



Universidade do Porto
Faculdade de Engenharia
FEUP

Faculdade de Engenharia da Universidade do Porto
Doutoramento em Engenharia Industrial e Gestão



Uma nova abordagem para a programação integrada de veículos e tripulações

Bruno de Athayde Prata

Orientador: Teresa Galvão Dias

Co-orientador: Jorge Pinho de Sousa

Porto

Abril de 2011

Uma nova abordagem para a programação integrada de veículos e tripulações

Bruno de Athayde Prata

Tese submetida ao corpo docente do Programa Doutoral em Engenharia Industrial e Gestão da Faculdade de Engenharia da Universidade do Porto como requisito parcial para a obtenção do grau de Doutor em Engenharia Industrial e Gestão.

Porto
Abril 2011

AGRADECIMENTOS

Primeiramente, agradeço a Deus, pela saúde e capacidade de discernimento.

Agradeço ao Governo da República Federativa do Brasil por ter custeado meu ensino médio/técnico em Edificações, minha graduação como Engenheiro Civil e meu mestrado em Logística e Pesquisa Operacional. Com a conclusão de meu Doutorado, me prontifico a contribuir incisivamente para o desenvolvimento de meu país.

Agradeço à Universidade do Porto por todas as condições que propiciaram o desenvolvimento de minha tese de Doutorado.

Agradeço profundamente ao Professor João Falcão e Cunha por todo o inestimável apoio dado desde o meu ingresso na FEUP até a conclusão de minha tese.

Agradeço à Professora Maria Teresa Galvão Dias por toda a disponibilidade e carinho com os quais sempre me atendeu. Sua tenacidade, raciocínio rápido e, sobretudo, amizade, foram imprescindíveis para a conclusão desta tese. “Profa.”, meu sincero muito obrigado!

Agradeço ao Professor Jorge Pinho de Sousa pelo aprendizado que me propiciou e pela imensa consideração e estima que sempre teve para com a minha pessoa. Sua visão crítica e sistêmica foi fundamental para a elaboração deste projecto de investigação. Muito obrigado!

Agradeço ao Professor Alcibíades Guedes pela formidável convivência.

Agradeço aos Professores José Fernando Oliveira e José António Sarsfield Pereira Cabral pelo fundamental apoio concedido.

Agradeço aos Professores António Brito, Américo Azevedo, José Soeiro, José Faria e Filomena Jordão pelas disciplinas lecionadas no DGEI, que tanto contribuíram para minha formação académica.

Agradeço à amiga Vera Miguéis, colega de Doutorado, pelo fundamental apoio no período de conclusão da tese.

Agradeço à Soledade, Isabel e Mónica pelo instimável apoio às minhas actividades.

Agradeço à empresa Optimização e Planeamento de Transportes S.A. (OPT) pela disponibilização de dados reais para a presente investigação.

Agradeço aos Professores Giovanni Cordeiro Barroso e Ernesto Ferreira Nobre Júnior por todo apoio dado para que eu iniciasse e concluísse o meu Doutorado, bem como pela incondicional amizade.

Agradeço ao Professor Mário Ângelo de Azevedo Filho pelos dados concedidos e pela amizade.

Agradeço, em ordem alfabética, aos meus amigos Alexandre, Archimedes Júnior, Carlildson, Ciro, Eloneide, Érico George, Gildemir, João César, Leonardo, Lindemberg, Magno, Nauri e Rosiane pela incondicional amizade.

Um agradecimento especial ao amigo Luís Henrique pelo apoio nas idéias algorítmicas.

Agradeço ao amigo Anselmo, pelas elucidativas conversas sobre Investigação Operacional, bem como pela sincera amizade.

Agradeço aos amigos Regis, Glawther e Lucas pela amizade fraternal e pela formidável convivência.

Agradeço à Universidade Federal do Ceará, à Universidade de Fortaleza e ao Instituto Federal de Educação, Ciência e Tecnologia pelas condições de trabalho que apoiaram o desenvolvimento de minha tese.

Agradeço às minhas tias Myrtes e Teresa Prata, bem como aos meus primos Mary Rose e Filipe, por terem me acolhido com tamanho afeto quando cheguei ao Porto. O apoio de todos foi imprescindível para a conclusão desta Tese. Meu muito obrigado!

Agradeço ao meu tio Archimedes Fortes Avelino pelo inestimável apoio, pelos conselhos e pelos inúmeros conhecimentos de Matemática, Física e Engenharia concedidos nestes últimos anos.

Agradeço ao meu cunhado Antônio Rodolfo Paulino Pessoa pelo inestimável apoio.

Agradeço aos meus sogros Ivan Linhares Pessoa e Maria Carmosa Paulino Sena pelos incomensuráveis apoio e consideração.

Agradeço aos meus irmãos Tárík de Athayde Prata e Ananda Badaró de Athayde Prata pelo inestimável apoio concedido.

Agradeço aos meus queridos pais, Benedito José Nogueira Prata e Helena Badaró de Athayde Prata, por moldarem o meu caráter de maneira singular e por me apoiarem com tamanho afinco ao longo dos meus anos de vida. Sem dúvida alguma, todos os postos que galguei até hoje foram alcançados pelas condições propiciadas por eles.

Agradeço à Nadja Paulino Pessoa Prata, minha querida esposa, por tudo o que representa em minha vida. Princesa, sem teu apoio jamais teria concluído este trabalho. Lamento pelo trajeto tão árduo, mas, enfim, deu tudo certo.

“É preciso estar sempre do lado da razão com tal firmeza que nem a paixão do povo nem a violência tirânica façam com que se desvie dela. Quem será, porém, essa fênix da equidade? A integridade tem poucos seguidores constantes. Muitos a elogiam, mas poucos a praticam. Outros a seguem até o limite do perigo: então os falsos a renegam e os políticos a simulam. As pessoas íntegras não se importam de opor-se à amizade, ao poder e à própria conveniência. Já os astutos elaboram desculpas sutis, seja por motivos superiores, seja por razões de Estado. O homem constante, no entanto, considera a dissimulação uma traição e valoriza mais a tenacidade que a sagacidade. Está ao lado da verdade e, se acaba se afastando dos demais, não é por inconstância, mas porque em primeiro lugar está a razão”.

Baltasar Gracián

Lista de acrónimos

BDSP	<i>Bus Driver Scheduling Problem</i>
B&B	<i>Branch and Bound</i>
CSP	<i>Crew Scheduling Problem</i>
ETUFOR	Empresa de Transportes Públicos de Fortaleza
GA	<i>Genetic Algorithm</i>
GAUG	<i>Genetic Algorithm with Unexpressed Genes</i>
GIST	Gestão Integrada de Sistemas de Transporte
GRASP	<i>Greedy Randomized Adaptive Search Procedures</i>
HGA	<i>Hybrid Genetic Algorithm</i>
IP	<i>Integer Programming</i>
LP	<i>Linear Programming</i>
MCP	<i>Maximal Covering Problem</i>
MCPMR	<i>Maximal Covering Problem with Multiples Resources</i>
MCLP	<i>Maximal Covering Location Problem</i>
MFSP	<i>Minimum Fleet Size Problem</i>
MDVSP	<i>Multiple Depot Vehicle Scheduling Problem</i>
MOGA	<i>Multi-Objective Genetic Algorithm</i>
MOKP	<i>Multi-Objective Knapsack Problem</i>
MOMCP	<i>Multi-Objective Maximum Covering Problem</i>
NSGA	<i>Nondominated Sorting Genetic Algorithm</i>
PESA	<i>Pareto Envelope-based Selection Algorithm</i>
PR	<i>Path Relinking</i>
RCL	<i>Restricted Candidact List</i>
RSPP	<i>Relaxed Set Partitioning Problem</i>
SA	<i>Simulated Anenaling</i>
SAD	Sistemas de Apoio à Decisão
SCP	<i>Set Covering Problem</i>
SDVSP	<i>Single Depot Vehicle Scheduling Problem</i>
SPEA	<i>Strenght Pareto Evolutionay Algorithm</i>
SPP	<i>Set Partitioning Problem</i>

VCSP	<i>Vehicle ans Crew Scheduling Problem</i>
VEGA	<i>Vector Evaluated Genetic Algorithm</i>
VSP	<i>Vehicle Scheduling Problem</i>

Resumo

Tradicionalmente, a programação de veículos e de tripulações é realizada manualmente pelos técnicos das empresas de transporte público. Devido à sua complexidade, estas tarefas podem consumir dias ou mesmo semanas de planeamento, sendo que o uso de computadores tem vindo a disseminar-se no sector, com vista a agilizar e flexibilizar tal processo. Por outro lado, é usual efectuar a afectação de veículos e de tripulantes de uma maneira sequencial. Todavia, esta prática de separação dos problemas não aproveita os recursos da melhor forma, enquanto a programação simultânea de veículos e tripulantes permitirá reduzir mais significativamente os custos operacionais.

Tendo em consideração a gama de variáveis relacionadas com o planeamento dos veículos e dos motoristas, há uma série de características práticas do problema que não têm vindo a ser contempladas nas soluções geradas computacionalmente. Entre essas características, frequentemente ignoradas, conta-se a existência de múltiplos objectivos nessa programação.

A presente Tese teve como objectivo principal desenvolver uma abordagem multiobjectivo, baseada em metaheurísticas, para a optimização integrada da programação de veículos e tripulações em sistemas de transporte público.

Foram desenvolvidos modelos matemáticos para a programação de veículos e tripulantes, os quais possibilitam uma resolução do problema mais rápida do que as abordagens tradicionais. Para esses problemas, foram também desenvolvidos algoritmos heurísticos que apresentam excelentes resultados computacionais. Em seguida, efectuou-se a extensão multi-objectivo dos algoritmos propostos para o caso mono-critério.

As abordagens desenvolvidas foram testadas em dados reais de sistemas de transporte público, com o objectivo de avaliar os ganhos advindos de sua aplicação. Verificou-se que o uso das abordagens propostas conduziu a soluções de melhor qualidade do que as verificadas na prática corrente. Deste modo, os algoritmos desenvolvidos poderão ser utilizados com vantagem, no planeamento de empresas do sector.

Palavras-chave: Planeamento Operacional em Sistemas de Transportes Públicos, Optimização Combinatória, Metaheurísticas multiobjectivo, Sistemas de Apoio à Decisão.

Abstract

Traditionally, the vehicle and crew scheduling processes are performed manually by human planners in public transportation companies. Due to their complexity, these tasks may take days or even weeks of planning, although the use of computers has been disseminated in the area in order to accelerate such processes and make them more flexible. On the other hand, it is usual to schedule vehicles and crews in a sequential manner. However, this practice of separating problems does not take the full advantage from the available resources, as the simultaneous scheduling of vehicles and crews will in general lead to a significant reduction of the operational costs.

Taking into consideration the range of variables related to the planning process of vehicles and drivers, there are several practical characteristics of the problem that are not reflected in the solutions generated computationally. Among these characteristics that are frequently ignored, probably the most important is the existence of multiple objectives.

The main goal of this thesis was to develop a new approach for the integrated vehicle and crew scheduling problem.

Mathematical models were developed for this problem. These models produce solutions for the problem in a faster way than the traditional approaches. Heuristic algorithms were also developed, with excellent results. Next, multiobjective extensions of the algorithms proposed for the mono-criteria case have been developed.

The approaches were tested with real data of public transportation systems, to assess the gains resulting from their application. The solutions obtained by the proposed approaches were in general of better quality than the ones currently produced. These algorithms can therefore be used by companies, with significant advantages.

Key-words: Operational Planning in Public Transportation Systems, Combinatorial Optimization, Multiobjective metaheuristics, Decision Support Systems.

Resumé

Traditionnellement, la planification des véhicules et des équipages ou des chauffeurs est réalisée manuellement par des éléments des sociétés de transports en commun. Dû à sa complexité, cette planification peut prendre plusieurs jours voire des semaines. Si d'un côté, l'informatique est utilisée dans le secteur pour aider et flexibiliser le processus, d'un autre, il est habituel d'effectuer la programmation des véhicules et des équipages de manière séquentielle. Toutefois, cette pratique de séparation des problèmes n'optimise pas l'utilisation des ressources, puis que la planification simultanée des véhicules et des équipages permettrait de réduire considérablement les coûts opérationnels.

Si on tient compte de la variété des caractéristiques pratiques de la programmation des véhicules et des conducteurs, c'est évident que les approches informatiques actuelles sont, en général, insuffisantes. Parmi ces caractéristiques, fréquemment ignorées, il y a de multiples objectifs.

Cette thèse a pour objectif principal de développer une nouvelle approche pour la planification des véhicules et des équipages.

Des modèles mathématiques pour ce problème ont été créés permettant de trouver une solution plus rapidement que les approches traditionnelles. Des algorithmes heuristiques ont été développés avec d'excellents résultats. Ensuite, on a fait l'extension multi-objective des algorithmes proposées pour les cas monocritères.

Les approches développées ont été testées avec des données réelles des systèmes des transports en commun permettant d'évaluer les gains issus de leur application. Les résultats sont meilleurs que ceux obtenus actuellement en pratique, ce qui prouve le potentiel des approches développées par cette recherche.

Mots-clés : Planification opérationnel des systèmes des transports en commun, Optimisation Combinatoire, Méta heuristiques, Systèmes d'Appui à la Décision.

Lista de Figuras

Figura 1: Diagrama com as contribuições da Tese.....	5
Figura 2: O processo de planeamento de transportes públicos.....	10
Figura 3: Estrutura da abordagem proposta para o VCSP com <i>changeovers</i>	87
Figura 4: Exemplo de movimento de inserção caudal.....	90
Figura 5: Exemplo de movimento de inserção de blocos.	91
Figura 6: Visão esquemática de um Algoritmo Genético.	94
Figura 7: Codificação binária do cromossoma de um indivíduo.....	96
Figura 8: Cruzamento uniforme.	97
Figura 9: Operador de mutação.	98
Figura 10: Visão esquemática da aplicação dos algoritmos.....	101
Figura 11: Soluções obtidas em uma execução do GA – instância 833.	105
Figura 12: Histograma das soluções obtidas em uma execução do GA – instância 833.	106
Figura 13: Soluções possíveis para o exemplo do MOKP.....	112
Figura 14: Ilustração do conceito de hipercubo.....	131
Figura 15: Aproximação para a fronteira de Pareto – MOMCP (instância NW12).	141
Figura 16: Aproximação da Fronteira de Pareto – instância 905.	157
Figura 17: Aproximação da Fronteira de Pareto – instância 122.	159
Figura 18: Aproximação da Fronteira de Pareto – instância 070.	160
Figura 19: Aproximações obtidas para a fronteira de Pareto no caso em estudo.....	167

Lista de Tabelas

Tabela 1: Principais termos usados na Tese.....	7
Tabela 2: Principais características dos casos de estudo citados na literatura.	41
Tabela 3: Características das instâncias e resultados do <i>solver</i> – instâncias adaptadas do SPP.....	63
Tabela 4: Características das instâncias e resultados do <i>solver</i> – instâncias adaptadas do SCP.	63
Tabela 5: Soluções obtidas pela heurística GRASP proposta – instâncias adaptadas do SPP.....	67
Tabela 6: Soluções obtidas pela heurística GRASP proposta – instâncias adaptadas do SCP.	67
Tabela 7: Desempenho computacional da heurística GRASP proposta – instâncias adaptadas do SPP.....	69
Tabela 8: Desempenho computacional da heurística GRASP proposta – instâncias adaptadas do SCP.....	69
Tabela 9: Quantidade de soluções avaliadas pela heurística GRASP proposta – instâncias adaptadas do SPP.....	70
Tabela 10: Quantidade de soluções avaliadas pela heurística GRASP proposta – instâncias adaptadas do SCP.	70
Tabela 11: Comparação entre o método exacto e a heurística GRASP proposta – instâncias adaptadas do SPP.....	71
Tabela 12: Comparação entre o método exacto e a heurística GRASP proposta – instâncias adaptadas do SPP.....	71
Tabela 13: Características das instâncias e resultados do <i>solver</i>	84
Tabela 14: Resultado da programação de veículos para a instância 406 – um depósito.....	84
Tabela 15: Resultado da selecção de serviços para a instância 406 – um depósito.....	85
Tabela 16: Soluções obtidas pela abordagem proposta.	102
Tabela 17: Comparação entre o método exacto e o Algoritmo Genético proposto.....	104
Tabela 18: Comparação de performance entre o <i>branch-and-bound</i> e as abordagens propostas.....	107
Tabela 19: Dados do exemplo do MOKP.	111
Tabela 20: Soluções possíveis para o exemplo do MOKP.	112
Tabela 21: Síntese das características dos algoritmos analisados.	132
Tabela 22: Características das instâncias adaptadas do SPP.....	140
Tabela 23: Quantidade de soluções não-dominadas obtidas pela heurística PESA-II proposta.	140
Tabela 24: Desempenho computacional da heurística PESA-II proposta.	141
Tabela 25: Hipervolumes obtidos com base nas soluções não-dominadas.....	143
Tabela 26: Análise estatística dos hipervolumes obtidos.....	143
Tabela 27: Experimentos computacionais.....	153
Tabela 28: Comparação entre as soluções óptimas e as soluções geradas manualmente.	154
Tabela 29: Experimentos computacionais – instância 905.	157
Tabela 30: Soluções para a instância 905.	158
Tabela 31: Experimentos computacionais – instância 122.	158
Tabela 32: Soluções para a instância 122.	159

	xx
Tabela 33: Experimentos computacionais – instância 070.	160
Tabela 34: Soluções para a instância 070.	161
Tabela 35: Comparação entre GRASP_PESA-II e GRASP_PESA-II_B&B.	167
Tabela 36: Soluções não-dominadas para a instância de Portimão.	168

Lista de Quadros

Quadro 1: Heurística GRASP.....	54
Quadro 2: Adaptação da heurística de Chvátal para a fase de construção.....	55
Quadro 3: Procedimento de construção da RCL.	56
Quadro 4: Algoritmo de melhoria para a variante do MCP.	59
Quadro 5: Procedimento para geração de serviços.....	81
Quadro 6: Heurística construtiva para a programação de veículos.	88
Quadro 7: Algoritmo do movimento de inserção caudal.	89
Quadro 8: Algoritmo do movimento de inserção de blocos.....	90
Quadro 9: Heurística GRASP para a programação de veículos.....	91
Quadro 10: Algoritmo Genético padrão.....	93
Quadro 11: Algoritmo Genético proposto.....	95
Quadro 12: Procedimento para geração de um indivíduo da população inicial.....	96
Quadro 13: Procedimento DROP.	98
Quadro 14: Procedimento ADD.	99
Quadro 15: Procedimento para reposição da população.....	100
Quadro 16: Algoritmo PESA-II proposto para o VCSP.	134
Quadro 17: Procedimento para geração de um indivíduo população inicial.....	135
Quadro 18: Obtenção das soluções não-dominadas na população inicial.....	136
Quadro 19: Procedimento DROP.	138
Quadro 20: Procedimento ADD.	138
Quadro 21: Procedimento para geração de serviços.....	151

Índice

Capítulo 1 - Introdução.....	1
1.1 Motivação.....	1
1.2 Objectivos	3
1.3 Contribuições da Tese	4
1.4 Sinopse da Tese.....	6
1.5 Glossário	7
Capítulo 2 - Programação integrada de veículos e motoristas: uma visão geral.....	9
2.1 Planeamento operacional em companhias de transporte público.....	9
2.2 Abordagens tradicionais para a programação de veículos e tripulantes.....	14
2.2.3 Sistemas de Apoio à Decisão em empresas de transporte público	24
2.3 Programação integrada de veículos e motoristas	25
2.3.1 Considerações iniciais	25
2.3.2 Modelos matemáticos para o VCSP.....	27
2.3.3 Abordagens para o VCSP.....	32
2.4 Considerações finais.....	40
Capítulo 3 - Uma variante do Problema de Máxima Cobertura para a programação de veículos e tripulantes sem <i>changeovers</i>.....	43
3.1 Introdução ao Problema de Máxima Cobertura	43
3.2 Formulação para o VCSP sem <i>changeovers</i>	46
3.3 Abordagens heurísticas para o MCP e o SPP.....	49
3.4 Uma heurística GRASP para a variante do MCP proposta	53
3.5 Resultados computacionais	60
3.6 Considerações finais.....	73
Capítulo 4 - O problema de programação integrada de veículos e tripulações.....	75
4.1 Uma formulação inovadora para o VCSP com <i>changeovers</i>	75
4.2 Resultados computacionais	81
4.3 Uma nova abordagem heurística para o VCSP com <i>changeovers</i>	86
4.4 Teste e avaliação dos algoritmos.....	101
4.5 Considerações finais.....	107

Capítulo 5 - Modelação multiobjectivo para a programação de veículos e tripulações	109
5.1. Fundamentos de optimização multiobjectivo.....	109
5.2 Natureza multiobjectivo do VCSP	113
5.3 Discussão dos possíveis objectivos	117
5.3.1 Objectivos para o VSP	117
5.3.2 Objectivos para o CSP.....	119
5.3.3 Objectivos para o VCSP.....	124
5.4 Uma abordagem multiobjectivo para o VCSP	125
5.4.1 Algoritmos Genéticos multiobjectivo	127
5.4.2 Selecção da heurística a utilizar	133
5.4.3 Um Algoritmo Genético multi-objectivo para o VCSP	134
5.5 Resultados computacionais	139
5.6 Considerações finais.....	144
Capítulo 6 - Aplicação aos casos de estudo	145
6.1 Considerações iniciais.....	145
6.2 Caso de estudo 1 – Fortaleza.....	146
6.2.1 O sistema de transporte público de Fortaleza.....	146
6.2.2 Modelo mono-objectivo e optimização	149
6.2.3 Extensão multiobjectivo	155
6.3 Programação de veículos e motoristas em Portimão.....	161
6.3.1 O sistema de transporte público de Portimão.....	161
6.3.2 Modelo mono-objectivo e optimização.....	163
6.3.3 Aplicação da extensão multiobjectivo.....	164
6.4 Considerações finais.....	168
Capítulo 7 - Conclusões	171
7.1 Principais contribuições da tese.....	171
7.2 Limitações do trabalho realizado	172
7.3 Sugestões para trabalhos futuros	173
Referências bibliográficas	175

Capítulo 1

Introdução

“A problem never exists in isolation; it is surrounded by other problems in space and time. The more of the context of a problem that a scientist can comprehend, the greater are his chances of finding a truly adequate solution.”

Russell Ackoff

Resumo

Este capítulo tem como objectivo apresentar uma visão geral da Tese de Doutoramento, a qual consiste no desenvolvimento de uma abordagem multiobjectivo, baseada em metaheurísticas, para o problema integrado de programação de veículos e motoristas. Na primeira secção, é apresentada a motivação da investigação. Na segunda secção, os objectivos do estudo são definidos. Na última secção, o escopo da Tese é apresentado.

1.1 Motivação

A operação de transportes urbanos representa elevados custos para as empresas do sector, sendo que os componentes do sistema que mais contribuem para tais custos são os veículos e os tripulantes. Deste modo, a programação de viaturas e de tripulantes é de suma relevância para uma utilização racional de tais recursos e, por conseguinte, para a eficiência operacional do sistema.

Sob o ponto de vista do operador do sistema, seja ele público ou privado, a redução de custos é primordial para o incremento da eficiência, podendo também contribuir, mesmo que indirectamente, para o aumento do nível de serviço oferecido ao cliente.

O planeamento dos transportes públicos de passageiros pode ser dividido nas seguintes etapas (SOUSA *et al.*, 2000; DIAS *et al.*, 2001; CEDER, 2002): definição da

rede de transportes, definição dos horários ao público, definição dos horários das viaturas, definição dos horários das tripulações e rotação das tripulações. As actividades de definição dos horários de viaturas e tripulações têm vindo a ser objecto de vários projectos de investigação, devido à redução de custos que pode resultar de sua optimização (DADUNA e PAIXÃO, 1995; WREN e ROSSEAU, 1995; HUISMAN, 2004; WEIDER, 2007).

Em muitas pequenas e médias empresas de transporte público, a definição dos horários das viaturas e a definição dos horários das tripulações é realizada manualmente, com base no conhecimento prático que os planeadores têm acerca de tais problemas. Dada a sua natureza combinatória, construir horários de autocarros e de pessoal tripulante é um processo laborioso, que pode consumir dias ou mesmo semanas de planeamento. O apoio de soluções informáticas permite agilizar o processo de planeamento, contribuindo, inclusive, para a obtenção de soluções de maior qualidade (AZEVEDO FILHO *et al.*, 1994; WREN e GUALDA, 1999).

De acordo com Daduna e Paixão (1995), a programação de veículos não pode ser realizada eficientemente sem a intervenção do planeador. Segundo os autores, é impossível, seja hoje ou no futuro, gerar blocos de viagens exclusivamente com o computador, sem qualquer intervenção humana. Isto ocorre pois, na prática dos sistemas de transporte público, normalmente existe uma gama de restrições as quais não podem ser contempladas por um modelo. Raciocínio análogo poderia ser apresentada para a programação de tripulações.

Nesse contexto, é de notar a importância de serem desenvolvidos métodos que tenham em conta a natureza multiobjectivo do problema. Dessa forma, as soluções informáticas podem se aproximar das necessidades práticas dos sistemas modelados.

Diversos trabalhos vêm ressaltando os ganhos, expressos em termos de reduções de custos operacionais, da afectação de veículos e tripulações efectuada em conjunto (BALL *et al.*, 1983; BODIN *et al.*, 1983; FRELING *et al.*, 1999b). O problema de optimização combinatória associado a esta abordagem é denominado *Vehicle and Crew Scheduling Problem* – VCSP.

O VCSP, ao resultar da combinação de dois difíceis problemas de optimização combinatória, constitui um problema extremamente complexo. Diversos trabalhos, como, por exemplo, Gaffi e Nonato (1999), Friberg e Haase (1999), Haase *et al.* (2001), Freling *et al.* (2003), Huisman (2004), Laurent e Hao (2007), Steizen (2007), Mesquita e Pias (2008), têm apresentado abordagens para o VCSP. Apesar de tais trabalhos

constituem excelentes contribuições para a resolução do VCSP, aspectos tais como modularidade, baixa dependência de parâmetros, soluções de qualidade em baixo tempo de processamento, necessitam ainda de ser desenvolvidos.

Outro aspecto relevante é a consideração de múltiplos objectivos na resolução do problema. Conforme discutido acima, Daduna e Paixão (1995), bem como Wren e Gualda (1995), apontam para a necessidade de uma análise crítica, por parte dos planeadores, das soluções geradas por programas de computador, em problemas de planeamento de sistemas de transportes públicos.

Modelos de programação matemática, bem como algoritmos heurísticos tradicionais, usualmente primam por obter o valor extremo (máximo ou mínimo) de uma função de desempenho, ou mesmo uma aproximação satisfatória deste. Em abordagens dessa natureza, geralmente não são contemplados alguns critérios que são relevantes para a qualidade de uma solução na prática. Portanto, percebe-se a necessidade de desenvolver métodos para a resolução do VCSP que possam considerar múltiplos objectivos.

Os *changeovers* representam o número de mudanças de motoristas por veículo, em um serviço. Quando os *changeovers* não são permitidos, o motorista e o veículo consistem em um único recurso; caso contrário, os motoristas podem mudar de veículo na execução de um dado serviço.

Os *changeovers* são uma importante característica do VCSP, visto que, conforme o caso de serem ou não permitidos, podem ser empregadas abordagens distintas para a resolução do problema.

1.2 Objectivos

O objectivo geral da investigação é desenvolver uma abordagem multiobjectivo, baseada em metaheurísticas, para a optimização integrada da programação de veículos e de motoristas em sistemas de transporte público.

Como objectivos específicos do presente estudo, podem ser destacados:

- a caracterização da interacção entre os problemas de programação de veículos e de tripulações, a fim de ampliar a compreensão acerca da natureza do problema;
- o desenvolvimento de modelos matemáticos para o problema integrado de programação de veículos e tripulações;

- a resolução exacta dos modelos desenvolvidos em instâncias geradas aleatoriamente e em instâncias de teste existentes na literatura;
- o desenvolvimento de metaheurísticas mono-objectivo para a abordagem integrada do problema;
- a identificação e selecção dos diversos objectivos relacionados com o problema integrado;
- o desenvolvimento de metaheurísticas multiobjectivo para a abordagem integrada do problema;
- a aplicação da abordagem proposta em casos de estudo reais, de modo a validá-la sob o ponto de vista prático;
- a comparação das soluções geradas pela abordagem proposta com as soluções utilizadas na prática da operação dos sistemas de transporte.

Com base no exposto, foram formuladas as seguintes *questões de investigação*:

- Como se dá a interacção entre os problemas de programação de veículos e de tripulações?
- Que modelos matemáticos deverão ser adoptados para tratar a complexidade do problema e, ao mesmo tempo, permitir a sua resolução?
- Quais as metaheurísticas mais adequadas para a resolução deste problema?
- Para as metaheurísticas seleccionadas, qual o tipo de codificação e de estruturas de vizinhança a utilizar na resolução do problema integrado de programação de veículos e tripulações?
- Quais os objectivos relevantes para o VCSP? Que atributos deverão ser considerados para a avaliação desses objectivos?
- Quais os benefícios da abordagem proposta em relação às abordagens que são empregadas correntemente na prática?

1.3 Contribuições da Tese

A seguir, na Figura 1, é apresenta-se um esquema com as principais contribuições desta trabalho. Em amarelo salientam-se os aspectos mais inovadores dessas contribuições.

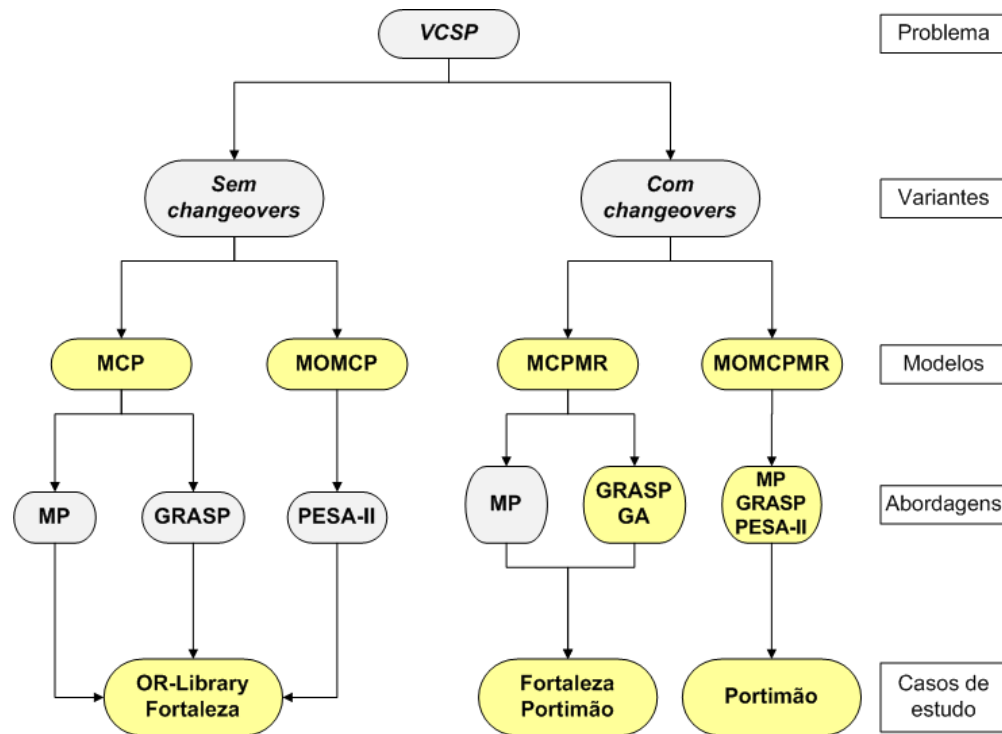


Figura 1: Diagrama com as contribuições da Tese.

O problema VCSP é estudado em suas duas variantes: com e sem permissão de *changeovers* (ver glossário, na secção 1.5).

No que se refere ao VCSP sem *changeovers*, foi proposta uma nova formulação baseada no *Maximal Covering Problem* – MCP. Em seguida, esta formulação foi estendida para um problema envolvendo múltiplos objectivos, dando origem ao *Multiple Objective Maximal Covering Problem* – MOMCP.

No que tange ao VCSP com *changeovers*, foi proposta uma nova formulação, para uma variante inovadora do MCP designada *Maximal Covering Problem with Multiple Resources* – MCPMR. Em seguida, a nova formulação foi estendida para um problema envolvendo múltiplos objectivos, dando origem ao *Multiple Objective Maximal Covering Problem with Multiples Resources* – MOMCPMR.

As formulações propostas permitiram a resolução, de forma exacta, de problemas reais, facto que, conforme a revisão de literatura (vide Capítulo 2), não ocorria com as outras abordagens reportadas.

As extensões multiobjectivo também se mostraram de grande relevância, visto que são escassos trabalhos que abordem o VCSP com foco em múltiplos critérios, notadamente no que concerne à geração de conjuntos de soluções não-dominadas.

Dentre as abordagens genéricas para resolução dos modelos propostos, tem-se a programação matemática (*Mathematical Programming* – MP) e técnicas heurísticas, tais como o GRASP, os Algoritmos Genéticos (GA) ou o PESA-II.

Por fim, foram realizados múltiplos estudos de caso. Primeiramente, foram estudadas instâncias da OR-Library, muitas delas advindas de problemas reais do *Airline Crew Scheduling*. Os sistemas de transportes colectivos das cidades de Fortaleza (Nordeste do Brasil) e Portimão (Sul de Portugal) foram modelados e solucionados, seja por métodos exactos, seja por técnicas heurísticas.

1.4 Sinopse da Tese

Esta Tese de Doutorado está dividida neste capítulo introdutório e em mais seis capítulos, que se apresentam sucintamente, de seguida.

No Capítulo 2 é apresentada uma visão geral do problema de programação de veículos e motoristas. As abordagens tradicionais para solucionar a programação de veículos e tripulantes separadamente são comentadas. Os benefícios da integração entre estas duas programações são salientados. Por fim, os modelos matemáticos, a complexidade computacional e abordagens heurísticas para o VCSP são apresentadas.

No Capítulo 3 é proposto um novo modelo matemático para o VCSP sem *changeovers*, baseado em uma variante do problema de máxima cobertura (*Maximal Covering Problem* – MCP). Propõe-se uma heurística GRASP reactiva para o MCP. Instâncias constantes na literatura, advindas de problemas reais, são estudadas.

No Capítulo 4 é proposto um novo modelo matemático para o VCSP com *changeovers*, baseado em um problema de máxima cobertura com múltiplos recursos. Propõe-se um Algoritmo Genético híbrido para o problema. Instâncias adaptadas de problemas reais são analisadas.

No Capítulo 5 é apresentada uma abordagem multiobjectivo para a programação integrada de viaturas e tripulantes. A capacidade do algoritmo em gerar conjuntos de soluções não-dominadas é avaliada em testes efectuados com instâncias da literatura.

No Capítulo 6 são analisados problemas reais. As instâncias oriundas de problemas práticos agora são estudadas por meio da optimização tradicional e por meio da optimização multiobjectivo. Comparam-se os resultados fornecidos pelos algoritmos propostos com a prática corrente dos sistemas de transporte público analisados.

O Capítulo 7 consiste na apresentação das conclusões da Tese, salientando-se os principais resultados obtidos pelas abordagens propostas, as limitações do estudo, bem como sugestões para investigação futura.

1.5 Glossário

A seguir, na Tabela 1, são apresentados os principais termos utilizados na Tese, com sua terminologia em língua inglesa, seguida do seu significado.

Tabela 1: Principais termos usados na Tese.

Termo usado em inglês	Termo usado no trabalho	Significado
<i>Changeover</i>	<i>Changeover</i>	Troca de tripulante por veículo.
<i>Covering</i>	Cobertura	Atendimento a demanda por parte de um recurso, sendo permitido que mais de um recurso efectue o atendimento.
<i>Crew</i>	Tripulante ou tripulação	Funcionário(s) que efectua(m) a condução dos veículos em sistemas de transportes públicos.
<i>Deadhead</i>	Viagem em vazio ou viagem não-produtiva	Viagem, realizada entre dois terminos, em que não é efectuada a condução de passageiros.
<i>Depot</i>	Depósito ou garagem	Facilidade que acolhe a frota de veículos de uma determinada empresa.
<i>Duty</i>	Serviço	Conjunto de tarefas realizadas por uma tripulação.
<i>Leftover</i>	<i>Leftover</i>	Tramo não coberto por um serviço.
<i>Overcovers</i>	<i>Overcovers</i>	Sobreposição da cobertura de tramos por um dado serviço.
<i>Piece-of-work</i>	Tramo	Menor unidade de trabalho de uma tripulação.
<i>Pull-in</i>	<i>Pull-in</i>	Viagem não-produtiva realizada entre o depósito e o ponto de início de uma viagem.
<i>Pull-out</i>	<i>Pull-out</i>	Viagem não-produtiva realizada entre o ponto de término de uma viagem e o depósito.
<i>Relief point</i>	Ponto de rendição	Local onde os tripulantes podem iniciar ou concluir uma viagem.
<i>Rostering</i>	Rotação	Plano de trabalho de um tripulante ao longo de um período de tempo.
<i>Scheduling</i>	Programação	Afectação de recursos para a realização de actividades.
<i>Timetable</i>	Tabela de horários	Estrutura de dados que contém as viagens a serem realizadas por um sistema de transportes públicos.
<i>Trip</i>	Viagem	Deslocamento de um veículo, com origem e destino determinados, e com horários estabelecidos para início e fim.
<i>Vehicle</i>	Veículo ou viatura	Equipamento que realiza o transporte de passageiros.

Deve-se observar que, para alguns dos termos, foi utilizada a palavra ou expressão originais, em inglês. Procedeu-se desta forma, pois, para tais termos, não foi encontrada uma tradução adequada ou consensual.

A seguir, no Capítulo 2, será apresentada a revisão de literatura sobre a programação de veículos e de tripulantes.

Capítulo 2

Programação integrada de veículos e motoristas: uma visão geral

“Although we have described crew schedules in terms of vehicles, this by no means suggests that vehicles should be scheduled prior to and separately from crews. Unfortunately, in all practical settings this is the approach taken. In the case of aircraft scheduling, this approach is justifiable since aircraft costs dominate air crew costs. In the public transport case, however, crew costs dominate vehicle operating costs and in some cases reach as high 80% of total operating costs. Under these circumstances, it seems foolish to allow vehicles to be scheduled independently of, and without regard to crews”.

Bodin *et al.* (1983).

Resumo

Este capítulo tem como objectivo efectuar um levantamento bibliográfico das abordagens para a programação integrada de veículos e motoristas. O estado da arte foi explorado em uma extensa revisão bibliográfica, sendo que os trabalhos considerados mais importantes foram destacados no presente Capítulo. Na primeira secção, o processo de planeamento de sistemas de transportes públicos é apresentado. Na segunda secção, as abordagens para a programação de veículos e de motoristas são descritas. Na terceira secção, o problema de programação integrada de veículos e motoristas é discutido, em termos de princípios, motivação, abordagens exactas e heurísticas. Por fim, são feitas algumas considerações gerais sobre a revisão bibliográfica realizada.

2.1 Planeamento operacional em companhias de transporte público

De acordo com Manheim (1980) um sistema de transportes é composto pelos passageiros e bens a serem transportados, pelos veículos que efectuem estes deslocamentos e pela rede de infra-estrutura de transportes que permite a operação do sistema. Nesse contexto, a Engenharia de Transportes prima por planear, projectar, construir, operar, manter e gerenciar sistemas de transportes.

O transporte urbano refere-se ao transporte realizado nas cidades ou em áreas metropolitanas, seja ele de passageiros ou de cargas. Já o transporte público diz respeito ao transporte de passageiros, o qual geralmente é organizado em linhas regulares, com rotas e horários pré-estabelecidos.

O planeamento dos transportes públicos é composto pelas seguintes principais etapas (SOUSA *et al.*, 2000; DIAS *et al.*, 2001; CEDER, 2002): definição da rede de transportes, definição da tabela de horários, afectação dos veículos às viagens e afectação das tripulações aos veículos. Weider (2007) divide o planeamento dos transportes públicos em duas grandes fases: o planeamento estratégico, que compreende o projecto da rede de transportes, o planeamento das linhas e a definição da tabela de horários; e o planeamento operacional, o qual consiste na programação dos veículos, na afectação de serviços e na rotação das tripulações. Na Figura 2, é ilustrado, de forma esquemática, o processo de planeamento de transportes públicos.

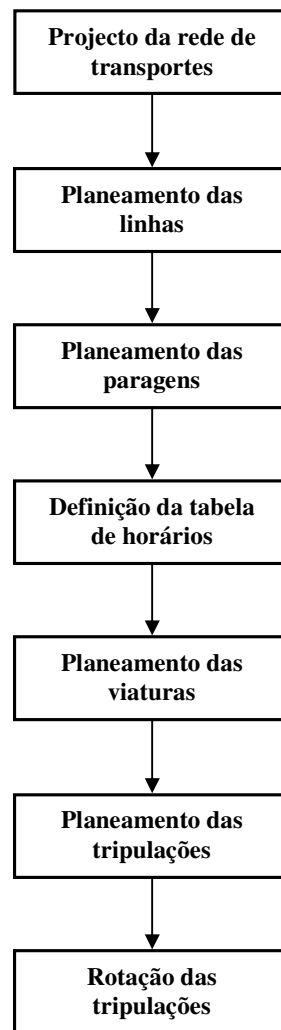


Figura 2: O processo de planeamento de transportes públicos.

O projecto da rede consiste na definição das rotas a serem seguidas pelos veículos, de forma a satisfazer uma procura prognosticada. O planeamento das linhas

consiste na definição das rotas entre pontos de origem-destino para satisfazer a procura estipulada, assim como as frequências associadas a cada rota. O planeamento das paragens consiste na determinação de pontos intermédios, nos quais os passageiros irão embarcar e desembarcar nos veículos. A definição da tabela de horários vai transformar rotas, frequências e a topologia da rede em viagens. O planeamento das viaturas consiste na afectação de veículos às viagens definidas na tabela de horários. O planeamento das tripulações ou dos motoristas consiste na afectação de tripulantes aos veículos que irão efectuar as viagens planeadas, gerando um conjunto de serviços. Por fim, o planeamento das rotações das tripulações consiste na determinação do plano de trabalho de cada tripulante ao longo de um período, seja uma semana ou um mês de trabalho, por exemplo.

Estes problemas são de natureza diferente, visto que o planeamento das linhas, das paragens e a definição da tabela de horário são problemas em um nível estratégico e a programação dos serviços de viaturas e tripulações, bem como as rotações, são problemas em um nível operacional.

Apesar do processo apresentado na Figura 2 ser sequencial, o planeamento de sistemas de transportes públicos possui retro-alimentações (*feedbacks*) entre cada elo e os demais, pois todas estas etapas estão interligadas, umas mais fortemente do que outras. Deste modo, torna-se importante efectuar um planeamento integrado de todas estas etapas, com vistas a obter um sistema de transportes mais eficaz e eficiente.

Ceder (2002) destaca os seguintes dados a serem levantados, no que se refere ao planeamento operacional de transportes públicos:

- *Topologia da rede de atendimento:* (i) número de rotas; (ii) nós, paragens e tempos em uma rota; (iii) padrões (sequências de nós em uma rota).
As rotas (linhas) consistem em itinerários efectuados pelas viaturas. O tempo decorrido entre o início e o término de uma viagem é denotado tempo de ciclo. Uma rede de transportes possui diversos pontos de interface com a procura, tais como as paragens ou as estações. A sequência de nós em uma rota envolvem tanto as paragens quanto as estações.
- *Procura de passageiros e nível de serviço:* (i) carregamentos de passageiros entre dois pontos adjacentes em uma rota; (ii) número desejado de passageiros a bordo do veículo em trânsito; (iii) política de frequência dos veículos.

A quantidade de passageiros entre dois nós adjacentes em uma rota é um indicador da qualidade do serviço prestado. Outro indicador de nível de serviço é o *headway*, o qual consiste na frequência do serviço oferecido, ou seja, no tempo decorrido entre a passagem de dois veículos da mesma rota em um dado nó da rede de transportes.

- *Características dos veículos:* (i) tipo de veículo; (ii) capacidade do veículo; (iii) tempo de viagem do veículo entre pontos notáveis de uma rota.

As viaturas podem ser de diversos tipos, com características distintas, tais como dimensões, capacidade, velocidade ou combustível requerido para operação. Cada tipo de veículo possui um tempo de autonomia, no qual pode operar sem a necessidade de reabastecimento.

- *Características das viagens:* (i) tempos de escalas (*layovers*) máximos e mínimos; (ii) tolerâncias nas partidas das viagens (atraso de partida máximo e adiantamento de partida máximo).

Em sistemas de transporte público, um *layover* representa a paragem de um motorista e/ou veículo no fim de uma viagem antes da operação da rota reversa ou da próxima viagem. Podem ser estipulados valores máximos e mínimos para a duração dos *layovers*, bem como podem ser tolerados atrasos ou adiantamentos.

- *Informações sobre viagens em vazio (deadheads):* (i) nome e localização das garagens; (ii) lista dos locais de início e término das viagens; (iii) tempos de viagens em vazio a partir da garagem para os locais de início das viagens (*pull-outs*); (iv) tempos de viagens em vazio dos locais de término das viagens para a garagem (*pull-ins*); (v) matriz de tempos de viagens em vazio entre todas as origens e destinos das viagens.

As viagens podem ser classificadas como produtivas, quando os veículos efectuam o transporte de passageiros, ou improdutivas, caso contrário.

- *Informações sobre pontos de rendição:* (i) localização dos pontos de rendição; (ii) tempos de viagens entre os pontos de rendição.

Um ponto de rendição é um local no qual pode ocorrer a troca de motoristas, na execução de um serviço. A rendição pode ocorrer no início de uma viagem, ou, em alguns casos, durante a mesma.

- *Restrições nos serviços de pessoal tripulante:* (i) tipo de serviço; (ii) duração do serviço; (iii) número de mudanças de veículo em um serviço (*changeovers*); (iv) intervalos para refeição; (v) composição do serviço; (vi) outras regras de regime de trabalho.

Cada empresa de transportes públicos pode permitir a ocorrência de diferentes tipos de serviços. Por exemplo, podem-se permitir serviços contínuos (sem interrupções) ou serviços com paragens. Uma questão importante no que se refere aos tipos de serviços se refere à jornada de trabalho. Geralmente as legislações laborais impõem que, quando a duração de um serviço excede a um dado valor padrão de jornada de trabalho, o funcionário seja remunerado com horas de trabalho extraordinário. A legislação trabalhista também impõe, de forma usual, que o funcionário tenha direito a um intervalo para refeição, não remunerado, durante a sua jornada de trabalho.

- *Prioridade de tripulantes e regras de rotação:* (i) lista dos tripulantes por nome e tipo; (ii) prioridade do tripulante ou regras de igualdade; (iii) descrição da jornada de trabalho.

A rotação (*rostering*) se refere ao planeamento das actividades dos tripulantes durante um determinado período. Trata-se de um problema mais amplo do que o *crew scheduling*.

Huisman (2004) analisa as diferenças entre planeamento de transportes públicos de autocarros, aviões e comboios e conclui que o planeamento de autocarros e aviões tem o mesmo grau de dificuldade, enquanto que o planeamento do transporte ferroviário é muito mais complexo do que as demais modalidades de transporte mencionadas. A complexidade do planeamento do transporte ferroviário diz respeito às restrições operacionais dos veículos, que dificultam sua programação.

É importante salientar que cada uma destas componentes do processo de planeamento operacional de sistemas de transportes públicos é extremamente complexo. Diante deste facto, considerar todas as etapas em um processo de planeamento integrado

tornaria tal prática praticamente inviável. Portanto, o planeador deve sempre gerir o *trade-off* entre os ganhos de efectividade advindos da integração e a complexidade do sistema a ser planeado.

No que concerne à afectação de veículos e de tripulantes, a prática recente tem demonstrado os ganhos ao se considerar uma abordagem integrada. Na secção 2.3 do presente Capítulo serão detalhados os benefícios desta abordagem.

Ceder (2001) apresenta uma metodologia para integrar, de forma eficiente, a definição das tabelas de horário e a programação dos veículos. No estudo citado, a geração de tabelas de horários de modo a impor uma correspondência do tempo de partida das viaturas com a procura por parte dos passageiros, pode implicar na redução dos recursos de frota empregados.

Tendo em vista que decisões estratégicas envolvem outro nível de recursos, é muito difícil realizar modificações da rede de transportes de um sistema de transportes públicos, pois tal procedimento envolve custos com remoção e/ou adição de paragens, alteração de tabelas de horários, construção de novas facilidades e elaboração de campanhas publicitárias para informação dos passageiros sobre as mudanças.

Deste modo, a retro-alimentação entre os planeamentos operacional e estratégico de sistemas de transporte público é bastante restrita sob o ponto de vista prático. As possibilidades de integração residem na sua realização no início da concepção do sistema de transporte ou em um processo contínuo de retro-alimentação no planeamento das etapas do planeamento operacional.

As programações de veículos e de pessoal tripulantes são alvos de inúmeras investigações devido à potencialidade de redução de custos advinda de sua optimização (DADUNA e PAIXÃO, 1995; WREN e ROSSEAU, 1995, FRELING *et al.*, 2003; WEIDER, 2007). A seguir, estes dois problemas serão analisados isoladamente, como introdução ao processo de suas integrações.

2.2 Abordagens tradicionais para a programação de veículos e tripulantes

Nesta secção serão apresentadas as abordagens tradicionais para a programação de veículos e tripulantes, sem considerar sua integração. Será dada ênfase às formulações matemáticas mais usuais dos referidos problemas.

2.2.1 *Vehicle scheduling problem*

O problema da programação de veículos (*Vehicle Scheduling Problem – VSP*) consiste em afectar um conjunto de veículos a um conjunto de viagens, de modo a minimizar uma função de desempenho relacionada aos custos de investimento e aos custos operacionais (CARRARESI e GALLO, 1984; FRELING *et al.*, 1999a; BAITA *et al.*, 2000). Tal função normalmente expressa o número de veículos alocados, o custo operacional e os tempos não produtivos (tempos de espera, viagens em vazio, etc.).

De acordo com Daduna e Paixão (1995), os três principais tipos de VSP são: o VSP com um único depósito (*Single Depot Vehicle Scheduling Problem – SDVSP*), o VSP com número fixo de veículos e o VSP com múltiplos depósitos (*Multiple Depot Vehicle Scheduling Problem – MDVSP*). Freling *et al.* (1999a) apresentam algoritmos polinomiais para o SDVSP; contudo, o MDVSP é NP-difícil (DADUNA e PAIXÃO, 1995), conforme será apresentado a seguir.

No VSP, as viagens devem ser efectuadas por uma quantidade de veículos, não definida *a priori*, de forma que as viaturas requeridas estejam alojadas em um único depósito. O VSP com número fixo de veículos assemelha-se ao VSP simples, excepto pelo facto de que o número de veículos é definido *a priori*. No MDVSP, uma quantidade de veículos, em princípio não definida, é requerida para a cobertura de viagens, de modo que as viaturas podem estar alojadas em dois ou mais depósitos.

Segundo Daduna e Paixão (1995), em termos da teoria dos grafos, o VSP pode ser descrito como a determinação do conjunto de circuitos de menor custo em um grafo, de modo a serem cobertas todas as viagens. Se o custo fixo associado aos veículos é elevado, o VSP corresponde a encontrar o número mínimo de veículos necessários para operar todas as viagens, o que correspondente ao serviço com mínimo custo operacional.

Com base na premissa apresentada acima, existem, para este problema, diversos tipos de formulações matemáticas, a saber: modelo de afectação (*assignment model*), modelo do transporte (*transportation model*), modelo de fluxo em redes (*network flow model*), modelo de quase-afectação (*quase-assignment model*) e modelo de emparelhamento (*matching model*). A seguir, será apresentado um modelo de quase-afectação proposto por Freling *et al.* (1999a, 2001a).

Single Depot Vehicle Scheduling Problem

Seja N o conjunto de vértices incluindo as origens das viagens, os destinos das viagens e os depósitos; A o conjunto de arcos compreendendo as viagens e as viagens em vazio; c_{ij} o custo do arco $(i, j) \in A$; e y_{ij} uma variável de decisão binária que é igual a 1 se o veículo cobre a viagem j exactamente após a viagem i , sendo 0 caso contrário. Pode-se assim estabelecer o seguinte modelo matemático:

[SDVSP]

$$\text{Minimizar } \sum_{(i,j) \in A} c_{ij} y_{ij} \quad (2.1)$$

Sujeito a:

$$\sum_{j:(i,j) \in A} y_{ij} = 1 \quad \forall i \in N \quad (2.2)$$

$$\sum_{i:(i,j) \in A} y_{ij} = 1 \quad \forall j \in N \quad (2.3)$$

$$\sum_{j:(i,j) \in A} y_{id} \leq n \quad (2.4)$$

$$\sum_{i:(i,j) \in A} y_{dj} \leq n \quad (2.5)$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in A \quad (2.6)$$

A expressão (2.1) representa a função objectivo, na qual o custo de alocação dos veículos às viagens deve ser minimizado. Os conjuntos de restrições (2.2) e (2.3) garantem que cada viagem será atendida por apenas um veículo. Os conjuntos de restrições (2.4) e (2.5) permitem que no máximo n arcos incidam no depósito. As restrições (2.6) impõem o carácter binário às variáveis de decisão.

Multiple Depot Vehicle Scheduling Problem

Carraresi e Gallo (1984) apresentam um modelo baseado em fluxo em redes com múltiplos produtos (*multicommodity network flow model*) para o MDVSP, o qual é formulado matematicamente como segue:

[MDVSP]

$$\text{Minimizar } \sum_{k=1}^l \sum_{(i,j) \in A \cup A^*} c_{ij} x_{ij}^k \quad (2.7)$$

Sujeito a:

$$\sum_{j \in J(l)} x_{ij}^k = y_{ik} \quad \forall i \in N, \forall k \in L \quad (2.8)$$

$$\sum_{i \in I(l)} x_{ij}^k = y_{jk} \quad \forall j \in N, \forall k \in L \quad (2.9)$$

$$\sum_{(i,j) \in A^*} x_{ij}^k \leq a_k \quad \forall k \in L \quad (2.10)$$

$$\sum_{k=1}^l y_{ik} = 1 \quad \forall i \in N \quad (2.11)$$

$$x_{ij}^k \geq 0 \quad \forall (i,j) \in A \cup A^*, \forall k \in L \quad (2.12)$$

$$y_{ik} \in \{0,1\} \quad \forall i \in N, \forall k \in L \quad (2.13)$$

O MDVSP é modelado em um grafo bipartido do tipo $N = (S, T, A \cup A^*)$, em que $s_i, i=1, \dots, n$, consistem no nós com oferta de fluxo; $t_j, j=1, \dots, n$, consistem nos nós com procura de fluxo, $A = \{(s_i, t_j):(v_i, v_j)\}$ é o conjunto de arcos entre os nós de oferta e procura e as viagens compatíveis, e $A^* = \{(s_i, t_j):(s_i, t_j) \notin A\}$.

No modelo são consideradas n viagens e k depósitos, cada qual com capacidade a_k . Para todo arco $(i, j) \in A$, a variável x_{ij}^k é igual a 1 se um veículo, alojado no depósito k , efectua as viagens v_i e v_j em sequência, sendo igual a 0 caso contrário. Para todo arco $(i, j) \in A^*$, a variável x_{ij}^k é igual a 1 se as viagens v_i e v_j são, respectivamente, a primeira e a última viagens do veículo alocado ao depósito k . Para todo $i=1, \dots, n$, a variável y_{ik} é igual a 1 se a viagem v_i é coberta por um veículo alojado no depósito k , sendo 0 caso contrário.

A expressão (2.7) consiste na função objectivo a ser minimizada, na qual o custo de afectação de veículos às viagens deve ser minimizado, sendo que os veículos podem partir e chegar a múltiplos depósitos. Os conjuntos de restrições (2.8) e (2.9) impõem que uma viagem não seja coberta por mais de um veículo. O conjunto de restrições (2.10) impõe que a capacidade dos depósitos seja respeitada. O conjunto de restrições (2.11) impõe que a viagem v_i seja alocada a apenas um veículo, advindo do depósito k . As restrições (2.12) e (2.13) referem-se à definição das variáveis de decisão.

O MDVSP é NP-difícil (DADUNA e PAIXÃO, 1995), não existindo, por isso, algoritmos polinomiais para sua resolução à optimalidade.

Carraresi e Gallo (1984) descrevem duas heurísticas para o MDVSP: *primeiro agrupa – depois aloca* (*cluster first – schedule second*) e *primeiro aloca – depois agrupa* (*schedule first – cluster second*). Na heurística *primeiro agrupa – depois aloca*, as viagens são agrupadas por depósito para, em seguida, serem programadas. Na heurística *primeira aloca – depois agrupa*, as viagens são programadas em um primeiro estágio para, em seguida, serem agrupadas por depósito.

Pepin *et al.* (2006) apresentam um estudo comparativo entre cinco diferentes abordagens heurísticas para o MDVSP: a heurística de Kliewer *et al.* (2006)¹ para geração de um modelo a ser executado em um *solver* para programação linear inteira mista, uma heurística Lagrangeana, uma heurística de geração de colunas, uma heurística de busca em vizinhança de grande dimensão (*Large Neighborhood Search*) usando geração de colunas para avaliação da vizinhança e um algoritmo Pesquisa Tabu. Esta comparação demonstrou que a técnica de geração de colunas produz soluções de melhor qualidade, quando há tempo computacional disponível.

Baita *et al.* (2000) analisam, na prática, diferentes abordagens de solução do VSP e constatam que muitos requisitos dos problemas reais não são considerados nos métodos tradicionais de solução, baseados em programação matemática ou em heurísticas.

No que concerne ao VSP, os seguintes objectivos podem ser destacados (Baita *et al.*, 2000): minimização do número de veículos; minimização do número de mudanças de linhas; minimização do número e do comprimento das viagens em vazio; e minimização dos tempos ociosos dos veículos nos terminos.

Baita *et al.* (2000) ressaltam a natureza multiobjectivo do problema e propõem alguns algoritmos para a solução do VSP, considerando múltiplos objectivos: um modelo de alocação com ponderação de critérios, uma abordagem baseada em programação lógica e um Algoritmo Genético para produzir um conjunto de soluções não-dominadas. Os autores concluem que o Algoritmo Genético não apresenta performance competitiva, se comparado às outras duas abordagens.

Haghani *et al.* (2003) apresentam um estudo comparativo de modelos para a programação de veículos: dois modelos com um depósito e restrições de tempo de

¹ A heurística de Kliewer *et al.* (2006) consiste em um procedimento para agregar o número de arcos correspondentes a viagens em vazio, de modo a reduzir o porte do modelo.

percurso (*single depot models with route time constraints*) e um modelo com múltiplos depósitos e restrições de tempo de percurso (*multiple depot model with route time constraints*). Ao resolverem instâncias reais de grande porte advindas do sistema de transporte público de Baltimore, Maryland, Estados Unidos, os autores concluem que a velocidade de viagem em vazio (*deadhead speed*) é um importante parâmetro para a programação de veículos. Na verdade, os tempos de viagem influenciam nas compatibilidades entre as viagens e os custos das viagens em vazio. Em geral, quanto maior a velocidade de viagem em vazio, maior é o número de compatibilidades entre as viagens e menor é o tempo de viagem em vazio, resultando num decréscimo do valor das funções objectivos consideradas pelos autores citados.

Haghani *et al.* (2003) demonstram, em diversos experimentos computacionais, que o modelo considerando um único depósito se comporta melhor do que os modelos que consideram múltiplos depósitos.

2.2.2 Crew scheduling problem

O problema de programação de pessoal tripulante (*Crew Scheduling Problem – CSP*) consiste em alocar tripulações para o cumprimento de viagens em sistemas de transportes. Um caso particular do CSP é o problema de programação de motoristas (*Bus Driver Scheduling Problem – BDSP*) no qual se deve construir um conjunto legal de tarefas efectuadas por um motorista em um dia de trabalho, de modo que todos os motoristas cubram as viagens dos veículos (WREN e ROSSEAU, 1995).

O CSP consiste na concepção de serviços de pessoal tripulante para uma predeterminada programação de veículos, sendo que um serviço corresponde ao trabalho diário realizado pelos tripulantes. O tempo consumido pela tripulação corresponde a viagens produtivas, viagens não produtivas (*deadheads*, *pull-in trips* e *pull-out trips*) e tempos de espera. De acordo com as legislações trabalhistas de cada cidade, o CSP possui uma série de restrições inerentes à jornada de trabalho dos tripulantes.

Os locais onde os tripulantes podem iniciar ou terminar uma viagem, ou mesmo um segmento da viagem, são denominados pontos de rendição (*relief points*). O trabalho decorrido entre dois pontos de rendição é denominado tramo (*piece-of-work*) e este corresponde à menor unidade de trabalho de uma tripulação. O serviço de um tripulante é um conjunto de tramos.

O CSP está entre os mais difíceis problemas de optimização combinatória, pertencendo à classe NP-difícil (MARTELLO e TOTH, 1986). Geralmente, são utilizadas duas principais abordagens para resolver o problema: Cobrimento de Conjuntos (*Set Covering Problem – SCP*) e Particionamento de Conjuntos (*Set Partitioning Problem – SPP*).

A modelagem por meio de cobertura de conjuntos é mais desejável sobre o ponto de vista computacional, pois incorre em um menor tempo de processamento se comparada à modelagem baseada em particionamento de conjuntos. Contudo, as soluções geradas pelo modelo SCP precisam ser corrigidas, pois mais de um tripulante pode cobrir um tramo ao mesmo tempo.

Tal problema não ocorre com o modelo SPP, que se mostra mais robusto para a resolução do CSP. Todavia, o SPP possui um espaço de busca extremamente restrito, facto que dificulta bastante a obtenção de soluções admissíveis.

De seguida, apresenta-se um pequeno exemplo que ilustra os modelos SCP e SPP. Uma matriz, denotada por A , representa o conjunto de serviços gerados para um CSP. O número de linhas desta matriz representa o número de tramos e o número de colunas o número de serviços.

Considere-se, então, um exemplo no qual existem 5 tramos a serem cobertos e em que devem ser escolhidos serviços de entre um conjunto de 7 serviços. Logo, tem-se que o número de linhas da matriz A é igual a 5 ($m=5$) e o número de colunas é igual a 7 ($n=7$), conforme ilustrado a seguir.

$$\begin{vmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{vmatrix}$$

O significado da matriz acima é o seguinte. Se $a(i,j)=1$, o i -ésimo tramo é coberto pelo j -ésimo serviço. O facto do elemento $a(4,1)$ ser igual a 1 significa que o quarto tramo ($i=4$) é coberto pelo primeiro serviço ($j=1$). Já o facto do elemento $a(1,2)$ ser nulo significa que o primeiro tramo não é coberto pelo segundo serviço.

Admitindo-se que a cada coluna da matriz ilustrada acima seja associado um custo, pode-se determinar o custo de cobrir os tramos com os serviços disponíveis. Seja

um vector de custos $\mathbf{c} = \{7 \ 12 \ 10 \ 8 \ 15 \ 12 \ 10\}$, de modo que a j -ésima componente de \mathbf{c} represente o custo do j -ésimo serviço. O custo de uma solução será a soma dos custos dos serviços presentes na solução.

Pode verificar-se facilmente que a solução de custo mínimo consiste na selecção dos serviços 5 e 6, incorrendo em um custo igual a 27. Contudo, o terceiro tramo é coberto concomitantemente pelos dois serviços seleccionados. Na prática, o significado da sobreposição destes dois serviços representa o uso simultâneo de dois motoristas em um mesmo veículo, para a cobertura do terceiro tramo. Esta sobreposição de serviços é conhecida como *overcover*. Na prática os *overcovers* são indesejáveis, pois acarretam ineficiência na utilização dos recursos empregados.

Ao serem seleccionados os serviços 3, 4 e 7, tem-se uma solução com um custo igual a 28 e nenhuma sobreposição de serviços.

Ao serem seleccionados os serviços 6 e 7, tem-se uma solução com um custo igual a 22 e nenhuma sobreposição de serviços. Todavia, pode-se constatar que o quarto tramo não é coberto nessa solução. Diz-se então que ocorreu um *leftover*.

Embora o modelo SPP seja o mais adequado para solucionar o problema de programação de tripulações, não existe garantia da obtenção de uma solução viável, pois as restrições de igualdade podem impedir que haja uma solução em que todas as linhas da matriz A sejam cobertas sem interseções de colunas

Sejam dados uma matriz A , de ordem $m \times n$, cujos elementos $a_{ij} \in \{0,1\}$; um vector de custos \mathbf{c} , com n elementos, em que cada componente representa o custo associado à coluna j da matriz A ; e x_j uma variável de decisão binária que é igual a 1 se o serviço está na solução, sendo 0 caso contrário; o modelo matemático para o CSP, baseado no SCP, é apresentado a seguir:

$$\text{Minimizar} \quad \sum_{j=1}^n c_j x_j \quad (2.14)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad \forall i = 1, \dots, m. \quad (2.15)$$

$$x_j \in \{0,1\} \quad \forall j = 1, \dots, n. \quad (2.16)$$

A função objectivo representada pela equação (2.14) prima por cobrir os tramos com os serviços disponíveis ao menor custo possível. As restrições (2.15) garantem o cobrimento de todas as linhas da matriz A . As restrições (2.16) resultam do carácter binário das variáveis de decisão. Caso as restrições (2.15) sejam convertidas para uma relação de igualdade, tem-se o modelo SPP.

De acordo com Park e Ryu (2006), ao invés de se adoptar um modelo SCP para a optimização de serviços de pessoal tripulante, pode-se usar um modelo de máxima cobertura (*Maximal Covering Problem – MCP*) para encontrar um subconjunto de serviços que cubram a maior quantidade de tramos possível. Segundo a experiência destes autores, a abordagem por meio de um MCP deixa muito menos tramos descobertos do que um modelo SCP.

O MCP pode ser formulado matematicamente como segue:

$$\text{Minimizar} \quad \sum_{i=1}^m y_i \quad (2.17)$$

Sujeito a:

$$\sum_{j=1}^n x_j = d \quad (2.18)$$

$$\sum_{j=1}^n a_{ij}x_j + y_i \geq 1 \quad \forall i = 1, \dots, m. \quad (2.19)$$

$$x_j \in \{0,1\} \quad \forall j = 1, \dots, n. \quad (2.20)$$

$$y_i \in \{0,1\} \quad \forall i = 1, \dots, m. \quad (2.21)$$

A função objectivo representada pela expressão (2.17) consiste na minimização dos tramos a serem descobertos. O conjunto de restrições (2.18) impõe que um número pré-determinado de serviços d seja seleccionado na solução. As restrições (2.19) garantem o cobrimento de todas as linhas da matriz A . Os conjuntos de restrições (2.20) e (2.21) impõem o carácter binário das variáveis de decisão. Nota-se que o modelo acima ainda permite que haja a sobreposição de serviços ao cobrir um tramo (*overcovers*).

Para solucionar este problema, Dias *et al.* (2002) propõem um modelo de particionamento de conjunto relaxado (*Relaxed Set Partitioning Problem – RSPP*) no qual a restrição de cobrimento de todas as linhas da matriz A é relaxada (ou seja, *leftovers* são permitidos). Deste modo, pode-se garantir a geração de soluções viáveis, sem a ocorrência de *overcovers*.

O modelo RSPP pode ser formulado como segue:

$$\text{Minimizar} \quad \sum_{j=1}^n c_j x_j + \sum_{i=1}^m z_i y_i \quad (2.22)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j + y_i = 1 \quad \forall i = 1, \dots, m. \quad (2.23)$$

$$x_j \in \{0,1\} \quad \forall j = 1, \dots, n. \quad (2.24)$$

$$y_i \in \{0,1\} \quad \forall i = 1, \dots, m. \quad (2.25)$$

A função objectivo representada pela expressão (2.22) prima por cobrir o maior número de viagens, com os serviços gerados, ao menor custo possível. Caso uma viagem não possa ser coberta, o modelo permite que a solução seja viável, mas a penaliza severamente (factor de penalidade z_i associado ao tramo descoberto y_i). As restrições (2.23) impõem o cobrimento das linhas da matriz A . As restrições (2.24) e (2.25) garantem o carácter binário das variáveis de decisão.

De acordo com Freling *et al.* (1999b), o CSP possui similaridades ao VSP; contudo, é muito mais complexo devido à gama de restrições envolvidas, geralmente relacionadas ao regime de trabalho do pessoal tripulante. Em linhas gerais, tais restrições relacionam-se ao tempo das tarefas, à localização das actividades e ao tipo de serviços.

Também se deve salientar a natureza multiobjectivo do problema. Como exemplos de objectivos para o CSP, podem ser citados: minimizar o custo dos serviços seleccionados; minimizar o número de tramos descobertos (*leftovers*); minimizar o número de serviços na solução; minimizar o número de trocas de motoristas por veículo (*changeovers*); minimizar a duração dos *leftovers* e minimizar os tempos ociosos. Uma descrição detalhada dos múltiplos objectivos para o CSP é feita no Capítulo 5 da presente Tese.

Deste modo, algoritmos que visem a construção de conjuntos de soluções de Pareto também devem ser desenvolvidos para uma resolução mais útil do problema (DIAS, 2005). Lourenço *et al.* (2000) e Dias (2005) apresentam metaheurísticas para a programação de tripulações envolvendo múltiplos objectivos.

2.2.3 Sistemas de Apoio à Decisão em empresas de transporte público

Diante do exposto, pode-se ressaltar que programar veículos e tripulações são dois processos extremamente complexos que, se realizados manualmente, provavelmente incorrerão em soluções sub-ótimas e numa elevada quantidade de tempo de planeamento.

Devido a este facto, existem diversas aplicações informáticas comerciais para apoiar o planeamento de sistemas de transporte público. A variedade de Sistemas de Apoio à Decisão (SAD) desenvolvidos para este fim é muito grande pois, para cada país ou região, a legislação de trabalho e os padrões de nível de serviço podem mudar drasticamente, fazendo que a concepção de um SAD geral seja extremamente difícil.

Em seguida, serão referidos os sistemas considerados pelo autor como os mais interessantes. Para o conhecimento de outras aplicações informáticas comerciais, o leitor pode consultar Ceder (2002).

Wren (2004) apresenta uma retrospectiva histórica dos 40 anos de experiência da Universidade de Leeds, no Reino Unido, na concepção deste tipo de *software* para apoiar o planeamento de sistemas de transporte público.

No período de 1961 a 1964, o primeiro *software* para a programação de veículos foi desenvolvido em um consórcio entre a Universidade de Leeds e a *British Railways*. No período de 1967 a 1978, foram desenvolvidas ferramentas para o *bus driver scheduling*, as quais resultaram no SAD conhecido como TRACS. No período de 1970 a 1991, bem como após 1994, foram dirigidos esforços de investigação na resolução do *bus scheduling*, os quais deram origem aos sistemas VAMPIRES, TASC e BOOST. Após a década de 80, com os aprofundamentos do conhecimento nas áreas de computação e programação linear inteira mista, foram desenvolvidos os sistemas BUSMAN e IMPACS (SMITH, 1988; CHAMBERLAIN e WREN, 1992).

A Universidade de Montreal, no Canadá, realizou, desde a década de 80, diversos estudos na área de optimização de sistemas de transporte público, os quais consubstanciaram na empresa GIRO (www.giro.com). GIRO, em cooperação com o Centro de Pesquisa em Transportes da Universidade de Montreal, desenvolveram o sistema HASTUS, que é um dos mais utilizados do mundo (LESSARD *et al.*, 1981; BLAIS *et al.*, 1990).

Em Portugal, a ferramenta Gestão Integrada de Sistemas de Transporte (GIST) foi desenvolvida por um consórcio entre grupos de Investigação e Desenvolvimento

(I&D) de duas universidades e cinco companhias de transporte público. Sousa *et al.* (2000) apresentam a ferramenta GIST, enquanto Dias *et al.* (2001) apresentam uma retrospectiva dos 10 anos de experiência da implantação do GIST em companhias de transporte público.

Na Itália, foi desenvolvido o sistema TURNI (www.turni.it) para a programação de tripulações pode ser ressaltado. Kroon *et al.* (2004) receberam o “*Daniel H. Wagner Prize for Excellence in Operations Research Practice*” em 2004, devido a uma aplicação do TURNI na empresa holandesa de transporte ferroviário *NS Rezigers*. A aplicação consistiu na programação de mais de 6500 motoristas, a qual incorreu em redução de custos anual da ordem de 7 milhões de dólares.

Na Nova Zelândia, pode-se destacar o sistema EXPRESS, o qual consiste em um *software* para a programação de motoristas em sistemas de transporte público por autocarros (FALKNER e RYAN, 1992).

O Brasil, apesar de ser um país de dimensões continentais e ter grandes cidades com sistemas de transportes públicos extremamente complexos, possui poucos avanços no que tange ao desenvolvimento de SADs para o sector em foco. Existem diversos trabalhos que remontam ao desenvolvimento de estudos, oriundos do estabelecimento de parcerias entre universidades e companhias de transporte público, para resolução de problemas de programação de veículos e tripulações, notadamente no transporte por autocarros (AZEVEDO FILHO *et al.*, 1994; RODRIGUES *et al.*, 2006, SILVA *et al.*, 2006; ATZINGEN *et al.*, 2007; SOUZA, 2010); todavia, ainda não foi consolidada uma ferramenta comercial. Um dos poucos sistemas disponíveis é o WPLEX (www.wplex.com.br).

2.3 Programação integrada de veículos e motoristas

2.3.1 Considerações iniciais

Tradicionalmente, a programação de veículos e de motoristas é realizado de uma maneira sequencial, de modo que a programação de veículos (VSP) é uma etapa inicial, sendo completada pela programação das tripulações (CSP). O CSP é mais restrito que o VSP, que, por sua vez, é mais flexível. Assim, escalonar veículos sem considerar as tripulações, que constituem o gargalo do processo poderá não ser a abordagem mais interessante.

Diversos investigadores têm relatado a forte interação entre estes dois problemas e os ganhos em considerá-los conjuntamente (BODIN *et al.*, 1983; FRIBERG e HAASE, 1999; GAFFI e NONATO, 1999; FRELING *et al.*, 1999b). Tal problema é denominado *Vehicle and Crew Scheduling Problem* – VCSP.

Steizen *et al.* (2007) definem o VCSP da seguinte forma: para um horizonte fixo de planeamento, uma tabela de horários e uma frota de veículos, que podem ser alocados em conjuntos de depósitos, o VCSP consiste em obter conjuntos de blocos de veículos e serviços de tripulantes de tal modo que as programações dos veículos e dos tripulantes sejam viáveis e mutuamente compatíveis.

Conforme Freling *et al.* (1999b), os benefícios da abordagem integrada para o VCSP são maiores nos seguintes casos: número restrito de trocas de motoristas por veículo; tempo elevado em que uma tripulação é utilizada no veículo; permanências obrigatórias dos tripulantes enquanto os veículos estão à espera; dominação dos custos das tripulações sobre os custos dos veículos e ocorrência da rendição dos motoristas somente nos depósitos. A seguir, serão explicadas as razões pelas quais a abordagem integrada é pertinente nestes casos.

Quando o número de trocas de motoristas por veículo é restrito, ou mesmo proibido, a viatura e o tripulante tendem a se comportar como um único recurso. Assim, programar veículos e motoristas de forma integrada representa uma melhor utilização dos mesmos.

Se o tempo em que uma tripulação opera uma viatura é elevado, como ocorre, por exemplo, no transporte interurbano, a realização de trocas de motoristas se mostra mais difícil. No transporte interurbano, como o motorista passa grande parte ou a totalidade da sua jornada de trabalho para realizar uma viagem, ou mesmo parte dela, não se mostra atrativo, ou mesmo viável, a utilização de *changeovers*. Logo, existe uma forte interação do tripulante com o veículo que ele ocupa, beneficiando o uso de uma programação integrada de tais recursos.

As trocas de motoristas podem ocorrer nas folgas entre duas viagens consecutivas. Porém, caso se imponha que os tripulantes permaneçam no veículo nesse período, perde-se grande parte das oportunidades de trocas, justificando uma programação integrada.

Quando o custo com as tripulações sobrepuja acentuadamente o custo com as viaturas, estas passam a ter uma importância secundária na programação, devendo se adequar à programação dos tripulantes. A flexibilização da programação de veículos em

favor da programação dos motoristas significa uma interacção mais forte entre estes dois recursos, beneficiando a abordagem integrada.

Caso as rendições dos tripulantes só possam ocorrer nos depósitos, não podem ser realizadas trocas de motoristas ao longo da operação da linha, restringindo as trocas a ocorrerem nos depósitos. Deste modo, os motoristas tendem a guiar apenas um veículo em sua jornada de trabalho, de modo que veículo e tripulante tendem a se comportar como um único recurso. Tal interacção facilita a abordagem integrada, conforme discutido anteriormente.

Freling *et al.* (2003) afirmam que existem três abordagens para a programação de viaturas e de tripulantes: a abordagem tradicional, na qual se resolve primeiro o VSP e em seguida o CSP; a abordagem independente, na qual os dois problemas são solucionados separadamente, sem a preocupação dos impactos mútuos entre as duas programações; e a abordagem integrada, na qual os dois problemas são resolvidos simultaneamente. Esta última abordagem será enfocada pelo presente trabalho.

Bodin *et al.* (1983) criticam severamente a estratégia de decomposição entre a programação de veículos e motoristas aplicada na abordagem sequencial. Os autores salientam que, no caso do transporte aéreo, em que os custos dos veículos são preponderantes diante dos custos com tripulações, tal decomposição ainda é justificável. Todavia, no caso do transporte público em autocarros e comboios, casos nos quais os custos com tripulantes representam a maior parte dos custos operacionais, programar veículos e tripulantes separadamente é uma abordagem pouco eficiente.

2.3.2 Modelos matemáticos para o VCSP

Nesta secção serão apresentadas algumas formulações matemáticas para o VCSP. Não é objectivo da secção ser exaustiva neste levantamento, mas apresentar diferentes tipos de modelos, os quais podem elucidar o leitor sobre a complexidade do problema. Mais informações sobre formulações matemáticas para o VCSP podem ser obtidas em Friberg e Haase (1999), Haase *et al.* (2001), Fischetti *et al.*, (2001), Freling *et al.* (2003), Huisman (2004) e Weider (2007).

2.3.2.1 VCSP com um único depósito

Um exemplo de modelo matemático para o VCSP com um único depósito é a formulação proposta por Freling *et al.* (1999b), a qual é baseada nos seguintes pressupostos:

- O planeamento dos veículos corresponde ao SDVSP conforme definido previamente, ou seja, um depósito, veículos idênticos, tempos de início e fim das viagens fixos e nenhuma restrição de tempo.
- A função custo para o VCSP é simplesmente o somatório das funções de custo do VSP e CSP definidas previamente. O objectivo primário do VSP é minimizar o número de veículos enquanto o objectivo primordial do CSP é minimizar o número de tripulantes.
- Um tramo (*piece-of-work*) é definido como uma sequência de tarefas em um bloco de veículo que podem ser realizadas por uma tripulação sem interrupção. Esta sequência de tarefas é somente restrita pela sua duração a qual deve atender certos limites de tempo.
- É feita distinção entre as viagens que correspondem ao cumprimento de tarefas e as viagens em vazio.

O VCSP contém uma formulação do SDVSP, definida sobre um grafo $G=(N, A)$, onde N é o conjunto de vértices incluindo as origens das viagens, os destinos das viagens e os depósitos; A o conjunto de arcos compreendendo as viagens e as viagens em vazio. Seja c_{ij} o custo do arco $(i, j) \in A$ e y_{ij} uma variável de decisão binária que é igual a 1 se o veículo cobre a viagem j exactamente após a viagem i , sendo 0 caso contrário.

Seja K o conjunto de todos os serviços viáveis e $K(p)$ o conjunto de serviços cobrindo a viagem produtiva $p \in I_1$ ou viagens em vazio $p \in I_2$, em que $I_2(i,j)$ representa o conjunto de tarefas relacionadas a viagens em vazio $(i,j) \in A$.

O modelo matemático é dado por:

[VCSP₁]

$$\text{Minimizar} \quad \sum_{(i,j) \in A} c_{ij} y_{ij} + \sum_{k \in K} d_k x_k \quad (2.26)$$

Sujeito a:

$$\sum_{j:(i,j) \in A} y_{ij} = 1 \quad \forall i \in N \quad (2.27)$$

$$\sum_{i:(i,j) \in A} y_{ij} = 1 \quad \forall j \in N \quad (2.28)$$

$$\sum_{k \in K(q)} x_k = 1 \quad \forall q \in I_1 \quad (2.29)$$

$$\sum_{k \in K(q)} x_k - y_{ij} = 0 \quad \forall (i, j) \in A, \forall q \in I_2(i, j) \quad (2.30)$$

$$x_k, y_{ij} \in \{0, 1\} \quad \forall k \in K, \forall (i, j) \in A \quad (2.31)$$

A função objectivo (2.26) a ser minimizada considera a soma total dos custos de designação de veículos e de pessoal tripulante. As restrições (2.27) e (2.28) garantem que cada viagem será atendida por apenas um veículo. O conjunto de restrições (2.29) impõem que cada tarefa seja coberta por um serviço. O conjunto de restrições (2.30) garante que as viagens em vazio serão consideradas na solução. Por fim, as restrições (2.31) estão relacionadas ao carácter binário das variáveis de decisão.

Huisman (2004) apresenta uma formulação mais compacta para o VCSP com um único depósito, no caso em que não são permitidas mudanças de tripulantes entre veículos (*changeovers*). Nesse caso, pode-se definir um custo do serviço que combina os custos com o veículo e pessoal tripulante.

No caso em que os *changeovers* são permitidos, viaturas e tripulantes se comportam como dois recursos distintos; contudo, quando os *changeovers* são proibidos, viaturas e tripulantes constituem um único recurso. Deste modo, pode-se adoptar um custo ao par veículo-tripulante, bem como reduzir a quantidade de variáveis e restrições.

Diante do exposto, pode-se utilizar um modelo de cobertura de conjuntos, formulado como segue:

[VCSP₂]

$$\text{Minimizar} \quad \sum_{j=1}^n g_j x_j \quad (2.32)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad \forall i = 1, \dots, m. \quad (2.33)$$

$$x_j \in \{0, 1\} \quad \forall j = 1, \dots, n. \quad (2.34)$$

O modelo acima é análogo ao modelo apresentado nas equações (2.14), (2.15) e (2.16), diferindo deste apenas pela consideração do custo combinado g , que contempla veículos e tripulações.

Se o VCSP, no caso de *changeovers* não serem permitidos, se comporta basicamente como um CSP, então também podem ser utilizados os modelos de máxima cobertura, particionamento de conjuntos ou particionamento de conjuntos relaxado apresentados na secção 2.2.2 deste capítulo.

Freling *et al.* (2003) apresentam diversos modelos, relaxações e algoritmos para a abordagem integrada. Os autores comparam as abordagens sequencial, integrada e o problema designado como *Independent Crew Scheduling Problem* (ICSP), o qual considera apenas a programação das tripulações e ignora a programação dos veículos, considerando a permissão ou proibição de *changeovers*. Os autores constatam que os benefícios da integração são maiores, em relação às outras duas abordagens, quando *changeovers* não são permitidos.

2.3.2.2 VCSP com múltiplos depósitos

Gaffi e Nonato (1999) apresentam outra formulação para o VCSP, voltada para o transporte público intermunicipal (*ex-urban mass transit*). Os autores ressaltam que a abordagem integrada para a programação de veículos e de pessoal tripulante é ainda mais importante para o transporte público intermunicipal do que para o transporte público urbano, devido à necessidade de um melhor aproveitamento dos motoristas, dados os elevados custos de mão-de-obra e a dispersão geográfica dos depósitos.

Seja V o conjunto de viagens, DH o conjunto de viagens em vazio, D o conjunto de depósitos com capacidade, e T o conjunto de tipos de jornadas (*runs*). Considerando as seguintes variáveis: x_j a variável de decisão associada com a tarefa (*run*) j , a um custo c_j ; z_i a variável de decisão relacionada com o serviço (*duty*) i com custo a_i e y_{uv} a variável que representa uma viagem em vazio entre u e v , a um custo b_{uv} . Pode-se, assim, estabelecer o seguinte modelo matemático para o VCSP (GAFFI e NONATO, 1999):

[VCSP₃]

$$\text{Minimizar} \quad \sum_{j \in J} c_j x_j + \sum_{i \in I} a_i z_i + \sum_{(u,v) \in DH} b_{uv} y_{uv} \quad (2.35)$$

Sujeito a:

$$\sum_{u \in V \cup D} y_{vu} = 1 \quad \forall v \in V \quad (2.36)$$

$$\sum_{u \in V \cup D} y_{uv} = 1 \quad \forall v \in V \quad (2.37)$$

$$\sum_{j \in J(u,v)} x_j = y_{uv} \quad \forall (u,v) \in DH \quad (2.38)$$

$$\sum_{i \in I(u,v)} z_i = y_{uv} \quad \forall (u,v) \in DH \quad (2.39)$$

$$\sum_{j \in J(u,v)} x_j = 1 \quad \forall v \in V \quad (2.40)$$

$$\sum_{i \in I(u,v)} z_i = 1 \quad \forall v \in V \quad (2.41)$$

$$\sum_{i \in I_k} z_i \leq C^k \quad \forall d_k \in D \quad (2.42)$$

$$\sum_{j \in J(t)} x_j \leq R^t \quad \forall t \in T \quad (2.43)$$

$$x_j, z_i, y_{uv}, y_{vu} \in \{0,1\} \quad \forall j \in J, \forall i \in I, \forall (u,v) \in DH. \quad (2.44)$$

A expressão (2.35) é a função objectivo a ser minimizada, a qual representa os custos de afectação dos veículos, pessoal tripulante e as viagens em vazio. As restrições (2.36) e (2.37) impõem que um motorista conduza um veículo entre duas viagens consecutivas. Os conjuntos de restrições (2.38), (2.39), (2.40) e (2.41) são destinadas a garantir que todos os serviços são cobertos. As restrições (2.42) modelam a capacidade de cada depósito. As restrições (2.43) limitam o número máximo de tipos de jornadas. Por fim, as restrições (2.44) relacionam-se à natureza binária das variáveis de decisão.

Com relação ao modelo acima, dois aspectos podem ser salientados. Em primeiro lugar, a formulação aplica-se a um problema envolvendo múltiplos depósitos. Em segundo lugar, os conjuntos de restrições (2.40) e (2.41) acabam por tornar-se redundantes diante das restrições (2.36) e (2.37) e podem ser removidos do modelo.

2.3.2.3 Considerações sobre as formulações

Diante das formulações apresentadas, deve-se destacar que são dois os aspectos mais importantes para a utilização de um dado modelo: permissão ou não *changeovers* e consideração de um ou múltiplos depósitos.

Quando os *changeovers* não são permitidos, viatura e tripulante podem ser modelados como um único recurso, visto que as trocas de motoristas por veículo são proibidas. Deste modo, o VCSP se reduz ao CSP, diminuindo o porte do modelo matemático empregado para sua representação.

No que se refere à quantidade de depósitos, o VCSP pode ser modelado considerando um único depósito ou múltiplos depósitos. Tais modelos basicamente agregam as formulações do VSP e do CSP em um único modelo matemático.

Como visto na secção 2.2.1, no SDVSP todos os veículos são alojados em um único depósito, enquanto no MDVSP, os veículos são alojados em depósitos distintos, facto que aumenta o número de soluções possíveis e, por conseguinte, a complexidade do problema.

O SDVSP possui algoritmos polinomiais para a sua resolução, enquanto o MDVSP é NP-difícil. Mesmo no caso em que o VCSP modela um SDVSP, os modelos matemáticos não conseguem fornecer soluções óptimas em tempo computacional aceitável, como reportam Friberg e Haase (1999) e Freling *et al.* (1999b).

Percebe-se que o uso de modelos matemáticos para a resolução exacta do VCSP ainda não se mostra viável. Assim, ao se trabalhar com tais formulações, usualmente se aplicam métodos tais como relaxação lagrangeana e geração de colunas para a obtenção de soluções, como será descrito na secção seguinte.

No caso em que o VCSP modela um MDVSP, a complexidade do modelo matemático é ainda maior. Groot e Huisman (2008) trabalham com a decomposição do MDVSP, de modo a obter soluções de qualidade em tempo útil.

2.3.3 Abordagens para o VCSP

Tendo em vista a complexidade do VCSP, para instâncias reais de médio e grande porte, ainda não é possível obter a solução exata para o problema; deste modo, abordagens heurísticas vêm sendo propostas na literatura para sua resolução. A seguir, é feita uma revisão bibliográfica acerca dos trabalhos que apresentam abordagens para resolução do VCSP.

2.3.3.1 Abordagens baseadas em métodos exactos

Freling *et al.* (1999b) apresentam um modelo matemático para o *Single Depot Vehicle and Crew Scheduling* (vide equações (2.26) a (2.31) no presente capítulo) e testam a abordagem em cinco instâncias reais do sistema de transporte por autocarros em Roterdão, Holanda. Na solução de três das cinco instâncias analisadas, foi possível obter uma redução da soma de veículos e de motoristas, comparando-se as abordagens integrada e sequencial. Para resolução do modelo, foram aplicadas uma heurística de relaxação lagrangeana e a técnica de geração de colunas.

Friberg e Haase (1999) apresentam um modelo matemático para o VCSP com um único depósito. Os autores avaliaram o modelo proposto em um conjunto de 10 instâncias teóricas, de pequeno porte (número de viagens entre 10 e 30), as quais foram geradas aleatoriamente. Para resolver o modelo, foi utilizado um algoritmo de geração de colunas. Aplicou-se uma relaxação linear ao problema de particionamento de conjuntos e, em seguida, aplicou-se um algoritmo *branch-and-bound* para obter a solução ótima do modelo. Para melhorar os limites inferiores, foram utilizadas algumas desigualdades válidas. Os autores constataram que a abordagem proposta só obtém as soluções ótimas para instâncias com até 20 viagens.

Gaffi e Nonato (1999) apresentam uma abordagem para o VCSP no caso do transporte público intermunicipal (vide equações (2.35) a (2.44) no presente capítulo). O modelo matemático proposto é o primeiro a considerar a possibilidade de múltiplos depósitos. A abordagem é testada em dois conjuntos de dados: um com oito instâncias típicas de transporte intermunicipal, e outro com seis instâncias de transporte urbano e suburbano. Os conjuntos de dados foram obtidos por meio do *software* MTRAM (www.maior.it), utilizado por diversas cidades na Itália. A resolução do modelo é baseada em uma heurística lagrangeana. Os autores constataram melhorias da abordagem integrada em relação à abordagem sequencial, nos conjuntos de dados analisados.

Haase *et al.* (2001) apresentam um modelo matemático para a integração de veículos e motoristas em sistemas de transporte público, considerando uma frota de veículos homogênea e um único depósito. Para resolução do modelo, é proposta uma heurística de geração de colunas (somente para a programação dos motoristas) integrada com um *branch-and-bound*. Avaliando o desempenho da abordagem proposta para um

conjunto de 10 instâncias geradas aleatoriamente, os autores constataram que a proposta apresentou bons resultados, mesmo para instâncias de grande porte.

Freling *et al.* (2001b) apresentam um caso de estudo no qual ilustram uma comparação entre as abordagens sequencial e integrada para a programação de veículos e tripulações. O caso de estudo foi efectuado em instâncias reais advindas do sistema de transporte público por autocarros de Roterdão. Os resultados da abordagem integrada superaram os resultados da abordagem sequencial em uma faixa de 2% a 8%, para os conjuntos de dados analisados.

Fischetti *et al.* (2001) propõem um modelo de programação linear binária para o problema da programação integrada de veículos e tripulantes. O modelo é aplicável ao caso em que múltiplos depósitos estão disponíveis e em que tanto o tempo de início e término de um serviço quanto o tempo total do mesmo, são impostos. Uma heurística de separação, em conjunto com um algoritmo exacto do tipo *branch-and-cut*, foram aplicados em instâncias aleatórias disponíveis na literatura, oferecendo resultados competitivos.

Huisman (2004) apresenta diversas abordagens para a programação integrado de veículos e motoristas, para os casos com um ou múltiplos depósitos. Para o caso envolvendo múltiplos depósitos, apresenta diversos modelos matemáticos e abordagens de solução, constituídas por algoritmos que combinam uma heurística lagrangeana com técnicas de geração de colunas. O autor soluciona diversas instâncias teóricas e reais, obtendo excelentes resultados. Por fim, apresenta a programação dinâmica² de veículos e motoristas (*Dynamic Vehicle and Crew Scheduling*), dissertando sobre os benefícios de tal abordagem e apresentando modelos matemáticos e métodos para resolução do problema.

Huisman *et al.* (2005) apresentam dois modelos diferentes para o VCSP com múltiplos depósitos, cujas abordagens de solução são baseadas em uma combinação entre geração de colunas e relaxação lagrangeana. Os autores apresentam experimentos computacionais em 8 instâncias reais de uma companhia de transporte público holandesa, bem como uma nova metodologia para gerar instâncias aleatórias para o problema.

² Tradicionalmente, consideram-se as tabelas de horário com tempos determinísticos. Na prática, podem ocorrer variações aleatórias nos tempos das viagens (adiantamentos e atrasos). A programação de sistemas de transportes colectivos que contempla esta aleatoriedade é dita dinâmica.

Weider (2007) apresenta o algoritmo IS-OPT, o qual integra a programação de veículos e serviços no planeamento de sistemas de transporte público por autocarros. Este algoritmo combina uma série de técnicas, a saber: relaxação lagrangeana, geração de colunas, método aproximativo (*proximal bundle method*), o qual é utilizado para minimizar uma função irrestrita, contínua, convexa e possivelmente não sinuosa) e *rapid branching* (uma nova variante heurística do algoritmo *branch-and-bound*). As programações geradas pelo IS-OPT incorreram na redução de mais de 5% dos custos das programações existentes das companhias de transporte público urbano e regional estudadas.

Fleurent e Rosseuau (2007) apresentam resultados da aplicação prática dos sistemas HASTUS, MINBUS e CREWPLAN na integração de veículos e motoristas. Casos de estudos nas cidades de Minneapolis (EUA), Montpellier (França), Hamburgo (Alemanha) e Colónia (Alemanha) demonstraram que a abordagem integrada incorreu em expressivos ganhos de produtividade.

Groot e Huisman (2008) solucionam instâncias reais de grande porte, de uma empresa de transporte público holandesa. Os autores desenvolvem uma metodologia para divisão de um problema de grande porte em subproblemas menores, os quais podem ser solucionados separadamente e levam à resolução do problema inicial. Foi demonstrado que o efeito da divisão das instâncias não deteriorou substancialmente a qualidade das soluções geradas. Além disso, foi evidenciada a redução de custos da abordagem integrada em relação à abordagem sequencial.

Mesquita e Paias (2008) apresentam uma abordagem para o VCSP baseada em modelos de cobertura e particionamento de conjuntos. A abordagem consiste nas seguintes etapas: geração de um conjunto inicial de serviços viáveis, relaxação do modelo matemático por meio de geração de colunas e aplicação da técnica *Branch-and-Bound* para obtenção de soluções viáveis para o VCSP. Em testes computacionais realizados com as instâncias de Huisman *et al.* (2005), a abordagem proposta mostrou ter o melhor desempenho da literatura, tanto em qualidade das soluções quanto em tempo computacional.

2.3.3.2 Abordagens baseadas em técnicas heurísticas

Tendo em vista a complexidade matemática e computacional do VCSP, uma das abordagens mais presentes na literatura é o uso de métodos heurísticos. As heurísticas e as metaheurísticas consistem em métodos aproximativos, que buscam a obtenção de

soluções de boa qualidade, não necessariamente ótimas, em tempo computacional aceitável. Nesse sentido, estas soluções são satisfatórias e podem ser usadas no cotidiano das empresas de transporte público.

Conforme Reeves e Beasley (1995), uma heurística é uma técnica que pesquisa boas soluções (ou seja, próximas do ótimo) com um custo computacional razoável, sem estar apta a garantir a optimalidade, ou a distância à optimalidade de uma dada solução.

De acordo com Blum e Roli (2003), uma metaheurística é um conjunto de conceitos os quais podem ser usados para definir métodos heurísticos que podem ser aplicados a uma ampla gama de diferentes problemas. Em outras palavras, uma metaheurística pode ser vista como um algoritmo genérico o qual pode ser aplicado em diferentes problemas de otimização com relativamente poucas modificações a serem realizadas na adaptação a um problema específico.

Ball *et al.* (1983) são os primeiros autores a propor um procedimento para programar veículos e tripulações simultaneamente. O algoritmo proposto consiste em compor um grafo de programação, particionando os nós em diferentes níveis. Em seguida, usa-se iterativamente um algoritmo de emparelhamento (*matching*) para compor os serviços.

Falkner e Ryan (1992) apresentam o caso do sistema EXPRESS, utilizado na cidade de Christchurch, Nova Zelândia. A programação dos motoristas inicia-se por uma programação inicial dos veículos, e desenvolve-se ao longo de um processo iterativo. Nota-se uma preocupação com a integração; porém a abordagem é centrada na programação dos motoristas.

Patrikalakis e Xerocostas (1992) apresentam um novo esquema de decomposição para o problema de planeamento em sistemas de transporte público: ao invés de solucionar a programação dos veículos antes da dos tripulantes, esta ordem é invertida. A programação é efectuada em dois estágios: em um primeiro estágio, é feita a afectação dos motoristas por meio de um modelo de cobertura de conjuntos, e, em seguida, em um segundo estágio, a programação de veículos e motoristas é completada por meio de um modelo de fluxo em redes.

Azevedo Filho *et al.* (1994) relatam os resultados de um estudo de afectação de veículos e tripulações em linhas de autocarros da cidade de Fortaleza, Nordeste do Brasil, considerando as suas características e usando *softwares* desenvolvidos na Universidade de Leeds e um aplicativo desenvolvido na Universidade Federal do Ceará.

Os seguintes passos foram implementados neste caso: (i) criar um conjunto de linhas de acordo com a procura de passageiros observada; (ii) aplicar o *software* de programação de veículos; (iii) aplicar o *software* de programação de tripulações; e (iv) rever a programação dos veículos com relação à programação dos tripulantes. Neste estudo, foram conseguidas reduções de até 14%, em comparação com o método manual utilizado na prática.

Wren e Gualda (1999) apresentam experimentos com *softwares* desenvolvidos na Universidade de Leeds, para a integração da programação de veículos e tripulações em duas cidades brasileiras: Fortaleza e Sorocaba. Os autores reportam os experimentos realizados por Azevedo Filho *et al.* (1994) para Fortaleza e fazem um estudo semelhante para Sorocaba. Em ambos os casos, foi observada uma redução substancial dos custos, em termos de horas de tripulação pagas, advinda do uso da abordagem integrada. Deve-se observar que nestes estudos, as programações de viaturas e tripulantes eram feitas separadamente, para uma posterior correção das soluções. Além deste facto, as abordagens não contemplavam múltiplos objectivos.

Valouxis e Housos (2002) apresentam uma abordagem combinada para a programação de veículos e motoristas. Analisando o caso de companhias de transporte público na Grécia, em que ocorrem práticas tais como programações diárias não fixas e a restrição à *changeovers*, os autores propõem um algoritmo, denominado CGQS (*Column Generation Quick Shifts*), o qual soluciona um problema de programação linear inteira. A heurística *Quick Shifts* (QS) trabalha com problemas de emparelhamento, particionamento e caminho mais curto, com vista a construir serviços de mínimo custo. A abordagem proposta foi testada em seis instâncias advindas de problemas reais de sistemas de transporte público na Grécia, apresentando bons resultados.

Rodrigues *et al.* (2006) apresentam o desenvolvimento de uma ferramenta computacional para solucionar o VCSP em São Bernardo do Campo, na região metropolitana de São Paulo, Brasil. A abordagem proposta consiste em uma combinação de modelos de programação inteira e heurísticas, sendo composta por quatro módulos: obtenção do conjunto primário de tempos de início das viagens, construção dos blocos de veículos, construção da programação e heurística para espaçamento de partidas consecutivas. A abordagem proposta foi testada em 7 instâncias reais e incorreu em diversos benefícios em relação ao planeamento realizado na prática, o qual é efectuado manualmente.

Silva *et al.* (2006) apresentam uma abordagem que combina a metaheurística *Simulated Annealing* – SA (KIRKPATRICK *et al.*, 1983; CERNY, 1985) com a técnica de geração de colunas para a resolução do VCSP. Experimentos computacionais realizados em instâncias reais obtidas junto a uma empresa de transporte público de Belo Horizonte, Brasil, apontam para a potencialidade de redução de custos por intermédio da aplicação da metodologia proposta.

Laurent e Hao (2007) apresentam uma abordagem, baseada na meta-heurística SA, para solução do VCSP em uma aplicação real de uma empresa de transporte de *limousines*. Tal problema tem elevada complexidade, pois os veículos são bastante diferenciados devido aos acessórios (aparelhos de som e DVD, por exemplo) e os motoristas também são diferenciados de acordo com suas aptidões (fluência em uma determinada língua, por exemplo). A abordagem proposta se divide em duas etapas: inicialmente utiliza-se programação por restrições (*constraint programming*) para a geração de uma solução inicial, e, em seguida, tal solução é refinada por um algoritmo SA.

Estes autores consideram também múltiplos objetivos em sua estratégia de solução. Consideram um objetivo principal (minimização da duração total de viagens não atendidas) e quatro objetivos secundários (minimização do número de veículos e de motoristas em uso, minimização do número de actualizações (*upgrades*), minimização das viagens em vazio e minimização do tempo total de espera). É pertinente ressaltar que os autores atribuíram empiricamente pesos aos critérios envolvidos, não sendo praticada uma abordagem baseada na construção de um conjunto de soluções de Pareto.

Bartodziej *et al.* (2007) propõem três metaheurísticas para o problema de programação de veículos e de tripulações em uma empresa de transporte aéreo de cargas: *Simulated Annealing* (SA), *Great Deluge Algorithm* (GDA) e *Record-to-Record Travel* (RRT) (DUECK, 1993). A complexidade do problema reside nas janelas de tempo, na legislação trabalhista para os motoristas e na alta heterogeneidade da frota. Os algoritmos foram implementados em um Sistema de Apoio à Decisão que vem sendo utilizado no planeamento do sistema real.

Steinzen *et al.* (2007) apresentam um algoritmo evolucionário híbrido para o problema de programação de veículos e motoristas envolvendo múltiplos depósitos. A abordagem híbrida combina as técnicas de relaxação lagrangeana, geração de colunas e Algoritmos Genéticos (HOLLAND, 1975; GOLDBERG, 1989). Na resolução do problema, este é decomposto em três subproblemas: afectação de viagens a depósitos,

construção das escalas dos veículos, construção das escalas dos motoristas. O Algoritmo Genético usa uma heurística lagrangeana, baseada em geração de colunas, para calcular a aptidão dos indivíduos da população. Os autores realizaram testes em instâncias geradas aleatoriamente, disponíveis na internet, e obtiveram resultados competitivos em relação às outras abordagens presentes na literatura.

Steizen (2007) estuda a programação de veículos e tripulações em transporte intermunicipal com tabelas de horário irregulares (*Ex-Urban Vehicle and Crew Scheduling with Irregular Timetables*). Para resolução do problema, o autor utiliza esquemas de ramificação local (*local branching*) em conjunto com metaheurísticas multiobjectivo. A ideia básica da ramificação local é definir regiões desejáveis a serem eficientemente exploradas por um *solver* genérico de programação inteira mista. Os objetivos avaliados são custo e regularidade dos serviços, em instâncias aleatórias disponíveis para o problema de programação de veículos e motoristas com múltiplos depósitos. O autor salienta que tal abordagem pode prover o tomador de decisões de informação adicional, a qual pode enriquecer a qualidade das soluções geradas.

Atzingen *et al.* (2007) apresentam uma heurística *Variable Neighborhood Search* – VNS (MLADENOVIC e HANSEN, 1997) para o VCSP. Em experimentos computacionais realizados com dados reais advindos de uma empresa de autocarros de uma grande cidade brasileira, a abordagem integrada propiciou melhores resultados do que a abordagem sequencial.

Gomes *et al.* (2008) apresentam uma abordagem heurística para a afectação de veículos e tripulações em Fortaleza. A heurística divide a jornada de trabalho em dois turnos e efectua movimentos de trocas de viagens entre os blocos, com vista a minimizar a quantidade de horas-extras pagas. A heurística foi testada em 10 instâncias reais e a redução de horas-extras em relação à solução real encontrou-se no intervalo de 9,6% a 100%, apresentando uma redução média de 54,4%.

Laurent e Hao (2008) apresentam uma meta-heurística *Greedy Randomized Adaptive Search Procedures* – GRASP (FEO e RESENDE, 1989; FEO e RESENDE, 1995), para solução do VCSP em um estudo de caso em uma empresa de transporte rodoviário de passageiros de longo curso (*extra-urban transport*). Em sete conjuntos de dados advindos de um sistema real, os experimentos computacionais salientaram a dominância da abordagem integrada sobre a sequencial.

2.3.3.3 Considerações sobre técnicas heurísticas

Conforme discutido, o uso de métodos exactos para resolução do VCSP não permite a obtenção de soluções óptimas em tempos aceitáveis.

Por outro lado, o uso de heurísticas é adequado a um tipo de problema específico, não sendo uma abordagem genérica. As metaheurísticas são abordagens de carácter modular, visto que podem ser adaptadas mais facilmente a diversas classes de problemas.

Na revisão bibliográfica realizada, constatou-se que, no âmbito das metaheurísticas aplicadas ao VCSP, os algoritmos baseados em pesquisa de vizinhança (SA, VNS, GRASP, GDA, RRT) foram mais utilizados que os algoritmos evolucionários. A utilização destes algoritmos não foi devidamente justificada pelos autores. Entretanto, esta opção deve-se possivelmente, ao facto do espaço de pesquisa do VCSP ser bastante restrito, sendo difícil a tarefa de obtenção de soluções viáveis, tornando menos atrativa a alternativa de adoptar algoritmos populacionais.

As metaheurísticas apresentadas, em geral, não tiveram seu comportamento avaliado com relação a algum outro método ou marco de referência (*benchmark*). Assim, uma análise mais acurada da eficiência e eficácia de tais métodos é necessária. Uma excepção é o Algoritmo Genético híbrido de Steizen *et al.* (2007) que foi comparado com resultados presentes na literatura, mostrando ser competitivo.

2.4 Considerações finais

O presente capítulo apresentou uma visão geral sobre o problema integrado de programação de veículos e tripulações (*Vehicle and Crew Scheduling Problem – VCSP*). Para tanto, foram introduzidos conceitos sobre o planeamento de sistemas de transporte público e sobre as programações de veículos e tripulações isoladamente. Em seguida, foi apresentada uma revisão bibliográfica sobre modelos matemáticos para o VCSP, bem como de abordagens heurísticas para sua resolução.

A seguir, na Tabela 2, é feito um resumo dos casos referidos para o VCSP no presente capítulo, enfatizando os principais pressupostos adoptados em cada estudo.

Deve-se destacar que foi dado enfoque no caso de sistemas de transporte públicos urbanos e intermunicipais por meio de autocarros. Todavia, existe uma nova tendência de trabalhos voltados para a integração de veículos e motoristas no transporte aéreo. Para o leitor interessado, são recomendados os trabalhos: Stojkovic e Soumis

(2001), Klabjan *et al.* (2002), Yao *et al.* (2004), Mercier *et al.* (2005), Ramírez (2007), Mercier e Soumis (2007), Papadakos (2007) e Souai e Teghem (2008).

Tabela 2: Principais características dos casos de estudo citados na literatura.

Autor	Caso de estudo	Depósitos	Changeovers
Ball <i>et al.</i> (1983)	Baltimore (Estados Unidos)	Múltiplos	Não especifica nenhuma restrição
Falkner e Ryan (1992)	Christchurch (Nova Zelândia)	Único	Não especifica nenhuma restrição
Patrikalakis e Xerocostas (1992)	Atenas (Grécia)	Único	O motorista usualmente conduz 2 autocarros
Azevedo Filho <i>et al.</i> (1994)	Fortaleza (Brasil)	Múltiplos	Proibidos
Wren e Gualda (1999)	Sorocaba (Brasil)	Múltiplos	Permitidos
Freling <i>et al.</i> (1999b)	Rotterdam (Holanda)	Único	Permitidos
Gaffi e Nonato (1999)	Cidades italianas (não especificadas)	Múltiplos	Permitidos
Freling <i>et al.</i> (2001b)	Rotterdam (Holanda)	Único	Permitidos
Valouxis e Housos (2002)	Cidades na Grécia (não as especifica)	Múltiplos	Proibidos
Huisman (2004)	Rotterdam e outras cidades holandesas	Único e múltiplos	Permitidos
Rodrigues <i>et al.</i> (2006)	Regiões Metropolitanas de São Paulo e São Bernardo do Campo (Brasil)	Único	Não especifica nenhuma restrição
Silva <i>et al.</i> (2006)	Belo Horizonte (Brasil)	Único	Permitidos
Laurent e Hao (2007)	Paris (França)	Único	Permitidos
Bartodziej <i>et al.</i> (2007)	Companhia de Logística, na Alemanha	Único	Permitidos
Atzingen <i>et al.</i> (2007)	Grande metrópole brasileira (não especificada)	Único	Permitidos
Weider (2007)	Cidades alemãs	Único e múltiplos	Não especifica nenhuma restrição
Gomes <i>et al.</i> (2008)	Fortaleza (Brasil)	Múltiplos, sendo reduzido para um único (simplificação)	Proibidos
Laurent e Hao (2008)	Transporte intermunicipal, locais não especificados.	Único	Permitidos
Groot e Huisman (2008)	Rotterdam, Utrecht e Dordrecht	Múltiplos. Decomposição o problema.	Permitidos

A seguir, no Capítulo 3, será abordado o problema de máxima cobertura, com ênfase na proposta de uma variante do VCSP sem *changeovers*.

Capítulo 3

Uma variante do Problema de Máxima Cobertura para a programação de veículos e tripulantes sem *changeovers*

“O bom estrategista, para vencer uma batalha, faz antes muitos cálculos no seu templo, pois sabe que eles são a chave que o conduzirá a vitória. É calculando e analisando que o estrategista vence previamente a guerra na simulação feita no templo. Portanto, fazer muitos cálculos conduz à vitória, e poucos, à derrota.”

Sun Tzu

Resumo

Este capítulo tem como objectivo apresentar o Problema de Máxima Cobertura (*Maximal Covering Problem – MCP*) que consiste em um modelo de optimização combinatória bastante estudado, aplicado usualmente em problemas de localização de facilidades e *scheduling*. Na primeira secção, é apresentada uma introdução geral ao MCP, salientando-se como este modelo se relaciona com o VCSP. Na segunda secção, é apresentada uma formulação para o MCP que pode ser aplicada para a modelação do VCSP sem *changeovers*. Na terceira secção, são apresentadas algumas abordagens para a resolução do MCP, com ênfase nas abordagens heurísticas. Na quarta secção é proposta uma abordagem heurística GRASP para a resolução do MCP. Na quinta secção, são apresentados experimentos computacionais, com vista a comparar a heurística proposta com o *branch-and-bound*. Por fim, na sexta secção, são feitas algumas considerações sobre as abordagens adoptadas e os resultados obtidos.

3.1 Introdução ao Problema de Máxima Cobertura

O Problema de Máxima Cobertura (*Maximal Covering Problem – MCP*) é um problema de optimização combinatória amplamente estudado, com diversas aplicações,

tais como localização de facilidades, telecomunicações, *scheduling* em sistemas de transporte público, selecção de lista de clientes para o envio de mala direta, selecção de ferramentas em sistemas flexíveis de manufatura e modelagem do esquema da memória humana (CHURCH e ReVELLE, 1974; DOWNS e CAMM, 1996; RESENDE, 1998; ARAKAKI e LORENA, 2006; PARK e RYU, 2006).

Trata-se de um problema combinatório NP-difícil (ver referências anteriores). Deste modo, métodos aproximados, tais como heurísticas, são desejáveis para a obtenção de boas soluções em tempo computacional aceitável.

De acordo com Downs e Camm (1996), conceptualmente, o MCP é relacionado com o SCP, diferindo deste pela restrição do número de conjuntos que podem ser seleccionados. Enquanto o SCP requer a cobertura de toda a procura como o menor número de recursos, o MCP busca cobrir a máxima procura com um número restrito de recursos (BERMAN *et al.*, 2009).

Maximal Covering Problem

O MCP clássico é definido como um problema de maximização, no qual, às variáveis de decisão, são associados pesos. A formulação a seguir foi proposta por Church e ReVelle (1974).

Dada uma matriz A ($m \times n$), tal que $a_{ij} \in \{0,1\}$, o MCP consiste em cobrir³ a maior quantidade de linhas da matriz A com uma quantidade de colunas da matriz A igual um valor predefinido d . As variáveis y_i representam as linhas da matriz A , de modo que $y_i = 1$ se a i -ésima linha é coberta em uma solução, sendo $y_i = 0$ caso contrário. As variáveis x_j representam as colunas da matriz A , de modo que $x_j = 1$ se a j -ésima coluna faz parte da solução, sendo $x_j = 0$ caso contrário. A cada linha da matriz A associa-se um peso, o qual reflete o ganho em cobrir tal linha.

[MCP₁]

$$\text{Maximizar} \quad z = \sum_{i=1}^m w_i y_i \quad (3.1)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j \geq y_i \quad \forall i = 1, \dots, m. \quad (3.2)$$

³ Conforme ilustrado na secção 2.2.2, a i -ésima linha da matriz A é coberta pela j -ésima coluna se $A(i,j)=1$ e $x_j=1$.

$$\sum_{j=1}^n x_j = d \quad (3.3)$$

$$x_j \in \{0,1\} \quad \forall j= 1, \dots, n. \quad (3.4)$$

$$0 \leq y_i \leq 1 \quad \forall i= 1, \dots, m. \quad (3.5)$$

A expressão (3.1) é a função objectivo a ser maximizada, a qual representa o ganho em cobrir a procura. O conjunto de restrições (3.2) representa as relações de cobertura das variáveis y_i pelas variáveis x_j . A restrição (3.3) impõe que d colunas sejam utilizadas na solução. O conjunto de restrições (3.4) impõe o carácter binário das variáveis de decisão x_j . Neste caso, a integralidade das variáveis y_i pode ser relaxada.

Church e ReVelle (1974) formulam o MCP como um problema de minimização em que $h_i = 1 - y_i$. A cada variável h_i é associado um custo c_i , inerente ao não-cobertura da i -ésima linha de A . Deste modo, o objectivo é minimizar o número de linhas descobertas da matriz A (tal formulação poderia ser denotada como um problema de mínima não-cobertura).

A variante descrita do MCP pode ser formulada matematicamente como segue:

[MCP₂]

$$\text{Minimizar} \quad z = \sum_{i=1}^m c_i h_i \quad (3.6)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j + h_i \geq 1 \quad \forall i = 1, \dots, m. \quad (3.7)$$

$$\sum_{j=1}^n x_j = d \quad (3.8)$$

$$x_j \in \{0,1\} \quad \forall j= 1, \dots, n. \quad (3.9)$$

$$h_i \geq 0 \quad \forall i= 1, \dots, m. \quad (3.10)$$

Seja m o número de linhas a serem cobertas por um conjunto de d colunas, o número de soluções possíveis para o problema é dado por (CURTIN *et al.*, 2005):

$$\binom{m}{d} = \frac{m!}{d!(m-d)!}$$

Conforme Goldberg e Luna (2005), a formulação do MCP enquanto problema de minimização tem forte relação com o problema de p -medianas, o qual também é NP-difícil.

3.2 Formulação para o VCSP sem *changeovers*

Como visto no capítulo anterior, em problemas de planeamento operacional de sistemas de transporte público, abordagens baseadas em cobertura e particionamento de conjuntos são correntemente utilizadas.

Abordagens baseadas no SCP têm a vantagem de permitirem uma solução mais rápida; todavia, devido a poderem ocorrer *overcovers*⁴, tais soluções necessitam de correções para serem aplicadas na prática. Abordagens baseadas no SPP são de resolução mais difícil, porém, por não permitirem *overcovers*, são mais atractivas do ponto de vista prático.

Huisman (2004) salienta que, para o caso particular do VCSP em que *changeovers* não são permitidos, pode-se adoptar uma abordagem baseada no SCP. Por conseguinte, uma abordagem baseada no SPP também pode ser utilizada. Como o modelo baseado no problema de particionamento é mais desejado do ponto de vista prático, é pertinente analisá-lo em detalhe.

Conforme Klabjan *et al.* (2001), os principais motivos pelos quais o problema de *scheduling* de tripulações baseado no SPP torna-se difícil, são o grande número de serviços admissíveis, a sua estrutura complexa e a natureza de custos não-linear.

O número de serviços possíveis não está relacionado basicamente com o modelo matemático, mas às características práticas de um dado problema real.

Um dos principais motivos pelos quais a estrutura do SPP é complexa são as restrições de igualdade, as quais impõem que cada linha da matriz A seja coberta por apenas uma coluna. No entanto, esta é uma grande vantagem prática da abordagem, pois, deste modo, eliminam-se os *overcovers*.

A função objectivo do SPP consiste em um produto escalar de um vector de custos e um vector de variáveis de decisão. Logo, o problema tem uma estrutura de

⁴ Em alguns casos, os *overcovers* podem ser permitidos, pois representariam o *deadheading* de tripulantes.

custos *explícita* de carácter linear. Contudo, de um modo *implícito*, esta estrutura pode ser não-linear. Tal particularidade se evidencia em problemas do tipo *crew scheduling*.

Em problemas de planeamento operacional de transportes públicos, um serviço pode ser visto como a cobertura de um conjunto de tramos. O custo de um serviço está relacionado com o pagamento da tripulação. Devido às legislações trabalhistas em vigência na maioria das grandes cidades, os serviços de tripulações não são pagos por hora trabalhada, mas por jornada de trabalho. Deste modo, serviços que cubram quantidades de tramos diferentes podem vir a ter o mesmo custo. Por conseguinte, a relação entre o custo do serviço e a quantidade de tramos cobertos não é directamente proporcional. Em outras palavras, tal relação é não-linear.

Outros aspectos estão associados a essa não-linearidade: serviços de diferentes tipos (serviços seguidos ou serviços partidos) têm custos diferentes (não explícitos), assim como a percentagem de diferentes tipos de serviço na solução final. Esta relação é claramente não-linear e tem um custo na qualidade da solução

Outra fonte de não-linearidade da estrutura implícita de custos do SPP é o pagamento de horas-extras. Ao se exceder a duração da jornada de trabalho, para que um serviço possa cobrir mais tramos é necessária a adição de um valor extra a ser pago, o qual é proporcionalmente mais oneroso do que uma hora trabalhada em jornada normal. Tem-se, novamente, uma evidência da não-linearidade de custos do SPP.

Dias (1995, p.10) apresenta o seguinte comentário sobre a não-linearidade do problema de geração de serviços de tripulações:

“Um caso de não linearidade surge, por exemplo, quando se pretende calcular o custo de um serviço em horas extraordinárias; de facto, em algumas empresas, o custo associado a um período de quatro horas de trabalho extraordinário é superior ao custo associado a dois períodos de duas horas. Os actuais processos de lidar com a não linearidade tornam o modelo de difícil resolução”.

Uma forma de evitar, na medida do possível, sua natureza de custos não-linear, é trabalhar com uma função objectivo que, em vez de procurar minimizar os custos dos serviços a serem seleccionados, procure minimizar os tramos não cobertos.

Variante proposta para o Maximal Covering Problem

Diante do exposto, propõe-se a seguir uma variante do MCP, aplicada ao VCSP sem *changeovers*, a qual pode ser formulada matematicamente como segue:

[MCP₃]

$$\text{Minimizar} \quad z = \sum_{i=1}^m z_i \quad (3.11)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij}x_j + z_i = 1 \quad \forall i = 1, \dots, m. \quad (3.12)$$

$$\sum_{j=1}^n x_j \leq d \quad (3.13)$$

$$x_j \in \{0,1\} \quad \forall j = 1, \dots, n. \quad (3.14)$$

$$z_i \geq 0 \quad \forall i = 1, \dots, m. \quad (3.15)$$

A função objectivo representada pela expressão (3.11) procura minimizar o número de tramos descobertos. Caso o i -ésimo tramo não possa ser coberto, o conjunto de restrições (3.12) arbitra que $z_i = 1$. Em consequência desta restrição, tem-se que a integralidade das variáveis z_i pode ser relaxada. A restrição (3.13) impõe que um número máximo de d de colunas da matriz A seja seleccionado na solução. O conjunto de restrições (3.14) impõe o carácter binário das variáveis de decisão. As restrições do tipo (3.15) impõem a não-negatividade das variáveis z_i .

Com relação à formulação proposta acima, note-se que:

- As variáveis do tipo z podem ser consideradas como variáveis auxiliares, pois, na verdade, não se caracterizam efetivamente como decisões. Se $z_i = 1$, então ocorre um *leftover*.
- O conjunto de restrições (3.12) impede a ocorrência de *overcovers*, diferindo das formulações tradicionais para o MCP por adoptar restrições de igualdade.
- Ao contrário dos modelos clássicos de máxima cobertura, nos quais o uso de d colunas é imposto, a restrição (3.13) permite que um número de colunas inferior à d seja utilizado.
- A relaxação linear do conjunto de restrições (3.14) incorre na obtenção de um limite inferior para a solução do modelo.

3.3 Abordagens heurísticas para o MCP e o SPP

Como já foi ressaltado, o MCP pertence à classe de problemas NP-difícil. Como o foco do presente trabalho é a aplicação do MCP em problemas de *scheduling* em sistemas de transporte público, problemas nos quais a matriz A geralmente possui centenas de linhas e milhares de colunas, o uso de métodos exactos se mostra, por isso, pouco atrativo.

Assim, serão analisadas, de seguida, algumas abordagens heurísticas para o MCP e para o SPP encontradas na literatura. Com base nestes estudos reportados na literatura, será proposta uma heurística para o MCP aplicado ao VCSP sem *changeovers*.

Teitz e Bart (1968) apresentam uma heurística para o problema de p -medianas, o qual tem forte relação com o MCP. A filosofia do algoritmo de Teitz e Bart é a seguinte: dado um grafo ponderado, com vértices que fazem parte da solução e outros que não fazem, deve-se trocar iterativamente tais vértices, de modo a obter melhorias na função de avaliação. Tal procedimento é a base dos algoritmos de melhoria de soluções utilizados por diversos outros autores que abordam o MCP.

O problema de p -medianas pode ser modelado como um grafo no qual os nós correspondem aos clientes (aos quais são associados um valor de procura) e às potenciais facilidades que devem atendê-los. Os arcos representam os custos de ligações entre os vértices. O problema consiste em determinar um subconjunto de vértices a atender os demais vértices, ao menor custo possível. Percebe-se que o problema de p -medianas objectiva cobrir uma demanda com uma quantidade limitada de recursos, assemelhando-se bastante ao MCP.

Galvão e ReVelle (1996) apresentam uma heurística lagrangeana para o problema de localização de máxima cobertura (*Maximal Covering Location Problem – MCLP*). Os autores citados modelaram o MCPL por meio da formulação [MCP₁]. A abordagem proposta se mostrou extremamente eficiente, para as diversas instâncias testadas.

Posteriormente, Galvão *et al.* (2000) fazem um estudo comparativo entre heurísticas baseadas nas relaxações lagrangeana e *surrogate* e constatam que, para as diversas instâncias analisadas, a diferença entre os resultados das duas abordagens é pouco significativa.

Apesar de extremamente eficientes, tais tipos de abordagens possuem a limitação de requererem uma série de parâmetros, cujos valores podem variar bastante de acordo com o porte da instância.

Resende (1998) propõe uma heurística GRASP para o MCP. Para um conjunto de 90 instâncias de grande porte, a heurística conseguiu obter soluções próximas e, em muitos casos iguais, ao limite inferior obtido através de relaxação linear. O algoritmo, de fácil implementação, mostrou-se extremamente eficaz e eficiente para a resolução do MCP. Uma das principais vantagens do GRASP é a baixa quantidade de parâmetros a ser calibrada para a execução do algoritmo.

Park e Ryu (2006) apresentam três Algoritmos Genéticos, uma Pesquisa Tabu e um algoritmo *Simulated Annealing* para o MCP, aplicado na geração de serviço de pessoal tripulante. Os algoritmos foram testados em instâncias reais e de grande porte, advindas de sistemas de transporte por metro. Um novo tipo de Algoritmo Genético, denominado *Genetic Algorithm with Unexpressed Genes (GAUG)*⁵, obteve os melhores resultados.

É importante salientar que na abordagem de Park e Ryu (2006), *overcovers*, ou seja, situações em que a mesma viatura está atribuída a mais do que um tripulante, são permitidos.

Prata *et al.* (2009) propõem uma heurística *Binary Particle Swarm Optimization (BPSO)* para uma variante do MCP. Tal variante corresponde ao modelo ilustrado pelas equações (3.6) a (3.10), com todos os custos iguais a 1. Os autores citados resolvem 10 instâncias geradas aleatoriamente. As vantagens da heurística BPSO consistem na fácil implementação computacional da heurística e a baixa dependência de parâmetros. Contudo, como limitações da abordagem em foco, pode-se destacar que, para uma melhoria dos resultados obtidos pela heurística, requer-se a geração eficiente da população inicial, bem como a aplicação de procedimentos de melhoria (*local search*) nas soluções geradas.

Berman *et al.* (2009) apresentam três heurísticas para um caso particular do MCP, em que alguns pesos podem ser negativos⁶: um algoritmo baseado na heurística

⁵ O conceito de genes não-expressos consiste em gerar cromossomas com uma parte explícita e outra implícita. A parte implícita (não expressa) pode vir a ser incorporada em gerações futuras, evitando perda de boas informações sobre os indivíduos durante o processo de pesquisa.

⁶ De acordo com Berman *et al.* (2009), no caso do problema de localização de facilidades, o uso de pesos negativos pode atuar como uma penalização, para evitar que clientes indesejáveis situem-se próximos das

de Teitz e Bart, um algoritmo *Simulated Annealing* e um algoritmo Pesquisa Tabu. Ao adaptarem 40 instâncias do problema de p -medianas disponíveis na literatura à variante do MCP supracitada, os autores constataram que o *Simulated Annealing* implementado obteve os melhores resultados, obtendo as melhores soluções conhecidas em cerca de 90% das execuções da heurística.

Conforme discutido na secção 2.2.2 desta Tese, o SPP e o SCP possuem estreita relação. Na secção 3.1, abordou-se a relação conceitual entre o SCP e o MCP. Percebe-se, portanto, que o SPP e o MCP também se relacionam.

O SPP pode ser visto como um problema que se destina a dividir um conjunto de clientes de modo que cada *cluster* seja atendido por uma única facilidade. O MCP se diferencia do SPP devido ao facto de que uma facilidade pode atender clientes de *clusters* distintos.

Apesar do modelo proposto para o VCSP sem *changeovers* ser de máxima cobertura, ele se aproxima bastante de um modelo de particionamento de conjuntos, devido às restrições de igualdade (3.12).

A formulação proposta (vide equações (3.11) a (3.15)) possui forte relação com o *Relaxed Set Partitioning Problem* (RSPP) proposto por Dias *et al.* (2002), apresentado no Capítulo 2 (equações (2.22) a (2.25)). A diferença entre os dois modelos é que, no caso da variante do MCP, os custos das colunas são ignorados e há a limitação da quantidade de colunas a serem introduzidas na solução.

Deste modo, é importante destacar alguns trabalhos que abordam o SPP. Poucos são os trabalhos que abordam o SPP por meio de métodos heurísticos, dada a grande dificuldade em se gerar soluções viáveis para o problema.

Levine (1994, 1996) apresenta um Algoritmo Genético híbrido para o SPP. A forma de representação é binária e a população é gerada aleatoriamente, podendo-se infringir as restrições de igualdade, sendo utilizada uma função para penalizar as soluções inviáveis. Utiliza-se selecção por torneio binário e o método de reposição *steady-state*⁷. O autor emprega um *two-point crossover*, com probabilidade de 60%, ou

facilidades. Por exemplo, ao estudar a localização de um hospital, a proximidade deste para um depósito de lixo é indesejada.

⁷ Uma das questões cruciais em um GA é a forma como a prole gerada é introduzida na população. Na estratégia geracional, a cada iteração do algoritmo, ocorre uma substituição total (ou próxima disto) dos indivíduos da geração anterior pelas soluções geradas por recombinação e mutação. Na estratégia *steady-state*, ocorre a substituição de um ou de poucos indivíduos menos aptos da população anterior pela prole gerada (COLEY, 2010).

mutação clássica, com probabilidade de 40%. Para o refinamento das soluções, é utilizada uma heurística de pesquisa local denominada ROW, a qual é aplicada em um dos indivíduos da população.

A heurística ROW aborda o problema orientado pelas linhas da matriz A . Com base num dado critério de paragem (no caso, o número máximo de iterações), a heurística tenta reduzir a inviabilidade de soluções (*leftovers* e *overcovers*), ou mesmo repará-las, bem como inserir colunas que levem a uma melhoria da função de avaliação.

Beasley e Chu (1995, 1998) propõem um Algoritmo Genético para o SPP. A forma de representação escolhida é a binária. Emprega-se selecção por torneio binário e método de reposição de indivíduos *Steady-State*. O *crossover* escolhido é o uniforme. A mutação consiste em inverter uma quantidade especificada de bits de uma solução gerada por *crossover*. Para cada elemento da população é utilizada uma heurística de melhoria, a qual se destina a corrigir a inviabilidade das soluções.

Dias *et al.* (2002), propõem um Algoritmo Genético para o RSPP. A codificação utilizada é a não-binária, sendo que cada gene do cromossoma corresponde a uma coluna que cobre uma determinada linha da matriz A . A população inicial é composta apenas por soluções viáveis, geradas de forma construtiva e aleatória. O operador de cruzamento é desenvolvido de forma a gerar apenas soluções viáveis, a partir de dois pais seleccionados por torneio. Dois tipos de mutação são utilizadas: uma mutação básica (modificação aleatória de um gene) e uma mutação para melhoria das soluções, a qual consiste em um procedimento de pesquisa local.

Dentre as três abordagens citadas acima para problemas de particionamento, pode-se observar que o AG de Levine é o que apresenta os piores resultados. Isso se deve ao facto do autor permitir os *overcovers*, penalizando-os na função objectivo. Deste modo, em alguns casos, a heurística sequer consegue chegar a uma solução viável.

Com relação aos parâmetros, as três abordagens supracitadas necessitam de diversos parâmetros a serem ajustados, a saber: tamanho da população, número de gerações, probabilidade de cruzamento, probabilidade de mutação, tipo de mutação, bem como os parâmetros dos algoritmos de melhoria.

3.4 Uma heurística GRASP para a variante do MCP proposta

A variante do problema de máxima cobertura em estudo, por causa das restrições de igualdade, apresenta um espaço de soluções bastante restrito, sendo difícil a obtenção de soluções viáveis a partir da exploração da vizinhança de uma solução qualquer.

Assim, e com base nas abordagens heurísticas para o MCP e o SPP, optou-se por desenvolver um algoritmo GRASP (FEO e RESENDE, 1989; FEO e RESENDE, 1995) para a variante proposta do MCP. Tal escolha se baseia nos seguintes aspectos:

- a metaheurística GRASP é pouco dependente de parâmetros;
- possui uma implementação computacional extremamente simples;
- devido ao conceito de construção gulosa aleatorizada, o GRASP facilita a construção de soluções viáveis;
- o uso de uma heurística construtiva aleatorizada, em conjunto com uma pesquisa local, combina mecanismos de diversificação e intensificação.

Uma heurística construtiva é um procedimento que se destina a gerar uma solução para um problema de otimização. A cada passo do algoritmo, existem diversos candidatos que podem ser escolhidos para compor a solução, o que define uma Lista de Candidatos Restrita (ou, em inglês, *Restricted Candidate List* – RCL). Um algoritmo é dito guloso (*greedy*) se escolhe, a cada passo, aquele candidato que tem o melhor impacto para a função de avaliação utilizada.

Os procedimentos gulosos usualmente são de fácil implementação e apresentam (em muitos casos) boas soluções iniciais, com baixo custo computacional. Porém, a grande desvantagem de um procedimento guloso é a ausência de uma visão sistêmica sobre o conjunto de candidatos a comporem uma solução.

Hart e Shogan (1987) propuseram dois mecanismos para aprimorar a eficácia de um algoritmo deste tipo: um mecanismo baseado em cardinalidade, no qual os k melhores candidatos são adicionados na RCL; e um mecanismo baseado em valor, no qual todas as soluções que tiverem valores melhores do que “ $\alpha \times$ melhor valor”, devem compor a RCL, em que $\alpha \in [0,1]$.

Paralelamente e de forma independente, Feo e Resende (1989) propuseram o conceito de construção gulosa aleatorizada, o qual, seguido de um procedimento de pesquisa local, viria a compor a metaheurística GRASP – *Greedy Randomized Adaptive Search Procedures*.

Conforme Resende e Ribeiro (2003), o GRASP é uma metaheurística de multi-partida composta por duas fases: uma etapa de construção, a qual objetiva gerar uma solução inicial por uma heurística construtiva gulosa aleatorizada; e uma etapa de pesquisa local, a qual prima por efectuar uma busca na vizinhança da solução inicial gerada. Estas duas etapas são repetidas um número máximo de iterações (o qual é um parâmetro da metaheurística) e, no final do processo, a melhor solução obtida na busca é retornada.

A busca é dita gulosa e aleatorizada devido ao parâmetro α , o qual controla a cardinalidade da RCL. Em um problema de minimização, para $\alpha=0$ tem-se uma busca aleatória, e, para $\alpha=1$ tem-se um algoritmo guloso. Valores intermediários para o parâmetro α propiciam um *trade-off* entre aleatoriedade e gulosidade.

A busca é dita adaptativa, pois a função para avaliação da gulosidade é actualizada a cada iteração da heurística construtiva. Conforme Viana *et al.* (2003), no GRASP a propriedade de adaptabilidade representa um acúmulo dinâmico de conhecimento sobre o problema, geralmente conduzindo a geração de soluções de boa qualidade.

Ainda de acordo com Feo e Resende (1995), uma característica importante do GRASP é a facilidade com a qual a metaheurística pode ser implementada. Em implementações padrão, apenas dois parâmetros devem ser utilizados e ajustados (a cardinalidade da RCL e a quantidade de iterações GRASP). Portanto, o desenvolvimento pode ser direcionado para a optimização das estruturas de dados, aprimorando a eficiência das iterações GRASP. Feo e Resende (1995) também salientam que o GRASP é uma metaheurística que pode ser facilmente paralelizada.

Segundo Resende e Ribeiro (2008), o GRASP pode ser visto como uma técnica de amostragem, regida por uma função densidade de probabilidade desconhecida, sendo que o processo de amostragem é regido pela RCL. Se a lista de candidatos só tiver uma solução, a solução média será constante e a variância nula.

No Quadro 1 apresenta-se a estrutura geral da heurística GRASP. A seguir, a fase de construção e a fase de pesquisa local para a variante do MCP serão descritas em detalhe.

Quadro 1: Heurística GRASP.

```

Parâmetros:  $\alpha$ , num_iter, num_sol
enquanto  $i \leq \text{num\_iter}$  faça
    Fase de construção
    Algoritmo de melhoria
    Actualiza melhor solução encontrada
     $i \leftarrow i + 1$ 
fim-do-enquanto

```

Fase de construção

Tendo em vista que a variante do MCP apresentada difere dos modelos estudados na literatura, foi necessário desenvolver um procedimento específico para o caso em foco. A heurística clássica de Chvátal, procedimento para o *Set Covering Problem*, foi adaptada ao MCP, conforme ilustrado no Quadro 2.

No início do algoritmo todas as colunas pertencem à lista AV de colunas disponíveis e no máximo d colunas podem ser adicionadas à solução. Para avaliar a atratividade da inserção de uma coluna na solução, utilizou-se a seguinte função gulosa:

$$\gamma_j = \sum_{i=1}^m a_{ij}, \forall j = 1, \dots, n. \quad (3.16)$$

Ou seja, um vector γ irá armazenar a quantidade de linhas cobertas por cada coluna.

Quadro 2: Adaptação da heurística de Chvátal para a fase de construção.

```

AV ← xj, ∀ j= 1, ..., n
col ← 0
enquanto AV ≠ ∅ ou col ≤ d faça
  obter a coluna xj que cobre a maior
  quantidade de linhas da matriz A
  solução ← solução ∪ xj
  Actualiza AV
  Actualiza função gulosa
  col ← col +1
fim-do-enquanto

```

Em seguida, buscar-se-á a coluna que apresentar maior valor da função γ_j , sendo então adicionada na solução. A lista AV é então actualizada, sendo removida a coluna inserida e as demais colunas que cobrirem as mesmas linhas cobertas pela coluna inserida, de modo que as próximas inserções satisfaçam as restrições de igualdade. Por fim, deve-se actualizar a função gulosa, pois linhas que já foram cobertas pela coluna inserida não devem contabilizar atratividade para colunas a serem inseridas. As operações descritas são repetidas enquanto houver colunas disponíveis ou enquanto o número máximo de colunas na solução não tiver sido excedido.

É importante observar que no caso da variante do MCP, diante das restrições de igualdade, a actualização da função gulosa consiste na remoção de todas as colunas que possuem conflitos (sobreposições de coberturas de linhas) com a coluna previamente

inserida na fase de construção, pois uma linha não pode ser coberta por mais de uma coluna.

A RCL é construída pelo algoritmo descrito no Quadro 3. Sobre este algoritmo, um aspecto a ser destacado é que o procedimento guloso proposto não é determinístico. Em caso de empate de colunas que cubram a mesma quantidade máxima de linhas, todas irão compor a RCL e a coluna a ser inserida na solução será escolhida aleatoriamente.

Quadro 3: Procedimento de construção da RCL.

```

RCL  $\leftarrow \emptyset$ 
 $\gamma^* = \max\{\gamma_j\}$ 
para  $j=1$  até  $n$  faça
    se  $\gamma_j \geq \alpha \times \gamma^*$  então
        RCL  $\leftarrow$  RCL  $\cup \{j\}$ 
    fim-do-se
fim-do-para

```

Após a composição da RCL, a cada passo da fase de construção, a selecção das colunas na solução será feita de acordo com uma distribuição uniforme, isto é, todos os elementos que compõem a RCL possuem a mesma probabilidade de serem escolhidos. Existem diversos modos para a determinação do valor de α . Posteriormente, na secção 3.5, são descritos os experimentos realizados para a sua determinação.

Fase de pesquisa local

Para a variante do MCP em foco, a codificação de uma solução consiste em um vector \mathbf{x} de modo que a j -ésima componente é igual a 1 se x_j faz parte da solução, sendo igual a zero, caso contrário. Outro modo de representação seria a existência de duas listas: uma lista que armazena as colunas que fazem parte da solução e outra lista que armazena as colunas que não fazem parte da solução.

Para definir a estrutura de vizinhança utilizada, considera-se que uma solução vizinha da actual é obtida através de um “movimento”. Para o problema abordado, foram avaliados três movimentos:

- movimento de inserção: adicionar uma coluna não presente na solução;
- movimento de remoção: remover uma coluna presente na solução;
- movimento de troca: trocar uma coluna presente na solução por uma coluna não presente na solução.

Pode-se observar que o movimento de troca é uma combinação dos movimentos de inserção e remoção.

A grande dificuldade em explorar a vizinhança de uma solução é a obtenção de soluções vizinhas viáveis, visto que as restrições de igualdade são bastante restritivas. Ao retirar uma coluna da solução, pode ocorrer que não seja possível inserir nenhuma outra sem infringir as restrições de igualdade.

Para exemplificar este problema, considere-se a matriz abaixo, cujo significado foi detalhado na secção 2.2.2. O vector $\mathbf{x} = (1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0)$ é uma solução viável para a variante do MCP em foco. Caso se deseje explorar soluções vizinhas, pode-se, por exemplo, remover a primeira coluna da solução e tentar inserir as colunas 3, 4, 5, 6 e 7.

$$\begin{vmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{vmatrix}$$

Nenhuma das colunas 3, 4, 5 e 6 pode ser inserida na solução actual, pois tal implicaria o aparecimento de *overcovers*. Logo, a única troca viável seria a da coluna 1 pela coluna 7, pois esta não possui nenhum conflito com a coluna 2, a qual está fixa na solução corrente.

Neste sentido, o número de linhas, de colunas e a densidade da matriz A possuem impacto crucial para a dificuldade da instância, pois quão maiores são esses valores, tende-se a ter uma maior dificuldade em explorar a vizinhança de soluções viáveis.

Uma decisão crucial para a resolução de modelos de cobertura e particionamento diz respeito ao tratamento das soluções inviáveis. Existem duas alternativas para o problema: penalizar soluções inviáveis ou utilizar uma heurística de reparos⁸. Tais abordagens possuem suas vantagens e desvantagens, de acordo com o tipo de restrição e de problema a ser resolvido.

⁸ Do termo *repair heuristic*. Uma heurística de reparos consiste em um procedimento para tornar uma solução inviável numa solução viável, por meio de alterações na solução.

Bäck *et al.* (1995) apresenta um estudo comparativo entre a aplicação de funções de penalidade, uma heurística de reparos e operadores estocásticos no problema de cobertura de conjuntos. Os autores demonstram que um algoritmo guloso simples obtém sempre resultados melhores que um Algoritmo Genético sem uma heurística de reparos. Os autores salientam que o uso de uma heurística de reparos é uma opção melhor do que o uso de funções de penalidade.

A função de penalidade tem a vantagem de permitir que o algoritmo escape de ótimos locais, dirigindo a busca para regiões promissoras. Sabe-se que uma solução de alta qualidade pode estar rodeada de soluções inviáveis. Contudo, dependendo do tipo de restrição a ser penalizada, a penalização pode incorrer em uma expansão do conjunto de soluções a serem avaliadas, tornando lento e pobre o processo de busca.

A heurística de reparos possui a grande vantagem de reduzir o espaço de busca apenas a soluções viáveis, o que geralmente propicia melhores resultados que a penalização. Por outro lado, dependendo do problema e da restrição violada, a heurística de reparo pode confinar a busca em regiões do espaço de soluções, impedindo o acesso a ótimos locais de maior qualidade.

Na abordagem proposta, far-se-á uso tanto de funções de penalidade quanto de heurística de reparos.

Foram implementados os seguintes algoritmos de melhoria:

- adaptação da heurística de Berman *et al.* (2009);
- heurística de remoção aleatória de colunas;
- adaptação da heurística de Beasley e Chu (1998).

A heurística de Berman *et al.* (2009), baseada na heurística de Teitz e Bart (1968), consiste em efectuar y ($n - y$) trocas de colunas, em que y é o número de colunas na solução e n é o número de colunas no problema. Tendo em vista que existe uma imensa quantidade de trocas possíveis, as y ($n - y$) trocas consistem em uma amostragem da vizinhança. Deve-se observar que a pesquisa local é executada uma vez a cada iteração GRASP, incorrendo em um elevado custo computacional. Verificou-se que a grande maioria das trocas levava a soluções inviáveis. Mesmo com a implementação de uma heurística de reparos, esta abordagem não levou a bons resultados.

A heurística de remoção aleatória de colunas consiste em remover k colunas da solução e inserir outras k colunas de forma gulosa. Quando $k=1$, em alguns casos, não

existia solução vizinha viável. Na medida em que k se aproxima de d , perde-se o conceito de solução vizinha e o algoritmo de assemelha a uma busca aleatória. A vantagem desta heurística é que ela só trabalha com soluções viáveis. Em contrapartida, para diversas instâncias, a busca ficava retida em ótimos locais. Outra desvantagem era o custo computacional elevado da heurística, se comparada com as outras abordagens avaliadas.

Beasley e Chu (1998) apresentam um algoritmo de melhoria de soluções para o SPP. Em linhas gerais, tal procedimento consiste em corrigir os *overcovers* e inserir colunas que cubram linhas não cobertas. Como o objectivo do MCP é cobrir as linhas da matriz A , abordou-se o problema observando as linhas que estão descobertas.

A seguir, no Quadro 4, é apresentado o procedimento de pesquisa local utilizado, cujo número de execuções é controlado pelo parâmetro *num_sol*. A heurística gera uma lista das linhas não cobertas na solução corrente e selecciona uma linha a ser coberta. Em seguida, selecciona-se aleatoriamente uma coluna não presente na solução que cubra a linha seleccionada, adicionando-a na solução. Por fim, aplica-se uma heurística de reparos, que, a seguir, se descreve com detalhe.

O vector \mathbf{w} armazena as relações de cobertura de uma dada solução do problema. Se $w_i=0$, então a i -ésima linha está descoberta na solução actual; se $w_i=1$, então a i -ésima linha é coberta por apenas uma coluna na solução actual; e, por fim, se $w_i \geq 2$ então ocorre pelo menos um *overcover*. Então, enquanto a soma dos elementos de \mathbf{w} maiores que 1 for diferente de zero, existirão *overcovers* a serem corrigidos. A selecção das colunas a serem removidas da solução é feita de forma aleatória, de modo a evitar alguma convergência prematura de colunas na solução.

Quadro 4: Algoritmo de melhoria para a variante do MCP.

```

Passo 1: selecciona aleatoriamente uma linha a ser coberta
Passo 2: selecciona aleatoriamente uma coluna a cobrir a linha
escolhida
Passo 3: adiciona a coluna na solução
Passo 4: Aplicar a heurística de reparos:
 $\mathbf{w} \leftarrow A \times \mathbf{x}^T$ 
enquanto (soma( $\mathbf{w}>1$ ) $\neq 0$ ) faça
  para  $i=1$  até  $m$ 
    se  $w_i \geq 2$  então
      selecciona aleatoriamente coluna a descobrir a linha
      remove a coluna na solução
       $\mathbf{w} \leftarrow A \times \mathbf{x}^T$ 
    fim-do-se
  fim-do-para
fim-do-enquanto

```

Pode-se observar que na heurística de reparos não é analisada a restrição inerente ao número máximo de colunas. Após experimentos computacionais, constatou-se que, quando a viabilidade desta restrição era reparada, a busca ficava confinada em ótimos locais. Diante desta limitação da heurística de reparos, optou-se por hibridizá-la com uma função de penalidade, de modo a permitir que a busca fosse direcionada para regiões promissoras do espaço de soluções.

A função de penalidade consiste em penalizar, na função objectivo, soluções que tenham um número de colunas maior do que o valor máximo admissível. Seja y o número de colunas na solução corrente e d o número máximo de colunas na solução. Então, a função de penalidade p pode ser definida como:

$$p = \max\{\lambda(y - d), 0\} \quad (3.17)$$

Se $y \leq d$, tem-se uma penalidade nula; caso contrário, a função objectivo é penalizada de acordo com a diferença de y e d e um factor de penalidade λ . Conforme Levine (1996), ainda não existem métodos exactos para a determinação dos valores de uma penalização, e a escolha do valor ideal é uma tarefa difícil. Optou-se por ajustar o valor de λ empiricamente, e, após experimentos computacionais, chegou-se a um valor $\lambda = \lfloor n/10 \rfloor$.

3.5 Resultados computacionais

Tendo em vista que não foram encontradas na literatura, instâncias para a variante do MCP proposta, foram adaptadas instâncias dos problemas SPP e SCP disponíveis na biblioteca OR-Library (BEASLEY, 1990), disponível no sítio <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>. O objectivo essencial dos testes não é a resolução do problema real, mas a validação da abordagem proposta.

A conversão das instâncias dos problemas supracitados ocorreu da seguinte forma: foram utilizadas as matrizes A de cada instância, sendo ignorados os custos inerentes às colunas. Atribuiu-se um valor do número máximo de colunas equivalente a $\lceil 0.2 \times m \rceil$.

As instâncias foram rodadas no *software* LINGO para obtenção de marcos de referência. Caso a solução óptima não fosse obtida em 3600 segundos, o *solver* era parado e guardava-se a melhor solução obtida. Os experimentos foram realizados em um processador Genuine Intel 1.86 GHz com 1GB de memória RAM.

A seguir, nas Tabela 3 e 4, são apresentadas as instâncias adaptadas, respectivamente, do SPP e do SCP, assim como os resultados da resolução da relaxação linear e do problema binário. As instâncias adaptadas do SPP representam problemas reais de programação de tripulações no transporte aéreo (*Airline Crew Scheduling*) que foram modelados como um SPP puro, sem nenhum outro tipo de restrição.

Deve-se salientar que não foram adaptados os procedimentos de redução do SPP para o MCP. Beasley e Chu (1998) apresentam procedimentos de pré-processamento da matriz A do SPP, objectivando a redução do problema pela eliminação de colunas redundantes, as quais podem ser eliminadas sem alterar a qualidade da solução gerada.

As características das instâncias são as seguintes: número de linhas da matriz A (m), número de colunas da matriz A (n), número máximo de colunas a serem utilizadas na solução (d) e densidade da matriz A (ρ). O valor de ρ é obtido pela divisão do número de elementos da matriz A iguais a 1 pelo número total de elementos de A . Estas informações constam nas colunas 1 a 5 das Tabela 3 e 4. Na coluna 6 consta a solução obtida para o problema com relaxação da integralidade das variáveis x , na coluna 7 consta o tempo de processamento para resolução do problema relaxado e, na coluna 8, consta o número de iterações para a obtenção da solução. Na coluna 9 consta a solução obtida para o problema binário, na coluna 10 consta o tempo de processamento para resolução do problema binário, na coluna 11 consta o número de *branches* efectuado pelo *branch-and-bound*, na coluna 12 consta o número de iterações para a obtenção da solução. O *duality gap*, apresentado na coluna 13, é o desvio percentual relativo entre o zIP e o zLP .

Sobre as instâncias e os experimentos computacionais apresentados nas Tabela 3 e 4, podem ser destacados os seguintes pontos:

- O parâmetro m influencia decisivamente a dificuldade da instância. Problemas com mais linhas a serem cobertas, usualmente são mais difíceis de serem solucionados.
- Para um mesmo valor de m , o aumento de n tende a tornar o problema mais fácil de ser resolvido. Se existem mais colunas disponíveis, existe uma maior possibilidade de cobrir todas as linhas da matriz A sem a existência de *overcovers*.
- Adoptou-se um valor de d definido por $\lceil 0.2 \times m \rceil$ de modo a tornar os problemas difíceis. Se d é próximo de 0 ou de m , a instância tende a ser mais fácil de ser

solucionada. Além disto, este valor de d também tentou refletir a natureza de problemas de planeamento operacional de transportes públicos.

- A densidade (ρ) é um parâmetro crucial para a dificuldade de uma instância do problema. Para valores de densidade próximos de zero, o problema tende a ser mais fácil de ser solucionado. Na medida em que se aumenta o valor de ρ , o problema tende a se tornar mais difícil. Problemas com altas densidades e com poucas colunas podem ser considerados muito difíceis.
- O *solver* utilizado só fornece tempos com precisão de segundos.

Com base nas instâncias reportadas nas Tabela 3 e 4, é possível avaliar o comportamento da metaheurística GRASP desenvolvida para a variante do MCP em foco, a qual foi implementada em MATLAB.

A heurística GRASP desenvolvida requer o ajuste dos seguintes parâmetros:

- α : parâmetro que regula a gulosidade e a aleatoriedade da fase de construção;
- *num_sol*: número de vezes que o algoritmo de melhoria é executado em uma mesma iteração GRASP;
- *num_iter*: número de iterações GRASP (execução dos procedimentos de construção e pesquisa local).

Para a calibração do parâmetro α , as seguintes estratégias foram utilizadas:

- uso de valor fixo;
- uso de valor escolhido aleatoriamente em uma distribuição uniforme;
- uso de valor escolhido aleatoriamente em uma distribuição empírica;
- estratégia reactiva (PRAIS e RIBEIRO, 2000).

O uso de um valor α fixo é a estratégia menos sofisticada para calibração do parâmetro supracitado e, geralmente, é aquela que incorre em piores resultados. Prais e Ribeiro (2000) afirmam que, ao usar um valor α fixo, o algoritmo GRASP retarda o processo de obtenção de melhores soluções.

Tabela 3: Características das instâncias e resultados do *solver* – instâncias adaptadas do SPP.

1. instância	2. m	3. n	4. d	5. ρ (%)	6. zLP	7. tzLP(s)	8. número de iterações	9. zIP	10. tzIP(s)	11. número de branches	12. número de iterações	13. duality gap (%)
NW41	17	197	4	22,10	0	0	41	0	0	0	54	0
NW32	19	197	4	24,30	0,8	0	45	1	0	7	229	25
NW40	19	404	4	26,95	0	0	42	0	0	0	27	0
NW08	24	434	5	22,39	3	0	49	3	0	0	56	0
NW15	31	467	7	19,55	0	0	103	0	1	0	92	0
NW21	25	577	5	24,89	0	0	60	0	1	1	272	0
NW22	23	619	5	23,87	0	0	50	0	0	0	27	0
NW12	27	626	6	20,00	5	0	29	5	1	0	14	0
NW39	25	677	5	26,55	0	0	48	0	1	1	380	0
NW20	22	685	5	24,70	0	0	42	0	1	0	52	0
NW23	19	711	4	24,80	2	0	58	2	1	0	38	0
NW37	19	770	4	25,83	0	0	44	0	1	0	55	0
NW26	23	771	5	23,77	0	1	41	0	0	0	48	0
NW10	24	853	5	21,18	5	1	49	5	0	0	33	0
NW34	20	899	4	28,06	0	0	45	0	0	0	67	0
NW43	18	1072	4	25,18	0,4	1	92	1	0	0	108	150
NW42	23	1079	5	26,33	0	0	50	0	0	0	37	0
NW28	18	1210	4	39,27	0	0	43	0	1	0	25	0
NW25	20	1217	4	30,16	0	0	55	0	0	0	56	0
NW38	23	1220	5	32,33	0	1	56	0	0	0	55	0
NW27	22	1355	5	31,55	0	1	40	0	0	0	28	0
NW24	19	1366	4	33,20	0	0	74	0	1	0	201	0
NW35	23	1709	5	26,70	0	0	53	0	1	0	56	0
NW36	20	1783	4	36,90	0	0	74	0	0	0	113	0
NW29	18	2540	4	31,04	0	0	50	0	1	0	31	0
NW30	26	2653	6	29,63	0	1	61	0	0	0	23	0
NW31	26	2662	6	28,86	0	0	66	0	0	0	81	0
NW19	40	2879	8	21,88	0	0	112	0	1	0	331	0
NW33	23	3068	5	30,76	0	0	61	0	1	0	15	0
NW09	40	3103	8	16,20	3	0	417	3	1	0	505	0
NW07	36	5172	8	22,12	0	1	178	0	1	0	347	0

Tabela 4: Características das instâncias e resultados do *solver* – instâncias adaptadas do SCP.

1. instância	2. m	3. n	4. d	5. ρ (%)	6. zLP	7. tzLP(s)	8. número de iterações	9. zIP	10. tzIP(s)	11. número de branches	12. número de iterações	13. duality gap (%)
4.1	200	1000	40	2,04	0	1	541	24	3600	91775	11899207	2400
4.2	200	1000	40	2,03	0	0	615	23	3600	77610	9761213	2300
4.3	200	1000	40	2,03	0	1	580	23	3600	78110	10648710	2300
4.4	200	1000	40	2,04	0	0	574	24	3600	72709	9991443	2400
4.5	200	1000	40	2,00	0	1	584	23	3600	64236	8597983	2300
4.6	200	1000	40	2,07	0	0	615	22	3600	83140	10845473	2200
4.7	200	1000	40	1,99	0	1	600	22	3600	61229	9508278	2200
4.8	200	1000	40	2,05	0	0	630	18	3600	87629	13427162	1800
4.9	200	1000	40	2,01	0	0	616	23	3600	75649	9697923	2300
4.10	200	1000	40	1,98	0	0	654	21	3600	75972	10248598	2100

Por outro lado, a calibração precisa de α no intervalo $[0,1]$ pode vir a se tornar muito laboriosa, visto que diferentes instâncias podem procurar valores de α específicos. Nos experimentos realizados, constatou-se que valores de α inferiores a 0.8 geralmente não levavam a boas soluções, valores entre 0.8 e 0.9 levavam a soluções razoáveis e valores iguais a 1 levavam aos melhores resultados. Entretanto, ao usar um valor fixo, em alguns casos pode-se ficar preso em regiões do espaço de busca.

Testou-se o uso de um valor contínuo de α escolhido aleatoriamente no intervalo $[0,1]$. De acordo com os experimentos, constatou-se que esta abordagem não levou a bons resultados. Ao se refinar o intervalo para $[0.9,1]$, os resultados obtidos foram melhores do que ao usar o intervalo $[0,1]$.

Com base nos experimentos realizados com o valor fixo e com a distribuição uniforme, pode-se usar uma distribuição empírica. Utilizou-se a seguinte distribuição discreta: $P[\alpha=0.6]=0.025$, $P[\alpha=0.7]=0.025$, $P[\alpha=0.8]=0.025$, $P[\alpha=0.9]=0.025$, $P[\alpha=1]=0.9$. Esta distribuição levou a resultados melhores do que o uso de um valor contínuo de α no intervalo $[0.9,1]$.

A estratégia reactiva foi proposta por Prais e Ribeiro (2000) e consiste em escolher o valor de α de acordo com uma distribuição de probabilidades, a qual é actualizada dinamicamente de forma selectiva. Valores de α que levem a melhores soluções terão uma maior probabilidade de serem escolhidos ao longo das iterações GRASP.

Esta abordagem, além de usualmente permitir a obtenção de melhores resultados, incorre na eliminação dos experimentos para calibração de α , que podem vir a consumir demasiado tempo de acordo com o problema. Além disso, após uma análise da função discreta de probabilidades gerada pelo processo reactivo, pode-se decidir por adoptar o melhor valor de α , ou mesmo um valor médio respectivo à convergência da variável aleatória α .

Os parâmetros requeridos para a abordagem reactiva são:

- Ψ : conjunto de valores assumíveis para α ;
- δ : parâmetro relacionado com a convergência da distribuição de probabilidades de α ;
- φ : número de iterações para os quais se mantém uma mesma distribuição de probabilidades para α .

De modo geral, a estratégia reactiva é implementada da seguinte forma: seja $\Psi = \{\alpha_1, \dots, \alpha_k\}$ um conjunto discreto de k valores para α . A cada vez que for iniciada a fase de construção GRASP, um valor de α será escolhido aleatoriamente através de uma distribuição de probabilidades que é actualizada de acordo com informações recolhidas ao longo do processo de busca.

Na primeira iteração GRASP, a probabilidade de escolher o i -ésimo valor de α é obtida por uma distribuição uniforme, ou seja, $p_i = 1/k$, $i=1, \dots, k$. Sejam f^* a melhor solução obtida até o momento e M_i a média das soluções obtidas para o i -ésimo valor de α . Tem-se que o valor de q_i é dado por:

$$q_i = \left(\frac{f^*}{M_i} \right)^\delta \quad (3.18)$$

em que δ é um parâmetro de amplificação. Tal parâmetro pode ser calibrado; contudo, conforme Prais e Ribeiro (2000), o valor $\delta = 10$ conduz a excelentes resultados.

Após cada φ iterações, actualiza-se a distribuição de probabilidades de acordo com a expressão a seguir:

$$p_i = \frac{q_i}{\sum_{j=1}^k q_j} \quad (3.19)$$

No cômputo de p_i , o parâmetro de amplificação desempenha grande importância. Se $\delta = 1$, a distribuição irá demorar para convergir para a distribuição ideal. Valores maiores de δ contribuem para uma pressão selectiva para os melhores valores de α .

Diante da descrição geral da abordagem reactiva, pode-se elucidar os experimentos realizados para calibração de α . No caso em estudo, inicialmente adoptou-se um conjunto $\Psi = \{0, 0.1, 0.2, \dots, 0.9, 1\}$ com vista a efectuar uma análise geral sobre o impacto de α para a qualidade das soluções GRASP geradas. Como se tratavam de 11 valores a serem escolhidos, foi adoptado um valor $\varphi = 100$ iterações para que houvesse uma maior possibilidade de cada valor de α ser seleccionado.

Para os experimentos realizados, verificou-se realmente a tendência da qualidade das soluções geradas pelo valor $\alpha=1$. Constatou-se a cardinalidade usada para a lista Ψ e o elevado valor de φ deixavam o algoritmo lento para as instâncias com maiores valores

de n , pois o mesmo demorava excessivamente para ajustar a distribuição de probabilidades adequada.

Com base nos experimentos realizados para valores fixos de α , adoptou-se uma nova lista $\Psi = \{0.8, 0.9, 1\}$ e $\varphi=20$. Desta forma, a fase de construção não ficava confinada a um valor fixo de α e não demorava excessivamente a convergir para uma distribuição de probabilidade adequada.

Como o GRASP é um procedimento de amostragem repetido por um número de iterações, naturalmente a melhor solução GRASP tende a melhorar com o aumento do número de iterações. Portanto, deve-se gerir o *trade-off* entre qualidade da solução gerada e tempo de processamento. Nesse contexto, adoptou-se um valor $num_iter = 200$.

O parâmetro num_sol diz respeito ao número de vezes que é executado o algoritmo de melhoria. Em outras palavras, num_sol representa a quantidade de soluções vizinhas pesquisadas no processo de pesquisa local. Após experimentos, adoptou-se um valor $num_sol = m$, pois se verificou que valores elevados de num_sol incorriam em maiores tempos de processamento e não traziam benefícios proporcionais a esse acréscimo de tempo.

Tendo em vista que a metaheurística GRASP é um processo de amostragem aleatória, optou-se por rodar o algoritmo 10 vezes, para avaliar o comportamento médio do mesmo, em termos de qualidade de solução obtida e de tempo de processamento. Os experimentos foram realizados em um processador Genuine Intel 1.86 GHz com 1GB de memória RAM.

A seguir, nas Tabela 5 e 6, são apresentadas as soluções obtidas pelo GRASP, em cada uma das 10 execuções do algoritmo.

Com relação ao conjunto de dados do SPP adaptados para o MCP, pode-se observar que, para as 30 instâncias analisadas, o algoritmo GRASP conseguiu obter a solução óptima para todos os problemas. Em 26 destas instâncias, o algoritmo obteve a solução óptima em todas as 10 execuções.

Com relação ao conjunto de dados do SCP adaptados para o MCP, pode-se observar que, para as 10 instâncias testadas (cuja solução óptima não foi determinada), em duas delas o algoritmo obteve o mesmo valor de zIP , em duas o algoritmo obteve resultados piores do que o de zIP e em 6 instâncias o algoritmo obteve melhores resultados do que zIP .

Tabela 5: Soluções obtidas pela heurística GRASP proposta – instâncias adaptadas do SPP.

Problema	Melhor solução GRASP em cada uma das 10 rodadas										média
	1	2	3	4	5	6	7	8	9	10	
NW41	0	0	0	0	0	0	0	0	0	0	0
NW32	1	1	1	1	1	1	1	1	1	1	1
NW40	0	0	0	0	0	0	0	0	0	0	0
NW08	3	4	4	4	4	3	4	4	4	4	3,8
NW15	0	0	0	0	0	0	0	0	0	0	0
NW21	1	0	0	1	1	0	0	0	0	0	0,3
NW22	0	0	0	0	0	0	0	0	0	0	0
NW12	5	5	5	5	5	5	5	5	5	5	5
NW39	0	0	0	0	0	0	0	0	0	0	0
NW20	0	0	0	0	0	0	0	0	0	0	0
NW23	4	4	4	2	3	2	2	4	3	4	3,2
NW37	0	0	0	0	0	0	0	0	0	0	0
NW26	0	0	0	0	0	0	0	0	0	0	0
NW10	5	6	6	6	6	6	6	6	6	6	5,9
NW34	0	0	0	0	0	0	0	0	0	0	0
NW43	1	1	1	1	1	1	1	1	1	1	1
NW42	0	0	0	0	0	0	0	0	0	0	0
NW28	0	0	0	0	0	0	0	0	0	0	0
NW25	0	0	0	0	0	0	0	0	0	0	0
NW38	0	0	0	0	0	0	0	0	0	0	0
NW27	0	0	0	0	0	0	0	0	0	0	0
NW24	0	0	0	0	0	0	0	0	0	0	0
NW35	0	0	0	0	0	0	0	0	0	0	0
NW36	0	0	0	0	0	0	0	0	0	0	0
NW29	0	0	0	0	0	0	0	0	0	0	0
NW30	0	0	0	0	0	0	0	0	0	0	0
NW31	0	0	0	0	0	0	0	0	0	0	0
NW19	0	0	0	0	0	0	0	0	0	0	0
NW33	0	0	0	0	0	0	0	0	0	0	0
NW07	0	0	0	0	0	0	0	0	0	0	0

Tabela 6: Soluções obtidas pela heurística GRASP proposta – instâncias adaptadas do SCP.

Problema	Melhor solução GRASP em cada uma das 10 rodadas										média
	1	2	3	4	5	6	7	8	9	10	
4.1	24	23	22	26	24	26	24	24	26	23	24,2
4.2	23	23	23	23	23	20	23	23	22	23	22,6
4.3	23	23	23	24	24	24	23	24	24	24	23,6
4.4	23	25	23	23	23	23	23	23	23	23	23,2
4.5	24	21	22	23	21	23	24	24	21	23	22,6
4.6	24	25	24	25	25	25	25	25	21	26	24,5
4.7	22	22	23	23	23	25	24	24	24	25	23,5
4.8	24	24	25	24	23	23	22	26	23	23	23,7
4.9	24	26	21	26	26	27	26	23	25	26	25,0
4.10	27	26	28	25	24	27	27	26	25	25	26,0

Tais resultados apontam para a eficácia da abordagem proposta, em termos de conseguir obter soluções de qualidade para conjuntos de dados com diferentes

características (quantidades de linhas e colunas, densidades, número máximo de colunas).

A seguir, nas Tabelas 7 e 8, é apresentada a performance computacional da heurística GRASP desenvolvida, expressa em termos de tempo de processamento, para cada uma das 10 execuções do algoritmo.

Nos testes realizados nas instâncias adaptadas do SPP, em 14 dos 30 conjuntos de dados observou-se que O GRASP obteve um tempo de processamento médio inferior a 1 segundo. Nos testes efectuados com as instâncias adaptadas do SCP, observou-se que O GRASP utilizou um tempo de processamento cerca de 6 vezes menor do que o método exacto, e, ainda assim, conseguiu obter melhores soluções.

A seguir, nas Tabelas 9 e 10, são apresentadas as quantidades de soluções pesquisadas para cada uma das 10 execuções do algoritmo. Estas quantidades podem ser vistas como outro indicador da eficiência da heurística desenvolvida. Beasley e Chu (1998) utilizam este indicador para avaliar um AG para o SPP.

O tempo de processamento é o indicador de performance mais comumente empregado em estudos envolvendo heurísticas. Entretanto, no caso de problemas que requerem a correção da viabilidade de soluções, boa parte do tempo de processamento pode ser consumida na execução da heurística de reparos. Ao avaliar o número de soluções pesquisadas, pode-se ter um indicador auxiliar para a eficiência da heurística.

Com base neste indicador pode-se constatar que, em algumas execuções do GRASP, a solução óptima foi obtida ainda na fase de construção.

A qualidade de uma solução para o MCP é normalmente medida através do percentual de cobertura c , determinado pela expressão (3.20), em que m é o número de linhas a serem cobertas e z é o valor da função objectivo obtida, que, neste caso, representa o número de linhas não cobertas:

$$c = \frac{(m - z)}{m} \quad (3.20)$$

Nas Tabelas 11 e 12, são apresentados o percentual de cobertura para cada abordagem adoptada, bem como a diferença (*gap*) entre os percentuais de cobertura obtidos.

Tabela 7: Desempenho computacional da heurística GRASP proposta – instâncias adaptadas do SPP.

Problema	Tempo computacional (s) em cada uma das 10 rodadas de GRASP										média (s)
	1	2	3	4	5	6	7	8	9	10	
NW41	0,2	0,3	0,2	0,0	0,1	0,1	0,1	0,0	0,2	0,1	0,1
NW32	0,1	0,0	0,0	0,3	0,0	0,0	0,2	0,2	0,2	0,0	0,1
NW40	0,1	0,0	0,0	0,0	0,2	0,1	0,0	0,1	0,0	0,1	0,1
NW08	0,1	25,5	25,3	25,6	24,5	0,1	25,3	25,8	25,1	25,6	20,3
NW15	14,5	4,1	0,3	0,0	2,8	1,6	4,2	11,2	17,6	6,8	6,3
NW21	19,0	5,5	3,8	17,0	17,1	5,9	6,6	3,7	16,8	3,8	9,9
NW22	0,4	0,1	0,2	0,3	0,1	0,1	0,0	0,0	0,0	0,0	0,1
NW12	0,5	0,0	0,5	0,1	0,0	0,3	0,0	0,0	0,0	0,0	0,1
NW39	2,6	1,9	0,6	0,8	1,4	0,5	0,5	0,3	2,2	0,8	1,1
NW20	0,3	0,2	0,2	0,1	0,4	0,0	0,0	0,3	0,0	0,1	0,2
NW23	15,7	16,0	15,3	0,0	15,8	0,1	8,3	15,1	15,8	15,1	11,7
NW37	0,5	1,5	0,5	0,8	1,1	0,4	0,6	0,9	0,5	0,0	0,7
NW26	0,2	0,0	0,0	0,1	0,0	0,2	0,0	0,0	0,0	0,4	0,1
NW10	23,6	42,1	41,7	42,6	43,0	40,4	46,7	36,0	38,0	35,9	39,0
NW34	1,3	0,0	4,9	1,1	1,7	2,0	1,2	0,0	2,5	0,2	1,5
NW43	0,7	1,4	0,6	0,4	1,8	3,0	2,3	2,0	0,5	1,7	1,4
NW42	0,1	0,7	0,3	0,3	0,5	0,0	0,0	0,5	0,9	0,0	0,3
NW28	0,5	0,0	1,3	0,0	0,8	0,0	1,9	1,5	0,0	0,7	0,7
NW25	0,7	0,3	0,9	1,0	0,5	0,4	0,4	0,0	0,3	2,2	0,7
NW38	8,1	4,8	7,6	3,2	3,3	4,6	2,9	7,9	4,6	3,2	5,0
NW27	1,6	1,4	0,0	0,1	1,4	0,0	0,1	0,5	0,0	0,5	0,6
NW24	7,5	30,6	0,4	3,1	16,7	2,1	3,9	8,3	5,1	9,6	8,7
NW35	12,1	3,6	6,5	4,2	3,5	16,6	21,3	12,5	3,2	0,1	8,4
NW36	2,4	0,6	0,1	5,5	4,1	4,4	7,4	3,8	11,9	8,1	4,8
NW29	4,9	2,5	5,4	6,6	0,2	0,2	3,2	13,5	2,4	9,5	4,8
NW30	0,2	0,2	0,2	0,2	0,2	0,2	0,2	5,7	0,2	0,2	0,7
NW31	26,8	0,2	0,2	15,0	61,3	29,7	69,4	11,8	23,3	21,6	25,9
NW19	21,5	0,3	0,3	11,0	9,8	0,3	0,3	28,9	17,5	0,3	9,0
NW33	0,3	0,3	0,2	0,3	0,3	0,2	0,3	0,2	0,2	0,2	0,2
NW07	0,7	282,6	252,7	47,2	70,8	0,7	122,5	50,5	128,0	22,3	97,8

Tabela 8: Desempenho computacional da heurística GRASP proposta – instâncias adaptadas do SCP.

Problema	Tempo computacional (s) em cada uma das 10 rodadas de GRASP										média (s)
	1	2	3	4	5	6	7	8	9	10	
4.1	604,7	587,5	610,5	604,6	589,3	628,0	598,7	607,8	595,4	586,1	601,3
4.2	703,9	729,6	748,5	632,4	589,5	579,5	605,1	593,1	588,8	579,2	634,9
4.3	606,8	583,0	608,4	606,0	655,1	635,3	615,0	668,4	648,0	612,4	623,8
4.4	612,1	631,9	656,3	665,1	657,9	671,2	648,7	645,6	668,5	676,3	653,4
4.5	624,5	649,9	640,5	658,2	601,0	615,9	602,1	597,2	586,7	606,6	618,3
4.6	593,2	585,1	599,8	599,3	603,3	615,9	587,1	596,9	612,8	604,0	599,7
4.7	575,7	588,0	591,9	567,3	570,8	585,7	574,7	592,7	581,9	627,3	585,6
4.8	589,6	602,7	608,3	595,4	675,6	599,3	604,5	614,5	597,7	594,7	608,2
4.9	644,0	603,6	578,1	586,1	589,9	578,8	597,8	585,1	610,0	603,5	597,7
4.10	616,4	638,4	604,3	627,1	588,9	607,8	575,3	615,1	612,8	633,4	612,0

Tabela 11: Comparação entre o método exacto e a heurística GRASP proposta – instâncias adaptadas do SPP.

Problema	percentual de cobertura (<i>c</i>)				<i>gap</i>		
	zIP	média	melhor	pior	média	melhor	pior
NW41	100%	100%	100%	100%	0%	0%	0%
NW32	94,7%	94,7%	94,7%	94,7%	0%	0%	0%
NW40	100%	100%	100%	100%	0%	0%	0%
NW08	87,5%	84,2%	87,5%	83,3%	3,3%	0%	4,2%
NW15	100%	100%	100%	100%	0%	0%	0%
NW21	100%	98,8%	100%	96%	1,2%	0%	4,0%
NW22	100%	100%	100%	100%	0%	0%	0%
NW12	81,5%	81,5%	81,5%	81,5%	0%	0%	0%
NW39	100%	100%	100%	100%	0%	0%	0%
NW20	100%	100%	100%	100%	0%	0%	0%
NW23	89,5%	83,2%	89,5%	78,9%	6,3%	0%	10,5%
NW37	100%	100%	100%	100%	0%	0%	0%
NW26	100%	100%	100%	100%	0%	0%	0%
NW10	79,2%	75,4%	79,2%	75,0%	3,7%	0%	4,2%
NW34	100%	100%	100%	100%	0%	0%	0%
NW43	94,4%	94,4%	94,4%	94,4%	0%	0%	0%
NW42	100%	100%	100%	100%	0%	0%	0%
NW28	100%	100%	100%	100%	0%	0%	0%
NW25	100%	100%	100%	100%	0%	0%	0%
NW38	100%	100%	100%	100%	0%	0%	0%
NW27	100%	100%	100%	100%	0%	0%	0%
NW24	100%	100%	100%	100%	0%	0%	0%
NW35	100%	100%	100%	100%	0%	0%	0%
NW36	100%	100%	100%	100%	0%	0%	0%
NW29	100%	100%	100%	100%	0%	0%	0%
NW30	100%	100%	100%	100%	0%	0%	0%
NW31	100%	100%	100%	100%	0%	0%	0%
NW19	100%	100%	100%	100%	0%	0%	0%
NW33	100%	100%	100%	100%	0%	0%	0%
NW07	100%	100%	100%	100%	0%	0%	0%

Tabela 12: Comparação entre o método exacto e a heurística GRASP proposta – instâncias adaptadas do SPP.

Problema	percentual de cobertura (<i>c</i>)				<i>gap</i>		
	zIP	média	melhor	pior	média	melhor	pior
4.1	88,0%	87,9%	89,0%	87,0%	0,1%	-1,0%	1,0%
4.2	88,5%	88,7%	90,0%	88,5%	-0,2%	-1,5%	0,0%
4.3	88,5%	88,2%	88,5%	88,0%	0,3%	0%	0,5%
4.4	88,0%	88,4%	88,5%	87,5%	-0,4%	-0,5%	0,5%
4.5	88,5%	88,7%	89,5%	88,0%	-0,2%	-1,0%	0,5%
4.6	89,0%	87,8%	89,5%	87,0%	1,3%	-0,5%	2,0%
4.7	89,0%	88,3%	89,0%	87,5%	0,8%	0%	1,5%
4.8	91,0%	88,2%	89,0%	87,0%	2,9%	2,0%	4,0%
4.9	88,5%	87,5%	89,5%	86,5%	1,0%	-1,0%	2,0%
4.10	89,5%	87,0%	88,0%	86,0%	2,5%	1,5%	3,5%

Com base nos resultados apresentados nas Tabelas 11 e 12, pode-se salientar que o GRASP obteve soluções de qualidade comparável às do método exacto.

Em termos da média dos resultados, para as 30 instâncias adaptadas do SPP, o GRASP obteve um *gap* médio de 0,6%, enquanto que para as 10 instâncias adaptadas do SCP, o GRASP obteve um *gap* médio de 0,8%.

Em termos dos melhores resultados, para as 30 instâncias adaptadas do SPP, o GRASP obteve um *gap* médio de 0%, enquanto que para as 10 instâncias adaptadas do SCP, o GRASP obteve um *gap* médio de -0,2%.

Com relação aos *gaps* entre algumas soluções obtidas pelo GRASP e o método exacto, podem ser salientados os seguintes aspectos. A variante do MCP com restrições de igualdade é de difícil resolução, devido ao espaço de busca ser demasiadamente restrito. Tendo em vista que o vector x é bastante esparso, a distância de Hamming⁹ entre a solução incumbente e o óptimo global pode ser elevada, dificultando sua obtenção.

O trabalho de Beasley e Chu (1998) é o que apresenta os melhores resultados da aplicação de metaheurísticas para o SPP. Em 10 rodadas de um AG para o SPP, em um conjunto de 43 instâncias de Hoffman e Padberg presentes na biblioteca OR-Library, obtiveram a solução óptima em 39 instâncias. Contudo, com relação ao comportamento médio do algoritmo, foram observados *gaps* médios elevados, da magnitude de 6%. Em uma instância, o AG de Beasley e Chu chegou a obter um desvio percentual relativo de 13,4% em relação à solução óptima.

No que diz respeito à discrepância entre os tempos computacionais da heurística GRASP e do *solver* LINGO, os seguintes comentários podem ser destacados. O método aproximativo consome elevado tempo computacional para se direccionar para soluções viáveis, visto que o espaço de busca é bastante restrito.

Beasley e Chu (1998) apresentam uma comparação dos tempos computacionais de um AG e do CPLEX para a resolução do SPP. Para 43 instâncias de Hoffman e Padberg presentes na biblioteca OR-Library, em 42 foram apresentadas as soluções óptimas. Destas 42 instâncias, obteve-se um tempo médio de 42 segundos para obtenção da solução óptima, enquanto o AG apresentou tempo médio de 4902 segundos, em dez

⁹Dados dois vectores com mesmo número de componentes, a distância de Hamming consiste no número de componentes nos quais estes dois vectores são diferentes. Por exemplo, dadas as seguintes *strings* 100100 e 110101, a distância de Hamming entre os dois vectores é igual a 2.

rodadas da heurística. Ou seja, em média, o AG apresentou um tempo de processamento cerca de 116 vezes maior do que o CPLEX.

Com relação ao desempenho do GRASP nas instâncias adaptadas do SPP, diante do exposto, pode-se ressaltar que apresenta comportamento condizente com outras abordagens ilustradas na literatura.

Nas instâncias adaptadas do SCP, o GRASP obteve, em geral, melhores soluções que o *branch-and-bound*, mesmo tendo um tempo de processamento bem inferior.

3.6 Considerações finais

No presente capítulo propôs-se uma formulação do MCP, com restrições de igualdade, para o *Vehicle and Crew Scheduling Problem* sem *changeovers*.

Deve-se salientar que a formulação proposta também pode ser aplicada ao *Crew Scheduling Problem*, de acordo com os pressupostos apresentados. Tendo em conta que no VCSP sem *changeovers* as viaturas e os tripulantes se comportam como um único recurso, tal problema se reduz ao CSP. Portanto, se a abordagem proposta modela a variante supracitada do VCSP, também pode modelar o CSP.

Sobre a potencialidade de aplicação da formulação proposta em outros problemas de planeamento de transportes públicos, deve ser destacado o comentário de Lourenço *et al.* (2000). Tais autores destacam que algumas companhias de transporte permitem que tramos não sejam cobertos em uma jornada de trabalho. Nestes casos, o objectivo principal seria minimizar a quantidade de tramos descobertos. Percebe-se, portanto, a aderência da modelagem aqui proposta com esta classe de problemas.

Visando desenvolver uma abordagem para solucionar problemas reais de grande porte, foi realizada uma revisão bibliográfica sobre abordagens para o MCP e o SPP presentes na literatura.

Optou-se por desenvolver uma heurística GRASP para o MCP. A heurística foi aprimorada após diversos experimentos computacionais. Foi incorporado um mecanismo reactivo, o qual, ao utilizar memória acerca das soluções obtidas, permitiu a eliminação da necessidade de calibração do parâmetro alfa.

O uso de um valor de alfa variável, cuja escolha é regida por uma função de densidade de probabilidade ajustada durante o processo de busca, apresentou melhores resultados do que o uso de um valor fixo desse parâmetro.

A abordagem proposta foi testada em um conjunto de 40 instâncias adaptadas de problemas da literatura, apresentando resultados competitivos em relação ao *branch-and-bound*, do *solver* utilizado.

Para as instâncias adaptados do SPP, o GRASP conseguiu obter a solução ótima em todas as instâncias, tendo um desempenho, expresso em tempos de processamento, inferior ao *branch-and-bound*. Para as instâncias adaptadas do SCP, o GRASP conseguiu obter, em média, melhores soluções que o *branch-and-bound*, mesmo tendo um tempo de processamento bastante inferior.

No próximo capítulo, será apresentada uma nova formulação para o VCSP, considerando o caso em que os *changeovers* são permitidos.

Capítulo 4

O problema de programação integrada de veículos e tripulações

“If I were again beginning my studies, I would follow the advice of Plato and start with mathematics.”

Galileo Galilei

Resumo

Este capítulo tem como objectivo principal reportar uma nova formulação para o caso geral do VCSP, em que as trocas de motoristas por veículo (*changeovers*) são permitidas. Na primeira secção, é apresentado um modelo inovador para o problema. Na segunda secção, são apresentados resultados computacionais, com base neste modelo, em instâncias oriundas de um sistema de transportes públicos real. Na terceira secção, é apresentada uma abordagem, baseada em técnicas heurísticas, para o problema. Na quarta secção são apresentados os resultados computacionais da abordagem proposta. Por fim, na quinta secção, são apresentadas algumas considerações gerais sobre o trabalho realizado.

4.1 Uma formulação inovadora para o VCSP com *changeovers*

No segundo capítulo da presente tese, foram apresentadas diversas formulações matemáticas para o VCSP. De acordo com a revisão bibliográfica realizada, constatou-se que o uso de tais modelos era limitado, pois, mesmo para instâncias de pequeno porte, os modelos matemáticos não conseguiam oferecer as soluções óptimas em tempo computacional aceitável.

Nesse contexto, buscou-se a formulação de um modelo que pudesse suplantar esta limitação, assumindo hipóteses que simplificam o problema, mas que preservam a utilidade do modelo.

As seguintes premissas nortearam a concepção deste novo modelo matemático:

- No que se refere à programação dos veículos, não estão sendo considerados os custos operacionais, mas apenas os custos fixos, ou seja, deseja-se cobrir todas as viagens com a menor quantidade de veículos.
- No que se refere à programação de veículos e tripulantes, estão sendo consideradas as viagens constantes na tabela de horários e as viagens em vazio.
- No que se refere à programação de tripulações, não estão sendo considerados os custos operacionais de forma explícita. Na abordagem proposta procura-se minimizar a quantidade de tramos não cobertos por serviços gerados. Tais tramos serão cobertos por serviços extraordinários ou por serviços constituídos especificamente para este fim, conforme sugerido por Dias (2005).

A seguir, tais simplificações serão justificadas.

Conforme apresentado no Capítulo 3 da presente tese, a natureza dos custos associados a serviços de tripulações é de carácter não-linear. Os tripulantes não são pagos pelas horas efectivamente trabalhadas, mas por jornada de trabalho. Assim, os custos acabam por não refletir adequadamente a realidade. Trabalhar com uma abordagem que lide com a cobertura de tramos por um conjunto limitado de serviços permitiu uma resolução mais simples de problemas do tipo VCSP sem *changeovers*. Constatou-se que tal abordagem também é plenamente aplicável ao BDSP.

Diante destas constatações, decidiu-se investigar os impactos de formulações para o VCSP com *changeovers* que primassem pela minimização do número de veículos e tramos descobertos. Evidentemente, tais formulações não representarão explicitamente custos operacionais.

Nos modelos matemáticos para o VCSP apresentados em Gaffi e Nonato (1999), Freling *et al.* (2003) e Mesquita e Paias (2008), apenas para considerar alguns exemplos, pode-se constatar que as restrições que regulam as viagens em vazio tornam os modelos mais complexos. Um modo de simplificar o modelo seria considerar as viagens em vazio apenas na função objetivo. Desta forma, seriam eliminadas as restrições que relacionam as viagens não produtivas, viaturas e serviços, as quais dificultam o uso de métodos exactos para o VCSP.

Deve-se observar que, em algumas situações práticas, não se tem conhecimento sobre as viagens em vazio. Por exemplo, no sistema de transporte público de Fortaleza, no Brasil, em princípio, não se sabe qual a empresa que vai realizar uma dada viagem.

Deste modo, os tempos não produtivos de *pull-ins* e *pull-outs* não são considerados pelo sistema de transporte no planeamento operacional das linhas. Neste caso, não se utiliza a prática de efectuar viagens em vazio entre os términos das linhas.

O modelo proposto por Friberg e Haase (1999) não considera explicitamente a questão das viagens não produtivas, seja na função objectivo ou nas restrições. A complexidade desta formulação reside em um conjunto de restrições que garantem que cada veículo é dirigido por apenas um motorista.

Relaxar a imposição da cobertura de todos os tramos, ou seja, permitir *leftovers*, relaxa um conjunto de restrições relacionando veículos e serviços, impondo que cada tramo seja coberto por um par veículo-tripulação. Este tipo de restrição torna o modelo muito mais complexo de ser solucionado, visto que aumenta consideravelmente o número de restrições.

Permitir *leftovers* facilita bastante o processo de resolução do problema integrado; no entanto, tal permissividade resulta em uma integração mais fraca entre o VSP e o CSP. A integração é dita mais fraca, pois, no momento da programação, não se obriga que um tramo seja coberto por uma viatura e uma tripulação.

No caso de um sistema que consiste em dois sistemas independentes, sem variáveis em comum, e apenas ligados pela função objectivo, a solução do modelo pode ser obtida pela resolução dos dois modelos separadamente, adicionando-se os objectivos destes para a obtenção da solução do modelo inicial (DANTZIG e THAPA, 2003).

Assim, pode-se concluir que, nos modelos citados acima, ao não se considerar restrições que relacionem veículos e tripulantes, o VSP e o CSP podem ser modelados como dois subsistemas separados. Ao não serem consideradas restrições que modelem viagens em vazio e ao serem permitidos os *leftovers*, não haverá conjuntos de restrições que relacionem viaturas e tripulações. Procedendo-se desta forma, o modelo só irá relacionar os recursos supracitados na função objectivo.

Nesta tese, buscam-se formulações matemáticas para o VCSP que sejam mais compactas, de modo a permitir sua resolução de forma exacta em tempo computacional aceitável. Logo, de acordo com as premissas elucidadas no início da secção 4.1, ter-se-á um modelo cuja integração entre o VSP e o CSP é fraca.

Na modelagem proposta, a inovação consiste no facto que os serviços são gerados directamente sobre as viagens e não sobre blocos de viaturas definidos previamente, como ocorre na abordagem sequencial.

Com base em uma tabela de horários e em restrições laborais, são gerados serviços de pessoal tripulante, de modo a conceber uma matriz de serviços (matriz A , conforme notação usada nesta tese). Com base na mesma tabela de horários, serão programados os veículos para realizarem as viagens. Assim, a programação dos tripulantes não está atrelada à programação dos veículos, como ocorre na abordagem sequencial. Ambas as programações (veículos e tripulações) são relacionadas com a tabela de horários. O objectivo seria cobrir os pontos de procura (viagens) pelos recursos disponíveis (veículos e tripulações). Tem-se, então, um *problema de cobertura com múltiplos recursos*.

Maximal Covering Problem with Multiple Resources

Propõe-se aqui uma formulação matemática inovadora para o VCSP, no caso em que as trocas de motoristas por veículo são permitidas. O modelo consiste em uma junção de um problema de mínima decomposição e um problema de máxima cobertura. O modelo proposto é denominado *Maximal Covering Problem with Multiple Resources* (MCPMR).

Em um mesmo modelo matemático, uma parcela (problema de mínima decomposição) destina-se a modelar a programação de veículos, enquanto a outra parcela (problema de máxima cobertura) destina-se a modelar a programação de tripulantes. Tais problemas se relacionam entre si apenas pela função objectivo.

Seja T um conjunto de viagens em uma tabela de horários, de modo que $|T| = m$. A relação $i \alpha k$ representa a compatibilidade entre as viagens i e k , ou seja, a viagem k pode ser efectuada após a viagem i , pelo mesmo veículo (c_{ik} é o parâmetro que modela esta possibilidade). Seja y_{ik} uma variável de decisão binária que é igual a 1 se o veículo cobre a viagem k exactamente após a viagem i , sendo 0, no caso contrário. As variáveis z_i representam as linhas não cobertas da matriz A , de modo que $z_i = 1$ se a i -ésima linha não é coberta em uma solução, sendo $z_i = 0$, no caso contrário. As variáveis x_j representam as colunas da matriz A , de modo que $x_j = 1$ se a j -ésima coluna faz parte da solução, sendo $x_j = 0$, no caso contrário.

[MCPMR]

$$\text{Maximizar } \sum_{i=1}^m \sum_{k=1}^m c_{ik} y_{ik} - \sum_{i=1}^m z_i \quad (4.1)$$

Sujeito a:

$$\sum_{i=1}^m y_{ik} \leq 1 \quad \forall k \in N \quad (4.2)$$

$$\sum_{k=1}^m y_{ik} \leq 1 \quad \forall i \in N \quad (4.3)$$

$$\sum_{j=1}^n a_{ij} x_j + z_i = 1 \quad \forall i = 1, \dots, m. \quad (4.4)$$

$$\sum_{j=1}^n x_j \leq d \quad (4.5)$$

$$x_j \in \{0,1\} \quad \forall j = 1, \dots, n. \quad (4.6)$$

$$z_i \geq 0 \quad \forall i = 1, \dots, m. \quad (4.7)$$

$$y_{ik} \geq 0 \quad \forall i, k \in M \quad (4.8)$$

com $c_{ik}=1$ se $i \alpha k$, caso contrário $c_{ik} = -\infty$

A expressão (4.1) representa a função objectivo do problema e é composta por duas parcelas: a primeira, que procura cobrir todas as viagens com a menor quantidade de veículos possível e a segunda, que se destina a minimizar a quantidade de tramos descobertos por serviços de tripulação. Os conjuntos de restrições (4.2) e (4.3) impõem que uma viagem seja coberta apenas uma vez. Caso o i -ésimo tramo não possa ser coberto, o conjunto de restrições (4.4) impõe que $z_i = 1$. Em consequência desta restrição, tem-se que a integralidade das variáveis z_i pode ser relaxada. A restrição (4.5) impõe que um número máximo de d colunas da matriz A seja seleccionado na solução. Os conjuntos de restrições (4.6), (4.7) e (4.8) dizem respeito à definição das variáveis do modelo.

A função objectivo penaliza os tramos não cobertos, expressos pelas variáveis z_i iguais a 1. Os parâmetros c_{ik} , que modelam a conectividade entre as viagens, podem ser usados para contemplar, na função objectivo, as viagens em vazio. $c_{ik} = -\infty$ significa que não existe conectividade entre as viagens i e k . Todavia, pode-se usar o parâmetro c_{ik} como um factor de penalização. Por exemplo, admitindo que uma viagem k possa ser realizada após uma viagem i , sendo requerida uma viagem em vazio entre o término da

viagem i e o local de início da viagem k , não é inviável efectuar a conexão entre i e k , mas tais viagens também não podem ser vistas como viagens em que *deadheads* não são requeridos. Portanto, o parâmetro c_{ik} pode ser usado como um factor que penaliza, na função objectivo, a conexão entre as viagens i e k .

A primeira fração da função objectivo (expressão (4.1)), procura cobrir as viagens com a menor quantidade de veículos. Os conjuntos de restrições (4.2) e (4.3) impõem que uma viagem seja coberta apenas uma vez. Como a matriz composta pelos conjuntos de restrições (4.2) e (4.3) é unimodular, a integralidade das variáveis de decisão pode ser relaxada e substituída pelas restrições de não-negatividade das variáveis de decisão, conforme equação (4.8). Caso as viagens i e k sejam compatíveis, adopta-se $c_{ik}=1$; caso contrário, adopta-se $c_{ik}= -\infty$. Deste modo, são modeladas as relações de conectividade entre as viagens.

A restrição (4.5) tem um papel crucial para a resolução do problema. O valor do limite superior de colunas na solução, expresso por d , além de influenciar a complexidade do problema de máxima cobertura, possui também repercussões na utilização prática das soluções geradas.

Se o valor de d for elevado, a restrição (4.5) não será activa na solução óptima. Porém, também podem ser seleccionados mais serviços do que a quantidade mínima requerida para a cobertura dos tramos. Se o valor de d for pequeno, a quantidade de *leftovers* na solução gerada tenderá a ser maior.

Para os planeadores, uma sensibilidade sobre a amplitude do valor de d é de grande relevância para que a determinação do contingente de tripulantes requeridos para a execução dos serviços, bem como das possíveis horas de serviço extraordinário.

A restrição que limita o número de serviços na solução constitui-se em uma ferramenta de apoio à decisão. Na prática, os planeadores não estão interessados em minimizar a quantidade de serviços em uma solução, pois este é um valor bem determinado por critérios práticos.

A restrição de empregar até d serviços em uma solução representa o que uma solução pode perder quando se limita a quantidade de serviços. O que é o melhor que pode ser feito com uma quantidade fixa de recursos? O que pode ocorrer se houver uma variação dessa quantidade de serviços? O valor de d deve, por isso, ser objecto de uma análise de sensibilidade cuidadosa.

Deve-se salientar que o modelo apresentado acima diz respeito ao caso em que existe apenas um depósito. Mesmo o caso do VCSP com apenas um depósito vem sendo

mostrado pela literatura como de difícil resolução (FRELING *et al.* 1999b; FRELING *et al.* 2003).

4.2 Resultados computacionais

Diante do facto que não foram encontradas na literatura instâncias para o VCSP com um único depósito, com vista a avaliar o comportamento da formulação proposta, foram efectuados experimentos computacionais em instâncias reais, advindas do sistema de transporte público de Fortaleza.

No sistema de transportes colectivos por autocarros de Fortaleza, *changeovers* não são permitidos. Contudo, tal restrição foi relaxada para uma avaliação teórica do modelo proposto. Outras questões intrínsecas ao sistema real, tal como a consideração de serviços inviáveis, não foram contempladas¹⁰ nos presentes experimentos. No Capítulo 6, um caso de estudo será apresentado em detalhe.

Para a geração dos serviços, foram consideradas as seguintes regras:

- A jornada de trabalho é de 07h20, que é o valor mínimo remunerado à tripulação, sem a adição de horas extras. Na programação, busca-se gerar serviços com esta exacta duração, ou o mais próximo possível desta.
- É obrigatório um intervalo (não remunerado) para lanche, que deve ter uma duração mínima de 30 minutos. O intervalo de lanche deve, preferencialmente, estar compreendido entre 30 minutos e 1 hora.
- Os pontos de rendição ocorrem nos términos das viagens.

Optou-se por gerar serviços heurísticamente, conforme o procedimento ilustrado no Quadro 5.

Quadro 5: Procedimento para geração de serviços.

```

Enquanto  $k \leq w$  faça
  para  $i=1$  até  $m$  faça
    para  $j=1$  até  $m$  faça
      alocar viagens aos serviços com a menor folga possível,
      alocando uma folga por serviço
    fim-do-para
  fim-do-para
fim-do-enquanto

```

¹⁰Na programação de serviços de pessoal tripulante em Fortaleza, um acordo entre a empresa que gere o sistema de transportes colectivos e os tripulantes permite que uma parcela dos serviços a serem realizados não satisfaça todas as regras impostas na legislação. Neste capítulo, serviços inviáveis não serão considerados, ou seja, só serão gerados serviços viáveis.

Este algoritmo procura gerar serviços com uma folga para lanche, alocando as demais viagens com a menor folga possível (se possível, com folga nula). Deste modo, a heurística procura gerar bons serviços. A matriz A foi gerada com folgas com duração entre 30 e 60 minutos.

A enumeração da quantidade total de serviços factíveis, por meio de algoritmos do tipo *depth-first-search*, pode levar a uma quantidade de serviços que torne proibitivo o uso de métodos exactos. Muitos dos serviços gerados por um processo de enumeração podem ser considerados ruins (por exemplo, serviços com muitos tempos ociosos por parte dos tripulantes).

A heurística proposta destina-se a realizar uma amostragem do conjunto total de serviços factíveis.

Ao se permitir a geração de poucos serviços, as soluções geradas apresentam muitos *leftovers*. Ao serem gerados muitos serviços, aumenta-se demasiadamente a quantidade de variáveis e restrições do modelo. Em alguns experimentos, verificou-se que esse acréscimo de variáveis não implica nenhuma melhoria no valor da solução óptima obtida.

Como critério de paragem do algoritmo, utilizou-se um número máximo de iterações igual à $10m$ (em que m é o número de viagens). Não foram gerados serviços com horas-extras, mas apenas serviços com uma duração máxima de 07h20.

As instâncias foram rodadas no *software* LINGO, em um computador com processador Genuine Intel 1.86 GHz e 1GB de memória RAM.

Na Tabela 13, são apresentados os resultados dos experimentos. A coluna 1 consiste na identificação da instância, que é representada pelo número da linha. A coluna 2 apresenta o número de viagens da linha. A coluna 3 apresenta o número de serviços gerados. Na coluna 4, é apresentado o tempo de ciclo da linha¹¹. Deve-se ressaltar que, no sistema de transporte público por autocarros de Fortaleza, o tempo de ciclo varia ao longo do dia. Portanto, é apresentada a moda da variável tempo de ciclo¹². Na coluna 5, é apresentada a densidade da matriz de serviços gerados.

¹¹O tempo de ciclo de uma linha consiste no tempo decorrido entre o início e o final da execução de uma viagem desta linha.

¹²Por questões de carácter operacional, em Fortaleza, o tempo de ciclo não é constante em uma mesma linha. Nos momentos nos quais a quantidade de passageiros transportada é menor, utilizam-se viagens com tempos de ciclo menores. Usualmente, tais viagens mais curtas têm lugar no início do dia ou no final da noite e seu número é pequeno diante do número total de viagens. Assim, a média e a moda acabam por tender para um mesmo valor. Para se trabalhar com um número inteiro, emprega-se a moda.

Na coluna 6 é apresentada a quantidade máxima de colunas (serviços) a comporem a solução. Na coluna 7, apresenta-se o número de serviços utilizado na solução. Na coluna 8, apresenta-se o número de veículos obtido na solução óptima do modelo. Na coluna 9, consta o número de tramos descobertos. Na coluna 10, é apresentado o percentual de cobertura das viagens.

Na coluna 11, consta o tempo de processamento para obtenção da solução óptima. Na coluna 12, consta o número de nós do *branch-and-bound*. Por fim, na coluna 13, é apresentado o número de iterações do *branch-and-bound*.

A título de ilustração, será considerada a instância 406. A programação dos veículos (primeira parte da função objectivo) é apresentado na Tabela 14. A programação dos tripulantes (segunda parte da função objectivo) é apresentada na Tabela 15.

Para a cobertura das 85 viagens programadas na tabela de horários, foi necessário um número mínimo de 12 veículos. Em termos médios, cada veículo efectua 7 viagens; entretanto, percebe-se um certo desbalanceamento entre as viagens realizadas pelos veículos 2 e 3. Enquanto o veículo 2 realiza 9 viagens, o veículo 3 efectua apenas 4 viagens.

Na prática, as duas últimas viagens programadas pelo veículo 2 (viagens 80 e 85) poderiam ser alocadas para o veículo 3 (após a viagem 65), de modo a equilibrar a taxa de utilização dos recursos disponíveis. Constata-se que existem outros critérios de cunho operacional a serem considerados nas soluções. Neste exemplo, outro objectivo seria minimizar o desvio entre o número médio de viagens para cada veículo. Outros objectivos serão destacados no Capítulo 5.

A solução representada na Tabela 15 contém *changeovers*. Tomando como exemplo o serviço 40: na sua primeira viagem (viagem 40), este serviço é executado no veículo 1, e as três viagens subseqüentes (viagens 51, 61 e 73) são realizadas no veículo 12.

A segunda parcela da função objectivo diz respeito aos tramos não cobertos pelos serviços disponíveis. No caso da instância 406, as viagens 7, 13, 15, 19 e 48 não puderam ser cobertas pelos serviços regulares gerados sem a sobreposição dos mesmos, sendo requeridos outros meios (como, por exemplo, serviços com horas extras, aproveitamento de serviços menores) para sua cobertura.

Tabela 13: Características das instâncias e resultados do *solver*.

1. Instância	2. m	3. n	4. ciclo (min)	5. ρ (%)	6. d	7. $ x $	8. # veículos	9. z	10. c (%)	11. t (s)	12. # nós (B&B)	13. número de iterações
406	85	712	110	4,5	21	21	12	5	94,1	2	8	1668
504	64	446	48	13,9	7	6	4	10	84,4	5	84	20211
905	49	382	72	12,0	8	7	4	7	85,7	4	647	22018
13	75	530	55	10,6	9	8	5	11	85,3	45	11217	216440
833	80	344	84	6,1	15	14	8	10	87,5	5	252	19854
15	111	882	56	7,1	14	12	7	15	86,5	137	12489	587530
501	64	450	48	13,9	7	6	4	10	84,4	9	144	35892
407	67	608	94	5,8	14	14	10	11	83,6	3	189	9558
466	74	515	52	10,7	9	8	5	11	85,1	4	24	7902
609	69	513	90	6,2	14	14	9	9	87,0	1	0	330
907	72	550	72	8,2	12	11	6	7	90,3	4	49	10848
810	64	435	48	14,0	7	6	4	10	84,4	6	31	19620
605/606	86	766	100	4,4	20	19	14	11	87,2	7	133	21740
81	59	392	48	15,2	6	5	4	14	76,3	2	16	8780
70	83	744	110	4,6	21	20	11	7	91,6	1	0	514
102	104	741	50	7,6	12	12	6	9	91,3	2	0	600
201	73	505	62	9,5	10	9	6	10	86,3	3	26	10206
411	84	620	62	8,2	12	10	8	15	82,1	7	232	15520
316	55	421	91	7,4	12	12	7	7	87,3	1	0	255
913	80	554	42	12,4	8	7	4	10	87,5	21	1969	94485

Tabela 14: Resultado da programação de veículos para a instância 406 – um depósito.

Veículo	Viagens cobertas
1	1 – 11 – 23 – 40 – 52 – 63 – 75
2	2 – 13 – 25 – 38 – 46 – 60 – 72 – 80 – 85
3	3 – 15 – 29 – 65
4	4 – 16 – 28 – 36 – 47 – 55 – 67 – 77
5	5 – 17 – 27 – 37 – 45 – 62 – 74 – 81
6	6 – 18 – 31 – 42 – 50 – 59 – 71
7	7 – 19 – 30 – 48 – 56 – 68 – 78 – 83
8	8 – 20 – 32 – 44 – 53 – 64 – 76 – 82
9	9 – 21 – 34 – 43 – 57 – 69
10	10 – 22 – 35 – 49 – 58 – 70 – 79 – 84
11	12 – 24 – 33 – 41 – 54 – 66
12	14 – 26 – 39 – 51 – 61 – 73

Tabela 15: Resultado da selecção de serviços para a instância 406 – um depósito.

Serviço	Viagens cobertas
40	40 – 51 – 61 – 73
142	58 – 70 – 81
144	60 – 72 – 82
147	63 – 75 – 83
148	64 – 76 – 84
172	3 – 14 – 25 – 36
175	6 – 18 – 28 – 39
178	9 – 21 – 30 – 41
179	10 – 22 – 31 – 42
180	11 – 23 – 32 – 43
206	37 – 45 – 53 – 67
225	56 – 68 – 78 – 85
510	1 – 8 – 20 – 29
511	2 – 12 – 24 – 33
513	4 – 16 – 26 – 34
514	5 – 17 – 27 – 35
547	38 – 46 – 54 – 65
553	44 – 52 – 62 – 74
556	47 – 55 – 66 – 77
558	49 – 57 – 69 – 79
559	50 – 59 – 71 – 80

Para as vinte instâncias analisadas, em quinze delas a cardinalidade do conjunto x foi inferior à quantidade máxima de colunas d . Nestes casos, a restrição (4.5) não foi activa na solução óptima. Em termos operacionais, isto significa que a quantidade de serviços requerida para a cobertura dos tramos foi menor do que a estimada. Com a redução da quantidade de serviços, espera-se uma redução do número de tripulantes.

Deve-se observar que o valor de d é uma estimativa da quantidade mínima de serviços requerida para cobrir as viagens de uma dada linha, que pode ser calculada multiplicando-se o tempo de ciclo pelo número de viagens e dividindo-se esse resultado pela duração da jornada de trabalho.

Deve-se ainda observar outro aspecto da natureza multiobjectivo do problema. O parâmetro d é usado na restrição (4.5). Contudo, outro objectivo interessante seria minimizar a quantidade de colunas na solução. Esse objectivo possui uma relação de *trade-off* com o objectivo maximizar a cobertura dos tramos. O valor de d é um instrumento de análise de sensibilidade que pode ser de grande importância.

Em uma análise geral dos experimentos realizados, pode-se salientar que o modelo se comportou de modo satisfatório, pois permitiu a obtenção de soluções ótimas em baixos tempos de processamento: para as 20 instâncias analisadas, obteve-se um tempo médio de processamento igual a 13 segundos, muito satisfatório, na prática.

Deve-se enfatizar que as instâncias de Fortaleza são de pequeno porte, pois os veículos andam em uma só linha. No entanto, se os *changeovers* fossem de facto permitidos, o modelo teria milhares de linhas e milhões de colunas, dificultando bastante o planeamento. Assim, percebe-se que não permitir *changeovers* pode viabilizar o processo de planeamento operacional dos autocarros em Fortaleza.

No caso do modelo de máxima cobertura estudado no capítulo anterior, como os *changeovers* eram proibidos, os veículos e tripulantes se comportavam como uma única entidade e pôde-se considerar que um par veículo-tripulante deveria cobrir todas as viagens. No caso do modelo proposto no presente capítulo, tendo em vista que os *changeovers* são permitidos, os serviços de pessoal tripulante podem ser executados em viaturas distintas. O problema consiste na cobertura da procura com múltiplos recursos: cobertura de viagens por meio de veículos e cobertura de viagens por meio de serviços.

4.3 Uma nova abordagem heurística para o VCSP com *changeovers*

4.3.1 Estrutura da abordagem proposta

No caso do VCSP com um depósito, constatou-se que, para as 20 instâncias analisadas, a média dos tempos de processamento foi de 13 segundos, evidenciando que a formulação permitiu a obtenção de soluções ótimas em tempos computacionais aceitáveis. No entanto, deve-se salientar a importância de se usar métodos heurísticos para este tipo de problema:

- Os métodos heurísticos são mais flexíveis, no que concerne à consideração de novas restrições ou de novas funções objectivo.
- Os métodos heurísticos permitem que as abordagens de planeamento desenvolvidas sejam adaptadas ao paradigma multiobjectivo com uma maior facilidade.
- Devem ser desenvolvidos algoritmos e interfaces que facilitem o uso por parte dos planeadores em empresas de transporte público.

- O uso de *solvers* comerciais (baseados em programação matemática) requer, em geral, um conhecimento aprofundado de técnicas de optimização, dificultando o seu uso no quotidiano das empresas de transporte público.

Assim, será desenvolvida uma abordagem heurística para o VCSP. Tendo em vista o grande número de variáveis do problema, a abordagem heurística proposta seguirá o princípio da decomposição desenvolvido por Steinzen *et al.* (2007), em que o *vehicle scheduling* e o *crew scheduling* são decompostos, mas há uma concatenação entre os dois problemas.

A seguir, na Figura 3, é ilustrada a abordagem proposta para o VCSP com *changeovers*. Nesta proposta, o problema é solucionado sobre o uso dos dois recursos (viaturas e tripulantes) na cobertura das viagens. Uma heurística GRASP é proposta para a programação das viaturas e um Algoritmo Genético (*Genetic Algorithm – GA*) é proposto para a programação das tripulações. Deve-se observar que a intersecção entre os algoritmos consiste nos procedimentos de melhoramento do GRASP, que se destinam a integrar os blocos de viaturas aos serviços.



Figura 3: Estrutura da abordagem proposta para o VCSP com *changeovers*.

A inovação desta abordagem está no facto de que a programação de veículos e de tripulações ocorre directamente sobre as viagens ao contrário do que ocorre com a abordagem tradicional, em que os serviços são programados sobre os blocos gerados para as viaturas.

4.3.2 Heurística GRASP proposta para a programação de veículos

A seguir, será apresentada a heurística utilizada para a programação de veículos. O GRASP proposto é dividido em duas fases: uma fase de construção de uma programação de veículos inicial, e uma fase de melhoria, com vistas a concatenar a programação dos veículos com a programação de tripulações.

Fase de construção

A heurística construtiva é descrita no Quadro 6. As variáveis da fase de construção são m viagens, nv veículos e um vector RCL o qual consiste na lista de candidatos restrita. A RCL armazena os veículos que podem ser alocados a uma dada viagem. A cardinalidade da RCL é representada pela variável k . Uma função gulosa Γ é utilizada para seleccionar veículos armazenados na RCL.

O número de veículos é representado pela variável nv . Inicialmente, é requerida a obtenção de um limite superior para o número de veículos necessário para cobrir um conjunto de viagens. Esta aproximação pode ser obtida dividindo-se o tempo de ciclo da linha pelo seu *headway*¹³. Tendo em vista que o *headway* usualmente varia ao longo do dia, pode-se calcular essa quantidade de veículos separadamente para diversos períodos da operação da linha. Na fase de construção aqui reportada, foi adoptado o menor *headway*¹⁴.

Quadro 6: Heurística construtiva para a programação de veículos.

```

para  $i=1$  até  $m$  faça
  para  $j=1$  até  $nv$  faça
    se  $\Gamma(j) \geq 0$  e  $\Gamma(j) \leq \min(\Gamma)$  então
       $k=k+1$ ;
       $RCL(k)=j$ ;
    fim-do-se
  fim-do-para
  Sortear um veículo da RCL
  Aloca a viagem ao veículo sorteado
fim-do-para

```

Em seguida, é criada uma matriz com nv linhas para armazenar as viagens da tabela de horários cobertas por cada veículo. Deve-se salientar que o número de

¹³ Na área de transportes, usa-se o termo *headway* para representar a frequência de um serviço, ou seja, o intervalo de tempo decorrido entre a passagem de dois veículos de uma mesma linha.

¹⁴ Ao se considerar o menor *headway* obtém-se uma estimativa da quantidade de veículos que contempla a pior situação possível. Deste modo, não há como a estimativa de nv ser menor que a quantidade real de veículos requerida para a execução das viagens.

veículos requerido para cobrir as viagens pode vir a ser menor do que nv , visto que este número é uma estimativa.

Uma função gulosa Γ é usada como critério de alocação de viagens a um veículo. Para cada veículo, calcula-se a diferença entre o tempo de término da última viagem alocada naquele veículo e o tempo de início das demais viagens. Se a diferença entre o término da última viagem alocada ao veículo j e o início da viagem i é maior ou igual a zero (ou seja, $\Gamma(j) \geq 0$), o veículo j pode executar a viagem i . Portanto, o veículo j é inserido na RCL.

Para não serem gerados blocos com folgas grandes, adoptou-se o seguinte critério: serão inseridos na RCL veículos que tenham folga mínima em relação à viagem i (isto é, $\Gamma(j) \leq \min(\Gamma)$). Procedendo-se dessa forma, espera-se que o número de veículos nv fornecido pela heurística seja mínimo.

Fase de melhoramento

Após a solução inicial construída, tem-se uma programação de veículos. Essa programação não leva em conta os serviços de tripulantes, por isso, na prática, não seria exequível. A fase de melhoria da heurística proposta consiste em concatenar a programação de veículos com a programação de tripulações. Portanto, é na fase de melhoramento do algoritmo de programação de veículos que reside a integração entre os dois *schedulings*.

Tal integração consiste em distribuir, de uma forma mais equilibrada, as viagens entre os veículos. Além disso, a fase de melhoramento tenta reduzir os tempos que os veículos ficam à espera de viagens.

A fase de *melhoramento* é composta por dois tipos de movimentos: movimento de inserção caudal e movimento de inserção de blocos.

O movimento de inserção caudal será descrito a seguir, no Quadro 7. A denominação “caudal” vem da propriedade do movimento de realocar as últimas viagens de um bloco para outro, de modo a equilibrar as viagens por veículo.

Quadro 7: Algoritmo do movimento de inserção caudal.

Passo 1: Gerar uma lista com as últimas viagens de cada bloco.
Passo 2: Se o tempo de início da última viagem do bloco i é maior ou igual ao tempo de término da viagem j , pode haver a reafectação.
Passo 3: Gerar matriz de conectividades entre os blocos.
Passo 4: Se os blocos são conexos e se o número de viagens do bloco i é maior do que o número de viagens do bloco j mais uma viagem, realocar. Actualizar a matriz de conectividades.

Se o número de viagens alocadas ao i -ésimo veículo é maior do que o número de viagens alocadas ao j -ésimo veículo mais uma viagem, o algoritmo de melhoria irá realocar as viagens. A seguir, na Figura 4, é ilustrada uma aplicação do movimento de inserção caudal em uma instância com 12 veículos e 85 viagens. As viagens 84, 79, 82 e 85, as quais, *a priori*, eram alocadas, aos veículos 1, 6, 10 e 11, respectivamente, foram realocadas aos veículos 3, 7, 4 e 2, respectivamente.

O movimento de inserção de blocos é descrito no Quadro 8. Este procedimento de melhoria tem como meta equilibrar as quantidades de viagens alocadas entre os veículos, por meio da inserção de subconjuntos de viagens (blocos) dos veículos mais utilizados para veículos menos utilizados.

A seguir, na Figura 5, é ilustrado um exemplo de movimento de inserção de blocos. Pode-se observar que há uma discrepância da quantidade de viagens alocadas aos veículos 1 e 2, bem como grande intervalo, no veículo 2, entre as viagens 16 e 33. Deste modo, o bloco de viagens 22 – 25 – 28 pode ser removido do veículo 1 e inserido no veículo 2, distribuindo, de forma mais equilibrada, as viagens entre os dois veículos.

1	7	19	31	39	47	55	66	78	84	1	7	19	31	39	47	55	66	78		
2	2	12	24	33	41	49	57	69	2	2	12	24	33	41	49	57	69	85		
3	15	27	37	45	53	63	75	3	15	27	37	45	53	63	75	84				
4	5	17	64	76	4	5	17	64	76	82										
5	4	16	26	35	43	51	61	73	5	4	16	26	35	43	51	61	73			
6	11	23	32	40	48	56	68	79	6	11	23	32	40	48	56	68				
7	1	8	20	29	58	70	7	1	8	20	29	58	70	79						
8	10	22	60	72	81	8	10	22	60	72	81									
9	9	21	30	38	46	54	65	9	9	21	30	38	46	54	65					
10	6	18	28	36	44	52	62	74	82	10	6	18	28	36	44	52	62	74		
11	3	14	25	34	42	50	59	71	80	85	11	3	14	25	34	42	50	59	71	80
12	13	67	77	83	12	13	67	77	83											

Figura 4: Exemplo de movimento de inserção caudal.

Quadro 8: Algoritmo do movimento de inserção de blocos.

Passo 1: Calcular uma matriz com os <i>gaps</i> entre os tempos de término e de início de todas as viagens.
Passo 2: Determinar os veículos com maior e menor número de viagens alocadas.
Passo 3: Localizar o maior <i>gap</i> do veículo com menor número de viagens.
Passo 4: Determinar o tamanho do bloco das viagens do veículo com maior número de viagens e inserir o bloco na brecha obtida no Passo 3.

1	3	7	11	15	19	22	25	28	31	35	39	42
2	4	8	12	16	33	37						

1	3	7	11	15	19	31	35	39	42
2	4	8	12	16	22	25	28	33	37

Figura 5: Exemplo de movimento de inserção de blocos.

A heurística GRASP proposta para a programação de veículos é ilustrada a seguir, no Quadro 9. Nessa variante o GRASP tem apenas como parâmetro, o número de iterações. Inicialmente, a heurística construtiva gera blocos de viagens, que, *a posteriori*, são refinados pelos movimentos de inserção caudal e o movimento de inserção de blocos.

Quadro 9: Heurística GRASP para a programação de veículos.

```

Parâmetros: num_iter
Fase de construção
Movimento de inserção caudal
i ← 0
enquanto i ≤ num_iter faça
    Movimento de inserção de blocos
    i ← i + 1
fim-do-enquanto

```

4.3.3 Algoritmo Genético proposto para a programação de tripulações

Para a componente do *crew scheduling* do VCSP, foi desenvolvido um Algoritmo Genético (GA) híbrido com base no problema de máxima cobertura. Este GA é dito híbrido, pois ao algoritmo clássico é acrescentado uma heurística de melhoramento.

A seguir, serão discutidas as justificativas pela adoção do GA, em detrimento do algoritmo GRASP apresentado no Capítulo 3.

É importante destacar que o algoritmo GRASP desenvolvido para o MCP, que foi reportado no Capítulo 3 desta tese, também foi testado nas instâncias apresentadas no presente capítulo. Apesar de ter apresentado um excelente comportamento para 40 instâncias da OR-Library, o GRASP não foi eficaz na resolução das instâncias reportadas no Capítulo 4. De facto, nos conjuntos de dados da OR-Library, as quantidades de linhas cobertas por cada coluna eram em geral muito diferentes. Nas instâncias consideradas acima, como não são considerados serviços com horas extras, as quantidades de linhas cobertas por cada coluna tendem a ter um valor constante. Deste modo, a função gulosa de GRASP não resulta, e a fase de construção funciona como uma busca aleatória.

Ao inserir colunas aleatoriamente na solução, verifica-se que a quantidade de colunas inseridas é bem inferior ao limite superior d . Isso ocorre devido às restrições de igualdade, pois, ao se inserir uma determinada coluna, muitas outras são excluídas do problema, para evitar a ocorrência de *overcovers*.

Como uma alternativa à função gulosa apresentada no Capítulo 3, também foi desenvolvida uma função gulosa baseada nos conflitos entre colunas. Ao se tentar inserir as colunas que apresentem uma menor quantidade de conflitos com as demais colunas, tende-se a inserir uma maior quantidade de colunas na solução. Essa abordagem tendeu a apresentar resultados melhores do que a função gulosa anterior, mas, ainda assim, não se conseguiu obter as soluções ótimas das instâncias.

Desenvolveu-se, então, uma abordagem do tipo *Path Relinking* – PR. A metaheurística PR, que foi desenvolvida por Glover *et al.* (2000), vem sendo amplamente estudada e vários relatos vêm salientando os benefícios da hibridização do GRASP com o PR (RESENDE e RIBEIRO, 2008).

O PR é uma abordagem populacional, pois um conjunto de soluções elite é gerado e caminhos entre boas soluções são investigados, de forma a obter melhores soluções. No caso binário, o PR assemelha-se ao cruzamento uniforme de um GA; porém, fazendo uma busca bem mais profunda na vizinhança de uma solução do que esse operador genético.

Com base nas soluções elite geradas pela heurística gulosa que minimiza os conflitos entre as colunas presentes na solução, uma série de ótimos locais era gerada. O PR não conseguia aprimorar estes ótimos locais, pois eles padeciam de pouca diversidade.

Percebeu-se, então, a necessidade de se implementar um algoritmo que trabalhasse com uma população de soluções bem diversificadas, tendo-se, por isso, optado por um Algoritmo Genético, o qual trabalha com uma população de indivíduos.

Os indivíduos de uma população são recombinações entre si, através de cruzamento, e herdam características dos pais que os vão fazendo mais aptos ao longo das gerações. Do ponto de vista matemático, os indivíduos podem ser vistos como possíveis valores de uma função; logo, a aptidão dos indivíduos pode ser expressa em termos do valor dessa função.

Os Algoritmos Genéticos propostos por Holland (1975) foram, ao longo dos anos, amplamente estudados. Todavia, sob o ponto de vista de sua aplicação prática, notadamente no campo da Engenharia, o trabalho de Goldberg (1989) foi um marco

sobre o tema. A partir deste trabalho, os GAs passaram a ser aplicados, com sucesso, em muitos problemas reais.

De acordo com Gen e Cheng (2000, p.8): “Algoritmos Genéticos são uma classe de métodos de busca de objectivo geral, combinando elementos de busca dirigida e estocástica, os quais podem fazer um bom balanço entre intensificação e diversificação”.

Os Algoritmos Genéticos são uma heurística de busca global que atua com dois tipos de paralelismo: um explícito e outro implícito. O paralelismo explícito consiste na geração e na avaliação de conjuntos de soluções, denominados populações, que permitem a busca em diferentes regiões do espaço de soluções. O paralelismo implícito consiste na geração de bons esquemas.

Conforme Linden (2006), um esquema consiste em um padrão descrevendo um subconjunto dentro o conjunto de todos os indivíduos possíveis. Para cada indivíduo da população, o GA manipula uma grande quantidade de esquemas (dezenas, centenas ou mesmo milhares) e vai seleccionando os melhores dentre todos eles.

Os GAs combinam mecanismos de intensificação e diversificação. Por trabalhar com populações de soluções, um GA explora diferentes regiões do espaço de busca e por causa da pressão seletiva, o GA tende a migrar para regiões promissoras do espaço de soluções e concentrar a busca na recombinação de indivíduos de alta aptidão. O pseudocódigo de um GA genérico é apresentado no Quadro 10.

Quadro 10: Algoritmo Genético padrão.

<p>Passo 1: Gerar uma população de soluções.</p> <p>Passo 2: Avaliar as soluções geradas.</p> <p>Enquanto um critério de parada não for satisfeito, faça</p> <p style="padding-left: 20px;">Passo 3: Seleccionar um conjunto de k pais.</p> <p style="padding-left: 20px;">Passo 4: Realizar o cruzamento de k pais, com uma dada probabilidade.</p> <p style="padding-left: 20px;">Passo 5: Realizar mutação das w soluções geradas, com uma dada probabilidade.</p> <p style="padding-left: 20px;">Passo 6: Avaliar a aptidão das soluções geradas.</p> <p style="padding-left: 20px;">Passo 7: Actualizar a população.</p> <p>Fim-do-enquanto</p> <p>Imprimir a melhor solução obtida</p>

A seguir, na Figura 6, é ilustrada uma visão esquemática do funcionamento de um GA.

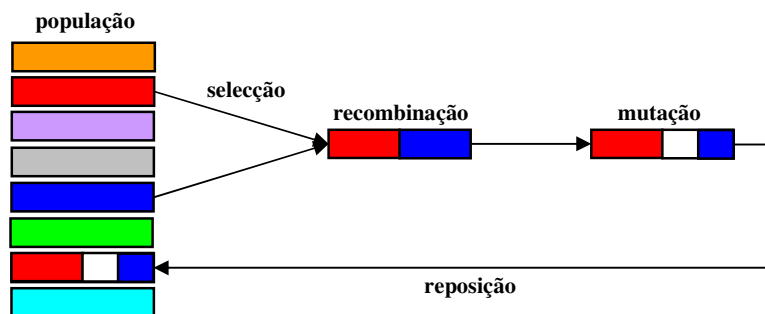


Figura 6: Visão esquemática de um Algoritmo Genético.

No campo dos problemas de otimização em transporte público, Wren e Wren (1995) fazem um estudo inicial acerca da viabilidade do uso dos GAs na programação de motoristas de autocarros. Os resultados iniciais do estudo mostraram o potencial deste tipo de algoritmos nessa classe de problemas.

Conforme já afirmado anteriormente, o problema de programação de tripulações usualmente recai em problemas de cobertura e particionamento. Beasley e Chu (1996) apresentam um GA para o SCP. Levine (1996) e Beasley e Chu (1995, 1998) apresentam GAs para o SPP. Nestes trabalhos, a codificação adoptada foi a binária.

Beasley e Chu (1996), defendem que a codificação binária é a melhor no caso do SCP. Para estes autores, a representação não binária pode tornar-se ambígua, pois uma mesma solução pode ser representada de diferentes formas. Todavia, Emereev (1999) e Dias *et al.* (2002) demonstram que a codificação não-binária pode, em alguns casos, ser mais eficiente do que a representação binária.

O GA proposto neste trabalho consiste em uma adaptação do algoritmo de Beasley e Chu (1995, 1998), adaptando à formulação de máxima cobertura.

Conforme Coley (2010), os GAs nem sempre são eficientes na obtenção de soluções óptimas; conseguem, contudo, prover uma diversificação da busca no espaço de pesquisa e alcançar soluções sub-óptimas. Para aprimorar os resultados obtidos por um GA, a literatura sugere a hibridização de tais algoritmos com outras técnicas de otimização, resultando nos chamados *Hybrid Genetic Algorithms* – HGA's.

No campo da otimização combinatória, as técnicas comumente hibridizadas com o GA são as técnicas de pesquisa local e heurísticas de reparos. No âmbito dos HGA's, também podem ser desenvolvidos operadores genéticos diferenciados, que sejam planeados de acordo com as características específicas do problema modelado.

No GA proposto neste trabalho, os procedimentos DROP e ADD destinam-se, respectivamente, a corrigir a viabilidade de uma solução após recombinação, para, em seguida, efectuar uma busca na vizinhança da solução corrigida.

A seguir, no Quadro 11, faz-se uma breve descrição do Algoritmo Genético proposto para resolução do MCP.

Quadro 11: Algoritmo Genético proposto.

Passo 1: Gerar aleatoriamente uma população de soluções viáveis.
Passo 2: Avaliar as soluções geradas.
Enquanto geração actual \leq número máximo de gerações, faça
 Passo 3: Efectuar selecção por torneio.
 Passo 4: Realizar o cruzamento uniforme com $p_c=100\%$.
 Passo 5: Realizar mutação da prole com $p_m=100\%$.
 Passo 6: Aplicar os procedimentos **DROP** e **ADD**.
 Passo 7: Avaliar a aptidão da solução gerada.
 Passo 8: Substituir o pior indivíduo da população pela solução gerada.
Fim-do-enquanto
Imprimir a melhor solução obtida

Codificação

Os principais benefícios da adoção da codificação binária para o problema de máxima cobertura, são os seguintes:

- facilitam o projecto, a implementação e a aplicação dos operadores genéticos;
- permitem o uso de estruturas matriciais para verificar a qualidade e a viabilidade de uma solução.

Por outro lado, sabe-se que a codificação binária apresenta as seguintes desvantagens:

- devido a matriz A do problema de máxima cobertura geralmente ser bastante esparsa, o uso da codificação binária requer uma maior quantidade de memória para armazenar as instâncias e as soluções,
- a codificação binária porta menos informação sobre as soluções do que a codificação não-binária.

Diante dos benefícios e limitações da codificação binária, optou-se pelo uso da mesma, conforme ilustrado na Figura 7. Seja um vector de dimensão n , em que n é o número de variáveis de decisão (ou seja, quantidade de serviços gerados), se $x_j = 1$ a j -

ésima variável de decisão fará parte da solução; caso contrário, se $x_j = 0$, a j -ésima variável de decisão não fará parte da solução.

gene	1	2	3	...	$n-1$	n
bit	1	0	0	...	1	1

Figura 7: Codificação binária do cromossoma de um indivíduo.

Avaliação das soluções

Uma solução pode ser vista como um vector \mathbf{x} . O vector \mathbf{w} armazena as relações de cobertura de uma dada solução do problema. A relação entre os vectores \mathbf{x} e \mathbf{w} é dada por $\mathbf{w} = \mathbf{A} \times \mathbf{x}^T$. Se $w_i = 0$, então a i -ésima linha está descoberta na solução corrente; se $w_i = 1$, então a i -ésima linha é coberta por apenas uma coluna na solução actual; e, por fim, se $w_i \geq 2$ então ocorre pelo menos um *overcover*.

A função de avaliação é dada pela soma dos elementos do vector \mathbf{w} iguais à zero (ou, seja, os *leftovers*).

Deve-se observar que soluções inviáveis, isto é, *overcovers* e soluções com mais de d colunas, não são permitidas.

Geração da população inicial

A população inicial é gerada de forma aleatória, levando em conta a viabilidade das soluções geradas, de modo que são gerados apenas indivíduos viáveis. A seguir, no Quadro 12, é ilustrado o procedimento para a construção de uma solução. No procedimento a seguir, AV é um vector que armazena as colunas disponíveis, que podem ser inseridas a cada interacção.

Quadro 12: Procedimento para geração de um indivíduo da população inicial.

```

AV ← xj, ∀ j= 1, ..., n
col ← 0
enquanto AV ≠ ∅ ou col ≤ d faça
    seleccionar aleatoriamente uma coluna xj
    solução ← xj
    Actualiza AV
    col ← col +1
fim-do-enquanto

```


No início do algoritmo todas as colunas pertencem à lista AV e no máximo d colunas podem ser adicionadas à solução.

Em seguida, uma coluna j , seleccionada aleatoriamente, é adicionada à solução. A lista AV é então actualizada, sendo removida a coluna inserida e as demais colunas que cobrirem as mesmas linhas cobertas pela coluna inserida, de modo que as próximas inserções satisfaçam as restrições de igualdade.

As operações descritas são repetidas enquanto houver colunas disponíveis ou enquanto o número máximo de colunas na solução não tiver sido excedido.

O número de soluções geradas é controlado por um parâmetro *maxpop* que representa a quantidade de indivíduos na população.

Seleccção

A seleccção dos pais a serem cruzados é efectuada por torneio binário: seleccionam-se aleatoriamente 4 indivíduos da população e são realizados torneios com dois indivíduos. Os torneios funcionam da seguinte forma: os quatro indivíduos seleccionados são divididos em 2 pares. Efectua-se um torneio em cada par, comparando-se as aptidões das soluções. Os indivíduos mais aptos, isto é, aqueles com menores valores da função objectivo, serão escolhidos como pais.

Cruzamento

O operador de cruzamento (*crossover*) é utilizado nos pais seleccionados para reprodução. O tipo de cruzamento utilizado foi o uniforme, ilustrado na Figura 8. Para cada gene da solução filho a ser gerada, é realizado um experimento, regido por uma distribuição de Bernoulli, no qual uma variável aleatória discreta b recebe um valor 0 ou 1. Se b igual a 0, o filho receberá o valor correspondente ao gene do primeiro pai. Se b igual a 1, o filho receberá o valor correspondente ao gene do segundo pai.

Pai 1
1 0 0 1 1 1
Pai 2
0 1 1 0 1 0
Filho gerado
0 0 1 1 1 1

Figura 8: Cruzamento uniforme.

O cruzamento uniforme foi adoptado por dois motivos: (i) possui fácil implementação computacional; e (ii) é capaz de combinar todo e qualquer esquema existente.

Mutação

A operação de mutação ocorre da seguinte forma: é gerado um número aleatório compreendido no intervalo $[0,100]$ e se este é menor ou igual à probabilidade de mutação considerada, o operador de mutação é acionado. É gerado um número aleatório r entre $[0,n]$ e o bit correspondente ao número aleatório gerado é invertido (por exemplo, se $x_r = 1$ muda-se x_r para 0 e se $x_r = 0$ muda-se x_r para 1).

A seguir, na Figura 9, é apresentado um exemplo simples de mutação.

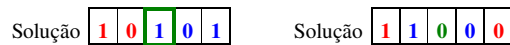


Figura 9: Operador de mutação.

Procedimento de melhoramento

Embora os indivíduos da população inicial sejam viáveis, após as operações de recombinação e mutação, esta viabilidade certamente será violada. Deste modo, é de fundamental importância a correção da inviabilidade da solução, bem como a melhoria da solução corrigida.

Neste sentido, foram adaptados para o problema de máxima cobertura, os procedimentos de melhoria propostos para o SPP por Beasley e Chu (1995, 1998).

O procedimento DROP é descrito no Quadro 13. DROP tem por objectivo eliminar os *overcovers*. Enquanto a soma dos elementos de w maiores que 1 for diferente de zero, existirão *overcovers* a serem corrigidos. A selecção das colunas a serem removidas da solução é realizada de forma aleatória, de modo a evitar alguma convergência prematura de colunas na solução.

Quadro 13: Procedimento DROP.

```

w ←  $A \times \mathbf{x}^T$ 
enquanto (soma( $w > 1$ ) ≠ 0) faça
  para  $i=1$  até  $n$ 
    se  $w_i \geq 2$  então
      selecciona aleatoriamente coluna a descobrir a linha
      remove a coluna na solução
      w ←  $A \times \mathbf{x}^T$ 
    fim-do-se
  fim-do-para
fim-do-enquanto

```

Pode-se observar que nesta heurística de reparos não é analisada a restrição inerente ao número máximo de colunas. Esta análise é realizada pelo procedimento ADD, descrito no Quadro 14, que actualiza a lista de colunas disponíveis (AV) e a quantidade de colunas da solução (*col*). ADD permite a inserção de colunas que não infringem as restrições de igualdade na prole gerada, cuja viabilidade foi devidamente corrigida.

Quadro 14: Procedimento ADD.

```

Actualiza AV
Actualiza col
enquanto AV ≠ ∅ ou col ≤ d faça
    seleccionar aleatoriamente uma coluna xj
    solução ← xj
    Actualiza AV
    col ← col + 1
fim-do-enquanto

```

Deve-se ressaltar que ADD não permite a geração de soluções inviáveis, de modo que não foi necessária a penalização de soluções que infringissem as restrições do problema, conforme ocorreu com o GRASP reactivo apresentado no Capítulo 3 da presente tese.

Reposição dos indivíduos da população

A operação de reposição consiste na substituição de elementos da população em uma geração passada pela prole gerada na geração corrente. O operador de reposição tem forte parcela de contribuição na eficiência da busca, visto que é ele que gere o processo de actualização da população. Se a população for actualizada de forma errada, regiões não-promissoras do espaço de busca podem ser visitadas constantemente ou a busca poderá ser confinada a pontos de mínimo (ou máximo) locais.

A estratégia de reposição dos indivíduos da população foi a incremental ou de estado estacionário (*steady state*). Nesta estratégia, cada novo indivíduo irá substituir um elemento da população. A estratégia *steady state* apresenta excelentes resultados, mas pode implicar uma convergência prematura da população.

Para mitigar o problema da convergência prematura da população, foi desenvolvido o seguinte procedimento para reposição das soluções, conforme ilustrado no Quadro 15.

Se a prole gerada tiver melhor aptidão do que o indivíduo com pior aptidão na população, a prole ingressa na população no lugar do pior indivíduo. Caso a prole gerada não tenha uma melhor aptidão do que o indivíduo de pior aptidão na população, ocorrerá a substituição com uma probabilidade p_a .

Caso só sejam aceitas soluções que incorram em melhoria, existe uma grande chance de o GA convergir para um ótimo local antes mesmo da execução de grande parte das gerações do algoritmo. Esse mecanismo permite a manutenção da diversidade da população ao longo das gerações do GA.

Quadro 15: Procedimento para reposição da população.

```

se fitness_offspring < fitness_worse então
  para j=1 até n faça
    pop(i_worse, j) ← offspring(j)
  fim-do-para
  fitness(i_worse) ← fitness_offspring
senão
  r ← random[0,100]
  se r ≤ pa
    para j=1 até n faça
      pop(i_worse, j) ← offspring(j)
    fim-do-para
    fitness(i_worse) ← fitness_offspring
  fim-do-se
fim-do-senão

```

Uma opção para a aceitação de soluções que degradam a qualidade da solução incumbente é a utilização, tal como ocorre na metaheurística *Simulated Annealing* (SA), do factor de probabilidade de Boltzmann. No SA, com base em um parâmetro de temperatura T e em um factor de redução de temperatura α , uma solução que deteriora a função objectivo pode ser aceita com uma dada probabilidade, a qual depende de T .

No início da busca, quando a temperatura é elevada, diversas soluções que deterioram a função objectivo são aceitas, implicando uma diversificação da amostragem das soluções. Ao longo das iterações do algoritmo, com o decréscimo da temperatura advindo de α , soluções que melhoram a função objectivo são preponderantemente aceitas, facto que intensifica a busca em uma vizinhança promissora.

Na opinião do autor, no caso de um GA, esse comportamento pode vir a ser indesejável. Nas primeiras gerações, o GA tem de ser seletivo na escolha de boas soluções (pressão seletiva), de modo a migrar para regiões promissoras do espaço de busca. A diversificação será obtida por meio da aplicação dos operadores genéticos. Ao contrário do que ocorre no SA, no GA, é nas últimas iterações que se necessita mais da aceitação de soluções piores, por forma a evitar a convergência prematura da população.

Deste modo, se justifica a adoção de uma probabilidade fixa para aceitação de soluções que não incorram em melhoria. Após testes computacionais, constatou-se que o valor do parâmetro p_a a ser adoptado seria de 5%. Valores pequenos de p_a não evitam a convergência prematura, enquanto valores elevados de p_a acabam por prejudicar a pressão selectiva do algoritmo.

Deve-se salientar que foi usada a estratégia do elitismo, isto é, a melhor solução da população é preservada para a geração seguinte.

4.4 Teste e avaliação dos algoritmos

Para cada instância, as heurísticas para programação de veículos e de tripulações foram rodadas em sequência, sendo chamadas no mesmo bloco de execução. Conforme relatado anteriormente, o GRASP destina-se à programação dos veículos e o GA destina-se à programação de tripulações. A seguir, na Figura 10, é ilustrada uma visão esquemática da aplicação dos algoritmos. Os resultados obtidos constam na Tabela 16.

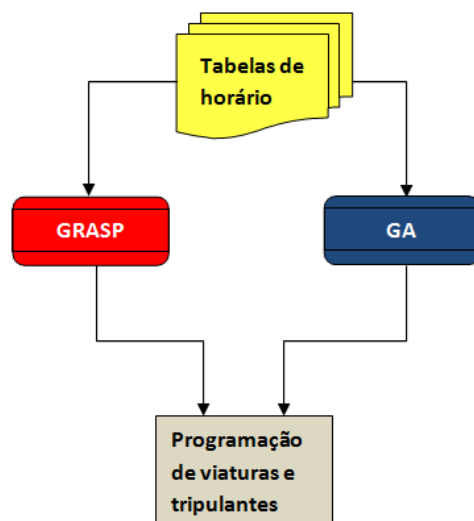


Figura 10: Visão esquemática da aplicação dos algoritmos.

Tabela 16: Soluções obtidas pela abordagem proposta.

Instância	teste	1	2	3	4	5	6	7	8	9	10
406	z	7	5	7	7	7	7	7	7	7	7
	nv	12	12	12	12	12	12	12	12	12	12
	t (s)	366,5	61,1	347,6	341,0	343,7	377,8	363,4	342,4	345,1	332,8
504	z	10	10	10	10	10	10	10	10	10	10
	nv	4	4	4	4	4	4	4	4	4	4
	t (s)	0,7	0,7	0,7	1,0	0,7	0,7	0,7	0,7	1,7	0,7
905	z	7	7	7	7	7	7	7	7	7	7
	nv	4	4	4	4	4	4	4	4	4	4
	t (s)	0,8	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6
13	z	11	11	11	11	11	11	11	11	11	11
	nv	5	5	5	5	5	5	5	5	5	5
	t (s)	1,3	2,0	1,1	1,7	1,1	1,1	1,1	1,1	1,5	2,6
833	z	10	10	10	10	10	10	10	10	10	10
	nv	8	8	8	8	8	8	8	8	8	8
	t (s)	4,9	7,4	5,4	10,2	6,9	5,4	6,9	6,4	8,7	6,8
15	z	15	15	15	15	15	15	15	15	15	15
	nv	7	7	7	7	7	7	7	7	7	7
	t (s)	46,8	26,7	38,0	29,0	33,7	18,0	19,3	15,8	36,5	38,2
501	z	10	10	10	10	10	10	10	10	10	10
	nv	4	4	4	4	4	4	4	4	4	4
	t (s)	0,7	0,7	0,8	1,2	0,9	0,7	0,8	0,7	0,7	0,7
407	z	11	11	11	11	11	11	11	11	11	11
	nv	10	10	10	10	10	10	10	10	10	10
	t (s)	17,3	16,1	13,9	8,4	2,4	9,9	4,3	13,6	25,1	5,2
466	z	11	11	11	11	11	11	11	11	18	11
	nv	5	5	5	5	5	5	5	5	5	5
	t (s)	3,8	14,8	102,3	115,6	6,9	48,5	3,4	17,8	113,5	12,3
609	z	13	13	13	13	9	13	9	9	13	13
	nv	9	9	9	9	9	9	9	9	9	9
	t (s)	151,0	149,9	151,7	158,3	18,5	159,0	14,2	123,2	148,4	153,5
907	z	8	7	12	8	8	7	8	8	7	8
	nv	6	6	6	6	6	6	6	6	6	6
	t (s)	139,8	6,1	135,5	133,2	140,6	11,4	134,9	142,2	16,3	153,0
810	z	10	10	10	10	10	10	10	10	10	10
	nv	4	4	4	4	4	4	4	4	4	4
	t (s)	1,1	1,1	0,9	1,1	0,8	0,6	0,8	0,6	1,2	0,8
605/606	z	11	11	11	11	12	11	11	12	12	11
	nv	14	14	14	14	14	14	14	14	14	14
	t (s)	105,1	35,3	26,8	20,1	348,5	37,2	37,1	374,6	373,0	29,6
81	z	14	14	14	14	14	14	14	14	14	14
	nv	4	4	4	4	4	4	4	4	4	4
	t (s)	1,6	0,5	0,5	1,8	2,2	0,7	0,9	11,2	0,6	5,4
70	z	9	10	10	9	10	9	10	10	10	10
	nv	11	11	11	11	11	11	11	11	11	11
	t (s)	357,9	363,4	353,2	360,7	371,7	366,3	358,3	366,9	360,9	372,2
102	z	9	9	17	9	9	10	9	9	9	9
	nv	6	6	6	6	6	6	6	6	6	6
	t (s)	24,5	37,4	297,2	20,5	67,7	297,2	178,8	11,6	28,6	15,9
201	z	10	10	10	10	10	10	10	10	10	10
	nv	6	6	6	6	6	6	6	6	6	6
	t (s)	4,7	3,4	3,6	3,3	3,6	3,3	3,9	3,8	1,1	3,2
411	z	15	15	15	15	15	15	15	15	15	15
	nv	8	8	8	8	8	8	8	8	8	8
	t (s)	166,2	14,7	8,5	4,3	8,6	7,1	9,8	7,0	5,8	2,8
316	z	7	7	7	7	7	7	7	7	7	7
	nv	7	7	7	7	7	7	7	7	7	7
	t (s)	2,6	8,5	27,9	3,2	5,7	45,0	10,9	77,3	58,7	3,2
913	z	10	10	10	10	10	10	10	10	10	10
	nv	4	4	4	4	4	4	4	4	4	4
	t (s)	49,2	16,8	47,9	20,7	13,8	1,9	13,4	1,1	5,6	95,5

As heurísticas foram implementadas no MATLAB. Os testes foram realizados em um processador Genuine Intel 1.86 GHz com 1GB de memória RAM.

A seguir, são apresentados os parâmetros utilizados para as heurísticas propostas. Tais parâmetros foram ajustados empiricamente após alguns testes computacionais, permitindo a análise do comportamento dos algoritmos.

Parâmetros do GRASP:

- $num_iter: \lfloor nv/2 \rfloor$

Parâmetros do GA:

- Tamanho da população ($maxpop$): 100
- Número de gerações ($maxgen$): 10000
- Probabilidade de mutação (p_{mut}): 100%

Tendo em vista que os algoritmos propostos consistem em processos de amostragem aleatória, optou-se por rodá-los 10 vezes, para avaliar o comportamento médio dos mesmos, em termos de qualidade de solução obtida e de tempo de processamento.

Deve-se observar que a heurística GRASP proposta para a programação de veículos obteve o número ótimo de viaturas em todas as replicações do algoritmo. Deste modo, a comparação das soluções obtidas pelo método exacto e pelas técnicas heurísticas propostas foi realizada apenas para a programação de tripulações, pois, para essa parte do problema, houve discrepância entre os resultados obtidos pelo GA e pelo *branch-and-bound*.

No que concerne ao GA desenvolvido para a programação de tripulantes, foi obtida a solução ótima em 19 das 20 instâncias analisadas (a solução ótima do problema 70 não foi encontrada pelo GA).

Na Tabela 17 é apresentada uma comparação entre as soluções obtidas pelo GA e as soluções ótimas obtidas pelo *branch-and-bound*. Para as 200 execuções do GA, este obteve a solução ótima em 163 rodadas, o que representa um aproveitamento de 81,5%. Em 14 das 20 instâncias analisadas, o GA conseguiu obter a solução ótima em todas as 10 execuções do algoritmo.

Com base nos resultados apresentados na Tabela 17, pode-se salientar que o Algoritmo Genético obteve soluções de qualidade comparável às do método exacto.

Tabela 17: Comparação entre o método exacto e o Algoritmo Genético proposto.

Problema	percentual de cobertura (<i>c</i>)				<i>gap</i> (%)			Soluções óptimas em 10 execuções do GA
	zIP	melhor	Pior	média	melhor	pior	média	
406	94,1	94,1	91,8	92,0	0,0	2,4	2,1	1
504	84,4	84,4	84,4	84,4	0,0	0,0	0,0	10
905	85,7	85,7	85,7	85,7	0,0	0,0	0,0	10
13	85,3	85,3	85,3	85,3	0,0	0,0	0,0	10
833	87,5	87,5	87,5	87,5	0,0	0,0	0,0	10
15	86,5	86,5	86,5	86,5	0,0	0,0	0,0	10
501	84,4	84,4	84,4	84,4	0,0	0,0	0,0	10
407	83,6	83,6	83,6	83,6	0,0	0,0	0,0	10
466	85,1	85,1	75,7	84,2	0,0	9,5	0,9	10
609	87,0	87,0	81,2	82,9	0,0	5,8	4,1	3
907	90,3	90,3	83,3	88,8	0,0	6,9	1,5	3
810	84,4	84,4	84,4	84,4	0,0	0,0	0,0	10
605/606	87,2	87,2	86,0	86,9	0,0	1,2	0,3	8
81	76,3	76,3	76,3	76,3	0,0	0,0	0,0	10
70	91,6	89,2	88,0	88,3	2,4	3,6	3,3	0
102	91,3	91,3	83,7	90,5	0,0	7,7	0,9	8
201	86,3	86,3	86,3	86,3	0,0	0,0	0,0	10
411	82,1	82,1	82,1	82,1	0,0	0,0	0,0	10
316	87,3	87,3	87,3	87,3	0,0	0,0	0,0	10
913	87,5	87,5	87,5	87,5	0,0	0,0	0,0	10

A seguir, serão analisados os aspectos que, segundo o autor, propiciaram um melhor desempenho do GA, se comparada ao GRASP reportado no Capítulo 3.

O GA, por trabalhar com uma população inicial aleatória, consegue gerar soluções diversificadas, as quais estão espalhadas por diversas regiões do espaço de busca. Entretanto, essas soluções iniciais são de baixa qualidade. O processo de recombinação dirige a busca para regiões mais promissoras do espaço investigado. A recombinação de soluções leva à geração de uma prole inviável, o que é devidamente tratado pela heurística de reparos e pelo algoritmo de inserção de novas colunas.

O GA, apesar de iniciar com soluções de baixa qualidade, consegue migrar para soluções de elevada qualidade, atingindo, em geral, as soluções óptimas das instâncias

analisadas. Assim, constata-se que o GA desenvolvido explora, de forma eficaz, os processos de diversificação e intensificação.

A seguir, com base nas Figuras 11 e 12, são apresentadas algumas informações sobre o comportamento do GA proposto. Na Figura 11, pode-se observar que o GA desenvolvido tem um comportamento do tipo do que se observa nas metaheurísticas clássicas. Ao longo das iterações (gerações), o algoritmo investiga diversas soluções, conseguindo melhorias sucessivas. No gráfico apresentado, estas melhorias incorreram na obtenção da solução ótima (no caso da instância 833, $z = 10$).

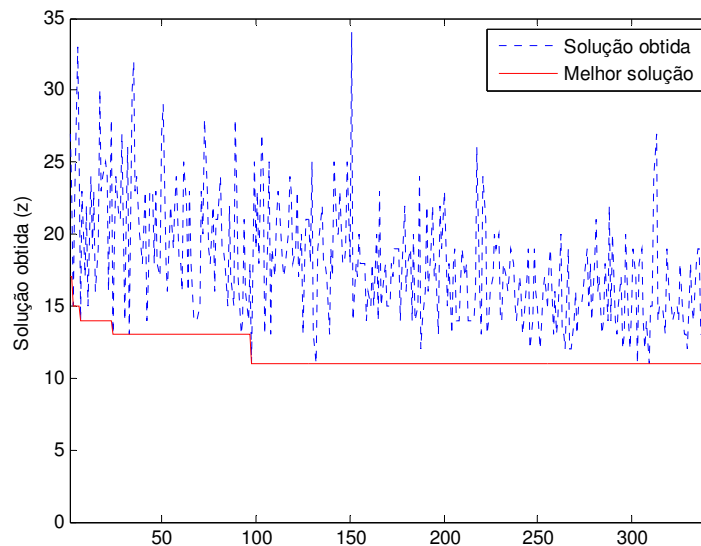


Figura 11: Soluções obtidas em uma execução do GA – instância 833.

O histograma da Figura 12 corrobora a hipótese de que o GA dirigiu a busca para soluções de elevada qualidade. Tendo em vista que a solução ótima da instância 833 é igual a 10 e que, no histograma apresentado, grande parte das frequências observadas dos valores de z situa-se entre 10 e 20 *leftovers*, o GA pesquisou diversas soluções próximas do ótimo global, demonstrando a eficácia do algoritmo.

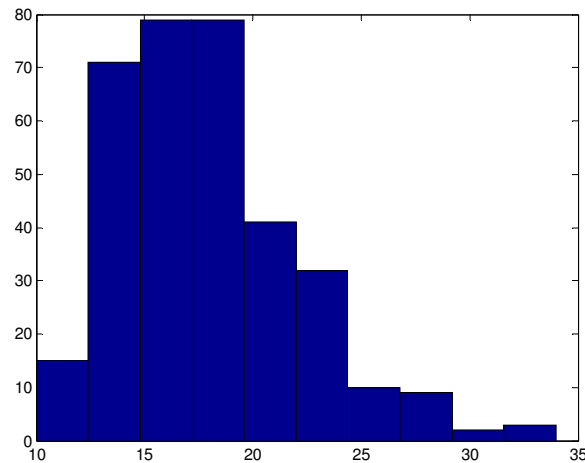


Figura 12: Histograma das soluções obtidas em uma execução do GA – instância 833.

A seguir, na Tabela 18, é feita uma análise comparativa entre o desempenho computacional (expresso em termos do tempo de processamento) do *branch-and-bound* e das heurísticas propostas.

Com relação aos tempos de processamento, com base nos dados da Tabela 18, pode-se constatar que as heurísticas apresentaram desempenho inferior ao *branch-and-bound*, em 7 das 20 instâncias analisadas.

No que concerne à composição do tempo de processamento das heurísticas propostas, pode-se observar que a maior parte (quase a totalidade) diz respeito ao GA. Os tempos de processamento referentes à heurística GRASP para a programação de veículos tendem à zero.

Esta discrepância entre os tempos de processamento das heurísticas e do método exacto, para o *Set Partitioning Problem*, já foi observada em outros trabalhos, como, por exemplo, Beasley e Chu (1995, 1998). No Capítulo 3, esta relação dos tempos de processamento entre métodos exactos e métodos heurísticos, para a variante do MCP estudada, já foi discutida.

Analisando as instâncias cujo GA obteve performance inferior ao *branch-and-bound*, pode-se observar que, exceto na instância 102, todas as demais instâncias possuíam tempos de ciclo elevados. Deste modo, evidencia-se que, no VCSP, a tabela de horários tem grande influência na complexidade da resolução de um problema.

Tabela 18: Comparação de performance entre o *branch-and-bound* e as abordagens propostas.

Instância	$t_{ZIP}(s)$	$t_{GA}(s)$			gap (%)		
		melhor	pior	média	melhor	pior	média
406	2	61	378	322	2950,0	18800,0	16000,0
504	5	1	2	1	-80,0	-60,0	-80,0
905	4	1	1	1	-75,0	-75,0	-75,0
13	45	1	3	1	-97,8	-93,3	-97,8
833	5	5	10	7	0,0	100,0	40,0
15	137	16	47	30	-88,3	-65,7	-78,1
501	9	1	1	1	-88,9	-88,9	-88,9
407	3	2	25	12	-33,3	733,3	300,0
466	4	3	116	44	-25,0	2800,0	1000,0
609	1	14	159	123	1300,0	15800,0	12200,0
907	4	6	153	101	50,0	3725,0	2425,0
810	6	1	1	1	-83,3	-83,3	-83,3
605/606	7	20	375	139	185,7	5257,1	1885,7
81	2	1	11	3	-50,0	450,0	50,0
70	1	353	372	363	35200,0	37100,0	36200,0
102	2	12	297	98	500,0	14750,0	4800,0
201	3	1	5	3	-66,7	66,7	0,0
411	7	3	166	23	-57,1	2271,4	228,6
316	1	3	77	24	200,0	7600,0	2300,0
913	21	1	96	27	-95,2	357,1	28,6

Se os tempos de ciclos são curtos, uma maior quantidade de tramos pode ser coberta por um serviço; caso contrário, se os tempos de ciclo são longos, uma menor quantidade de tramos pode ser coberta por um serviço. Deve-se observar ainda que a cada serviço deve ser alocado pelos menos uma folga para refeição. Deste modo, viagens com tempos de ciclo longos apresentam uma maior dificuldade na determinação de serviços que contemplem as folgas e cubram a maior quantidade de tramos possível.

4.5 Considerações finais

Neste capítulo foi apresentada uma nova formulação para o *Vehicle and Crew Scheduling Problem* com *changeovers*. O modelo proposto consiste em uma abordagem inovadora que busca utilizar o conceito de máxima cobertura com múltiplos recursos.

O VCSP foi decomposto em dois subproblemas, por forma a permitir sua resolução, seja de forma exacta, seja de forma heurística. Tal abordagem se diferencia da abordagem seqüencial tradicional devido ao facto dos serviços serem gerados

directamente sobre as viagens e não com base em blocos de viagens executados pelos veículos. Deste modo, tem-se um problema de cobertura com múltiplos recursos: viaturas cobrem viagens e serviços cobrem viagens.

Um conjunto de dados de teste baseados em 20 linhas de autocarros operantes em Fortaleza, foram resolvidos à optimalidade por meio do *solver* LINGO.

Foram também apresentadas duas heurísticas para o VCSP, que conseguiram resolver, de forma eficaz e eficiente, as instâncias analisadas. Uma variante do algoritmo GRASP foi proposta para a programação das viaturas e um GA híbrido foi desenvolvido para a programação de tripulantes.

Os algoritmos GRASP e GA foram executados de forma seqüencial para as instâncias avaliadas.

O GRASP obteve a solução óptima do problema de programação de tripulantes em todas as execuções do algoritmo. Os tempos de processamento consumidos por GRASP tendem a zero.

O GA híbrido proposto obteve a solução óptima em 19 das 20 instâncias analisadas. No que se refere aos tempos de processamento, em termos médios, o GA apresentou desempenho inferior ao *branch-and-bound*.

No próximo capítulo, será apresentada a extensão do GA desenvolvido, ao problema de máxima cobertura multiobjectivo.

Capítulo 5

Modelação multiobjectivo para a programação de veículos e tripulações

“Measure what is measurable, and make measurable what is not so.”

Galileo Galilei

Resumo: Este capítulo tem como objectivo apresentar extensões multiobjectivo para o problema de programação de veículos e motoristas. Na primeira secção, são apresentados conceitos básicos sobre optimização multiobjectivo. Na segunda secção é apresentada a natureza multiobjectivo do VCSP, com ênfase para suas restrições e objectivos. Na terceira secção, são apresentados os objectivos mais relevantes para o VSP, CSP e o VCSP. Na quarta secção é proposta uma abordagem inovadora para o VCSP multiobjectivo. Na quinta secção, são apresentados os resultados computacionais que validam a heurística desenvolvida. Por fim, na sexta secção são apresentadas as considerações finais acerca do capítulo.

5.1. Fundamentos de optimização multiobjectivo

Segundo Czyzak e Jaskiewicz (1998), em determinados contextos, um problema de decisão pode ser visto sob o prisma de actores diferentes, facto que envolve a necessidade de serem considerados múltiplos objectivos, de forma que a decisão a ser tomada propicie uma intervenção que satisfaça os critérios considerados pelos actores envolvidos no processo de decisão. Nestes casos, abordagens monocritério podem propiciar resultados pouco condizentes com o problema de decisão real.

Conforme Cohon (1978), a programação multiobjectivo representa uma generalização das abordagens mais tradicionais mono-objectivo aos problemas de

planeamento. O autor citado acrescenta que, ao serem considerados diversos critérios no processo de planeamento, três grandes melhorias são obtidas na resolução de problemas:

- A programação multiobjectivo promove regras mais apropriadas para os participantes no processo de planeamento e tomada de decisão;
- Uma gama maior de alternativas de intervenção usualmente é identificada quando uma metodologia multiobjectivo é empregada;
- Modelos (quando usados) ou a percepção do analista tornam-se mais realistas quando múltiplos objectivos são considerados no processo de planeamento e tomada de decisão.

Um problema com múltiplos objectivos pode ser convertido em um problema mono-objectivo por meio de uma função de agregação. Tal função consiste num vector de pesos associado a cada objectivo e a conversão pode ser efectuada pelo produto escalar do vector de pesos pelo vector de variáveis de decisão.

No entanto, o processo de obtenção de pesos, além de muitas vezes ser extremamente laborioso, em alguns casos ainda pode não vir a refletir adequadamente os interesses dos actores envolvidos no processo decisório.

A optimização multiobjectivo procura obter soluções designadas por soluções não-dominadas. Graficamente, tais soluções assumem a forma de uma curva denominada fronteira de Pareto (DEB, 2001).

Considere-se um problema de maximização, onde \mathbf{x} é um vector de variáveis de decisão, $f_j(\mathbf{x})$ um conjunto de funções objectivo e D o conjunto de soluções viáveis. Uma solução $\mathbf{x} \in D$ é não-dominada (óptima de Pareto) se não existe nenhum $\mathbf{x}' \in D$ tal que $\forall j, f_j(\mathbf{x}') \geq f_j(\mathbf{x})$ e $f_j(\mathbf{x}') > f_j(\mathbf{x})$ para pelo menos um j .

Em seguida, será demonstrado um exemplo simples de um problema de optimização combinatoria multiobjectivo, no caso o problema da mochila multiobjectivo (*Multiobjective Knapsack Problem* – MOKP).

Considere-se um conjunto discreto com n itens, de modo que a cada um deles associa-se um lucro c_j e um peso w_j . O problema da mochila consiste em determinar quais itens a serem alocados a uma mochila de capacidade W , de modo a maximizar o lucro. O problema da mochila clássico pode ser formulado como segue:

[KP]

$$\text{Maximizar } z = \sum_{j=1}^n c_j x_j \quad (5.1)$$

Sujeito a:

$$\sum_{j=1}^n w_j x_j \leq W \quad (5.2)$$

$$x_j \in \{0,1\} \quad (5.3)$$

A equação 5.1 representa a função objectivo do problema, em que um item de valor c_j fará parte da solução se x_j igual a 1, não fazendo parte da mesma se x_j igual a 0. A restrição 5.2 impõe que o produto escalar do vector de pesos pelo vector de variáveis de decisão seja menor ou igual à capacidade W da mochila. Para que uma solução seja viável, a soma dos pesos dos itens que compõem a solução deve ser menor ou igual a W . Por fim, as condições 5.3 denotam o carácter binário das variáveis de decisão.

Neste exemplo, apenas a fim de ilustrar o conceito de soluções não-dominadas, o problema será considerado irrestrito e serão utilizadas duas funções objectivo a serem maximizadas:

[MOKP]

$$\text{Maximizar } z_1 = \sum_{j=1}^n c_j x_j \quad (5.4)$$

$$\text{Maximizar } z_2(X) = W - \sum_{j=1}^n w_j x_j \quad (5.5)$$

Sujeito a:

$$x_j \in \{0,1\} \quad (5.6)$$

A função objetivo z_2 está relacionada à restrição de capacidade do problema da mochila clássica. Com o objectivo de exemplificar o MOKP, será considerado um problema da mochila com capacidade $W = 100$ e 5 itens a serem empacotados, com as seguintes características, ilustradas na Tabela 19.

Tabela 19: Dados do exemplo do MOKP.

Item	1	2	3	4	5
w_j	25	15	14	25	15
c_j	80	120	80	15	30

Tendo em vista que se trata de um problema de optimização combinatória binário, existem 2^n soluções possíveis, em que n é a quantidade de variáveis de decisão do problema. Deste modo, no exemplo em questão, existem 32 soluções possíveis, as quais são ilustradas na Figura 13.

Pode-se constatar que as soluções 1, 3, 4, 11, 17, 25, 31 e 32 (vide Tabela 20) não são dominadas por nenhuma outra solução, tendo em conta os dois objectivos em análise. Deste modo, tais soluções formam a frente de Pareto (conjunto de soluções não-dominadas). As demais soluções para o MOKP em análise podem ser superadas por pelo menos uma das soluções óptimas de Pareto listadas acima.

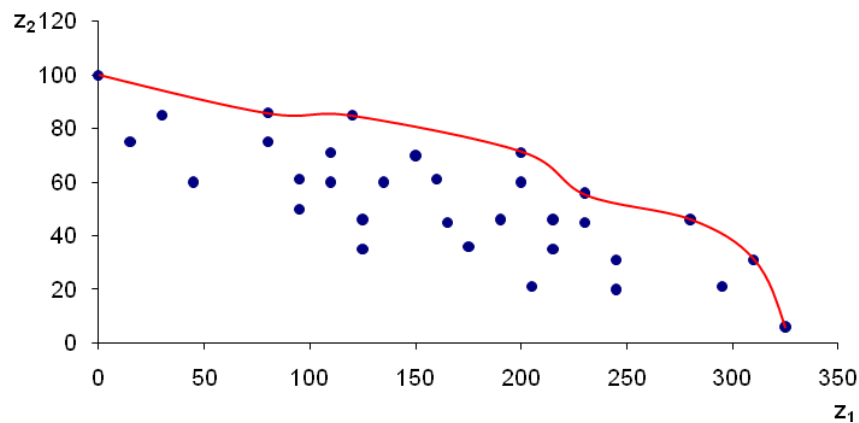


Figura 13: Soluções possíveis para o exemplo do MOKP.

Tabela 20: Soluções possíveis para o exemplo do MOKP.

Solução	$z_1(X)$	$z_2(X)$	Solução	$z_1(X)$	$z_2(X)$	Solução	$z_1(X)$	$z_2(X)$	Solução	$z_1(X)$	$z_2(X)$
1	0	100	9	95	50	17	280	46	25	230	56
2	80	75	10	110	60	18	215	35	26	125	46
3	120	85	11	200	71	19	230	45	27	295	21
4	80	86	12	135	60	20	175	36	28	245	31
5	15	75	13	150	70	21	190	46	29	205	21
6	30	85	14	95	61	22	125	35	30	245	20
7	200	60	15	110	71	23	215	46	31	310	31
8	160	61	16	45	60	24	165	45	32	325	6

De acordo com Ulungu *et al.* (1999), além da característica NP-difícil de muitos problemas de optimização combinatória, os problemas de optimização combinatória multiobjectivo são particularmente intratáveis por um método exacto.

Nos últimos anos, a resolução de problemas de optimização combinatória multiobjectivo tem-se caracterizado como uma área de investigação bastante activa, sendo as metaheurísticas um dos métodos mais utilizados para a resolução de tais problemas (VIANA, 2004; DIAS, 2005).

No caso dos problemas de optimização combinatória com um único objectivo, as metaheurísticas destinam-se a gerar movimentos que visem diversificar as soluções pesquisadas no espaço de busca e, após diversas iterações, intensificar a busca em uma região promissora e chegar a uma solução tão próxima do óptimo global quanto possível, em tempo computacional aceitável.

No que concerne aos problemas de optimização combinatória multiobjectivo, as metaheurísticas devem combinar mecanismos de diversificação e de intensificação para construir um conjunto diversificado de soluções não-dominadas tão próximas quanto possível da fronteira de Pareto, em tempo computacional admissível.

5.2 Natureza multiobjectivo do VCSP

Conforme apresentado no Capítulo 2 desta tese, o *Vehicle and Crew Scheduling Problem* (VCSP) consiste num problema integrado de optimização combinatória no qual a programação de veículos e de tripulações, geralmente relacionados com um sistema de transporte público, são solucionados de forma conjunta.

Uma das grandes dificuldades ao programar veículos e tripulações é que estas duas programações devem satisfazer critérios de viabilidade e de compatibilidade. Conforme Steizen *et al.* (2007), o VCSP deve contemplar estes dois critérios.

De acordo com Freling *et al.* (2003), ao solucionar o problema de programação de veículos e tripulações, pode-se proceder de três maneiras: realizar as duas programações separadamente, sem considerar as interações entre os dois problemas; programar os veículos, e, em seguida, as tripulações; e solucionar os dois problemas de uma forma integrada.

Programar veículos e tripulações de forma independente facilita o processo de solução, por tornar a modelagem mais simples; entretanto, as soluções geradas certamente serão de pior qualidade, em termos de diversos requisitos práticos que devem ser satisfeitos. Esta abordagem independente pode acarretar na não satisfação dos critérios de viabilidade e de compatibilidade entre as duas programações, facto que torna esta abordagem pouco atractiva sob o ponto de vista prático.

Devido ao facto destes dois recursos (viaturas e tripulações) serem utilizados de uma forma distinta, programar o uso de um destes recursos, sem contemplar o outro, pode incorrer na inviabilidade da programação do outro recurso. Em certas circunstâncias, mesmo que as duas programações sejam viáveis, elas podem ser incompatíveis, visto que não possam ser executadas concomitantemente.

Quando se efectua primeiramente a programação dos veículos para, em seguida, escalonar o pessoal tripulante, usualmente usa-se o termo *abordagem tradicional*. De certo modo, pode-se dizer que esta abordagem hierarquiza os dois problemas, dando ênfase à programação das viaturas.

Na abordagem tradicional, a programação de veículos é efectuada num primeiro estágio, sendo sucedida pela programação dos tripulantes. A geração de serviços de pessoal tripulante é feita com base nos blocos de viagens executadas por viaturas, definidos na programação de veículos.

Diante do facto que as tipulações usualmente são responsáveis por grande parte do custo operacional do sistema (BODIN *et al.*, 1983), programar os serviços de tripulantes com base em uma programação de viaturas levam a soluções que não utilizam o recurso tripulação da melhor forma possível.

Tal abordagem não é a mais pertinente, visto que a programação dos tripulantes é que constitui o gargalo do VCSP. A satisfação das restrições do CSP é, em geral, bem mais difícil do que a satisfação das restrições do VSP, em particular quando a sobreposição de serviços (*overcovers*) não é permitida. Além desta complexidade, usualmente os tripulantes contribuem de forma mais incisiva (BODIN *et al.*, 1983) do que os veículos para o custo operacional do sistema de transporte.

Deve-se ressaltar que Patrikalakis e Xerokostas (1992) propõem uma inversão da abordagem seqüencial na qual, em uma primeira etapa, efectua-se a programação dos tripulantes, para, em seguida, ser realizada a programação dos veículos.

Ao proceder-se de forma sequencial, isto é, resolver uma das programações, para, em seguida, resolver a segunda, com base nas informações obtidas na primeira, tende-se a se obter resultados melhores do que ao se usar a abordagem independente.

A abordagem integrada consiste em efectuar as programações de veículos e de tripulações concomitantemente. Procedendo-se desta forma, garante-se que as duas programações sejam viáveis e mutuamente compatíveis, além de incorrer-se numa maior eficiência na utilização dos recursos. Tal vantagem é ainda mais significativa

quando os dois recursos em análise são inseparáveis, ou seja, quando os *changeovers* não são permitidos.

Assim, percebe-se a importância do VCSP como uma abordagem para garantir a viabilidade e a compatibilidade entre a programação de veículos e de tripulações. Na opinião do autor, este é a principal justificação para a abordagem do problema de uma forma integrada.

Ao serem considerados simultaneamente as duas programações supracitadas obtêm-se as condições favoráveis para garantir a satisfação dos critérios de viabilidade e compatibilidade. Acrescido a este facto, existe a possibilidade de reduzir os custos operacionais, por ter uma maior possibilidade de afectar os recursos da forma mais adequada. Estas vantagens têm um preço: a abordagem integrada torna o processo de modelagem e resolução do problema muito mais complexo.

Na opinião do autor, a programação de veículos e tripulações não pode ser vista, de uma forma simplista, como *integrada* ou *não integrada*. Esta integração pode ser efectuada em níveis, isto é, pode-se ter uma integração mais forte, uma integração mais ténue, bem como uma não-integração.

Neste contexto, no enfoque proposto na presente tese, far-se-á a seguinte separação: a programação será dita *integração total* ou *integração forte* quando os veículos e os tripulantes não puderem ser separados, ou seja, quando os *changeovers* são proibidos. No caso em que os tripulantes puderem fazer trocas de veículos durante sua jornada de trabalho, isto é, quando os *changeovers* forem permitidos, a programação será denotada *integração parcial* ou *integração leve*.

No Capítulo 3 desta tese apresentou-se uma nova formulação matemática para o VCSP, a qual contempla uma integração total. Já no quarto Capítulo, foi proposta uma nova formulação para o VCSP que considera uma integração parcial.

Diante do facto da programação de tripulantes ser mais complexa do que a programação de veículos, deve-se observar as questões que mais contribuem para esta dificuldade de modelagem e resolução. A permissão ou a proibição de *overcovers* (sobreposição de serviços) e de *leftovers* (ocorrência de tramos descobertos) são dois factores de grande impacto na complexidade do VCSP.

Deve-se ressaltar que a permissão ou proibição de *leftovers* possui um impacto maior na complexidade de modelagem, conforme será elucidado posteriormente.

Ao se permitir a ocorrência de *overcovers*, facilita-se o processo de resolução do CSP, e, por conseguinte, do VCSP. Contudo, ao permitir a ocorrência de *overcovers*,

paga-se o tempo da facilidade de resolução com uma perda de qualidade das soluções geradas.

Ao permitir que haja a sobreposição de serviços, admite-se que duas tripulações estejam em um mesmo veículo durante a cobertura de um tramo. Salvo sob circunstâncias específicas, as empresas de transporte público tendem a não admitir a ocorrência de tais sobreposições, pois tais soluções tendem a ser onerosas.

Ao se pagar duas tripulações para efectuar a cobertura de um mesmo tramo, sendo que apenas uma tripulação efetivamente realiza a tarefa, consiste em uma ineficiência. Portanto, ou se impede a existência de *overcovers*, ou estes devem ser restritos por um limite superior.

Caso o processo de modelagem permita a ocorrência dos *overcovers*, estes devem ser posteriormente retirados das soluções. Este processo é denominado *correção* das soluções. Este processo de correção, além de poder ser complexo, incorre geralmente numa perda de qualidade das soluções.

Dias (2005) salienta o aspecto de que soluções que possuam muitos *overcovers* tendem a ter tramos cobertos por muitos serviços, sendo que cada serviço sozinho cobre poucas viagens. Ao corrigir estas soluções obtêm-se *serviços curtos*, geralmente não admissíveis, ou seja, serviços que não cumprem as regras definidas. Tais serviços podem ser vistos como *leftovers* e devem, portanto, ser tratados como *leftovers*. Percebe-se então que a permissão de *overcovers* não tem consistência prática.

Conforme Dias *et al.* (2001), ao se adoptar uma abordagem baseada em particionamento de conjuntos para a programação de tripulações, ou seja, quando os *overcovers* não são permitidos, não se tem a garantia da obtenção de uma solução viável. Ao se permitir a ocorrência de *leftovers*, isto é, tramos que não são cobertos por serviços regulares, pode-se garantir a viabilidade das soluções geradas. Posteriormente estes *leftovers* devem ser cobertos, seja por serviços especiais gerados para o efeito, seja por serviços de horas extraordinárias ou combinando *leftovers* de diferentes horários.

Os *leftovers* possuem papel importante na modelagem e resolução do VCSP. Dos modelos matemáticos para o VCSP, usualmente tem-se um conjunto de restrições no qual se relacionam variáveis correspondentes aos serviços e aos veículos. Este tipo de restrição torna o modelo matemático mais extenso e de resolução mais difícil. Todavia, ao se permitir a ocorrência de *leftovers*, não se tem a obrigatoriedade de considerar restrições relacionando serviços e viaturas.

Tal situação deve-se ao seguinte facto: ao se permitir que tramos fiquem descobertos em uma solução, não se tem a imposição de que cada tramo seja coberto por um par veículo-tripulação. Deste modo, pode-se observar que a permissão de *leftovers* diminui consideravelmente o número de restrições do modelo, facilitando a resolução do mesmo.

5.3 Discussão dos possíveis objectivos

Nesta secção, são apresentados e explicados os principais objectivos para o VSP e para o CSP. Em seguida, discorre-se sobre como estes objectivos podem influenciar o problema integrado.

Alguns dos objectivos reportados são inerentes às características do problema como, por exemplo, a minimização do número de viaturas na solução. Estes tipos de objectivos podem ser enquadrados como *estruturais*. Por outro lado, alguns objectivos estão relacionados com os métodos de resolução do problema como, por exemplo, minimizar a inviabilidade de uma solução, com base na formulação de particionamento de conjuntos. Estes objectivos podem ser enquadrados como *instrumentais*.

5.3.1 Objectivos para o VSP

Baita *et al.* (2000) enumeram cinco objectivos a serem considerados na programação de veículos:

Minimizar a quantidade de veículos (objectivo estrutural)

Quando a minimização do número de veículos é o objectivo principal, também se pode chamar o problema de programação de veículos de *Minimum Fleet Size Problem* – MFSP (BERTOSSO *et al.*, 1987).

A minimização da quantidade de viaturas está intimamente relacionada com os custos fixos (custos de capital). Se a frota em operação é menor, obviamente os custos operacionais e de manutenção também serão menores.

Minimizar o número de mudanças de linha (objectivo estrutural)

Ainda que não haja geralmente nenhuma restrição que limite que um dado veículo deva operar na mesma linha durante toda a jornada de trabalho, tal limitação é um requisito importante para a qualidade das soluções geradas.

As trocas de linhas por veículo são bastante inconvenientes para os tripulantes, além de poderem também ter um impacto negativo num outro objectivo da programação de veículos, o da minimização de viagens não produtivas.

Também se pode ressaltar que uma programação de veículos que não permita a troca de veículos entre linhas facilita o processo de programação de serviços de tripulações, visto que o número de viagens não produtivas é reduzido drasticamente.

Minimizar a quantidade de viagens em vazio (objectivo estrutural)

As viagens em vazio, além de não serem produtivas para as companhias, geram uma série de malefícios para diversos actores, como, por exemplo, aumento dos custos operacionais, aumento do tempo de espera por parte dos utentes, aumento do congestionamento no tráfego urbano de veículos, aumento do impacto no meio ambiente (emissão de poluentes, ruído, vibração, intrusão visual, etc.).

Deve-se destacar que, usualmente, o apoio financeiro do Estado às companhias privadas de transporte público é proporcional à duração das viagens produtivas, sendo as viagens improdutivas não subsidiadas.

Em algumas circunstâncias, as companhias minimizam as viagens em vazio por meio do estacionamento dos veículos em pontos distintos da cidade, quando eles não estão em operação efetiva. Deste modo, eliminam-se duas viagens não produtivas (*pull-in* e *pul-out*).

Minimizar a duração das viagens em vazio (objectivo estrutural)

A descrição deste objectivo é análoga ao do objectivo anterior,

Minimizar os tempos ociosos nos terminos (objectivo estrutural)

Um objectivo importante da programação de veículos é a redução do tempo em que um veículo fica ocioso ao término de uma viagem, à espera de realizar viagens produtivas. Souza (2010) acrescenta que também deve ser minimizado o tempo ocioso entre duas viagens consecutivas, descontando-se os tempos de embarque e desembarque de passageiros, os quais são obrigatórios.

O tempo em que o veículo fica parado acaba por incorrer em ineficiência operacional, pois os tripulantes que estão conduzindo os veículos serão pagos por estes tempos ociosos, salvo condições específicas.

Deste modo, é pertinente que o veículo execute as viagens de uma forma mais contínua possível, ou seja, sem interrupções, salvo nos casos em que são obrigatórios os intervalos para refeição dos tripulantes.

No caso em que os *changeovers* são permitidos, durante o intervalo para refeição de uma tripulação que conduzia um determinado veículo, este pode ser guiado por outra tripulação. Todavia, no caso em que os *changeovers* são proibidos, o intervalo para refeição de uma tripulação incorre em ociosidade da viatura.

5.3.2 Objectivos para o CSP

A seguir, serão apresentados os principais objectivos reportados para o CSP, com base nas referências consultadas.

Minimizar o custo dos serviços seleccionados (objectivo estrutural)

Conforme apresentado no Capítulo 2 desta tese, este objectivo é usualmente adoptado na modelação do CSP.

Com base no problema de cobertura ou de particionamento de conjuntos, dada uma matriz de serviços, na qual cada coluna possui um custo associado, pretende-se seleccionar as colunas (serviços) que incorram no menor custo global.

Conforme discutido no terceiro capítulo desta tese, usualmente os tripulantes são pagos por jornada de trabalho e não por hora efectivamente trabalhada. Assim, o custo de um serviço não é directamente proporcional à quantidade de horas trabalhadas pelo tripulante, isto é, a relação entre custo e duração de um serviço é de carácter não-linear. Quando ocorrem serviços com horas extraordinárias, ao custo de uma jornada trabalhada é acrescido um factor de custo, o qual é proporcionalmente superior ao custo de uma jornada padrão.

Deve-se observar que no caso em que os custos dos serviços são admitidos como unitários, o objectivo em foco reduz-se a minimizar o número de serviços na solução.

Minimizar o número de tramos descobertos – *letfovers* (objectivo instrumental)

Conforme Lourenço *et al.* (2000), tendo em vista que, em alguns casos práticos, se permitem tramos descobertos nas soluções, obter uma solução que maximize a cobertura dos tramos, ou seja, que minimize a não cobertura dos tramos, é um critério adoptável na realidade de empresas de transporte público.

Hoffman e Padberg (1993) afirmam que não existe garantia de obtenção de soluções ótimas, ou mesmo de soluções viáveis, quando o CSP é modelado como um problema de particionamento de conjuntos. Conforme discutido no Capítulo 2 da presente tese, pode-se adoptar um modelo relaxado (*Relaxed Set Partitioning Problem*) o qual permite a não-cobertura de alguns tramos. Na modelação proposta, tais tramos seriam cobertos *a posteriori*, por serviços extraordinários ou por novos serviços gerados para tanto.

Minimizar a inviabilidade de uma solução, com base na formulação de cobertura de conjuntos (objectivo instrumental)

De acordo com Lourenço *et al.* (2000) adoptam uma formulação de cobertura de conjuntos, *overcovers* (coberturas de tramos por mais de um serviço) são permitidos. Esta função objectivo destina-se a minimizar as sobreposições de serviços, aproximando a solução a um particionamento de conjuntos.

Dias (2005) salienta que a formulação por cobertura de conjuntos pode vir a gerar serviços inexecutáveis na prática, sendo necessários processos de correção das soluções. Estas soluções, mesmo corrigidas, podem vir a apresentar baixa qualidade, expressa em termos de muitos *leftovers*.

Ao se permitir a ocorrência de *overcovers*, permite-se, por conseguinte, que mais de uma tripulação esteja concomitantemente a cobrir um tramo. Procedendo-se dessa forma, tende-se a acarretar em uma infeciência na utilização do recurso tripulação.

Minimizar o número de serviços na solução (objectivo estrutural)

Conforme elucidado anteriormente, este objectivo é um caso particular do objectivo minimizar o custo dos serviços seleccionados.

Este critério é proposto por Lourenço *et al.* (2000) e por Dias (2005). Lourenço *et al.* (2000) destacam que os planeadores de algumas companhias de transportes público preferem soluções com um menor número de serviços, pois estas facilitam o processo de planeamento e implementação das soluções.

Pode-se destacar que, nas abordagens apresentadas nos Capítulos 3 e 4 da presente tese, a quantidade de serviços na solução é modelada por um limite superior d , o qual consiste em uma restrição para o problema. Esta restrição implicitamente representa um importante objectivo a ser considerado nas soluções.

Também se pode observar que, no caso em que existam múltiplos tipos de serviços a serem considerados, pode-se buscar a minimização de um número de serviços de um tipo específico.

Por exemplo, no caso em que sejam permitidos serviços compostos por uma ou duas etapas, soluções podem ser compostas por vários serviços de uma única etapa. Tal solução é viável, porém pode não ser desejável pelos planeadores ou pelos tripulantes.

Minimizar o número de trocas de motoristas por veículo – changeovers (objectivo estrutural)

Critério proposto por Lourenço *et al.* (2000) e por Dias (2005). Os *changeovers* podem quebrar a fluidez da operação de uma companhia, bem como gerar reclamações por parte dos funcionários.

Outra questão importante no que concerne aos *changeovers* diz respeito aos custos de manutenção. Quando diversos motoristas guiam um mesmo veículo, os custos de manutenção da frota aumentam, pois não há um controlo rígido acerca da responsabilidade sobre a conservação do veículo. Quando poucos motoristas guiam um determinado veículo, estes irão fazê-lo com um maior zelo, pois sabem que aquele veículo encontra-se sobre sua exclusiva responsabilidade. Deste modo, os custos de manutenção diminuem.

Esta é uma das principais motivações práticas pela qual os *changeovers* são proibidos por diversas companhias de transporte público.

Minimizar a duração dos leftovers (objectivo estrutural)

Dias (2005) destaca que não somente a quantidade de *leftovers* deve ser minimizada, mas também a duração destes. Dependendo da duração dos tramos, pode-se ter *leftovers* de curta ou de longa duração. Seria mais atractivo ter uma maior quantidade de *leftovers* de curta duração ou uma menor quantidade de *leftovers* de longa duração? Percebe-se que existe uma relação de *trade-off* entre os objectivos número de *leftovers* e duração dos *leftovers*.

No caso em todos os tramos possuem a mesma duração, a distinção entre minimização do número de *leftovers* e a minimização da duração dos *leftovers* pode ser vista como redundante.

Minimizar os tempos ociosos (objectivo estrutural)

Dias (2005) e Souza (2010) reportam este objectivo. Os tripulantes são remunerados por jornada de trabalho, sendo que, usualmente, apenas a paragem para refeição é não-remunerada.

Logo, se forem gerados serviços nos quais os tripulantes fiquem ociosos, estas paragens serão remuneradas e tais serviços serão pouco atractivos do ponto de vista económico.

Dias (2005) salienta que, apesar de indesejáveis, estas pequenas folgas (tempos não-productivos) nos serviços são fundamentais para a qualidade das soluções. Isto ocorre porque, ao se permitirem pequenas interrupções nos serviços, aumenta-se substancialmente o número de serviços viáveis, o que usualmente incorre em soluções com menos serviços e menos *leftovers*.

Minimizar os tempos de horas extras (objectivo estrutural)

Souza (2010) destaca este objectivo. Tendo em vista que horas de trabalho que excedem a jornada padrão geralmente são pagas com valor bem superior às horas da jornada padrão, ao serem reduzidos os serviços que contenham horas-extras, reduzem-se, também, os custos operacionais.

Deve-se ressaltar que existem empresas em que é usual gerar serviços com horas extras, enquanto em outras tal conduta não é permitida.

Minimizar o desvio entre a duração média desejada e a duração média de cada tipo de serviço (objectivo estrutural)

De acordo com Dias (2005), este é um dos mais importantes critérios analisados pelos planeadores de companhias de transportes colectivos. Em algumas companhias, outros critérios importantes, tais como a quantidade de serviços, são sacrificados de modo a atingir a duração desejada para cada tipo de serviço.

Existe uma relação de *trade-off* entre a duração média de um serviço e a quantidade de serviços em uma solução. Por exemplo, uma dada solução pode consistir num número de serviços de longa duração média, os quais irão gerar grande insatisfação entre os tripulantes. Assim, os planeadores podem preferir ter uma solução na qual haja serviços a mais de menor duração média, por questões de cunho operacional.

Mesmo que possam ocorrer serviços longos, em termos médios, a duração dos serviços não pode exceder a um limite superior. Isto ocorre para facilitar o processo de planeamento da rotação (*rostering*) dos tripulantes.

Minimizar o número de tripulantes necessário para efectuar todas as viagens (objectivo estrutural)

Critério proposto por Dias (2005). Deve-se salientar que este objectivo é distinto do objectivo que procura minimizar o número de serviços. Obter a quantidade mínima de motoristas nem sempre é uma tarefa trivial, devido à natureza dos *leftovers*.

Em alguns casos, um *leftover* pode ser designado para um motorista como trabalho extraordinário; entretanto, em outros casos, um *leftover* pode ser alocado para um motorista como um serviço incompleto (e inviável) ou ser combinado com outro *leftover* para compor um serviço viável.

Minimizar o número serviços alocados por veículo (objectivo estrutural)

Critério reportado por Dias (2005). Conforme comentado anteriormente, a partir do momento em que diversos tripulantes conduzem um mesmo veículo, os custos de manutenção da frota aumentam substancialmente. Deste modo, as companhias desejam que um veículo seja designado para uma quantidade mínima de motoristas. Em geral, este número não excede dois motoristas.

Maximizar a satisfação dos tripulantes (objectivo estrutural)

Este objectivo é proposto por Kohl e Karisch (2004). Em diversas companhias, nomeadamente no que concerne ao transporte aéreo, os tripulantes indicam os serviços de sua preferência.

Indicadores para medir este objectivo podem variar de acordo com a natureza do problema. Alguns exemplos de indicadores são: compatibilidade na realização do serviço, tempo de descanso entre jornadas de trabalho e padrões de dias de descanso.

Atender aos desejos dos tripulantes pode acarretar em diversos benefícios para a companhia, pois, estando satisfeitos, estes podem oferecer um serviço de maior qualidade aos utentes do sistema de transportes, bem como serem mais eficientes na realização de suas actividades.

5.3.3 Objectivos para o VCSP

Diante do facto que o VCSP consiste numa junção do VSP e do CSP, a extensão dos objectivos destes problemas, considerados isoladamente, para o problema integrado é natural.

Quando os *changeovers* não são permitidos, os objectivos devem considerar que os recursos modelados, ou seja, veículos e tripulantes, atuam como um único recurso. Quando os *changeovers* são permitidos, veículos e tripulações podem ser considerados como dois recursos distintos.

Além dos objectivos mencionados *a priori*, outras funções objectivo para o VCSP foram propostas, dentre as quais consta a função objectivo do modelo matemático inovador proposto no Capítulo 4 desta tese.

Minimizar o custo total de afectação de viaturas e tripulantes (objectivo estrutural)

Critério reportado por Freling (1999b). A função objectivo (vide equação (2.26) no Capítulo 2) é composta por duas parcelas: uma representa o custo de alocação das viaturas e a outra o custo de alocação das tripulações.

Na opinião do autor, agrupar os custos de viaturas com os de pessoal tripulante não é tarefa simples, pois estes dois recursos possuem naturezas distintas. No caso dos custos com veículos, em muitos casos o mais importante é minimizar os custos fixos (ou seja, a quantidade de viaturas em uma programação) e não os custos operacionais. No que se refere aos tripulantes, o custo refere-se ao pagamento da jornada de trabalho mais horas extraordinárias (caso necessário).

Maximizar o atendimento à procura (objectivo estrutural)

Critério reportado por Laurent e Hao (2007). Os autores citados reportam uma aplicação do VCSP numa empresa de transporte de luxo (*limousines*) em Paris. O objectivo primordial dessa modelação é obter uma programação que atenda, ao máximo possível, a procura dos clientes.

Caso não seja possível atender plenamente à procura, uma opção seria subcontratar o serviço a outras empresas para a realização destas viagens não cobertas.

Maximizar a cobertura de tramos (objectivo estrutural)

Crítério reportado nesta tese. De certo modo, este critério assemelha-se bastante ao objectivo maximizar o atendimento à procura, o qual foi reportado por Laurent e Hao (2007). A diferença reside em que o objectivo de maximizar a cobertura de tramos é visto como um problema de cobertura com múltiplos recursos. As viagens devem ser cobertas por dois recursos distintos: viaturas e tripulações. Caso haja viagens descobertas, estas devem ser tratadas como *leftovers*.

5.4 Uma abordagem multiobjectivo para o VCSP

Com base nos objectivos escolhidos para o *Multiple Objective Maximal Covering Problem* (MOMCP), são desenvolvidos procedimentos específicos para avaliar a aptidão de soluções para cada um dos critérios em análise.

O MOMCP estudado possui a seguinte estrutura:

[MOMCP₁]

$$\text{Minimizar} \quad z_1 = \sum_{i=1}^m z_i \quad (5.4)$$

$$\text{Minimizar} \quad z_2 = \sum_{j=1}^n c_j x_j \quad (5.5)$$

Sujeito a:

$$\sum_{j=1}^n a_{ij} x_j + z_i = 1 \quad \forall i = 1, \dots, m. \quad (5.6)$$

$$\sum_{j=1}^n x_j \leq d \quad (5.7)$$

$$x_j \in \{0,1\} \quad \forall j = 1, \dots, n. \quad (5.8)$$

$$z_i \geq 0 \quad \forall i = 1, \dots, m. \quad (5.9)$$

A função objectivo representada pela equação (5.4) procura minimizar o número de tramos descobertos. A função objectivo representada pela equação (5.5) procura minimizar os custos associados às colunas presentes na solução. Caso o *i*-ésimo tramo não possa ser coberto, o conjunto de restrições do tipo (5.6) arbitra que $z_i = 1$. Em

consequência desta restrição, tem-se que a integralidade das variáveis z_i pode ser relaxada. A restrição (5.7) impõe que um número máximo de d de colunas da matriz A seja seleccionado na solução. O conjunto de restrições do tipo (5.8) ilustra o carácter binário das variáveis de decisão. As restrições do tipo (5.9) impõem a não-negatividade das variáveis z_i .

Em relação às funções objectivo, são pertinentes os seguintes comentários. No caso de uma integração forte (*changeovers* proibidos), o objectivo de minimizar tramos descobertos diz respeito a tramos que não são cobertos por um par veículo-tripulante. A minimização dos custos associados às colunas presentes na solução diz respeito ao custo de afectação do par veículo-tripulante na cobertura de um conjunto de tramos. Assim, os dois objectivos relatados podem estar relacionados tanto com serviços como com viaturas.

Na optimização tradicional, a utilização de heurísticas procura obter um escalar que representa uma solução tão próxima quanto possível da solução óptima do problema. No âmbito da optimização multiobjectivo, deseja-se obter um vector de soluções, ditas não-dominadas, as quais se aproximam, o quanto seja possível, da frente de Pareto.

Logo, ao se utilizar heurísticas que trabalhem com populações de soluções, espera-se, em geral, que tais soluções, ao longo da simulação do processo evolucionário, consigam aproximar-se da fronteira de Pareto de uma forma mais adequada do que algoritmos que trabalham com uma única solução.

Outro motivo da adoção de um GA diz respeito ao seu desempenho eficaz e eficiente na resolução do VCSP, conforme elucidado no Capítulo 4 da presente tese. No Capítulo mencionado, verificou-se que o GA suplantou um algoritmo GRASP, o qual trabalha sobre uma única solução, na resolução do VCSP. Tal resultado justifica a escolha do GA como uma heurística adequada para receber as extensões multiobjectivo na presente investigação.

Outro aspecto favorável à adoção do GA foi o facto de as estruturas de dados e os operadores genéticos implementados para o problema mono-objectivo serem, em grande parte, aproveitados nas extensões multiobjectivo. Logo, adoptando-se o GA tem-se um ganho significactivo no tempo de implementação.

Deve-se ressaltar que, diante do foco da presente investigação, o método a ser utilizado não é uma questão crítica, mas sim como a abordagem multiobjectivo pode trazer ganhos na resolução do VCSP.

5.4.1 Algoritmos Genéticos multiobjectivo

Nesta secção, discorre-se sobre a definição do algoritmo a ser projectado e implementado para a resolução do VCSP envolvendo múltiplos critérios. Diante dos diversos algoritmos heurísticos para optimização multiobjectivo, optou-se por um Algoritmo Genético (GA). As justificativas da adoção de um GA são apresentadas a seguir.

A literatura reporta diversos GA's direccionados para a optimização multiobjectivo. Com base nos *surveys* apresentados por Coello (2001), Deb (2001) e Konak *et al.* (2006), foram analisados os algoritmos considerados mais importantes.

Weight Based Genetic Algorithm (WBGA)

Abordagem proposta por Hajela e Lin (1987). Um vector de pesos sorteados aleatoriamente, cujo somatório é igual à unidade, converte um problema multiobjectivo em um problema com um único objectivo.

Como vantagens do WBGA, podem ser destacadas as seguintes: aproveitar a estrutura e a implementação do GA tradicional (mono-objectivo); ser fácil de implementar e de performance eficiente; e apresentar bons resultados em espaços de busca convexos e em problemas com poucos objectivos.

Como desvantagens do WBGA, podem ser salientadas as seguintes: a dificuldade em gerar um conjunto de pesos que reflitam adequadamente os objectivos quando o conhecimento do problema é pequeno e não conseguir gerar adequadamente soluções não-dominadas quando a frente de Pareto é não-convexa.

Tendo em vista que tal tipo de algoritmo requer frentes de Pareto convexas, facto que não pode ser garantido para muitos problemas de optimização combinatoria multiobjectivo, não se justifica a adoção desta abordagem.

Vector Evaluated Genetic Algorithm (VEGA)

Abordagem proposta por Schaffer (1985). Num problema com k objectivos, a população P é dividida em k sub-populações, cada uma de tamanho P/k . O VEGA difere de um GA tradicional apenas pelo operador de selecção, de modo que, em cada geração, um número de sub-populações é gerado por meio de uma selecção proporcional relacionada com cada objectivo em análise.

Como vantagens do algoritmo VEGA destacam-se o aproveitamento de boa parte da estrutura de um GA padrão (mono-objectivo) e a performance eficiente. Como desvantagens, por trabalhar basicamente como um WPGA, padece das suas mesmas limitações.

Tendo em vista que o VEGA também não trabalha de forma eficaz em problemas cuja frente de Pareto possui formato não-convexo, não se justifica a adoção desta técnica.

Multi-Objective Genetic Algorithm (MOGA)

Heurística proposta por Fonseca e Fleming (1993), esta abordagem consiste num esquema de ordenação dos indivíduos da população com base no critério de dominância. Às soluções não-dominadas, são atribuídas classificação 1, enquanto as soluções dominadas são penalizadas de acordo com a densidade da população na região da frente de Pareto.

Como vantagens do MOGA, pode-se ressaltar que este consiste numa extensão simples de um GA mono-objectivo e possui uma performance eficiente. Contudo, o MOGA usualmente apresenta uma baixa convergência para a frente de Pareto e existem problemas reportados sobre o impacto do parâmetro relacionado ao tamanho dos nichos sobre a qualidade das soluções obtidas.

Tendo em vista as desvantagens reportadas sobre o MOGA, também não se mostra adequada a adoção deste tipo de algoritmo.

Nondominated Sorting Genetic Algorithm (NSGA)

O NSGA, proposto por Srinivas e Deb (1994), utiliza um procedimento designado por compartilhamento de aptidão (*fitness sharing*), o qual se destina a dividir a população em diversos nichos, os quais se comportam como sub-populações. Conforme Arroyo (2002), a lógica da formação dos nichos relaciona-se com a proximidade das soluções. Soluções próximas entre si, num mesmo nicho, são penalizadas. Assim, soluções que pertencem a nichos menos densos são preferíveis. Procedendo-se desta forma, dá-se maior prioridade à exploração de soluções não-dominadas em regiões inexploradas da fronteira de Pareto.

No NSGA, as soluções são divididas em diversas frentes de soluções não-dominadas. As frentes dominadas são retiradas, enquanto todos os indivíduos da população são classificados.

Conforme Coello (2001), diversos investigadores têm reportado que o NSGA apresenta um desempenho geral inferior ao do MOGA, tanto em termos de eficiência computacional quanto em termos da qualidade das soluções geradas.

Deb *et al.* (2000) apontam os seguintes problemas no NSGA clássico: (i) elevada complexidade computacional na ordenação (*sorting*) de soluções não-dominadas; (ii) falta de elitismo; e (iii) necessidade da especificação do parâmetro de compartilhamento de aptidão.

Deb *et al.* (2000) incorporam, no NSGA, procedimentos para suprir todas estas limitações, propondo o NSGA-II, o qual sobrepuja o NSGA em performance e em qualidade das soluções obtidas.

Strength Pareto Evolutionary Algorithm (SPEA)

O SPEA, proposto por Zitzler e Thiele (1999), utiliza o conceito de dominância de Pareto como forma de designar valores de aptidão aos indivíduos. A população é dividida em dois sub-conjuntos: uma população inicial P e uma população externa de soluções não-dominadas P' . A aptidão de um indivíduo é determinada somente com base nas soluções constantes na população externa, enquanto as demais soluções não são consideradas para tanto.

O tamanho da população P' influencia o desempenho do SPEA, pois se a quantidade de soluções não-dominadas aumenta indefinidamente, pode ocorrer uma redução da pressão seletiva e a busca pode perder a qualidade da aproximação da fronteira de eficiência.

Para controlar o tamanho do conjunto de soluções não-dominadas, o SPEA trabalha com um agrupamento (*clustering*) para reduzir o número de soluções não-dominadas armazenadas em P' , sem destruir as características da frente de Pareto.

Um ponto positivo do SPEA é que não requer nenhum parâmetro para o agrupamento das soluções. Contudo, uma desvantagem do SPEA é que, quando a quantidade de soluções não-dominadas excede um valor especificado *a priori*, um algoritmo de agrupamento é executado. Deste modo, além da dificuldade de implementação, tem-se, ainda, um elevado custo computacional.

Zitzler *et al.* (2001) apontam três potenciais fraquezas do SPEA tradicional: (i) esquema de compartilhamento de aptidão; (ii) falta de utilização de informação sobre a densidade da população, de modo a guiar a busca mais eficientemente; e (iii) perda de

dispersão no conjunto das soluções não-dominadas devida ao modo de truncamento da população.

Ao incorporarem mecanismos que mitigam tais limitações, Zitzler *et al.* (2001) desenvolvem o SPEA-2, o qual supera o SPEA tradicional.

Pareto Envelope-based Selection Algorithm (PESA)

O PESA, proposto por Corne *et al.* (2000), consiste numa metaheurística evolucionária multiobjectivo baseada no conceito de hiper grade (*hiper grid*). A hiper grade é uma grade no hiperplano, a qual tem por objectivo controlar a densidade de soluções no espaço normalizado dos objectivos. Tal mecanismo é de fundamental importância para a eficiência dos operadores de selecção, bem como para a manutenção da diversidade da população.

A estratégia de subdividir o espaço de soluções em hipercubos tem a finalidade de permitir a avaliação da densidade de soluções em uma dada célula do *grid*. Células com menor densidade de soluções são preferíveis a células com maior densidade de soluções. Isto ocorre porque hipercubos com menor densidade de soluções representam regiões promissoras da frente de Pareto que apresentam uma baixa representatividade na população corrente.

A seguir, na Figura 14, é ilustrado o conceito de hiper-cubo. Considere-se um problema de optimização envolvendo duas funções objectivo f_1 e f_2 . Admitiu-se que cada eixo fosse dividido em 4 partes. O número de hiper-cubos é dado por n^k , em que n é o número de grades e k é o número de objectivos. Logo, no exemplos ilustrado a seguir, tem-se 4^2 hipercubos.

A densidade de um hiper-cubo é dada pelo número de soluções não-dominadas nele contido. Cada hiper-cubo é definido por faixas de valores admissíveis das funções objectivo. Ao avaliar a aptidão de uma solução com base nos objectivos em análise, pode-se determinar a qual hiper-cubo a solução pertence.

No exemplo abaixo, tem-se um hiper-cubo com densidade igual a 1, dois hipercubos com densidade igual a 2, um hiper-cubo com densidade igual a 3 e os demais hipercubos com densidade nula.

Visualmente percebe-se que, se um hiper-cubo é denso, há uma concentração de soluções em torno de uma aproximação para a frente de Pareto. Em hipercubos esparsos, percebe-se a possibilidade de se ter uma melhor exploração desta região.

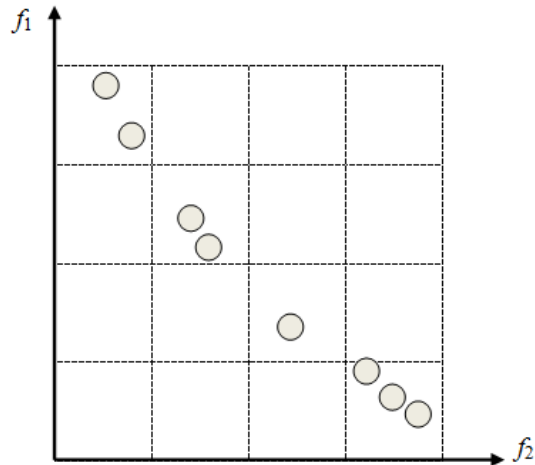


Figura 14: Ilustração do conceito de hipercubo.

Explorar células com menor densidade de soluções, além de incorrer em uma melhor exploração das regiões próximas da fronteira de Pareto, ajuda também a evitar uma possível convergência prematura da população.

Corne *et al.* (2001) entendem o conceito de densidade das células como um critério para a qualidade das soluções, consubstanciado no algoritmo PESA-II. Nesta abordagem, na aplicação de um operador de selecção, como, por exemplo, o torneio binário, uma célula que é esparsamente ocupada tem maior chance de ser seleccionada do que uma célula densamente ocupada. Uma vez que uma dada célula é seleccionada, uma de entre as soluções que compõem a célula é seleccionada aleatoriamente para o emprego dos operadores genéticos.

Conforme Corne *et al.* (2001), o PESA-II, ao utilizar um procedimento de selecção baseado em regiões, e não em indivíduos, apresenta uma probabilidade muito maior de explorar regiões pouco densas na fronteira de Pareto. Deste modo, tende a obter uma aproximação de soluções não-dominadas melhor distribuída no entorno da curva de eficiência.

Uma vantagem do PESA é que os mecanismos de controle da diversidade e de selecção de indivíduos na população funcionam por meio do mesmo mecanismo, enquanto em outras heurísticas multiobjectivo, são requeridos procedimentos distintos para estas duas operações.

Uma desvantagem do PESA é o seu desempenho possui uma dependência do parâmetro que controla a hiper grade.

A seguir, na Tabela 21, é ilustrada uma síntese dos Algoritmos Genéticos multi-objectivos analisados.

Tabela 21: Síntese das características dos algoritmos analisados.

Algoritmo	Princípio	Vantagens	Desvantagens
WBGA	Um vector de pesos converte um problem mono-objectivo e multiobjectivo.	(i) aproveita a implementação do GA; (ii) performance eficiente; (iii) trabalha bem em espaços convexos.	(i) requerem algum conhecimento do problema; (ii) ineficaz quanto o espaço é não-convexo.
VEGA	População dividida em sub-populações, cada qual avaliada para um dos objectivos.	(i) idem ao WBGA.	(i) idem ao WBGA.
MOGA	Ordenação dos indivíduos com base no conceito de dominância de Pareto.	(i) é uma extensão simples do GA; (ii) fácil implementação e performance eficiente.	(i) baixa convergência para frente de Pareto; (ii) forte dependência do parâmetro relacionado ao tamanho dos nichos.
NSGA	Compartilhamento da aptidão (<i>fitness sharing</i>) por meio da divisão da população em nichos	(i) rápida convergência para frente de Pareto.	(i) elevado custo na ordenação; (ii) falta de elitismo; e (iii) parâmetro de compartilhamento de aptidão.
NSGA-II	Ordenação eficientes das soluções.	(i) redução da complexidade computacional; (ii) redução da dependência de parâmetros.	(i) trabalha apenas no espaço dos objectivos.
SPEA	Trabalha com duas populações (interna e externa), usando <i>clustering</i> para controlar a população de soluções não-dominadas.	(i) bastante testado, apresentando excelente performance; (ii) não carece de parâmetro para o <i>clustering</i> .	(i) compartilhamento de aptidão; (ii) falta de utilização de informação sobre a densidade da população; (iii) perda de dispersão devida ao <i>clustering</i> .
SPEA-II	Considera-se, para cada indivíduo, quantos indivíduos o dominam e quantos são dominados por ele.	(i) uso de informações sobre densidade; (ii) aprimoramento do processo de <i>clustering</i> .	(i) elevado custo computacional.
PESA	Divide o espaço dos objectivos em hiper-cubos.	(i) melhor exploração no entorno da frente de Pareto; (ii) mitiga a convergência prematura da população; (iii) controle de diversidade e selecção operam por um mesmo mecanismo. (iv) fácil implementação; (v) performance eficiente.	(i) dependência do parâmetro de formação dos hiper-cubos.
PESA-II	Trabalha com hiper-cubos, ao invés de trabalhar com soluções.	(i) idem PESA. (ii) maior probabilidade em explorar regiões pouco densas na frente de Pareto.	(i) dependência do parâmetro de formação dos hiper-cubos.

5.4.2 Seleção da heurística a utilizar

Após a análise dos algoritmos reportados acima, procurou-se-se optar por um deles com base nos seguintes critérios: qualidade das soluções, desempenho computacional e facilidade de implementação algorítmica.

Foi possível constatar que os algoritmos WBGA, VEGA e MOGA estavam em um patamar inferior de eficácia em relação aos algoritmos NSGA, SPEA e PESA. Assim, a análise será centrada nestes últimos algoritmos.

Nos diversos resultados reportados em Zitzler e Thiele (1999), Corne *et al.* (2000), Zitzler *et al.* (2001) e Corne *et al.* (2001), pode-se ressaltar que os algoritmos que apresentaram os melhores resultados foram SPEA, SPEA-II, NSGA-II, PESA e PESA-II.

Em Zitzler e Thiele (1999), o SPEA foi testado em diversos conjuntos de dados e suplantou todos os outros algoritmos comparados, dentre eles, o VEGA e o NSGA, os quais foram descritos acima. Em Corne *et al.* (2000), o PESA sobrepujou o SPEA. Em Zitzler *et al.* (2001), após extensos experimentos computacionais, o SPEA-II sobrepujou o SPEA. SPEA-II e o NSGA-II apresentaram o melhor desempenho geral, embora o PESA apresentasse uma convergência mais rápida. Em problemas com elevado número de objectivos, o SPEA-II apresentou vantagens sobre o NSGA-II e o PESA. Em Corne *et al.* (2001), o PESA-II suplantou o PESA. Não foram reportadas comparações entre o SPEA-II e o NSGA-II com o PESA-II.

Diante do levantamento sucintamente descrito no parágrafo acima, com base em critérios de eficácia e eficiência, não se pode concluir sobre um *melhor* algoritmo. Entretanto, sob o prisma da implementação, a abordagem PESA-II é mais simples do que SPEA-II e NSGA-II. Portanto, com base neste último critério, justifica-se a adoção do PESA-II como extensão multiobjectivo da presente tese.

Mais uma vez, deve-se salientar que o foco da investigação não consiste no método em si, mas em avaliar como a otimização multiobjectivo pode contribuir, em termos práticos, para a resolução do VCSP. Com base nos resultados reportados nas referências citadas anteriormente, espera-se que a adoção de SPEA-II, NSGA-II ou PESA-II leve a resultados de qualidade equiparável.

5.4.3 Um Algoritmo Genético multi-objectivo para o VCSP

O algoritmo que se propõe neste trabalho é baseado na heurística PESA-II. A seguir, no Quadro 16, é ilustrado o algoritmo PESA-II proposto para resolução do MOMCP, aplicado ao VCSP.

A população de soluções é denominada P_i e a população de soluções não-dominadas é denominada P_e . Após a geração da população inicial, é gerada a população de soluções eficientes, a qual é actualizada a cada geração. A selecção é efectuada com base nos hiper-cubos e não nas soluções, como ocorre com o GA original.

A cada geração é gerada uma nova prole, a qual sofre mutação. Em seguida, são aplicados os procedimentos de melhoria (reparo da viabilidade e inserção de novas colunas, caso possível). Por fim, actualizam-se P_i e P_e . Todas as etapas do algoritmo proposto são detalhadas a seguir.

Quadro 16: Algoritmo PESA-II proposto para o VCSP.

Passo 1: Gerar aleatoriamente e avaliar uma população de soluções.
Passo 2: Dividir o espaço normalizado dos objectivos em $ngrid^k$ hipercubos.
Passo 3: Actualizar o conjunto de soluções não-dominadas P_e .
Enquanto geração actual \leq número máximo de gerações, faça
 Passo 4: Efectuar selecção por torneio entre hipercubos.
 Passo 5: Realizar o cruzamento uniforme com $p_c=100\%$.
 Passo 6: Realizar mutação da prole com $p_m=100\%$.
 Passo 7: Aplicar os procedimentos **DROP** e **ADD**.
 Passo 8: Avaliar a aptidão da solução gerada.
 Passo 9: Actualizar P_i e P_e . Incrementar o número de gerações.
Fim-do-enquanto
Imprimir o conjunto de soluções não-dominadas obtido

Codificação

Com base nos benefícios reportados na secção 4.3 desta tese, adoptou-se a codificação binária.

Considere-se um vector de dimensão n , em que n é o número de variáveis de decisão (ou seja, quantidade de serviços gerados). Se $x_j = 1$ a j -ésima variável de decisão fará parte da solução; caso contrário, se $x_j = 0$, a j -ésima variável de decisão não fará parte da solução.

Avaliação das soluções

Uma solução pode ser vista como um vector \mathbf{x} . O vector \mathbf{c} armazena os custos de cada serviço. A relação entre os vectores \mathbf{x} e \mathbf{w} é dada por $\mathbf{w} = \mathbf{A} \times \mathbf{x}^T$.

Se $w_i = 0$, então a i -ésima linha está descoberta na solução corrente; se $w_i = 1$, então a i -ésima linha é coberta por apenas uma coluna na solução actual; e, por fim, se $w_i \geq 2$ então ocorre pelo menos um *overcover*.

A cada serviço é associado um custo, sendo os custos armazenados num vector **c**.

A função de avaliação do objectivo 1 (vide equação (5.4)) é dada pela soma dos elementos do vector **w** iguais a zero (ou, seja, os *leftovers*). A função de avaliação do objectivo 2 (vide equação (5.5)) é dada pelo produto escalar dos vectores **c** e **x**.

Deve-se observar que soluções inviáveis, isto é, soluções com *overcovers* e soluções com mais de d colunas, não são permitidas.

Formação dos hipercubos

A formação de hipercubos é controlada por um parâmetro *ngrid* que define o número de subdivisões dos eixos do espaço das funções objectivo. O espaço das funções objectivo deve ser limitado e normalizado pelos valores mínimos e máximos das funções objectivo em análise.

Deve-se observar que o espaço será dividido em $ngrid^k$ hipercubos, em que k é o número de objectivos. Assim, constata-se que a quantidade de hipercubos cresce exponencialmente com a quantidade de objectivos.

Geração da população inicial

A população inicial é gerada de forma aleatória, levando em conta a viabilidade das soluções geradas, de modo que são gerados apenas indivíduos viáveis tendo em conta o modelo adoptado (MOMCP), vide equações (5.4) a (5.9). A seguir, no Quadro 17, é ilustrado o procedimento para a construção de uma solução, em que AV é o número de colunas disponíveis na solução, a cada iteração.

Quadro 17: Procedimento para geração de um indivíduo população inicial.

```

AV ← xj, ∀ j= 1, ..., n
col ← 0
enquanto AV ≠ ∅ ou col ≤ d faça
    seleccionar aleatoriamente uma coluna xj
    solução ← xj
    Actualiza AV
    col ← col +1
fim-do-enquanto

```

No início do algoritmo todas as colunas pertencem à lista AV. Em seguida, uma coluna x_j , seleccionada aleatoriamente, é adicionada à solução. A lista AV é então actualizada, sendo removida a coluna inserida e as demais colunas que cobrirem as mesmas linhas cobertas pela coluna inserida, de modo que as próximas inserções satisfaçam as restrições de igualdade.

Avaliação das soluções não dominadas

Após a geração da população, esta é subdividida em duas populações: uma população interna P_i e uma população externa P_e . As soluções não-dominadas são armazenadas em P_e enquanto as demais soluções são armazenadas em P_i . A seguir, no Quadro 18, é descrito o algoritmo para obtenção de soluções não-dominadas.

Quadro 18: Obtenção das soluções não-dominadas na população inicial.

```

p=2;
para i=1:size_front-1 faça
  para j=p:size_front faça
    se front(i)=0 então
      se  $\forall k f_k(\mathbf{x}') \leq f_k(\mathbf{x})$  e  $f_k(\mathbf{x}') < f_k(\mathbf{x})$  para pelo menos um k
        front(j)←1;
      fim-do-se
    senão
      front(i)←1;
    fim-do-senão
  fim-do-se
fim-do-para
p=p+1;
fim-do-para

```

Na população inicial, o vector *front* tem cardinalidade *maxpop* e distingue os elementos dominados dos não-dominados. A *i*-ésima componente de *front* representa o *i*-ésimo indivíduo da população e sendo o vector inicializado com elementos nulos. Cada indivíduo é comparado com os demais. De acordo com o conceito de dominância de Pareto definido da secção 5.1 do presente Capítulo, rotula-se uma solução dominada com o valor 1. Portanto, aquelas soluções em que $front(i)=0$ formarão P_e e aquelas soluções em que $front(i)=1$ formarão P_i . Nas gerações seguintes, *front* terá a cardinalidade do conjunto de soluções não-dominadas.

Após a actualização das soluções não-dominadas, as soluções dominadas (caso ocorram) são incorporadas na população interna. Existe um mecanismo para o controlo do tamanho da população interna: a soma das soluções nas populações interna e externa

não pode exceder ao tamanho da população. Caso ocorra que esta soma seja maior do que o tamanho da população, soluções dominadas são descartadas aleatoriamente, de forma a manter-se a relação de igualdade.

Seleccção

A seleccção dos pais a serem cruzados é efectuada por um torneio com características variáveis, as quais se relacionam com a densidade dos hipercubos.

No algoritmo proposto, os pais a serem seleccionados não são soluções, mas sim hipercubos. Os hipercubos mais aptos, ou seja, aqueles com menores valores de densidade, serão escolhidos como pais.

Por densidade de um hipercubo, entende-se o número de soluções não-dominadas nele contidas. Após a seleccção de um hipercubo, escolhe-se, de forma aleatória, uma solução nele contida para a operação de cruzamento.

A seleccção será feita por torneio binário tradicional sempre que existiram quatro ou mais hipercubos com densidade maior ou igual a 1. Neste caso, seleccionam-se aleatoriamente 4 hipercubos e são realizados torneios com dois deles.

Quando o número de hipercubos com densidades maiores ou iguais a 1 é inferior a 4, situação esta que ocorre mais frequentemente nas primeiras gerações da busca, procede-se conforme descrito a seguir. Seleccionam-se aleatoriamente 2 hipercubos para a realização de um torneio. Após a seleccção do hipercubo, selecciona-se aleatoriamente uma solução pertencente ao hipercubo vencedor do torneio. Esta solução será recombinada com uma solução pertencente à população interna, escolhida aleatoriamente.

No caso em que só existe um hipercubo com densidade maior ou igual a 1, selecciona-se aleatoriamente uma solução deste. Em seguida, selecciona-se aleatoriamente uma solução da população interna.

Percebe-se, portanto, que se pode efectuar o cruzamento entre soluções dominadas e não-dominadas, conduta que promove o espalhamento das soluções não-dominadas ao longo da aproximação de frente de Pareto.

Recombinação

O cruzamento é efectuado com probabilidade igual a 100%, ou seja, a cada geração, é gerada uma nova solução. O tipo de cruzamento utilizado foi o uniforme. Cada gene da solução filho a ser gerada é gerado com base numa distribuição de

Bernoulli, na qual uma variável aleatória discreta b recebe um valor 0 ou 1. Se b igual a 0, o filho receberá o valor correspondente ao gene do primeiro pai. Se b igual a 1, o filho receberá o valor correspondente ao gene do segundo pai.

O cruzamento uniforme foi adoptado por dois motivos: (i) possui fácil implementação computacional; e (ii) é capaz de combinar todo e qualquer esquema existente.

Mutação

A operação de mutação ocorre da seguinte forma: é gerado um número aleatório compreendido no intervalo $[0,100]$ e se este é menor ou igual à probabilidade de mutação considerada, o operador de mutação é acionado. É gerado um número aleatório r compreendido no intervalo $[0,n]$ e o bit correspondente ao número aleatório gerado é invertido (por exemplo, se $x_r = 1$ muda-se x_r para 0 e se $x_r = 0$ muda-se x_r para 1).

Procedimento de melhoria

Conforme reportado no Capítulo 4, secção 4.3, a correção da viabilidade das soluções é vital para a qualidade das soluções da serem geradas no MCP. A seguir, nos Quadros 19 e 20 são apresentados os procedimentos de melhoria. Para uma descrição mais detalhada destes, consultar a secção 4.3.

Quadro 19: Procedimento DROP.

```

w ←  $A \times \mathbf{x}^T$ 
enquanto (soma(w>1)≠0) faça
  para  $i=1$  até  $n$ 
    se  $w_i \geq 2$  então
      selecciona aleatoriamente coluna a descobrir a linha
      remove a coluna na solução
      w ←  $A \times \mathbf{x}^T$ 
    fim-do-se
  fim-do-para
fim-do-enquanto

```

Quadro 20: Procedimento ADD.

```

Actualiza AV
Actualiza  $col$ 
enquanto  $AV \neq \emptyset$  ou  $col \leq d$  faça
  seleccionar aleatoriamente uma coluna  $x_j$ 
  solução ←  $x_j$ 
  Actualiza AV
   $col \leftarrow col + 1$ 
fim-do-enquanto

```

5.5 Resultados computacionais

5.5.1 Apresentação dos testes

Com vista a validar o algoritmo PESA-II proposto para o VCSP, foram realizados testes computacionais com instâncias do MOMCP.

Foram adaptadas instâncias do SPP disponíveis na biblioteca de Investigação Operacional OR-Library (<http://people.brunel.ac.uk/~mastjib/jeb/info.html>). Atribuiu-se um valor do número máximo de colunas (d) equivalente a $\lceil 0.2 \times m \rceil$.

Deve-se ressaltar que seleccionando os objectivos minimizar *leftovers* e minimizar custos, tem-se um caso particular do VCSP. Não vai ser solucionado um problema genérico, pois o intuito destes testes é ilustrar os resultados que a abordagem heurística proposta pode proporcionar, adoptando o MOMCP com estes dois objectivos. Admite-se como natural a extensão desta abordagem com a consideração de outros objectivos.

As características das instâncias são as seguintes: número de linhas da matriz A (m), número de colunas da matriz A (n), número máximo de colunas a serem utilizadas na solução (d) e densidade da matriz A (ρ). O valor de ρ é obtido pela divisão do número de elementos da matriz A iguais a 1 pelo número total de elementos de A . Estas informações constam nas colunas 1 a 5 da Tabela 22.

A seguir, são apresentados os parâmetros utilizados no algoritmo PESA-II proposto. Tais parâmetros foram ajustados empiricamente após alguns testes computacionais, permitindo a análise do comportamento dos algoritmos.

- Tamanho da população (*maxpop*): 100
- Número de gerações (*maxgen*): 10000
- Probabilidade de mutação (*p_{mut}*): 100%
- Espaçamento da grade (*ngrid*): 25

Tendo em vista que a metaheurística PESA-II é um processo de amostragem aleatória, optou-se por rodar o algoritmo 10 vezes, para avaliar o comportamento médio do mesmo, em termos de qualidade de solução obtida e de tempo de processamento. Os experimentos foram realizados em um processador Genuine Intel 1.86 GHz com 1GB de memória RAM.

Tabela 22: Características das instâncias adaptadas do SPP.

1. Instância	2. m	3. n	4. d	5. ρ (%)
NW41	17	197	4	22,10
NW32	19	197	4	24,30
NW40	19	404	4	26,95
NW08	24	434	5	22,39
NW15	31	467	7	19,55
NW21	25	577	5	24,89
NW22	23	619	5	23,87
NW12	27	626	6	20,00
NW39	25	677	5	26,55
NW20	22	685	5	24,70

5.5.2 Resultados obtidos

A seguir, na Tabela 23, são apresentadas as quantidades de soluções não-dominadas obtidas por PESA-II, em cada uma das 10 execuções do algoritmo. Na Tabela 24, é apresentado o desempenho computacional, expresso em tempos de processamento, da heurística proposta.

Com base nos resultados obtidos, pode-se observar que existe uma maior dificuldade em obter aproximações da frente de Pareto em problemas com uma maior quantidade de linhas da matriz A . A quantidade de colunas da matriz A apresenta um maior impacto no tempo de processamento.

A seguir, na Figura 15, é ilustrada uma aproximação da fronteira de Pareto obtida pelo algoritmo PESA-II proposto.

Tabela 23: Quantidade de soluções não-dominadas obtidas pela heurística PESA-II proposta.

Problema	Número de soluções não-dominadas em cada uma das 10 execuções do PESA-II										Média
	1	2	3	4	5	6	7	8	9	10	
NW41	12	10	9	12	12	10	10	11	12	9	10,7
NW32	9	9	10	8	8	12	10	9	10	10	9,5
NW40	13	13	13	13	12	12	12	13	11	13	12,5
NW08	11	13	13	10	12	13	13	9	9	11	11,4
NW15	6	6	5	7	4	7	6	5	5	7	5,8
NW21	8	8	10	12	8	8	8	9	10	10	9,1
NW22	8	7	11	9	6	7	9	8	8	7	8
NW12	10	10	13	13	15	10	9	9	14	11	11,4
NW39	12	13	9	9	11	8	13	10	13	7	10,5
NW20	10	6	12	13	10	11	9	9	12	7	9,9

Tabela 24: Desempenho computacional da heurística PESA-II proposta.

Problema	Tempo computacional (s) em cada uma das 10 execuções do PESA-II										Média (s)
	1	2	3	4	5	6	7	8	9	10	
NW41	13,8	14,0	13,5	13,7	14,0	14,0	13,7	14,2	13,7	13,6	13,8
NW32	15,7	15,8	16,2	15,1	14,7	14,4	15,9	14,8	15,7	16,2	15,4
NW40	26,3	28,2	28,2	28,2	28,8	27,2	28,3	26,6	27,8	27,2	27,7
NW08	33,1	32,3	35,7	36,2	33,8	37,4	36,7	34,7	35,1	33,4	34,8
NW15	49,4	50,3	51,1	50,7	49,2	50,7	52,1	52,0	50,4	51,3	50,7
NW21	43,7	43,5	52,0	45,7	45,6	45,2	45,3	45,8	46,4	45,2	45,8
NW22	47,6	49,8	51,6	51,6	43,9	43,5	49,4	51,9	49,9	49,6	48,9
NW12	57,9	62,5	57,2	64,4	61,7	61,2	59,4	55,7	62,9	54,9	59,8
NW39	58,5	52,5	65,5	66,3	65,9	61,2	69,0	64,3	61,7	49,2	61,4
NW20	61,2	59,0	68,5	63,7	60,4	61,3	63,5	65,0	65,4	59,2	62,7

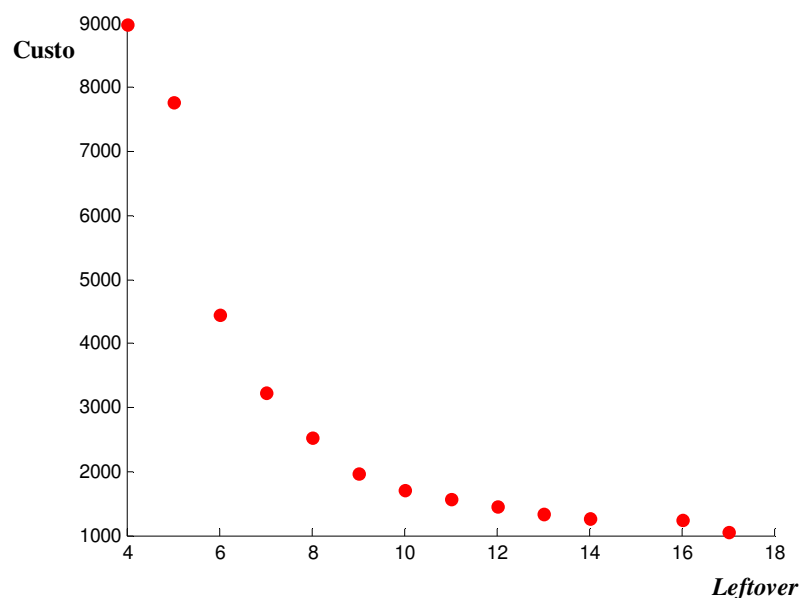


Figura 15: Aproximação para a fronteira de Pareto – MOMCP (instância NW12).

5.5.3 Avaliação da qualidade das soluções

De acordo com Deb (2000), a otimização multiobjectivo possui duas finalidades distintas: (i) descobrir soluções tão próximas quanto possível da frente de Pareto; e (ii) encontrar soluções tão diversas quanto possível na fronteira de Pareto. Estes objectivos são conflituosos: o primeiro reflecte a proximidade das soluções não-dominadas relativamente à frente de Pareto e o segundo refere-se à diversidade das soluções não-dominadas ao longo da aproximação da fronteira de Pareto.

Uma dificuldade da avaliação de metaheurísticas multiobjectivo é que, em muitos casos, a fronteira de Pareto não é conhecida. Deste modo, a determinação de quão próximas e diversificadas são as soluções geradas pelas metaheurísticas pode ser

um pouco imprecisa. Na opinião do autor, a avaliação de performance de metaheurísticas multiobjectivo ainda é um campo que necessita de maior aprimoramento.

Deb (2001) apresenta diversas métricas para avaliação de métodos de otimização multiobjectivo, os quais são classificados em três grupos: (i) métricas que avaliam a proximidade à fronteira de Pareto; (ii) métricas que avaliam a diversidade das soluções não-dominadas; e (iii) métricas que avaliam proximidade e diversidade.

Nesse contexto, optou-se por adoptar uma métrica que avaliasse tanto a proximidade quanto a diversidade, visto que estes dois objectivos são fundamentais quando se trata de otimização multiobjectivo.

A métrica escolhida foi a do hipervolume, devido sua fácil implementação e ao facto de que, mesmo desconhecendo com exatidão a frente de Pareto para um dado problema, o hipervolume de um conjunto de soluções não-dominadas pode propiciar informação acerca da qualidade das soluções geradas.

A métrica do hipervolume pode ser explicada como segue: considere-se um problema de otimização multiobjectivo no qual todos os objectivos são de minimização. Dado um ponto de referência W , cada solução não-dominada delimita um hipercubo com relação à W . O ponto de referência W pode representar as coordenadas equivalentes aos piores valores possíveis no espaço das funções objectivo. Seja $|Q|$ a cardinalidade do conjunto de soluções não-dominadas, a união dos i volumes (mutuamente exclusivos) destes hipercubos compreende o hipervolume HV :

$$HV = volume \left(\bigcup_{i=1}^{|Q|} v_i \right) \quad (5.10)$$

Deve-se observar que quanto maior o valor de HV , mais distante do ponto W se encontram as soluções geradas, e, por conseguinte, mais próximas da frente e mais e diversificadas ao longo da frente elas estão.

A seguir, na Tabela 25, são apresentados os hipervolumes obtidos em cada execução do PESA-II, para o conjunto de instâncias avaliado.

Embora não se tenha um marco de comparação, como, por exemplo, soluções geradas por outra abordagem multiobjectivo, o hipervolume permite uma avaliação da qualidade das soluções entre as diversas execuções do PESA-II.

Por exemplo, considerem-se as execuções 7 e 10 da instância NW41. A sétima execução, apesar de gerar uma quantidade de soluções não-dominadas maior do que a décima execução, corresponde a um hipervolume menor. Isso significa que a aproximação da décima execução tende a ser melhor do que a da sétima execução.

Tabela 25: Hipervolumes obtidos com base nas soluções não-dominadas.

Problema	Hipervolume cada uma das 10 execuções do PESA-II									
	1	2	3	4	5	6	7	8	9	10
NW41	375612	363087	357450	375978	376428	363681	349416	366156	376035	357234
NW32	399957	399540	408303	384639	385524	425712	414798	381135	413604	413613
NW40	416409	418167	416583	418803	416649	405045	395532	416097	413496	418803
NW08	983662	984250	1032408	1027952	987626	1023868	1029920	1031826	1023754	985288
NW15	6858300	6801777	6815391	6866730	6692784	6867384	6802371	6815391	6813969	6867393
NW21	658214	668374	675862	657924	670456	670238	652810	674780	674484	675050
NW22	564036	562774	558030	567186	561060	561788	554682	566452	559456	550140
NW12	1210336	1255572	1208992	1260748	1259708	1193494	1221960	1221960	1263966	1111520
NW39	700434	702684	695811	703116	732624	693201	719886	735111	719265	683532
NW20	562860	555093	592545	574362	565236	571161	575535	587991	591804	571413

Como os hipervolumes gerados para cada instância possuem magnitudes diferentes, não podem ser comparados de uma forma direta. Um modo de comparar tais valores é por meio do Coeficiente de Variação (CV), o qual corresponde à razão entre o desvio padrão σ e a média μ . Usualmente o CV é expresso em valor percentual.

Tendo em vista que os hipervolumes para cada instância possuem distribuições diferentes, o uso do CV permite avaliar quanto o desvio padrão está oscilando em torno da média. Logo, por meio do CV, podem ser comparados os resultados, expressos em termos de hipervolumes, de instâncias diferentes. A seguir, na Tabela 26, são ilustradas as análises estatísticas dos hipervolumes obtidos.

Tabela 26: Análise estatística dos hipervolumes obtidos.

Instância	σ	μ	CV(%)
NW41	9661,7	366107,7	2,6
NW32	15088,9	402682,5	3,7
NW40	7505,2	413558,4	1,8
NW08	22449,8	1011055,4	2,2
NW15	52754,5	6820149,0	0,8
NW21	8422,2	667819,2	1,3
NW22	5250,5	560560,4	0,9
NW12	46093,4	1220825,6	3,8
NW39	17262,5	708566,4	2,4
NW20	12586,9	574800,0	2,2

Analisando os resultados de uma forma geral, pode-se constatar que os CV's dos hipervolumes obtidos são de pequena magnitude (todos inferiores a 4%). Tal comportamento representa a estabilidade dos resultados obtidos pelo PESA-II, visto que a variabilidade dos resultados não foi elevada.

5.6 Considerações finais

Este capítulo apresentou uma nova extensão multiobjectivo do Algoritmo Genético desenvolvido para o VCSP. Diante das diversas abordagens evolucionárias para a otimização multiobjectivo, optou-se pela implementação de um algoritmo PESA-II, devido sua fácil implementação computacional e os excelentes resultados da abordagem, reportados por outros trabalhos.

O algoritmo PESA-II foi implementado para o MOMCP, como uma abordagem para a resolução do VCSP. A variante do MOMCP não consiste em um problema geral, mas antes um caso particular. Não foi objectivo dos testes computacionais solucionar um problema real, mas avaliar o comportamento do algoritmo proposto, com vista a validá-lo para posterior aplicação prática.

Os objectivos seleccionados para o VCSP foram minimização dos *leftovers* e minimização dos custos dos serviços. Ao serem adoptados outros objectivos, ocorre uma natural extensão da abordagem proposta.

Para avaliar a capacidade da heurística desenvolvida em obter aproximações para a fronteira de Pareto, foram realizados experimentos computacionais em um conjunto de 10 instâncias advindas da OR-Library. Foi empregada a métrica do hipervolume para a avaliação da qualidade das soluções geradas. Os resultados obtidos apontam para a eficácia e a eficiência do algoritmo proposto.

Diante destes resultados, a extensão multiobjectivo será aplicada em casos de estudo reais, no Capítulo 6 da tese.

Capítulo 6

Aplicação aos casos de estudo

“Give me the fruitful error any time, full of seeds, bursting with its own corrections. You can keep your sterile truth for yourself”.

Vilfredo Pareto

Resumo: Este capítulo tem como objectivo apresentar os resultados da aplicação das abordagens desenvolvidas em problemas reais. Na primeira secção são apresentadas informações de carácter geral sobre os casos de estudo. Na segunda secção é descrito o caso de estudo de Fortaleza. Na terceira secção, é apresentado o caso de estudo de Portimão. Por fim, na quarta secção, são apresentadas as considerações finais sobre o capítulo.

6.1 Considerações iniciais

No presente capítulo, as abordagens desenvolvidas na presente tese, isto, é as duas novas formulações para o VCSP, bem como a metaheurística multiobjectivo proposta, são aplicadas em problemas reais. Os casos de estudo são referentes às cidades de Fortaleza (Brasil) e Portimão (Portugal).

No que diz respeito ao processo de amostragem das instâncias a serem testadas, não foi realizado um estudo estatístico rigoroso de determinação do tamanho da amostra. Em primeiro lugar, devido a restrições de tempo para realização dos testes computacionais, bem como da disponibilidade de informação, não se mostra viável solucionar uma grande parcela das instâncias (ou seja, linhas de transportes colectivos). Em segundo lugar, não é objectivo desta tese avaliar, de uma forma mais contundente, o método a ser utilizado, ou mesmo se o método utilizado é o melhor para a resolução dos

problemas em foco. O foco central da presente investigação é avaliar como as abordagens propostas podem contribuir, em termos práticos, para a resolução do VCSP.

6.2 Caso de estudo 1 – Fortaleza

6.2.1 O sistema de transporte público de Fortaleza

A cidade de Fortaleza, capital do Ceará, Estado situado na Região Nordeste do Brasil, é uma das metrópoles regionais brasileiras em crescente expansão. É a cidade brasileira com maior densidade populacional, possuindo cerca de 2,4 milhões de habitantes, distribuídos em apenas dois milésimos do território cearense, os quais representam aproximadamente um terço da população do Estado. Considerando sua Região Metropolitana, a população de Fortaleza é superior aos três milhões de habitantes.

Em Fortaleza, os empregos estão concentrados principalmente na área central e a maioria das viagens é efectuada pelo sistema de transporte por autocarros. A maior parte da população possui baixos rendimentos, dependendo do sistema de autocarros para viagens de trabalho, estudo, compras e lazer (AZEVEDO FILHO *et al.*, 1994).

O sistema de transporte público em Fortaleza constitui uma complexa rede, compreendendo uma série de linhas e terminais de transbordo. Apesar de tal complexidade, a programação dos veículos ainda é efectuada manualmente, o que incorre num elevado tempo de planeamento e em soluções cujas qualidades ainda poderiam ser significativamente melhoradas.

O transporte público em Fortaleza é composto por quatro modos: os autocarros, o transporte complementar (*topics ou vans*), os táxis e os mototáxis. Os modos que realizam o transporte de uma grande quantidade de passageiros são os autocarros e o transporte complementar (também chamado de transporte alternativo).

Apesar de operar com veículos de baixa e média capacidade, o transporte alternativo efectua a locomoção de diversos cidadãos fortalezenses. As *topics* possuem rotas que complementam as linhas de transporte por autocarros, mas, em diversos casos, competem com tais veículos nos grandes corredores de tráfego. A Empresa de Transporte Urbano de Fortaleza – ETUFOR é a empresa responsável pela programação dos veículos que operam no serviço de transporte alternativo em Fortaleza.

O transporte por autocarros pode-se realizar de duas formas: convencional e selectivo. O transporte selectivo é realizado por um número limitado de companhias e consiste na oferta de um serviço de maior qualidade, com número de linhas restrito e

valor de tarifa diferenciado. O transporte convencional por autocarros é aquele que será o objeto de estudo deste capítulo.

O sistema de transporte colectivo por autocarros em Fortaleza é denominado Sistema Integrado de Transportes – SITFOR. O SITFOR opera de acordo com um sistema de integração tarifária física e temporal, a qual possibilita um maior acesso aos passageiros a diversos pontos da cidade com o pagamento de uma única tarifa por viagem.

Actualmente, a rede de atendimento do SITFOR é composta por:

- sete estações de autocarros fechadas e integradas;
- duas estações de autocarros abertas e não integradas;
- rede de linhas, em sistema tronco-alimentado¹⁵; e
- cerca de 3650 paragens para autocarros.

Nas sete estações fechadas, o utente pode trocar de veículo quantas vezes desejar, sem a necessidade do pagamento de uma tarifa adicional, enquanto nas estações abertas, o utente necessita efectuar o pagamento de nova tarifa para mudar de condução.

Existe também um sistema de integração temporal, no qual o utente que detiver créditos em alguns tipos de cartões eletrónicos pré-determinados poderá trocar de veículo uma vez, dentro de uma janela de tempo relacionada com o tempo de ciclo da linha, sem ter a necessidade de ingressar num dos terminais fechados.

De um modo geral, as linhas que operam no SITFOR podem ser agrupadas em três grandes classes: linhas que fazem integração de carácter bairro-terminal, linhas que fazem integração entre um terminal e o centro da cidade e linhas que integram duas ou mais estações.

Programação de veículos

As características da programação de veículos são as seguintes:

- A programação dos autocarros é de responsabilidade da ETUFOR, empresa de capital misto que gere o transporte na cidade.

¹⁵ Sistema definido por um conjunto de estações para transbordo e conexão em pontos estratégicos de uma cidade. Com este formato de rede, há linhas que fazem a ligação entre o centro da cidade e as estações (sistemas radiais), com capacidade ampliada. Nas estações, os utentes fazem conexões com as linhas alimentadoras, que efectuem a distribuição dos passageiros para bairros ou outras estações. Procedendo-se desta forma, diminui-se o número de autocarros circulando no centro da cidade.

- A programação é realizada para cada linha, individualmente.
- Não são permitidas trocas de veículos entre linhas durante os intervalos para folgas.
- Cada linha é operada por um conjunto de empresas.
- Os tempos de viagens em vazio (*deadheads*) no início e no término da operação de um veículo (*pull-in* e *pull-out*) não são considerados na programação, uma vez que não se sabe, *a priori*, quais as empresas que vão operar uma determinada linha.
- Cada veículo tem autonomia de combustível para realizar a jornada.
- 22 empresas operam na cidade, com uma frota de cerca de 1700 veículos e um total de 221 linhas regulares e 22 linhas de operação especial (corujão). Existem ainda 16 linhas de transporte complementar.

Programação de tripulações

As características da programação de tripulações são as seguintes:

- A programação da tripulação não é de responsabilidade do órgão gestor (ETUFOR); todavia, as programações dos veículos devem respeitar a legislação laboral relativa às tripulações.
- A tripulação de um autocarro é formada por um motorista e um cobrador.
- A jornada de trabalho é de 07h20min, que corresponde ao valor mínimo remunerado à tripulação, sendo permitida uma jornada máxima de 09h20min, com a adição de horas extras, que são pagas com acréscimo de 50%.
- É obrigatório um intervalo (não remunerado) para lanche, que deve ter uma duração mínima de 30min. O intervalo de lanche deve, preferencialmente, estar compreendido num intervalo de 30min a 1h. Intervalos entre 1h e 2h são permitidos, mas geram reclamações entre os tripulantes.
- Um valor máximo de 30% das viagens pode possuir intervalos superiores a 2h (conhecidos como “duplas pegadas” ou “chupitilhas”).
- O intervalo do lanche deve ocorrer, fora dos horários de pico, preferencialmente entre as 9:00 e as 11:00, no turno da manhã, e entre as 14:30 e as 16:30, no turno da tarde.
- Não é permitido um intervalo para lanche a dois veículos que operem seguidamente numa mesma linha.

- É imposto que uma tripulação trabalhe no mesmo veículo durante sua jornada de trabalho, ou seja, *changeovers* não são permitidos.

Uma questão pertinente é a relação intrínseca entre a programação de veículos e pessoal tripulante em Fortaleza, visto que não são permitidas trocas de tripulações entre veículos.

Deste modo, faz pouco sentido prático programar os veículos sem levar em consideração os serviços do pessoal tripulante, o que torna necessária uma abordagem integrada para o problema.

6.2.2 Modelo mono-objetivo e otimização

Diante do exposto, o autor decidiu propor uma extensão do MCP ao VCSP, conforme será ilustrado a seguir. No Capítulo 3, é proposta uma variante do MCP para o VCSP. Tal variante será adequada às características do problema real da cidade de Fortaleza.

Dada uma matriz A , de ordem $m \times n$, com elementos $a_{ij} \in \{0,1\}$; uma matriz B , de ordem $m \times p$, com elementos $b_{ik} \in \{0,1\}$. A matriz A , a qual possui m linhas e n colunas, contém serviços viáveis que respeitem a legislação trabalhista, enquanto a matriz B , que possui m linhas e p colunas, contém serviços inviáveis. É importante destacar que não existe penalização em não se cobrir uma linha da matriz B , tendo em vista que este tipo de serviço não é prioritário.

Seja x_j uma variável de decisão binária que é igual a um se o j -ésimo serviço da matriz A está na solução, sendo zero caso contrário; y_k uma variável de decisão binária que é igual a um se o k -ésimo serviço da matriz B está na solução, sendo zero caso contrário; z_i uma variável de decisão binária que é igual a um se a i -ésimo tramo não é coberto, sendo zero caso contrário e γ o número máximo de serviços da matriz B permitidos.

O modelo matemático para o VCSP sem *changeovers*, baseado no MCP, é definido como segue:

$$\text{Minimizar} \quad \sum_{i=1}^m z_i \quad (6.1)$$

sujeito a:

$$\sum_{j=1}^n x_j + \sum_{k=1}^p y_k \leq d \quad (6.2)$$

$$\sum_{j=1}^n a_{ij} x_j + \sum_{k=1}^p b_{ik} y_k + z_i = 1 \quad \forall i = 1, \dots, m. \quad (6.3)$$

$$\sum_{k=1}^p y_k \leq \gamma \quad \forall k = 1, \dots, p. \quad (6.4)$$

$$x_j \in \{0,1\} \quad \forall j = 1, \dots, n. \quad (6.5)$$

$$y_k \in \{0,1\} \quad \forall k = 1, \dots, p. \quad (6.6)$$

$$z_i \in \{0,1\} \quad \forall i = 1, \dots, m. \quad (6.7)$$

A função objectivo representada pela equação (6.1) procura minimizar o número de tramos descobertos. Caso o i -ésimo tramo não possa ser coberto, $z_i = 1$. A restrição (6.2) impõe que um número máximo d de colunas das matrizes A e B sejam seleccionadas na solução. O conjunto de restrições do tipo (6.3) garante que os tramos serão cobertos por colunas sem interseções de linhas em colunas distintas, e, caso isso não seja possível, $z_i=1$. A restrição do tipo (6.4) impõe que a quantidade máxima de serviços inviáveis não seja excedida. Por serviço inviável entende-se um serviço que não atenda plenamente às regras trabalhistas. As expressões (6.5), (6.6) e (6.7) impõem o carácter binário das variáveis. Deve-se observar que as variáveis z_i são auxiliares, sendo que não representam, de facto, decisões.

Conforme Huisman (2004), como os *changeovers* são proibidos, considera-se que cada viagem constitui um tramo (*piece-of-work*). Deste modo, aloca-se um par veículo-tripulação a cada tramo, de modo que todas as viagens sejam cobertas com o menor número de veículos possível e que uma menor quantidade de horas pagas a tripulantes seja requerida.

A lógica para a programação de veículos e tripulantes utilizada foi a seguinte (i) geração de serviços possíveis (viáveis e inviáveis); e (ii) resolução do modelo matemático. A seguir, descrevem-se sucintamente estas fases.

Geração de serviços

A fase de geração de serviços é de fundamental importância para a qualidade das soluções geradas no processo de optimização. A técnica de busca em grafos conhecida como busca em profundidade (ou, em inglês, *Depth-First Search*) usualmente é empregada para a geração de todos os serviços factíveis. É do conjunto destes serviços, gerados em número muito elevado, que são posteriormente seleccionados os serviços a implementar na prática.

Entretanto, diversos trabalhos (KLABJAN *et al.*, 2001; KORNILAKIS e STAMATOPOULOS, 2002; AHMADBEYGI e COHN, 2006) reportam que a geração de todos os serviços factíveis pode dificultar, ou mesmo inviabilizar, o processo de optimização, tendo em vista que o espaço de soluções viáveis cresce exponencialmente com a quantidade de serviços gerados.

Diante do facto de que a programação de veículos em Fortaleza requer a geração de um conjunto de serviços inviáveis, um processo enumerativo, tal como *Depth-First Search*, poderia incorrer em custos computacionais proibitivos no processo de optimização. Portanto, optou-se por gerar serviços heurísticamente, conforme ilustrado no Quadro 21.

Quadro 21: Procedimento para geração de serviços.

```

Enquanto  $k \leq w$  faça
  para  $i=1$  até  $m$  faça
    para  $j=1$  até  $m$  faça
      alocar viagens aos serviços com a menor folga possível,
      alocando uma folga por serviço
    fim-do-para
  fim-do-para
fim-do-enquanto

```

Tal procedimento para geração de serviços pode, em alternativa, ser descrito como segue: o algoritmo procura gerar serviços com uma folga para lanche, alocando as demais viagens com a menor folga possível (se possível, com folga nula). Deste modo, a heurística procura gerar serviços com folgas tão pequenas quanto possível. A matriz *A* foi gerada com folgas com duração no intervalo [30;60], enquanto a matriz *B* foi gerada superiores a 60 minutos. Não foram gerados serviços com horas-extras, mas apenas serviços com uma duração máxima de 07h20min, devido à definição do modelo descrita na Secção 3 do corrente Capítulo.

Resultados computacionais

A seguir, na Tabela 27, são apresentados os resultados dos testes computacionais para o caso em que os veículos são alocados a um único depósito. A coluna 1 consiste na identificação da instância, a qual é representada pelo número da linha. Na coluna 2 apresenta-se o número de viagens da linha. Na coluna 3 apresenta-se o número de serviços viáveis gerados, enquanto na coluna 4 apresenta-se o número de serviços inviáveis gerados.

Na coluna 5 é ilustrado o tempo de ciclo da linha. Deve-se ressaltar que, no sistema de transporte público por autocarros de Fortaleza, o tempo de ciclo varia ao longo do dia. Assim, é apresentada a moda da variável tempo de ciclo. Na coluna 6 é apresentada a quantidade máxima de colunas a comporem a solução. Na coluna 7, é apresentada a densidade da matriz de serviços viáveis gerados, enquanto na coluna 8 é apresentada a densidade da matriz de serviços inviáveis gerados.

Na coluna 9, consta o número de tramos descobertos. Na coluna 10, consta o tempo de processamento para obtenção da solução ótima. Na coluna 11, apresenta-se o número de nós do B&B. Na coluna 12, é apresentado o número de iterações do B&B. Por fim, na coluna 13, é apresentada a percentagem de cobertura das viagens.

Os experimentos foram realizados em um processador Genuine Intel 1.86 GHz com 1GB de memória RAM, utilizando-se o *solver* LINGO.

Dos resultados obtidos, pode-se concluir que os maiores benefícios da abordagem desenvolvida (com base em um novo modelo matemático para o problema) dizem respeito à qualidade das soluções geradas e à redução nos tempos de planejamento.

Em termos de qualidade das soluções (vide Tabela 28), a abordagem proposta obteve, em termos médios, uma redução da ordem de 5% do percentual de cobertura dos tramos. Estes resultados serão discutidos posteriormente.

Enquanto a geração de escalas de veículos gerada manualmente pode levar horas ou mesmo dias para ser realizada pelos técnicos da ETUFOR, a abordagem proposta permite que a programação seja realizada em tempo computacional insignificante (geralmente inferior a 1 minuto), possibilitando uma maior flexibilidade no planejamento por parte dos técnicos da empresa. Estas economias em tempo têm um efeito adicional importante, já que permitem a realização de múltiplos ensaios num

reduzido período de trabalho, com a possibilidade de uma análise de sensibilidade das soluções propostas.

Diante da rapidez de reposta do modelo matemático proposto, não se justifica a implementação de uma heurística, visto que a obtenção das soluções ótimas já ocorre num tempo de processamento reduzido.

Tabela 27: Experimentos computacionais.

1. instância	2. m	3. n	4. p	5. Ciclo (min)	6. d	7. ρ_A	8. ρ_B	9. ZIP	10. t (s)	11. Número de <i>branches</i>	12. Número de iterações	13. c (%)
11	179	1005	1079	65	30	3,2	3,1	3	10	124	39718	98,3
466	74	509	509	56	10	9,2	9,0	4	1	89	5089	94,6
112	109	657	657	60	16	5,5	5,4	2	20	610	131278	98,2
407	67	247	247	90	20	5,5	5,4	4	1	38	2755	94,0
340	75	627	627	40	8	11,1	10,8	4	2	2	2793	94,7
360	214	1301	1301	60	32	2,8	2,8	7	476	14543	2225640	96,7
220	142	1004	1004	55	20	5,0	4,9	6	11	465	74471	95,8
501	64	489	503	48	8	12,2	12,0	3	1	16	1644	95,3
401	166	1164	1164	52	24	4,2	4,1	7	158	2680	858401	95,8
810	64	489	489	48	8	11,9	11,6	3	4	88	10945	95,3
905	49	258	258	72	8	21,2	10,0	1	1	0	427	98,0
15	111	801	804	56	14	6,5	6,4	4	106	3007	560602	96,4
122	64	440	440	52	8	10,7	10,5	4	20	635	204087	93,8
314	74	325	325	77	14	5,9	5,9	4	1	3	1085	94,6
70	83	295	302	100	22	4,4	4,3	1	1	4	900	98,8

Análise dos resultados obtidos

A comparação entre os resultados fornecidos pelo modelo e as soluções geradas manualmente foi realizada, mas não é completamente justa. Isto por que as soluções geradas manualmente flexibilizam ainda mais restrições laborais (por exemplo, permitem intervalos de folga inferiores a 30 minutos ou jornadas superiores à 9:20hs), o que não é permitido nos testes realizados. De qualquer modo, mesmo flexibilizando estas restrições, as soluções geradas manualmente, muitas vezes, requerem muitos serviços com horas-extras para cobertura das viagens.

A seguir, na Tabela 28, é ilustrada uma comparação entre as soluções geradas pelo modelo matemático e as soluções geradas manualmente, comparação esta efectuada em termos de tramos descobertos e de percentagem de cobertura.

Em relação aos resultados obtidos, podem ser ressaltadas as seguintes conclusões:

- para as 15 comparações efectuadas, o modelo matemático incorreu na melhoria do factor de cobertura em 11 instâncias, obteve resultados inferiores às soluções manuais em 3 instâncias e ocorreu 1 empate.
- o desvio da percentagem de cobertura encontra-se no intervalo $[-3,1;14,7]$, enquanto o desvio médio da percentagem de cobertura, para as 15 instâncias, foi de 4,7%.

Tabela 28: Comparação entre as soluções óptimas e as soluções geradas manualmente.

Instância	Viagens	Tramos descobertos		Cobertura (%)		Desvio (%)
		Modelo	Manual	Modelo	Manual	
11	179	3	5	98,3	97,2	1,1
466	74	4	12	94,6	83,8	10,8
112	109	2	4	98,2	96,3	1,8
407	67	4	4	94,0	94,0	0,0
340	75	4	6	94,7	92,0	2,7
360	214	7	30	96,7	86,0	10,7
220	142	6	17	95,8	88,0	7,7
501	64	3	1	95,3	98,4	-3,1
401	166	7	6	95,8	96,4	-0,6
810	64	3	1	95,3	98,4	-3,1
905	49	1	8	98,0	83,7	14,3
15	111	4	8	96,4	92,8	3,6
122	64	4	10	93,8	84,4	9,4
314	74	4	5	94,6	93,2	1,4
70	83	1	12	98,8	85,5	13,3

Diante do exposto, pode-se constatar que a abordagem de optimização proposta traz grandes benefícios relativamente à prática corrente do sistema de transporte público por autocarros em Fortaleza, seja na redução dos tempos de planeamento, seja na obtenção de soluções de maior qualidade. Também deve se destacar que os resultados obtidos corroboram a importância em se considerar uma abordagem integrada para o caso de Fortaleza.

6.2.3 Extensão multiobjectivo

Apesar da qualidade e consistência dos resultados obtidos meio da formulação matemática proposta, deve-se ressaltar que o problema em análise possui uma forte componente multiobjectivo. A seguir, serão descritos os objectivos escolhidos para serem avaliados.

(i) minimização de tramos descobertos

A ocorrência de tramos que não são cobertos por serviços regulares requer a adição de horas extras, que são pagas com acréscimo de 50%. Deste modo, ao se minimizar este não-cobrimto, são geradas soluções que reduzem os custos operacionais.

(ii) minimização de tempos ociosos

Os tempos em que uma viatura fica parada entre a operação de duas viagens consecutivas gera diversos problemas. O primeiro deles é o uso ineficiente das viaturas, que consistem em recursos onerosos. Quando uma viatura possui um tempo ocioso superior a duas horas, esta é recolhida para a garagem (movimento de *pull-out*). Assim, percebe-se que o aumento dos tempos ociosos também corrobora para o incremento das viagens em vazio, as quais incorrem em diversos problemas, tais como ineficiência operacional e problemas de congestionamento viário.

Outro problema que os tempos ociosos acarretam é a ineficiência do uso das tripulações, visto que, como os *changeovers* são proibidos, veículos e tripulações formam um só recurso. Durante a jornada de trabalho, obrigatoriamente ocorre uma parada (não-remunerada) para refeição. Os demais tempos em que as viaturas ficam ociosas incorrem em tempos que os tripulantes também não executam actividades, mesmo sendo remunerados.

A outra questão referente aos tempos ociosos é que jornadas com paragens longas, ou seja, entre uma e duas horas, geram reclamações entre as tripulações. Portanto, os tempos ociosos das viaturas também influenciam a qualidade dos serviços gerados, sob a óptica dos tripulantes.

Diante do exposto, pode-se constatar que os tempos ociosos consistem um critério que influencia incisivamente na eficiência operacional do sistema de transporte em análise.

É pertinente ressaltar aspectos práticos relacionados com outro critério relevante para o VCSP, que é o número de viaturas. Para os planeadores da ETUFOR, o número de viaturas não é uma variável, ou seja, não se aceitam soluções com um número de viaturas superior ao mínimo, mesmo que acarretem em algum *trade-off* para a qualidade dos serviços gerados.

Resultados computacionais

Com base nos dois objectivos elencados anteriormente, adaptou-se a heurística PESA-II reportada no Capítulo 5 da presente tese para os objectivos em foco, bem como para a variante do MCP proposta na 6.2.2 do presente capítulo. É importante salientar que o algoritmo proposto trabalha apenas com soluções que respeitem as restrições expostas na secção anterior.

Foram escolhidas arbitrariamente três instâncias a serem testadas, com vista a avaliar os benefícios da abordagem multiobjectivo em relação à prática corrente. Para a compreensão dos resultados reportados, deve-se recordar que $|Q|$ representa a cardinalidade do conjunto de soluções não-dominadas, t representa o tempo de execução do algoritmo e HV denota o hipervolume compreendido pelo conjunto de soluções não-dominadas obtido em cada execução do PESA-II.

Os parâmetros utilizados no algoritmo PESA-II são apresentados abaixo. Tais parâmetros foram ajustados empiricamente após alguns testes computacionais, permitindo a análise do comportamento dos algoritmos.

- Tamanho da população (*maxpop*): 100
- Número de gerações (*maxgen*): 10000
- Probabilidade de mutação (p_{mut}): 100%
- Espaçamento da grade (*ngrid*): 25

Tendo em vista que a metaheurística PESA-II é um processo de amostragem aleatória, optou-se por executar o algoritmo 10 vezes, para avaliar o comportamento médio do mesmo, em termos de qualidade de solução obtida e de tempo de processamento. Os testes foram realizados num processador Genuine Intel 1.86 GHz com 1GB de memória RAM.

Instância 905

A seguir, na Tabela 29, são apresentados os resultados dos testes computacionais para a instância 905, em 10 execuções do PESA-II.

Tabela 29: Experimentos computacionais – instância 905.

Instância 905	10 execuções do PESA-II										Média
	1	2	3	4	5	6	7	8	9	10	
$ Q $	17	9	15	18	13	15	19	12	17	17	15,2
$t(s)$	100,1	104,5	102,8	104,7	104,3	102,1	105,1	96,3	99,1	102,4	102,1
HV	148776	145814	151593	152129	152540	149063	155748	145056	152068	152218	150501

Dentre as 10 execuções do PESA-II para a instância 905, a sétima execução incorreu em um maior hipervolume, e, portanto, será considerada a solução de maior qualidade. A seguir, na Figura 16, é ilustrada a aproximação da frente de Pareto obtida na sétima execução do PESA-II, pois esta incorreu em um maior hipervolume. Com base na Figura 16, pode-se observar que a solução manual da ETUFOR é dominada por todas as soluções geradas pelo algoritmo proposto.

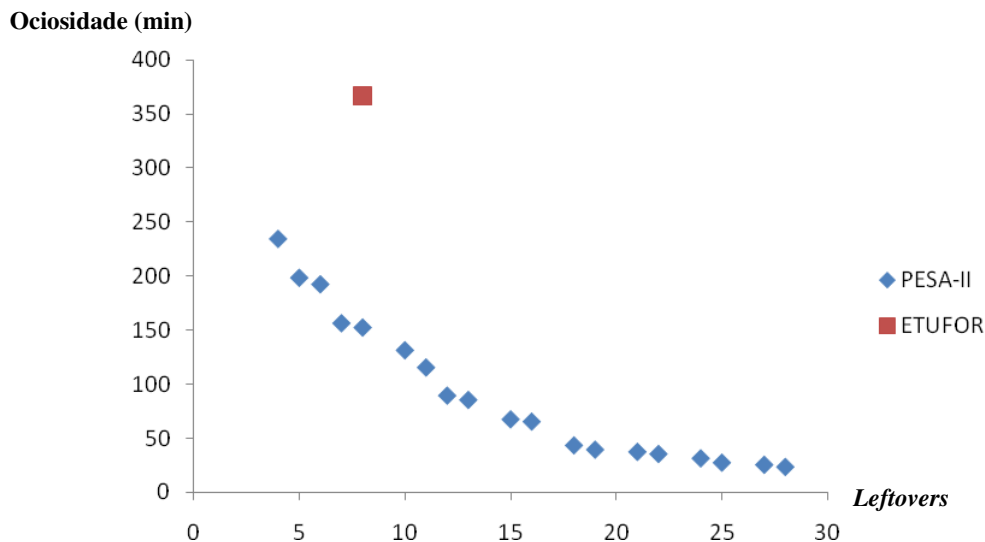


Figura 16: Aproximação da Fronteira de Pareto – instância 905.

Tendo em vista que soluções que tenham muitos *leftovers* não são desejáveis na prática, optou-se por descartar as soluções que tivessem mais *leftovers* do que a solução implementada. A seguir, na Tabela 30, são apresentadas a solução gerada manualmente pelos planeadores da ETUFOR e um conjunto de 5 soluções não-dominadas geradas

pelo algoritmo PESA-II. A solução ND5 possui a mesma quantidade de tramos descobertos constante na solução da ETUFOR, mas possui um tempo ocioso que corresponde a 42% do tempo ocioso da solução da empresa supracitada.

Tabela 30: Soluções para a instância 905.

Solução	Tramos descobertos	Tempos ociosos (min)
ETUFOR	8	367
ND1	4	235
ND2	5	199
ND3	6	193
ND4	7	157
ND5	8	153

Instância 122

A seguir, na Tabela 31, são apresentados os resultados dos testes computacionais para a instância 122, em 10 execuções do PESA-II.

Tabela 31: Experimentos computacionais – instância 122.

Instância 122	10 execuções do PESA-II										Média
	1	2	3	4	5	6	7	8	9	10	
$ Q $	21	20	20	18	18	25	24	19	19	24	20,8
$t(s)$	277,2	275,9	281,7	265,4	239,2	262,8	249,7	245,4	277,5	232,1	260,7
HV	190434	189644	189412	183209	174032	196520	183668	183722	188619	166091	184535

Dentre as 10 execuções do PESA-II para a instância 122, a sexta execução incorreu em um maior hipervolume, e, portanto, será considerada a solução de maior qualidade. A seguir, na Figura 17, é ilustrada a aproximação da frente de Pareto obtida na sexta execução do PESA-II, pois esta incorreu em um maior hipervolume. Com base na Figura 17, pode-se observar que a solução manual da ETUFOR é dominada por todas as soluções geradas pelo algoritmo proposto.

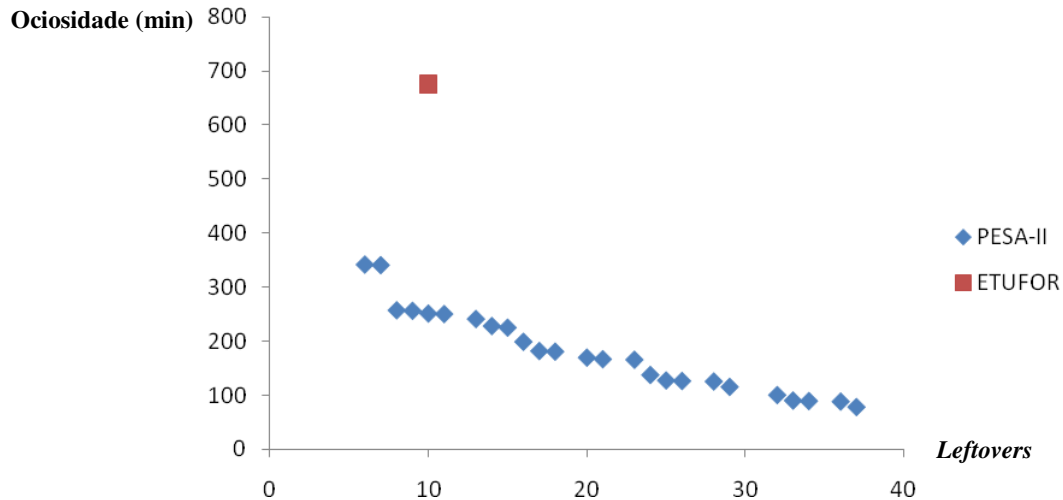


Figura 17: Aproximação da Fronteira de Pareto – instância 122.

Tendo em vista que soluções que tenham muitos *leftovers* não são desejáveis na prática, optou-se por descartar as soluções que tivessem mais *leftovers* do que a solução implementada. A seguir, na Tabela 32, são apresentadas a solução gerada manualmente pelos planeadores da ETUFOR e um conjunto de 5 soluções não dominadas geradas pelo algoritmo PESA-II. A solução ND5 possui a mesma quantidade de tramos descobertos constante na solução da ETUFOR, mas possui um tempo ocioso que corresponde a 37% do tempo ocioso da solução da empresa supracitada.

Tabela 32: Soluções para a instância 122.

Solução	Tramos descobertos	Tempos ociosos (min)
ETUFOR	10	676
ND1	6	341
ND2	7	340
ND3	8	257
ND4	9	256
ND5	10	251

Instância 070

A seguir, na Tabela 33, são apresentados os resultados dos experimentos computacionais para a instância 070, em 10 execuções do PESA-II.

Tabela 33: Experimentos computacionais – instância 070.

Instância 070	10 execuções do PESA-II										Média
	1	2	3	4	5	6	7	8	9	10	
$ Q $	8	12	11	10	9	5	10	11	9	11	10
$t(s)$	329,7	339,7	324,2	371,9	350,2	318,7	305,4	318,6	334,0	326,7	331,9
HV	707863	726740	743178	725934	704636	698300	669251	700386	706808	696232	707933

Dentre as 10 execuções do PESA-II para a instância 070, a terceira execução incorreu em um maior hipervolume, e, portanto, será considerada a solução de maior qualidade. A seguir, na Figura 18, é ilustrada a aproximação da frente de Pareto obtida na sexta execução do PESA-II, pois esta incorreu em um maior hipervolume. Com base na Figura 18, pode-se observar que a solução manual da ETUFOR é dominada por todas as soluções geradas pelo algoritmo proposto.

Tal como nas instâncias anteriores, optou-se por descartar as soluções que tivessem mais *leftovers* do que a solução implementada. Na Tabela 34, é apresentada a solução gerada manualmente pelos planeadores da ETUFOR, assim como um conjunto de 5 soluções não dominadas geradas pelo algoritmo PESA-II.

As cinco soluções não-dominadas apresentadas apresentam melhor desempenho, em relação à solução manual gerada pela ETUFOR, para os dois critérios em análise.

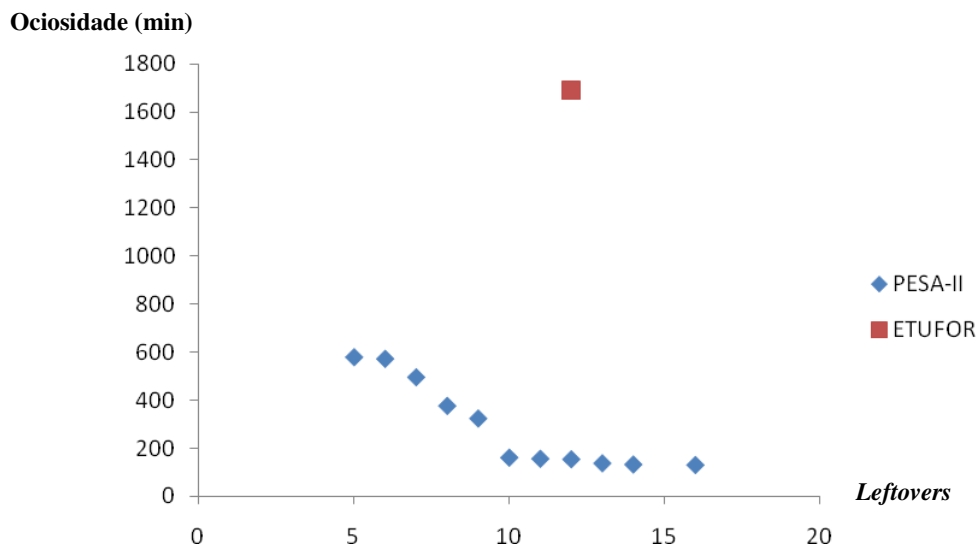


Figura 18: Aproximação da Fronteira de Pareto – instância 070.

Tabela 34: Soluções para a instância 070.

Solução	Tramos descobertos	Tempos ociosos (min)
ETUFOR	12	1691
ND1	5	581
ND2	6	574
ND3	7	497
ND4	8	377
ND5	9	324

Considerações finais

Diante dos resultados obtidos, pode-se verificar que a abordagem multiobjectivo trouxe benefícios em relação à prática do sistema em foco. Para as três instâncias apresentadas, foi possível constatar que as soluções manuais implementadas na prática do sistema foram dominadas por todas as soluções não-dominadas gerados pelo algoritmo proposto.

Dos resultados obtidos, percebe-se que há uma relação de *trade-off* entre o objectivo que procura minimizar o número de tramos descobertos e o que procura minimizar tempos ociosos. As soluções com uma menor quantidade de tramos descobertos incorrem em maiores tempos ociosos.

A abordagem proposta consiste em uma importante ferramenta de apoio à decisão para os planeadores do sistema, visto que podem fornecer soluções de qualidade, com baixo custo computacional e considerando múltiplos critérios, sem a necessidade da determinação de pesos para a ponderação destes.

6.3 Programação de veículos e motoristas em Portimão

6.3.1 O sistema de transporte público de Portimão

O Algarve é uma região situada no sul de Portugal, compreendendo 16 municípios, os quais possuem uma população da ordem de 400 mil habitantes. É uma região fortemente marcada pelo turismo, nomeadamente no período do Verão, quando a população do Algarve praticamente triplica, devido aos visitantes.

De entre os municípios que compõem o Algarve, Portimão é uma cidade com cerca de 50 mil habitantes, cuja economia é fortemente marcada pela pesca e pelo turismo. De modo a proporcionar melhores condições de acesso, seja aos habitantes da

cidade, seja aos turistas, foi implantado no ano de 2003 um sistema de transportes colectivos.

O sistema Vai Vem é um serviço regular de transportes colectivos, da responsabilidade da empresa Frota Azul (Algarve), cuja rede é composta por 14 linhas: azul, verde, amarela, turquesa, violeta, laranja, rosa, encarnada, cinzenta, roxa, ameixa, lima e salmão.

As principais características da programação de veículos e motoristas em Portimão são as seguintes:

- Para a saída e recolha de viaturas, é utilizado um parque da cidade. Assim, o problema pode ser considerado como tendo apenas um depósito.
- Os tempos de viagens não produtivas não pôde ser determinado para todas as situações. Admitiu-se que os tempos de tais viagens seriam de 20 minutos.
- Permite-se que as viaturas aguardem até 15 minutos nos terminos.
- Os serviços podem ser executados em 2 ou 3 etapas, com tempos variando entre uma hora e quatro horas e meia.
- Existe um intervalo obrigatório para refeição, seja almoço ou jantar, os quais ocorrem, respectivamente, entre os intervalos 11:00-14:30 ou 19:30-22:00.
- O intervalo entre as etapas é compreendido entre uma e três horas, correspondendo a um intervalo total de três horas.
- Os *changeovers* são permitidos, no intervalo para a refeição ou no início de uma etapa.
- Os pontos de rendição ocorrem nas paragens Largo do Dique, Alameda e Hospital.
- A jornada de trabalho é de 8 horas diárias, podendo chegar às 14 horas, com a adição de horas extraordinárias.

Na programação de veículos e tripulações em Portimão, as 14 linhas são consideradas conjuntamente, de modo que os recursos, ou seja, viaturas e motoristas podem ser compartilhados entre as linhas. Percebe-se que é permitida a mudança de viatura por linha e a mudança de motorista por viatura (*changeover*).

Ao contrário do que ocorre em Fortaleza (caso de estudo anterior), em que o problema de planeamento operacional dos transportes urbanos é decomposto por linha, o problema de Portimão contempla todas as linhas numa só instância. Assim, a instância

de Portimão tem uma dimensão maior do que as instâncias analisadas para o caso de estudo de Fortaleza.

Deve-se ressaltar que não encontrava disponível a solução implementada na prática para o caso de estudo de Portimão, facto que não permitiu uma comparação entre as soluções geradas pelas abordagens em proposição e a prática corrente.

6.3.2 Modelo mono-objectivo e optimização

Com base na formulação matemática para o VCSP com *changeovers* proposta no Capítulo 4 desta tese, modelou-se o caso de estudo de Portimão por meio de programação linear inteira mista.

De acordo com os argumentos reportados na secção 6.2.2 do presente Capítulo, não foi usada uma abordagem exaustiva *depth-first-search* para a geração dos serviços, sendo estes gerados de forma heurística.

Na heurística de geração de serviços, foram gerados serviços com uma duração máxima de 8 horas, não sendo gerados serviços com horas extraordinárias. Foram gerados serviços com duas ou três etapas, com intervalos para refeição de uma, duas e três horas. Deve-se ressaltar que nos serviços gerados, *changeovers* são permitidos, visto que um motorista pode realizar tarefas em linhas diferentes. As mudanças eram permitidas após o intervalo para refeição, nos pontos de rendição.

As características da instância são as seguintes:

- Número de tramos (m): 401
- Número de serviços (n): 22706
- Número máximo de colunas (d): 65
- Densidade da matriz A (ρ): 0,9%

O modelo matemático teve 88711 variáveis, sendo 22006 delas inteiras (variáveis binárias que representam os serviços). As 65705 variáveis restantes são lineares e dizem respeito ao modelo de fluxo em redes requerido para a programação dos veículos.

O modelo matemático foi executado no *solver* LINGO, em um processador Genuine Intel 1.86 GHz com 1GB de memória RAM.

Apesar da dimensão da instância de Portimão, esta pôde ser solucionada de forma óptima num tempo computacional da mesma magnitude das instâncias de Fortaleza. Deve-se observar que o modelo proposto MCPMR permitiu a obtenção da solução óptima mesmo para uma instância de grande porte.

As características da solução obtida são as seguintes:

- Número de tramos descobertos: 68
- Número de serviços na solução: 65
- Número de viaturas: 38
- Tempo para obtenção da solução: 73s
- Número de nós de B&B: 61
- Número de iterações: 122337
- Percentual de cobertura: 83,0%

A solução óptima obtida consiste numa solução de qualidade em termos práticos, visto que o percentual de cobertura obtido foi elevado, mesmo sem a geração de serviços com horas extraordinárias.

6.3.3 Aplicação da extensão multiobjectivo

Tendo em vista que o VCSP é um problema com forte carácter multiobjectivo, o problema do planeamento operacional do sistema de transportes colectivos em Portimão também foi analisado sob o prisma da optimização multiobjectivo. A seguir, são apresentados os critérios considerados na presente análise.

(i) minimização de tramos descobertos

Conforme descrito na secção 6.2.3, a ocorrência de *leftovers* acarreta em uma maior ineficiência operacional. Portanto, a não-cobertura de tramos deve ser minimizada.

(ii) minimização de tempos ociosos

Conforme descrito na secção 6.2.3, os tempos ociosos apresentam impactos negativos para a operação de sistemas de transportes colectivos, devendo, portanto, ser minimizados.

Como as viagens das 14 linhas são agregadas em uma única tabela de horários, a duração dos tramos é bastante heterogênea. Assim, na geração de serviços, tende-se a ter serviços que cubram diferentes quantidades de tramos.

Neste contexto, percebe-se que gerar soluções de forma aleatória não é a opção mais adequada, visto que o uso de heurísticas construtivas pode incorrer em soluções de maior qualidade.

Logo, a geração da população inicial do PESA-II proposto foi efectuada conforme a heurística construtiva do algoritmo GRASP, reportado no Capítulo 3 da presente tese (vide Quadros 2 e 3). Devido a esta hibridização, usar-se-á a denominação GRASP_PESA-II para a heurística proposta para a instância de Portimão.

Os parâmetros utilizados no algoritmo GRASP_PESA-II são apresentados a seguir. Tais parâmetros foram ajustados empiricamente após alguns testes computacionais, permitindo a análise do comportamento dos algoritmos.

- Parâmetro da fase de construção (α): 100
- Tamanho da população ($maxpop$): 50
- Número de gerações ($maxgen$): 500
- Probabilidade de mutação (p_{mut}): 100%
- Espaçamento da grade ($ngrid$): 25

Conforme reportado no Capítulo 3 da presente tese, a abordagem reactiva para o GRASP suplantou a abordagem com parâmetro α fixo. No entanto, tendo em vista que a construção gulosa aleatorizada ocorre apenas para a geração da população inicial, não seria possível usar a abordagem reactiva. Assim, adoptou-se o parâmetro α fixo.

Devido à dimensão do problema, os tempos computacionais de uma população de 100 indivíduos e de 10000 gerações eram bastante elevados. Assim, optou-se por uma redução destes parâmetros, com vista à obtenção de soluções mais rapidamente.

Devido aos elevados tempos de execução, não foi realizada uma análise estatística do desempenho do algoritmo GRASP_PESA-II para a instância de Portimão. Todavia, com base nos testes apresentados nas seções 5.5 e 6.2.3, o comportamento da heurística proposta, seja em termos de qualidade das soluções ou de tempo de processamento, é estável, não estando usualmente sujeito a *outliers*.

Numa primeira fase, utilizou-se a construção gulosa aleatorizada como mecanismo de geração de uma população inicial. Entretanto, tendo em vista a grande quantidade de tramos a serem cobertos ($m=401$) e o elevado porte da *string* ($n=22706$), o GA apresenta dificuldade em migrar para soluções de elevada qualidade, em termos de cobertura de tramos.

Objectivando melhorar a qualidade da aproximação da fronteira de Pareto para o caso de Portimão buscou-se uma hibridização entre o método heurístico com o método exacto (*Mathematical Programming* – MP). A solução óptima obtida na execução da instância mono-objectivo (vide secção anterior) foi inserida na população inicial. As premissas desta inserção são justificadas a seguir.

A solução óptima obtida via B&B consiste em uma solução de elevada qualidade para o objectivo de minimização do número de tramos descobertos. Logo, esta solução consiste num ponto de uma das extremidades da frente de Pareto. Como as demais soluções geradas por GRASP não têm a mesma qualidade, a solução advinda de B&B estará isolada num hipercubo.

Dadas as premissas do PESA-II, este hipercubo terá uma maior probabilidade de ser seleccionado, e, por conseguinte, esta solução terá uma maior probabilidade de ser utilizada nas operações de recombinação. Deste modo, espera-se que esta solução compartilhe as suas características com a prole futura, incorrendo em uma aproximação mais eficiente para a fronteira de Pareto.

As aproximações para a fronteira de Pareto baseadas em GRASP_PESA-II e MP_GRASP_PESA-II serão denotadas respectivamente por F1 e F2.

Na Figura 19, são ilustradas as aproximações F1. Visualmente, pode-se perceber como F2 supera F1, seja em aproximação ou em diversidade. A aproximação F2 foi mais bem distribuída, enquanto a aproximação F1 apresentou uma descontinuidade na região central.

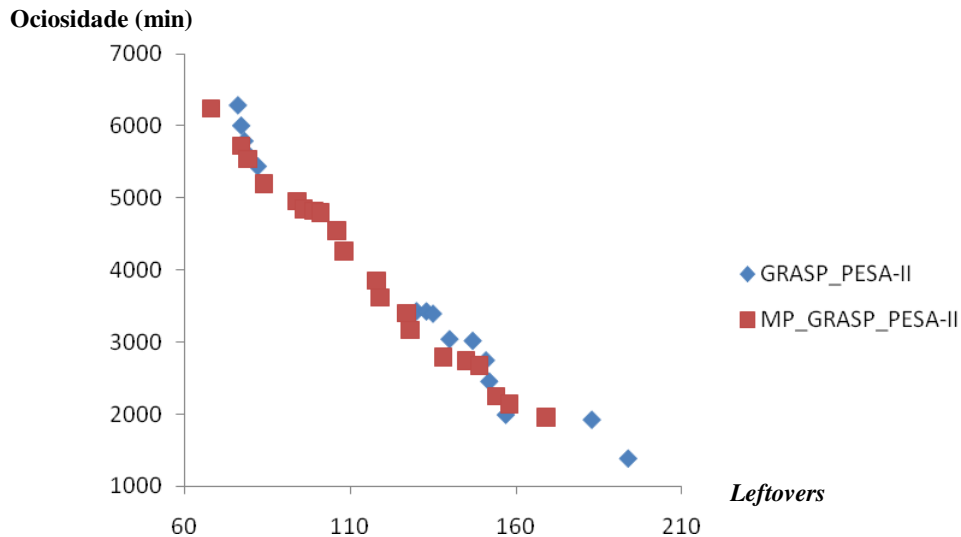


Figura 19: Aproximações obtidas para a fronteira de Pareto no caso em estudo.

Na Tabela 35, são apresentados os resultados, para uma execução, das duas abordagens testadas para o caso de Portimão. Com base nos resultados obtidos, pode-se constatar que F2 apresentou uma maior quantidade de soluções não-dominadas que F1, embora F1 apresente um volume cerca de 2% superior ao de F2.

Duas das soluções não-dominadas de F1 (vide soluções isoladas à direita do gráfico ilustrado na Figura 19) foram de má qualidade e acabaram por aumentar o hipervolume de F1. Deve-se observar que a métrica de avaliação do hipervolume, apesar de ser um indicador da qualidade de aproximações de frentes de Pareto, pode vir a incorrer em análises inadequadas.

Tabela 35: Comparação entre GRASP_PESA-II e GRASP_PESA-II_B&B.

Instância Portimão	F1	F2
$ Q $	15	20
$t(s)$	5852	5738
HV	2512710	2474946

A seguir, na Tabela 36, são apresentadas 5 soluções não-dominadas geradas pela abordagem MP_GRASP_PESA-II. Tendo em vista que soluções que tenham muitos *leftovers* não são desejáveis na prática, optou-se por apresentar apenas estas 5 soluções. Diante destes dados, pode-se constatar que, no caso de Portimão, soluções que apresentam uma menor quantidade de *leftovers* tendem a apresentar maiores tempos ociosos e vice-versa.

Tabela 36: Soluções não-dominadas para a instância de Portimão.

Solução	Tramos descobertos	Tempos ociosos (min)
ND1	68	6244
ND2	77	5726
ND3	79	5543
ND4	84	5194
ND5	94	4956

Dado o porte da instância ($m=401$ e $n=22006$), o Algoritmo Genético apresentou grande dificuldade em obter soluções para o problema. Manusear as estruturas de dados, gerar a população inicial, calcular os valores das funções objectivo, corrigir a viabilidade de uma solução e actualizar a frente de Pareto foram operações que consumiram elevado tempo de processamento.

Uma solução viável para o problema em foco consiste num vector bastante esparso, no qual podiam ser escolhidos até 65 serviços de uma gama de 22006 serviços. Os operadores genéticos podem ter dificuldade em guiar a busca para regiões promissoras do espaço de pesquisa. Nesse contexto, o uso da construção de soluções baseada em GRASP foi de grande relevância para o êxito da abordagem proposta.

Sobre os benefícios da hibridização da metaheurística multiobjectivo com o B&B, devem ser salientados os seguintes aspectos. Devido ao porte da instância, foi necessário correr o algoritmo por apenas 50 gerações, de modo a obter uma solução em tempo de processamento hábil.

Se o número de gerações fosse maior, a solução óptima obtida via B&B teria uma maior probabilidade de ser seleccionada e recombinada com outras soluções, direccionando o processo de pesquisa. Em experimentos realizados em uma instância de menor porte, verificou-se que, para um número de gerações maior, a abordagem MP_GRASP_PESA-II apresentou resultados bem superiores a GRASP_PESA-II.

6.4 Considerações finais

Em geral, sob o prisma da optimização multiobjectivo, parece claro que a importância do VSP reside muito mais na sua interacção com o CSP do que na sua optimização propriamente dita, ou seja, o VSP desempenha um papel secundário com relação ao CSP.

O VSP é um problema relativamente simples, se comparado com o CSP. Obter a quantidade mínima de viaturas a cobrir um conjunto de viagens, para o caso de um único depósito, consiste em um problema trivial. Dificilmente haverá algum *trade-off* relacionado com a quantidade de viaturas, visto que os planeadores provavelmente refutariam uma solução com um número de viaturas superior ao mínimo.

A importância de serem considerados critérios do VSP no VCSP reside nos impactos sobre a viabilidade das soluções do CSP. Para o autor, este é o principal motivo ao estudar o problema integrado.

No que concerne ao caso de estudo de Fortaleza, pode-se salientar que as abordagens baseadas em programação matemática e em optimização multiobjectivo usualmente incorreram em soluções de melhor qualidade do que as implementadas na prática corrente, sob o prisma dos critérios analisados.

O automatismo na obtenção de soluções também é um grande benefício das abordagens propostas, visto que, em Fortaleza, apesar do porte do sistema de transportes colectivos desta cidade, este ainda é planeado manualmente, exigindo um esforço desproporcionado e que pode ser significativamente reduzido pela adopção de um procedimento automático.

No que se refere ao caso de estudo de Portimão, tendo em vista que não havia dados acerca das soluções manuais do sistema de transporte público em análise, não foi possível efectuar uma análise comparativa tal como se procedeu no caso de estudo anterior. De qualquer modo, os resultados obtidos se mostraram metodologicamente consistentes, além do benefício do automatismo, também relatado para o caso de estudo anterior.

Em conclusão, pode-se afirmar que uma abordagem multi-objectivo para o VCSP, tal como a desenvolvida neste trabalho, deverá permitir um planeamento de melhor qualidade, assim como mais flexível, no sentido de fornecer ao decisor a possibilidade de escolher soluções entre um conjunto de alternativas, sem a necessidade de arbitrar pesos para a ponderação dos critérios em análise.

A seguir, no Capítulo 7, serão apresentadas as principais conclusões da presente investigação, bem como suas limitações e sugestões para futuros estudos.

Capítulo 7

Conclusões

“If I have seen a little further it is by standing on the shoulders of Giants.”

Sir Isaac Newton

Resumo: A presente tese versou sobre novas abordagens para a programação de veículos e tripulações. Neste contexto, o capítulo final do trabalho reporta, em sua primeira secção, os principais resultados obtidos na investigação; na segunda secção, as limitações do estudo; e, por fim, na terceira secção, as sugestões para futuras investigações.

7.1 Principais contribuições da tese

O planeamento operacional de sistemas de transporte público é de grande relevância para a eficácia e a eficiência da operação de transportes colectivos. As programações de veículos e de tripulações são fundamentais, quer para a qualidade do serviço prestado, quer para a redução dos custos operacionais. A integração destas duas programações vem sendo reportada pela literatura como de grande valia para um melhor planeamento.

Apesar de diversas investigações sobre o VCSP terem sido desenvolvidas nos últimos 20 anos, ainda existem diversos aspectos a serem melhor elucidados. Neste sentido, esta tese apresenta as seguintes principais contribuições:

- Desenvolveu-se uma abordagem multiobjectivo, baseada numa meta-heurística evolucionária, para a resolução do VCSP. Uma heurística PESA-II foi

projectada e implementada, de forma modular, para a resolução do problema supracitado. Assim, a abordagem pode ser aplicada em diversos contextos, com algumas adaptações.

- Desenvolveram-se novas formulações matemáticas para o VCSP, seja no caso em que os *changeovers* são proibidos, seja quando estes permitidos. As formulações propostas permitiram a obtenção de soluções óptimas, mesmo para instâncias de médio e grande porte, facto que não era contemplado pelas formulações tradicionais presentes na literatura.
- Constatou-se que permitir a ocorrência de *leftovers* tem forte impacto na complexidade da resolução do VCSP. Ao se permitir a ocorrência de *leftovers*, o custo computacional na resolução do VCSP decresce drasticamente, embora a interacção entre o VSP e o CSP se torne mais tênue.
- Constatou-se que, no âmbito do VCSP, a importância do VSP consiste na sua interacção com o CSP, de modo a garantir a viabilidade das soluções obtidas.
- Foram desenvolvidas heurísticas para a resolução do VCSP com e sem *changeovers*, para o caso mono-objectivo. As heurísticas desenvolvidas conseguiram ser competitivas, tanto em qualidade como em performance computacional, com o B&B.
- Foram identificados os principais objectivos relacionados ao VCSP e as inter-relações entre eles.
- As heurísticas desenvolvidas foram implementadas no ambiente MATLAB, podendo ser aplicadas em problemas reais, de modo a apoiar a tomada de decisão de planeadores de empresas de transportes colectivos.
- As abordagens desenvolvidas foram testadas em diversos conjuntos de dados, advindos de problemas reais.
- Nos casos em que foi possível realizar uma comparação entre as soluções geradas e as soluções implementadas na prática, constatou-se que a abordagens propostas apresentaram soluções de melhor qualidade.

7.2 Limitações do trabalho realizado

Com relação aos objectivos da investigação, estes foram atingidos, conforme pode ser observado ao longo desta tese. Deve-se, entretanto, fazer algumas ressalvas:

- Não foi estudada a variante do VCSP para o caso em existem múltiplos depósitos. Contudo, deve-se observar que Haghani *et al.* (2003) demonstram, com base diversos experimentos computacionais, que um modelo considerando um único depósito se comportou melhor do que os modelos analisados que consideram múltiplos depósitos. Groot e Huisam (2008), ao estudarem instâncias do VCSP com múltiplos depósitos, constataram que o efeito da divisão das instâncias não apresenta grande influência na qualidade das soluções obtidas. Assim, mesmo que as abordagens desenvolvidas para o VCSP no presente trabalho contemplem apenas um depósito, elas podem ser adaptadas para o caso com múltiplos depósitos, gerando soluções de qualidade.
- Apesar das heurísticas desenvolvidas terem sido implementadas no ambiente MATLAB e poderem ser utilizadas como ferramenta de apoio à tomada de decisão, não foi desenvolvida uma interface que permitisse uma maior interação com o utilizador. Deve-se observar que o ambiente MATLAB permite tanto a concepção de interfaces gráficas quanto a geração de um arquivo executável, o qual pode ser utilizado em qualquer computador
- No que se refere aos casos de estudo verificou-se que num deles as soluções implementadas não estavam disponíveis. Assim, uma comparação entre as soluções reais e as soluções obtidas pelas abordagens desenvolvidas não foi realizada.

7.3 Sugestões para trabalhos futuros

O tema explorado neste trabalho é bastante amplo, pois muitos desdobramentos podem surgir tanto na tentativa de generalizar os resultados obtidos como na utilização das especificidades dos problemas reais. Deste modo, são reportadas sugestões para aprofundamento do tema investigado, bem como de outros temas afins.

- O problema de geração das tabelas de horário possui forte relação tanto com o VSP quanto com o CSP. Deste modo, é importante que se aprofundem os estudos sobre a integração destes três problemas.
- O problema de escalamento de tripulantes (*rostering*) possui forte impacto na escolha por serviços de tripulações. Logo, é relevante que se aprofundem os estudos sobre a integração do escalamento dos tripulantes e o VCSP.

- O VSP possui uma forte relação com problemas tais como o *Vehicle Routing Problem* e o *Job Shop Scheduling Problem*. Da mesma forma, o MCP relaciona-se com diversos problemas, tais como problemas de *scheduling*, localização de facilidades, definição *layout*, dentre outros. Portanto, as abordagens desenvolvidas, nomeadamente no campo multiobjectivo, podem ser adequadas e avaliadas para outros problemas de otimização combinatória.
- Em diversos contextos, a programação de viaturas e de tripulantes possui uma componente aleatória, visto que, de acordo com o contexto operacional, podem ocorrer alterações nas programações realizadas *a priori*. Nessa conjuntura, se enquadram os problemas de *scheduling* dinâmico (HUISMAN e WAGELMANS, 2006), *rescheduling* (POTTHOFF *et al.*, 2010) e *robust scheduling* (BURKE *et al.*, 2010). O estudo do carácter dinâmico do planeamento operacional de sistemas de transportes públicos, principalmente no que concerne ao prisma da otimização multiobjectivo, deve ser aprofundado. Deve-se observar que situações tais como *changeovers* e mudanças de linha certamente possuem impacto nas abordagens citadas acima.
- Na área de Logística, a integração do VSP e do CSP com o *Vehicle Routing Problem* (HOLLIS *et al.*, 2006) vem sendo estudada. Deve ser investigado como a abordagem proposta nesta tese pode ser estendida para esta nova classe de problemas.
- No que se refere à resolução do SPP, bem como da variante do MCP com restrições de igualdade, a qual se assemelha bastante ao SPP, os métodos exactos apresentam-se como mais adequados do que heurísticas. Devem ser avaliados os benefícios da hibridização de heurísticas e métodos exactos para a resolução destes tipos de problemas.
- O desenvolvimento de interface para um melhor uso das abordagens propostas, bem como a integração dos algoritmos com outros SAD's, também é recomendada.

Referências bibliográficas

ARAKAKI, R. G. I.; LORENA, L. A. N. A constructive Genetic Algorithm for the Maximal Covering Location Problem. In: 4th Metaheuristics International Conference, Porto, 2006.

ARROYO, J. E. C. **Heurísticas e metaheurísticas para otimização combinatorial multiobjetivo**. 2002. Tese de Doutorado (Doutorado em Engenharia Elétrica). Universidade Estadual de Campinas – UNICAMP, Campinas.

ATZINGEN, J.; CUNHA, C. B.; SILVA, G. P. Uma abordagem para o problema de programação de veículos e tripulantes de ônibus. In: XXI Congresso de Ensino e Pesquisa em Transportes, Rio de Janeiro, 2007.

AZEVEDO FILHO, M. A. N.; KWAN, R. S. K.; WREN, A. A alocação de ônibus e motoristas no Brasil: alguma experiência prática. In: VII Congresso Brasileiro de Pesquisa e Ensino em Transportes, Recife, 1994.

BÄCK, T.; SCHUTZ, M.; KHURI, S. A comparative study of a penalty functions, a repair heuristic, and stochastic operators with the set covering problem. In: Proceedings of the European Conference on Artificial Evolution, Springer, 1995.

BAITA, F.; PESENTI, R.; UKOVICH, W.; FAVARETTO, D. A comparison of different solution approaches to the vehicle scheduling problem in a practical case. **Computers & Operations Research**, vol. 27, p. 1249–1269, 2000.

BALL, M.; BODIN, L.; DIAL, R. A matching based heuristic for scheduling mass transit crews and vehicles. **Transportation Science**, vol. 17, p. 4–31, 1983.

BARTODZIEJ, P.; DERIGS, U.; MALCHEREK, D. VOGEL, U. Models and algorithms for solving combined vehicle and crew scheduling problems with rest constraints: an application to road feeder service planning in air cargo transportation. **OR Spectrum**, 2007.

BEASLEY, J.E. OR-Library: Distributing Test Problems by Electronic Mail, **Journal of the Operational Research Society**, vol. 41, p.1069–1072, 1990.

BEASLEY, J. E.; CHU, P. C. **A Genetic Algorithm for the set partitioning problem**. Technical report, Imperial College, The Management School, London, 1995.

BEASLEY, J. E.; CHU, P. C. A Genetic Algorithm for the set covering problem. **European Journal of Operations Research**, vol. 94, 392 – 404, 1996.

BEASLEY, J. E.; CHU, P. C. Constraint handling in genetic algorithms: the set partitioning problem. **Journal of Heuristics**, vol. 11, p. 323–357, 1998.

BERMAN, O.; DREZNER, Z.; WESOLOWSKY, G. O. The Maximal Covering Problem with some negative weights. **Geographical Analysis**, vol. 41, p. 30–42, 2009.

BERTOSSI, A.A.; CARRARESI, P.; GALLO, G. On Some matching problems arising in vehicle scheduling models. **Network**, vol. 17, p. 271–281, 1987.

BLAIS, J. Y.; LAMONT, J.; ROSSEAU, M. The HASTUS vehicle and manpower scheduling system at the Société de transport de la Communauté urbaine de Montréal. **Interfaces**, vol. 20, p. 26–42, 1990.

BLUM, C.; ROLI, A. Metaheuristics in Combinatorial Optimization: Overview and conceptual comparison. **ACM Computing Surveys**, vol. 35, p. 268–308, 2003.

BODIN, L.; GOLDEN, B.; ASSAD, A.; BALL, M. Routing and Scheduling of Vehicles and Crews - The State of the Art. **Computers and Operations Research**, vol. 10, p. 63–212, 1983.

BORNDÖRFER, R.; LÖBEL, A.; WEIDER, S. A bundle method for integrated multi-depot vehicle and duty scheduling in public transit. **Technical Report**, Zentrum für Informationstechnik Berlin (ZIB), Berlin, 2004.

BUNTE, S.; KLIOWER, N. An overview on vehicle scheduling models. In: Proceedings of the 10th International Conference on Computer-Aided Scheduling of Public Transport – CASPT, Leeds, 2006.

BURKE, E.K.; MAERE, G.; MULDER, J.; PAELINCK, M.; BERGHE, G. V.A multi-objective approach for robust airline scheduling. **Computers and Operations Research**, vol. 37, p. 822–832.

CARRARESI, P.; GALLO, G. Network models for vehicle and crew scheduling models. **Transportation Research Part B**, vol. 37, p. 301–322, 1984.

CEDER, A. Efficient timetabling and vehicle scheduling for public transport. In: Voß, S.; Daduna, J. R. (Eds.) **Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems**, Springer, 2001, vol. 505, p. 37–52.

CEDER, A. Urban transit scheduling: framework, review and examples. **Journal of Urban Planning and Development**, vol. 128, p. 225 – 244, 2002.

CERNY, V. Thermodynamics approach to the traveling salesman problem: an efficient simulation algorithm. **Journal of Optimization Theory**, vol. 45, p. 41–51, 1985.

CHAMBERLAIN, M.P.; WREN, A. Developments and recent experience with BUSMAN and BUSMAN II. In: Desrochers, M.; Rosseuau, J.M. (Eds.) **Computer-Aided Transit Scheduling: Proceedings of the Fifth International Workshop**, Springer, 1992, p. 1–15.

COELLO, C. A. C. A short tutorial on evolutionary multiobjective optimization. In: Proceedings of First International Conference on Evolutionary Multi-Criterion Optimization, Zurich, 2001.

COHON, J. L. **Multiobjective programming and planning**. New York: Academic Press, 1978.

COLEY, D. A. **An introduction to Genetic Algorithms for scientists and engineers**. Singapore: World Scientific Publishing, 2010.

CORNE, D. W.; KNOWLES, J. D.; OATES, M. J. The Pareto Envelope-based selection algorithm for multiobjective optimization. In: Proceedings of sixth International Conference on parallel problem solving from Nature, Paris, 2000.

CORNE, D. W.; JERRAM, N. R.; KNOWLES, J. D.; OATES, M. J. PESA-II: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO – 2001), San Francisco, 2001.

CURTIN, K. M.; QIU, F.; HAYSLETT-MCCALL, K.; BRAY, T. M. **Integrating GIS and Maximal Covering Models to Determine Optimal Police Patrol Areas**. USA: University of Dallas, 2005.

CZYZAK, P.; JASZKIEWICZ, A. Pareto Simulated Annealing – a metaheuristic technique for multiple-objective combinatorial optimization problems. **Journal of Multi-Criteria Decision Analysis**; vol. 7: 34–47, 1998.

DADUNA, J. R.; PAIXÃO, J. M. P. Vehicle Scheduling for public mass transit – an Overview. In: Daduna, J. R.; Branco, I.; Paixão, J. M. P. (Eds.) **Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems, Springer**, vol. 430, p. 76–90, 1995.

DANTZIG, G. B.; THAPA, M. N. **Linear programming: 2: Theory and extensions**. Nova Iorque: Springer-Verlag, 2003.

DEB, K. **Multi-objective optimization using evolutionary algorithms**. Chichester: John Wiley & Sons, 2001.

DEB, K.; AGRAWAL, S.; PRATAP, A.; MEYARIVAN, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceedings of sixth International Conference on parallel problem solving from Nature, Paris, 2000.

DIAS, M. T. G. **Aplicação de algoritmos genéticos ao problema da geração de serviços de tripulações**. 1995. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores). Faculdade de Engenharia, Universidade do Porto, Porto.

DIAS, T.G.; FERREIRA, J. V.; FALCÃO E CUNHA, J. Evaluating a DSS for Operational Planning in Public Transport Systems: Ten Years of Experience with the GIST System. In: Voß, S.; Daduna, J. R. (Eds.) **Computer-aided Transit Scheduling**,

Lecture Notes in Economics and Mathematical Systems, Springer, 2001, vol. 505, p. 167–180.

DIAS, M. T. G.; SOUSA, J. M. P.; CUNHA, J. F. Genetic algorithms for the bus driver scheduling problem: a case study. **Journal of the Operational Research Society**, vol. 53, p. 1–12, 2002.

DIAS, M. T. G. **A new approach to the bus driver scheduling problem using multiobjective genetic algorithms**. 2005. Ph.D. Thesis (Doctorate in Engineering Sciences). Faculty of Engineering, University of Porto, Porto.

DOWNS, B. T.; CAMM, J. D. An exact algorithm for the Maximal Covering Problem. **Naval Research Logistics**, vol. 43, p. 435–461, 1996.

DUECK, G. **New Optimization Heuristics: the Great Deluge Algorithm and the Record-to-Record Travel**. *Journal of Computational Physics*, vol. 104, p. 86–92.1993.
EREMEEV, A. V. A Genetic Algorithm with a non-binary representation for the set covering problem. In: *Proceedings of Operations Research '98*, Zurich, 1999.

FALKNER, J. C.; RYAN, D. M. EXPRESS: Set partitioning for bus crew scheduling in Christchurch. In: Desrochers, M.; Rosseau, J. M. (Eds.) **Computer-Aided Transit Scheduling: Proceedings of the Fifth International Workshop**, Springer, 1992, p. 359–378.

FEO, T.A.; RESENDE, M.G.C. A probabilistic heuristic for a computationally difficult set covering problem. **Operations Research Letters**, vol. 8, p.67–71, 1989.

FEO, T.A.; RESENDE, M.G.C. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, vol. 6, p.109–133, 1995.

FISCHETTI, M.; LODI, A.; MARTELLO, S.; TOTH, P. A polyhedral approach to simplified crew scheduling and vehicle scheduling problems. **Management Science**, vol. 47, n. 6, p. 833–850, 2001.

FLEURENT, C.; ROSSEAU, J.M. Integrated vehicle and crew scheduling in practice. In: Technical Report GIRO, Montreal, 2007.

FONSECA, C.M.; FLEMING, P.J. Multiobjective genetic algorithms. In: IEEE colloquium on Genetic Algorithms for Control Systems Engineering, London, 1993.

FRELING, R.; PAIXÃO, J.M.P.; WAGELMANS, A.P.M. Models and algorithms for vehicle scheduling. In: Report 9562/A, Econometric Institute, Erasmus University Rotterdam, 1999a.

FRELING, R.; WAGELMANS, A.P.M.; PAIXÃO, J.M.P. An overview of models and techniques for integrating vehicle and crew scheduling. In: Wilson, N. H. M. (Ed.) **Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems**, Springer, vol. 471, p. 441–460, Springer, 1999b.

FRELING, R.; PAIXÃO, J. M. P.; WAGELMANS, A. P. M. Models and algorithms for single-depot vehicle scheduling. **Transportation Science**, vol. 35, 165 – 180, 2001a.

FRELING, R.; HUISMAN, D.; WAGELMANS, A.P.M. Applying an integrated approach to vehicle and crew scheduling in practice. In: Voß, S.; Daduna, J. R. (Eds.) **Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems**, Springer, vol. 505, p. 73–90, 2001b.

FRELING, R.; HUISMAN, D.; WAGELMANS, A.P.M. Models and algorithms for integration of vehicle and crew scheduling. **Journal of Scheduling**, vol. 6, p. 63–85, 2003.

FRIBERG, C. HAASE, K. An exact branch and cut algorithm for the vehicle and crew scheduling problem. In: Wilson, N. H. M. (Ed.) **Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems**, Springer, vol. 471, p. 63–80, Springer, 1999.

GAFFI, A.; NONATO, M. An integrated approach to ex-urban crew and vehicle scheduling problem. In: Wilson, N. H. M. (Ed.) **Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems**, Springer, vol. 471, p. 103–128, Springer, 1999.

GALVÃO, R. D.; ReVELLE, C. A Lagrangean heuristic for the maximal covering location problem. **European Journal of Operational Research**, vol. 88, p. 114–123, 1996.

GALVÃO, R. D.; ESPEJO, L. G. A.; BOFFEY, B. A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem. **European Journal of Operational Research**, vol. 124, p. 377–389, 2000.

GEN, M.; CHENG, R. **Genetic Algorithms and Engineering design**. England: John Wiley & Sons, 1997.

GLOVER, F.; LAGUNA, M.; MARTI, R. Fundamentals of scatter search and path relinking. **Control and Cybernetics**, vol. 39, p. 653–684, 2000.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear – modelos e algoritmos**. Rio de Janeiro: Elsevier, 2005.

GOLDBERG, D. E. **Genetic Algorithms in search, optimization and machine learning**. Reading: Addison Wesley, 1989.

GOMES, F.R.A.; PRATA, B.A.; AZEVEDO FILHO, M. A. N. Desenvolvimento de software para alocação de veículos e tripulações no sistema de transporte público em Fortaleza. In: XXI Congresso Brasileiro de Pesquisa e Ensino em Transportes, Fortaleza, 2008.

GROOT, S. W.; HUISMAN, D. (2008) Vehicle and crew scheduling: solving large real-world instances with an integrated approach. In: Hickman, M.; Mirchandani; Voß, S.

(Eds.) **Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems**, Springer, vol. 600, p. 43 – 56, Springer.

HAASE, K.; DESAULNIERS, G.; DESROSIERS, J. Simultaneous vehicle and crew scheduling in urban mass transit systems. **Transportation Science**, vol. 35, p. 286–303, 2001.

HAGHANI, A.; BANIHASHEMI, M.; CHIANG, K.H. A comparative analysis of bus transit vehicle scheduling models. **Transportation Research Part B**, vol. 37, 301 – 322, 2003.

HAJELA, P; LIN, C.Y. Genetic search strategies in multicriterion optimal design. **Struct Optimization**, vol. 4, p.99–107, 1992.

HART, J.P.; SHOGAN, A.W. Semi-greedy heuristics: an empirical study. **Operations Research Letters**, vol. 6, p.107–114, 1987.

HOFFMAN, K. L.; PADBERG, M. Solving Airline Crew Scheduling Problems by Branch and Cut. **Management Science**, vol. 39, p.657–682, 1993.

HOLLAND, J. H. **Adaptation in natural artificial systems**. Michigan: University of Michigan Press, 1975.

HOLLIS, B.L.; FORBES, M.A.; DOUGLAS, B.E. Vehicle routing and crew scheduling for metropolitan mail distribution at Australia Post. **European Journal of Operational Research**, vol. 173, p. 133–150, 2006.

HUISMAN, D. **Integrated and dynamic vehicle and crew scheduling**. 2004. Ph.D. Thesis. Tinbergen Institute, Erasmus University Rotterdam, Rotterdam.

HUISMAN, D.; FRELING, R.; WAGELMANS, A.P.M. Multiple-depot integrated vehicle and crew scheduling. **Transportation Science**, vol. 39, p. 491–5025, 2005.

HUISMAN, D.; WAGELMANS, A. P. M. A solution approach for dynamic vehicle and crew scheduling. **European Journal of Operational Research**, vol. 172, p. 453 – 471, 2006.

KIRKPATRICK, S.; GELATT JR., C. D.; VECHI, M. P. Optimization by simulated annealing. **Science**, vol. 220, p. 671–680, 1983.

KLABJAN, D., JOHNSON, E. L., NEMHAUSER, G. L., GELMAN, E., RAMANASWAMY, S. Solving large airline crew scheduling problems: random pairing generation and strong branching. **Computational Optimization and Applications**, vol. 20, p. 73–91, 2001.

KLIEWER, N.; MELLOULI, T.; SUHL, L. A Time-Space Network Based Exact Optimization Model for Multi-Depot Bus Scheduling. **European Journal of Operational Research**, vol. 175, p. 1616–1627, 2006.

KOHL, N.; KARISCH, S. E. Airline Crew Rostering: Problem Types, Modeling and Optimization. **Annals of Operations Research**, vol. 127, p. 223–257, 2004.

KONAK, A.; COIT, D. W.; SMITH, A. E. Multiobjective optimization using genetic algorithms: a tutorial. **Reliability Engineering and System Safety**, vol. 91, p. 992–1007, 2006.

KRON, L.; ABBINK, E.; VROMANS, M.; FISCHETTI, M. Reinventing Crew Scheduling at Netherlands Railways. In: Technical Report, Turni, Padova, 2004.

LAURENT, B.; HAO, J.K. Simultaneous vehicle and driver scheduling: a case study in a limousine rental company. **Computers & Industrial Engineering**, vol. 53, p. 542–558, 2007.

LAURENT, B.; HAO, J. K. Simultaneous vehicle and crew scheduling for extra urban transports. In: XXI International Conference on Industrial Engineering & Other Applications of Applied Intelligent Systems, 2008, Wroclaw.

LESSARD, R.; ROSSEUAU, J.M.; DUPUIS, D. Hastus I: a mathematical programming approach to the bus driver scheduling problem. In: Computer Scheduling of Public Transport: Urban Passenger and Vehicle and Crew Scheduling, Amsterdam, 1981.

LEVINE, D. **Application A parallel genetic algorithm for the set partitioning problem**. 1994. Ph.D. Thesis (Doctorate in Computer Science). Illinois Institute of Technology.

LEVINE, D. Application of a hybrid genetic algorithm to airline crew scheduling. **Computers and Operations Research**, vol. 23, p. 547–558, 1996.

LINDEN, R. **Algoritmos Genéticos – uma importante ferramenta da inteligência computacional**. Rio de Janeiro: Brasport, 2006.

LOURENÇO, H. R.; J. P. PAIXÃO; PORTUGAL, R. Metaheuristics for the bus-driver scheduling problem. In: Economic Working Papers Series, n. 304, Universitat Pompeu Fabra, 1998.

LOURENÇO, H. R.; J. P. PAIXÃO; PORTUGAL, R. Multiobjective metaheuristics for the bus-driver scheduling problem. **Transportation Science**, vol. 35, p. 331–342, 2000.

MANHEIM, M.L. **Fundamentals of Transport System Analysis – Vol. 1: Basic Concepts**. Boston: MIT Press, 1980.

MARTELLO, S.; TOTH, P. A heuristic approach to the bus driver scheduling problem. **European Journal of the Operational Research Society**, vol. 24, p. 106–117, 1986.

MESQUITA, M.; PAIAS, A. Set partitioning/covering-based approach for the integrated vehicle and crew scheduling problem. **Computers & Operations Research**, vol. 35, p. 1562–1575, 2008.

MLADENOVIC, N.; HANSEN, P. Variable neighborhood Search. **Computers and Operations Research**, vol. 24, p. 1097–1100, 1997.

PARK, T.; RYU, K. R. Crew pairing optimization by a genetic algorithm with unexpressed genes. **Journal of Intelligent Manufacturing**; vol. 17, p. 375–383, 2006.

PATRIKALAKIS, G.; XEROKOSTAS, D. Experimentation with a new decomposition scheme of the urban public transport scheduling. In: Desrochers, M.; Rosseau, J. M. (Eds.) **Computer-Aided Transit Scheduling: Proceedings of the Fifth International Workshop**, Springer, 1992, p. 407–425.

PEPIN, A.S.; DESAULNIERS, G.; HERTZ, A., HUISMAN, D. Comparison of heuristic approaches for the multiple depot vehicle scheduling problem. In: Report EI2006-34, Econometric Institute, Erasmus University Rotterdam, 2006.

POTTHOFF, D.; D. HUISMAN, D.; G. DESAULNIERS, G. Column Generation with Dynamic Duty Selection for Railway Crew Rescheduling. **Transportation Science**, vol. 44, p. 493–505, 2010.

PRAIS, M.; RIBEIRO, C.C. Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment. **INFORMS Journal on Computing**, vol. 12, p. 164–176, 2000.

PRATA, B. A.; SOUSA, J. M. P.; DIAS, M. T. G. A binary particle swarm optimization algorithm for a variant of the maximum covering problem In: 10th European Meeting on Metaheuristics, Porto, 2009.

REEVES, C.R.; BEASLEY, J. E. Introduction. In: Reeves, C.R. (Ed.) **Modern heuristic techniques for combinatorial problems**, London: McGraw-Hill, p. 1–14, 1995.

RESENDE, M. G. C. Computing approximate solutions of the maximum covering problem with GRASP. **Journal of Heuristics**, vol. 4: 161–177, 1998.

RESENDE, M. G. C.; RIBEIRO, C. C. Greedy Randomized Adaptive Search Procedures. In: Glover, F.; Kochenberger, G. A. (Eds.) **Handbook of metaheuristics**, Kluwer Academic Publishers, p. 219–250, 2003.

RESENDE, M. G. C.; RIBEIRO, C. C. Greedy Randomized Adaptive Search Procedures: advances and applications. In: Gendreau, M.; Potvin, J. Y. (Eds.) **Handbook of metaheuristics**, Kluwer Academic Publishers, p. 219–250, 2008.

RODRIGUES, M. K.; SOUZA, C.C.; MOURA, A.V. Vehicle and crew scheduling for urban bus lines. **European Journal of Operational Research**, vol. 170, p. 844–862, 2006.

SCHAFFER, J.D. Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the international conference on genetic algorithm and their applications, Pittsburgh, 1985.

SILVA, G. P.; REIS, J. V. A.; SOUZA, M. J. F. Resolução Integrada do Problema de Programação de Veículos e Tripulações no Sistema de Transporte Público. In: Congresso Luso-brasileiro para o Planejamento Urbano, Regional, Integrado e Sustentável, Braga, 2006.

SMITH, B.M., IMPACS - a bus crew scheduling system using linear programming, **Mathematical Programming**, vol. 42, p. 181-187, 1988.

SOUSA, J. P.; FALCÃO E CUNHA, J. GUIMARÃES, R.; PAIXÃO, J. M. P. GIST – Um Sistema de Apoio à Decisão para o Planejamento Operacional de Transportes Colectivos. In: **Casos de Aplicação da Investigação Operacional**, McGraw-Hill, 2000, p. 109 –130.

SOUZA, M. J. F. Programação integrada de veículos e tripulações de ônibus urbano. In: Relatório Técnico, Universidade Federal de Ouro Preto, 2010.

SRINIVAS, N.; DEB, K. Multi-Objective function optimization using non-dominated sorting genetic algorithms. **Evolutionary Computation**, vol. 2, p. 221–248, 1995.

STEIZEN, I. Topics in integrated vehicle and crew scheduling in public transport. 2007. Dr. Rer. Pol. Thesis. Universität Paderborn, Paderborn.

STEIZEN, I.; BECKER, M.; SUHL, L. A hybrid evolutionary algorithm for the vehicle and crew scheduling problem in public transit. In: 2007 IEEE Congress on Evolutionary Computation – CEC 2007, Singapore, 2007.

VALOUXIS, C.; HOUSOS, E. Combined bus and driver scheduling. **Computers and Operations Research**, vol. 170, p. 843–862, 2002.

VIANA, A., SOUSA, J.P., MATOS, M. Using GRASP to solve the Unit Commitment Problem. **Annals of Operations Research**, vol. 120, p. 117–132, 2003.

WEIDER, S. **Integration of vehicle and duty scheduling in public transport**. 2007. Der. Rer. nat. Thesis. Technischen Universität Berlin, Berlin.

WREN, A.; ROSSEAU, J. M. Bus Driver Scheduling – an Overview. In: Daduna, J. R.; Branco, I.; Paixão, J. M. P. (Eds.) **Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems**, Springer, vol. 430, p. 173–183, 1995.

WREN, A.; GUALDA, N. D. F. Integrated Scheduling of Buses and Drivers. In: Wilson, N. H. M. (Ed.) **Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems**, Springer, vol. 471, p. 155–176, Springer, 1999.

WREN, A. Scheduling Vehicles and their Drivers-Forty Years' Experience. In: Technical. Report 2004.03, University of Leeds, School of Computing, 2004.

WREN, A., WREN, D. A Genetic Algorithm for public transport driver scheduling. **Computers and Operations Research**, vol. 22, p. 101–110, 1995.

ZITZLER, E.; THIELE, L. Multiobjective evolutionary algorithms - a comparative case study and the strength pareto approach. **IEEE Transactions on Evolutionary Computation**, vol. 3, p. 257–271, 1999.

ZITZLER, E.; LAUMANN, M.; THIELE, L. SPEA 2: improving the Strength Pareto Evolutionary Algorithm. In: Technical Report, Swiss Federal Institute Technology, Zurich, 2001.