierer mit lokaler Suche weiter, sofern eine Neuberechnung erfolgen soll. Die Aktionsgenerierung wird in einem getrennten Modul realisiert, das sowohl von MO-MCTS als auch von der lokalen Suche genutzt werden kann. Die Zustandsbewertung und Planzeitanpassung erfolgt ebenfalls in einem separaten Submodul, das anderen Teilsystemen zur Verfügung steht.

Das zweite Teilsystem bildet eine Anwendervisualisierung zur Experimentplanung und Ergebnisvisualisierung.

Das dritte Teilsystem bildet die Produktionssimulation. Diese ereignisdiskrete Simulation fungiert als Repräsentant einer stochastischen Produktion. Die Simulation kann unterschiedliche Produktionssysteme mit stochastischen Bearbeitungs-, Rüst-, und Transportzeiten sowie Störungen der Produktionsmittel abbilden. Die Simulation übermittelt zu Beginn Initialisierungsdaten unter anderem über das Produktionsprogramm, die Kapazitäten, Rüstzeit- und Transportmatrizen. Im laufenden Betrieb sendet die Simulation die aktuellen Zustandsdaten an die Steuerung, die auf Basis des aktuellen Zustands eine Steuerungsentscheidung trifft und diese wieder an die Simulation übermittelt. Die Steuerung muss hierzu aufeinanderfolgende Anfragen identifizieren können und aus den rückgemeldeten Daten der Simulation ableiten, ob Abweichungen zur Planung vorliegen. Die realisierten Bearbeitungs-, Rüst- und Transportdauern dienen als Ausgangspunkt für die Anpassung der Planzeiten.

Das vierte Teilsystem bildet eine Referenzoptimierung zum Generieren von Referenzlösungen zur Bewertung der Belegungspläne. Diese mathematische Optimierung kann als MILP-Problem formulierte Steuerungsprobleme lösen und das optimale Ergebnis zur Darstellung an das Steuerungssystem übermitteln. Dieser Optimierungsbaustein dient ausschließlich der Generierung von Referenzlösungen.

5 Untersuchung der Produktionssteuerung

Das folgende Kapitel untersucht die entwickelte Produktionssteuerung vor dem Hintergrund der drei gestellten Forschungsfragen:

[F1] Wie kann eine reaktive und vorausschauende Mehrziel-Steuerung für die Matrixproduktion unter Berücksichtigung realer Restriktionen gestaltet werden?

[F2] Kann durch automatisierte Anpassung an das vorliegende Steuerungsproblem eine hohe Lösungsgüte bei unterschiedlichen Steuerungsproblemen und Optimierungszielen erreicht werden?

[F3] Wie können die Dauern der Vorgänge in der Produktion fortlaufend überwacht, Verteilungsänderungen detektiert und realistische Dauern abgeschätzt werden?

Im ersten Schritt wird in Abschnitt 5.2 auf die Anpassung von Planzeiten basierend auf Realdaten eingegangen. Um Forschungsfrage **F3** zu beantworten, werden die Verteilungsschätzung, die Erkennung von Verteilungsänderungen und die Korrektur von saisonalen Effekten untersucht.

Der Schwerpunkt des Kapitels widmet sich der Beantwortung der Forschungsfragen **F1** und **F2**. Hierzu wird die Produktionssteuerung nach drei Gesichtspunkten evaluiert. Zur Bewertung der Eignung als vorausschauende Mehrziel-Steuerung wird untersucht, wie sich der Suchfokus des Verfahrens lenken lässt. Im Zentrum steht die Frage, welcher Bereich des Lösungsraums durch die Optimierung erreicht wird und welche Möglichkeiten bestehen, den Suchfokus zu beeinflussen. Hierfür wird die Aktionsgenerierung untersucht, da dieser Schritt bestimmt, welche Zustände an den Suchbaum angefügt werden. Anschließend werden die Möglichkeiten der Parallelisierung zur zeitgleichen Suche in unterschiedlichen Bereichen des Lösungsraums betrachtet sowie der Ansatz, den Fokus der Suche während der Optimierung sequentiell zu verschieben. Den Abschluss bildet die Betrachtung des Einflusses der lokalen Suche auf den erreichten Bereich des Lösungsraums. Zur Bewertung der gleichmäßigen Abdeckung des Lösungsraums wird das Hypervolumen als aggregierte Maßzahl für die Ausdehnung der Pareto-Front herangezogen.

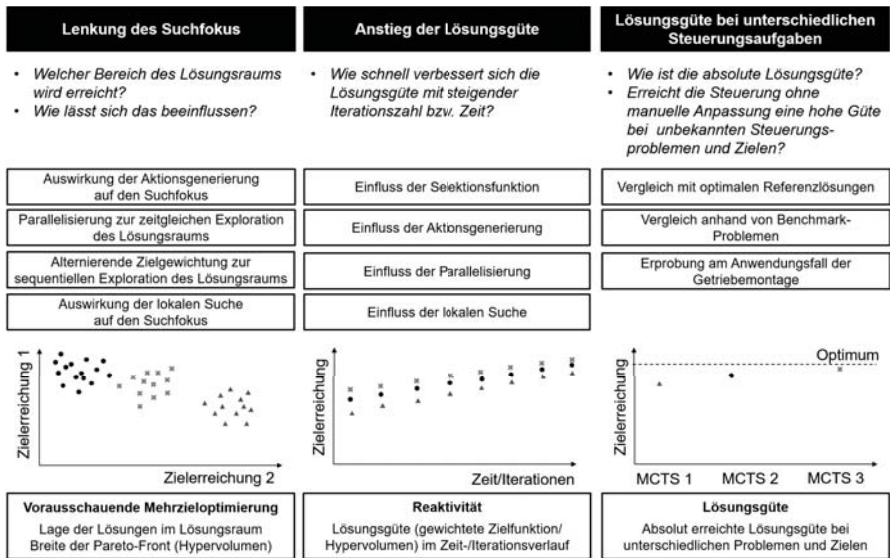


Abbildung 5.1: Vorgehen zur Untersuchung der Produktionssteuerung

Um den Aspekt der Reaktivität der Produktionssteuerung zu bewerten, wird der Anstieg der Lösungsgüte betrachtet. Ein schneller Anstieg ermöglicht, auch kurzfristig steuernd in den Produktionsablauf einzugreifen. Betrachtet wird hierfür der Anstieg der Lösungsgüte mit steigender Iterationszahl beziehungsweise steigender Ausführungszeit. Die Lösungsgüte kann als gewichteter Zielfunktionswert oder als Breite der Pareto-Front ausgedrückt werden. Da die Balance zwischen der Priorisierung des Ausbaus des Suchbaums in bereits bekannten Bereichen mit guten Lösungen (*Exploitation*) und der Erkundung noch unbekannter Bereiche (*Exploration*) durch die Selektionsfunktion bestimmt wird, bildet die Untersuchung des Einflusses der Selektionsfunktion auf den Anstieg der Lösungsgüte den Ausgangspunkt. Anschließend wird der Einfluss der Aktionsgenerierung beleuchtet. Durch die Einschränkung zulässiger Aktionen wird der Lösungsraum verkleinert. Dies kann zu einem schnelleren Anstieg der Lösungsgüte führen, wenn dadurch ungünstige Lösungen ausgeschlossen werden. Zudem wird untersucht, ob Parallelisierung einen schnelleren Anstieg der Lösungsgüte bewirkt. Abschließend werden die Auswirkungen der lokalen Suche auf die Suchgeschwindigkeit beleuchtet. Die lokale Suche benötigt als zusätzlicher Optimierungsschritt Zeit, kann jedoch durch die Postoptimierung der Ausgangslösungen zu besseren Ergebnissen führen.

Zur Beurteilung der Forschungsfrage F2 wird als letzter Fokus der Auswertungen untersucht,

wie sich die Lösungsgüte bei unterschiedlichen Steuerungsaufgaben verhält. Betrachtet wird die absolute Lösungsgüte anhand eines Vergleichs zu optimalen Referenzlösungen. Die Anwendung der Steuerung auf ein Benchmark-Problem aus der Literatur mit einer von der Matrixproduktion abweichenden Struktur und weiteren Zielgrößen zeigt, wie gut sich die Steuerung automatisiert an das neue Steuerungsproblem und die Zielgrößen anpassen kann. Die Erprobung anhand eines realen Anwendungsfalls aus der Getriebemontage demonstriert den Umgang der Steuerung mit stochastischen Dauern und Maschinenausfällen.

Für die Untersuchungen bezüglich der Lenkung des Suchfokus und des Anstiegs der Lösungsgüte werden generische Steuerungsprobleme einer Matrixproduktion mit Rüst- und Transportzeiten genutzt. Diese werden im Folgenden vorgestellt.

5.1 Anwendungsfälle

Zur Untersuchung der Produktionssteuerung werden drei Anwendungsfälle genutzt. Für die Untersuchung des Verhaltens der Produktionssteuerung wird erstens ein generisches Anwendungsbeispiel mit drei verschiedenen Problemgrößen verwendet. Dieser Anwendungsfall ist so ausgelegt, dass er in allen Zieldimensionen herausfordernd ist und eine gute Evaluation des Verfahrens ermöglicht. Zweitens wird ein Anwendungsfall aus dem Maschinenbau herangezogen. Dieses Praxisbeispiel dient der Erprobung unter realen Bedingungen. Drittens wird ein Benchmark-Anwendungsfall aus der Literatur genutzt, um den Vergleich zu anderen Verfahren zu ziehen.

5.1.1 Generisches Anwendungsproblem

Das generische Anwendungsproblem ist in drei Problemgrößen ausgeführt, die sich in der Anzahl der Produktionsaufträge und der Produktionsmittel unterscheiden (Anzahl Produktionsaufträge \times Anzahl Produktionsmittel). Der kleinste Anwendungsfall wird genutzt, um mit Hilfe eines exakten Verfahrens eine optimale Lösung als Referenz zu ermitteln. Alle Anwendungsfälle bilden eine Matrixproduktion mit alternativen Vorranggraphen ab. Im Folgenden werden die genutzten Beispielprobleme aufgeführt.

5.1.1.1 Anwendungsproblem 18x8

Das größte Anwendungsproblem besteht aus 18 Produktionsaufträgen in drei Produktvarianten und acht Produktionsmitteln. Jedes Produktionsmittel ist in der Lage, mehrere Vorgänge

durchzuführen. Jeder Produktionsauftrag hat einen frühesten Start- und einen spätesten Fertigstellungszeitpunkt (Anhang A2 Tabelle A1) und benötigt drei Vorgänge. Es gibt alternative Bearbeitungsvorgänge (Tabelle A2) sowie Rüst- und Transportzeiten. Eine Verspätung ist bei diesem Anwendungsfall unvermeidlich. Für maximale Flexibilität kann jedes Produktionsmittel jeden Vorgang durchführen. Die Transportzeiten zwischen den Produktionsmitteln betragen jeweils 5 ZE. Die Rüstzeiten belaufen sich auf 4 ZE. Die Bearbeitungszeiten und initialen Rüstzustände sind im Anhang der Tabelle A3 zu entnehmen.

5.1.1.2 Anwendungsproblem 12x5

Die Probleminstanz mit zwölf Produktionsaufträgen und fünf Produktionsmitteln ist eine Teilmenge des 18x8 Problems, wobei die Produktionsaufträge eins bis zwölf und die Produktionsmittel eins bis fünf zum Einsatz kommen. Transport- und Rüstzeiten sind unverändert.

5.1.1.3 Anwendungsproblem 5x3

Die kleinste Probleminstanz mit fünf Produktionsaufträgen und drei Produktionsmittel dient zur Generierung einer Referenzlösung mit Hilfe eines exakten Verfahrens. Die Transport- und Rüstzeiten entsprechen denen der anderen Probleminstanzen. Die drei Produktvarianten können auf fünf verschiedene Arten hergestellt werden, siehe Anhang Tabelle A4. Die drei Produktionsmittel entsprechen M1 bis M3 der anderen Probleminstanzen und befinden sich initial im selben Rüstzustand. Die fünf betrachteten Produktionsaufträge sind mit frühesten Start- und spätesten Fertigstellungszeitpunkten versehen, siehe Anhang Tabelle A5.

5.1.2 Anwendungsfall der Getriebemontage

Als Anwendungsfall aus der Praxis dient eine Endmontage von Getriebemotoren. Die Produktion wurde ausgehend von einer Linie flexibilisiert. Die Endmontage läuft in fünf Phasen ab. Den Anfang bildet die Getriebemontage. Dieser erste Schritt entspricht einem flexiblen Job Shop mit freier Wahl der Getriebemontageinsel für jeden Produktionsschritt. Im zweiten Schritt folgt die Befüllung mit Öl. Die Dauer der automatischen Ölbefüllung richtet sich nach Öltyp und Größe der Motorvariante. Wird der Schritt nicht benötigt, geht der Produktionsauftrag direkt in den Motoranbau. Im dritten Schritt wird der Anbau des Getriebes an den Motor vorgenommen. Der Motoranbau ist vergleichbar mit der Getriebemontage und kann an jeder Motoranbaustation durchgeführt werden. Die Funktionsprüfung (Test) bildet den vierten Schritt. Wird beim Test ein Fehler festgestellt, müssen alle vorangegangenen Vorgänge erneut

durchlaufen werden. Als fünfter und letzter Schritt erfolgt die Lackierung. Einzelne Varianten können diesen Schritt überspringen. Abbildung 5.2 zeigt den Aufbau schematisch.

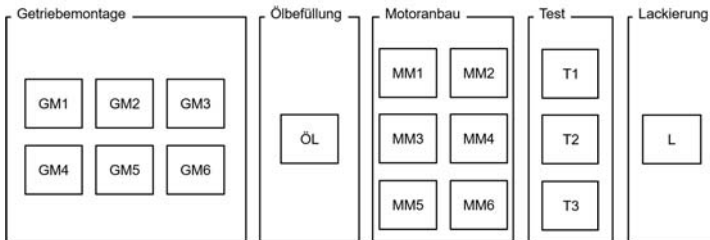


Abbildung 5.2: Schematisches Layout der Endmontage für Getriebemotoren

Insgesamt werden fünf Produktvarianten hergestellt. Gelegentlich werden auch Getriebe ohne Motor montiert. Diese dienen als Vorprodukt für die Endmontage an einem anderen Standort. Die Produktvarianten unterscheiden sich vor allem im Volumen und der Motorleistung. Daher gleicht sich die Abfolge der Vorgänge, jedoch bestehen deutliche Unterschiede in den Bearbeitungszeiten. Die Anteile der Varianten am Produktionsprogramm unterscheiden sich stark. Die Rennvariante P5 macht 56% des Produktionsvolumens aus, gefolgt von P2 mit 20% und P1 mit 13%. Auf die Exoten P3 und P4 entfallen 7% und 4% des Produktionsvolumens. Die Zuordnung der Vorgänge auf die Produktionsmittel, die benötigten Vorgänge pro Produktvariante und die Dauer der Vorgänge finden sich in Anhang A3.

Die Materialbereitstellung erfolgt nach dem Kanban-Prinzip an den einzelnen Produktionsmitteln, sodass eine kurzfristige Änderung der Produktionsmittelallokation möglich ist.

5.1.3 Benchmark-Problem aus der Literatur

Um den Vergleich zu anderen Ansätzen herzustellen, wird das Kacem-Benchmark-Problem verwendet (Kacem & Hammadi u. a. 2002). Das Kacem-Problem ist ein Sub-Problem des betrachteten Steuerungsproblems der Matrixproduktion. Im Gegensatz zu der betrachteten Steuerungsaufgabe werden beim Kacem-Problem keine Rüst- und Transportzeiten berücksichtigt. Zudem gibt es keine alternativen Vorgangssequenzen. Für die einzelnen Vorgänge stehen teils mehrere Produktionsmittel zur Verfügung, die unterschiedliche Bearbeitungszeiten benötigen. Die Optimierungsziele des Kacem-Problems sind die Gesamtarbeitsmenge, die maximale Arbeitsbelastung und der Makespan, siehe Kapitel 4.1. Keine der betrachteten Zielgrößen ist auftragsbezogen, zudem verhalten sich alle drei Zielgrößen komplementär.

5.2 Anpassung der Planzeiten basierend auf Realdaten

Als Planzeit wird die Zeit bezeichnet, welche die Steuerung für einen bestimmten Vorgang bei der Erstellung der Belegungspläne nutzt. Planzeiten werden für Bearbeitungs-, Transport- und Rüstvorgänge benötigt. Im Folgenden wird das in Kapitel 4.2 erarbeitete Vorgehen zur Schätzung der Planzeiten und zur Detektion von Verteilungsänderungen anhand realer Bearbeitungszeiten angewendet und evaluiert. Die Daten beziehen sich auf ein automatisiertes Produktionsmittel aus der Montage. Die Ergebnisse sind für die Beantwortung der Forschungsfrage **F3** wichtig.

5.2.1 Ermittlung der theoretischen Verteilung

Als vorbereitender Schritt wird ein Histogramm aus dem Datensatz erstellt. Liegen genügend Beobachtungen vor, wird die theoretische Verteilung anhand der Stichprobe geschätzt. Ausreißer werden herausgefiltert. Hierfür wird die Betrachtung auf das 1% - 99% Quantil eingeschränkt. Anschließend werden mit der Maximum-Likelihood-Methode die Parameter der angenommenen Verteilung geschätzt. Abbildung 5.3 zeigt Realdaten einer automatisierten Montagestation und die geschätzte Verteilung für die Normal- und Log-Normalverteilung. Der Datensatz besteht aus mehr als 10.000 Beobachtungen.

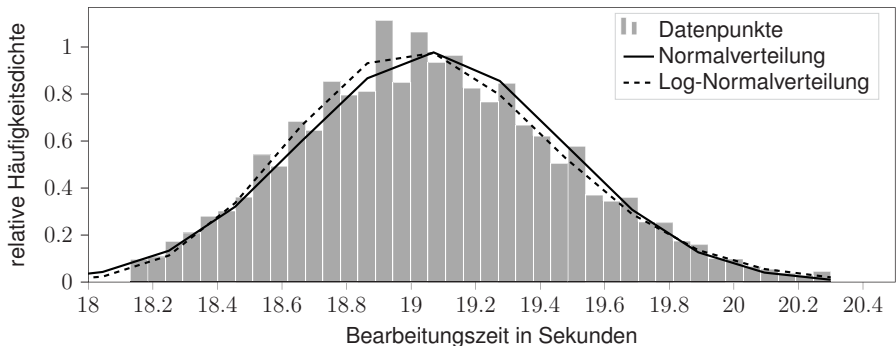


Abbildung 5.3: Reale Bearbeitungszeiten (1% - 99% Quantil) eines repräsentativen automatisierten Produktionsmittels sowie die geschätzte Log-Normalverteilung und die geschätzte Normalverteilung, angelehnt an (A_Wolf 2019)

Die Schätzung der Verteilungsfunktion anhand der Maximum-Likelihood-Methode für die Log-Normalverteilung ergibt für den Skalenparameter μ einen Wert von 3,909, für den

Formparameter σ einen Wert von 0,104 und einen Lageparameter von 15,134. Für diese dreiparametrische Log-Normalverteilung gilt, dass $Y = \ln(X - \lambda)$ normalverteilt ist gemäß $N(\mu, \sigma^2)$. Nach der Schätzung der Verteilungsfunktion wird die Güte der Schätzung mit einem Hypothesentest bewertet. Die Nullhypothese besagt, dass die Stichprobe aus der geschätzten Verteilung entstammt. Für ein Konfidenzniveau von 0,05 ergibt der Kolmogorov-Smirnov-Anpassungstest für die Log-Normalverteilung einen p-Wert von 0,152 und für die Normalverteilung einen p-Wert von $1 \cdot 10^{-6}$. Die Betrachtung der p-Werte zeigt, dass für die Log-Normalverteilung die Nullhypothese nicht verworfen werden kann. Dagegen kann die Nullhypothese für die Normalverteilung verworfen werden. Zur Ermittlung der Planzeiten sollte in diesem Fall die geschätzte Log-Normalverteilung herangezogen werden.

5.2.2 Erkennung von Verteilungsänderungen

Zur Untersuchung der Fähigkeit Verteilungsänderungen zu erkennen, werden Realdaten eines Produktionsmittels als Basis verwendet. Nach einem Monat wurde eine Parameteränderung durchgeführt, die zu einer Reduktion des Mittelwerts um etwa 5% in den beiden Folgemonaten geführt hat, siehe Abbildung 5.4.

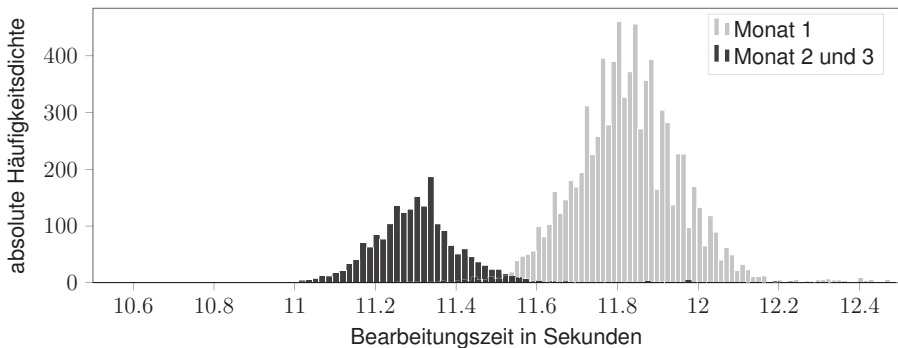


Abbildung 5.4: Realdaten eines Produktionsmittels über drei Monate. Nach dem ersten Monat findet ein Concept Drift statt mit einer Reduktion des Mittelwerts. Das Produktionsvolumen des ersten Monats liegt deutlich über dem Niveau der beiden Folgemonate, angelehnt an (A_Wolf 2019)

Für die Untersuchung der Verteilungsänderungsdetektion wird zuerst die Verteilung aus den Realdaten geschätzt. Anschließend werden 150 Punkte anhand der initialen Verteilung erzeugt, ab der 151. Beobachtung wird die neue Verteilung als Grundlage genutzt. Die

Beobachtungen werden anhand der unterschiedlichen Methoden zur *Concept-Drift*-Detektion auf eine Verteilungsänderung untersucht. Bewertet wird die Dauer bis zur Erkennung der Änderung sowie die Zuverlässigkeit der *Concept-Drift*-Detektion.

Die aus den Realdaten geschätzte Verteilung für den ersten Monat ist eine Normalverteilung mit den Parametern $N(\mu = 11,83, \sigma = 0,13)$, ab dem zweiten Monat werden die Beobachtungen aus der neuen Verteilung $N(\mu = 12,42, \sigma = 0,13)$ erzeugt.

5.2.2.1 Änderungsdetektion durch statistische Prozesskontrolle

Im Folgenden wird die EWMA-Regelkarte als Beispiel der Ansätze der statistischen Prozesskontrolle zur Detektion einer Verteilungsänderung vorgestellt. Diese Ansätze beruhen darauf, dass auf Basis der Kenntnis über die stationäre Verteilung untere und obere Eingriffsgrenzen definiert werden, deren Erreichen eine Veränderung der Verteilung bedeuten.

Für die Festlegung der Eingriffsgrenzen werden die ersten 15 Datenpunkte herangezogen. Bis diese Datenpunkte erreicht sind, ist keine Aussage möglich. Je größer die Anzahl betrachteter Datenpunkte für die Festlegung der Eingriffsgrenzen ist, umso besser wird die stationäre Verteilung abgebildet. Gleichzeitig sinken die Einsatzmöglichkeiten aufgrund der benötigten Datenpunkte. Für die EWMA-Regelkarte wurden vorab geeignete Parameter experimentell bestimmt (hier $L = 3$ und $\lambda = 0,15$).

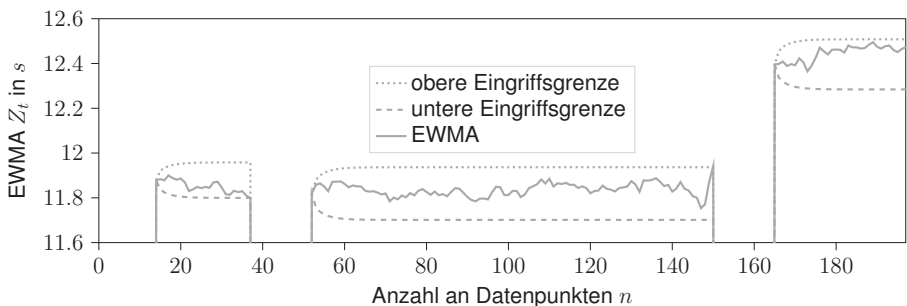


Abbildung 5.5: EWMA-Regelkarte zur *Concept-Drift*-Detektion mit der Schätzung des Mittelwerts und den Eingriffsgrenzen, angelehnt an (A_Wolf 2019)

Abbildung 5.5 zeigt die resultierende EWMA-Regelkarte mit der Mittelwertschätzung und den Eingriffsgrenzen. Die Festlegung der Eingriffsgrenzen anhand der ersten 15 Beobachtungen ist am Anfang gut erkennbar. Die Karte detektiert fälschlicherweise nach 37 Datenpunkten

einen *Concept-Drift* und startet daraufhin die Festlegung der Eingriffsgrenzen erneut. Der tatsächliche *Concept-Drift* wird nach 151 Beobachtungen erkannt, also mit einer Verzögerung von einer Beobachtung.

5.2.2.2 Änderungsdetektion durch Verteilungsvergleich

Aus dem Bereich des Verteilungsvergleichs wird die ADWIN-Methode untersucht, siehe Abschnitt 4.2.1.1. Die ADWIN-Methode betrachtet alle benachbarten Teilfenster und vergleicht die Stichproben der betrachteten Teilfenster. Im Fall einer Verteilungsänderung wird das stabile Fenster auf die maximale Größe zurückgesetzt, bei dem die Nullhypothese nicht verworfen werden kann. Die Nullhypothese besagt, dass die beiden Stichproben der gleichen Verteilung angehören.

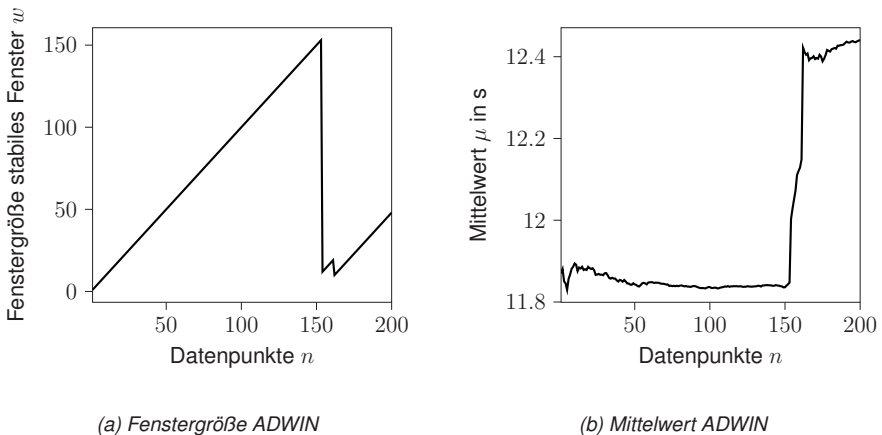


Abbildung 5.6: Größe des stabilen Zeitfensters der ADWIN-Methode und Mittelwertschätzung der ADWIN-Methode, angelehnt an (A_Wolf 2019)

Abbildung 5.6 zeigt die Fenstergröße der ADWIN-Methode sowie den errechneten Mittelwert. ADWIN erkennt die Änderung der Verteilung nach 153 Beobachtungen. Kurz darauf erfolgt eine zweite, falsche Änderungsdetektion. Bei der Änderungsdetektion wird das stabile Fenster nicht vollständig zurückgesetzt, sondern startet mit einem Wert von drei.

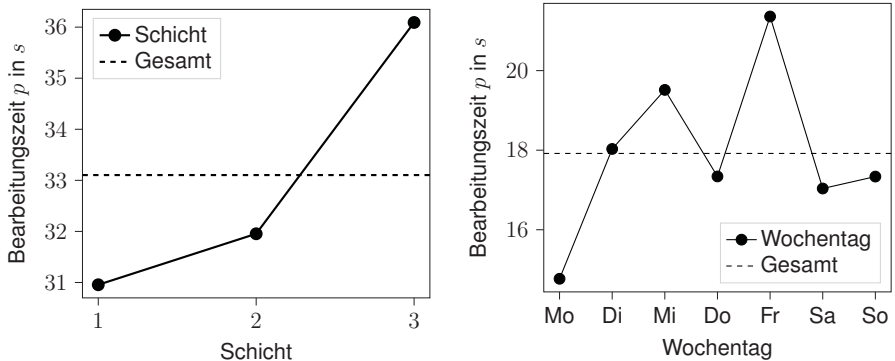
Zusammenfassend lässt sich feststellen, dass sowohl die ADWIN-Methode als auch die EWMA-Karte die Verteilungsänderung nach 150 Datenpunkten detektieren. Die statistischen Methoden erfordern jedoch Kenntnis der Verteilung und eine Parametrisierung der Prozessregelkarte. ADWIN hingegen hat den Vorteil, dass die Fenstergrößen nicht vorab festgelegt,

sondern dynamisch angepasst werden und bei einer Änderung ein Teil der Beobachtungen wieder verwendet wird. Aufgrund der erzielten Genauigkeit der Änderungsdetektion sowie der Geschwindigkeit der Erkennung der Verteilungsänderung wird ADWIN empfohlen.

5.2.3 Erkennung von Trends und Saisonalitäten

Die Zeiten in der Produktion unterliegen systematischen Schwankungen, die in den Planzeiten berücksichtigt werden sollten. Typischerweise bestehen Unterschiede zwischen den Tag- und Nachtschichten (Graf 1960) und zwischen den Wochentagen. Im Folgenden soll untersucht werden, wie das vorgestellte System diese beiden Phänomene erkennt. Dies erfolgt anhand von Realdaten aus einem Zeitraum von neun Monaten.

Verglichen werden die Mittelwerte der Bearbeitungszeiten eines Produktionsmittels für die drei Schichten mit dem Mittelwert über alle Schichten. Die Bearbeitungszeit der Frühschicht (Schicht 1) weicht um -6% ab, die der Spätschicht (Schicht 2) um -4% und die der Nachtschicht (Schicht 3) um 9%. Alle Werte sind statistisch signifikant. Die Anwendung der Methodik auf die Daten eines weiteren Montageplatzes aus einem Zeitraum von ebenfalls neun Monaten zeigt deutliche Unterschiede nach Wochentagen. Details können der Abbildung 5.7 entnommen werden. Nicht alle Produktionsmittel zeigen klare Abhängigkeiten der Bearbeitungszeiten nach Schichten oder Wochentagen.



(a) Bearbeitungszeitunterschiede pro Schicht

(b) Bearbeitungszeitunterschiede pro Wochentag

Abbildung 5.7: Bearbeitungszeit pro Schicht und durchschnittliche Bearbeitungszeit über alle Schichten für ein Produktionsmittel sowie die Bearbeitungszeit pro Wochentag für einen ausgewählten Montageplatz, angelehnt an (A_Wolf 2019)

Diese Abweichungen der Durchschnittswerte werden als Korrekturfaktor bei der Ermittlung der Planzeiten berücksichtigt, sodass die Produktionssteuerung auch schicht- und tageabhängig die Planzeit anpassen kann. Im Sinne der Forschungsfrage **F3** kann somit geschlossen werden, dass die vorgestellte Methodik in der Lage ist, aus Realdaten Verteilungsparameter zu schätzen, Verteilungsänderungen zu erkennen und tage- sowie schichtbezogene Korrekturen der Planzeiten vorzunehmen.

5.3 Lenkung des Suchfokus im mehrdimensionalen Lösungsraum

Der folgende Abschnitt befasst sich mit dem Aspekt der vorausschauenden Mehrzieloptimierung, thematisiert in Forschungsfrage **F1**. Im Zentrum der Untersuchungen steht die Frage, welcher Bereich des Lösungsraums durch das Optimierungsverfahren abgesucht wird und wie der Suchfokus beeinflusst werden kann.

Eingangs werden die Auswirkungen der Aktionsgenerierung auf den Suchfokus untersucht. Im ersten Schritt wird der Zusammenhang zwischen der Aktionsgenerierung und der Gestalt des Suchbaums aufgezeigt, da der Suchbaum den Lösungsraum abbildet. Anschließend wird der Einfluss der Ansätze zur Einschränkung zulässiger Aktionen auf den erreichten Bereich des mehrdimensionalen Lösungsraums analysiert, siehe Abschnitt 4.3.3. Den zweiten Schwerpunkt bildet die Betrachtung der Parallelisierung zum zeitgleichen Absuchen unterschiedlicher Bereiche des Lösungsraums. Als Drittes werden Ansätze der alternierenden Suche in verschiedenen Bereichen des Lösungsraums adressiert. Den vierten Schwerpunkt bildet die Untersuchung der Auswirkungen der lokalen Suche auf den erreichten Lösungsraum.

5.3.1 Auswirkungen der Aktionsgenerierung auf den Suchfokus

Da der aufgebaute Suchbaum den Lösungsraum repräsentiert, wird eingangs untersucht, wie sich die Ansätze der Aktionsgenerierung auf die **Gestalt des Suchbaums** auswirken.

Die Aktionsgenerierung im Rahmen der Expansionsphase läuft in drei Schritten ab. Im ersten Schritt werden die Produktionsaufträge anhand von Kriterien sortiert. Herrscht nach einem Kriterium Indifferenz, wird das nächste Kriterium herangezogen, bis die Produktionsaufträge eindeutig sortiert sind. Hierfür stehen mehrere Kriterien zur Auswahl, die in eine Anwen-dungsreihenfolge gebracht werden müssen. Ist die Sortierung erfolgt, werden anschließend in Schritt 2 für die betrachtete Teilmenge der Produktionsaufträge alle möglichen Aktionen ermittelt. Eine Aktion gibt an, welcher Vorgang auf welches Produktionsmittel in welchem Zeitfenster allokiert wird. Im letzten Schritt werden die resultierenden Aktionen durch Anwendung

von Filtern reduziert. Hierfür stehen mehrere Filterheuristiken zur Verfügung, die beliebig kombiniert werden können. Um die Effekte des Vorgehens einzeln aufzuzeigen, wird der Einfluss der **Sortierung der Produktionsaufträge**, die Auswirkung der **Anzahl der parallel betrachteten Produktionsaufträge** und der Einfluss der **Einschränkung der zulässigen Aktionen** durch Filter anhand von vier Filterkombinationen verdeutlicht. Anschließend wird die entwickelte Methode zur intelligenten Einschränkung der Aktionen untersucht. Diese Methode ermittelt selbstständig geeignete Sortierkriterien und Filterkombinationen. Es ist zu erwarten, dass die Anzahl parallel betrachteter Produktionsaufträge und die Filterkombinationen die größte Einschränkung des Lösungsraums bewirken, da diese beiden Faktoren sich direkt auf die Anzahl der Aktionen auswirken, während die Sortierung der Produktionsaufträge lediglich einen Einfluss darauf hat, welche Produktionsaufträge als Ausgangspunkt der Aktionsermittlung genutzt werden.

5.3.1.1 Auswirkungen der Aktionsgenerierung auf die Gestalt des Suchbaums

Da der Suchbaum den Lösungsraum für das MO-MCTS-Verfahren abbildet, wird zuerst der Einfluss der Aktionsgenerierung auf den Suchbaum dargestellt. Es wird untersucht, wie sich die Anzahl parallel betrachteter Produktionsaufträge in Schritt 1 der Aktionsgenerierung und die Wahl der Filter zur Einschränkung der zulässigen Aktionen in Schritt 3 der Aktionsgenerierung auf die Gestalt des Suchbaums auswirken, da diese beiden Faktoren einen direkten Einfluss auf die Menge der Aktionen haben.

Hierzu werden vier Filterkombinationen zur Einschränkung der zulässigen Aktionen untersucht, die den Lösungsraum zunehmend verkleinern, siehe Tabelle 5.1. Die Betrachtung wird mit Hilfe der 18x8 Problem Instanz durchgeführt, siehe Tabelle 5.2. Die Anzahl gleichzeitig betrachteter Produktionsaufträge gibt an, wie viele Produktionsaufträge in einer Entscheidungsiteration simultan evaluiert werden. Für dieses Problem beträgt die maximale Anzahl 18 Produktionsaufträge. Wird nur eine Teilmenge der Produktionsaufträge betrachtet, verkleinert sich in der nachfolgenden Iteration der Lösungsraum, da bereits ein Vorgang im Belegungsplan allokiert wurde. Gleichzeitig sinkt dadurch die Komplexität des Problems deutlich, was im Folgenden gezeigt wird.

Die Größe, Tiefe und Breite des Suchbaums wird anhand folgender Maßzahlen bewertet:

- **Verzweigungsfaktor:** Der Verzweigungsfaktor für Bäume gibt die durchschnittliche Anzahl Kindknoten eines Elternknotens an.
- **Baumtiefe:** Maximale Tiefe des Suchbaums nach der letzten Expansion

Tabelle 5.1: Konfigurationen zur Untersuchung der Auswirkungen der Aktionsgenerierung auf die Gestalt des Suchbaums
 nicht aktiv: ○, aktiv: ●, Reihenfolge der Anwendung der Sortierkriterien: 1 - 7, variabel: ~

Konfiguration	Auftragslimit	Auswahl betrachteter Aufträge (Sortierung)							Einschränkung zulässiger Aktionen (Filter)										
		Priorität	Status	Früherster Startzeitpunkt	Früherster Endzeitpunkt	Kürzeste Bearbeitungszeit	Längste Bearbeitungszeit	Name	Dringlichkeit	Vorgang/Maschine	Rüst- und Transportzeiten	Rüstvorgänge	Transportvorgänge	Produktionsstatus	Anzahl Vorgänge	Alternative Aufträge	Makespan-Grenze	Früherster Startzeitpunkt	
Keine Filter	~	5	3	4	2	1	6	7	○	○	○	○	○	○	○	○	○	○	○
Geringe Filter	~	5	3	4	2	1	6	7	○	○	●	○	○	○	○	●	○	○	○
Mittlere Filter	~	5	3	4	2	1	6	7	●	●	○	○	○	○	●	●	○	○	○
Hohe Filter	~	5	3	4	2	1	6	7	○	●	●	○	○	●	●	●	●	●	●

Tabelle 5.2: Wirkung der Sortierkriterien und Filterkombinationen auf den Suchbaum und Wirkung der Filterkombinationen auf den erreichten Lösungsraum

Problem- instanz	Terminierungs- kriterium	Ziel- funktion	Aktionsgenerierung (1. Sortierkriterium, Filterkombinationen)	C-Wert
18x8	2.000 Iterationen	Durchlaufzeit	Kürzeste Bearbeitungszeit, 4 Filterkombinationen	0,15

- **Baumgröße:** Anzahl expandierter Knoten
- **Explorationsgrad:** Anteil der bekannten Aktionen, die expandiert wurden

Abbildung 5.8 zeigt für die vier betrachteten Filterkombinationen den durchschnittlichen Verzweigungsfaktor, die Baumtiefe, die Anzahl Blattknoten und den Explorationsgrad, abhängig von der Anzahl parallel betrachteter Produktionsaufträge in der Aktionsgenerierung. Die Ergebnisse bleiben bestehen, wenn ein anderes Zielkriterium als die Durchlaufzeit herangezogen wird. Die dargestellten Werte sind der Mittelwert aus zehn MCTS-Ausführungen. In Grafik (a) wird deutlich, dass der Verzweigungsfaktor etwa linear mit der Anzahl der betrachteten Produktionsaufträge wächst. Werden keine Einschränkungen vorgenommen, wird ein sehr hoher Verzweigungsfaktor erreicht, der bei gleichzeitiger Betrachtung aller Produktionsaufträge zu durchschnittlich 350 zulässigen Aktionen pro Elternknoten führt. Geringe Einschränkungen führen nur zu einer marginalen Reduktion der Aktionsanzahl. Die Filter-

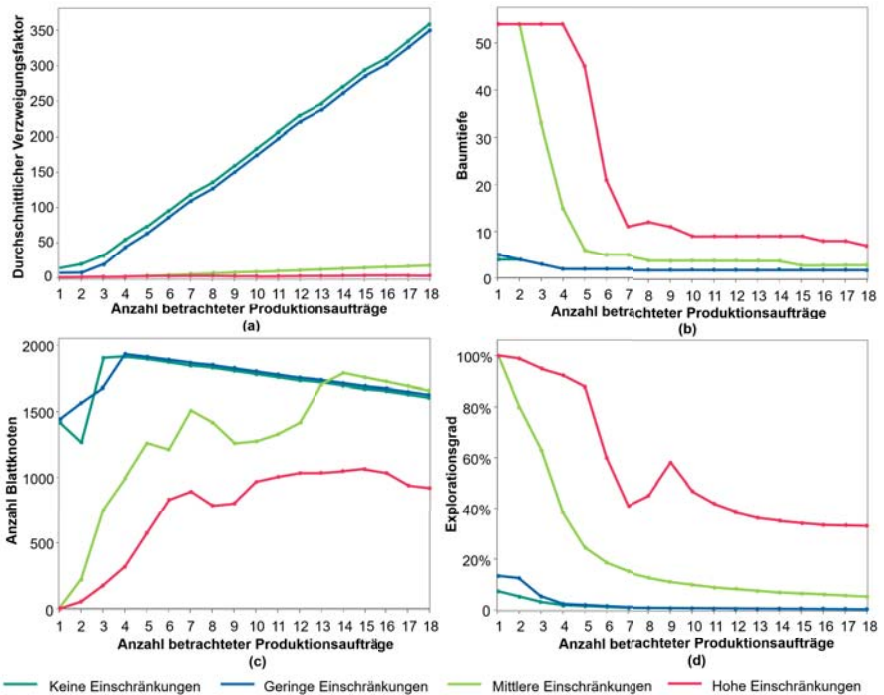


Abbildung 5.8: Einfluss der Anzahl betrachteter Produktionsaufträge und der Einschränkung zulässiger Aktionen auf den Suchbaum, angelehnt an (A_Deining 2020)

kombination mit mittleren und hohen Einschränkungen können die durchschnittliche Anzahl zulässiger Aktionen drastisch reduzieren. Grafik (b) zeigt, dass diese Filterkombinationen mit mittleren und hohen Einschränkungen für eine geringe Anzahl an parallel betrachteten Produktionsaufträgen eine große Baumtiefe in der Expansion erreichen. Hierfür muss der Suchbaum entweder relativ klein sein und daher bis in die Tiefe aufgebaut werden oder vorrangig nur ein Bereich des Lösungsraums durchsucht werden. Grafik (c) zeigt, dass die Anzahl der Blattknoten bei Filterkombinationen mit mittlerer oder starker Einschränkung geringer ist als bei den Filterkombinationen mit wenig Einschränkungen. Diese Kennzahl steht für die unterschiedliche Baumgröße, die durch die Einschränkungen des Lösungsraums erreicht wird. Besonders deutlich ist der Effekt bei hohen Einschränkungen und gleichzeitig wenigen betrachteten Aufträgen. In diesen Fällen besteht der Suchbaum nur aus wenigen Knoten. Daraus lässt sich schließen, dass die hohe Baumtiefe, siehe Grafik (b), aus der Größe des Suchbaums resultiert und nicht aus einer selektiven Suche. Grafik (d) stellt den Explorationsgrad

grad der unterschiedlichen Filterkombinationen dar, abhängig von der Anzahl betrachteter Produktionsaufträge. In allen Fällen wurden 2.000 Iterationen durchgeführt. Nur die Anzahl der zur Verfügung stehenden Aktionen ist unterschiedlich. Je größer die Anzahl betrachteter Produktionsaufträge, umso geringer fällt der Explorationsgrad aus, da die Anzahl möglicher Aktionen zunimmt. Die Lage der Kurven zueinander zeigt, dass die stark einschränkenden Filterkombinationen den Suchraum deutlich reduzieren. Bei starken Einschränkungen und einer geringen Anzahl Produktionsaufträge wird sogar der gesamte Suchraum aufgespannt, Explorationsgrad gleich 1. Es fällt auf, dass der Explorationsgrad bei hohen Einschränkungen steigt, sobald etwa genauso viele Aufträge wie Produktionsmittel zeitgleich betrachtet werden. Auch weitere Experimente konnten jedoch keinen klaren Zusammenhang belegen.

Die Untersuchungen zeigen, dass die Anzahl parallel betrachteter Produktionsaufträge und die Wahl der Filterkombinationen einen starken Einfluss auf die Größe des erzeugten Suchbaums und damit auf den abgebildeten Lösungsraum haben. Im Weiteren wird der Einfluss der Sortierkriterien, der Anzahl parallel betrachteter Produktionsaufträge und der Filterkombinationen auf die Lage der Lösungen im Lösungsraum untersucht. Die Abdeckung des Lösungsraums durch die Pareto-Punkte wird anhand des Hypervolumens analysiert. Im folgenden Abschnitt wird zuerst auf die Effekte der Einschränkungen der zulässigen Aktionen durch verschiedene Filterkombinationen eingegangen. Danach wird der Einfluss der Anzahl parallel betrachteter Produktionsaufträge in Kombinationen mit der besten Filterkombinationen untersucht (Abschnitt 5.3.1.2). Im anschließenden Abschnitt steht der Einfluss der Sortierkriterien zur Bildung einer Rangfolge der Produktionsaufträge im ersten Schritt der Aktionsgenerierung im Vordergrund (Abschnitt 5.3.1.3).

5.3.1.2 Einfluss der Einschränkung zulässiger Aktionen durch Filterkombinationen und der Anzahl parallel betrachteter Produktionsaufträge

Um zu bewerten, wie die Wahl der Filterkombinationen den erreichten Lösungsraum beeinflusst, werden die drei Zielgrößen kumulierte Durchlaufzeit, kumulierte Verspätung und Makespan optimiert. Abgesehen von den Zielgrößen wird die MO-MCTS-Konfiguration aus der vorangegangenen Untersuchung verwendet, siehe Tabelle 5.2. Die Auswertung nach dem Hypervolumen fasst die Ausdehnung der Pareto-Front zusammen.

Abbildung 5.9 zeigt in Grafik (a) das erreichte Hypervolumen abhängig von der Anzahl der parallel betrachteten Produktionsaufträge und der angewandten **Einschränkungen zulässiger Aktionen** durch verschiedene Filterkombinationen. Der Einfluss der Anzahl parallel betrachteter Produktionsaufträge wird anschließend getrennt analysiert. Die Lage der Kurven zueinander macht deutlich, dass die angewandten Filterkombinationen einen Einfluss auf

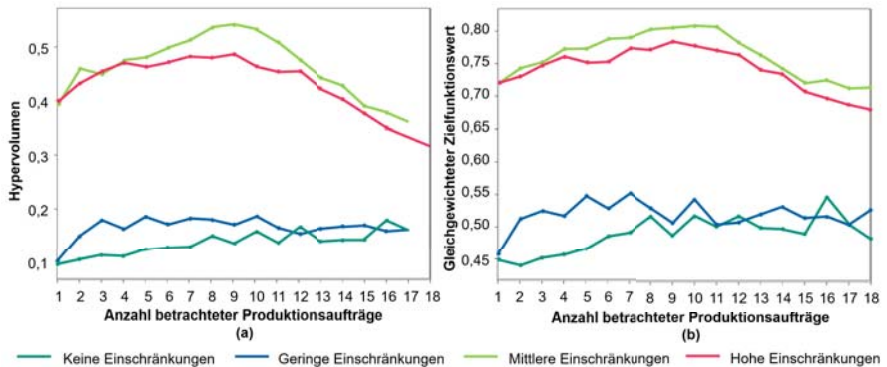


Abbildung 5.9: Hypervolumen und gleichgewichteter Zielfunktionswert für die Zielkriterien Durchlaufzeit, Termintreue und Makespan, angelehnt an (A_Deining 2020)

die Größe des Hypervolumens haben. Aus Grafik (b) ist ersichtlich, dass die Qualität der Lösungen für Filterkombinationen mit geringen Einschränkungen niedrig ist. Dies deckt sich mit den Erkenntnissen über die breiten und flachen Suchbäume aus Abschnitt 5.3.1.1. Für den Suchfokus bedeutet das, dass die Filterkombinationen mit nur geringen Einschränkungen für die gegebene Anzahl Iterationen nur einen Bereich unterhalb der Paretofront erreichen. Übereinstimmend mit dem Stand der Technik zeigen die Auswertungen, dass die Reduktion des Lösungsraums zu verbesserten Ergebnissen führen kann. Die Ergebnisse aus Abschnitt 5.3.1.1 belegen, dass für diese Einschränkungen der Suchbaum deutlich verkleinert wird.

Für die Untersuchung des Einflusses der **Anzahl parallel betrachteter Produktionsaufträge** wird ebenfalls die Mehrzieloptimierung nach Gesamtdurchlaufzeit, Makespan und kumulierter Verspätung vorgenommen. Es wird jedoch die Filterkombination mit mittlerer Einschränkung genutzt, da für diese Einstellung das erreichte Hypervolumen am größten und damit die Pareto-Front am breitesten ist, siehe Abbildung 5.9.

Abbildung 5.10 zeigt die Lage der generierten Lösungen im normierten Lösungsraum für 18 MO-MCTS-Varianten, die sich in der Anzahl parallel betrachteter Produktionsaufträge unterscheiden. Jeder Punkt einer Farbe stellt den Belegungsplan eines Terminalzustands am Ende der *Rollout*-Phase dar. Die Pareto-Front ist durch eine Linie gekennzeichnet. Grafik (a) zeigt deutlich, dass der erreichte Bereich des Lösungsraums stark von der Anzahl der parallel betrachteten Produktionsaufträge abhängt. Für die Zielgrößen Verspätung und Durchlaufzeit ist eine positive Korrelation zwischen der Lösungsqualität und der Anzahl parallel betrachteter Produktionsaufträge sichtbar. Grafik (b) stellt die Beziehung zwischen Makespan und Durchlaufzeit dar. Hier zeigt sich ein anderes Bild. Die Werte für die Anzahl

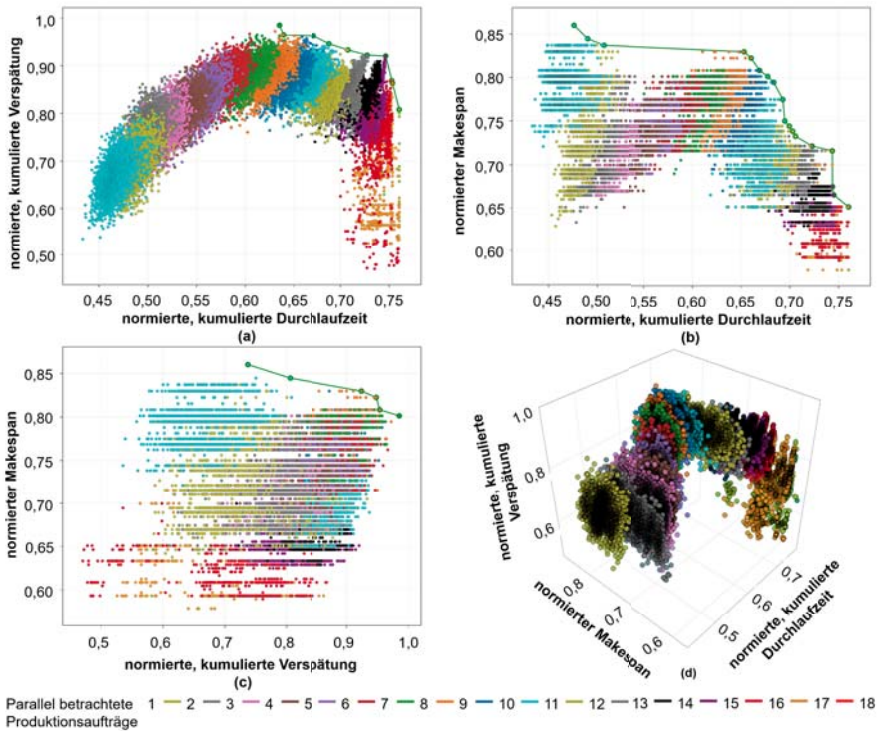


Abbildung 5.10: Auswirkung der Anzahl parallel betrachteter Produktionsaufträge auf den erreichten Lösungsraum für Durchlaufzeit, Termintreue und Makespan, angelehnt an (A_Deiningner 2020)

parallel betrachteter Produktionsaufträge, die für die Makespan-Optimierung günstig sind, zeigen eine geringe Eignung für die Optimierung der Durchlaufzeit. Für die kumulierte Durchlaufzeit ist eine relativ große Lücke zwischen den Pareto-Punkten bei 0,51 und 0,65 zu erkennen. Auch wenn die genutzte Filterkombination mit mittleren Einschränkungen die breiteste Pareto-Front liefert, wird in diesem Bereich der Lösungsraum nur bedingt abgesucht. Grafik (c) zeigt den Zusammenhang für die Zielkriterien Verspätung und Makespan. Hier ist keine Korrelation erkennbar. Grafik (d) stellt die drei Zielgrößen in einem dreidimensionalen Koordinatensystem dar. In dieser aggregierten Darstellung ist die konzentrierte Lage der Lösungen für einen Wert der Anzahl parallel betrachteter Produktionsaufträge an der Ballung der einzelnen Punkte gut erkennbar.

Es wird ersichtlich, dass alle betrachteten Optimierungsziele durch die Produktionssteuerung erreichbar sind. Allerdings sind abhängig vom angestrebten Ziel, eine andere Anzahl parallel betrachteter Produktionsaufträge und andere Filterkombinationen zu wählen. Die Unterschiedlichkeit der erzielten Lösungen kann anhand der resultierenden Belegungspläne gut beobachtet werden. Abbildung A5.1 in Anhang 5 zeigt den optimalen Belegungsplan nach der kumulierten Durchlaufzeit, erzielt mit Filterkombinationen mittlerer Einschränkung und einem parallel betrachteten Produktionsauftrag. Demgegenüber steht der Belegungsplan in Abbildung A5.2 mit sehr gutem Makespan, der mit sehr stark einschränkenden Filterkombinationen und 18 parallel betrachteten Produktionsaufträgen erzielt wurde.

5.3.1.3 Einfluss der Sortierung der Produktionsaufträge

Nach der Untersuchung der Auswirkungen der Filterkombinationen und der Anzahl parallel betrachteter Produktionsaufträge in der Aktionsgenerierung auf den erreichbaren Teil des Lösungsraums, werden die Effekte der **Sortierung der Produktionsaufträge** untersucht. Hierfür werden drei verschiedene Kombinationen aus Sortierkriterien verglichen: Erstens die vorrangige Sortierung der Aufträge nach der längsten Bearbeitungszeit, zweitens nach der kürzesten Bearbeitungszeit und drittens aufsteigend nach dem spätestmöglichen Fertigstellungszeitpunkt. Für die Experimente wurde die Filterkombination mit mittleren Einschränkungen der zulässigen Aktionen in Kombination mit zehn parallel betrachteten Produktionsaufträgen genutzt, siehe Tabelle 5.3 und Tabelle 5.4.

Tabelle 5.3: Untersuchung der Sortierung der Produktionsaufträge in der Expansionsphase nicht aktiv: ○, aktiv: ●, Anwendung der Sortierkriterien: 1 - 7

Konfiguration	Auftragslimit	Auswahl betrachteter Aufträge (Sortierung)							Einschränkung der zulässigen Lösungen (Filter)										
		Priorität	Status	Frühster Startzeitpunkt	Frühster Endzeitpunkt	Kürzeste Bearbeitungszeit	Längste Bearbeitungszeit	Name	Dringlichkeit	Vorgang/Maschine	Rüst- und Transportzeiten	Rüstvorgänge	Transportvorgänge	Produktionsstatus	Anzahl Vorgänge	Alternative Aufträge	Makespan-Grenze	Frühster Startzeitpunkt	
Längste Bearbeitung	10	6	5	4	2	3	1	7	●	●	○	○	○	○	○	●	●	○	○
Kürzeste Bearbeitung	10	6	5	4	2	1	3	7	●	●	○	○	○	○	○	●	●	○	○
Frühster Endzeitpunkt	10	6	5	4	1	3	2	7	●	●	○	○	○	○	○	●	●	○	○

Tabelle 5.5 zeigt die besten Ergebnisse aus jeweils zehn Versuchen für jede der drei analysierten Ansätze zur Sortierung der Produktionsaufträge. Es ist ersichtlich, dass sich je nach Zielkriterium eine andere Auswahl der Produktionsaufträge empfiehlt. Diese Auswertung zeigt, dass auch die Sortierung, genauso wie die Anzahl parallel betrachteter Produktionsaufträge, abhängig von der angestrebten Zielgröße gewählt werden sollte. Dieses Ergebnis deckt sich mit den Erkenntnissen aus dem Stand der Technik für regelbasierte Heuristiken. Die unterschiedliche Größe des Hypervolumens zeigt zudem, dass der erreichte Bereich des Lösungsraums unterschiedlich groß ist. Die Streuung der Ergebnisse für die betrachteten Sortierkombinationen kann Abbildung 5.11 entnommen werden.

Tabelle 5.4: Auswirkung der Sortierungskriterien der Produktionsaufträge auf den erreichten Lösungsraum

Problem- instanz	Terminierungs- kriterium	Ziel- funktion	Aktionsgenerierung (1. Sortierkriterium, Filterkombinationen)	C-Wert
18X8	2.000 Iterationen	Durchlaufzeit, Verspätung, Makespan	3 Sortierkombinationen, Mittlere Einschränkung	0,15

Tabelle 5.5: Maximal erreichter normierter Zielfunktionswert für die untersuchten Sortierkombinationen der Produktionsaufträge, angelehnt an (A_Deiningner 2020)

Ziel- kriterium	Längste Bearbeitungszeit	Kürzeste Bearbeitungszeit	Frühster Fertigstellungstermin
Kumulierte Durchlaufzeit	0,6123	0,6619	0,6532
Kumulierte Verspätung	0,8333	0,9704	1,0
Makespan	0,8297	0,8297	0,8013
Hypervolumen	0,4463	0,5411	0,5300

5.3.1.4 Auswirkung der intelligenten Einschränkung zulässiger Aktionen

Die bisherigen Untersuchungen der Aktionsgenerierung zeigen zwei grundlegende Eigenschaften in Bezug auf den erreichbaren Lösungsraum. Erstens haben sowohl die Wahl der Anzahl parallel betrachteter Produktionsaufträge als auch die Filterkombinationen und Sortierkriterien einen Einfluss auf die Lage der Pareto-Punkte im Lösungsraum. Zweitens sind durch die geeignete Ausgestaltung der Aktionsgenerierung alle Optimierungsziele erreichbar. Es ist jedoch kein Ansatz in der Lage, alle Bereiche gleichermaßen zu erreichen. Diese Erkenntnisse legen nahe, dass es mit einer statischen Herangehensweise nicht möglich ist, den Lösungsraum für unterschiedliche Ziele und Anwendungsfälle effizient und breit

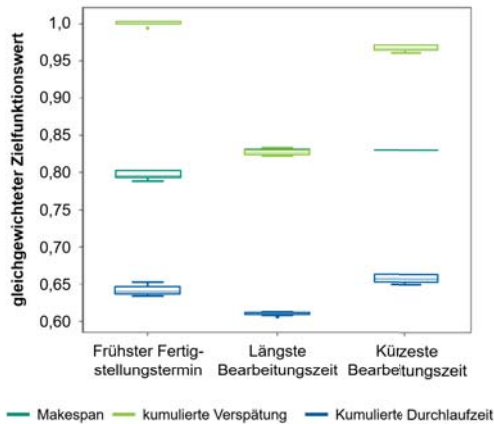


Abbildung 5.11: Boxplot für die unterschiedlichen Auswahlverfahren und die Zielkriterien Durchlaufzeit, Termintreue und Makespan, angelehnt an (A_Deininger 2020)

abzusuchen. Ein statisches Vorgehen eignet sich demnach nicht für eine vorausschauende Mehrzieloptimierung.

Der in Kapitel 4.3.3.4 vorgestellte Ansatz zur intelligenten Einschränkung zulässiger Aktionen ermittelt in der ersten Phase, dem Experimentierbetrieb, geeignete Kombinationen aus Filterkombinationen und Sortierkriterien sowie eine passende Anzahl parallel betrachteter Produktionsaufträge zur Aktionsgenerierung. Hierbei stehen alle bislang vorgestellten Mechanismen der Aktionsgenerierung zur Verfügung. In der Produktivphase werden die besten Kombinationen zu einer hohen Wahrscheinlichkeit angewandt, während zu einer geringen Wahrscheinlichkeit weiterhin zufällige Kombinationen erprobt werden. Die intelligente Einschränkung der Aktionen wird im Produktivbetrieb sowohl in der Expansionsphase als auch in der *Rollout*-Phase genutzt.

Für den Experimentierbetrieb werden 20 MO-MCTS-Ausführungen mit je 1.000 Iterationen durchgeführt. Für eine hohe Exploration wird ein c -Wert von 2 genutzt. Ziel ist die Maximierung des Hypervolumens, um eine möglichst gute Abdeckung des gesamten Lösungsraums zu erreichen, siehe Tabelle 5.7.

Die Ergebnisse aus den 20.000 MO-MCTS-Iterationen der 20 MO-MCTS-Ausführungen dienen als Grundlage für die Auswahl der besten Sortierkriterien, Filterkombinationen und der Anzahl parallel betrachteter Produktionsaufträge. Es werden alle Kombinationen behalten, die zu den fünf besten Pareto-Fronten geführt haben. Des Weiteren werden die Kombinationen

Tabelle 5.6: Konfiguration für Rollout und Expansion im Experimentierbetrieb
 nicht aktiv: ○, aktiv: ●, Anwendung der Sortierkriterien: 1 - 7, veränderbarer Wert
 mit Startwert $x: \bar{x}$

Konfiguration	Auftragslimit	Auswahl betrachteter Aufträge (Sortierung)							Einschränkung der zulässigen Lösungen (Filter)										
		Priorität	Status	Früherster Startzeitpunkt	Früherster Endzeitpunkt	Kürzeste Bearbeitungszeit	Längste Bearbeitungszeit	Name	Dringlichkeit	Vorgang/Maschine	Rüst- und Transportzeiten	Rüstvorgänge	Transportvorgänge	Produktionsstatus	Anzahl Vorgänge	Alternative Aufträge	Makespan-Grenze	Früherster Startzeitpunkt	
Expansionsphase	12	4	2	5	1	6	3	7	●	●	○	○	○	○	○	●	●	○	○
Rollout-Phase	12	5	4	6	1	2	3	7	○	○	○	○	○	○	○	○	○	○	○

Tabelle 5.7: Konfiguration der Experimente im Experimentierbetrieb

Problem- instanz	Terminierungs- kriterium	Ziel- funktion	Aktionsgenerierung (1. Sortierkriterium, Filterkombinationen)	C-Wert
12x5	1.000 Iterationen	Durchlaufzeit, Verspätung, Makespan	siehe Tabelle 5.6	2

genutzt, die mindestens in 50% der Fälle und mindestens zwei Mal zu einer Lösung geführt haben, die maximal 10% von der Pareto-Front entfernt ist. Um auch in der Produktivphase die Erprobung neuer Kombinationen und damit eine Anpassung der Aktionsgenerierung an eine sich wandelnde Steuerungsaufgabe zu gewährleisten, werden weiterhin zu 5% zufällige Kombinationen erprobt.

Abbildung 5.12 zeigt die Ergebnisse für die 12x5 Probleminstanz. Dargestellt sind die Lösungen der Terminalzustände des Experimentier- und Produktivbetriebs. Grafik (a)-(c) zeigen, dass die Lösungen, die im Produktivbetrieb gefunden werden, im relevanten Bereich des Lösungsraums liegen und die gesamte Pareto-Front abdecken. Grafik (d) zeigt den Zielfunktionswert als Linearkombination aus den drei betrachteten Zielen. Die Lösungen, die im Produktivbetrieb erreicht werden, konzentrieren sich auf die höchsten Zielfunktionswerte. Es wird deutlich, dass das Verfahren der intelligenten Einschränkung der zulässigen Aktionen in der Lage ist, eine breite Pareto-Front zu bilden und gleichzeitig hohe Zielfunktionswerte zu erreichen. Im Gegensatz zu den bisher vorgestellten Ansätzen wird automatisiert eine geeignete Parametrisierung erzeugt, die den Lösungsraum in allen Dimensionen breit abdecken

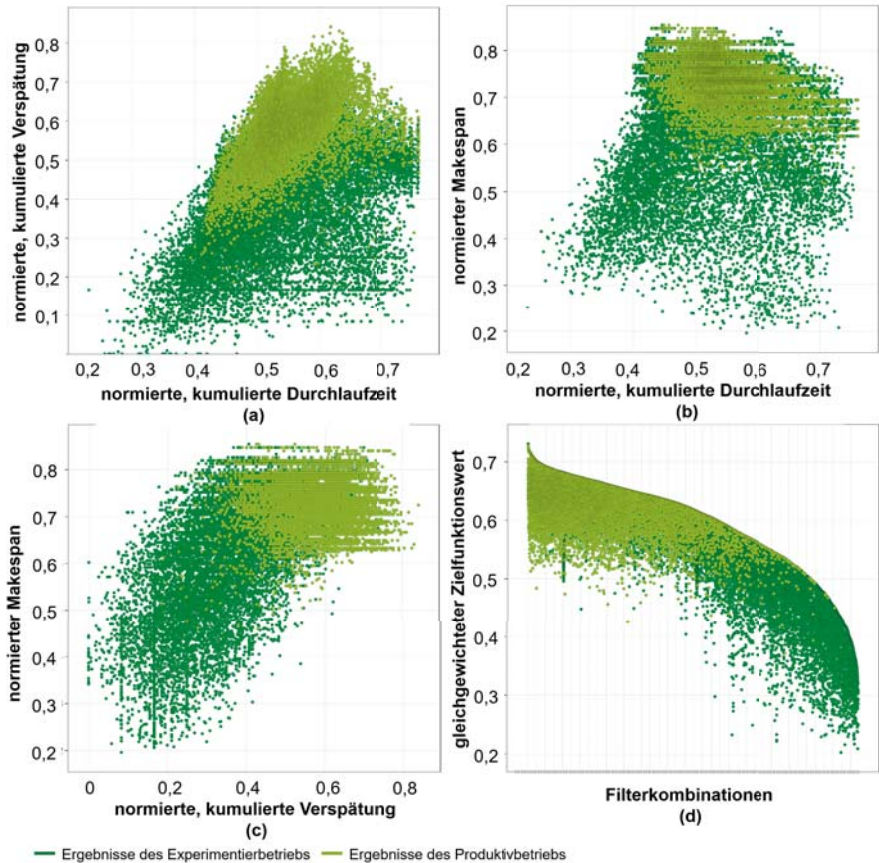


Abbildung 5.12: Ergebnisse des Experimentier- und des Produktivbetriebs hinsichtlich Durchlaufzeit, Verspätung, Makespan und des gleichgewichteten Zielfunktionswerts, angelehnt an (A_Deiningner 2020)

kann. Die intelligente Einschränkung der Aktionen ermöglicht somit eine vorausschauende Mehrzieloptimierung im Sinne der Forschungsfrage **F1**. Zudem erfolgt die Abstimmung der Produktionssteuerung auf die Optimierungsziele selbstständig, sodass von einer automatisierten Anpassung an unterschiedliche Optimierungsziele gesprochen werden kann, wie in Forschungsfrage **F2** gefordert.

5.3.2 Parallelisierung zur zeitgleichen Exploration des Suchraums

Parallelisierung kann genutzt werden, um zeitgleich mehrere Filterkombinationen in der Aktionsgenerierung anzuwenden, um unterschiedliche Bereiche des Lösungsraums zu erreichen. Die Auswertungen zu den Auswirkungen der Anzahl parallel betrachteter Produktionsaufträge haben einen großen Einfluss auf die Lage der Pareto-Lösungen im Lösungsraum gezeigt, siehe Abschnitt 5.3.1.2. Um die Effekte der Parallelisierung für den Aufbau einer breiten Pareto-Front durch unterschiedliche Mechanismen in der Aktionsgenerierung aufzuzeigen, werden sieben *Threads* mit unterschiedlicher Anzahl parallel betrachteter Produktionsaufträge verwendet. Der achte *Thread* der vier Dual-Core Prozessoren dient der Zusammenführung der Ergebnisse. Jeder *Thread* startet im gleichen Ausgangszustand s_0 und hat die gleiche Aktionsmenge A zur Verfügung (Root-Parallelisierung). Durch die Unterschiede in der Aktionsgenerierung werden verschiedene Bereiche des Lösungsraums erreicht. Damit jeder *Thread* einen anderen Bereich des Lösungsraums zielgerichtet absucht, wird ein geringer Explorationsfaktor gewählt. Gleichzeitig werden die Gewichte der Zielgrößen in der Selektionsphase so verschoben, dass auch hier Lösungen gut bewertet werden, die in dem Bereich liegen, der durch die Aktionsgenerierung in der Expansionsphase gut erreichbar ist. Für eine geringe Anzahl parallel betrachteter Produktionsaufträge wird der Fokus in der Zielfunktion ebenfalls zu Gunsten der Durchlaufzeit gelegt. Mit zunehmender Anzahl parallel betrachteter Produktionsaufträge wird die Gewichtung zu Gunsten der Auslastung verschoben.

Tabelle 5.8: Filterkombinationen in der Aktionsgenerierung der einzelnen parallelen Prozesse bei den Experimenten mit Root-Parallelisierung

Problem- instanz	Terminierungs- kriterium	Zielfunktion Durchlaufzeit:Auslastung	Aktionsgenerierung	C-Wert
12x5	20 Sekunden		Kürzeste Bearbeitungszeit, mittlere Einschränkung,	0,01
		Thread 1: 3:2	Thread 1: 4 Aufträge	
		Thread 2: 3:2	Thread 2: 5 Aufträge	
		Thread 3: 2:3	Thread 3: 6 Aufträge	
		Thread 4: 2:3	Thread 4: 7 Aufträge	
		Thread 5: 3:7	Thread 5: 8 Aufträge	
		Thread 6: 1:4	Thread 6: 10 Aufträge	
		Thread 7: 1:9	Thread 7: 12 Aufträge	

Im Kontext der Parallelisierung sollte das Terminierungskriterium zeitbezogen gewählt werden, da sich die Anzahl Iterationen pro Zeit aufgrund der unterschiedlichen Aktionsgenerierung zwischen den parallelen Prozessen unterscheiden wird. Tabelle 5.8 stellt die Wahl der Parameter des Experiments dar.

Abbildung 5.13 zeigt die Lage der Terminallösungen der sieben *Threads* im Lösungsraum,

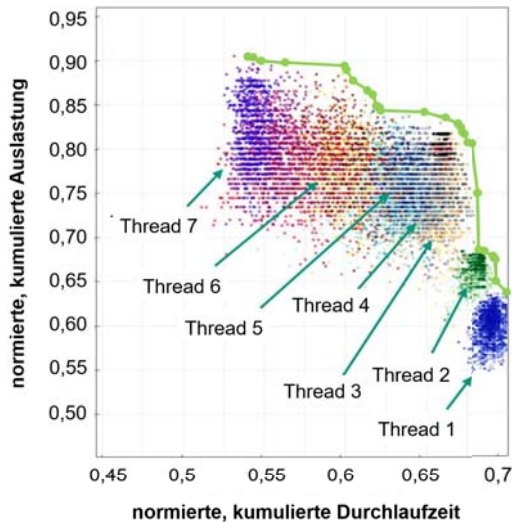


Abbildung 5.13: Lösungen am Ende der Rollout-Phase bei Root-Parallelisierung mit sieben Threads mit unterschiedlichen Auftragslimits und Zielgewichtungen für den 12x5 Anwendungsfall, angelehnt an (A_Wurster 2019)

aufgespannt durch die beiden normierten Zielkriterien, Auslastung und kumulierte Durchlaufzeit. Es ergibt sich ein vergleichbares Bild zu den Auswertungen auf dem 18x8-Problem in Abschnitt 5.3.1.2. Insgesamt erzielt die Root-Parallelisierung den Aufbau einer breiten Pareto-Front mit einer guten Abdeckung des Lösungsraums. Jedoch ist Domänenwissen zur Parametrisierung der unabhängigen Threads nötig, sodass diese unterschiedliche Bereiche des Lösungsraums abdecken.

5.3.3 Alternierende Zielgewichtung zur sequentiellen Exploration des Suchraums

Das Suchverhalten des MO-MCTS-Verfahrens wird auch durch die Wahl des zu expandierenden Knotens in der Selektionsphase beeinflusst. Im Fall der Mehrzieloptimierung bestimmt die Gewichtung der Ziele in der Selektionsfunktion, in welchem Bereich des Lösungsraums der Suchbaum weiter aufgebaut wird. Eine Möglichkeit den Suchfokus während der MO-MCTS-Ausführung zu steuern, ist es, die Selektion alternierend und unikriteriell nach den unterschiedlichen Zielfunktionswerten vorzunehmen, siehe Abschnitt 4.3.2.2.

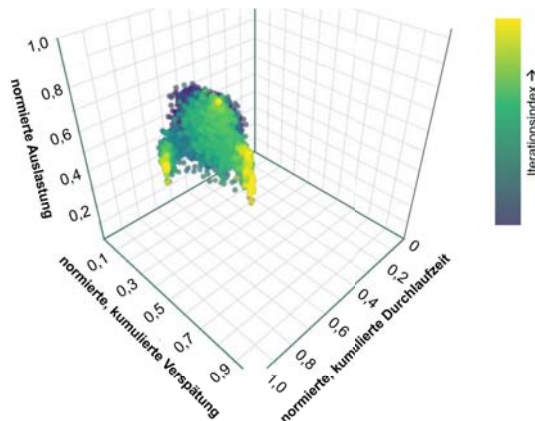


Abbildung 5.14: Terminalzustände je Iteration bei rollierender Zielgewichtungen der Einzelziele in der Selektionsphase, angelehnt an (A_Wurster 2019)

Abbildung 5.14 zeigt das Ergebnis für eine alternierende Selektion nach den drei Zielkriterien kumulierte Verspätung, kumulierte Durchlaufzeit und Auslastung sowie der gleichgewichteten Linearkombination aus den drei Zielgrößen. Zusätzlich zu der unikriteriellen Auswahl nach einem Zielkriterium wird gemäß der gleichgewichteten Linearkombination gewählt, um den Bereich zwischen den Einzelzielen zu erfassen. Jede der Suchrichtungen erhält abwechselnd den Fokus. Die Auswertung wurde mit der Konfiguration aus Tabelle 5.9 vorgenommen.

Tabelle 5.9: Alternierende Zielgewichtung

Problem- instanz	Terminierungs- kriterium	Zielfunktion Durchlaufzeit:Auslastung	Anzahl parallel betrachteter Produktionsaufträge	C-Wert
12x5	10.000 Iterationen	Alternierend	Kürzeste Bearbeitungszeit, mittlere Einschränkung	2

Die zu den drei Zielgrößen gehörenden Suchrichtungen sind deutlich an der Konzentration gelber Punkte erkennbar, die für die Terminalzustände späterer Iterationen stehen. Die gleichgewichtete Linearkombination zeichnet sich nicht als abgegrenzte Region im Suchraum ab. Dies liegt daran, dass unterschiedliche Kombinationen aus den Einzelzielen zu einem guten gleichgewichteten Zielfunktionswert führen.

Die Experimente zur *Root*-Parallelisierung und zur alternierenden Zielgewichtung zeigen, dass auf diese Weise der Lösungsraum gleichmäßiger abgesucht werden kann. Jedoch erfolgt die Suche in beiden Fällen durch spezialisierte Konfigurationen des MO-MCTS, die

entweder zeitgleich oder iterativ angewendet werden. Weiterhin gilt, dass der erreichbare Teil des Lösungsraums stark von der manuellen Wahl der Parameter abhängt.

5.3.4 Auswirkung der lokalen Suche auf den Suchfokus

Als vierte Untersuchung im Zusammenhang mit der Lenkung des Suchfokus wird der Einfluss der lokalen Suche auf den erreichten Lösungsraum betrachtet. Durch lokale Suche wird der benachbarte Lösungsraum einer Ausgangslösung abgesucht. Daher ist mit einer Vergrößerung des Hypervolumens zu rechnen. Im Folgenden werden zuerst unterschiedliche Algorithmen der lokalen Suche evaluiert. Das am besten geeignete Verfahren wird für die weiteren Auswertungen genutzt. Danach wird auf die modifizierten Nachbarschaftsstrukturen und die Sortierung der Belegungsintervalle und Vorgänge eingegangen, bevor die Effekte der Auswahl der Ausgangslösung beleuchtet werden. Einen Schwerpunkt der Analysen bildet die Betrachtung der Effekte der Reallokation kritischer Vorgänge in einer Iteration der lokalen Suche. In diesem Zusammenhang werden ebenfalls unterschiedliche Akzeptanzkriterien einer Lösung betrachtet. Den Abschluss der Auswertungen bildet die Untersuchung, in wie weit lokale Suche zu einer verbesserten Abdeckung des Lösungsraums führen kann, wenn das Basisoptimierungsverfahren ungünstig parametrisiert ist und den Lösungsraum nur lokal begrenzt absucht.

Tabelle 5.10: Konfiguration des MO-MCTS als Basisverfahren zur Generierung der Ausgangslösung zur Untersuchung der lokalen Suche

Problem- instanz	Terminierungs- kriterium	Ziel- funktion	Aktionsgenerierung (1. Sortierkriterium, Filterkombinationen)	C-Wert
18x8	2.000 Iterationen	Termintreue Durchlaufzeit Makespan Gesamtarbeitsmenge Maximale Arbeitslast	Kürzeste Bearbeitungszeit, Mittlere Einschränkung	C=0,25

5.3.4.1 Verfahren der lokalen Suche

Untersucht werden die Tabu-Suche mit verschiedenen Nachbarschaftsstrukturen, lokale Suche basierend auf Tauschoperationen und auf Verschiebeoperationen sowie LSONE. Die Ausgangslösungen werden durch MO-MCTS als Basisoptimierungsverfahren erzeugt, siehe Tabelle 5.10.

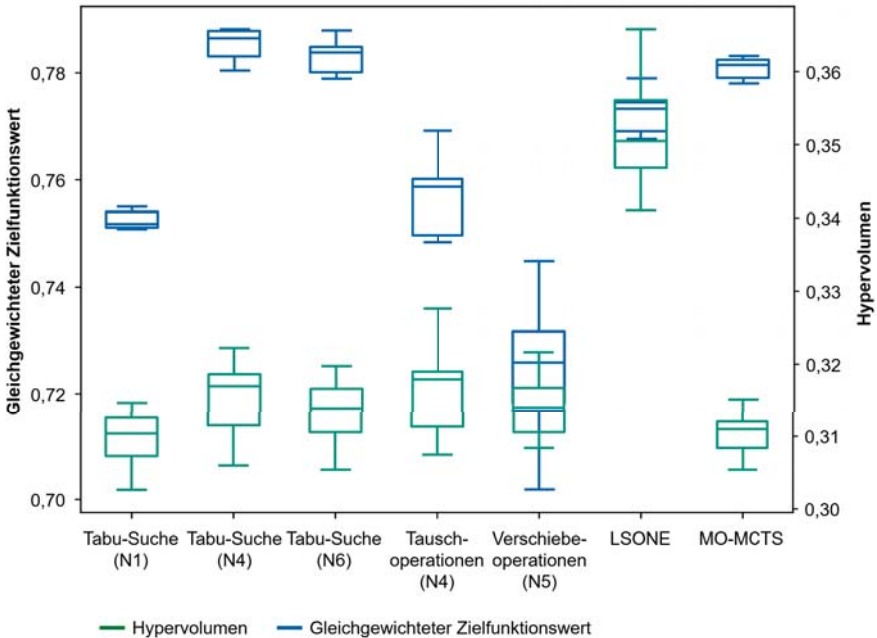


Abbildung 5.15: Hypervolumen und gleichgewichteter Zielfunktionswert für die Tabu-Suche mit Nachbarschaftsstruktur N1, N4, N6, Tauschoperationen mit Nachbarschaftsstruktur N4 und Verschiebeoperationen mit Nachbarschaftsstruktur N5 sowie LSONE und MO-MCTS, angelehnt an (A_Liu 2021)

Abbildung 5.15 zeigt den gleichgewichteten Zielfunktionswert und das Hypervolumen für die unterschiedlichen Verfahren der lokalen Suche aus zehn Ausführungen. Die Verfahren basierend auf Tabu-Suche, siehe Abschnitt 4.5.4.1, mit den Nachbarschaftsstrukturen N1, N4 und N6 zeigen ein vergleichbares Hypervolumen. Die gleichgewichteten Zielfunktionswerte unterscheiden sich etwas. Wird aber die Nachbarschaftsstruktur N4 mit Tauschoperation ohne Tabu-Suche verwendet, siehe Abschnitt 4.5.4.2, sinkt der Zielfunktionswert. Verschiebeoperationen führen zu den schlechtesten Ergebnissen. Im Gegensatz zu den bisherigen Verfahren der lokalen Suche kann LSONE das Hypervolumen deutlich steigern. Der gleichgewichtete Zielfunktionswert liegt jedoch in einer vergleichbaren Größe wie bei den Tabu-Suche-Verfahren. Dies zeigt, dass die Pareto-Front vor allem in die Breite wächst. Da im Vergleich mit den bisherigen Verfahren LSONE das Hypervolumen deutlich stärker vergrößert, wird im Weiteren ausschließlich LSONE für die Betrachtungen herangezogen.

5.3.4.2 Modifikation der LSONE-Nachbarschaftsstruktur

Da die ursprüngliche LSONE-Datenstruktur weder Rüst- noch Transportzeiten berücksichtigt, kann dies dazu führen, dass Intervalle, die durch LSONE ermittelt werden, unzulässig sind. Daher wurde LSONE weiterentwickelt, siehe Abschnitt 4.5.3.

Abbildung 5.16 zeigt das Hypervolumen und den gleichgewichteten Zielfunktionswert für die Modifikation von LSONE, siehe Abschnitt 4.5.3, verglichen mit dem LSONE nach Gao & Sun u. a. (2008) und der Ausgangslösung durch MCTS.

Die Modifikation von LSONE, die Transport- und Rüstzeiten berücksichtigt, führt zu deutlich besseren Ergebnissen sowohl hinsichtlich der Breite der Pareto-Front (a) als auch hinsichtlich des gleichgewichteten Zielfunktionswerts (b). Die Streuung der Lösungen nimmt durch die lokale Suche zu, jedoch liegen alle Werte oberhalb der Referenzlösungen.

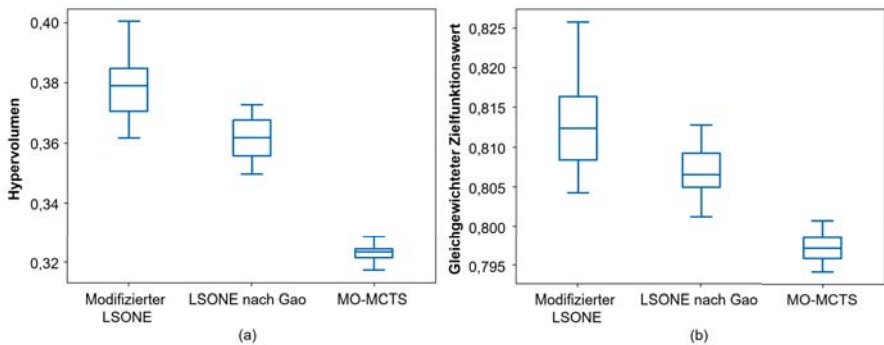


Abbildung 5.16: Hypervolumen und gleichgewichteter Zielfunktionswert für die Modifikation von LSONE, LSONE nach Gao & Sun u. a. (2008) und für MO-MCTS, angelehnt an (A_Liu 2021)

Durch die Modifikationen konnte zudem in über 25% der Iterationen der lokalen Suche eine Verbesserung erzielt werden, bei der Basisversion von LSONE liegt dieser Wert bei knapp unter 20%.

5.3.4.3 Sortierung der Belegungsintervalle und kritischen Vorgänge

Die Vorsortierung der kritischen Vorgänge und der Belegungsintervalle, siehe Abschnitt 4.5.3.3, ermöglicht, vielversprechende Veränderungen der Nachbarschaft in den ersten Iterationen der lokalen Suche zu erproben. Die Regeln zur Sortierung der Vorgänge und der

Belegungsintervalle beziehen sich auf die gleichen Kriterien. Bei der abgestimmten Sortierung werden die Regeln zur Sortierung der kritischen Vorgänge und der Belegungsintervalle aufeinander abgestimmt. So werden beispielsweise die Vorgänge mit den meisten Rüstzeiten auf die Produktionsmittel mit den geringsten Rüstzeiten verlagert. Dies muss nicht zwingend die beste Sortierung sein, da parallel existierende Restriktionen nicht berücksichtigt werden.

Der Effekt der Sortierung ist dann am deutlichsten, wenn wenig Iterationen und keine Reallokation erlaubt sind. Die Möglichkeit zur Reallokation führt dazu, dass automatisch unterschiedliche Optionen evaluiert werden. Abbildung 5.17 zeigt das Hypervolumen (a) und den Anteil der erfolgreichen Veränderungen (b) abhängig von der angewendeten Sortierung. Es ist ersichtlich, dass eine Sortierung sowohl zu einer breiteren Pareto-Front als auch zu einem größeren Anteil an erfolgreichen lokalen Suchiterationen führt. Die Abstimmung der Sortierregeln aufeinander sorgt jedoch nicht für ein besseres durchschnittliches Ergebnis bezüglich des Hypervolumens. Dies liegt wahrscheinlich daran, dass die Regeln nur einen Aspekt des Problems abbilden, aber andere Restriktionen außer Acht lassen. Beobachtbar ist allerdings auch, dass die Abstimmung der Regeln zu einer geringeren Varianz im Anteil der erfolgreichen Züge führt.

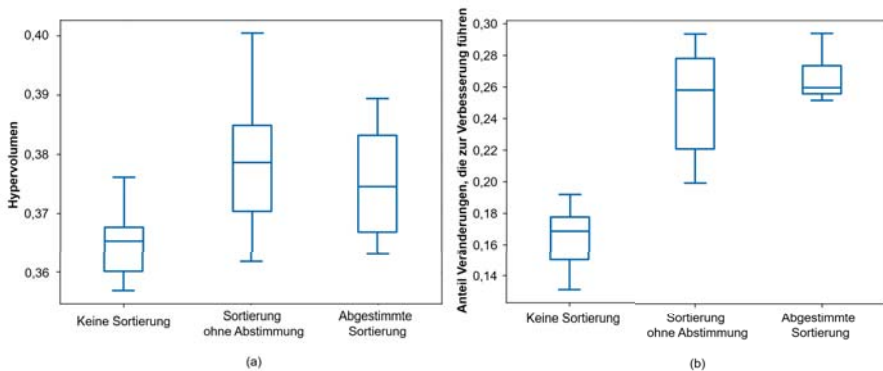


Abbildung 5.17: Hypervolumen und Anteil der erfolgreichen Veränderungen abhängig von der Vorsortierung, angelehnt an (A_Liu 2021)

5.3.4.4 Auswahl der Ausgangslösung

Die Methode zur Auswahl der Ausgangslösungen für die lokale Suche hat auch Einfluss auf die erreichbaren Ergebnisse in der lokalen Suche. Untersucht werden ein zufälliges *Sampling* mit Wahrscheinlichkeit 0,1 nach jedem *Rollout* (*Rollout-Sampling*), die besten 25

Pareto-Punkte nach jeweils 80 Iterationen (BPI) und die 25 besten Pareto-Punkte nach der Ausführung (BPA) hinsichtlich des erzielten Hypervolumens, siehe Abschnitt 4.5.1.

Abbildung 5.18 zeigt das Hypervolumen abhängig von der Anzahl gleichzeitig betrachteter Produktionsaufträge für unterschiedliche Auswahlverfahren der Ausgangslösung für die lokale Suche. Die zufällige Auswahl des besten Pareto-Punktes aller 80 Iterationen und die Verwendung der besten 25 Pareto-Punkte am Ende der MCTS-Ausführung zeigen sehr vergleichbare Ergebnisse. Das zufällige *Sampling* von Terminalzuständen am Ende des *Rollouts* führt zu leicht besseren Ergebnissen insbesondere in den Randbereichen. Da das *Rollout* als Basis verwendet wird, startet die lokale Suche auch von schlechteren Ausgangslösungen, die nicht Teil der Pareto-Menge sind. Dadurch scheinen die Randbereiche stärker repräsentiert zu sein als bei Verfahren, welche die Pareto-Menge als Grundlage nehmen. Zudem führt dieses Auswahlverfahren zu etwa vier mal mehr Iterationen der lokalen Suche als die Pareto-Mengen-basierten Verfahren. Die Kombination aller drei Verfahren liefert die besten Ergebnisse, wenn auch auf vergleichbarem Niveau.

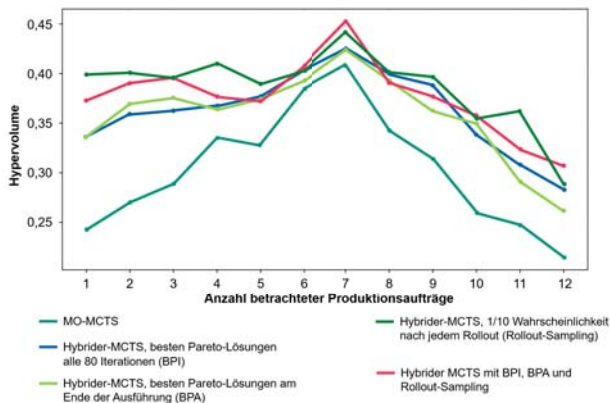


Abbildung 5.18: Hypervolumen abhängig von der Anzahl betrachteter Aufträge und der Wahl der Ausgangslösungen für das 12x5 Problem, angelehnt an (Liu 2021)

5.3.4.5 Reallokation kritischer Vorgänge

Stehen der lokalen Suche mehrere Belegungsintervalle für einen kritischen Vorgang zur Verfügung, muss eine Entscheidung getroffen werden, in welches Belegungsintervall der kritische Vorgang verschoben werden sollte. Durch eine ungünstige Wahl kann sich die Lösungsgüte verschlechtern, sodass die lokale Suche zu keiner Verbesserung führt. Der

Mechanismus der Reallokation ermöglicht es der lokalen Suche, ausgehend von einem Ausgangszustand mehrere Belegungsintervalle für einen kritischen Vorgang auszuprobieren, um die beste Allokation auszuwählen. Entspricht das Ergebnis durch die Verschiebung eines kritischen Vorgangs nicht den Anforderungen, wird die Ausgangssituation durch ein *Rollback* wiederhergestellt, siehe Abschnitt 4.5.5. Im Folgenden wird zuerst auf unterschiedliche Gütekriterien zur Bewertung, ob ein Zustand tolerierbar ist und als neuer Ausgangszustand der lokalen Suche genutzt wird, eingegangen. Anschließend wird die maximale Anzahl der Reallokationen untersucht.

Damit die lokale Suche in der Lage ist, lokale Optima zu überbrücken, kann auch eine Verschlechterung der Lösung toleriert werden. Entweder wird mit statischen oder dynamischen Toleranzwerten gearbeitet, die sich mit der Anzahl der Iterationen verändern.

Abbildung 5.19 stellt die Ergebnisse für konstante und dynamische Toleranzwerte dar. Im Fall der dynamischen Toleranzwerte können diese über einen konstanten Faktor oder über *Simulated Annealing* ermittelt werden. Im Fall des konstanten Faktors wird der Folgewert auf jeweils 80% des aktuellen Werts reduziert. Im Fall von *Simulated Annealing* wird die Wahrscheinlichkeit, mit der eine Verschlechterung akzeptiert wird, auf 0,25 festgesetzt und unterschiedliche Verringerungsraten von 0,4 bis 0,9 betrachtet. Grafik (a) stellt das Hypervolumen abhängig von der Toleranz dar. Ein negativer Wert erlaubt eine Verschlechterung, ein positiver Wert setzt eine Mindestverbesserung in jeder Iteration voraus. Es ist ersichtlich, dass mit einer Toleranz von 0 die breiteste Pareto-Front erzielt wird. Werden deutliche Verschlechterungen erlaubt, bspw. -0,01, sinkt das Hypervolumen erkennbar. Die Verschlechterung, die in einer Iteration erzielt wird, kann somit in Folgeiterationen nicht ausgeglichen werden. Für die Werte 0,005 und 0,01 sinkt der Zielfunktionswert erneut erkennbar. Dies weist darauf hin, dass die Mindestanforderung zu strikt gewählt wurde. Grafik (b) zeigt die Auswertung nach dem gleichgewichteten Zielfunktionswert. Auch diese Grafik belegt, dass eine statische Toleranz von 0 zu sehr guten Ergebnissen führt. Ist die Anforderung an eine Verbesserung hoch (0,005 oder 0,01), sinkt die Zielerreichung deutlich. Grafik (c) stellt den Anteil der Pareto-Lösungen an der Menge der Pareto-Lösungen dar, die durch die lokale Suche gefunden wurden. Während bis 0,0005 durchschnittlich über 80% der Pareto-Lösungen durch die lokale Suche gefunden wurden, sinkt der Wert auf unter 25% für 0,005 und 0,01. Diese Auswertung zeigt klar, dass bei der richtigen Wahl der Toleranz die lokale Suche die Pareto-Front deutlich erweitert. Grafik (d) fasst die Ergebnisse für dynamische Toleranzwerte zusammen. *Simulated Annealing* führt bei einer geeigneten Wahl der Verringerung der Akzeptanz zu guten Ergebnissen für das Hypervolumen, kann aber die Werte des besten statischen Toleranzwertes nicht erreichen. Ist die Verringerung zu groß (0,8 und 0,9), sinkt das erreichte Hypervolumen deutlich. Die Anpassung des Toleranzwertes um einen konstanten Faktor, ausgehend von 0,01 und 0,02

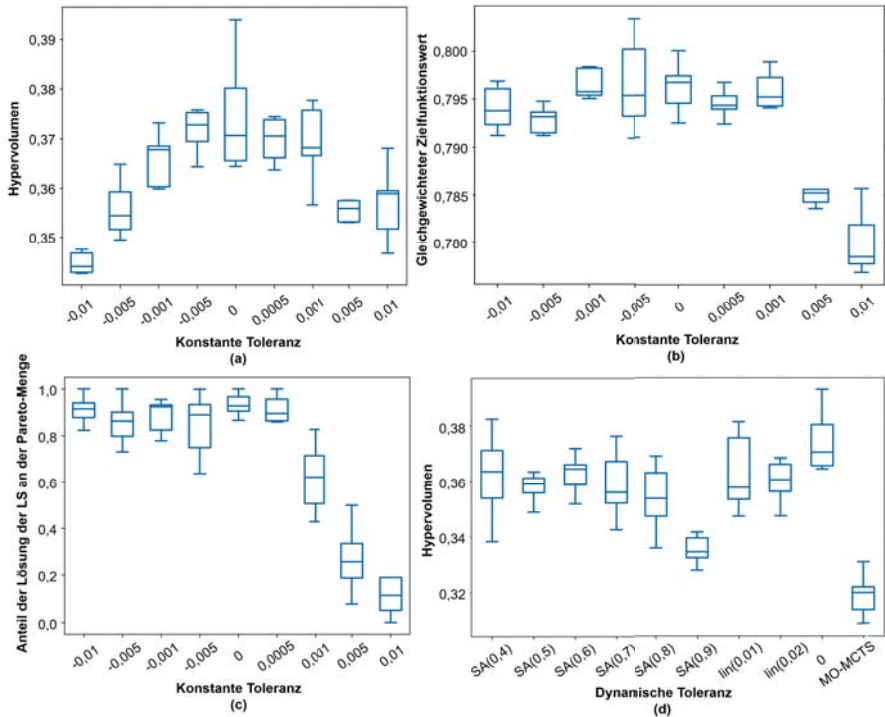


Abbildung 5.19: Auswertung statischer Toleranzwerte nach Hypervolumen (a), gleichgewichteter Zielfunktionswert (b) und Anteil der Pareto-Lösungen durch lokale Suche an der Pareto-Menge (c). Auswertung dynamischer Toleranzwerte nach dem Hypervolumen (d), angelehnt an (A_Liu 2021)

(lin(0,01) und lin(0,02)), führt zu vergleichbaren Ergebnissen mit *Simulated Annealing*.

Die Experimente zeigen, dass eine statische Toleranz von 0 zu sehr guten Ergebnissen führt und gleichzeitig keine zusätzliche Komplexität in das Gesamtsystem einbringt.

Die **Reallokation** erlaubt es der lokalen Suche, ausgehend von einem Ausgangszustand mehrere Belegungsintervalle zu evaluieren, um die beste Option zu identifizieren. Die Anzahl der Reallokationen gibt an, wie viele Belegungsintervalle maximal geprüft werden, die Anzahl Iterationen gibt an, nach wie vielen erfolgreichen Tauschoperationen das Verfahren endet (Abbruchkriterium).

Abbildung 5.20 zeigt das Hypervolumen abhängig vom Abbruchkriterium, der Auswahl der

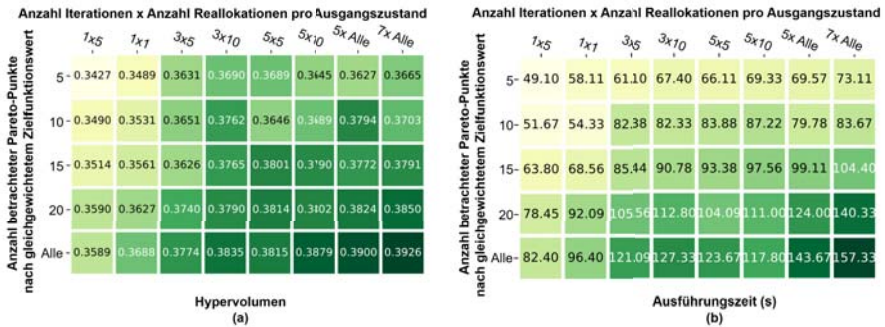


Abbildung 5.20: Hypervolumen (a) und Ausführungszeit (b) abhängig von der Anzahl Iterationen und Reallokationen sowie der Anzahl betrachteter Pareto-Punkte, angelehnt an (A_Liu 2021).

Ausgangslösung und der Anzahl erlaubter Reallokationen. Als Abbruchkriterium, siehe Abschnitt 4.5.6, wird *Best Improvement* kombiniert mit einer Iterationsbeschränkung angewendet. Die Auswahl der Ausgangslösungen, siehe Abschnitt 4.5.1, erfolgt nach dem BPA-Kriterium, bei dem eine feste Anzahl Pareto-Punkte nach dem gleichgewichteten Zielfunktionswert am Ende des MO-MCTS-Laufs ausgewählt wird. Verwendung findet der modifizierte LSONE-Ansatz. Die Anzahl Iterationen legt fest, wie oft die lokale Suche ausgehend vom verbesserten Zustand der Voriteration erneut begonnen wird. Die Anzahl Reallokationen gibt an, wie viele unterschiedliche Veränderungen ausgehend von einem Zustand evaluiert werden. Diese Dimension (Horizontale der Abbildung 5.20) zeigt, wie umfassend die Nachbarschaft um eine Ausgangslösung abgesucht wird. Die Anzahl untersuchter Pareto-Punkte (Vertikale der Abbildung 5.20) bestimmt, wie viele verschiedene Terminalzustände als initialer Ausgangszustand der lokalen Suche dienen. Diese Dimension gibt die Breite der Suche an.

Die Abbildung zeigt deutlich, dass das Hypervolumen steigt, je intensiver die Nachbarschaft hinsichtlich Tiefe und Breite erkundet wird. Der Vergleich entlang der Achsen belegt allerdings, dass eine tiefe Suche (7xAlle) auf nur fünf Pareto-Lösungen bessere Ergebnisse liefert als eine breite Suche auf allen Pareto-Punkten mit nur einer Iteration und ohne Möglichkeit zur Reallokation. Es wird daher empfohlen, eine umfassende Suche auf wenigen Ausgangslösungen vorzuziehen. Gleichzeitig wächst die Ausführungszeit in dieser Dimension langsamer als bei einer flacheren Suche basierend auf einer größeren Zahl an Ausgangslösungen.

5.3.4.6 Kompensation nicht-optimaler MO-MCTS-Parameter durch lokale Suche zur breiten Abdeckung des Lösungsraums

Die bisherigen Auswertungen zeigen, dass die Mechanismen der Aktionsgenerierung einen großen Einfluss auf die Lage der Pareto-Punkte im Lösungsraum haben. Im folgenden Abschnitt wird untersucht, inwieweit die lokale Suche einen MO-MCTS Basis-Optimierer ausgleichen kann, der den Lösungsraum nur selektiv absucht.

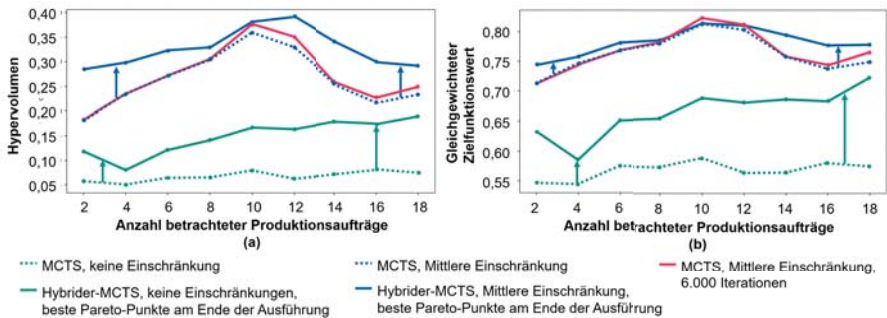


Abbildung 5.21: Hypervolumen und gleichgewichteter Zielfunktionswert abhängig von den Filterkombinationen zur Einschränkung zulässiger Aktionen und der Anzahl parallel betrachteter Produktionsaufträge, angelehnt an (A_Liu 2021)

Abbildung 5.21 zeigt das durchschnittlich erreichte Hypervolumen (a) und den gleichgewichteten Zielfunktionswert (b) abhängig von der Anzahl betrachteteter Produktionsaufträge für unterschiedlich starke Einschränkungen des Lösungsraums anhand des 18x8 Problems. Verglichen werden zwei MO-MCTS-Konfigurationen mit ihrem hybriden Pendant und einer reinen MO-MCTS-Ausführung mit 6.000 statt 2.000 Iterationen. Es wird deutlich, dass die lokale Suche zu einer starken Verbesserung führt, wenn der MO-MCTS die Lösungsmenge nicht einschränkt. Das liegt an der insgesamt sehr geringen Leistungsfähigkeit des MO-MCTS in dieser Konfiguration. Bei einem besser abgestimmten MO-MCTS mit mittlerer Einschränkung führt die lokale Suche weiterhin zu einer Verbesserung. Allerdings ist diese Verbesserung in Bereichen minimal, in denen die Anzahl gleichzeitig betrachteter Produktionsaufträge passend gewählt ist. Diese Beobachtung lässt den Schluss zu, dass die lokale Suche bei einem gut eingestellten MO-MCTS-Verfahren nur geringe Verbesserungen ermöglicht. Ist der MO-MCTS jedoch nicht passend konfiguriert, kann die lokale Suche dies teilweise kompensieren, sodass der Lösungsraum gut abgedeckt wird. Diesen Effekt sieht man im Randbereich für große und kleine Auftragslimits besonders deutlich.

Abbildung 5.22 stellt die Lage der Lösungen im Lösungsraum für die Ziele Durchlaufzeit,

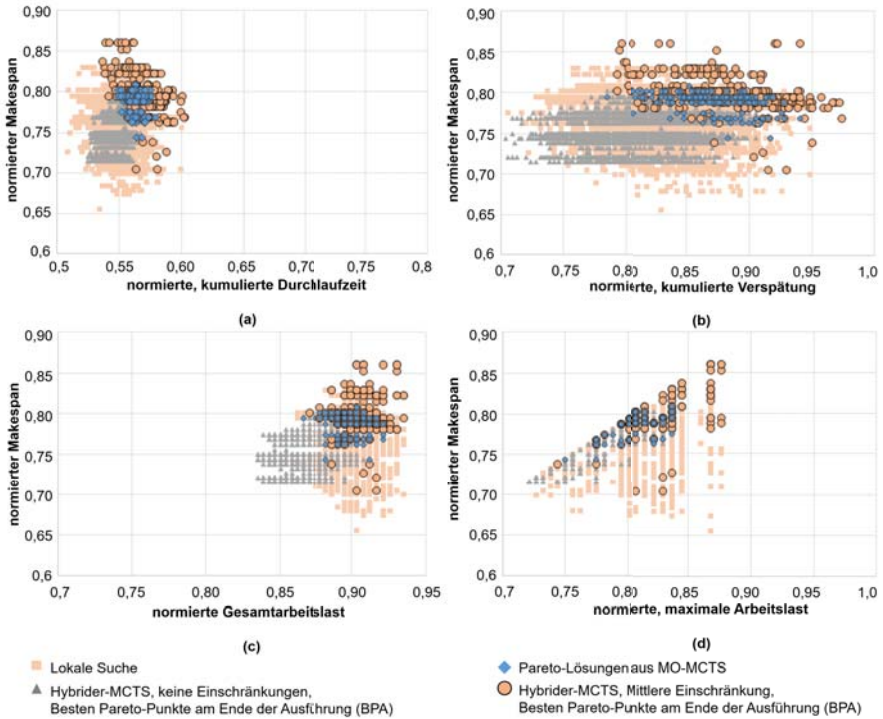


Abbildung 5.22: Lage der MO-MCTS-Lösungen und der Lösungen der lokalen Suche im Lösungsraum für Makespan, Durchlaufzeit, Verspätung, maximale Arbeitslast und Gesamtarbeitslast, angelehnt an (A_Liu 2021)

Makespan, Verspätung, Gesamtarbeitsmenge und maximale Arbeitslast für verschiedene hybride Ansätze sowie den MO-MCTS-Basisalgorithmus dar. Die Punkte der lokalen Suche (Quadrat) weisen eine große Streuung auf. Der hybride Ansatz mit MO-MCTS ohne Einschränkung der zulässigen Lösungsmenge (Dreieck) schneidet schlechter ab als der Basis-MO-MCTS-Algorithmus (Raute) und der hybride Ansatz auf Basis des MO-MCTS-Algorithmus mit mittlerer Einschränkung der zulässigen Lösungsmenge (Kreis). Die Startlösungen für die lokale Suche sind beim MO-MCTS ohne Einschränkungen deutlich schlechter, sodass auch durch die lokale Suche kein hohes Lösungsniveau erreicht werden kann. Wird lokale Suche auf die Ausgangslösungen des MO-MCTS mit mittlerer Einschränkung angewendet, kommt es zu einer Verbesserung in allen Zielkriterien. Die Pareto-Punkte dieses hybriden Ansatzes liegen über den Pareto-Punkten des MO-MCTS-Verfahrens.

Bezogen auf die Lösungsgüte ist für das hybride System festzustellen, dass die lokale Suche in der Lage ist, die Effekte einer nicht-optimalen Wahl der Parameter zu dämpfen. Die Ergebnisse aus Abschnitt 5.4.4 zeigen, dass lokale Suche bei einem optimal eingestellten MO-MCTS zu keiner deutlichen Verbesserung führt.

Zusammenfassend lässt sich festhalten, dass lokale Suche die Breite der Paretofront erhöht. Die angepasste LSONE-Nachbarschaftsstruktur führt zu den besten Ergebnissen. Durch die Möglichkeit der Reallokation kann die Lösungsgüte gesteigert werden. Sowohl die Toleranz von Verschlechterungen als auch eine Mindestanforderung an die Lösungsgüte führen zu einer Verschlechterung bei der Reallokation. Die Ergebnisse zeigen, dass eine umfassende Suche ausgehend von wenigen Ausgangslösungen zu empfehlen ist. Hierzu eignet sich das *Sampling* aus den Pareto-Punkten am Ende der MO-MCTS-Ausführung oder eine zufällige Wahl der Terminalzustände am Ende des *Rollouts*.

5.3.5 Zusammenfassung

Die Auswertungen zeigen, dass die Einschränkung des Aktionsraums einen großen Einfluss auf das Baumwachstum des Suchbaums und damit den abgebildeten Lösungsraum hat. Je nach gewähltem Ansatz für die Verringerung der Aktionen werden unterschiedliche Bereiche des Lösungsraums erreicht. Durch Parallelisierung und Domänenwissen können zeitgleich unterschiedliche Bereiche des Lösungsraums erkundet werden. Alternativ kann durch die sequentielle Änderung der Zielgewichtung in der Selektionsfunktion der Suchbaum in anderen Bereichen des Lösungsraums aufgebaut werden. Beide Ansätze erfordern genaues Domänenwissen über das Verhalten des Optimierers auf dem gegebenen Steuerungsproblem. Das alternative Vorgehen zur intelligenten Einschränkung zulässiger Aktionen findet durch selbstständige Anpassung geeignete Filterkombinationen, um die Aktionen so einzuschränken, dass ohne vorherige manuelle Abstimmung der Steuerung durch Domänenexperten eine breite Pareto-Front mit guter Abdeckung des Lösungsraums erzielt wird. Lokale Suche basierend auf einem modifizierten LSONE führt ebenfalls zu einer breiteren Pareto-Front. Die Verbesserung durch die lokale Optimierung fällt stärker aus, wenn das MO-MCTS-Basisverfahren ungünstig parametrisiert ist. Im Sinne der Forschungsfrage **F1** kann gefolgert werden, dass der MO-MCTS mit den entwickelten Anpassungen für die Umsetzung einer vorausschauenden Mehrzieloptimierung für die Matrixproduktion geeignet ist. Der Aspekt der Reaktivität der Steuerung wird im folgenden Abschnitt untersucht. Die Methode zur intelligenten Einschränkung zulässiger Aktionen führt zudem dazu, dass sich die Produktionssteuerung selbstständig an das vorliegende Steuerungsproblem und die

angestrebten Ziele anpasst. Daher leistet diese Methode ebenfalls einen wichtigen Beitrag zur positiven Beantwortung der Forschungsfrage **F2**.

5.4 Anstieg der Lösungsgüte

In diesem Kapitel wird der Anstieg der Lösungsgüte über die Iterationen untersucht. Erstrebenswert ist ein möglichst schneller Anstieg der Lösungsgüte. Die Entwicklung der Lösungsgüte ist im Rahmen der Forschungsfrage **F1** von besonderem Interesse, da bei einer reaktiven Steuerung die Lösungsgüte schnell ein hohes Niveau erreichen sollte.

Die Entwicklung der Lösungsgüte wird maßgeblich durch den Explorationsfaktor C der Selektionsfunktion bestimmt. Im Gegensatz zur Aktionsgenerierung schränkt der C -Wert den Lösungsraum nicht ein, sondern beeinflusst den zeitlichen Verlauf der Erkundung.

5.4.1 Einfluss der Selektionsfunktion

Eine Möglichkeit zur Steuerung des Suchfokus durch die Selektionsfunktion wurde in Abschnitt 5.3.3 am Beispiel alternierender Zielgewichtungen dargestellt. Im Folgenden stehen die Auswirkungen des Explorationsfaktors C auf die Entwicklung der Lösungsgüte im Mittelpunkt. Betrachtet werden statische C -Werte und dynamische C -Werte.

5.4.1.1 Statischer Explorationsfaktor

Durch die Wahl des C -Werts $C > 0$ wird der zeitliche Verlauf der Erkundung beeinflusst. Zeitbezogene Zielkriterien, wie die Durchlaufzeit oder die Verspätung, reagieren sensibler auf die Wahl des C -Wertes, da die Einzelentscheidungen der Steuerung hier einen größeren Einfluss haben als bei Makespan oder Auslastung. Zur Demonstration der Auswirkung des C -Werts wird die Lösungsgüte für die Zielgröße Verspätung über 70.000 Iterationen untersucht, siehe Tabelle 5.11 (A_Funk 2019).

Grafik (a) der Abbildung 5.23 zeigt den Verlauf der durchschnittlichen Zielerreichung aus je 25 Experimenten über die Iterationen. Der asymptotische Verlauf der Ergebnisse ist klar ersichtlich. Es bestehen zudem große Unterschiede im Anstieg der Lösungsgüte und geringere Unterschiede in der absolut erreichten Güte der Ergebnisse. Zuerst wird deutlich, dass kleine C -Werte zu einem schnellen Anstieg der Lösungsgüte führen, da sie die Suche im Lösungsbereich mit hoher Güte verstärken. Die absolute Lösungsgüte kleiner C -Werte

Tabelle 5.11: Konfiguration der Experimente zur Untersuchung des Explorationsfaktors

Problem- instanz	Terminierungs- kriterium	Ziel- funktion	Aktionsgenerierung (1. Sortierkriterium, Filterkombinationen)	C-Wert
12x5	70.000 Iterationen	Verspätung	Kürzeste Bearbeitungszeit, Mittlere Einschränkung	~

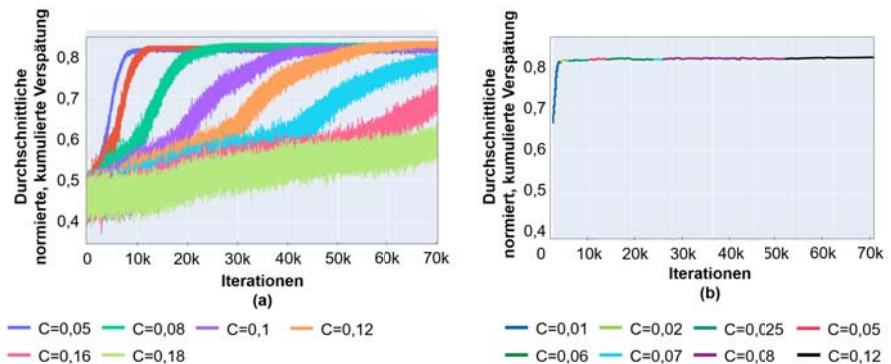


Abbildung 5.23: (a) Entwicklung der Lösungsgüte über 70.000 Iterationen für die Optimierung der kumulierten Verspätung für die 12x5 Probleminstanz. (b) Bester C-Wert abhängig von der Iterationszahl, angelehnt an (A_Funk 2019)

stagniert jedoch mit zunehmender Iterationszahl und bleibt leicht hinter den Ergebnissen der Konfigurationen mit höherer Exploration zurück. Das geringere Lösungsniveau ist darin begründet, dass das MO-MCTS-Verfahren in einem lokalen Optimum gefangen ist. Als zweites fällt die zunehmende Streuung der Lösungsgüte mit steigender Exploration auf. Dies zeigt der Vergleich zwischen den Ergebnissen für den Explorationsfaktor $C = 0,05$ (geringe Streuung) und beispielsweise den Ergebnissen, erzielt mit einem Explorationsfaktor von $C = 0,18$ (größere Streuung). Die Zunahme der Streuung hängt mit der erweiterten Exploration zusammen. Diese bedingt, dass auch unattraktive, aber neue Bereiche des Lösungsraums erschlossen werden.

Grafik (b) der Abbildung 5.23 stellt die beste Wahl eines statischen Explorationsfaktors abhängig von der Iterationszahl dar. Für diese Auswertung wurden kleinere Inkremente des C -Werts gewählt, was sich insbesondere in den ersten 10.000 Iterationen niederschlägt. Die Auswertung zeigt, dass die Wahl des Explorationsfaktors von der geplanten Iterationszahl abhängt und kein statischer Wert stets zu guten Ergebnissen führt.

5.4.1.2 Dynamischer Explorationsfaktor

Ein dynamischer Explorationsfaktor verändert sich über die Iterationen. In Abschnitt 4.3.2 wurden Ansätze zur linearen Veränderung des Explorationsfaktors vorgestellt. Durchgeführte Experimente zum abnehmenden Explorationsfaktor zeigen, dass durch die geringe Exploration in späteren Iterationen der Algorithmus kaum noch die Fähigkeit besitzt, lokale Optima zu überbrücken. Bei linear wachsendem Explorationsfaktor kommt es nach einem schnellen Anstieg der Lösungsgüte zu einem Abflachen, da das Verfahren die relevanten Bereiche des Lösungsraums zunehmend seltener besucht. Aufgrund dieser Beobachtungen wird im Folgenden nur der kombinierte Ansatz aus Abschnitt 4.3.2.2 vorgestellt.

Der in Abschnitt 4.3.2.2 beschriebene Ansatz reiht Phasen der linearen Absenkung des C -Werts (Inkrement) aneinander. Wird ein Terminalzustand in der Expansionsphase erreicht und damit der Suchbaum lokal vollständig aufgebaut, endet diese *greedy*-Phase und es kommt zu einer sprunghaften Erhöhung des Explorationsfaktors, um das lokale Optimum zu verlassen.

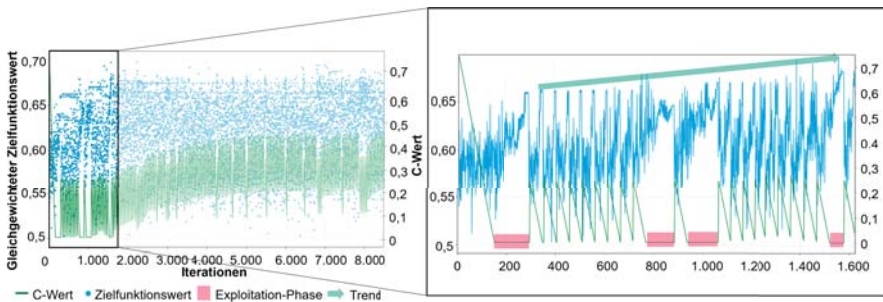


Abbildung 5.24: Verlauf des Zielfunktionswerts und des dynamischen C -Werts über 8.000 Iterationen für die 12×5 Problem Instanz, angelehnt an (A_Deininger 2020)

Abbildung 5.24 zeigt den Verlauf des normierten gleichgewichteten Zielfunktionswerts und des dazugehörigen C -Werts. Es ist erkennbar, dass der Zielfunktionswert in *Exploitation*-Phasen mit niedrigem C -Wert nahe oder gleich null stark ansteigt. Durch den sprunghaften Anstieg kann über die Iterationen eine Erhöhung des Zielfunktionswerts erreicht werden. Dieses Verhalten ist am Sägezahnmuster des C -Werts erkennbar. Mit zunehmender Iterationszahl zeigt sich auch eine Erhöhung des durchschnittlichen C -Werts. Dies belegt, dass das Verfahren nur noch selten Bereiche im Suchbaum findet, die nicht aufgebaut sind.

Im Folgenden sollen systematisch die Auswirkungen des initialen C -Werts, der Erhöhungskonstante und des Inkrements untersucht werden. Tabelle 5.12 stellt die Werte beziehungsweise

die Wertebereiche der einzelnen Parameter dar.

Tabelle 5.12: Sensitivitätsanalysen des dynamischen C-Werts

Analyse	Initialer C-Wert	Inkrement	Erhöhung
Initialer Wert	[0,1]	-0,005	0.25
Erhöhung	0.25	-0,005	[0,1]
Inkrement	0.25	[-0.001,-0.01]	0.25

Abbildung 5.25 zeigt die Entwicklung des durchschnittlichen Verzweigungsfaktors und des durchschnittlichen C -Werts abhängig von der Wahl des initialen C -Werts, des Inkrements und der Erhöhungskonstante. Der initiale C -Wert hat keinen klar gerichteten Einfluss auf den Verlauf des durchschnittlichen Verzweigungsfaktors und des durchschnittlichen C -Werts. Die Grafiken (a) und (b) zeigen vergleichbare Werte für den gesamten untersuchten Wertebereich von 0 bis 1. Dieses Verhalten liegt darin begründet, dass der initiale C -Wert lediglich den Beginn des Verfahrens beeinflusst. Sobald das erste Mal ein Terminalzustand in der Expansionsphase erreicht wurde, ist das weitere Verhalten unabhängig vom initialen C -Wert. Einschränkend sei vermerkt, dass bei geringen Iterationszahlen und hohen initialen C -Werten der Effekt des dynamischen Verfahrens ausbleibt. In dieser speziellen Konstellation verhält sich dieses Verfahren analog zur linearen Anpassung des C -Werts.

Die Grafiken (e) und (f) belegen den Einfluss der Wahl des Inkrements auf den Verzweigungsfaktor und den durchschnittlichen C -Wert. Hier zeigt sich ein vergleichbares Bild. Auch die Wahl des Inkrements ändert das grundlegende Verhalten des Verfahrens nicht.

Die Wahl der Erhöhungskonstante zeigt jedoch eine deutlich positive Korrelation mit dem Verzweigungsfaktor, siehe Grafik (i). Je höher die Erhöhungskonstante, umso stärker steigt die durchschnittliche Zahl der Kindknoten pro Zustand. Ist die Erhöhungskonstante 0, wird der Mechanismus ausgehebelt und es liegt eine lineare Anpassung des C -Werts vor. Der Verzweigungsfaktor scheint mit steigenden C -Werten zu konvergieren. Dies ist erwartbar, da der maximal möglich Verzweigungsfaktor durch das Steuerungsproblem beschränkt ist. Der durchschnittliche C -Wert wächst linear mit zunehmender Erhöhungskonstante, siehe Grafik (j). In späten Iterationen ist der Suchbaum bereits stärker erkundet und das Suchverfahren findet nur selten Bereiche, in denen noch viele Iterationen möglich sind, bevor ein Terminalzustand erreicht wird. Der resultierende Anstieg des C -Werts ist im Beispiel in Abbildung 5.24 zu beobachten. Ist die Erhöhungskonstante 0, nähert sich der durchschnittliche C -Wert dem Wert 0 an, da kein Mechanismus existiert, den Wert zu erhöhen.

Grafiken (c),(g),(k) und (d),(h),(l), zeigen das Hypervolumen und die Zielerreichung für die Einzelziele kumulierte Verspätung, kumulierte Durchlaufzeit und Makespan sowie für das

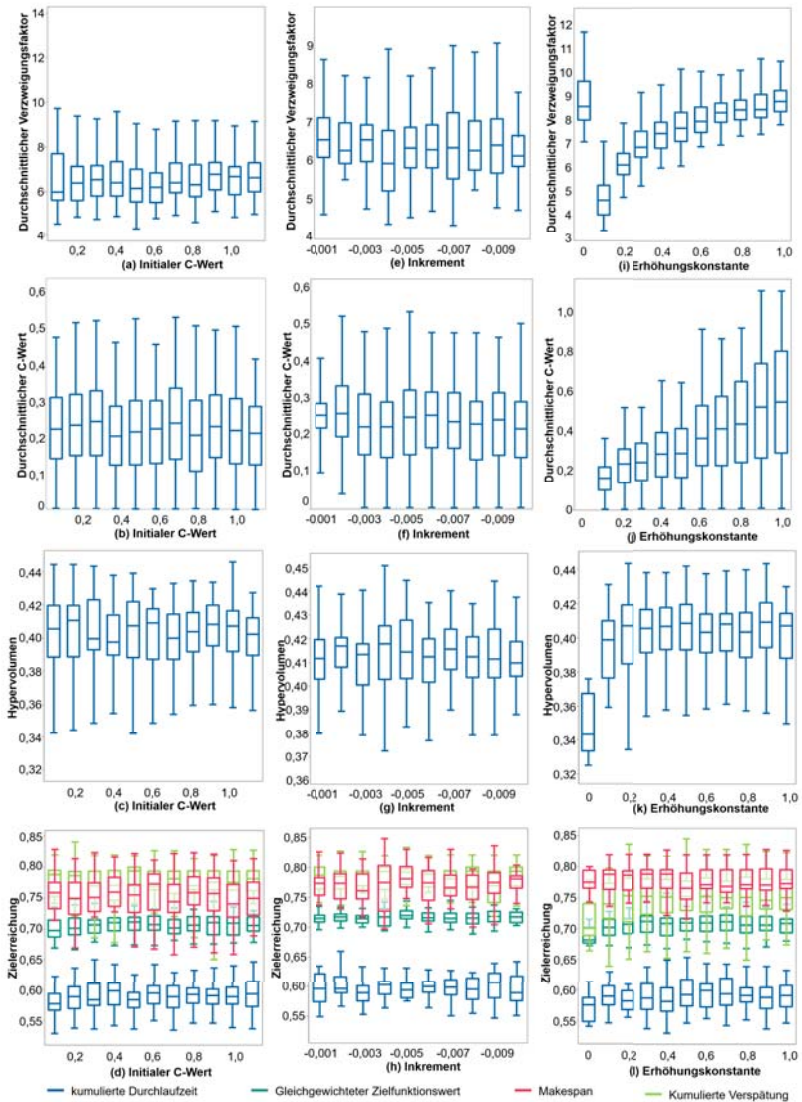


Abbildung 5.25: Verzweigungsfaktor, C-Wert, Hypervolumen und Zielerreichung für die 12x5 Probleminstanz, angelehnt an (A_Deinger 2020)

gleichgewichtete Mittel aus den drei Zielkriterien. Grafik (k) belegt, dass das Hypervolumen deutlich kleiner ist, wenn anstelle der dynamischen Anpassung des C -Werts nur eine lineare Verringerung vorliegt (Erhöhung = 0). In den anderen Fällen sind die Ergebnisse für die betrachteten Parameter des dynamischen Ansatzes vergleichbar. Insgesamt lässt sich daraus schließen, dass die dynamische Wahl des C -Werts einen positiven Einfluss aufweist und zudem robust gegenüber der Wahl der Parameter des Verfahrens ist.

5.4.2 Einfluss der intelligenten Einschränkung der Aktionen auf den Anstieg der Lösungsgüte

Die Einschränkung der zulässigen Aktionsmenge kann dazu führen, dass vor allem vielversprechende Aktionen verbleiben. Dadurch steigt die Lösungsgüte über die Iterationen schneller an, verglichen mit einer Aktionsmenge, die auch schlechtere Aktionen enthält. Im Weiteren soll untersucht werden, wie sich die intelligente Einschränkung der Aktionsmenge auf die Entwicklung der Lösungsgüte auswirkt. Für die Auswertung wurden die Experimente gemäß Tabelle 5.7 durchgeführt.

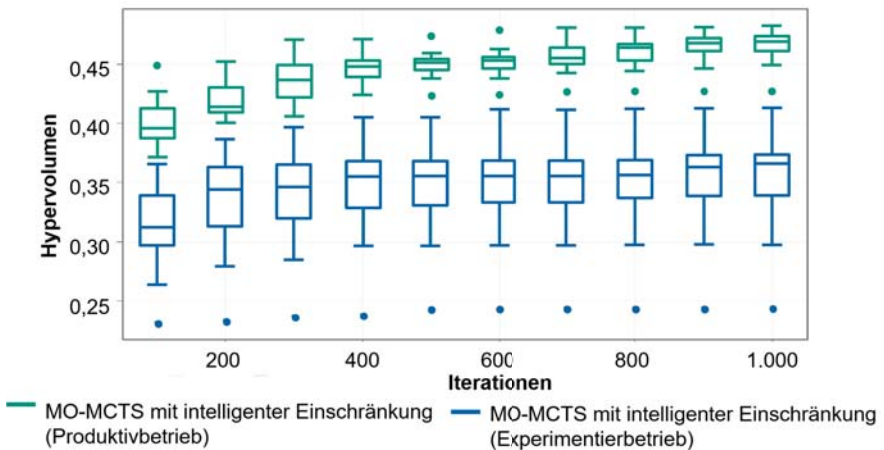


Abbildung 5.26: Entwicklung des Hypervolumens während des Experimentierbetriebs und im Produktivbetrieb (12x5 Problem Instanz), angelehnt an (A_Deiningger 2020)

Abbildung 5.26 zeigt die Entwicklung des Hypervolumens während des Experimentierbetriebs mit 20 Ausführungen zu je 1.000 Iterationen und im Produktivbetrieb. Durch die automatisierte Auswahl passender Filterkombinationen wird zu jedem Zeitpunkt ein deutlich höheres

Hypervolumen bei gleichzeitig geringerer Streuung erzielt. Gemessen an dem erreichten Hypervolumen von 0,44 nach 10.000 Iterationen ohne die intelligente Einschränkung des Aktionsraums, siehe Anhang A6.1, kann dieses Hypervolumen durch die intelligente Einschränkung des Aktionsraums bereits nach durchschnittlich etwa 350 Iterationen erreicht werden. Dies entspricht einer Beschleunigung der Suchgeschwindigkeit um den Faktor 28.

5.4.3 Einfluss der Parallelisierung auf den Anstieg der Lösungsgüte

In diesem Abschnitt werden die Effekte der Parallelisierung auf die Entwicklung der Lösungsgüte thematisiert. Untersucht werden die Root-Parallelisierung und die Subtree-Parallelisierung. Für die Root-Parallelisierung werden die Einstellung aus Tabelle 5.8 genutzt, allerdings mit einer Laufzeit von 5, 10 und 20 Sekunden. Für die Subtree-Parallelisierung wurden die Einstellung gemäß Tabelle 5.13 verwendet.

Tabelle 5.13: Untersuchung der Subtree-Parallelisierung

Problem- instanz	Terminierungs- kriterium	Zielfunktion Durchlaufzeit:Auslastung	Anzahl parallel betrachteter Produktionsaufträge	C-Wert
12x5	5, 10, 20 Sekunden	1:1	Kürzeste Bearbeitungszeit, mittlere Einschränkung	0,1

Abbildung 5.27 zeigt die Untersuchung der Parallelisierungsansätze, siehe Abschnitt 4.4. Drei Effekte sind erkennbar. Erstens steigt die Anzahl Iterationen pro Zeiteinheit um den Faktor 0,5, multipliziert mit der Anzahl paralleler MO-MCTS-Prozesse. Das bedeutet, dass zwei MO-MCTS-Prozesse genauso schnell sind wie ein einzelner, vier aber bereits doppelt so schnell. Diese Verlangsamung des einzelnen MO-MCTS-Prozesses liegt an dem gesteigerten Aufwand für die Zusammenführung der Ergebnisse. Dieser Effekt ist abhängig von der gewählten Implementierung, daher ist dieses Ergebnisse nicht verallgemeinerbar. Abbildung (a) zeigt die Anzahl Iterationen, realisiert durch die *Root*- und *Subtree*-Parallelisierung mit 8 virtuellen CPU-Kernen, verglichen mit einem MO-MCTS-Ansatz ohne Parallelisierung. Entsprechend der Erhöhung der Iterationen um den Faktor 0,5 ergibt sich für die acht Kerne eine Erhöhung der Iterationszahl etwa den Faktor 4. Zweitens kann in der Grafik (b) für die *Subtree*-Parallelisierung beobachtet werden, dass der durchschnittliche Zielfunktionswert des parallelisierten Systems um etwa 2.5% sinkt. Durch die Aufteilung in Unterbäume wird über die gesamte MO-MCTS-Ausführung auch eine Erkundung in ungünstigen Bereichen des Lösungsraums betrieben. Gleichzeitig sinkt die Geschwindigkeit pro MCTS-Prozess, sodass auch im vielversprechenden Bereich des Lösungsraums weniger Fortschritt erzielt wird. Die Aufteilung in Unterbäume hebt somit zum Teil die steuernde Wirkung der Selektionsfunktion

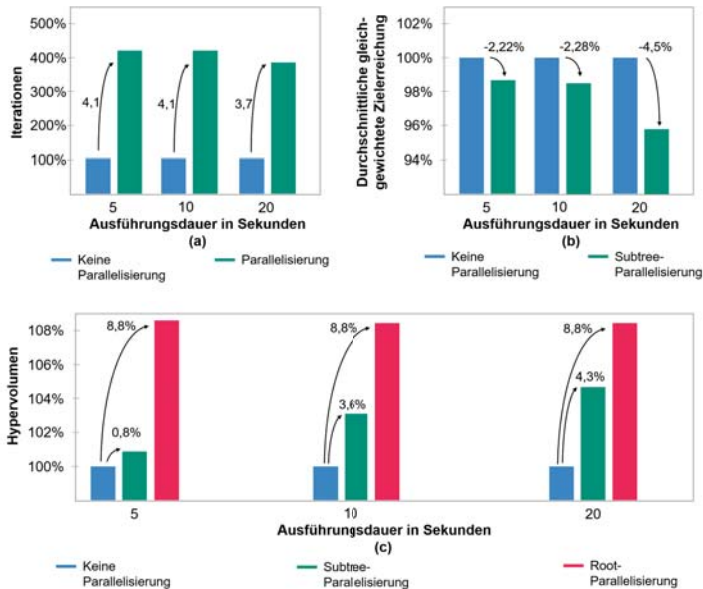


Abbildung 5.27: Anzahl der Iterationen mit und ohne Parallelisierung (a), durchschnittlicher gleichgewichteter Zielfunktionswert ohne Parallelisierung und mit Subtree-Parallelisierung (b), Hypervolumen ohne Parallelisierung, mit Subtree-Parallelisierung und mit Root-Parallelisierung (c) für 8 virtuelle CPU-Kerne, angelehnt an (A_Wurster 2019)

aus. Die dritte Beobachtung betrifft das Hypervolumen, Grafik (c). Durch die Aufteilung in Unterbäume wird auch durch die *Subtree*-Parallelisierung ein größerer Bereich des Lösungsraums untersucht, sodass das Hypervolumen ansteigt. Bei der *Root*-Parallelisierung steht weniger die Suchgeschwindigkeit als die Abdeckung des Lösungsraums im Fokus. Durch die unterschiedliche Ausrichtung der MO-MCTS-Varianten pro MCTS-Prozess kann der Lösungsraum breiter abgesucht werden, siehe Abbildung 5.13. Der Vergleich zwischen der *Root*-Parallelisierung, der *Subtree*-Parallelisierung und des MO-MCTS ohne Parallelisierung zu verschiedenen Ausführungszeitpunkten zeigt, dass das Hypervolumen der parallelisierten Ansätze den Wert des MO-MCTS ohne Parallelisierung übersteigt. Zudem kann durch die *Root*-Parallelisierung ein großer Teil des Lösungsraums erkundet werden.

Parallelisierung beschleunigt linear die Suchgeschwindigkeit. Aufgrund des exponentiellen Charakters des Steuerungsproblems ist Parallelisierung kein geeignetes Mittel, um eine hohe Lösungsgüte auch bei komplexen Problemen sicherzustellen. Hierfür sind Ansätze der

Komplexitätsreduktion durch Einschränkung des Lösungsraums besser geeignet.

5.4.4 Einfluss lokaler Suche auf den Anstieg der Lösungsgüte

Lokale Suche bewirkt durch eine Post-Optimierung eine Erhöhung der Lösungsgüte, aber auch des Rechenaufwands. Im Folgenden wird der Anstieg der Lösungsgüte des hybriden Verfahrens mit Postoptimierung mit den Ergebnissen ohne die Postoptimierung durch lokale Suche verglichen. Um den Effekt der Komplexitätssteigerung durch die lokale Suche abzubilden, erfolgt die Auswertung zeitbezogen und nicht iterationsbezogen.

Die Experimente werden auf der 18x8 Probleminstanz gemäß Tabelle 5.14 durchgeführt. Um den Kompensationseffekt einer ungünstigen MO-MCTS-Konfiguration aufzuzeigen, werden zwölf Produktionsaufträge in der Aktionsgenerierung parallel betrachtet. Die Auswertungen aus Abschnitt 5.3.1.2 haben gezeigt, dass zehn parallel betrachtete Produktionsaufträge für dieses Steuerungsproblem zu besseren Ergebnissen führen.

Tabelle 5.14: Untersuchung des Einflusses der lokalen Suche auf den Anstieg der Lösungsgüte

Problem- instanz	Terminierungs- kriterium	Zielfunktion Durchlaufzeit:Auslastung	Anzahl parallel betrachteter Produktionsaufträge	C-Wert
18x8	140 Sekunden	Termintreue Durchlaufzeit Makespan Gesamtarbeitsmenge Maximale Arbeitslast	Kürzeste Bearbeitungszeit Mittlere Einschränkung 12 parallel betrachtete Produktionsaufträge	0,25

Abbildung 5.28 zeigt den Verlauf des Hypervolumens über die Ausführungszeit für zwei MO-MCTS-Algorithmen und einen hybriden Algorithmus, aggregiert für zehn Experimente. Das Basisoptimierungsverfahren des hybriden Algorithmus ist der MO-MCTS mit zwölf parallel betrachteten Produktionsaufträgen und mittlerer Einschränkung der zulässigen Lösungsmenge (blau). Der hybride Ansatz nutzt die Lösungen dieses MO-MCTS als Ausgangspunkt für die Optimierung durch lokale Suche. Der hybride MO-MCTS (grün) führt nach fünf Sekunden MO-MCTS jeweils fünf Sekunden lang eine lokale Suche auf den besten Pareto-Punkten (BPI) durch. Als dritte Referenz ist das erreichte Hypervolumen für einen MO-MCTS mit zehn gleichzeitig betrachteten Produktionsaufträgen am Ende der Ausführungszeit (rot) als Referenz eingezeichnet. Dieses Beispiel verdeutlicht, dass die lokale Suche in der Lage ist, den Effekt einer sub-optimalen Einschränkung des Lösungsraums, hier mit zwölf parallel betrachteten Produktionsaufträgen, teilweise zu kompensieren. Der hybride Ansatz führt zu einem höheren Hypervolumen über die gesamte Laufzeit. Vergleichbar mit den anderen

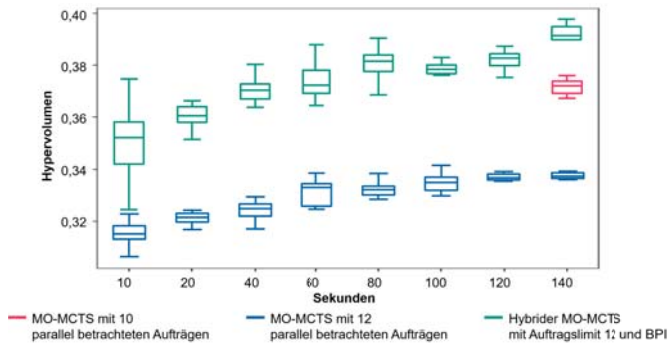


Abbildung 5.28: Entwicklung des Hypervolumens über die Laufzeit für den Basisalgorithmus und die hybride Version für das 18x8 Problem, angelehnt an (A_Liu 2021)

Ergebnissen ist auch in diesem Fall die Streuung größer als bei reinem MO-MCTS. Das hybride Verfahren erzielt zudem ein besseres Ergebnis als der gut konfigurierte MO-MCTS mit mittlerer Einschränkung und einem Auftragslimit von zehn Aufträgen (rot). Trotz des höheren Rechenaufwands für die lokale Suche beschleunigt sich der Anstieg der Lösungsgüte.

Im Folgenden soll der Beitrag der lokalen Suche im Vergleich zu einem sehr gut eingestellten MO-MCTS untersucht werden. Hierfür wird der in Tabelle 5.14 beschriebene MO-MCTS mit intelligenter Einschränkung zulässiger Aktionen als Referenz genutzt. Alle zehn Sekunden wird auf der aktuellen Pareto-Menge zehn Sekunden lang eine lokale Suche durchgeführt (BPI). Die erste Lösung der lokalen Suche steht daher nach 20 Sekunden zur Verfügung. Im Vorfeld wird die intelligente Einschränkung zulässiger Aktionen anhand von 20 Trainingsdurchläufen mit jeweils 2.000 Iterationen parametrisiert.

Grafik (a) der Abbildung 5.29 zeigt die aggregierten Ergebnisse des Hypervolumens und des gleichgewichteten Zielfunktionswerts aus zehn Durchläufen über die Laufzeit.

Die Hypervolumen des hybriden Ansatzes und des MO-MCTS-Ansatzes mit intelligenter Einschränkung der Aktionen, der als Basisoptimierer für den hybriden Ansatz genutzt wird, zeigen einen ähnlichen Verlauf. Der hybride Ansatz zeigt ein leicht verbessertes Hypervolumen zu jedem Zeitpunkt und weist, wie bereits in den vorangegangenen Experimenten eine größere Streuung auf. Gegenüber dem ebenfalls hybriden MO-MCTS, der die mittlere Einschränkung und ein ungünstigeres Auftragslimit von zwölf parallel betrachteten Produktionsaufträgen in der Aktionsgenerierung nutzt (rot), zeigt sich jedoch eine deutliche Verbesserung. Der Vergleich gegenüber demselben MO-MCTS jedoch ohne Postoptimierung durch lokale Suche (hellgrün) fällt die Verbesserung noch deutlicher aus.

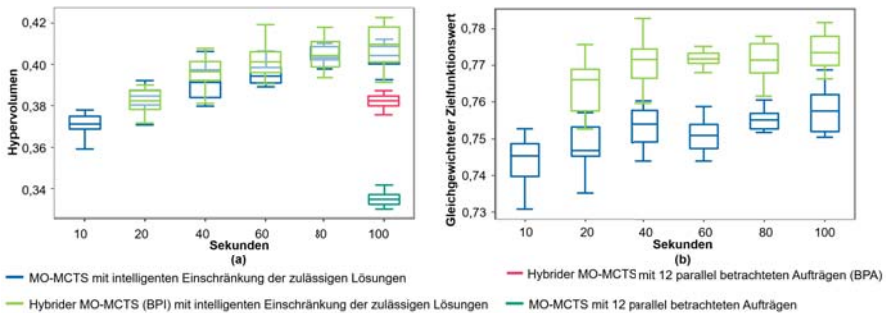


Abbildung 5.29: Hypervolumen und Zielerreichung über die Laufzeit für hybride Ansätze mit und ohne intelligente Einschränkung zulässiger Aktionen (18x8 Probleminstanz), angelehnt an (A_Liu 2021)

Die Auswertung nach dem gleichgewichteten Zielfunktionswert in Grafik (b) belegt eine klare aber leichte Steigerung über die gesamte Ausführungszeit. Die lokale Suche führt auch bei einem MO-MCTS mit intelligenter Einschränkung der Aktionen zu einer Beschleunigung des Anstiegs der Lösungsgüte.

5.4.5 Zusammenfassung

Die Auswertung zum Anstieg der Lösungsgüte haben gezeigt, dass kein statischer C -Wert optimal ist, da die Lösungsgüte für die geplante Iterationszahl entweder zu flach ansteigt oder das Verfahren lokale Optima nicht überwinden kann. Der vorgestellte dynamische Explorationsfaktor überkommt dieses Dilemma und führt zu einem schnelleren Anstieg der Lösungsgüte. Die detaillierten Untersuchungen haben eine geringe Abhängigkeit gegenüber den Parametern des Ansatzes zur dynamischen Regelung des C -Wertes gezeigt. Die Aktionsgenerierung führt durch die Einschränkung der zulässigen Aktionen dazu, dass besonders relevante Aktionen verbleiben, sodass ein schnellerer Anstieg der Lösungsgüte resultiert. Parallelisierung führt zu einer Beschleunigung des Anstiegs der Lösungsgüte, insbesondere bei der Root-Parallelisierung. Allerdings ist dieser Effekt nur linear und durch den zusätzlichen Aufwand zur Zusammenführung der Teilbäume und Teilergebnisse gedämpft. Obwohl die lokale Suche zusätzliche Rechenschritte mit sich bringt, kann der Anstieg der Lösungsgüte beschleunigt werden. Dieser Effekt fällt bei einem gut eingestellten MO-MCTS-Verfahren, das beispielsweise die intelligente Einschränkung der Aktionen nutzt, geringer aus. Im Sinne der Forschungsfrage **F1** konnte gezeigt werden, dass die entwickelten Methoden zur Anpassung

des C -Werts, zur intelligenten Einschränkung der Aktionen und die Hybridisierung des MO-MCTS den Anstieg der Lösungsgüte beschleunigen. Somit sind sie geeignet, die Reaktivität der Produktionssteuerung zu verbessern. Das Verfahren zur dynamischen Anpassung des C -Werts ermöglicht zudem eine automatisierte Anpassung an unterschiedliche Steuerungsprobleme. Es stellt daher zusammen mit der intelligenten Einschränkung der Aktionen einen weiteren Baustein zur positiven Beantwortung der Forschungsfrage **F2** dar.

5.5 Bestimmung der Lösungsgüte

In diesem Kapitel wird die Lösungsgüte anhand verschiedener Steuerungsaufgaben überprüft. Die genutzten Steuerungsprobleme unterscheiden sich sowohl in der Struktur des Produktionssystems als auch bei den genutzten Zielkriterien. Um die absolute Lösungsgüte zu bestimmen, wird zum einen der Vergleich zu den Lösungen eines exakten Verfahrens gezogen. Zum anderen werden die Lösungen des MO-MCTS-Optimierers und des hybriden Verfahrens mit anderen Ansätzen aus der Literatur anhand des Kacem-Benchmarks verglichen. Anschließend wird am Beispiel eines realen Anwendungsfalls aus der Getriebemontage das Verhalten der Steuerung bei Ausfällen von Produktionsmitteln und stochastischen Bearbeitungszeiten verdeutlicht.

5.5.1 Vergleich mit exakten Verfahren

Zur Ermittlung der absoluten Lösungsgüte werden die Lösungen des MO-MCTS-Verfahrens mit den Lösungen eines exakten Verfahrens verglichen. Das Steuerungsproblem wird hierfür als *Mixed Integer Linear Programming* Problem ähnlich den im Abschnitt 3.2.1 vorgestellten Arbeiten von Özgüven & Özbakır u. a. (2010) modelliert. Der Ansatz von Özgüven wurde hierfür um Transport- und Rüstzeiten erweitert. Die Mehrzieloptimierung erfolgt mit Hilfe der gewichteten Summe und wird für zwei Optimierungsziele jeweils im zweidimensionalen Lösungsraum durchgeführt. Die mathematische Formulierung des Optimierungsproblems finden sich im Anhang 7. Die Lösung des Optimierungsproblems wird mit Hilfe der Python-Implementierung der Optimierungssoftware Gurobi Optimizer Version 9.0 errechnet und basiert auf dem Branch-and-Cut-Verfahren¹.

Da es sich um ein NP-vollständiges Optimierungsproblem handelt, wird der kleinste Anwendungsfall mit drei Produktionsmitteln und fünf Aufträgen verwendet. Zur Generierung

¹Gurobi Optimization (2020), 'Gurobi Optimizer Reference Manual', <https://www.gurobi.com/documentation/9.0/refman/index.html>, aufgerufen am 27.10.2020

der Pareto-Punkte des exakten Verfahrens wird zum einen die Zielkombination kumulierte Durchlaufzeit und kumulierte Verspätung und zum anderen die Zielkombination kumulierte Durchlaufzeit und Makespan optimiert. Um mehrere Pareto-Punkte zu erhalten, werden die Zielgewichtungen variiert. Dabei werden die Gewichte der einzelnen Optimierungsziele zwischen den Zielgrößen in Schritten von 0,25 verschoben, sodass die Pareto-Front gleichmäßig aufgebaut wird. Es ergeben sich damit folgende Gewichtungen zwischen den Zielen $[1; 0]$, $[0,75; 0,25]$, $[0,5; 0,5]$, $[0,25; 0,75]$, $[0; 1]$. Jeder Optimierungslauf dauert 60 Minuten. Innerhalb der 60 Minuten konnte der Optimalitätsbeweis nicht erbracht werden, sodass keine Garantie besteht, dass es sich bereits um optimale Lösungen handelt. Da keine Verbesserung der Lösungen über einen längeren Zeitraum mehr beobachtet werden konnte, kann davon ausgegangen werden, dass die Optimalitätslücke gering ist.

Tabelle 5.15: Untersuchung der absoluten Lösungsgüte anhand des 5x3 Problems

Problem- instanz	Terminierungs- kriterium	Ziel- funktion	Aktionsgenerierung	C-Wert
5x3	10 Sek. / 120 Sek.	Durchlaufzeit, Verspätung / Durchlaufzeit, Makespan	Intelligente Einschränkung der zulässigen Lösungen	Dynamisch

Zum Erzeugen der Vergleichslösungen wird der MO-MCTS mit dynamischem C -Wert, siehe Abschnitt 4.3.2.2, und mit intelligenter Einschränkung zulässiger Aktionen, siehe Abschnitt 4.3.3.4, konfiguriert. Der MO-MCTS wurde 10 Sekunden und 120 Sekunden durchgeführt, siehe Tabelle 5.16.

Abbildung 5.30 zeigt die Ergebnisse des exakten Verfahrens und des MO-MCTS für die Untersuchung des zweidimensionalen Lösungsraum aus kumulierter Durchlaufzeit und kumulierter Verspätung sowie aus kumulierter Durchlaufzeit und Makespan. Grafik (a) und Grafik (b) stellen das Ergebnis des MO-MCTS nach 10 Sekunden Laufzeit dar. Ein Vergleich der Punkte $DLZ_{MO-MCTS}^*$ und DLZ_{opt}^* belegt, dass der MO-MCTS die gleiche Lösungsgüte (unikriteriell) hinsichtlich der kumulierten Durchlaufzeit erreicht, wie das exakte Verfahren. Dies gilt ebenfalls für die Zielkriterien Verspätung, $V_{MO-MCTS}^*$ und V_{opt}^* sowie Makespan $Mks_{MO-MCTS}^*$ und Mks_{opt}^* . Die Pareto-Front des MO-MCTS wird jedoch in einigen Bereichen durch die exakten Lösungen dominiert. In Grafik (b) und (c) gibt es jeweils einen Bereich, in dem der MO-MCTS zu besseren Lösungen führt als das exakte Verfahren. Dies liegt an der gewählten Zielgewichtung des exakten Verfahrens.

Grafik (c) und (d) zeigen die Ergebnisse für 120 Sekunden Ausführungszeit. Die Lösungen des MO-MCTS liegen in diesem Fall nahe der Pareto-Front des exakten Verfahrens und erweitern für das Zieltupel aus kumulierter Durchlaufzeit und Makespan sogar teilweise die

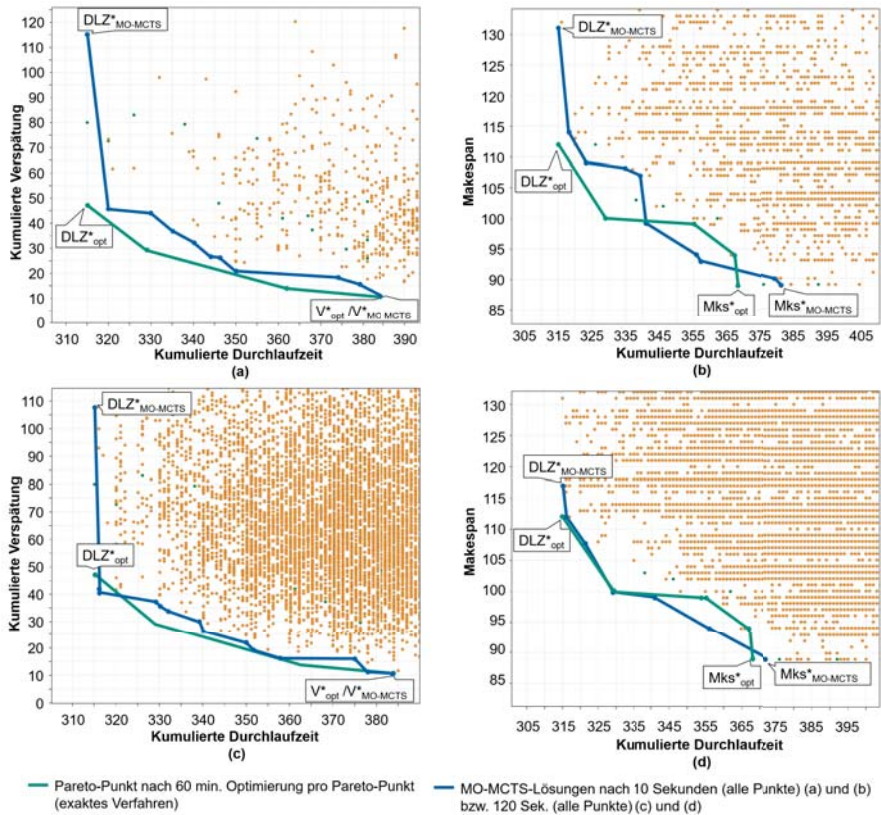


Abbildung 5.30: Pareto-Punkte des exakten Verfahrens nach je 60 Minuten sowie Lösungen des MO-MCTS nach 10 Sekunden Optimierung (a) und (b) bzw. nach 120 Sekunden (c) und (d) (5x3 Problem), angelehnt an (A_Deiningner 2020)

Pareto-Front. Dies steht nicht im Widerspruch zu den Eigenschaften eines exakten Verfahrens, da das exakte Verfahren nur für einige diskrete Zielgewichte durchgeführt und zudem der Optimalitätsbeweis nicht erbracht wurde. Die Belegungspläne für die optimale kumulierte Durchlaufzeit und den optimalen Makespan finden sich im Anhang in Abbildung A7.1.

Die Auswertungen zeigen, dass bereits nach 10 Sekunden die unikriteriellen Lösungen des exakten Verfahrens gefunden werden und die Pareto-Front angenähert wird. Der geringe Zeitbedarf des MO-MCTS-Verfahrens im Vergleich zur exakten Optimierung begründen die gute Eignung als reaktives Steuerungsverfahren für die Matrixproduktion. Bezüglich

der Anforderung an eine möglichst hohe Lösungsgüte wird deutlich, dass das vorgestellte MO-MCTS-Verfahren ebenfalls in der Lage ist, Punkte nahe des Optimums zu erreichen.

5.5.2 Vergleich anhand von Benchmark-Problemen aus der Literatur

Das Kacem-Problem kann als vereinfachtes Steuerungsproblem der Matrixproduktion ohne Rüst- und Transportzeiten abgebildet werden. Zur Evaluierung wird ein MO-MCTS mit den drei Zielgrößen Makespan, Gesamtarbeitsmenge und maximale Arbeitslast des Kacem-Problems unter Verwendung der dynamischen Anpassung des C -Werts und der intelligenten Einschränkung zulässiger Aktionen verwendet. Die Gesamtarbeitsmenge errechnet sich als Summe aller Bearbeitungsvorgänge. Da die Dauer der Bearbeitungsvorgänge von der Wahl des Produktionsmittels abhängt, kann dieser Wert minimiert werden. Die maximale Arbeitslast ist die größte Summe der Bearbeitungsvorgänge eines Produktionsmittels.

Tabelle 5.16: Konfiguration der Experimente zur Lösung des Kacem-Problems

Problem- instanz	Terminierungs- kriterium	Ziel- funktion	Aktionsgenerierung	C-Wert
Kacem: (4x5), (10x7), (8x8), (10x10), (15x10)	120 Sek.	Gesamtarbeitsmenge, Maximale Arbeitslast, Makespan	Intelligente Einschränkung der zulässigen Lösungen	Dynamischer C-Wert

Als zweiter Lösungsansatz wird das hybride Verfahren aus MO-MCTS und lokaler Suche mit der modifizierten Nachbarschaftsstruktur LSONE und BPA angewendet.

Tabelle 5.17 zeigt die Ergebnisse der fünf Problemgrößen des Kacem-Benchmarks. Der eigene Ansatz ist in der Lage, 15 der 18 aus der Literatur bekannten Pareto-Punkte zu finden. Auf der größten Probleminstanz wurden mit MO-MCTS die Punkte (12, 91, 11) und (13, 93, 10) erreicht. Das entspricht einer Abweichung beim Makespan von einer Zeiteinheit bzw. zwei Zeiteinheiten. Übereinstimmend mit den anderen Auswertungen findet das hybride Verfahren die Pareto-Punkte der vier kleineren Probleminstanzen früher als der reine MO-MCTS. Die Pareto-Punkte der 15x10 Probleminstanz können nicht identifiziert werden, jedoch wird ausgehend vom Punkt (12, 92, 11) des MO-MCTS der Punkt (11, 93, 11) durch die lokale Suche identifiziert. Dies stellt eine Verbesserung gegenüber der Lösung des reinen MO-MCTS-Verfahrens dar.

Insgesamt lässt sich festhalten, dass der eigene Ansatz in der Lage ist, die meisten der bekannten Pareto-Punkte zu finden. Die als Vergleich herangezogenen Verfahren sind stark

Tabelle 5.17: Ergebnisse des Kacem-Vergleichs, nach (A_Deiningger 2020), (A_Liu 2021)
 nicht erreicht: ○, erreicht: ●, keine Angabe: -, optimale Lösung: *

	(4x5)	(10x7)	(8x8)	(10x10)	(15x10)
Makespan*	11 11 12 13	11 11 12	14 15 15 16 16	7 7 8 8	11 11
Gesamtarbeitsmenge*	32 34 32 33	61 62 60	77 75 81 73 77	42 43 42 41	91 93
Maximale Arbeitslast*	10 9 8 7	11 10 12	12 12 11 13 11	6 5 5 7	11 10
Multi-objective Genetic algorithm 1	● ● ● ○	- - -	○ ● ● ● ○	● ○ ● ●	● ○
Simple evolutionary algorithm (SEA) 2	● ● ● ●	● ● ●	● ● ○ ● ●	● ● ● ●	● ●
Mememic algorithm (GA + LS) 3	● ● ● ●	● ● ●	● ● ○ ● ●	● ● ● ●	● ●
MO-MCTS + Local Search (Wu) 4	● ● ● ●	● ● ●	● ● ○ ● ●	● ● ● ●	● ●
MO-MCTS + Local Search (Chou) 5	● ● ● ●	● ● ●	● ○ ● ●	● ● ● ●	● ●
MO-MCTS (Eigener Ansatz)	● ● ● ●	● ● ●	● ● ○ ● ●	● ● ● ●	○ ○
MO-MCTS + LS (Eigener Ansatz)	● ● ● ●	● ● ●	● ● ○ ● ●	● ● ● ●	○ ○

auf das Kacem-Problem abgestimmt, siehe Abschnitt 3.3. Aufgrund des Einflusses der angewendeten Einschränkungen auf den Suchfokus, der im Laufe dieser Arbeit gezeigt werden konnte, ist anzunehmen, dass diese Verfahren keine vergleichbare Lösungsgüte auf anderen Problemen erreichen können. Der eigene Ansatz hingegen ist durch seine automatischen Anpassungsmechanismen für die Wahl des *C*-Werts und die Einschränkung zulässiger Aktionen in der Lage, sich automatisch an die Problemcharakteristik anzupassen.

5.5.3 Erprobung am Anwendungsfall der Getriebemontage

Um das Verhalten der Steuerung über mehrere Steuerungsentscheidungen im Kontext von Störungen der Produktionsmittel und stochastischen Bearbeitungszeiten zu zeigen, wird die hybride Steuerung auf den Anwendungsfall Getriebemontage angewendet.

¹(Wang & Gao u. a. 2010)

²(Chiang & Lin 2013)

³(Chiang & Lin 2012)

⁴(Wu & Wu u. a. 2013)

⁵(Chou & Liang u. a. 2015)

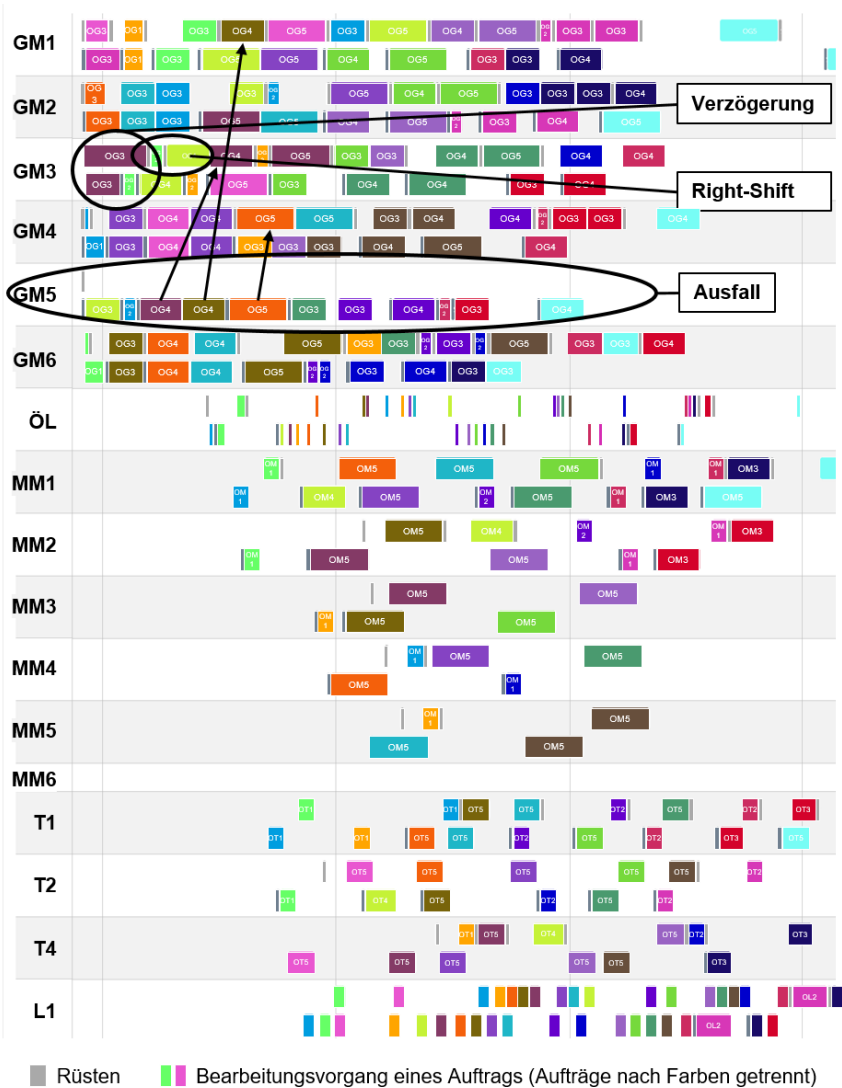


Abbildung 5.31: Belegungsplan der Getriebemontage. Untere Zeile: initialer Belegungsplan, obere Zeile: letzter Belegungsplan, angelehnt an (A_Liu 2021)

Die hybride Steuerung nutzt die intelligente Einschränkung zulässiger Aktionen, den dynamischen C -Wert und die besten Pareto-Lösungen am Ende der Ausführung als Ausgangslösung für die lokale Suche. Die Ausführungszeit ist auf zehn Sekunden beschränkt.

Die Getriebemontage wird über ein ereignisdiskretes Simulationsmodell abgebildet, das die Steuerungsentscheidungen ausführt und den aktuellen Zustand an die Steuerung zurückmeldet. Die Steuerung überprüft bei jeder Rückmeldung, ob eine Neuplanung erforderlich ist. Dies ist der Fall, wenn es zu einer Abweichung des Plans gekommen ist oder wenn Produktionsaufträge oder Produktionsmittel den Status verändern, beispielsweise ausfallen, neu dazukommen oder fertiggestellt werden. Es handelt sich folglich um eine ereignisbezogene Neuplanung. Aufgrund der Vorbereitungen an den Produktionsstationen für die einzelnen Aufträge sind innerhalb von 20 Minuten keine Reallokationen von Produktionsaufträgen erlaubt. Innerhalb dieses zeitlichen Korridors sind lediglich Verschiebungen der bereits eingeplanten Produktionsaufträge gestattet.

Abbildung 5.31 zeigt einen Belegungsplan aus Sicht der Produktionsmittel. In diesem Plan sind daher die Bearbeitungs- und Rüstvorgänge enthalten. In der unteren Zeile ist der initiale Belegungsplan abgebildet. In der oberen Zeile ist der letzte Belegungsplan zu sehen. Die dargestellten Pläne bauen nicht direkt aufeinander auf, sondern stellen zwei Planungsstände zu Beginn und gegen Ende dar. Das Produktionsmittel MM6 ist über den gesamten Zeitraum nicht verfügbar. Das Beispiel der Getriebemontage 5 (GM5) zeigt einen Ausfall. Es ist erkennbar, dass im Weiteren diese Station nicht mehr belegt wird. Die ursprünglich eingeplanten Vorgänge werden außerhalb des 20-Minuten-Korridors auf anderen Produktionsmittel eingeplant. Für den unmittelbar betroffenen Produktionsauftrag SO10 (hellgrün) wird auch die Reihenfolge der Vorgänge angepasst, siehe Abbildung 5.32. Der Vorgang OG3 auf dem Produktionsmittel GM3 verzögert sich. Man erkennt deutlich den Right-Shift, also die Verschiebung der nachfolgenden Vorgänge innerhalb des 20-Minuten-Korridors. In diesem Anwendungsfall konnte die lokale Suche durchschnittlich eine Reduktion des Makespan gegenüber der Ausgangslösung von 12,7 % erzielen.

Der dargestellte Anwendungsfall kann nicht herangezogen werden, um die Güte der Steuerungsentscheidungen zu bewerten, da Referenzwerte fehlen und aufgrund der Stochastik und Komplexität nicht exakt ermittelt werden können. Es wird dennoch deutlich, dass das hybride System im Stande ist, über einen längeren Zeitraum die Produktion vorausschauend und reaktiv im Sinne der Forschungsfrage **F1** zu steuern. Entsprechend der Forschungsfrage **F2** ist die Produktionssteuerung durch die Methoden zur dynamischen Anpassung des C -Werts und die intelligente Einschränkung zulässiger Aktionen in der Lage, ohne manuelle Abstimmungen diesen unbekanntem und komplexen Anwendungsfall zu steuern.

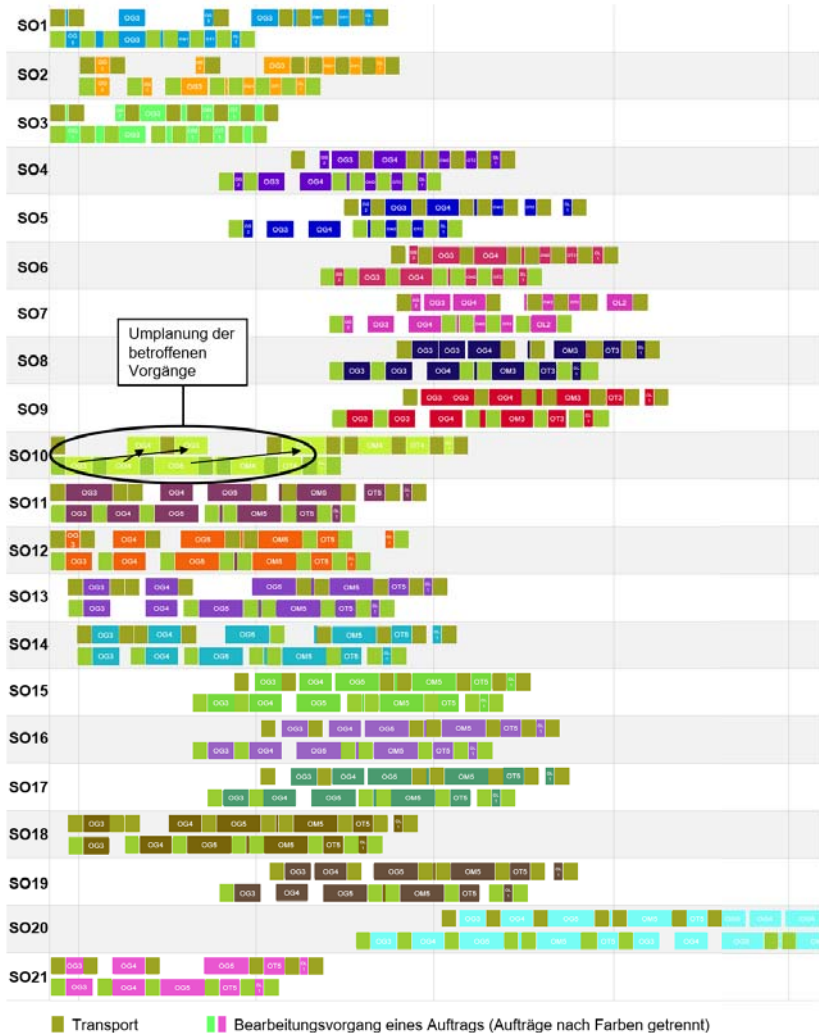


Abbildung 5.32: Belegungsplan aus Produktionsauftrags-sicht des Beispielfalls aus der Getriebemontage, angelehnt an (A_Li & Chang u. a. 2009)

6 Diskussion und Ausblick

Im folgenden Kapitel werden die Ergebnisse im Hinblick auf die formulierten Forschungsfragen eingeordnet. Darauf aufbauend werden Themen für weitere Forschungsarbeiten abgeleitet.

6.1 Diskussion

Die hohe Komplexität der Matrixproduktion stellt jede Produktionssteuerung vor große Herausforderungen. Durch die gute Modularität sind zudem häufig Änderungen des Produktionssystems möglich. Für das Steuerungsproblem bringt dies stets eine Änderung des Zustandsraums mit sich. MCTS-basierte Vorgehen haben den Vorteil, dass der Zustandsraum nicht vollständig abgebildet werden muss, sondern in Form des Suchbaums dynamisch aufgebaut und abgesucht wird. Diese Eigenschaft ermöglicht die Anwendung des Verfahrens ohne vorherige Einlernphase bei einem sich wandelnden Produktionssystem.

Die Forschungsfrage **F1** widmet sich der Fragestellung, wie eine reaktive und vorausschauende Mehrziel-Steuerung für eine Matrixproduktion unter Berücksichtigung realer Restriktionen erzielt werden kann. Um die Anforderungen einer Matrixproduktion adäquat abzubilden, werden Transport- und Rüstvorgänge sowie alternative Vorgangssequenzen modelliert. Die in dieser Arbeit konzipierte und prototypisch umgesetzte Steuerung basiert auf MO-MCTS, kombinierbar mit lokaler Suche. Diese ist in der Lage unterschiedliche Zielgrößen zu optimieren. Durch die Evaluation von Endzuständen werden auch auftragsbezogenen Zielgrößen, wie beispielsweise die Verspätung und Durchlaufzeit, bei denen Einzelentscheidungen eine wichtige Rolle spielen, erfolgreich optimiert. Die Auswertungen zeigen, dass die Aktionsgenerierung einen großen Einfluss auf den erreichbaren Bereich des Lösungsraums hat. Durch den Einsatz von problemspezifischen Heuristiken zur Einschränkung zulässiger Lösungen (Filterkombinationen) in der Aktionsgenerierung kann die Zielerreichung verbessert werden. Je nach Zielgröße führen jedoch andere Filterkombinationen in der Aktionsgenerierung zu guten Ergebnissen. Die entwickelte Methode zur intelligenten Einschränkung zulässiger Aktionen ermöglicht es, automatisiert geeignete problemspezifische Heuristiken in Form

von Filterkombinationen zur Einschränkung zulässiger Aktionen auszuwählen, sodass alle untersuchten Zielkriterien gleichmäßig und besser erreicht werden.

Bezüglich der Reaktionsfähigkeit der Steuerung bietet MO-MCTS bereits nach nur einer Iteration eine zulässige Lösung. Die Lösungsgüte steigt mit zunehmender Iterationszahl asymptotisch an. Diese Charakteristik ermöglicht eine reaktive Steuerung. Es kann festgestellt werden, dass die intelligente Einschränkung zulässiger Aktionen und die dynamische Anpassung des Explorationsfaktors den Anstieg der Lösungsgüte beschleunigen, was die Reaktionsfähigkeit der Steuerung weiter steigert. Durch die Kombination mit lokaler Suche kann dies zusätzlich verbessert werden. Ein wesentlicher Vorteil der lokalen Suche liegt zudem in der Steigerung der Lösungsgüte, insbesondere bei nicht-optimalen MO-MCTS-Parametern. Durch Maßnahmen der Parallelisierung kann die Suchgeschwindigkeit und die Reaktionsfähigkeit ebenfalls verbessert werden. Allerdings führen diese Ansätze nur zu einem linearen Anstieg der Suchgeschwindigkeit. Sie sind daher ungeeignet, um die Komplexität des Problems zu beherrschen.

Forschungsfrage **F2** befasst sich mit der Fragestellung, ob durch automatisierte Anpassung des Steuerungsverfahrens eine hohe Lösungsgüte bei unterschiedlichen Steuerungsproblemen und Zielgrößen erreicht werden kann. Die umfangreichen Auswertungen des Einflusses des Mechanismus zur Aktionsgenerierung auf den erreichbaren Bereich des Lösungsraums haben eine starke Abhängigkeit ergeben. Durch die vorgestellte Methode zur intelligenten Einschränkung zulässiger Aktionen ist es möglich, diese Abhängigkeit stark zu reduzieren. Es konnte gezeigt werden, dass die automatisch selektierten Filterkombinationen zur Einschränkung zulässiger Aktionen nicht nur die Suchgeschwindigkeit erhöhen, sondern auch zu einer breiten Abdeckung des Lösungsraums führen. Dies macht deutlich, dass durch diese Methode der nicht-relevante Bereich des Lösungsraums von der Suche ausgeschlossen wird. Der Explorationsfaktor, der die Balance zwischen der Erkundung des Lösungsraums ausgehend von bereits bekannten, guten Lösungen und der Erkundung des noch unbekanntem Lösungsraums steuert, hat ebenfalls einen starken Einfluss auf das erreichte Lösungs-niveau und den Anstieg der Lösungsgüte. Der vorgestellte Ansatz zur dynamischen Anpassung des Explorationsfaktors verbessert die Steuerung dahingehend, dass sie in der Lage ist, Phasen der verstärkten Suche aneinanderzureihen und selbstständig lokale Optima zu überwinden. Die Untersuchungen zur erreichbaren Lösungsgüte anhand eines Referenzproblems und mehrerer Benchmark-Probleme zeigen, dass die MO-MCTS-basierte Steuerung durch diese Methoden in der Lage ist, sich selbstständig an die Steuerungsprobleme und Zielgrößen anzupassen und die optimalen Steuerungsentscheidungen anzunähern oder zu erreichen.

Die Forschungsfrage **F3** adressiert die Anpassung der Dauern der Bearbeitungs-, Rüst- und Transportzeiten, die als Eingangsgröße der Produktionssteuerung genutzt werden. Realitäts-

nahe Vorgangsdauern sind eine Voraussetzung für realisierbare Steuerungsentscheidungen. Daher wurde eine Methode zur Schätzung der zugrundeliegenden Verteilungen und zur Detektion von Verteilungsänderungen basierend auf den realisierten Dauern entwickelt. Die Anwendung dieser Methode zeigt, dass passende Verteilungen geschätzt werden können und Verteilungsänderungen sowie saisonale Effekte zuverlässig detektiert werden. Die Anwendung der Steuerung auf das Praxisbeispiel der Getriebemontage verdeutlicht, dass die Steuerung mit stochastischen Vorgängen und Störungen von Produktionsmitteln umgehen kann und fortlaufend den Belegungsplan an die neuen Bedingungen anpasst. Die Steuerung ist in der Lage, unter Berücksichtigung eines Planungskorridors mit eingeschränkter Flexibilität, in dem nur Verschiebungen zulässig sind, umzuplanen.

6.2 Ausblick

Die untersuchte Steuerung deckt bereits die wichtigsten Charakteristika einer Matrixproduktion ab. Weitere Forschungsarbeiten könnten zusätzliche Restriktionen und Freiheitsgrade in die Betrachtung aufnehmen. Die Aufspaltung und Zusammenlegung von Losen und die Einbeziehung von Werkzeugen und Mitarbeitenden wären interessante Ergänzungen. Da diese Erweiterung die Komplexität erhöht, müssten im Zuge dessen die Verfahren zur Einschränkung des Lösungsraums erneut geprüft werden.

Obwohl die Matrixproduktion eine hohe Modularität bietet, nutzt der aktuelle Steuerungsansatz die Möglichkeiten der Wandlungsfähigkeit noch nicht als Freiheitsgrad aus. Zukünftige Arbeiten könnten die Anpassung des Produktionssystems durch Hinzunehmen oder Entfernen von Produktionsmitteln sowie die Änderung des Layouts als Steuerungsmöglichkeit berücksichtigen. Dies führt zu einer Aufteilung der Steuerungsentscheidung in eine operative und eine taktische Ebene. Die Berücksichtigung verminderter Kapazität in der Übergangsphase sollte ebenfalls Berücksichtigung finden.

Die Komplexität der Steuerungsaufgabe, verbunden mit dem Anspruch bei Planabweichungen steuernd einzugreifen, erfordert kurze Rechenzeiten und einen schnellen Anstieg der Lösungsgüte. Obwohl MCTS ein domänenunabhängiges Optimierungsverfahren ist, empfiehlt diese Komplexität den Einsatz von Domänenwissen zur Reduktion des Aktionsraums. Durch Domänenwissen sinkt jedoch die Leistungsfähigkeit auf domänenfremden Problemen. Der vorgestellte Ansatz zur automatischen Selektion geeigneter Filterkombinationen leistet einen wichtigen Beitrag, baut jedoch auf einer festen Menge definierter Heuristiken auf. Weitere Arbeiten im Bereich der Generierung von Heuristiken könnten die Anwendbarkeit des Verfahrens weiter steigern.

Die Anwendung der Steuerung auf das Praxisbeispiel der Getriebemontage ist ein erster Schritt zur Bewertung der Steuerung unter realen Bedingungen. Weitere Forschungsarbeiten könnten einen Vergleich zwischen der entwickelten Steuerung über einen längeren Zeitraum fokussieren. Um die ereignisdiskrete Simulation zu ersetzen, wäre die Einbindung in die reale Produktions-IT, insbesondere das MES-System, nötig.

Die Erhebung der Planzeiten aus den Rückmeldedaten der Produktion ist eine wichtige Voraussetzung einer realisierbaren Planung und Steuerung. Daher ist die Verfügbarkeit und Qualität realisierter Dauern ausschlaggebend. Diese Arbeit stellt eine Methode zur Abschätzung realistischer Planzeiten vor. Allerdings erfordert auch diese Methode das Vorliegen von realisierten Dauern aus der Produktion. Hierfür ist die Rückmeldung des Starts und des Endes der einzelnen Vorgänge erforderlich. Ungenaue Rückmeldungen verzerren das Bild und lassen keine realitätsnahe Steuerung zu. Der Digitalisierung der Arbeitsplätze kommt somit eine entscheidende Bedeutung auf dem Weg zu einer leistungsfähigen Steuerung der Matrixproduktion in der Praxis zu. Ansätze zur Digitalisierung bestehender Produktionsmittel und manueller Arbeitsplätze werden daher benötigt. Ein weiteres Feld für zukünftige Forschungsvorhaben besteht in der Adressierung der Frage, welches Quantil der geschätzten Verteilung der Dauern als Planzeit herangezogen werden sollte, um einen guten Kompromiss aus logistischer Zielerreichung und der Häufigkeit von Planänderungen zu erzielen. Diese Betrachtungen sollten insbesondere die Auswirkungen der Dauer des Korridors mit verminderter Flexibilität einbeziehen.

Die produktionsnahen Bereiche, insbesondere die Logistik, sind ein wichtiger Befähiger für die Umsetzung einer Matrixproduktion in der Praxis. Mit steigendem Vorlauf für Kommissionierung und Beschickung der Arbeitsplätze sinkt die Flexibilität der Produktionssteuerung, um auf Echtzeitereignisse zu reagieren. Auch wenn eine integrierte Betrachtung beider Domänen die Komplexität weiter erhöht, ließe sich hierdurch die Anwendbarkeit verbessern und durch den Verzicht auf statische Vorlaufzeiten, orientiert an der maximalen Dauer, die Flexibilität verbessern.

Aus Sicht des Produktionssystems als sozio-technisches System wäre eine Untersuchung der Auswirkungen einer solchen Steuerung auf die Mitarbeitenden interessant. Ein möglicher Schwerpunkt wäre die Betrachtung, wie die Mitarbeitenden auf häufige Plananpassungen reagieren und wie diese Änderungen nachvollziehbar kommuniziert werden könnten.

7 Zusammenfassung

Ein zunehmend diverseres Produktionsprogramm mit unsicheren Stückzahlen erschwert den wirtschaftlichen Betrieb von Produktionssystemen. Bewirkt die Produktindividualisierung unterschiedliche Bearbeitungszeiten an den Produktionsstationen, kommt es zu Taktzeitverlusten und insgesamt zu einer geringeren Auslastung sowie Verzögerungen. Schwankungen in den Anteilen der Produktvarianten am Produktionsprogramm können zudem zu dynamischen Engpässen im Produktionssystem führen.

Das Konzept der Matrixproduktion verfolgt eine Flexibilisierung der Produktionsstruktur durch Auflösung der starren Verkettung, der Taktzeitbindung und die Nutzung redundanter Mehrzweckstationen. Durch diese Maßnahmen kann die Produktionssteuerung die Vorgangsrufenfolge innerhalb der technischen Grenzen des Vorrangsgraphs variieren und die Route jedes Auftrags durch das Produktionssystem individuell anpassen. Damit diese Freiheitsgrade optimal genutzt werden können und den unterschiedlichen Zielsetzungen von Produktionssystemen Rechnung getragen werden kann, bedarf es einer reaktiven und vorausschauenden Mehrziel-Steuerung. Da sich die Struktur von Produktionssystemen, das Produktionsprogramm und die Zielsetzungen unterscheiden, ist die Fähigkeit der Steuerung, sich selbstständig auf das Steuerungsproblem und die Zielgrößen anzupassen, entscheidend, um für diese unterschiedlichen Steuerungsfälle eine hohe Lösungsgüte erzielen zu können. Die Dauern der Rüst-, Transport- und Bearbeitungszeiten sind wichtige Eingangsgrößen der Produktionssteuerung. Diese Dauern sind statistisch verteilt und unterliegen Verteilungsänderungen. Um Steuerungsentscheidungen hoher Güte zu treffen, müssen die zugrundeliegenden Verteilungen der Dauern fortlaufend geschätzt werden, sodass realistische Planzeiten als Eingangsgröße der Produktionssteuerung abgeleitet werden können. Aufgrund der Komplexität der Steuerungsentscheidung sind heuristische Verfahren besser geeignet als mathematische Optimierung. Insbesondere MCTS verfügt als iteratives Suchbaumverfahren über gute Eigenschaften für die Nutzung in der Produktionssteuerung. Bislang fehlten jedoch Ansätze in der Forschung, wie die Anforderungen der Steuerung einer Matrixproduktion erfüllt werden können.

Im Rahmen dieser Arbeit wurde eine reaktive Mehrziel-Steuerung auf Basis von MO-MCTS zur Produktionssteuerung einer Matrixproduktion unter Berücksichtigung von Rüst- und Trans-

portvorgängen entwickelt. Um die Anforderung der Matrixproduktion zu erfüllen, wurden Erweiterungen des MO-MCTS-Basis-Verfahrens in allen vier Phasen des Algorithmus vorgenommen. Zudem wurde eine Postoptimierung basierend auf lokaler Suche entwickelt, die in den MO-MCTS-Ablauf integriert werden kann.

Zur gleichmäßigen Erreichung unterschiedlicher Zielgrößen wurde eine Methode zur intelligenten Einschränkung zulässiger Aktionen entwickelt. Diese Methode ermittelt selbstständig geeignete Filterkombinationen, um die zulässigen Aktionen vor dem Hintergrund der angestrebten Zielgrößen bestmöglich zu reduzieren. Um die verstärkte Suche in Bereichen des Lösungsraums mit guten Lösungen und die Erkundung des noch unbekanntem Lösungsraums auszubalancieren, wurde ein Verfahren zur dynamischen Wahl des Explorationsfaktors vorgestellt, das die Erkundung adaptiv erhöht, wenn die Lösungsgüte stagniert, sodass lokale Optima effizient überbrückt werden können. Zur Verbesserung der Lösungsgüte und Beschleunigung des Anstiegs der Lösungsgüte wurde das MO-MCTS-Verfahren durch eine Postoptimierung basierend auf lokaler Suche komplettiert. Eine Methode zur Schätzung von Verteilung und Detektion von Verteilungsänderungen liefert realistische Planzeiten als Eingangsgrößen für die Steuerung.

Die Anwendung der Verteilungsschätzung und Änderungsdetektion konnte zeigen, dass aus realen Rückmeldedaten zuverlässig Verteilungen ermittelt werden können, sodass realistische Planzeiten als Eingangsgröße der Produktionssteuerung gewonnen werden können. Im Rahmen der Forschungsarbeit wurde detailliert untersucht, wie sich das Suchverhalten des hybriden Systems im Lösungsraum beeinflussen und steuern lässt. Als zweiter Schwerpunkt der Auswertungen wurde betrachtet, wie sich der Anstieg der Suchgeschwindigkeit beschleunigen lässt. Als dritter Schwerpunkt wurde die erreichte Lösungsgüte anhand unterschiedlicher Anwendungsfälle und Zielgrößen betrachtet. Hierzu wurde die Lösungsgüte mit Hilfe von Referenzlösungen bewertet. Ein Vergleich zu anderen Steuerungsverfahren konnte durch das Kacem-Benchmark-Problems gezogen werden. Um das Verhalten der Steuerung im Kontext einer realen stochastischen Produktion zu beobachten, wurde die Steuerung anhand eines Simulationsmodells einer Getriebemontage erprobt.

Die vorliegende Arbeit stellt ein umfassendes Verfahren zur Steuerung einer realitätsnahen Matrixproduktion vor und präsentiert Ansätze, wie die Steuerung automatisiert auf unterschiedliche Steuerungsfälle und Optimierungsziele angepasst werden kann.

Eigene Veröffentlichungen

- Hofmann, C.; Brakemeier, N.; Krahe, C.; Stricker, N. & Lanza, G. (2018), 'The Impact of Routing and Operation Flexibility on the Performance of Matrix Production Compared to a Production Line'. *Advances in Production Research*. Hrsg. von R. Schmitt & G. Schuh. 8th Congress of the German Academic Association for Production Technology (WGP), 18.-19. Nov. 2018, Aachen, Deutschland: Springer, Cham, S. 155–165.
- Hofmann, C.; Krahe, C.; Stricker, N. & Lanza, G. (2020), 'Autonomous production control for matrix production based on deep Q-Learning'. *Procedia CIRP*. Hrsg. von R. Teti & D. M. D'Addona. Bd. 88. 13th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 17.-19. Juli 2019, Golf von Neapel, Italien: Elsevier B.V., Amsterdam, S. 25–30.
- Hofmann, C.; Lauber, S.; Haefner, B. & Lanza, G. (2018), 'Development of an agile development method based on Kanban for distributed part-time teams and an introduction framework'. Hrsg. von D. Mourtzis & G. Chryssolouris. Bd. 23. 8th Conference on Learning Factories, 12.-13. April 2018, Patras, Griechenland: Elsevier B.V., Amsterdam, S. 45–50.
- Hofmann, C.; Patschkowski, C.; Haefner, B. & Lanza, G. (2020), 'Machine Learning Based Activity Recognition To Identify Wasteful Activities In Production'. Hrsg. von C. Ramsauer; M. Wolf & M. Hulla. Bd. 45. 10th Conference on Learning Factories, 16.-17. April 2018, Graz, Österreich: Elsevier B.V., Amsterdam, S. 171–176.
- Hofmann, C.; Schoof, J.; Kuhnle, A. & Lanza, G. (2021), 'Flexibilitätsoptimale Austaktung der Fließmontage', *wt Werkstatttechnik online* 111.4, S. 260–264.
- Hofmann, C.; Staehr, T.; Cohen, S.; Stricker, N.; Haefner, B. & Lanza, G. (2019), 'Augmented Go & See: An approach for improved bottleneck identification in production lines'. Hrsg. von C. Herrmann & S. Thiede. Bd. 31. 9th Conference on Learning Factories, 26. - 29. März 2019, Braunschweig, Deutschland: Elsevier B.V., Amsterdam, S. 148–154.
- Lanza, G.; Hofmann, C.; Biehl, E. & Braun, Y. (2018), *Auf dem Weg zum Digitalen Shopfloor Management. Eine Studie zum Stand der Echtzeitentscheidungsfähigkeit und des Industrie 4.0-Reifegrads*. Techn. Ber. Karlsruhe: Karlsruhe Institut für Technologie.
- Stricker, N.; Kuhnle, A.; Hofmann, C. & Deininger, P. (2021), 'Self-adjusting multi-objective scheduling based on Monte Carlo Tree Search for matrix production assembly systems', *CIRP Annals* 70.1, S. 381–384.

Yang, S.; Liu, H.; Zhang, Y.; Arndt, T.; Hofmann, C.; Häfner, B. & Lanza, G. (2020), 'A data-driven approach for quality analytics of screwing processes in a global learning factory'. Hrsg. von C. Ramsauer; M. Wolf & M. Hulla. Bd. 45. 10th Conference on Learning Factories, 16.-17. April 2018, Graz, Österreich: Elsevier B.V., Amsterdam, S. 454–459.

Literaturverzeichnis

Acker 2011

Acker, I. J. (2011), *Methoden zur mehrstufigen Ablaufplanung in der Halbleiterindustrie*. Universität Hohenheim, Dissertation. Wiesbaden: Gabler Verlag.

Armbruster & De Beer u. a. 2006

Armbruster, D.; De Beer, C.; Freitag, M.; Jagalski, T. & Ringhofer, C. (2006), 'Autonomous control of production networks using a pheromone approach', *Physica A: Statistical Mechanics and its Applications* 363.1, S. 104–114.

Arnold & Isermann u. a. 2008

Arnold, D.; Isermann, H.; Kuhn, A.; Tempelmeier, H. & Furmans, K., Hrsg. (2008), *Handbuch Logistik*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Aydin & Öztemel 2000

Aydin, M. E. & Öztemel, E. (2000), 'Dynamic job-shop scheduling using reinforcement learning agents', *Robotics and Autonomous Systems* 33.2-3, S. 169–178.

Balas 1969

Balas, E. (1969), 'Machine Sequencing Via Disjunctive Graphs: An Implicit Enumeration Algorithm', *Operations Research* 17.6, S. 941–957.

Balas & Vazacopoulos 1998

Balas, E. & Vazacopoulos, A. (1998), 'Guided Local Search with Shifting Bottleneck for Job Shop Scheduling', *Management Science* 44.2, S. 262–275.

Bauernhansl & Ten Hompel u. a. 2014

Bauernhansl, T.; Ten Hompel, M. & Vogel-Heuser, B., Hrsg. (2014), *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung-Technologien-Migration*. Wiesbaden: Springer Vieweg.

Bifet & Gavaldà 2006

Bifet, A. & Gavaldà, R. (2006), 'Kalman Filters and Adaptive Windows for Learning in Data Streams'. *Discovery Science*. Hrsg. von T. Ljupčič; L. Nada & K. P. Jantke. Bd. 4265. DS: International Conference on Discovery Science, 7.-10. Okt. 2006, Barcelona, Spanien. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 29–40.

Bifet & Gavaldà 2007

Bifet, A. & Gavaldà, R. (2007), 'Learning from time-changing data with adaptive windowing'. *Proceedings of the 2007 SIAM international conference on data mining*. Hrsg. von C. Apte;

- D. Skillicorn; B. Liu & S. Parthasarathy. 7th SIAM International Conference on Data Mining, 26.-28. April 2007, Minneapolis, USA. Philadelphia: Society of Industrial und Applied Mathematics, S. 443–448.
- Blackstone & Phillips u. a. 1982
Blackstone, J. H.; Phillips, D. T. & Hogg, G. L. (1982), 'A state-of-the-art survey of dispatching rules for manufacturing job shop operations', *International Journal of Production Research* 20.1, S. 27–45.
- Blazewicz & Domschke u. a. 1996
Blazewicz, J.; Domschke, W. & Pesch, E. (1996), 'The job shop scheduling problem: Conventional and new solution techniques', *European Journal of Operational Research* 93.1, S. 1–33.
- Bochmann 2018
Bochmann, L. S. (2018), *Entwicklung und Bewertung eines flexiblen und dezentral gesteuerten Fertigungssystems für variantenreiche Produkte*. ETH Zürich, Dissertation. Zürich: ETH Zürich Verlag.
- Bonfill & Espuña u. a. 2005
Bonfill, A.; Espuña, A. & Puigjaner, L. (2005), 'Addressing Robustness in Scheduling Batch Processes with Uncertain Operation Times', *Industrial & Engineering Chemistry Research* 44.5, S. 1524–1534.
- Browne & Powley u. a. 2012
Browne, C.; Powley, E.; Whitehouse, D.; Lucas, S.; Cowling, P.; Rohlfshagen, P.; Tavener, S.; Liebana, D.; Samothrakis, S. & Colton, S. (2012), 'A Survey of Monte Carlo Tree Search Methods.', *IEEE Trans. Comput. Intellig. and AI in Games* 4.1, S. 1–49.
- Chaslot & De Jong u. a. 2006
Chaslot, G; De Jong, S; Saito, J. T. & Uiterwijk, J (2006), 'Monte-Carlo tree search in production management problems'. *Proceedings of the 18th BeNeLux Conference on Artificial Intelligence*. Hrsg. von P.-Y. Schobbens; W. Vanhoof & G. Schwanen. 18th BeNeLux Conference on Artificial Intelligence BNAIC'06, 5.-6. Oktober 2006, Namur, Belgien. Namur: University of Namur, S. 91–98.
- Chiang & Lin 2012
Chiang, T.-C. & Lin, H.-J. (2012), 'Flexible Job Shop Scheduling Using a Multiobjective Memetic Algorithm'. *Advanced Intelligent Computing*. Hrsg. von D. S. Huang; Y. Gan; V. Bevilacqua & J. C. Figueroa. 7th International Conference, ICIC 2011, 11.-14. August 2011, Zhengzhou, China. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 49–56.
- Chiang & Lin 2013
Chiang, T.-C. & Lin, H.-J. (2013), 'A simple and effective evolutionary algorithm for multiob-

- jective flexible job shop scheduling', *International Journal of Production Economics* 141.1, S. 87–98.
- Chou & Liang u. a. 2015
- Chou, J. J.; Liang, C. C.; Wu, H. C.; Wu, I. C. & Wu, T. Y. (2015), 'A new MCTS-based algorithm for multi-objective flexible job shop scheduling problem'. *Proceedings of a meeting held 20-22 November 2015, Tainan, Taiwan*. Hrsg. von H.-C. Wang; R.-M. Chen & B.-R. Chang. 2015 Conference on Technologies und Applications of Artificial Intelligence (TAAI 2015), 20.-22. November 2015, Tainan, Taiwan. New York: IEEE, S. 136–141.
- Church & Uzsoy 1992
- Church, L. K. & Uzsoy, R. (1992), 'Analysis of periodic and event-driven rescheduling policies in dynamic shops', *International Journal of Computer Integrated Manufacturing* 5.3, S. 153–163.
- Coulom 2006
- Coulom, R. (2006), 'Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search'. *CG'06: Proceedings of the 5th international conference on Computers and games*. Hrsg. von H. J. van den Herik & P. Ciancarini. CG'06. 5th International Conference on Computer and Games, 29.-31. Mai 2006, Turin, Italien. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 72–83.
- Cowling & Johansson 2002
- Cowling, P. & Johansson, M. (2002), 'Using real time information for effective dynamic scheduling', *European Journal of Operational Research* 139.2, S. 230–244.
- Cramer & Kamps u. a. 2008
- Cramer, E.; Kamps, U. & Steland, A. (2008), *Grundlagen der Wahrscheinlichkeitsrechnung und Statistik 4., korrigierte und erweiterte Auflage*. Berlin, Heidelberg: Springer-Lehrbuch.
- Dell'Amico & Trubian 1993
- Dell'Amico, M. & Trubian, M. (1993), 'Applying Tabu Search to the Job-Shop Scheduling Problem', *Annals of Operations Research* 41.3, S. 231–252.
- Domschke & Drexl u. a. 2015
- Domschke, W.; Drexl, A.; Klein, R. & Scholl, A. (2015), *Einführung in Operations Research*. Berlin Heidelberg: Springer Gabler.
- Domschke & Scholl u. a. 1997
- Domschke, W.; Scholl, A. & Voß, S. (1997), *Produktionsplanung*. Bd. 2. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg.
- Drake & Uurtamo 2007
- Drake, P. & Uurtamo, S. (2007), 'Move ordering vs heavy playouts: Where should heuristics be applied in Monte Carlo Go'. *Proceedings of the 3rd North American Game-On Confe-*

rence. Hrsg. von P. Fishwick & B. Lok. GAMEON-NA'2007, 10.-12. Sept. 2007, Gainesville, USA. Ostend: Eurosis-ETI Publication, S. 171–175.

Echsler 2020

Echsler, F. (2020), *Prädiktiv-reaktives Scheduling zur Steigerung der Robustheit in der Matrix-Produktion*. Karlsruher Institut für Technologie, Dissertation. Aachen: Shaker Verlag.

Echsler Minguillon & Lanza 2017

Echsler Minguillon, F. & Lanza, G. (2017), 'Maschinelles Lernen in der PPS', *wt Werkstatt-technik online* 107.9, S. 630–634.

EIMaraghy & Schuh u. a. 2013

EIMaraghy, H; Schuh, G; EIMaraghy, W; Piller, F; Schönsleben, P; Tseng, M & Bernard, A (2013), 'Product variety management', *CIRP Annals* 62.2, S. 629–652.

Eversheim 1989

Eversheim, W. (1989), *Organisation in der Produktionstechnik Band 4: Fertigung und Montage*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Eversheim 1992

Eversheim, W. (1992), 'Flexible Produktionssysteme'. *Handwörterbuch der Organisation*. Hrsg. von E. Frese. 3. Aufl. Poeschel Verlag Stuttgart, S. 2058–2066.

Fahrmeir & Heumann u. a. 2016

Fahrmeir, L.; Heumann, C.; Künstler, R.; Pigeot, I. & Tutz, G. (2016), *Statistik: Der Weg zur Datenanalyse*. Berlin, Heidelberg: Springer Spektrum.

Freitag & Hildebrandt 2016

Freitag, M. & Hildebrandt, T. (2016), 'Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization', *CIRP Annals* 65.1, S. 433–436.

Furuoka & Matsumoto 2017

Furuoka, R. & Matsumoto, S. (2017), 'Worker's knowledge evaluation with single-player Monte Carlo tree search for a practical reentrant scheduling problem', *Artificial Life and Robotics* 22.1, S. 130–138.

Gama & Žliobaitundefined u. a. 2014

Gama, J.; Žliobaitundefined, I.; Bifet, A.; Pechenizkiy, M. & Bouchachia, A. (2014), 'A Survey on Concept Drift Adaptation', *ACM Comput. Surv.* 46.4, S. 1–37.

Gao & Sun u. a. 2008

Gao, J.; Sun, L. & Gen, M. (2008), 'A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems', *Computers & Operations Research* 35.9, S. 2892–2907.

Garey & Johnson u. a. 1976

Garey, M. R.; Johnson, D. S. & Sethi, R. (1976), 'The complexity of flowshop and jobshop scheduling', *Mathematics of operations research* 1.2, S. 117–129.

Gelly & Silver 2007

Gelly, S. & Silver, D. (2007), 'Combining Online and Offline Knowledge in UCT'. *ICML '07: Proceedings of the 24th international conference on Machine learning*. Hrsg. von Z. Ghahramani. ICML '07 & ILP '07: The 24th Annual International Conference on Machine Learning held in conjunction with the 2007 International Conference on Inductive Logic Programming, 20.-24. Juni 2007, Corvallis, USA. New York: Association for Computing Machinery, S. 273–280.

Gelly & Silver 2011

Gelly, S. & Silver, D. (2011), 'Monte-Carlo tree search and rapid action value estimation in computer Go', *Artificial Intelligence* 175.11, S. 1856–1875.

Geyik & Cedimoglu 2004

Geyik, F. & Cedimoglu, I. H. (2004), 'The strategies and parameters of tabu search for job-shop scheduling', *Journal of Intelligent Manufacturing* 15.4, S. 439–448.

Glover 1989

Glover, F. (1989), 'Tabu Search—Part I', *ORSA Journal on Computing* 1.3, S. 190–206.

Glover 1990

Glover, F. (1990), 'Tabu Search: A Tutorial', *INFORMS Journal on Applied Analytics* 20.4, S. 74–94.

Graf 1960

Graf, O. (1960), 'Physiologische und psychologische Grundlagen menschlicher Arbeit und Leistung BT - Arbeitsphysiologie'. Hrsg. von O. Graf. Wiesbaden: Gabler Verlag, S. 9–17.

Graham & Lawler u. a. 1979

Graham, R.; Lawler, E.; Lenstra, J. & Kan, A. (1979), 'Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey'. *Discrete Optimization II*. Hrsg. von P. Hammer; E. Johnson & B. Korte. Bd. 5. Advanced Research Institute on Discrete Optimization, Systems Applications of the Systems Science Panel of NATO und of the Discrete Optimization Symposium, 1.-31. Aug. 1977, Vancouver, Kanada. Amsterdam: North-Holland Publishing Company, S. 287–326.

Greschke & Schönemann u. a. 2014

Greschke, P.; Schönemann, M.; Thiede, S. & Herrmann, C. (2014), 'Matrix Structures for High Volumes and Flexibility in Production Systems'. *Procedia CIRP*. Hrsg. von H. ElMaraghy. Bd. 17. 47th CIRP Conference on Manufacturing Systems (CMS 2014), 28.-30. April 2014, Windsor, Kanada. Amsterdam: Elsevier B.V., S. 160–165.

Greschke 2016

Greschke, P. I. (2016), *Matrix-Produktion als Konzept einer taktunabhängigen Fließfertigung*. Universität Braunschweig, Dissertation. Essen: Vulkan Verlag.

Grinninger 2012

Grinninger, J. (2012), *Schlanke Produktionssteuerung zur Stabilisierung von Auftragsfolgen in der Automobilproduktion*. Technische Universität München, Dissertation. München: Technische Universität München.

Hansen & Mladenović u. a. 2010

Hansen, P.; Mladenović, N. & Moreno Pérez, J. A. (2010), 'Variable neighbourhood search: methods and applications', *Annals of Operations Research* 175.1, S. 367–407.

Hedderich & Sachs 2016

Hedderich, J. & Sachs, L. (2016), *Angewandte Statistik*. Berlin, Heidelberg: Springer Spektrum.

Ho & Tay 2008

Ho, N. B. & Tay, J. C. (2008), 'Solving Multiple-Objective Flexible Job Shop Problems by Evolution and Local Search', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.5, S. 674–685.

Hu & Zhu u. a. 2008

Hu, S. J.; Zhu, X.; Wang, H & Koren, Y (2008), 'Product variety and manufacturing complexity in assembly systems and supply chains', *CIRP Annals* 57.1, S. 45–48.

Huang & Yang u. a. 2013

Huang, R.-H.; Yang, C.-L. & Cheng, W.-C. (2013), 'Flexible job shop scheduling with due window—a two-pheromone ant colony approach', *International Journal of Production Economics* 141.2, S. 685–697.

Hüttemann 2021

Hüttemann, G. (2021), *Model-based A Priori Analysis of Line-less Mobile Assembly Systems*. RWTH Aachen University, Dissertation. Aachen: Apprimus Verlag.

Jain & Ranganaswamy u. a. 2000

Jain, A. S.; Ranganaswamy, B. & Meeran, S. (2000), 'New and "Stronger" Job-Shop Neighbourhoods: A Focus on the Method of Nowicki and Smutnicki (1996)', *Journal of Heuristics* 6.4, S. 457–480.

Jamili 2016

Jamili, A. (2016), 'Robust job shop scheduling problem: Mathematical models, exact and heuristic algorithms', *Expert Systems with Applications* 55, S. 341–350.

Jiménez 2012

Jiménez, Y. M. (2012), *A generic multi-agent reinforcement learning approach for scheduling problems*. Universität Brüssel, Dissertation. Brüssel: VUBPRESS Brussels University Press.

Kacem & Hammadi u. a. 2002

Kacem, I.; Hammadi, S. & Borne, P. (Sep. 2002), 'Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic', *Mathematics and Computers in Simulation* 60.3-5, S. 245–276.

Kadwe & Suryawanshi 2015

Kadwe, Y. & Suryawanshi, V. (2015), 'A review on concept drift', *IOSR J. Comput. Eng* 17.1, S. 20–26.

Kar & Ramalingam 2013

Kar, S. S. & Ramalingam, A. (2013), 'Is 30 the magic number? Issues in sample size estimation', *National Journal of Community Medicine* 4.1, S. 175–179.

Keckl & Kern u. a. 2016

Keckl, S.; Kern, W.; Abou-Haydar, A. & Westkämper, E. (2016), 'An Analytical Framework for Handling Production Time Variety at Workstations of Mixed-model Assembly Lines'. *Procedia CIRP*. Hrsg. von R. Teti. Bd. 41. 48th CIRP Conference on Manufacturing Systems, 24.-26. Juni 2015, Ischia, Italien. Amsterdam: Elsevier B.V., S. 201–206.

Kern & Lämmermann u. a. 2017

Kern, W.; Lämmermann, H. & Bauernhansl, T. (2017), 'An Integrated Logistics Concept for a Modular Assembly System'. *Procedia Manufacturing*. Hrsg. von M. Pellicciari & M. Peruzzini. Bd. 11. 7th International Conference on Flexible Automation und Intelligent Manufacturing, FAIM2017, 27.-30. Juni 2017, Modena, Italien. Amsterdam: Elsevier B.V., S. 957–964.

Kirkpatrick & Gelatt u. a. 1983

Kirkpatrick, S; Gelatt, C. D. & Vecchi, M. P. (1983), 'Optimization by Simulated Annealing', *Science* 220.4598, S. 671–680.

Kocsis & Szepesvári 2006

Kocsis, L. & Szepesvári, C. (2006), 'Bandit Based Monte-Carlo Planning'. *Machine Learning: ECML 2006*. Hrsg. von J. Fürnkranz; T. Scheffer & M. Spiliopoulou. 17th European Conference on Machine Learning, 18.-22. Sept. 2006, Berlin, Deutschland. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 282–293.

Koren & Gu u. a. 2018

Koren, Y.; Gu, X. & Guo, W. (2018), 'Reconfigurable manufacturing systems: Principles, design, and future trends', *Frontiers of Mechanical Engineering* 13.2, S. 121–136.

Koren & Shpitalni 2010

Koren, Y. & Shpitalni, M. (2010), 'Design of reconfigurable manufacturing systems', *Journal of Manufacturing Systems* 29.4, S. 130–141.

Kuhnle 2020

Kuhnle, A. (2020), *Adaptive Order Dispatching based on Reinforcement Learning Ap-*

- plication in a Complex Job Shop in the Semiconductor Industry*. Karlsruher Institut für Technologie, Dissertation. Aachen: Shaker Verlag.
- Laarhoven & Aarts u. a. 1992
Laarhoven, P. J. M. v.; Aarts, E. H. L. & Lenstra, J. K. (1992), 'Job Shop Scheduling by Simulated Annealing', *Operations Research* 40.1, S. 113–125.
- Leitão & Restivo 2006
Leitão, P. & Restivo, F. (2006), 'ADACOR: A holonic architecture for agile and adaptive manufacturing control', *Computers in Industry* 57.2, S. 121–130.
- Lemaréchal 2001
Lemaréchal, C. (2001), 'Lagrangian relaxation'. *Computational Combinatorial Optimization. Lecture Notes in Computer Science, vol 2241*. Hrsg. von M. Jünger & D. Naddef. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, S. 112–156.
- Li & Chang u. a. 2009
Li, L.; Chang, Q. & Ni, J. (Sep. 2009), 'Data driven bottleneck detection of manufacturing systems', *International Journal of Production Research* 47.18, S. 5019–5036.
- Li & Gao 2016
Li, X. & Gao, L. (2016), 'An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem', *International Journal of Production Economics* 174, S. 93–110.
- Li & Ierapetritou 2008
Li, Z. & Ierapetritou, M. G. (2008), 'Reactive scheduling using parametric programming', *AIChE Journal* 54.10, S. 2610–2623.
- Loth & Sebag u. a. 2013
Loth, M.; Sebag, M.; Hamadi, Y.; Schoenauer, M. & Schulte, C. (2013), 'Hybridizing Constraint Programming and Monte-Carlo Tree Search: Application to the Job Shop Problem BT - Learning and Intelligent Optimization'. *Learning and Intelligent Optimization*. Hrsg. von G. Nicosia & P. Pardalos. 7th International Conference LION 7, 7.-11. Jan. 2013, Catania, Italien. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 315–320.
- Lu & Chiu u. a. 2016
Lu, C.-L.; Chiu, S.-Y.; Wu, J. & Chao, L.-P. (2016), 'Dynamic Monte-Carlo Tree Search Algorithm for Multi-Objective Flexible Job-shop Scheduling Problem', *Appl. Math* 10.4, S. 1531–1539.
- Lubosch & Kunath u. a. 2018
Lubosch, M.; Kunath, M. & Winkler, H. (2018), 'Industrial scheduling with Monte Carlo tree search and machine learning'. *Procedia CIRP*. Hrsg. von L. Wang. 51st CIRP Conference on Manufacturing Systems (CMS), 16.-18. May 2018, Stockholm, Schweden. Amsterdam: Elsevier B.V., S. 1283–1287.

Luczak & Eversheim u. a. 1999

Luczak, H.; Eversheim, W. & Schotten, M (1999), *Produktionsplanung und-steuerung: Grundlagen, Gestaltung und Konzepte*. Berlin Heidelberg: Springer Verlag.

Maravelakis & Panaretos u. a. 2005

Maravelakis, P. E.; Panaretos, J. & Psarakis, S. (2005), 'An Examination of the Robustness to Non Normality of the EWMA Control Charts for the Dispersion', *Communications in Statistics - Simulation and Computation* 34.4, S. 1069–1079.

Maravelias & Grossmann 2003

Maravelias, C & Grossmann, I (2003), 'A General Continuous State Task Network Formulation for Short Term Scheduling of Multipurpose Batch Plants with Due Dates.', *Computer Aided Chemical Engineering* 15, S. 274–279.

Marchand & Martin u. a. 2002

Marchand, H.; Martin, A.; Weismantel, R. & Wolsey, L. (2002), 'Cutting planes in integer and mixed integer programming', *Discrete Applied Mathematics* 123.1-3, S. 397–446.

Matsuo & Suh u. a. 1989

Matsuo, H.; Suh, C. J. & Sullivan, R. S. (1989), 'A controlled search simulated annealing method for the single machine weighted tardiness problem', *Annals of Operations Research* 21.1, S. 85–108.

Miller 1956

Miller, L. H. (März 1956), 'Table of Percentage Points of Kolmogorov Statistics', *Journal of the American Statistical Association* 51.273, S. 111–121.

Mladenović & Hansen 1997

Mladenović, N & Hansen, P (1997), 'Variable neighborhood search', *Computers & Operations Research* 24.11, S. 1097–1100.

Mouelhi-Chibani & Pierreval 2010

Mouelhi-Chibani, W. & Pierreval, H. (2010), 'Training a neural network to select dispatching rules in real time', *Computers & Industrial Engineering* 58.2, S. 249–256.

Müller-Merbach 1981

Müller-Merbach, H. (1981), 'Heuristics and their design: a survey', *European Journal of Operational Research* 8.1, S. 1–23.

Nowicki & Smutnicki 1996

Nowicki, E. & Smutnicki, C. (1996), 'A Fast Taboo Search Algorithm for the Job Shop Problem', *Management Science* 42.6, S. 797–813.

Nowicki & Smutnicki 2005

Nowicki, E. & Smutnicki, C. (2005), 'An Advanced Tabu Search Algorithm for the Job Shop Problem', *Journal of Scheduling* 8.2, S. 145–159.

Ouelhadj & Petrovic 2008

Ouelhadj, D. & Petrovic, S. (2008), 'A survey of dynamic scheduling in manufacturing systems', *Journal of Scheduling* 12.4, S. 417–431.

Özgüven & Özbakır u. a. 2010

Özgüven, C.; Özbakır, L. & Yavuz, Y. (2010), 'Mathematical models for job-shop scheduling problems with routing and process plan flexibility', *Applied Mathematical Modelling* 34.6, S. 1539–1548.

Petkov & Maranas 1997

Petkov, S. B. & Maranas, C. D. (1997), 'Multiperiod Planning and Scheduling of Multiproduct Batch Plants under Demand Uncertainty', *Industrial & Engineering Chemistry Research* 36.11, S. 4864–4881.

Pinedo 2016

Pinedo, M. L. (2016), *Scheduling: Theory, Algorithms, and Systems*. New York, Heidelberg, London: Springer International Publishing.

Policella & Cesta u. a. 2008

Policella, N.; Cesta, A.; Oddi, A. & Smith, S. F. (2008), 'Solve-and-robustify', *Journal of Scheduling* 12.3, S. 299–314.

Ross & Adams u. a. 2012

Ross, G. J.; Adams, N. M.; Tasoulis, D. K. & Hand, D. J. (2012), 'Exponentially weighted moving average charts for detecting concept drift', *Pattern Recognition Letters* 33.2, S. 191–198.

Runarsson & Schoenauer u. a. 2012

Runarsson, T. P.; Schoenauer, M. & Sebag, M. (2012), 'Pilot, rollout and monte carlo tree search methods for job shop scheduling'. *Learning and Intelligent Optimization*. Hrsg. von Y. Hamadi & M. Schoenauer. 6th International Conference LION 6, 16.- 20. Jan 2012, Paris, Frankreich. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 160–174.

Schlimmer & Granger 1986

Schlimmer, J. C. & Granger, R. H. (1986), 'Incremental learning from noisy data', *Machine learning* 1.3, S. 317–354.

Scholz-Reiter & GörgeS u. a. 2009b

Scholz-Reiter, B.; GörgeS, M.; Jagalski, T. & Mehrsai, A. (2009b), 'Modelling and Analysis of Autonomously Controlled Production Networks', *IFAC Proceedings Volumes* 42.4, S. 846–851.

Scholz-Reiter & GörgeS u. a. 2009a

Scholz-Reiter, B.; GörgeS, M. & Philipp, T. (2009a), 'Autonomously controlled production systems—Influence of autonomous control level on logistic performance', *CIRP Annals* 58.1, S. 395–398.

Scholz-Reiter & Hamann 2008

Scholz-Reiter, B. & Hamann, T. (2008), 'The behaviour of learning production control', *CIRP Annals* 57.1, S. 459–462.

Scholz-Reiter & Rekersbrink u. a. 2010a

Scholz-Reiter, B.; Rekersbrink, H & Görges, M (2010a), 'Dynamic Flexible Flow Shop Problems - Scheduling Heuristics vs. Autonomous Control', *CIRP Annals* 59.1, S. 465–468.

Scholz-Reiter & Rekersbrink u. a. 2010b

Scholz-Reiter, B.; Rekersbrink, H. & Görges, M. (Jan. 2010b), 'Dynamic flexible flow shop problems—Scheduling heuristics vs. autonomous control', *CIRP Annals* 59.1, S. 465–468.

Scholz-Reiter & Beer u. a. 2004

Scholz-Reiter, B.; Beer, C. de & Peters, K. (2004), 'Autonomous control of shop floor logistics', *Manufacturing, Modelling, Management and Control 2004* 47, S. 47–56.

Scholz-Reiter & Freitag u. a. 2005

Scholz-Reiter, B.; Freitag, M.; Beer, C. de & Jagalski, T. (2005), 'Modelling and analysis of autonomous shop floor control'. *Proceedings of the 38th CIRP International Seminar on Manufacturing Systems*. 38th CIRP International Seminar on Manufacturing Systems, 16.-18. Mai 2005, Florianopolis, Brasilien. CD-ROM, S. 1–6.

Schönemann & Herrmann u. a. 2015

Schönemann, M.; Herrmann, C.; Greschke, P. & Thiede, S. (2015), 'Simulation of matrix-structured manufacturing systems', *Journal of Manufacturing Systems* 37, S. 104–112.

Schuh 2006

Schuh, G., Hrsg. (2006), *Produktionsplanung und-steuerung: Grundlagen, Gestaltung und Konzepte*. Berlin: Springer.

Shang & Wu u. a. 2020

Shang, Y; Wu, W; Liao, J; Jianbo, G; Su, J; Liu, W & Huang, Y (2020), 'Stochastic Maintenance Schedules of Active Distribution Networks based on Monte-Carlo Tree search', *IEEE Transactions on Power Systems* 35.5, S. 3940–3952.

Silver & Huang u. a. 2016

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Driessche, G. van den; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T.; Leach, M.; Kavukcuoglu, K.; Graepel, T. & Hassabis, D. (2016), 'Mastering the game of Go with deep neural networks and tree search', *Nature* 529.7587, S. 484–489.

Silver & Schrittwieser u. a. 2017

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; Chen, Y.; Lillicrap, T.; Hui, F.; Sifre, L.; Van Den Driessche, G.;

- Graepel, T. & Hassabis, D. (2017), 'Mastering the game of Go without human knowledge', *Nature* 550.7676, S. 354–359.
- Statistisches Bundesamt 2020
Statistisches Bundesamt, Hrsg. (2020), *Bruttoinlandsprodukt Für Deutschland 2019 Beigleitmaterial zur Pressekonferenz am 15. Januar 2020 in Berlin*. Wiesbaden, S. 1–50.
- Stegherr 2000
Stegherr, F (2000), *Reinforcement-Learning zur dispositiven Auftragssteuerung in der Variantenreihenproduktion*. Technische Universität München, Dissertation. München: Herbert Utz Verlag.
- Stoop & Wiers 1996
Stoop, P. P. & Wiers, V. C. (1996), 'The complexity of scheduling in practice', *International Journal of Operations & Production Management* 16.10, S. 37–53.
- Streim 1975
Streim, H. (1975), 'Heuristische Lösungsverfahren Versuch einer Begriffsklärung', *Zeitschrift für Operations Research* 19.5, S. 143–162.
- Stricker & Kuhnle u. a. 2018
Stricker, N.; Kuhnle, A.; Sturm, R. & Friess, S. (2018), 'Reinforcement learning for adaptive order dispatching in the semiconductor industry', *CIRP Annals* 67.1, S. 511–514.
- Stricker & Kuhnle u. a. 2021
Stricker, N.; Kuhnle, A.; Hofmann, C. & Deininger, P. (2021), 'Self-adjusting multi-objective scheduling based on Monte Carlo Tree Search for matrix production assembly systems', *CIRP Annals* 70.1, S. 381–384.
- Sturges 1926
Sturges, H. A. (März 1926), 'The Choice of a Class Interval', *Journal of the American Statistical Association* 21.153, S. 65–66.
- Sutton & Barto 2015
Sutton, R. S. & Barto, A. G. (2015), *Reinforcement learning: An Introduction*. 2. Aufl. Cambridge, London: MIT Press.
- Taillard 1994
Taillard, D. (1994), 'Parallel Taboo Search Techniques for the Job Shop Scheduling Problem', *ORSA Journal on Computing* 6.2, S. 108–117.
- Verderame & Elia u. a. 2010
Verderame, P. M.; Elia, J. A.; Li, J. & Floudas, C. A. (2010), 'Planning and Scheduling under Uncertainty: A Review Across Multiple Sectors', *Industrial & Engineering Chemistry Research* 49.9, S. 3993–4017.

Vieira & Herrmann u. a. 2003

Vieira, G. E.; Herrmann, J. W. & Lin, E. (2003), 'Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods', *Journal of Scheduling* 6.1, S. 39–62.

Vodopivec & Samothrakis u. a. 2017

Vodopivec, T.; Samothrakis, S. & Ster, B. (2017), 'On Monte Carlo tree search and reinforcement learning', *Journal of Artificial Intelligence Research* 60, S. 881–936.

Wang & Gao u. a. 2010

Wang, X.; Gao, L.; Zhang, C. & Shao, X. (2010), 'A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem', *International Journal of Advanced Manufacturing Technology* 51.5-8, S. 757–767.

Wannenwetsch 2014

Wannenwetsch, H. (2014), *Integrierte Materialwirtschaft und Logistik: Beschaffung, Logistik, Materialwirtschaft und Produktion*. Berlin, Heidelberg: Springer-Verlag.

Waschneck & Reichstaller u. a. 2018

Waschneck, B.; Reichstaller, A.; Belzner, L.; Altenmüller, T.; Bauernhansl, T.; Knapp, A. & Kyek, A. (2018), 'Deep reinforcement learning for semiconductor production scheduling'. *Proceedings of the 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC 2018)*. 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), 30.04. - 03-05. 2018, New York, USA. New York: IEEE, S. 301–306.

Westkämper & Löffler 2016

Westkämper, E & Löffler, C (2016), *Strategien der Produktion-Technologien*. Berlin, Heidelberg: Springer Vieweg, Berlin, Heidelberg.

Wiendahl 1997

Wiendahl, H.-P. (1997), *Fertigungsregelung : Logistische Beherrschung von Fertigungsabläufen auf Basis des Trichtermodells*. Berlin: Hanser.

Wiendahl 2019

Wiendahl, H.-P. (2019), *Betriebsorganisation für Ingenieure*. München: Carl Hanser Verlag.

Wiendahl & Hegenscheidt 2006

Wiendahl, H.-P. & Hegenscheidt, M. (2006), 'Verfügbarkeit von Montagesystemen'. *Montage in der industriellen Produktion: Ein Handbuch für die Praxis*. Hrsg. von B. Lotter & H.-P. Wiendahl. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 369–406.

Wiendahl & Reichardt u. a. 2014

Wiendahl, H.-P.; Reichardt, J. & Nyhuis, P. (2014), *Handbuch Fabrikplanung: Konzept, Gestaltung und Umsetzung wandlungsfähiger Produktionsstätten*. Berlin: Hanser.

Wu & Wu u. a. 2013

Wu, T; Wu, I & Liang, C (2013), 'Multi-objective Flexible Job Shop Scheduling Problem Based on Monte-Carlo Tree Search'. *TAAI '13: Proceedings of the 2013 Conference on*

Technologies and Applications of Artificial Intelligence. Hrsg. von D. Li. 2013 Conference on Technologies und Applications of Artificial Intelligence, 6.-8. Dez 2013, Taipei, Taiwan. Washington: IEEE, S. 73–78.

Xia & Wu 2005

Xia, W. & Wu, Z. (2005), 'An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems', *Computers & Industrial Engineering* 48.2, S. 409–425.

Zäpfel & Braune 2005

Zäpfel, G. & Braune, R. (2005), *Moderne Heuristiken der Produktionsplanung: am Beispiel der Maschinenbelegung*. München: Vahlen.

Zhang & Li u. a. 2007

Zhang, C.; Li, P.; Guan, Z. & Rao, Y. (2007), 'A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem', *Computers & Operations Research* 34.11, S. 3229–3242.

Zhang & Zheng u. a. 2012

Zhang, Z.; Zheng, L.; Li, N.; Wang, W.; Zhong, S. & Hu, K. (2012), 'Minimizing mean weighted tardiness in unrelated parallel machine scheduling with reinforcement learning', *Computers and Operations Research* 39.7, S. 1315–1324.

Zimmermann & Stache 2010

Zimmermann, W. & Stache, U. (2010), *Operations research: quantitative Methoden zur Entscheidungsvorbereitung*. München: Oldenbourg Wissenschaftsverlag.

Zitzler & Thiele 1999

Zitzler, E. & Thiele, L. (1999), 'Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach', *IEEE transactions on Evolutionary Computation* 3.4, S. 257–271.

Abschlussarbeiten

Literaturquellen nach dem Schema (A_<Nachname> <Jahr>) beziehen sich auf studentische Arbeiten am wbk Institut für Produktionstechnik, die durch den Autor der Dissertation angeleitet wurden.

A_Bickel 2021

Bickel, M. (2021), 'Formalisierung einer Matrixproduktion: Formulierung eines Linearen Programms für das Flexible Job Shop Scheduling Problem mit alternativen Vorgangssequenzen (FJSSP-MPP)'. Seminararbeit, Karlsruher Institut für Technologie, Karlsruhe, wbk Institut für Produktionstechnik.

A_Deininger 2020

Deininger, P. (2020), 'Monte Carlo Tree Search for Production Scheduling in Matrix Production'. Masterarbeit, Karlsruher Institut für Technologie, Karlsruhe, wbk Institut für Produktionstechnik.

A_Funk 2019

Funk, R. (2019), 'Hyperparameter-Tuning für Monte Carlo Tree Search - Hyper parameter optimization for Monte Carlo Tree Search'. Seminararbeit, Karlsruher Institut für Technologie, Karlsruhe, wbk Institut für Produktionstechnik.

A_Köhler 2019

Köhler, A. (2019), 'Aufbau eines Optimierungsmodells zur Bestimmung der exakten Lösungen eines Flexible Job Shop Problems in gurobi'. Seminararbeit, Karlsruher Institut für Technologie, Karlsruhe, wbk Institut für Produktionstechnik.

A_Liu 2021

Liu, X. (2021), 'A Hybrid MCTS Local Search Approach for Matrix Production Scheduling Problems'. Masterarbeit, Karlsruher Institut für Technologie, Karlsruhe, wbk Institut für Produktionstechnik.

A_Wolf 2019

Wolf, L. (2019), 'Konzeptionierung und Implementierung einer Regelung der Plananpassung eines echtzeitnahen Steuerungssystems einer Matrixproduktion'. Masterarbeit, Karlsruher Institut für Technologie, Karlsruhe, wbk Institut für Produktionstechnik.

A_Wurster 2019

Wurster, M. (2019), 'Entwicklung einer Produktionssteuerung basierend auf Monte-Carlo Tree Search für das Konzept der Matrixproduktion'. Masterarbeit, Karlsruhe Institut für Technologie, Karlsruher, wbk Institut für Produktionstechnik.

Abbildungsverzeichnis

1.1	Matrixproduktion mit den Produktionsmitteln M1-M6 und den Prozessen A-C	2
2.1	Räumliche Struktur ausgewählter Fertigungsprinzipien, angelehnt an (Wiendahl 2019)	6
2.2	Eigenschaften, Forschungsfragen und Einflussfaktoren auf die Steuerung einer Matrixproduktion anhand eines Beispiels aus den Produktionsmitteln M1-M6, den Vorgängen A-D und einem Auftrag mit vier alternativen Routen 1-4	8
2.3	Aachner PPS-Modell nach Schuh (2006)	10
2.4	Einordnung unterschiedlicher Ansätze der dynamischen Produktionsplanung und -steuerung in Anlehnung an Vieira; Herrmann u. a. (2003)	16
2.5	Ablauf des Monte Carlo Tree Search Verfahrens	20
2.6	Gerichteter Graph für drei Produktionsaufträge und 3 Maschinen	26
3.1	Ausgehend vom Knoten S_1 werden drei Rollouts durchgeführt. Die Bewertung der Aktion a_1 ergibt sich als Mittelwert der Zielfunktionen der Terminalzustände T_1, T_2, T_3 , da bei allen Zuständen die Aktion a_1 gewählt wurde. Die Bewertung für a_2 basiert nur auf T_1, T_3	44
4.1	Gesamtsystem aus Planzeitanpassung, Basisoptimierungsverfahren MCTS und lokaler Suche als Post-Optimierungsschritt	52
4.2	Schematische Darstellung des Aufbaus und der Anpassung der Planzeiten, angelehnt an einen Regelkreis nach (A_Wolf 2019)	57
4.3	<i>Concept-Drift</i> -Erkennung	58
4.4	Schematische Darstellung des Aufbaus der Verteilungsanalyse	61
4.5	Erkennung von Saisonalitäten	64
4.6	(links) Belegungsplan für Produktionsmittel mit Bearbeitungs- und Rüstvorgängen (grau). Die Farben fassen Produktionsaufträge zusammen. (rechts) Belegungsplan für Produktionsaufträge mit Bearbeitungs- und Transportvorgängen (grün)	65

4.7	Schematische Darstellung des Suchbaums, der Zustände $s_0 - s_3$ und der Aktionen $a_1 - a_3$ für die Produktionsmittel M1 bis M3, den Vorgang OP1 und die Aufträge 5 (blau), 7 (rot) und 15 (grün)	66
4.8	Hypervolumen der Pareto-Punkte L, G, W im dreidimensionalen Raum	68
4.9	Suchbaum mit geringer Exploration (links) und hoher Exploration (rechts)	69
4.10	Generierung und Einschränkung von Aktionen in der Expansionsphase	72
4.11	(links) Subtree-Parallelisierung mit identischen MO-MCTS-Konfigurationen und geteilten Aktionsmengen. (Mitte) Root-Parallelisierung mit unterschiedlichen MO-MCTS-Konfigurationen und identischen Aktionsmengen. (rechts) Rollout-Parallelisierung	82
4.12	Schematischer Aufbau des hybriden Systems aus MCTS und lokaler Suche	83
4.13	Darstellung des gerichteten Graphen für ein fiktives Beispiel aus drei Maschinen und zwei Produktionsaufträgen mit Transport- und Rüstvorgängen	85
4.14	Schematische Darstellung der Ermittlung des geeigneten Intervalls für den Vorgang v	89
4.15	Gesamtarchitektur bestehend aus der Produktionssteuerung, einer Visualisierung, einer Produktionssimulation und einer Referenzoptimierung zur Generierung von Referenzlösungen	98
5.1	Vorgehen zur Untersuchung der Produktionssteuerung	101
5.2	Schematisches Layout der Endmontage für Getriebemotoren	104
5.3	Reale Bearbeitungszeiten (1% - 99% Quantil) eines repräsentativen automatisierten Produktionsmittels sowie die geschätzte Log-Normalverteilung und die geschätzte Normalverteilung, angelehnt an (A_Wolf 2019)	105
5.4	Realdaten eines Produktionsmittels über drei Monate. Nach dem ersten Monat findet ein <i>Concept Drift</i> statt mit einer Reduktion des Mittelwerts. Das Produktionsvolumen des ersten Monats liegt deutlich über dem Niveau der beiden Folgemonate, angelehnt an (A_Wolf 2019)	106
5.5	EWMA-Regelkarte zur <i>Concept-Drift</i> -Detektion mit der Schätzung des Mittelwerts und den Eingriffsgrenzen, angelehnt an (A_Wolf 2019)	107
5.6	Größe des stabilen Zeitfensters der ADWIN-Methode und Mittelwertschätzung der ADWIN-Methode, angelehnt an (A_Wolf 2019)	108
5.7	Bearbeitungszeit pro Schicht und durchschnittliche Bearbeitungszeit über alle Schichten für ein Produktionsmittel sowie die Bearbeitungszeit pro Wochentag für einen ausgewählten Montageplatz, angelehnt an (A_Wolf 2019)	109

5.8 Einfluss der Anzahl betrachteter Produktionsaufträge und der Einschränkung zulässiger Aktionen auf den Suchbaum, angelehnt an (A_Deiningner 2020)	113
5.9 Hypervolumen und gleichgewichteter Zielfunktionswert für die Zielkriterien Durchlaufzeit, Termintreue und Makespan, angelehnt an (A_Deiningner 2020)	115
5.10 Auswirkung der Anzahl parallel betrachteter Produktionsaufträge auf den erreichten Lösungsraum für Durchlaufzeit, Termintreue und Makespan, angelehnt an (A_Deiningner 2020)	116
5.11 Boxplot für die unterschiedlichen Auswahlverfahren und die Zielkriterien Durchlaufzeit, Termintreue und Makespan, angelehnt an (A_Deiningner 2020)	119
5.12 Ergebnisse des Experimentier- und des Produktivbetriebs hinsichtlich Durchlaufzeit, Verspätung, Makespan und des gleichgewichteten Zielfunktionswerts, angelehnt an (A_Deiningner 2020)	121
5.13 Lösungen am Ende der Rollout-Phase bei Root-Parallelisierung mit sieben Threads mit unterschiedlichen Auftragslimits und Zielgewichtungen für den 12x5 Anwendungsfall, angelehnt an (A_Wurster 2019)	123
5.14 Terminalzustände je Iteration bei rollierender Zielgewichtungen der Einzelziele in der Selektionsphase, angelehnt an (A_Wurster 2019)	124
5.15 Hypervolumen und gleichgewichteter Zielfunktionswert für die Tabu-Suche mit Nachbarschaftsstruktur N1, N4, N6, Tauschoperationen mit Nachbarschaftsstruktur N4 und Verschiebeoperationen mit Nachbarschaftsstruktur N5 sowie LSONE und MO-MCTS, angelehnt an (A_Liu 2021)	126
5.16 Hypervolumen und gleichgewichteter Zielfunktionswert für die Modifikation von LSONE, LSONE nach Gao; Sun u. a. (2008) und für MO-MCTS, angelehnt an (A_Liu 2021)	127
5.17 Hypervolumen und Anteil der erfolgreichen Veränderungen abhängig von der Vorsortierung, angelehnt an (A_Liu 2021)	128
5.18 Hypervolumen abhängig von der Anzahl betrachteter Aufträge und der Wahl der Ausgangslösungen für das 12x5 Problem, angelehnt an (Liu 2021)	129
5.19 Auswertung statischer Toleranzwerte nach Hypervolumen (a), gleichgewichteter Zielfunktionswert (b) und Anteil der Pareto-Lösungen durch lokale Suche an der Pareto-Menge (c). Auswertung dynamischer Toleranzwerte nach dem Hypervolumen (d), angelehnt an (A_Liu 2021)	131
5.20 Hypervolumen (a) und Ausführungszeit (b) abhängig von der Anzahl Iterationen und Reallokationen sowie der Anzahl betrachteter Pareto-Punkte, angelehnt an (A_Liu 2021).	132

5.21 Hypervolumen und gleichgewichteter Zielfunktionswert abhängig von den Filterkombinationen zur Einschränkung zulässiger Aktionen und der Anzahl parallel betrachteter Produktionsaufträge, angelehnt an (A_Liu 2021)	133
5.22 Lage der MO-MCTS-Lösungen und der Lösungen der lokalen Suche im Lösungsraum für Makespan, Durchlaufzeit, Verspätung, maximale Arbeitslast und Gesamtarbeitslast, angelehnt an (A_Liu 2021)	134
5.23 (a) Entwicklung der Lösungsgsgüte über 70.000 Iterationen für die Optimierung der kumulierten Verspätung für die 12x5 Problem Instanz. (b) Bester C-Wert abhängig von der Iterationszahl, angelehnt an (A_Funk 2019)	137
5.24 Verlauf des Zielfunktionswerts und des dynamischen C-Werts über 8.000 Iterationen für die 12x5 Problem Instanz, angelehnt an (A_Deiningner 2020)	138
5.25 Verzweigungsfaktor, C-Wert, Hypervolumen und Zielerreichung für die 12x5 Problem Instanz, angelehnt an (A_Deiningner 2020)	140
5.26 Entwicklung des Hypervolumens während des Experimentierbetriebs und im Produktivbetrieb (12x5 Problem Instanz), angelehnt an (A_Deiningner 2020)	141
5.27 Anzahl der Iterationen mit und ohne Parallelisierung (a), durchschnittlicher gleichgewichteter Zielfunktionswert ohne Parallelisierung und mit Subtree-Parallelisierung (b), Hypervolumen ohne Parallelisierung, mit Subtree-Parallelisierung und mit Root-Parallelisierung (c) für 8 virtuelle CPU-Kerne, angelehnt an (A_Wurster 2019)	143
5.28 Entwicklung des Hypervolumens über die Laufzeit für den Basisalgorithmus und die hybride Version für das 18x8 Problem, angelehnt an (A_Liu 2021)	145
5.29 Hypervolumen und Zielerreichung über die Laufzeit für hybride Ansätze mit und ohne intelligente Einschränkung zulässiger Aktionen (18x8 Problem Instanz), angelehnt an (A_Liu 2021)	146
5.30 Pareto-Punkte des exakten Verfahrens nach je 60 Minuten sowie Lösungen des MO-MCTS nach 10 Sekunden Optimierung (a) und (b) bzw. nach 120 Sekunden (c) und (d) (5x3 Problem), angelehnt an (A_Deiningner 2020)	149
5.31 Belegungsplan der Getriebemontage. Untere Zeile: initialer Belegungsplan, obere Zeile: letzter Belegungsplan, angelehnt an (A_Liu 2021)	152
5.32 Belegungsplan aus Produktionsauftragssicht des Beispielfalls aus der Getriebemontage, angelehnt an (A_Li; Chang u. a. 2009)	154
A5.1 Belegungsplan mit optimaler kumulierter Durchlaufzeit, für eine mittlere Einschränkung des Aktionsraums und einem parallel betrachteten Produktionsauftrag, angelehnt an (A_Deiningner 2020).	XXVIII

A5.2 Belegungsplan mit sehr gutem Makespan, für eine sehr starke Einschränkung des Aktionsraums und 18 parallel betrachtete Produktionsaufträge, angelehnt an (A_Deininger 2020)	XXIX
A6.1 Maximales Hypervolumen für unterschiedliche C-Werte für das 12x5 Problem bei Verwendung der Filterkombinationen mit mittlerer Einschränkung und 10 parallel betrachteten Produktionsaufträgen, angelehnt an (A_Deininger 2020)	XXX
A7.1 Grafik (a) zeigt den Belegungsplan aus Sicht der Produktionsmittel (oben) und aus Sicht der Produktionsaufträge (unten) für die optimale Lösung bezüglich der kumulierten Durchlaufzeit. Grafik (b) zeigt die analogen Abbildungen für den optimalen Makespan, angelehnt an (A_Deininger 2020)	XXXI

Tabellenverzeichnis

2.1	Maschinenart und Anordnung der mehrstufigen Produktion α_1	12
3.1	Bewertung der Scheduling-Ansätze aus dem Stand der Forschung anhand der Anforderungen der Matrixproduktion nicht erfüllt: ○, teils erfüllt: ◐, erfüllt: ●	48
5.1	Konfigurationen zur Untersuchung der Auswirkungen der Aktionsgenerierung auf die Gestalt des Suchbaums nicht aktiv: ○, aktiv: ●, Reihenfolge der Anwendung der Sortierkriterien: 1 - 7, variabel: ~	112
5.2	Wirkung der Sortierkriterien und Filterkombinationen auf den Suchbaum und Wirkung der Filterkombinationen auf den erreichten Lösungsraum	112
5.3	Untersuchung der Sortierung der Produktionsaufträge in der Expansionsphase nicht aktiv: ○, aktiv: ●, Anwendung der Sortierkriterien: 1 - 7	117
5.4	Auswirkung der Sortierungskriterien der Produktionsaufträge auf den erreichten Lösungsraum	118
5.5	Maximal erreichter normierter Zielfunktionswert für die untersuchten Sortierkombinationen der Produktionsaufträge, angelehnt an (A_Deiningger 2020)	118
5.6	Konfiguration für Rollout und Expansion im Experimentierbetrieb nicht aktiv: ○, aktiv: ●, Anwendung der Sortierkriterien: 1 - 7, veränderbarer Wert mit Startwert $x: \tilde{x}$	120
5.7	Konfiguration der Experimente im Experimentierbetrieb	120
5.8	Filterkombinationen in der Aktionsgenerierung der einzelnen parallelen Prozesse bei den Experimenten mit Root-Parallelisierung	122
5.9	Alternierende Zielgewichtung	124
5.10	Konfiguration des MO-MCTS als Basisverfahren zur Generierung der Ausgangslösung zur Untersuchung der lokalen Suche	125
5.11	Konfiguration der Experimente zur Untersuchung des Explorationsfaktors	137
5.12	Sensitivitätsanalysen des dynamischen C-Werts	139

5.13	Untersuchung der Subtree-Parallelisierung	142
5.14	Untersuchung des Einflusses der lokalen Suche auf den Anstieg der Lösungsgüte	144
5.15	Untersuchung der absoluten Lösungsgüte anhand des 5x3 Problems	148
5.16	Konfiguration der Experimente zur Lösung des Kacem-Problems	150
5.17	Ergebnisse des Kacem-Vergleichs, nach (A_Deininger 2020), (A_Liu 2021) nicht erreicht: ○, erreicht: ●, keine Angabe: -, optimale Lösung:*	151
A1	Maschinenart und Anordnung der einstufigen Produktion α_1	XV
A1	Produktvarianten, früheste Start- und Entzeitpunkte der Produktionsaufträge des 18x10 Problems	XVI
A2	Vorgangsreihenfolgen der Produktvarianten des 18x10 Problems	XVII
A3	Bearbeitungsdauer in Zeiteinheiten (ZE) der Vorgänge 1 - 3 und die initialen Rüstzustände der Produktionsmittel M1 - M8 des 18x10 Problems nicht gerüstet: ○, gerüstet: ●	XVII
A4	Mögliche Reihenfolgen der Vorgänge PP1 - PP3 der Produktvarianten des 5x3 Problems	XVII
A5	Produktvarianten, früheste Start- und Entzeitpunkte der Produktionsaufträge des 5x3 Problems	XVII
A1	Zuordnung der Vorgänge auf die Produktionsmittel (Station)	XVIII
A2	Zuordnung der Vorgänge pro Produktvariante	XIX
A3	Dauer des Vorgangs in Minuten	XIX

Anhang

A1 Klassifikation von Produktionssteuerungsproblemen für die einstufige Fertigung

Maschinenart und Anordnung α_1

Für die **Maschinenart und die Anordnung** α_1 werden für die einstufige Fertigung folgende Ausprägungen unterschieden.

- \emptyset : Alle Aufträge benötigen einen Prozessschritt und es steht nur eine Maschine zur Bearbeitung zur Verfügung (Ein-Maschinen-Problem).
- P : Im Gegensatz zu \emptyset stehen mehrere identische parallele Maschinen zur Verfügung.
- Q : Es stehen identische, parallele Maschinen zur Verfügung, deren Bearbeitungszeiten sich unterscheiden.
- R : Die Maschinen sind heterogen und parallel, sodass die Bearbeitungszeiten vom Produktionsauftrag und der Maschine abhängen.

Der Parameter α_2 gibt die Anzahl paralleler Maschinen an. Tabelle A1 fasst die Werte des Parameters α_1 zusammen.

Tabelle A1: Maschinenart und Anordnung der einstufigen Produktion α_1

α_1	Einstufige Produktion			
	\emptyset	P	Q	R
Maschinenanzahl	1	> 1	> 1	> 1
Maschinenart	gleich	gleich	gleich	verschieden
Bearbeitungszeit	gleich	gleich	maschinen- abh.	maschinen- & auftragsabh.
Prozessschritte	1	1	1	1
Prozesssequenz	-	-	-	-
Durchläufe pro Maschine	-	-	-	-
Alternative Maschinen	nein	ja	ja	ja

A2 Generische Anwendungsprobleme

Tabelle A1: Produktvarianten, früheste Start- und Endzeitpunkte der Produktionsaufträge des 18x10 Problems

Produktionsaufträge 1-9					Produktionsmittel 10-18				
Produktionsauftrag	Produktvariante	Früherster Startzeitpunkt	Spätester Endzeitpunkt	Priorität	Produktionsauftrag	Produktvariante	Früherster Startzeitpunkt	Spätester Endzeitpunkt	Priorität
1	P1	15:50	17:00	0	10	P1	15:50	18:00	0
2	P2	15:50	17:00	0	11	P2	15:50	18:00	0
3	P3	15:50	17:00	0	12	P3	15:50	18:00	0
4	P1	15:50	17:00	0	13	P1	15:50	18:00	0
5	P2	15:50	17:00	0	14	P2	15:50	18:00	0
6	P3	15:50	17:10	0	15	P3	15:50	18:10	0
7	P1	15:50	17:10	0	16	P1	15:50	18:10	0
8	P2	15:50	17:20	0	17	P2	15:50	18:30	0
9	P3	15:50	17:50	0	18	P3	15:50	18:30	0

Tabelle A2: Vorgangsreihenfolgen der Produktvarianten des 18x10 Problems

Produktvariante	Produktionssequenz		
Produktvariante 1	PP1	PP2	PP3
	PP2	PP1	PP3
	PP1	PP2	PP1
Produktvariante 2	PP1	PP2	PP3
	PP2	PP1	PP3
	PP3	PP2	PP1
Produktvariante 3	PP1	PP2	PP3
	PP2	PP1	PP3
	PP3	PP2	PP1

Tabelle A3: Bearbeitungsdauer in Zeiteinheiten (ZE) der Vorgänge 1 - 3 und die initialen Rüstzustände der Produktionsmittel M1 - M8 des 18x10 Problems
nicht gerüstet: ○, gerüstet: ●

	Dauer	Produktionsmittel							
		M1	M2	M3	M4	M5	M6	M7	M8
Vorgang 1	10	○	●	●	●	○	●	○	○
Vorgang 2	15	●	○	○	○	○	○	●	○
Vorgang 3	20	○	○	○	○	●	○	○	●

Tabelle A4: Mögliche Reihenfolgen der Vorgänge PP1 - PP3 der Produktvarianten des 5x3 Problems

Produktvariante	Produktionssequenz		
Produktvariante 1	PP1	PP2	PP3
	PP2	PP1	PP3
Produktvariante 2	PP1	PP2	PP3
	PP3	PP2	PP1
Produktvariante 3	PP2	PP1	PP3

Tabelle A5: Produktvarianten, früheste Start- und Endzeitpunkte der Produktionsaufträge des 5x3 Problems

Produktionsauftrag	Produktvariante	Frühester Startzeitpunkt	Spätester Endzeitpunkt	Priorität
1	P1	15:50	16:56	0
2	P2	15:50	16:56	0
3	P3	15:50	17:13	0
4	P1	15:50	17:21	0
5	P2	15:50	17:21	0

A3 Anwendungsfall aus der Getriebemontage

Es gibt sechs Produktionsstationen für die Getriebemontage (GM1 - GM6), eine Ölbefüllungsstation (ÖL), sechs Motormontagestationen (MM1 - MM6), drei Teststationen (T1 - T3) und eine Lackstation (L).

Es gibt fünf Prozessschritte der Getriebemontage (OG1 - OG5), die in jeweils drei Unterschritten aufgeteilt werden können. Somit ergeben sich die Prozessschritte (OG11 - OG53). Für die Montage der Motoren werden fünf Schritte durchgeführt (OM1 - OM5), bei der Ölbefüllung gibt es zwei unterschiedliche Prozessschritte abhängig von der Ölsorte (OO1, OO2). Der Funktionstest gliedert sich in fünf Testschritte (OT1 - OT5). In der Lackierung stehen Speziallacke und Standardlacke zur Verfügung (OL1, OL2).

Tabelle A1 listet auf, welche Vorgänge von welchem Produktionsmittel durchführbar sind.

Tabelle A1: Zuordnung der Vorgänge auf die Produktionsmittel (Station)

Station	Vorgang	Station	Vorgang	Station	Vorgang
GM1	OG11 - OG53	MM1	OM1 - OM5	ÖL	OEL1, OEL2
GM2	OG11 - OG53	MM2	OM1 - OM5	T1	OT1 - OT5
GM3	OG11 - OG53	MM3	OM1 - OM5	T2	OT1 - OT5
GM4	OG11 - OG53	MM4	OM1 - OM5	T3	OT1 - OT5
GM5	OG11 - OG53	MM5	OM1 - OM5	L1	OL1 - OL2
GM6	OG11 - OG53	MM6	OM1 - OM5		

Die Zuordnung der Vorgänge auf die Produktvarianten ist in Tabelle A2 zusammengefasst. Wird ein Defekt bei einem Test festgestellt, müssen alle Vorgänge erneut durchlaufen werden. Das Produkt P6 entspricht dem Getriebe des Getriebemotors P5. Die Variante P7 ist das Produkt P2 mit einer Speziallackierung. P8 und P9 entsprechen P1 und P3 mit Spezialölen.

Tabelle A2: Zuordnung der Vorgänge pro Produktvariante

Produktvariante	Benötigter Vorgang
P1	OG11, OG12, OG13, OEL1, OM1, OT1, OL1
P2	OG21, OG22, OG23, OEL1, OM2, OT2, OL1
P3	OG31, OG32, OG33, OEL1, OM3, OT3, OL1
P4	OG41, OG42, OG43, OEL1, OM4, OT4, OL1
P5	OG51, OG52, OG53, OEL1, OM5, OT5, OL1
P6	OG51, OG52, OG53, OT5, OL1
P7	OG21, OG22, OG23, OEL1, OM2, OT2, OL2
P8	OG11, OG12, OG13, OEL2, OM1, OT1, OL1
P9	OG31, OG32, OG33, OEL2, OM3, OT3, OL1

Die Dauer der Vorgänge in Minuten wird in Tabelle A3 zusammengefasst.

Tabelle A3: Dauer des Vorgangs in Minuten

Vorgang	Dauer	Vorgang	Dauer	Vorgang	Dauer	Vorgang	Dauer
OG11	5	OM1	4	OT1	4	OEL1	1
OG12, OG21	3	OM2	4	OT2	4	OEL2	2
OG13, OG22, OG31, OG32, OG41, OG51	9	OM3	11	OT3	6	OL1	3
OG23, OG33, OG42, OG52	11	OM4	11	OT4	8	OL2	9
OG43, OG53	15	OM5	15	OT5	7		

A4 Gemischt-ganzzahliges Optimierungsproblem

Formelzeichen

Bezeichner	Beschreibung	Wertebereich	Einheit
i	Produktionsauftrag ($i, i' \in I$)	I	-
j	Bearbeitungsvorgang ($j, j' \in O$)	O	-
$p_{(i)}$	Bearbeitungsvorgangsreihenfolge für Produktionsauftrag i ($p_{(i)} \in P_i$)	P_i	-
m	Produktionsmittel ($m \in M$)	M	-
J	Menge der Produktionsaufträge	-	-
O	Menge der Bearbeitungsvorgänge	-	-
P_i	Menge alternativer Bearbeitungsvorgangsreihenfolgen für den Produktionsauftrag i	-	-
M	Menge von verfügbaren Produktionsmitteln	-	-
M_j	Menge von Produktionsmitteln, welche Bearbeitungsvorgang j ausführen können ($M_j \subseteq M$)	-	-
$M_j \cap M_{j'}$	Menge von Produktionsmitteln, welche Bearbeitungsvorgang j und j' ausführen können	-	-
O_i	Menge von Bearbeitungsvorgängen aller Bearbeitungsvorgangsreihenfolgen für den Produktionsauftrag i ($O_i = \cup_{p_{(i)} \in P_i} O_{ip_{(i)}}$)	O	-
$O_{ip_{(i)}}$	Geordnete Menge von Bearbeitungsvorgängen der Bearbeitungsvorgangsreihenfolgen $p_{(i)}$ für den Produktionsauftrag i ($O_{ip_{(i)}} \in O$), wobei $O_{ip_{(i)}f_{(i)}}$ das erste und $O_{ip_{(i)}l_{(i)}}$ das letzte Element (respektive Bearbeitungsvorgang) von $O_{ip_{(i)}}$ darstellen	O	-
t_{jm}	Benötigte Zeit für Bearbeitungsvorgangs j auf Produktionsmittel m	\mathbb{R}_0^+	s
WI	Eingangslager	-	-
WO	Ausgangslager	-	-

$d_{mm'}$	Transportzeit von Produktionsmittel m zu m' , wobei $d_{vv} = 0$	\mathbb{R}_0^+	s
$d_{WI,m}$	Transportzeit vom Eingangslager WI zum Produktionsmittel m	\mathbb{R}_0^+	s
$d_{m,WO}$	Transportzeit vom Produktionsmittel m zum Ausgangslager WO	\mathbb{R}_0^+	s
f_j	Rüstzustand für Bearbeitungsvorgang j	F	-
f_{0m}	Errüstzustand eines Produktionsmittels m zu Beginn des Programms	F	-
F	Menge aller Rüstzustände ($F = \cup_{j \in O} \{f_j\}$)	-	-
$u_{f_j f_{j'}}$	Zeit des Umrüstens zwischen Bearbeitungsvorgang j und j' , wobei $u_{f_j f_j} = 0$	\mathbb{R}_0^+	s
E_i	Frühster Beginn von Bearbeitung und Transport des Produktionsauftrags i	\mathbb{R}_0^+	s
A_m	Frühster Beginn zur Umrüstung und Bearbeitung von Produktionsaufträgen auf Produktionsmittel m	\mathbb{R}_0^+	s
R_i	Spätester Fertigstellungszeitpunkt für Auftrag i ab Programmstart	\mathbb{R}_0^+	s
$Z_{ip(i)}$	Entscheidungsvariable: 1, wenn Bearbeitungsvorgangsreihenfolge $p_{(i)}$ ausgewählt wird; 0, sonst	$\{0, 1\}$	-
$X_{ip(i)jm}$	Entscheidungsvariable: 1, wenn Produktionsmittel m für Vorgang $O_{ip(i)j}$ ausgewählt wird; 0, sonst	$\{0, 1\}$	-
$S_{ip(i)jm}$	Entscheidungsvariable: Startzeit vom Bearbeitungsvorgang $O_{ip(i)j}$ auf Produktionsmittel m	\mathbb{R}_0^+	s
$Y_{ij'j'm}$	Entscheidungsvariable: 1, wenn Bearbeitungsvorgang $O_{ij'}$ ein Vorgänger von Bearbeitungsvorgang $O_{j'm}$ auf Produktionsmittel m ist; 0, sonst	$\{0, 1\}$	-
C_i	Entscheidungsvariable: Durchlaufzeit des Produktionsauftrags i	\mathbb{R}_0^+	s
C_{max}	Entscheidungsvariable: Makespan	\mathbb{R}_0^+	s
G_i	Entscheidungsvariable: Verspätung des Auftrages i	\mathbb{R}_0^+	s
L	Eine große Zahl	\mathbb{R}_0^+	s

Modellierung des gemischt-ganzzahligen Optimierungsproblems

Das im Folgenden vorgestellte Optimierungsmodell basiert auf den Arbeiten von (A_Bickel 2021) und (A_Köhler 2019). Das Optimierungsmodell minimiert entweder den Makespan, die Verspätung oder die kumulierte Durchlaufzeit unter Berücksichtigung der Einschränkungen und Freiheitsgrade der Matrixproduktion. Zum Startzeitpunkt $t_0 = 0s$ sind alle geplanten Produktionsaufträge $i, i' \in I$ noch nicht begonnen. Ein Produktionsauftrag i erfordert mehrere Bearbeitungsvorgänge $j \in J$. Alternative Bearbeitungsvorgangsreihenfolgen werden in einer Menge aus geordneten Bearbeitungsvorgängen P_i definiert.

Es stehen verschiedene Produktionsmittel $m, m' \in M$ zur Verfügung. Produktionsmittel können verschiedene Bearbeitungsvorgänge $j, j' \in J$ ausführen. Alle Produktionsmittel, die einen bestimmten Bearbeitungsvorgang j ausführen können, befinden sich in der Menge M_j . Um den Bearbeitungsvorgang j ausführen zu können, muss sich ein Produktionsmittel $m \in M$ im Rüstzustand s_j befinden. Zu Beginn befindet sich jedes Produktionsmittel im Zustand f_{0m} , welcher einem Zustand s_k entspricht.

Sowohl für Produktionsaufträge als auch für Produktionsmittel kann ein frühester Startzeitpunkt $E_i > 0$ definiert werden.

Produktionsaufträge i, i' befinden sich zu Beginn im Wareneingangslager WI und müssen von dort aus zum ersten Produktionsmittel m zur Bearbeitung gebracht werden. Nach dem letzten Arbeitsschritt auf Maschine m' muss der Produktionsauftrag i ins Warenausgangslager WO transportiert werden.

Restriktionen

$$\sum_{p_i \in P_{(i)}} Z_{ip_{(i)}} = 1, \quad \forall i \in I \quad 0.1$$

Formel 0.1 beschränkt die Zuweisung einer Bearbeitungsvorgangsreihenfolge je Produktionsauftrag. Diese Restriktion stellt sicher, dass für jeden Produktionsauftrag $i \in I$ genau eine Bearbeitungsvorgangsreihenfolge $p_i \in P_{(i)}$ ausgewählt ist.

$$\sum_{m \in M_j} X_{ip_{(i)}jm} = Z_{ip_{(i)}}, \quad \forall i \in I, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}} \quad 0.2$$

Die Zuweisung jedes Bearbeitungsvorgangs der gewählten Bearbeitungsvorgangsreihenfolge auf ein Produktionsmittel wird durch Formel 0.2 sichergestellt. Die Restriktion verlangt, dass für alle Produktionsaufträge $i \in I$ für jeden Bearbeitungsvorgang der ausgewählten Bearbeitungsvorgangsreihenfolge $j \in O_{ip(i)}$ ein Produktionsmittel ausgewählt sein muss.

$$S_{ip(i)jm} \leq X_{ip(i)jm} \cdot L, \quad \forall i \in I, \forall p(i) \in P_i, \forall j \in O_{ip(i)}, \forall m \in M_j \quad 0.3$$

Die Restriktion 0.3 beschränkt die Startzeitpunkte der Bearbeitungsvorgänge. Sie garantiert, dass der Startzeitpunkt jedes Bearbeitungsvorgangs j für alle Reihenfolgen aus P_i für alle Produktionsmittel aus M kleiner ist als das Produkt aus der Binärvariable $X_{ip(i)jm}$ und einer sehr großen Zahl L . Da für nicht aktive Produktionsmittel die Binärvariable den Wert null annimmt, wird sichergestellt, dass nicht benötigte Bearbeitungsvorgänge auf diesem Produktionsmittel die Startzeit null erhalten.

$$S_{ip(i)jm} + (1 - X_{ip(i)jm}) \cdot L \geq E_i + d_{WI,m}, \quad \forall i \in I, \forall p(i) \in P_i, \forall m \in M_j, j = O_{ip(i)f(i)} \quad 0.4$$

Restriktion 0.4 stellt sicher, dass der Start des ersten Bearbeitungsvorgangs auf dem Produktionsmittel m frühestens nach dem frühesten Startzeitpunkt des Produktionsauftrags E_i zuzüglich der Transportdauer von Wareneingangslager (WI) zum Produktionsmittel $d_{WI,m}$ beginnen kann. Für Produktionsmittel, die nicht für den Produktionsschritt ausgewählt wurden, ergibt sich eine sehr hohe frühe Startzeit.

$$S_{ip(i)jm'} + (2 - X_{ip(i)jm'} - X_{ip(i)j-1,m}) \cdot L \geq S_{ip(i)j-1,m} + t_{j-1,m} + d_{mm'}, \quad \forall i \in I, \forall p(i) \in P_i, j \in O_{ip(i)} \setminus \{O_{ip(i)f(i)}\}, \forall m \in M_{j-1}, \forall m' \in M_j \quad 0.5$$

Die Restriktion 0.5 behandelt die Einhaltung der Reihenfolgebeziehung innerhalb der Bearbeitungsvorgänge eines Produktionsauftrags auf den Produktionsmitteln. Wird für einen Produktionsauftrag i der Bearbeitungsvorgang $j - 1$ auf dem Produktionsmittel m ausgeführt ($X_{ip(i)j-1,m} = 1$) und ist gemäß der Bearbeitungsreihenfolge $p(i)$ der nächste Bearbeitungsvorgang j auf dem Produktionsmittel m' auszuführen ($X_{ip(i)jm'} = 1$), so kann der Start $S_{ip(i)jm'}$ von Schritt j frühestens nach Abschluss des vorhergehenden Schrittes $j - 1$ begonnen werden. Dieser Zeitpunkt ergibt sich als Summe aus dem Start dieses Vorgangs $S_{ip(i)j-1,m}$ und seiner Dauer $t_{j-1,m}$ zuzüglich der Transportdauer $d_{mm'}$ von Produktionsmittel m zu Produktionsmittel m' .

$$S_{ip(i)jm} + (1 - X_{ip(i)jm}) \cdot L \geq A_m + u_{f_{0m}f_j},$$

$$\forall i \in I, \forall p(i) \in P_i, \forall j \in O_{ip(i)}, \forall m \in M_j \quad 0.6$$

Die Restriktion 0.6 limitiert die Verfügbarkeit eines Produktionsmittels. Die früheste Belegung ergibt sich aus dem frühesten Belegungszeitpunkt A_m und der Dauer des Rüstvorgangs vom initialen Rüstzustand f_{0k} zum Rüstzustand des ersten Bearbeitungsvorgangs f_j .

$$S_{ip(i')j'm} + (1 - Y_{ij'j'm}) \cdot L \geq S_{ip(i)jm} + t_{jm} + u_{f_jf_{j'}},$$

$$\forall i \in I, \forall p(i) \in P_i, \forall j \in O_{ip(i)}, \forall i' \in I, \forall p(i') \in P_{i'}, \forall j' \in O_{ip(i')}, \forall m \in M_j \cap M_{j'} \quad 0.7$$

Die Restriktion 0.7 stellt sicher, dass auf einem Produktionsmittel stets nur ein Vorgang gleichzeitig möglich ist. Der Start des Bearbeitungsvorgangs j' auf dem Produktionsmittel m ist gegeben durch $S_{ip(i')j'm}$. Wenn j' Vorgänger des Bearbeitungsvorgangs j auf dem Produktionsmittel m ist ($Y_{ij'j'm} = 1$), ist der früheste Startzeitpunkt gegeben durch den Startzeitpunkt des Vorgängers $S_{ip(i)jm}$ zuzüglich der Bearbeitungszeit t_{jm} des Vorgangs j

des Auftrags i und der Rüstzeit $u_{f_j f_{j'}}$ zur Herstellung des Rüstzustands für den Vorgang j' ausgehend von j .

$$Y_{ijijk} = 0, \quad \forall i \in I, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}}, \forall m \in M_j \quad 0.8$$

Die Nebenbedingung 0.8 stellt sicher, dass der Bearbeitungsvorgang j nicht sein eigener Vorgänger sein darf.

$$C_i \geq (S_{ip_{(i)}j'k'} + t_{j'k'} + d_{k',WO} - S_{ip_{(i)}jk} + d_{WI,k}) \cdot X_{ip_{(i)}jk} \cdot X_{ip_{(i)}j'k'}, \\ \forall i \in I, \forall p_{(i)} \in P_i, \forall k \in M_j, j = O_{ip_{(i)}f_{(i)}}, \forall k' \in M_{j'}, j' = O_{ip_{(i)}l_{(i)}} \quad 0.9$$

Die Restriktion 0.9 ermittelt die Durchlaufzeit C_i des Produktionsauftrags i . Dieser ergibt sich aus der Differenz des Startzeitpunkts des letzten Vorgangs, zuzüglich der Bearbeitungszeit und der Transportzeit zum Ausgangslager, abzüglich der Startzeit des ersten Bearbeitungsvorgangs, ergänzt um den Transport zum ersten Produktionsmittel. Damit umfasst die Durchlaufzeit die Dauer vom Verlassen des Eingangslagers bis zur Ankunft im Ausgangslager.

Für die Optimierung gelten folgende Wertebereiche und Nichtnegativitätsbedingungen:

$$G_i \geq 0, \quad \forall i \in I \quad 0.10$$

$$Z_{ip_{(i)}} \in \{0, 1\}, \quad \forall i \in I, \forall p_{(i)} \in P_i \quad 0.11$$

$$X_{ip_{(i)}jm} \in \{0, 1\}, \quad \forall i \in I, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}}, \forall m \in M_j \quad 0.12$$

$$S_{ip_{(i)}jm} \geq 0, \quad \forall i \in I, \forall p_{(i)} \in P_i, \forall j \in O_{ip_{(i)}}, \forall m \in M_j \quad 0.13$$

$$Y_{ij'i'j'm} \in \{0, 1\}, \quad \forall i \in I, \forall j \in O_i, \forall i' \in I, \forall j' \in O_{i'}, \forall m \in M_j \cap M_{j'} \quad 0.14$$

$$Y_{ij'i'j'm} \in \{0, 1\}, \quad \forall i \in I, \forall j \in O_i, \forall i' \in I, \forall j' \in O_{i'}, \forall m \in M_j \cap M_{j'} \quad 0.15$$

$$C_i \geq 0, \quad \forall i \in I \quad 0.16$$

Zielfunktionen

Formel 0.17 minimiert den Makespan.

$$f_1 = \min\{\max(C_i)\}, \quad \forall i \in I \quad 0.17$$

Auf die Minimierung der kumulierten Durchlaufzeit zielt Formel 0.18.

$$f_2 = \min\left(\sum_{i \in J} C_i\right) \quad 0.18$$

Die Minimierung der kumulierten Verspätungen ergibt sich nach 0.19 und 0.20.

$$f_2 = \min\left(\sum_{i \in J} G_i\right) \quad 0.19$$

$$G_i \geq (S_{ip(i)jk} + t_{jk} + d_{k,WO} - R_i) \cdot X_{ip(i)jk}, \quad \forall i \in J, \forall p(i) \in P_i, \forall k \in M_j, j = O_{ip(i)l(i)} \quad 0.20$$

A5 Belegungspläne zur Demonstration des Einflusses der Aktionsgenerierung auf den Suchfokus

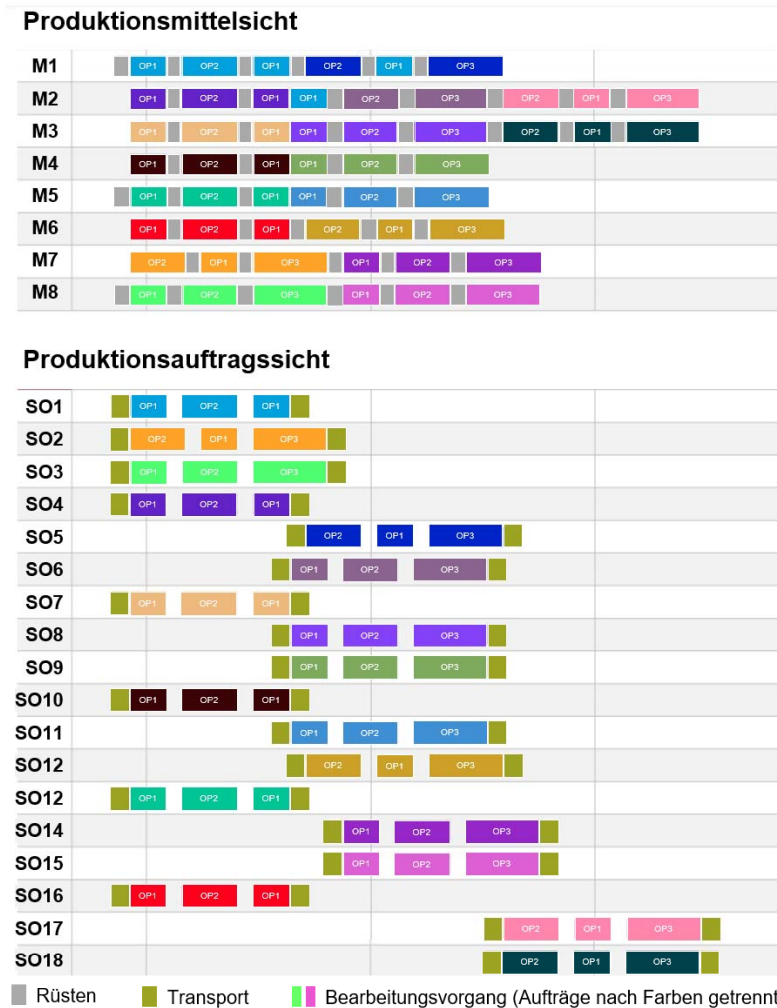
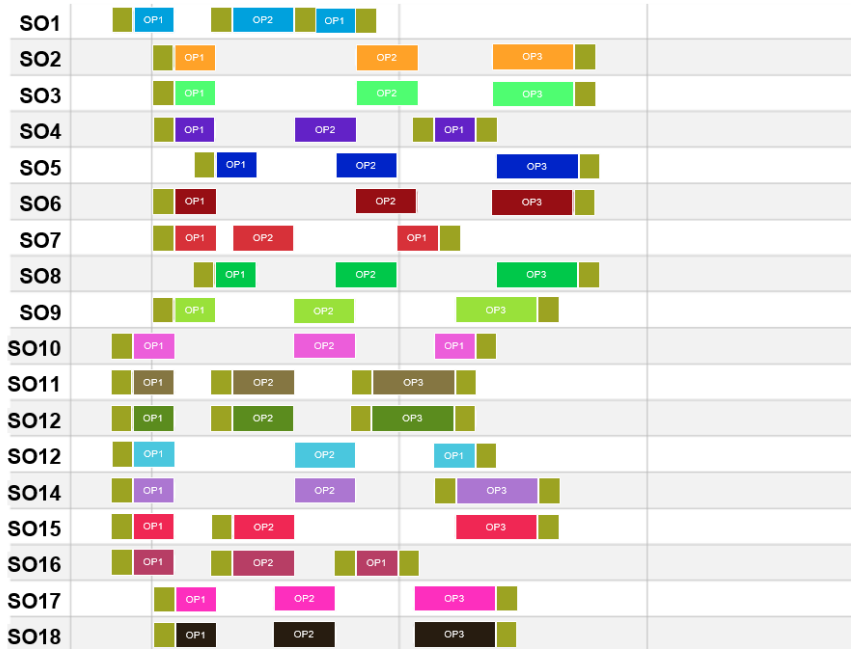


Abbildung A5.1: Belegungsplan mit optimaler kumulierter Durchlaufzeit, für eine mittlere Einschränkung des Aktionsraums und einem parallel betrachteten Produktionsauftrag, angelehnt an (A_Deining 2020).

Produktionsmittelsicht



Produktionsauftragssicht



■ Rüsten ■ Transport ■ ■ ■ Bearbeitungsvorgang (Aufträge nach Farben getrennt)

Abbildung A5.2: Belegungsplan mit sehr gutem Makespan, für eine sehr starke Einschränkung des Aktionsraums und 18 parallel betrachtete Produktionsaufträge, angelehnt an (A_Deininger 2020)

A6 Hypervolumen für das 12x5 Problem

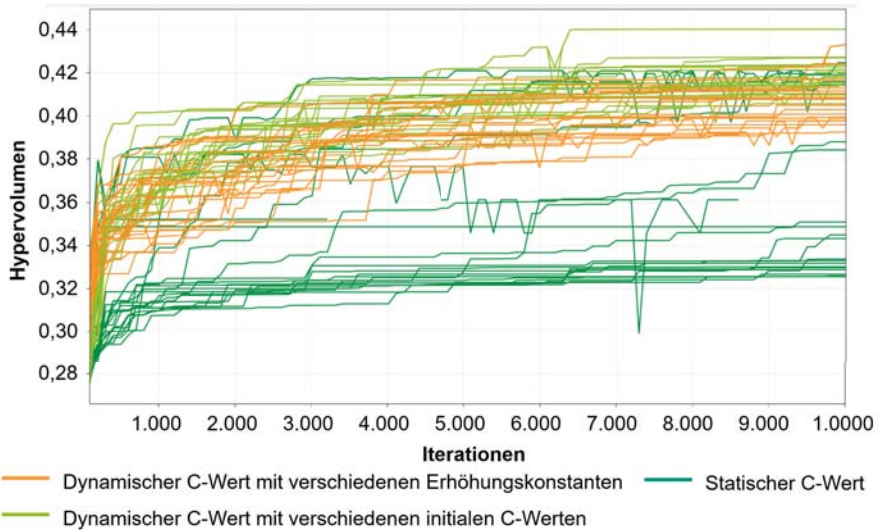


Abbildung A6.1: Maximales Hypervolumen für unterschiedliche C-Werte für das 12x5 Problem bei Verwendung der Filterkombinationen mit mittlerer Einschränkung und 10 parallel betrachteten Produktionsaufträgen, angelehnt an (A_Deiningner 2020)

A7 Optimale Belegungspläne für das exakte Referenzproblem

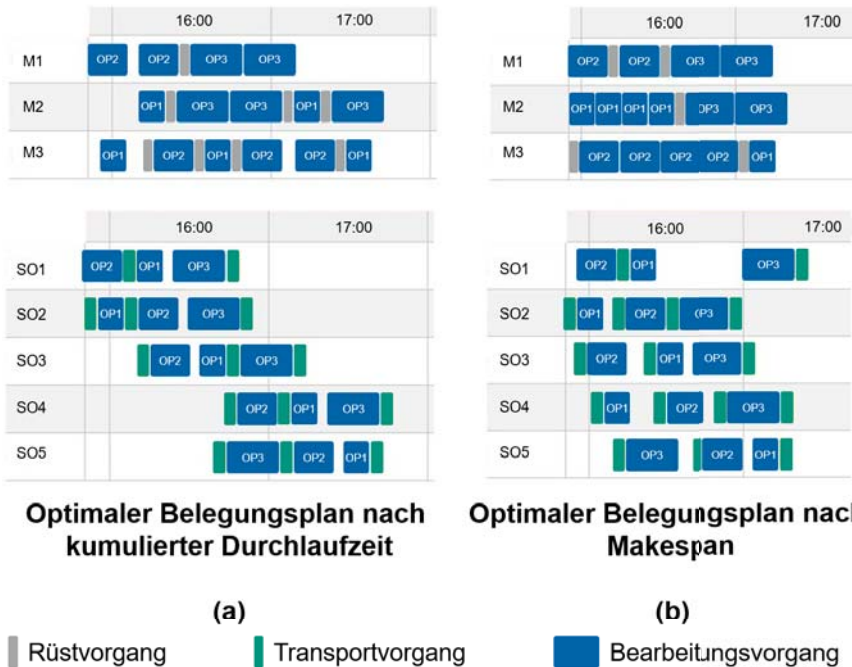


Abbildung A7.1: Grafik (a) zeigt den Belegungsplan aus Sicht der Produktionsmittel (oben) und aus Sicht der Produktionsaufträge (unten) für die optimale Lösung bezüglich der kumulierten Durchlaufzeit. Grafik (b) zeigt die analogen Abbildungen für den optimalen Makespan, angelehnt an (A_Deiningner 2020)

Forschungsberichte aus dem wbk
Institut für Produktionstechnik
Karlsruher Institut für Technologie (KIT)

Bisher erschienene Bände:

Band 0

Dr.-Ing. Wu Hong-qi

Adaptive Volumenstromregelung mit Hilfe von drehzahlgeregelten Elektroantrieben

Band 1

Dr.-Ing. Heinrich Weiß

**Fräsen mit Schneidkeramik - Verhalten des System
Werkzeugmaschine-Werkzeug-Werkstück und Prozessanalyse**

Band 2

Dr.-Ing. Hans-Jürgen Stierle

**Entwicklung und Untersuchung hydrostatischer Lager für die
Axialkolbenmaschine**

Band 3

Dr.-Ing. Herbert Hörner

Untersuchung des Geräuschverhaltens druckgeregelter Axialkolbenpumpen

Band 4

Dr.-Ing. Rolf-Dieter Brückbauer

**Digitale Drehzahlregelung unter der besonderen Berücksichtigung
von Quantisierungseffekten**

Band 5

Dr.-Ing. Gerhard Staiger

Graphisch interaktive NC-Programmierung von Drehteilen im Werkstattbereich

Band 6

Dr.-Ing. Karl Peters

**Ein Beitrag zur Berechnung und Kompensation von Positionierfehlern an
Industrierobotern**

Band 7

Dr.-Ing. Paul Stauss

Automatisierte Inbetriebnahme und Sicherung der Zuverlässigkeit und Verfügbarkeit numerisch gesteuerter Fertigungseinrichtungen

Band 8

Dr.-Ing. Günter Möckesch

Konzeption und Realisierung eines strategischen, integrierten Gesamtplanungs- und -bearbeitungssystems zur Optimierung der Drehteilorganisation für auftragsbezogene Drehereien

Band 9

Dr.-Ing. Thomas Oestreicher

Rechnergestützte Projektierung von Steuerungen

Band 10

Dr.-Ing. Thomas Selinger

Teilautomatisierte werkstattnahe NC-Programmerstellung im Umfeld einer integrierten Informationsverarbeitung

Band 11

Dr.-Ing. Thomas Buchholz

Prozessmodell Fräsen, Rechnerunterstützte Analyse, Optimierung und Überwachung

Band 12

Dr.-Ing. Bernhard Reichling

Lasergestützte Positions- und Bahnvermessung von Industrierobotern

Band 13

Dr.-Ing. Hans-Jürgen Lesser

Rechnergestützte Methoden zur Auswahl anforderungsgerechter Verbindungselemente

Band 14

Dr.-Ing. Hans-Jürgen Lauffer

Einsatz von Prozessmodellen zur rechnerunterstützten Auslegung von Räumwerkzeugen

Band 15

Dr.-Ing. Michael C. Wilhelm

Rechnergestützte Prüfplanung im Informationsverbund moderner Produktionssysteme

Band 16
Dr.-Ing. Martin Ochs

**Entwurf eines Programmsystems zur wissensbasierten Planung
und Konfigurierung**

Band 17
Dr.-Ing. Heinz-Joachim Schneider

**Erhöhung der Verfügbarkeit von hochautomatisierten
Produktionseinrichtungen mit Hilfe der Fertigungsleittechnik**

Band 18
Dr.-Ing. Hans-Reiner Ludwig

Beanspruchungsanalyse der Werkzeugschneiden beim Stirnplanfräsen

Band 19
Dr.-Ing. Rudolf Wieser

Methoden zur rechnergestützten Konfigurierung von Fertigungsanlagen

Band 20
Dr.-Ing. Edgar Schmitt

Werkstattsteuerung bei wechselnder Auftragsstruktur

Band 21
Dr.-Ing. Wilhelm Enderle

**Verfügbarkeitssteigerung automatisierter Montagesysteme
durch selbsttätige Behebung prozessbedingter Störungen**

Band 22
Dr.-Ing. Dieter Buchberger

Rechnergestützte Strukturplanung von Produktionssystemen

Band 23
Prof. Dr.-Ing. Jürgen Fleischer

**Rechnerunterstützte Technologieplanung für die flexibel
automatisierte Fertigung von Abkantteilen**

Band 24
Dr.-Ing. Lukas Loeffler

Adaptierbare und adaptive Benutzerschnittstellen

Band 25
Dr.-Ing. Thomas Friedmann

**Integration von Produktentwicklung und Montageplanung durch neue
rechnergestützte Verfahren**

Band 26

Dr.-Ing. Robert Zurrin

Variables Formhonen durch rechnergestützte Hornprozesssteuerung

Band 27

Dr.-Ing. Karl-Heinz Bergen

Langhub-Innenrundhonen von Grauguss und Stahl mit einem elektromechanischem Vorschubsystem

Band 28

Dr.-Ing. Andreas Liebisch

Einflüsse des Festwalzens auf die Eigenspannungsverteilung und die Dauerfestigkeit einsatzgehärteter Zahnräder

Band 29

Dr.-Ing. Rolf Ziegler

Auslegung und Optimierung schneller Servopumpen

Band 30

Dr.-Ing. Rainer Bartl

Datenmodellgestützte Wissensverarbeitung zur Diagnose und Informationsunterstützung in technischen Systemen

Band 31

Dr.-Ing. Ulrich Golz

Analyse, Modellbildung und Optimierung des Betriebsverhaltens von Kugelgewindetrieben

Band 32

Dr.-Ing. Stephan Timmermann

Automatisierung der Feinbearbeitung in der Fertigung von Hohlformwerkzeugen

Band 33

Dr.-Ing. Thomas Noe

Rechnergestützter Wissenserwerb zur Erstellung von Überwachungs- und Diagnoseexpertensystemen für hydraulische Anlagen

Band 34

Dr.-Ing. Ralf Lenschow

Rechnerintegrierte Erstellung und Verifikation von Steuerungsprogrammen als Komponente einer durchgängigen Planungsmethodik

Band 35

Dr.-Ing. Matthias Kallabis

Räumen gehärteter Werkstoffe mit kristallinen Hartstoffen

Band 36

Dr.-Ing. Heiner-Michael Honeck

Rückführung von Fertigungsdaten zur Unterstützung einer fertigungsgerechten Konstruktion

Band 37

Dr.-Ing. Manfred Rohr

Automatisierte Technologieplanung am Beispiel der Komplettbearbeitung auf Dreh-/Fräszellen

Band 38

Dr.-Ing. Martin Steuer

Entwicklung von Softwarewerkzeugen zur wissensbasierten Inbetriebnahme von komplexen Serienmaschinen

Band 39

Dr.-Ing. Siegfried Beichter

Rechnergestützte technische Problemlösung bei der Angebotserstellung von flexiblen Drehzellen

Band 40

Dr.-Ing. Thomas Steitz

Methodik zur marktorientierten Entwicklung von Werkzeugmaschinen mit Integration von funktionsbasierter Strukturierung und Kostenschätzung

Band 41

Dr.-Ing. Michael Richter

Wissensbasierte Projektierung elektrohydraulischer Regelungen

Band 42

Dr.-Ing. Roman Kuhn

Technologieplanungssystem Fräsen. Wissensbasierte Auswahl von Werkzeugen, Schneidkörpern und Schnittbedingungen für das Fertigungsverfahren Fräsen

Band 43

Dr.-Ing. Hubert Klein

Rechnerunterstützte Qualitätssicherung bei der Produktion von Bauteilen mit frei geformten Oberflächen

Band 44

Dr.-Ing. Christian Hoffmann

Konzeption und Realisierung eines fertigungsintegrierten Koordinatenmessgerätes

Band 45

Dr.-Ing. Volker Frey

Planung der Leittechnik für flexible Fertigungsanlagen

Band 46

Dr.-Ing. Achim Feller

Kalkulation in der Angebotsphase mit dem selbsttätig abgeleiteten Erfahrungswissen der Arbeitsplanung

Band 47

Dr.-Ing. Markus Klaiber

Produktivitätssteigerung durch rechnerunterstütztes Einfahren von NC-Programmen

Band 48

Dr.-Ing. Roland Minges

Verbesserung der Genauigkeit beim fünfachsigem Fräsen von Freiformflächen

Band 49

Dr.-Ing. Wolfgang Bernhart

Beitrag zur Bewertung von Montagevarianten: Rechnergestützte Hilfsmittel zur kostenorientierten, parallelen Entwicklung von Produkt und Montagesystem

Band 50

Dr.-Ing. Peter Ganghoff

Wissensbasierte Unterstützung der Planung technischer Systeme: Konzeption eines Planungswerkzeuges und exemplarische Anwendung im Bereich der Montagesystemplanung

Band 51

Dr.-Ing. Frank Maier

Rechnergestützte Prozessregelung beim flexiblen Gesenkbiegen durch Rückführung von Qualitätsinformationen

Band 52

Dr.-Ing. Frank Debus

Ansatz eines rechnerunterstützten Planungsmanagements für die Planung in verteilten Strukturen

Band 53

Dr.-Ing. Joachim Weinbrecht

Ein Verfahren zur zielorientierten Reaktion auf Planabweichungen in der Werkstattregelung

Band 54

Dr.-Ing. Gerd Herrmann

Reduzierung des Entwicklungsaufwandes für anwendungsspezifische Zellenrechnersoftware durch Rechnerunterstützung

Band 55

Dr.-Ing. Robert Wassmer

Verschleissentwicklung im tribologischen System Fräsen: Beiträge zur Methodik der Prozessmodellierung auf der Basis tribologischer Untersuchungen beim Fräsen

Band 56

Dr.-Ing. Peter Uebelhoer

Inprocess-Geometriemessung beim Honen

Band 57

Dr.-Ing. Hans-Joachim Schelberg

Objektorientierte Projektierung von SPS-Software

Band 58

Dr.-Ing. Klaus Boes

Integration der Qualitätsentwicklung in featurebasierte CAD/CAM-Prozessketten

Band 59

Dr.-Ing. Martin Schreiber

Wirtschaftliche Investitionsbewertung komplexer Produktionssysteme unter Berücksichtigung von Unsicherheit

Band 60

Dr.-Ing. Ralf Steuernagel

Offenes adaptives Engineering-Werkzeug zur automatisierten Erstellung von entscheidungsunterstützenden Informationssystemen

Band 62

Dr.-Ing. Uwe Schauer

Qualitätsorientierte Feinbearbeitung mit Industrierobotern: Regelungsansatz für die Freiformflächenfertigung des Werkzeug- und Formenbaus

Band 63

Dr.-Ing. Simone Loeper

Kennzahlengestütztes Beratungssystem zur Verbesserung der Logistikleistung in der Werkstattfertigung

Band 64

Dr.-Ing. Achim Raab

Räumen mit hartstoffbeschichteten HSS-Werkzeugen

Band 65,

Dr.-Ing. Jan Erik Burghardt

Unterstützung der NC-Verfahrenskette durch ein bearbeitungs-elementorientiertes, lernfähiges Technologieplanungssystem

Band 66

Dr.-Ing. Christian Tritsch

Flexible Demontage technischer Gebrauchsgüter: Ansatz zur Planung und (teil-)automatisierten Durchführung industrieller Demontageprozesse

Band 67

Dr.-Ing. Oliver Eitrich

Prozessorientiertes Kostenmodell für die entwicklungsbegleitende Vorkalkulation

Band 68

Dr.-Ing. Oliver Wilke

Optimierte Antriebskonzepte für Räummaschinen - Potentiale zur Leistungssteigerung

Band 69

Dr.-Ing. Thilo Sieth

Rechnergestützte Modellierungsmethodik zerspantechnologischer Prozesse

Band 70

Dr.-Ing. Jan Linnenbuerger

Entwicklung neuer Verfahren zur automatisierten Erfassung der geometrischen Abweichungen an Linearachsen und Drehschwenkköpfen

Band 71

Dr.-Ing. Mathias Klimmek

Fraktionierung technischer Produkte mittels eines frei beweglichen Wasserstrahlwerkzeuges

Band 72

Dr.-Ing. Marko Hartel

Kennzahlenbasiertes Bewertungssystem zur Beurteilung der Demontage- und Recyclingeignung von Produkten

Band 73

Dr.-Ing. Jörg Schaupp

Wechselwirkung zwischen der Maschinen- und Hauptspindeltriebsdynamik und dem Zerspanprozess beim Fräsen

Band 74

Dr.-Ing. Bernhard Neisius

Konzeption und Realisierung eines experimentellen Telemanipulators für die Laparoskopie

Band 75

Dr.-Ing. Wolfgang Walter

Erfolgsversprechende Muster für betriebliche Ideenfindungsprozesse. Ein Beitrag zur Steigerung der Innovationsfähigkeit

Band 76

Dr.-Ing. Julian Weber

Ein Ansatz zur Bewertung von Entwicklungsergebnissen in virtuellen Szenarien

Band 77

Dr.-Ing. Dipl. Wirtsch.-Ing. Markus Posur

Unterstützung der Auftragsdurchsetzung in der Fertigung durch Kommunikation über mobile Rechner

Band 78

Dr.-Ing. Frank Fleissner

Prozessorientierte Prüfplanung auf Basis von Bearbeitungsobjekten für die Kleinserienfertigung am Beispiel der Bohr- und Fräsbearbeitung

Band 79

Dr.-Ing. Anton Haberkern

Leistungsfähigere Kugelgewindetriebe durch Beschichtung

Band 80

Dr.-Ing. Dominik Matt

Objektorientierte Prozess- und Strukturinnovation (OPUS)

Band 81

Dr.-Ing. Jürgen Andres

Robotersysteme für den Wohnungsbau: Beitrag zur Automatisierung des Mauerwerkbaus und der Elektroinstallation auf Baustellen

Band 82

Dr.-Ing. Dipl.Wirtschaftsing. Simone Riedmiller

Der Prozesskalender - Eine Methodik zur marktorientierten Entwicklung von Prozessen

Band 83

Dr.-Ing. Dietmar Tilch

Analyse der Geometrieparameter von Präzisionsgewinden auf der Basis einer Least-Squares-Estimation

Band 84

Dr.-Ing. Dipl.-Kfm. Oliver Stiefbold

Konzeption eines reaktionsschnellen Planungssystems für Logistikketten auf Basis von Software-Agenten

Band 85

Dr.-Ing. Ulrich Walter

Einfluss von Kühlschmierstoff auf den Zerspansprozess beim Fräsen: Beitrag zum Prozessverständnis auf Basis von zerspantechnischen Untersuchungen

Band 86

Dr.-Ing. Bernd Werner

Konzeption von teilautonomer Gruppenarbeit unter Berücksichtigung kultureller Einflüsse

Band 87

Dr.-Ing. Ulf Osmers

Projektieren Speicherprogrammierbarer Steuerungen mit Virtual Reality

Band 88

Dr.-Ing. Oliver Doerfel

Optimierung der Zerspantechnik beim Fertigungsverfahren Wälzstossen: Analyse des Potentials zur Trockenbearbeitung

Band 89

Dr.-Ing. Peter Baumgartner

Stufenmethode zur Schnittstellengestaltung in der internationalen Produktion

Band 90
Dr.-Ing. Dirk Vossman

Wissensmanagement in der Produktentwicklung durch Qualitätsmethodenverbund und Qualitätsmethodenintegration

Band 91
Dr.-Ing. Martin Plass

Beitrag zur Optimierung des Honprozesses durch den Aufbau einer Honprozessregelung

Band 92
Dr.-Ing. Titus Konold

Optimierung der Fünffachsfräsbearbeitung durch eine kennzahlenunterstützte CAM-Umgebung

Band 93
Dr.-Ing. Jürgen Brath

Unterstützung der Produktionsplanung in der Halbleiterfertigung durch risikoberücksichtigende Betriebskennlinien

Band 94
Dr.-Ing. Dirk Geisinger

Ein Konzept zur marktorientierten Produktentwicklung

Band 95
Dr.-Ing. Marco Lanza

Entwurf der Systemunterstützung des verteilten Engineering mit Axiomatic Design

Band 96
Dr.-Ing. Volker Hüntrup

Untersuchungen zur Mikrostrukturierbarkeit von Stählen durch das Fertigungsverfahren Fräsen

Band 97
Dr.-Ing. Frank Reinboth

Interne Stützung zur Genauigkeitsverbesserung in der Inertialmesstechnik: Beitrag zur Senkung der Anforderungen an Inertialsensoren

Band 98
Dr.-Ing. Lutz Trender

Entwicklungintegrierte Kalkulation von Produktlebenszykluskosten auf Basis der ressourcenorientierten Prozesskostenrechnung

Band 99

Dr.-Ing. Cornelia Kafka

Konzeption und Umsetzung eines Leitfadens zum industriellen Einsatz von Data-Mining

Band 100

Dr.-Ing. Gebhard Selinger

Rechnerunterstützung der informellen Kommunikation in verteilten Unternehmensstrukturen

Band 101

Dr.-Ing. Thomas Windmüller

Verbesserung bestehender Geschäftsprozesse durch eine mitarbeiterorientierte Informationsversorgung

Band 102

Dr.-Ing. Knud Lembke

Theoretische und experimentelle Untersuchung eines bistabilen elektrohydraulischen Linearantriebs

Band 103

Dr.-Ing. Ulrich Thies

Methode zur Unterstützung der variantengerechten Konstruktion von industriell eingesetzten Kleingeräten

Band 104

Dr.-Ing. Andreas Schmälzle

Bewertungssystem für die Generalüberholung von Montageanlagen –Ein Beitrag zur wirtschaftlichen Gestaltung geschlossener Facility- Management- Systeme im Anlagenbau

Band 105

Dr.-Ing. Thorsten Frank

Vergleichende Untersuchungen schneller elektromechanischer Vorschubachsen mit Kugelgewindetrieb

Band 106

Dr.-Ing. Achim Agostini

Reihenfolgeplanung unter Berücksichtigung von Interaktionen: Beitrag zur ganzheitlichen Strukturierung und Verarbeitung von Interaktionen von Bearbeitungsobjekten

Band 107

Dr.-Ing. Thomas Barrho

Flexible, zeitfenstergesteuerte Auftragseinplanung in segmentierten Fertigungsstrukturen

Band 108

Dr.-Ing. Michael Scharer

Quality Gate-Ansatz mit integriertem Risikomanagement

Band 109

Dr.-Ing. Ulrich Suchy

**Entwicklung und Untersuchung eines neuartigen Mischkopfes für das Wasser
Abrasive Strahl schneiden**

Band 110

Dr.-Ing. Sellal Mussa

Aktive Korrektur von Verlagerungsfehlern in Werkzeugmaschinen

Band 111

Dr.-Ing. Andreas Hühsam

Modellbildung und experimentelle Untersuchung des Wälzschälprozesses

Band 112

Dr.-Ing. Axel Plutowsky

**Charakterisierung eines optischen Messsystems und den Bedingungen des
Arbeitsraums einer Werkzeugmaschine**

Band 113

Dr.-Ing. Robert Landwehr

**Konsequent dezentralisierte Steuerung mit Industrial Ethernet und offenen
Applikationsprotokollen**

Band 114

Dr.-Ing. Christoph Dill

Turbulenzreaktionsprozesse

Band 115

Dr.-Ing. Michael Baumeister

Fabrikplanung im turbulenten Umfeld

Band 116

Dr.-Ing. Christoph Gönninger

**Konzept zur Verbesserung der Elektromagnetischen Verträglichkeit (EMV) in
Produktionssystemen durch intelligente Sensor/Aktor-Anbindung**

Band 117

Dr.-Ing. Lutz Demuß

Ein Reifemodell für die Bewertung und Entwicklung von Dienstleistungsorganisationen: Das Service Management Maturity Modell (SMMM)

Band 118

Dr.-Ing. Jörg Söhner

Beitrag zur Simulation zerspanungstechnologischer Vorgänge mit Hilfe der Finite-Element-Methode

Band 119

Dr.-Ing. Judith Elsner

Informationsmanagement für mehrstufige Mikro-Fertigungsprozesse

Band 120

Dr.-Ing. Lijing Xie

Estimation Of Two-dimension Tool Wear Based On Finite Element Method

Band 121

Dr.-Ing. Ansgar Blessing

Geometrischer Entwurf mikromechatronischer Systeme

Band 122

Dr.-Ing. Rainer Ebner

Steigerung der Effizienz mehrachsiger Fräsprozesse durch neue Planungsmethoden mit hoher Benutzerunterstützung

Band 123

Dr.-Ing. Silja Klinkel

Multikriterielle Feinplanung in teilautonomen Produktionsbereichen – Ein Beitrag zur produkt- und prozessorientierten Planung und Steuerung

Band 124

Dr.-Ing. Wolfgang Neithardt

Methodik zur Simulation und Optimierung von Werkzeugmaschinen in der Konzept- und Entwurfsphase auf Basis der Mehrkörpersimulation

Band 125

Dr.-Ing. Andreas Mehr

Hartfeinbearbeitung von Verzahnungen mit kristallinen diamantbeschichteten Werkzeugen beim Fertigungsverfahren Wälzstoßen

Band 126

Dr.-Ing. Martin Gutmann

Entwicklung einer methodischen Vorgehensweise zur Diagnose von hydraulischen Produktionsmaschinen

Band 127

Dr.-Ing. Gisela Lanza

Simulative Anlaufunterstützung auf Basis der Qualitätsfähigkeiten von Produktionsprozessen

Band 128

Dr.-Ing. Ulf Dambacher

Kugelgewindetrieb mit hohem Druckwinkel

Band 129

Dr.-Ing. Carsten Buchholz

Systematische Konzeption und Aufbau einer automatisierten Produktionszelle für pulverspritzgegossene Mikroteile

Band 130

Dr.-Ing. Heiner Lang

Trocken-Räumen mit hohen Schnittgeschwindigkeiten

Band 131

Dr.-Ing. Daniel Nesges

Prognose operationeller Verfügbarkeiten von Werkzeugmaschinen unter Berücksichtigung von Serviceleistungen

Im Shaker Verlag erschienene Bände:

Band 132

Dr.-Ing. Andreas Bechle

Beitrag zur prozesssicheren Bearbeitung beim Hochleistungsfertigungsverfahren Wälzschälen

Band 133

Dr.-Ing. Markus Herm

Konfiguration globaler Wertschöpfungsnetzwerke auf Basis von Business Capabilities

Band 134

Dr.-Ing. Hanno Tritschler

Werkzeug- und Zerspanprozessoptimierung beim Hartfräsen von Mikrostrukturen in Stahl

Band 135

Dr.-Ing. Christian Munzinger

Adaptronische Strebe zur Steifigkeitssteigerung von Werkzeugmaschinen

Band 136

Dr.-Ing. Andreas Stepping

Fabrikplanung im Umfeld von Wertschöpfungsnetzwerken und ganzheitlichen Produktionssystemen

Band 137

Dr.-Ing. Martin Dyck

Beitrag zur Analyse thermische bedingter Werkstückdeformationen in Trockenbearbeitungsprozessen

Band 138

Dr.-Ing. Siegfried Schmalzried

Dreidimensionales optisches Messsystem für eine effizientere geometrische Maschinenbeurteilung

Band 139

Dr.-Ing. Marc Wawerla

Risikomanagement von Garantieleistungen

Band 140

Dr.-Ing. Ivesa Buchholz

Strategien zur Qualitätssicherung mikromechanischer Bauteile mittels multisensorieller Koordinatenmesstechnik

Band 141

Dr.-Ing. Jan Kotschenreuther

Empirische Erweiterung von Modellen der Makrozerspanung auf den Bereich der Mikrobearbeitung

Band 142

Dr.-Ing. Andreas Knödel

Adaptronische hydrostatische Drucktascheneinheit

Band 143

Dr.-Ing. Gregor Stengel

Fliegendes Abtrennen räumlich gekrümmter Strangpressprofile mittels Industrierobotern

Band 144

Dr.-Ing. Udo Weismann

Lebenszyklusorientiertes interorganisationelles Anlagencontrolling

Band 145

Dr.-Ing. Rüdiger Pabst

Mathematische Modellierung der Wärmestromdichte zur Simulation des thermischen Bauteilverhaltens bei der Trockenbearbeitung

Band 146

Dr.-Ing. Jan Wieser

Intelligente Instandhaltung zur Verfügbarkeitssteigerung von Werkzeugmaschinen

Band 147

Dr.-Ing. Sebastian Haupt

Effiziente und kostenoptimale Herstellung von Mikrostrukturen durch eine Verfahrenskombination von Bahnerosion und Laserablation

Band 148

Dr.-Ing. Matthias Schlipf

Statistische Prozessregelung von Fertigungs- und Messprozess zur Erreichung einer variabilitätsarmen Produktion mikromechanischer Bauteile

Band 149

Dr.-Ing. Jan Philipp Schmidt-Ewig

Methodische Erarbeitung und Umsetzung eines neuartigen Maschinenkonzeptes zur produktflexiblen Bearbeitung räumlich gekrümmter Strangpressprofile

Band 150

Dr.-Ing. Thomas Ender

Prognose von Personalbedarfen im Produktionsanlauf unter Berücksichtigung dynamischer Planungsgrößen

Band 151

Dr.-Ing. Kathrin Peter

**Bewertung und Optimierung der Effektivität von Lean Methoden
in der Kleinserienproduktion**

Band 152

Dr.-Ing. Matthias Schopp

Sensorbasierte Zustandsdiagnose und -prognose von Kugelgewindetrieben

Band 153

Dr.-Ing. Martin Kipfmüller

Aufwandsoptimierte Simulation von Werkzeugmaschinen

Band 154

Dr.-Ing. Carsten Schmidt

**Development of a database to consider multi wear mechanisms
within chip forming simulation**

Band 155

Dr.-Ing. Stephan Niggeschmidt

**Ausfallgerechte Ersatzteilbereitstellung im Maschinen- und Anlagenbau
mittels lastabhängiger Lebensdauerprognose**

Band 156

Dr.-Ing. Jochen Conrad Peters

**Bewertung des Einflusses von Formabweichungen in der
Mikro-Koordinatenmesstechnik**

Band 157

Dr.-Ing. Jörg Ude

**Entscheidungsunterstützung für die Konfiguration
globaler Wertschöpfungsnetzwerke**

Band 158

Dr.-Ing. Stefan Weiler

Strategien zur wirtschaftlichen Gestaltung der globalen Beschaffung

Band 159

Dr.-Ing. Jan Rühl

Monetäre Flexibilitäts- und Risikobewertung

Band 160

Dr.-Ing. Daniel Ruch

Positions- und Konturerfassung räumlich gekrümmter Profile auf Basis bauteilimmanenter Markierungen

Band 161

Dr.-Ing. Manuel Tröndle

Flexible Zuführung von Mikrobauteilen mit piezoelektrischen Schwingförderern

Band 162

Dr.-Ing. Benjamin Viering

Mikroverzahnungsnormal

Band 163

Dr.-Ing. Chris Becke

Prozesskrafttrichtungsangepasste Frässtrategien zur schädigungsarmen Bohrungsbearbeitung an faserverstärkten Kunststoffen

Band 164

Dr.-Ing. Patrick Werner

Dynamische Optimierung und Unsicherheitsbewertung der lastabhängigen präventiven Instandhaltung von Maschinenkomponenten

Band 165

Dr.-Ing. Martin Weis

Kompensation systematischer Fehler bei Werkzeugmaschinen durch self-sensing Aktoren

Band 166

Dr.-Ing. Markus Schneider

Kompensation von Konturabweichungen bei gerundeten Strangpressprofilen durch robotergestützte Führungswerkzeuge

Band 167

Dr.-Ing. Ester M. R. Ruprecht

Prozesskette zur Herstellung schichtbasierter Systeme mit integrierten Kavitäten

Band 168

Dr.-Ing. Alexander Broos

Simulationsgestützte Ermittlung der Komponentenbelastung für die Lebensdauerprognose an Werkzeugmaschinen

Band 169

Dr.-Ing. Frederik Zanger

Segmentspanbildung, Werkzeugverschleiß, Randschichtzustand und Bauteileigenschaften: Numerische Analysen zur Optimierung des Zerspanungsprozesses am Beispiel von Ti-6Al-4V

Band 170

Dr.-Ing. Benjamin Behmann

Servicefähigkeit

Band 171

Dr.-Ing. Annabel Gabriele Jondral

Simulationsgestützte Optimierung und Wirtschaftlichkeitsbewertung des Lean-Methodeneinsatzes

Band 172

Dr.-Ing. Christoph Ruhs

Automatisierte Prozessabfolge zur qualitätssicheren Herstellung von Kavitäten mittels Mikrobahnerosion

Band 173

Dr.-Ing. Steven Peters

Markoffsche Entscheidungsprozesse zur Kapazitäts- und Investitionsplanung von Produktionssystemen

Band 174

Dr.-Ing. Christoph Kühlewein

Untersuchung und Optimierung des Wälzschälverfahrens mit Hilfe von 3D-FEM-Simulation – 3D-FEM Kinematik- und Spanbildungssimulation

Band 175

Dr.-Ing. Adam-Mwanga Dieckmann

Auslegung und Fertigungsprozessgestaltung sintergefügter Verbindungen für μ MIM-Bauteile

Band 176

Dr.-Ing. Heiko Hennrich

Aufbau eines kombinierten belastungs- und zustandsorientierten Diagnose- und Prognosesystems für Kugelgewindetriebe

Band 177

Dr.-Ing. Stefan Herder

Piezoelektrischer Self-Sensing-Aktor zur Vorspannungsregelung in adaptiven Kugelgewindetrieben

Band 178

Dr.-Ing. Alexander Ochs

Ultraschall-Strömungsgreifer für die Handhabung textiler Halbzeuge bei der automatisierten Fertigung von RTM-Bauteilen

Band 179

Dr.-Ing. Jürgen Michna

Numerische und experimentelle Untersuchung zerspanungsbedingter Gefügeumwandlungen und Modellierung des thermo-mechanischen Lastkollektivs beim Bohren von 42CrMo4

Band 180

Dr.-Ing. Jörg Elser

Vorrichtungsfreie räumliche Anordnung von Fügepartnern auf Basis von Bauteilmarkierungen

Band 181

Dr.-Ing. Katharina Klimscha

Einfluss des Fügespalts auf die erreichbare Verbindungsqualität beim Sinterfügen

Band 182

Dr.-Ing. Patricia Weber

Steigerung der Prozesswiederholbarkeit mittels Analyse akustischer Emissionen bei der Mikrolaserablation mit UV-Pikosekundenlasern

Band 183

Dr.-Ing. Jochen Schädel

Automatisiertes Fügen von Tragprofilen mittels Faserwickeln

Band 184

Dr.-Ing. Martin Krauße

Aufwandsoptimierte Simulation von Produktionsanlagen durch Vergrößerung der Geltungsbereiche von Teilmodellen

Band 185

Dr.-Ing. Raphael Moser

Strategische Planung globaler Produktionsnetzwerke

Bestimmung von Wandlungsbedarf und Wandlungszeitpunkt mittels multikriterieller Optimierung

Band 186

Dr.-Ing. Martin Otter

Methode zur Kompensation fertigungsbedingter Gestaltabweichungen für die Montage von Aluminium Space-Frame-Strukturen

Band 187

Dr.-Ing. Urs Leberle

Produktive und flexible Gleitförderung kleiner Bauteile auf phasenflexiblen Schwingförderern mit piezoelektrischen 2D-Antriebsselementen

Band 188

Dr.-Ing. Johannes Book

Modellierung und Bewertung von Qualitätsmanagementstrategien in globalen Wertschöpfungsnetzwerken

Band 189

Dr.-Ing. Florian Ambrosy

Optimierung von Zerspanungsprozessen zur prozesssicheren Fertigung nanokristalliner Randschichten am Beispiel von 42CrMo4

Band 190

Dr.-Ing. Adrian Kölmel

Integrierte Messtechnik für Prozessketten unreifer Technologien am Beispiel der Batterieproduktion für Elektrofahrzeuge

Band 191

Dr.-Ing. Henning Wagner

Featurebasierte Technologieplanung zum Preforming von textilen Halbzeugen

Band 192

Dr.-Ing. Johannes Gebhardt

Strukturoptimierung von in FVK eingebetteten metallischen Lasteinleitungselementen

Band 193

Dr.-Ing. Jörg Bauer

Hochintegriertes hydraulisches Vorschubsystem für die Bearbeitung kleiner Werkstücke mit hohen Fertigungsanforderungen

Band 194

Dr.-Ing. Nicole Stricker

Robustheit verketteter Produktionssysteme

Robustheitsevaluation und Selektion des Kennzahlensystems der Robustheit

Band 195

Dr.-Ing. Anna Sauer

Konfiguration von Montagelinien unreifer Produkttechnologien am Beispiel der Batteriemontage für Elektrofahrzeuge

Band 196

Dr.-Ing. Florian Sell-Le Blanc

Prozessmodell für das Linearwickeln unrunder Zahnspulen

Ein Beitrag zur orthozyklischen Spulenwickeltechnik

Band 197

Dr.-Ing. Frederic Förster

Geregeltes Handhabungssystem zum zuverlässigen und energieeffizienten Handling textiler Kohlenstofffaserzuschnitte

Band 198

Dr.-Ing. Nikolay Boev

Numerische Beschreibung von Wechselwirkungen zwischen Zerspanprozess und Maschine am Beispiel Räumen

Band 199

Dr.-Ing. Sebastian Greinacher

Simulationsgestützte Mehrzieloptimierung schlanker und ressourceneffizienter Produktionssysteme

Band 200

Dr.-Ing. Benjamin Häfner

Lebensdauerprognose in Abhängigkeit der Fertigungsabweichungen bei Mikroverzahnungen

Band 201

Dr.-Ing. Stefan Klotz

Dynamische Parameteranpassung bei der Bohrungsherstellung in faserverstärkten Kunststoffen unter zusätzlicher Berücksichtigung der Einspannsituation

Band 202

Dr.-Ing. Johannes Stoll

Bewertung konkurrierender Fertigungsfolgen mittels Kostensimulation und stochastischer Mehrzieloptimierung

Anwendung am Beispiel der Blechpaketfertigung für automobiler Elektromotoren

Band 203

Dr.-Ing. Simon-Frederik Koch

Fügen von Metall-Faserverbund-Hybridwellen im Schleuderverfahren
ein Beitrag zur fertigungsgerechten intrinsischen Hybridisierung

Band 204

Dr.-Ing. Julius Ficht

Numerische Untersuchung der Eigenspannungsentwicklung für sequenzielle Zerspanungsprozesse

Band 205

Dr.-Ing. Manuel Baumeister

Automatisierte Fertigung von Einzelblattstapeln in der Lithium-Ionen-Zellproduktion

Band 206

Dr.-Ing. Daniel Bertsch

Optimierung der Werkzeug- und Prozessauslegung für das Wälzschälen von Innenverzahnungen

Band 207

Dr.-Ing. Kyle James Kippenbrock

Deconvolution of Industrial Measurement and Manufacturing Processes for Improved Process Capability Assessments

Band 208

Dr.-Ing. Farboud Bejnoud

Experimentelle Prozesskettenbetrachtung für Räumbauteile am Beispiel einer einsatzgehärteten PKW-Schiebemuffe

Band 209

Dr.-Ing. Steffen Dosch

Herstellungsübergreifende Informationsübertragung zur effizienten Produktion von Werkzeugmaschinen am Beispiel von Kugelgewindetrieben

Band 210

Dr.-Ing. Emanuel Moser

Migrationsplanung globaler Produktionsnetzwerke

Bestimmung robuster Migrationspfade und risiko-effizienter Wandlungsbefähiger

Band 211

Dr.-Ing. Jan Hochdörffer

Integrierte Produktallokationsstrategie und Konfigurationssequenz in globalen Produktionsnetzwerken

Band 212

Dr.-Ing. Tobias Arndt

Bewertung und Steigerung der Prozessqualität in globalen Produktionsnetzwerken

Band 213

Dr.-Ing. Manuel Peter

Unwuchtminimale Montage von Permanentmagnetrotoren durch modellbasierte Online-Optimierung

Band 214

Dr.-Ing. Robin Kopf

Kostenorientierte Planung von Fertigungsfolgen additiver Technologien

Band 215

Dr.-Ing. Harald Meier

**Einfluss des Räumens auf den Bauteilzustand in der Prozesskette
Weichbearbeitung – Wärmebehandlung – Hartbearbeitung**

Band 216

Dr.-Ing. Daniel Brabandt

**Qualitätssicherung von textilen Kohlenstofffaser-Preforms mittels
optischer Messtechnik**

Band 217

Dr.-Ing. Alexandra Schabunow

**Einstellung von Aufnahmeparametern mittels projektionsbasierter Qualitäts-
kenngrößen in der industriellen Röntgen-Computertomographie**

Band 218

Dr.-Ing. Jens Bürgin

Robuste Auftragsplanung in Produktionsnetzwerken

Mittelfristige Planung der variantenreichen Serienproduktion unter Unsicherheit
der Kundenauftragskonfigurationen

Band 219

Dr.-Ing. Michael Gerstenmeyer

**Entwicklung und Analyse eines mechanischen Oberflächenbehandlungs-
verfahrens unter Verwendung des Zerspanungswerkzeuges**

Band 220

Dr.-Ing. Jacques Burtscher

**Erhöhung der Bearbeitungsstabilität von Werkzeugmaschinen durch
semi-passive masseneinstellbare Dämpfungssysteme**

Band 221

Dr.-Ing. Dietrich Berger

**Qualitätssicherung von textilen Kohlenstofffaser-Preforms mittels prozess-
integrierter Wirbelstromsensor-Arrays**

Band 222

Dr.-Ing. Fabian Johannes Ballier

Systematic gripper arrangement for a handling device in lightweight production processes

Band 223

Dr.-Ing. Marielouise Schäferling, geb. Zaiß

Development of a Data Fusion-Based Multi-Sensor System for Hybrid Sheet Molding Compound

Band 224

Dr.-Ing. Quirin Spiller

Additive Herstellung von Metallbauteilen mit dem ARBURG Kunststoff-Freiformen

Band 225

Dr.-Ing. Andreas Spohrer

Steigerung der Ressourceneffizienz und Verfügbarkeit von Kugelgewindetrieben durch adaptive Schmierung

Band 226

Dr.-Ing. Johannes Fisel

Veränderungsfähigkeit getakteter Fließmontagesysteme

Planung der Fließbandabstimmung am Beispiel der Automobilmontage

Band 227

Dr.-Ing. Patrick Bollig

Numerische Entwicklung von Strategien zur Kompensation thermisch bedingter Verzüge beim Bohren von 42CrMo4

Band 228

Dr.-Ing. Ramona Pfeiffer, geb. Singer

Untersuchung der prozessbestimmenden Größen für die anforderungsgerechte Gestaltung von Pouchzellen-Verpackungen

Band 229

Dr.-Ing. Florian Baumann

Additive Fertigung von endlosfaserverstärkten Kunststoffen mit dem ARBURG Kunststoff-Freiform Verfahren

Band 230
Dr.-Ing. Tom Stähr

Methodik zur Planung und Konfigurationsauswahl skalierbarer Montagesysteme – Ein Beitrag zur skalierbaren Automatisierung

Band 231
Dr.-Ing. Jan Schwennen

Einbringung und Gestaltung von Lasteinleitungsstrukturen für im RTM-Verfahren hergestellte FVK-Sandwichbauteile

Band 232
Dr.-Ing. Sven Coutandin

Prozessstrategien für das automatisierte Preforming von bebinderten textilen Halbzeugen mit einem segmentierten Werkzeugsystem

Band 233
Dr.-Ing. Christoph Liebrecht

Entscheidungsunterstützung für den Industrie 4.0-Methodeneinsatz
Strukturierung, Bewertung und Ableitung von Implementierungsreihenfolgen

Band 234
Dr.-Ing. Stefan Treber

Transparenzsteigerung in Produktionsnetzwerken
Verbesserung des Störungsmanagements durch verstärkten Informationsaustausch

Band 235
Dr.-Ing. Marius Dackweiler

Modellierung des Fügewickelprozesses zur Herstellung von leichten Fachwerkstrukturen

Band 236
Dr.-Ing. Fabio Echsler Minguillon

Prädiktiv-reaktives Scheduling zur Steigerung der Robustheit in der Matrix-Produktion

Band 237
Dr.-Ing. Sebastian Haag

Entwicklung eines Verfahrensablaufes zur Herstellung von Batteriezellstapeln mit großformatigem, rechteckigem Stapelformat und kontinuierlichen Materialbahnen

Band 238

Dr.-Ing. Raphael Wagner

Strategien zur funktionsorientierten Qualitätsregelung in der Serienproduktion

Band 239

Dr.-Ing. Christopher Ehrmann

Ausfallfrüherkennung von Ritzel-Zahnstangen- Trieben mittels Acoustic Emission

Band 240

Dr.-Ing. Janna Hofmann

Prozessmodellierung des Fünf-Achs-Nadelwickelns zur Implementierung einer trajektoriebasierten Drahtzugkraftregelung

Band 241

Dr.-Ing. Andreas Kuhnle

Adaptive Order Dispatching based on Reinforcement Learning Application in a Complex Job Shop in the Semiconductor Industry

Band 242

Dr.-Ing. Andreas Greiber

Fertigung optimierter technischer Oberflächen durch eine Verfahrenskombination aus Fliehkraft-Tauchgleitschleifen und Laserablation Prozesseinflüsse und Prozessauslegung

Band 243

Dr.-Ing. Jan Niclas Eschner

Entwicklung einer akustischen Prozessüberwachung zur Porenbestimmung im Laserstrahlschmelzen

Band 244

Dr.-Ing. Sven Roth

Schädigungsfreie Anbindung von hybriden FVK/Metall-Bauteilen an metallische Tragstrukturen durch Widerstandspunktschweißen

Band 245

Dr.-Ing. Sina Kathrin Peukert

Robustheitssteigerung in Produktionsnetzwerken mithilfe eines integrierten Störungsmanagements

Band 246

Dr.-Ing. Alexander Jacob

Hochiterative Technologieplanung

Rekursive Optimierung produkt- und fertigungsbezogener Freiheitsgrade am Beispiel der hybrid-additiven Fertigung

Band 247

Dr.-Ing. Patrick Moll

Ressourceneffiziente Herstellung von Langfaser-Preforms im Faserblasverfahren

Band 248

Dr.-Ing. Eric Thore Segebade

Erhöhung der Verschleißbeständigkeit von Bauteilen aus Ti-6Al-4V mittels simulationsgestützter Zerspanung und mechanischer Mikrotextrurierung

Band 249

Dr.-Ing. Shun Yang

Regionalized implementation strategy of smart automation within assembly systems in China

Band 250

Dr.-Ing. Constantin Carl Hofmann

Vorausschauende und reaktive Mehrzieloptimierung für die Produktionssteuerung einer Matrixproduktion