

CLAUDIO BARBIERI DA CUNHA

**CONTRIBUIÇÃO À MODELAGEM DE PROBLEMAS
EM LOGÍSTICA E TRANSPORTES**

Texto de sistematização crítica de parte da obra do candidato, apresentado à Escola Politécnica da Universidade de São Paulo para o Concurso de Livre Docência na especialidade “Logística e Sistemas de Transporte” do Departamento de Engenharia de Transportes.

São Paulo
2006

FICHA CATALOGRÁFICA

Cunha, Claudio Barbieri da

Contribuição à modelagem de problemas em logística e transportes: texto de sistematização crítica de parte da obra do candidato / C.B. Cunha. -- São Paulo, 2006.

315 p.

Tese (Livre-Docência) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Transportes.

**1.Logística (Administração de materiais) 2.Transportes
3.Modelagem matemática I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Transportes II. t.**

Dedico este trabalho
à minha esposa Cristina,
a meus filhos Tiago e Natalia,
à minha mãe Cléa,
e, em memória, a meu pai Clóvis.

AGRADECIMENTOS

Não há como deixar de destacar e agradecer a todos que, ao longo dos anos, direta ou indiretamente contribuíram para que este texto, e a carreira acadêmica que o mesmo representa, se concretizassem.

Ao Prof. Nicolau D. Fares Gualda, orientador de mestrado e de doutorado, pela orientação e incentivo ao longo de quase dez anos; pelo apoio que proporcionou a possibilidade de eu adentrar e caminhar na carreira acadêmica na EPUSP; pelo companheirismo e amizade nesses mais de 8 anos juntos no LPT – Laboratório de Planejamento e Operação de Transportes.

Aos meus colegas docentes e aos funcionários do Departamento de Engenharia de Transportes, aos meus colegas do Programa de Mestrado em Engenharia de Sistemas Logísticos, Profs. Hugo T. Y. Yoshizaki, Miguel César Santoro, Marco A. Brinati e Rui Carlos Botter, e demais colegas docentes e funcionários da Escola Politécnica da Universidade de São Paulo, pelo respeito, pelo apoio e incentivo constantes, e pelo ambiente estimulante de trabalho que me proporcionaram desde o início da minha carreira.

Ao Eng^o Wagner Colombini Martins, pela oportunidade de ingressar no IPT – Instituto de Pesquisas Tecnológicas do Estado de São Paulo, ponto inicial da minha carreira científica como pesquisador, pelo constante incentivo, e pela sua sempre amizade.

Ao Prof. Joffre Swait, responsável pela minha iniciação nesse campo fascinante e desafiante que envolve o desenvolvimento de algoritmos e heurísticas para solução de problemas em logística e transportes, ao qual me dedico até hoje com muito prazer, emoção e satisfação; pela sua orientação em grande parte da minha pesquisa de mestrado; pela acolhida atenciosa e calorosa em Gainesville, Flórida, por ocasião do meu estágio de pós-doutoramento, sozinho, sem família, fazendo-me sentir em casa, e um pouco parte da sua família durante toda a minha estadia.

Meu especial agradecimento e gratidão ao Prof. Ravindra K. Ahuja, do *Department of Industrial and Systems Engineering*, da Universidade da Flórida, com quem tive o privilégio de trabalhar por ocasião do meu pós-doutoramento, entre agosto de 2003 e março de 2004. Foi uma experiência absolutamente gratificante e inestimável, tanto do ponto de vista científico, como também pessoal. Suas qualidades humanas, éticas, morais, científicas e

espirituais continuam a desempenhar o papel de farol e exemplo, no qual humildemente me espelho.

Aos muitos colegas docentes das comunidades de Engenharia de Transportes e de Pesquisa Operacional em todo o Brasil, com quem tive a oportunidade de passar momentos agradáveis, e de sempre aprender em bancas e em eventos científicos no Brasil e nos mais diferentes e exóticos locais do exterior. Seria impraticável nomear todos individualmente, além do risco de acabar esquecendo alguém.

A todas as pessoas com quem tive o privilégio de conviver, tanto nas relações acadêmicas e profissionais, como no contato pessoal, com quem sempre aprendi muito e descobri oportunidades interessantes e desafiadoras.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), à CAPES (Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior) e à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), por terem possibilitado o desenvolvimento de muitos dos trabalhos aqui apresentados, através da concessão de bolsas e auxílios.

À minha amada família, minha esposa Cristina, meus filhos Tiago e Natália, minha mãe Cléa e meu saudoso pai Clóvis, incansáveis incentivadores e admiradores da minha trajetória acadêmica, pela paciência, compreensão e apoio em todas as oportunidades que se viram privados da minha companhia, convívio e presença ao longo da minha carreira, em especial naqueles longos 8 meses em que nos separamos em função do meu pós-doutoramento.

Finalmente, gostaria de ressaltar o meu mais aprofundo agradecimento e reconhecimento aos meus alunos e orientados de Mestrado, Doutorado e Iniciação Científica, que pela sua dedicação e empenho conduziram pesquisas que me possibilitaram não só redigir este trabalho, como também e principalmente produzir artigos que me fazem sentir orgulhoso.

Enfim, a minha gratidão a todos que, de maneira direta ou indireta, contribuíram para que este trabalho se tornasse realidade.

RESUMO

CUNHA, CLAUDIO BARBIERI DA. CONTRIBUIÇÃO À MODELAGEM DE PROBLEMAS EM LOGÍSTICA E TRANSPORTES. 2006.

O presente trabalho sintetiza parte da pesquisa realizada e orientada pelo autor e relacionada à modelagem matemática de problemas em logística e transportes, como parte dos requisitos de um Concurso de Livre-Docência na especialidade *Logística e Sistemas de Transporte* do Departamento de Engenharia de Transportes da Escola Politécnica da Universidade de São Paulo.

Pretende-se ainda contribuir para a complexa tarefa de modelar, propor e desenvolver algoritmos e heurísticas para resolver problemas reais em logística e transportes encontrados na prática. Espera-se que este texto possa ser usado como uma referência básica introdutória para um curso em nível de pós-graduação que trate do desenvolvimento de algoritmos e heurísticas nos campos acima, especialmente direcionado para aqueles que não tem um conhecimento prévio na área. Conseqüentemente, atenção especial foi dada ao desenvolvimento de um texto bastante didático; nesse sentido, buscou-se aprimorar e ampliar as descrições dos algoritmos nas quais este texto se baseou.

A maioria dos problemas tratados são modelados matematicamente como problemas de programação linear inteira, de natureza combinatória e, conseqüentemente, extremamente difíceis de resolver. Assim, apresenta-se inicialmente uma revisão sucinta e abrangente das principais estratégias de solução, incluindo os principais aspectos do algoritmo “*branch-and-bound*” e da relaxação Lagrangeana, seguindo-se uma descrição das metaheurísticas mais importantes e comumente utilizadas: algoritmos genéticos, busca tabu, “*simulated annealing*”, GRASP e busca em vizinhança de grande porte.

Em seguida são apresentados os principais conceitos e aspectos da modelagem e solução de alguns problemas relevantes selecionados. Esses problemas são agrupados em três grandes categorias: roteirização e programação de veículos, localização e projeto de redes logísticas, e otimização ferroviária. O objetivo é ilustrar a aplicação das diferentes estratégias de solução, algoritmos exatos e heurísticos, e descrever os principais aspectos da sua implementação e dos resultados obtidos, assim como demonstrar como os mesmos podem ser usados em diferentes contextos para tratar problemas do mundo real.

Adicionalmente, busca-se analisar e identificar potenciais oportunidades e direções de pesquisa.

Palavras-chave: logística, transportes, modelagem matemática, algoritmos, heurísticas.

ABSTRACT

CUNHA, CLAUDIO BARBIERI DA. A CONTRIBUTION TO THE MODELING OF PROBLEMS IN LOGISTICS AND TRANSPORTATION. 2006.

This manuscript summarizes part of the research conducted and advised by the author, and related to mathematical modeling of problems in logistics and transportation, in partial fulfillment of the requirements for the level of “livre-docência” (roughly equivalent to associate professor in the US and other countries) in the field of *Logistics and Transportation Systems* of the Department of Transportation Engineering, Escola Politécnica, University of São Paulo.

It also aims to contribute to the complex task of modeling, devising and developing algorithms and heuristics to solve real-world problems in logistics and transportation found in practice. It is expected that this text can be used as a basic introductory reference in a graduate course related to the development of algorithms and heuristics in the above fields, especially targeted to those who do not have a previous background in the subject. Consequently, special attention was dedicated towards developing a very didactical text; in this sense, the descriptions of the algorithms in which this text was based were substantially improved.

Most of the problems described are modeled as linear integer programming problems, and are of combinatorial nature, and, as a result, extremely hard to solve. Thus, a concise and comprehensive review of the main solution strategies, both exact and heuristics is initially presented, including the main issues in branch-and-bound algorithms and Lagrangian relaxation, followed by the most relevant and widely used metaheuristics: genetic algorithms, tabu search, simulated annealing, GRASP and very large-scale neighborhood search.

We then provide the main concepts and details of some selected relevant problems in logistics and transportation that are efficiently and effectively modeled and solved. These problems are grouped in three major categories: vehicle routing and scheduling, network design, and railroad optimization. The goal is to illustrate the application of the different solution strategies, exact algorithms and heuristics, and to describe the main implementation issues and the results obtained, as well as to show how they can be used in different contexts for addressing real-world problems.

In addition, we aim to analyze and point out potential opportunities and directions for research.

Key words: logistics, transportation, mathematical modeling, algorithms, heuristics.

SUMÁRIO

1 INTRODUÇÃO	18
1.1 OBJETIVOS E ESCOPO	18
1.2 RELEVÂNCIA DO TEMA	21
1.3 ORGANIZAÇÃO DO TEXTO	23
2 MODELAGEM E ALGORITMOS EXATOS PARA PROBLEMAS EM LOGÍSTICA E TRANSPORTES	25
2.1 ALGORITMO “ <i>BRANCH-AND-BOUND</i> ”	26
2.1.1 Formas de Pesquisa do Espaço de Soluções pelo Algoritmo B&B	36
2.1.2 Avanços na Resolução de Problemas de Programação Inteira	41
2.1.3 Formulações Matemáticas mais Robustas	44
2.2 RELAXAÇÃO LAGRANGEANA	47
2.2.1 Solução do Problema Relaxado	49
2.2.2 O Problema de Caminho Mínimo Restrito	51
2.2.3 O Problema das Múltiplas Mochilas Binárias com K Restrições (K-MKP)	57
2.2.4 Relaxação Lagrangeana Aplicada ao Problema de Roteirização de Veículos	60
2.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO	64
3 ESTRATÉGIAS DE SOLUÇÃO HEURÍSTICAS	66
3.1 ALGORITMOS GENÉTICOS	69
3.1.1 Codificação de Indivíduos	72
3.1.2 Geração da População Inicial	73
3.1.3 Função de Avaliação da Aptidão	73
3.1.4 Seleção de Indivíduos para Reprodução	74
3.1.5 Operador Genético de Cruzamento	76
3.1.6 Operador Genético de Mutação	81
3.1.7 Atualização da População	83
3.1.8 Parâmetros para o Algoritmo Genético	85
3.1.9 Novos avanços	86
3.2 BUSCA TABU	87
3.2.1 Uso de Memória	89
3.2.2 Intensificação e Diversificação	90
3.2.3 Memória de Curto Prazo	91

3.2.4	Memória de Longo Prazo.....	94
3.2.5	Exploração da vizinhança.....	95
3.2.6	Estrutura Geral de Funcionamento da Busca Tabu.....	97
3.2.7	Outros Aspectos.....	98
3.3	GRASP.....	101
3.3.1	Construção da Lista Restrita de Candidatos (LRC).....	103
3.4	<i>SIMULATED ANNEALING</i>	105
3.5	BUSCA EM VIZINHANÇA DE GRANDE PORTE.....	107
3.5.1	Busca Local: Uma Visão Geral.....	108
3.5.2	Problemas de Particionamento.....	109
3.5.3	Vizinhanças Definidas como Ciclos.....	111
3.5.4	Grafo de Melhoria.....	113
3.5.5	Algoritmo de busca em vizinhança de grande porte.....	115
3.5.6	Exemplos de Aplicação.....	117
3.6	OUTRAS METAHEURÍSTICAS.....	120
3.6.1	Colônia de Formigas.....	120
3.6.2	<i>Scatter Search</i>	122
3.7	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	124

4 ROTEIRIZAÇÃO E PROGRAMAÇÃO DE VEÍCULOS 125

4.1	CARACTERIZAÇÃO DO PROBLEMA.....	125
4.1.1	Taxonomia dos Problemas de Roteirização.....	131
4.1.2	Principais Estratégias de Solução.....	132
4.2	O PROBLEMA DE DIMENSIONAMENTO E ROTEIRIZAÇÃO DE UMA FROTA HETEROGÊNEA DE VEÍCULOS.....	135
4.2.1	Heurísticas para Problemas com Frota Heterogênea.....	139
4.2.2	Experimento Computacional.....	140
4.2.3	Considerações Finais.....	142
4.3	ROTEIRIZAÇÃO E PROGRAMAÇÃO DO TRANSPORTE DE PESSOAS COM DEFICIÊNCIAS.....	143
4.3.1	Caracterização do Problema.....	144
4.3.2	Formulação Matemática.....	146
4.3.3	Estratégia de Solução.....	150
4.3.4	Aplicação do Modelo.....	151
4.3.5	Considerações Finais.....	154

4.4	O PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS COM CARGA COMPLETA E JANELAS DE TEMPO.....	155
4.4.1	Formulação Matemática.....	158
4.4.2	Estratégia de Solução Baseada em Busca Tabu.....	162
4.4.3	Experimentos Computacionais.....	165
4.4.4	Considerações Finais.....	168
4.5	O PROBLEMA DE ROTEIRIZAÇÃO DE CAMINHÕES-TANQUE PARA A ENTREGA DE COMBUSTÍVEIS EM POSTOS.....	168
4.5.1	Modelo Matemático.....	170
4.5.2	Heurística de Solução.....	174
4.5.3	Experimentos Computacionais.....	176
4.5.4	Considerações Finais.....	178
4.6	PROGRAMAÇÃO DE VEÍCULOS DE COLETA DE RESÍDUOS PARA GERAÇÃO DE ENERGIA.....	180
4.6.1	Formulação Matemática.....	181
4.6.2	Estratégia de Solução.....	184
4.6.3	Considerações Finais.....	188
4.7	ASPECTOS PRÁTICOS DA APLICAÇÃO DE MODELOS DE ROTEIRIZAÇÃO DE VEÍCULOS A PROBLEMAS REAIS.....	188
4.7.1	A Função Objetivo a ser Otimizada.....	189
4.7.2	Atributos Espaciais para Problemas de Roteirização.....	191
4.7.3	Requisitos de um Software para Roteirização de Veículos.....	194
4.8	CONCLUSÕES DO CAPÍTULO.....	198

5 PROJETO DE REDES LOGÍSTICAS.....200

5.1	CONCEITOS BÁSICOS DE LOCALIZAÇÃO DE INSTALAÇÕES.....	204
5.1.1	Modelos Matemáticos para a Configuração de Redes Logísticas.....	206
5.1.2	Estratégias de Solução para Problemas de Localização.....	214
5.1.3	Modelagem da Localização de Instalações com Múltiplos Critérios.....	216
5.2	CONFIGURAÇÃO DE REDES DO TIPO HUB-AND-SPOKE PARA O TRANSPORTE RODOVIÁRIO DE CARGA PARCELADA.....	221
5.2.1	Problemas de Configuração de Redes “Hub-and-Spoke” na Literatura.....	223
5.2.2	Formulação Matemática.....	226
5.2.3	Estratégia de Solução Proposta.....	227
5.2.4	Experimentos Computacionais.....	229
5.2.5	Estudo de Caso.....	231
5.2.6	Considerações Finais e Continuidade da Pesquisa.....	233

5.3	OTIMIZAÇÃO DA LOGÍSTICA DE PRODUÇÃO E DISTRIBUIÇÃO DE REVISTAS.....	235
5.3.1	Caracterização do Problema.....	237
5.3.2	Modelagem Matemática.....	239
5.3.3	Estruturação do Modelo em Ambiente de Planilha Eletrônica.....	242
5.3.4	Aplicação a um Problema Real.....	247
5.3.5	Considerações Finais.....	250
5.4	CONCLUSÕES DO CAPÍTULO.....	252
6	MODELAGEM DE PROBLEMAS FERROVIÁRIOS.....	255
6.1	RELEVÂNCIA E MOTIVAÇÃO.....	255
6.2	O PROBLEMA DE DESPACHO DE TRENS.....	261
6.2.1	Modelagem Matemática.....	264
6.2.2	Heurística de Solução.....	270
6.2.3	Considerações Finais.....	271
6.3	O PROBLEMA DE PROGRAMAÇÃO DE LOCOMOTIVAS.....	272
6.3.1	Formulação Matemática.....	275
6.3.2	Estratégia de Solução.....	279
6.3.3	Experimentos Computacionais.....	283
6.3.4	Considerações Finais.....	283
6.4	O PROBLEMA DE BLOCAGEM DE VAGÕES.....	284
6.4.1	Formulação Matemática.....	287
6.4.2	Estratégia de Solução.....	289
6.4.3	Experimentos Computacionais.....	292
6.4.4	Considerações Finais.....	293
6.5	CONCLUSÕES DO CAPÍTULO.....	294
7	CONSIDERAÇÕES FINAIS.....	296

LISTA DE FIGURAS

Figura 1.1: Capa do Livreto Distribuído pela INFORMS na campanha “The <i>Science of Better</i> ”.....	22
Figura 1.2: Exemplo de história de sucesso no âmbito da divulgação do <i>Science of Better</i>	23
Figura 2.1: Árvore de Enumeração	27
Figura 2.2: Árvores B&B para o problema de programação binária	29
Figura 2.3: Exemplo de branching em problemas com variáveis inteiras não binárias	32
Figura 2.4: Árvore “branch-and-bound” para o exemplo de programação inteira mista	33
Figura 2.5: Exemplo de árvore de busca.....	37
Figura 2.6: Exemplo da introdução de planos de corte (“cutting planes”).....	39
Figura 2.7: Problema de caminho mínimo com restrição de tempo.....	51
Figura 2.8: Função Lagrangeana $Z_D(\mu)$ para $T = 14$	56
Figura 2.9: Heurística baseada em relaxação Lagrangeana para o K-MKP	59
Figura 3.1: Esquema geral de funcionamento do Algoritmo Genético	71
Figura 3.2: Seleção por Roleta	75
Figura 3.3: Seleção por Ranking Linear.....	76
Figura 3.4: Seleção por Amostragem Universal Estocástica	76
Figura 3.5: Cruzamento em um único ponto	77
Figura 3.6: Cruzamento em múltiplos pontos	77
Figura 3.7: Cruzamento uniforme	78
Figura 3.8: Operador de cruzamento PMX	79
Figura 3.9: Operador de cruzamento OX	80
Figura 3.10: Operador de cruzamento CX	81
Figura 3.11: Mutação por inversão	82

Figura 3.12: Mutação por substituição	82
Figura 3.13: Mutação de troca recíproca.....	83
Figura 3.14: Mutação heurística.....	83
Figura 3.15: Árvore de decisão tabu	93
Figura 3.16: Diferentes regras de ativação tabu	96
Figura 3.17: Estrutura Geral da Busca Tabu	98
Figura 3.18: Oscilação Estratégica.....	99
Figura 3.19: Path Relinking	100
Figura 3.20: Pseudo-código da metaheurística GRASP.....	101
Figura 3.21: Pseudo-código da fase de construção	102
Figura 3.22: Pseudo-código refinado da fase de construção	104
Figura 3.23: Busca em vizinhança	107
Figura 3.24: Problema de particionamento	110
Figura 3.25: Problema de roteirização visto como um problema de particionamento.....	110
Figura 3.26: Vizinhança do tipo 2-troca.....	111
Figura 3.27: Troca cíclica (“cyclic exchange”).....	112
Figura 3.28: Troca de caminho (“path exchange”)	112
Figura 3.29: Arco (i, j) no grafo de melhoria	114
Figura 3.30: Ciclo direcionado no grafo de melhoria	115
Figura 3.31: Algoritmo básico de busca VLSN	116
Figura 3.32: Troca cíclica de duas sub-árvores.....	118
Figura 3.33: Dois possíveis ciclos para o problema das múltiplas mochilas	119
Figura 3.34: Esquema geral de funcionamento do “scatter search”	122

Figura 4.1: Pôster anunciando o desafio proposto pela P&G relacionado a um Problema de Caixeiro Viajante com 33 cidades.....	127
Figura 4.2: Dois “subtours” para o PCV com 6 cidades	129
Figura 4.3: Dados do exemplo de problema de roteirização de frota heterogênea	137
Figura 4.4: Localização dos pontos de atendimento	152
Figura 4.5: Representação do problema	159
Figura 4.6: Movimento de realocação de uma viagem	163
Figura 4.7: Movimento de troca entre viagens.....	163
Figura 4.8: Exemplo de cálculo do valor da aptidão.....	185
Figura 4.9: Estruturação do cromossomo e cálculo do valor da aptidão no Excel.....	186
Figura 5.1: Exemplo de Rede Logística.....	201
Figura 5.2: Estrutura de decisão hierárquica em 3 níveis.....	218
Figura 5.3: Estrutura hierárquica para o problema de localização do centro de distribuição.....	219
Figura 5.4: Exemplo de rede do tipo “hub-and-spoke”	222
Figura 5.5: Rede “hub-and-spoke” da DHL nos EUA Fonte: DHL.....	223
Figura 5.6: Codificação do Algoritmo Genético	229
Figura 5.7: Dados de entrada do modelo matemático referentes aos pontos de demanda j e às unidades industriais i	244
Figura 5.8: Dados de entrada do modelo matemático referentes a custos e transit times.	244
Figura 5.9: Dados de entrada do modelo matemático referentes aos pontos de demanda j e às unidades industriais i	245
Figura 5.10: Parte da planilha do modelo correspondente às variáveis de decisão Z_i e Y_{ijm} e às restrições (5.37), (5.42) e (5.43).....	245
Figura 5.11: Variáveis de decisão X_{ijmt} e restrições (5.38) e (5.40).....	246
Figura 5.12: Função objetivo e restrições (5.39) e (5.41)	246

Figura 6.1: Malha ferroviária dos EUA.....	257
Figura 6.2: Malha ferroviária brasileira e concessionárias.....	258
Figura 6.3: Parte de uma malha ferroviária com trechos de via singela e dupla.....	262
Figura 6.4: Exemplo de gráfico horário	265
Figura 6.5: Parte de uma rede espaço-tempo para um trem	268
Figura 6.6: Parte de uma rede espaço-tempo para uma estação	277
Figura 6.7: Exemplo de rede de blocos	286
Figura 6.8: Algoritmo de busca em vizinhança de grande porte (VLSN) para o problema de blocagem de vagões.....	290
Figura 7.1: Novos problemas teste benchmark de grande porte com 240 e 1200 clientes.....	298

LISTA DE QUADROS

Quadro 3.1: Comparação Algoritmo Genético versus “scatter search”	124
Quadro 4.1: Heurísticas de economia propostas por Golden <i>et al.</i> (1984) para problemas de roteirização com frota heterogênea.....	138
Quadro 4.2: Características das Heurísticas de Melhorias Baseadas em Busca Tabu	164
Quadro 4.3: Requisitos e características de sistemas para roteirização de veículos.....	197
Quadro 5.1: Lista de verificação proposta para seleção de localização	220
Quadro 6.1: Concessionárias Ferroviárias no Brasil e respectivas extensões de malha.....	259

LISTA DE TABELAS

Tabela 2.1: Soluções dos nós da árvore B&B	31
Tabela 2.2: Otimização por subgradiente para o problema de caminho mínimo restrito.....	55
Tabela 2.3: Custos e tempos em todos os caminhos para o problema de caminho mínimo restrito	56
Tabela 4.1: Economias ordenadas em ordem decrescente	137
Tabela 4.2: Solução da heurística de economias e ótima para o problema da Figura 4.3	137
Tabela 4.3: Desvios de custo obtidos para as três heurísticas em relação ao melhor resultado obtido	141
Tabela 4.4: Velocidades médias e faixas de distâncias	152
Tabela 4.5: Indicadores relativos à programação obtida pela heurística de inserção paralela	153
Tabela 4.6: Indicadores relativos à programação obtida após o processo de melhoria.....	154
Tabela 4.7: Resultados para problemas reduzidos	167
Tabela 4.8: Instâncias e Resultados do Processamento no ILOG CPLEX.....	173
Tabela 4.9: Exemplo de Cálculo de Serventia e Facilidade	175
Tabela 4.10: Resumo dos Resultados dos Testes Realizados.....	178
Tabela 5.1: Valores de $\alpha(T)$ de acordo com o fluxo médio diário de carga.....	232
Tabela 5.2: Hubs selecionados e economias para diferentes demandas e custos fixos.....	233
Tabela 5.3: Hubs selecionados e economias para diferentes custos fixos e variáveis.....	233
Tabela 5.4: Resumo das Alternativas Propostas e da Situação Atual	249
Tabela 6.1: Comparação da heurística proposta com as soluções da ferrovia CSX.....	283
Tabela 6.2: Economias obtidas com novos planos de blocos otimizados	292
Tabela 6.3: Economias obtidas com limite de duas trocas de trem por bloco.....	293

Capítulo**1****INTRODUÇÃO****1.1 OBJETIVOS E ESCOPO**

O presente trabalho foi redigido de maneira a constituir um Texto de Sistematização Crítica, como parte dos requisitos de um Concurso de Livre-Docência na especialidade *Logística e Sistemas de Transporte* da área de Planejamento e Operação de Transportes do Departamento de Engenharia de Transportes da Escola Politécnica da Universidade de São Paulo.

Segundo as normas que regem esse concurso, o candidato deve apresentar e defender uma tese original ou um texto que sistematize criticamente a obra do candidato ou parte dela. O que se está apresentando aqui é um trabalho que sintetiza parte dos estudos e pesquisas realizados ou orientados pelo autor, relacionados à modelagem e ao desenvolvimento de algoritmos e heurísticas visando à solução de problemas típicos em logística e transportes, com ênfase para aqueles mais complexos do ponto de vista do esforço computacional para a sua solução. Essas pesquisas vindo sendo diretamente apoiadas pelo CNPq desde 2002, através de uma bolsa de Produtividade em Pesquisa (PQ); pela CAPES, através de uma bolsa de estágio pós-doutoral no exterior entre 2003 e 2004; e também através de outros auxílios-reunião pela FAPESP.

Descrevem-se as principais pesquisas realizadas pelo autor desde a conclusão do seu doutorado em 1997, apresentando, com algum detalhamento, os desenvolvimentos metodológicos e os principais resultados obtidos em cada pesquisa e que contribuíram para o avanço na área de modelagem e desenvolvimento de algoritmos para a solução de problemas em logística e transportes. Embora resultados individuais já tenham sido apresentados em artigos científicos publicados, a sistematização apresentada busca não só demonstrar mais

claramente a evolução do conhecimento acumulado sobre o tema, como também a integração entre as pesquisas, além de apresentar criticamente os conhecimentos produzidos pelo autor e seus orientados. Assim, espera-se que este trabalho proporcione uma visão clara da carreira acadêmica do autor naquela que é a linha-mestra das suas pesquisas.

Pretende-se ainda que o trabalho preencha o objetivo de contribuir efetivamente para a pesquisa, o conhecimento e a aplicação de técnicas de Pesquisa Operacional, especialmente Modelagem Matemática, na solução de problemas típicos de sistemas logísticos e de transportes.

Nesse contexto, uma motivação adicional para o desenvolvimento deste trabalho foi a de produzir um texto que possa ser utilizado como referência básica para a disciplina de pós-graduação PTR-5001 - *Implementação Computacional de Algoritmos e Heurísticas aplicados a Transportes e Logística*, ministrada pelo docente. Conforme consta na ementa dessa disciplina, seu objetivo é “*proporcionar uma visão abrangente de técnicas, métodos e estratégias de solução e de implementação de algoritmos de PO e sua aplicação a problemas de otimização em transportes e logística*”. Assim, são apresentados na disciplina os principais algoritmos e heurísticas de solução, bem como os principais aspectos da sua implementação computacional. Trata-se de uma disciplina “*hands-on*”, em que os alunos devem implementar uma estratégia de solução para algum problema selecionado, incluindo todo o trabalho de codificação em alguma linguagem de programação (C, C++, C#, Pascal/Delphi, Java, etc.).

A parte inicial da disciplina, que engloba toda a parte de estrutura de dados (pilhas, filas, árvores e árvores binárias, *heaps*, conjuntos, grafos, ordenação, *hash tables*, etc.) possui boas referências bibliográficas, em particular o livro de Swait (1991) (que foi o primeiro docente a propor e ministrar essa disciplina entre 1989 e 1990 na EPUSP), além de Horowitz e Sahni (1978) e Cormen *et al.* (2001), entre outros. Por outro lado, a segunda parte da disciplina, que engloba o detalhamento das principais estratégias de solução com ênfase para as metaheurísticas, carece de um texto básico introdutório.

Assim, espera-se que esse texto possa contribuir para sanar essa lacuna, apresentando, de maneira introdutória, os principais algoritmos e metaheurísticas, como também estabelecendo a motivação para o uso de estratégias heurísticas, a partir de uma revisão do funcionamento dos algoritmos exatos, em particular o “*branch-and-bound*”, e sua dificuldade para resolver inúmeros problemas encontrados na prática. Evidentemente, não se tem a pretensão de

substituir os textos encontrados na literatura sobre cada uma das metaheurísticas, mas sim apresentar uma visão global de todas elas e de aspectos da sua implementação computacional, o que deve ser complementado por artigos específicos. Além da parte conceitual, busca-se ilustrar a aplicação dessas estratégias de solução a problemas de logística e transportes que foram objeto das pesquisas realizadas por este autor, seus orientados, colegas e colaboradores.

A disciplina PTR-5001 tem como pré-requisito uma disciplina introdutória de Pesquisa Operacional, ou de Modelagem Matemática, como por exemplo, a disciplina de pós-graduação PTR-5744 – *Pesquisa Operacional Aplicada ao Planejamento de Transportes*, também ministrada por este autor. Deve ficar claro que este texto não tem em nenhuma hipótese a finalidade de substituir algum livro-texto utilizado no âmbito dessa disciplina, embora as diferentes formulações matemáticas apresentadas, não encontradas nesses livros textos clássicos de otimização, possam ilustrar tópicos avançados de formulação matemática de problemas em logística e transportes.

Da mesma forma, este texto pode ser usado como material complementar para uma disciplina avançada em logística e transportes, cobrindo e ilustrando aspectos quantitativos, mais especificamente modelagem matemática, nessas áreas, em um nível de aprofundamento maior que o dos livros-texto disponíveis como, por exemplo, o de Chopra e Meindl (2001), embora menos genérico, uma vez que tópicos relacionados à produção e à gestão de estoques, só para citar alguns, não foram objeto das pesquisas do docente, assim como não o foram outras técnicas de Pesquisa Operacional distintas da modelagem matemática (por exemplo, simulação).

Para finalizar, embora se tenha buscado dar um caráter abrangente, conceitual e quantitativo ao tema, cobrindo diferentes tipos de problemas, e modelos matemáticos, estratégias e algoritmos aplicáveis à sua solução, não houve a pretensão de ser exaustivo em nenhum dos seus aspectos, e muito menos, de esgotar o assunto. Pretendeu-se, na realidade, contribuir para a continuidade e para o aprofundamento de linhas de pesquisa associadas à aplicação de técnicas de Pesquisa Operacional, especialmente Modelagem Matemática, na solução de problemas relevantes em Logística e Transportes.

1.2 RELEVÂNCIA DO TEMA

As mudanças que vêm sendo observadas nos cenários econômico e produtivo mundial, com a globalização e a crescente competição entre empresas por mercados mundiais, acrescentando-se, no caso brasileiro, a estabilização da economia e a redução da inflação (que já completaram uma década), têm proporcionado novos desafios em termos de desempenho e de qualidade de serviços, que afetam o planejamento da operação de sistemas logísticos e de transporte em seus aspectos estratégicos, táticos e operacionais.

São metas a diminuição de custos, o aumento da eficiência econômica e operacional, a confiabilidade e a qualidade dos serviços oferecidos, com foco em decisões e ações operacionais, gerenciais e estratégicas, apoiadas em modernas técnicas de modelagem matemática e informática como forma de assegurar decisões e ações corretas, no curto, médio e longo prazos.

Nesse contexto, surgem inúmeras oportunidades de desenvolvimento e de aplicação de modelos matemáticos a problemas reais em sistemas logísticos e de transportes. Essa é uma área de grande interesse de pesquisa, tendo em vista não só a sua relevância no contexto produtivo atual, como também em função da complexidade matemática de inúmeros tipos de problemas reais, principalmente aqueles de natureza combinatória, em que o esforço computacional para a sua resolução cresce exponencialmente com o tamanho do problema, tornando impossível a obtenção da solução ótima. Pode-se citar, entre outros, problemas de roteirização e programação de veículos, de localização de instalações, de projeto e configuração de redes logísticas e de cadeias de suprimentos, de alocação de veículos e de programação de tripulações, e de otimização ferroviária, vários dos quais serão abordados no presente trabalho.

Por outro lado, do ponto de vista prático, mais especificamente da utilização desse sofisticado ferramental para a solução de problemas reais, cada vez mais as empresas, organizações e dirigentes estão se dando conta da importância da Pesquisa Operacional (PO). Mas o que é Pesquisa Operacional? Pesquisa Operacional, do inglês americano "*Operations Research*", ou britânico "*Operational Research*", também conhecida como "*Management Science*", pode ser definida como a disciplina que busca aplicar métodos analíticos e quantitativos avançados para auxiliar a tomada de decisão, procurando assegurar melhores decisões.

Através da utilização de técnicas como modelagem matemática para analisar situações complexas, a Pesquisa Operacional proporciona aos executivos e às empresas a possibilidade de tomar as melhores decisões e solucionar problemas complexos de maneira efetiva.

Em uma entrevista que foi republicada, *in memoriam*, na edição de junho de 2005 da revista *OR/MS Today* (Nazareth, 2005), quando perguntado sobre sua definição de PO, George B. Dantzig, considerado “o pai da programação linear”, respondeu:

“Eu a chamo de ‘a ciência da tomada de decisão’, o que inclui todas as possibilidades de abstrair um problema e colocá-lo em uma forma matemática. Isto engloba desde as pessoas que se inclinam a ser matemáticos tentando resolver esses problemas abstratos, até as pessoas que têm um problema real e tem que responder ao seu superior. Pesquisa Operacional é tudo isso, e, dependendo de com que você conversar, você terá uma idéia diferente do que ela é.”

Uma coletânea de diferentes definições de PO pode ser encontrada em Nazareth (2005).

O INFORMS - *Institute for Operations Research and Management Science* (www.informs.org) é a maior sociedade profissional no mundo ligada à prática da Pesquisa Operacional. Foi criado em 1995, através da fusão da ORSA – *Operations Research Society of America* e do TIMS – *The Institute for Management Science*.

Em um esforço promovido pelo INFORMS para divulgar e tornar mais conhecida a PO, foi lançada em 2004 uma campanha de marketing em que se “batizou” a Pesquisa Operacional como “*The Science of Better*”, ou *A Ciência do Melhor*, como ilustrado na Figura 1.1.

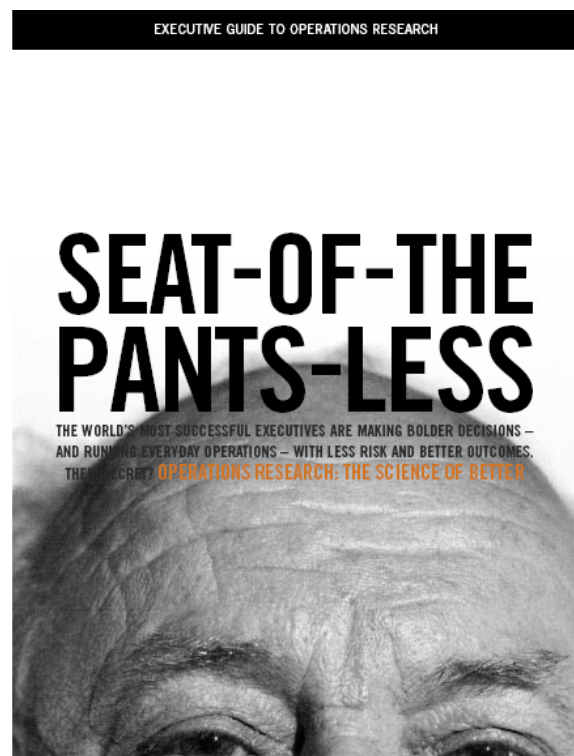


Figura 1.1: Capa do Livreto Distribuído pela INFORMS na campanha “*The Science of Better*”

Fonte: www.ScienceOfBetter.com

Foi inclusive criado um *site* na Internet só para promover e divulgar casos reais de grande sucesso na PO (www.ScienceOfBetter.com), utilizando linguagem para leigos e recursos gráficos, como exemplificado na Figura 1.2.

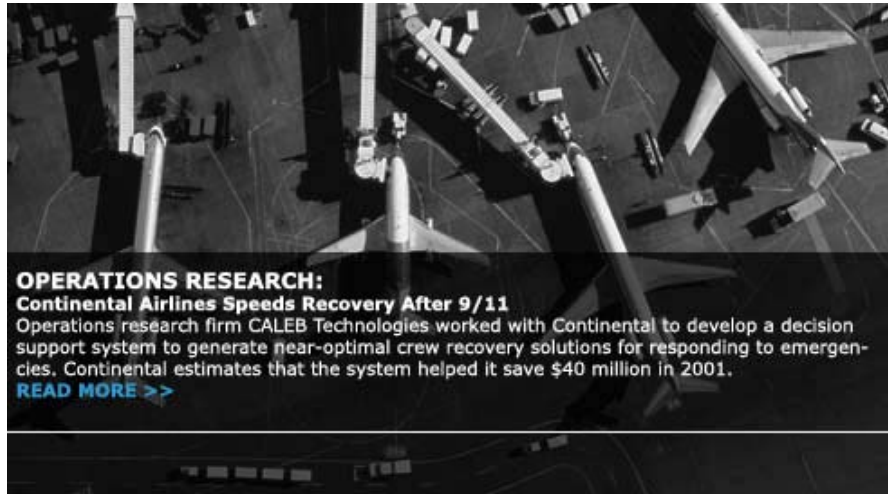


Figura 1.2: Exemplo de história de sucesso no âmbito da divulgação do *Science of Better*

Fonte: INFORMS (www.ScienceOfBetter.com)

A título de curiosidade, a PO é também denominada “*an MIT approach to decision making*” no âmbito do Centro de Pesquisa Operacional (“*Operations Research Center*”) do MIT (<http://web.mit.edu/orc/www/>).

A Pesquisa Operacional abrange tópicos como programação matemática (métodos de otimização), foco deste texto, bem como teoria de filas, simulação, árvores de decisão, entre outros. A programação Matemática é usualmente dividida em programação linear, problemas de fluxo em rede (“*network flow*”), programação inteira e programação não linear.

1.3 ORGANIZAÇÃO DO TEXTO

Este documento está dividido em duas partes principais: os dois primeiros capítulos apresentam os principais conceitos relacionados à modelagem e a resolução de problemas reais encontrados em logística e transportes, enquanto os três capítulos seguintes descrevem as mais importantes pesquisas e aplicações que compõem a pesquisa desenvolvida pelo autor.

Mais especificamente, no Capítulo 2 descrevem-se as principais estratégias exatas para a solução de problemas de programação inteira, com ênfase para o algoritmo “*branch-and-bound*” e a relaxação Lagrangeana, tendo em vista que muitos problemas encontrados nas áreas de logística e transportes são modelados como problemas de programação linear inteira.

No Capítulo 3 são apresentadas, com algum detalhamento, as mais importantes metaheurísticas utilizadas para resolver problemas reais em logística e transportes, muitos dos quais de natureza combinatória, o que inviabiliza o uso de algoritmos exatos. Mais especificamente, são abordadas as seguintes metaheurísticas: algoritmos genéticos, busca tabu, GRASP, “*simulated annealing*” e busca em vizinhança de grande porte.

O Capítulo 4 trata de problemas de roteirização e programação de veículos. Mais especificamente, são apresentados os principais conceitos, as taxonomias e as principais estratégias de solução para problemas de roteirização. Na seqüência, são apresentados diversos problemas que envolvem a roteirização e a programação de veículos: (i) dimensionamento e roteirização de uma frota heterogênea de veículos, (ii) roteirização e programação do transporte de pessoas com deficiências, (iii) roteirização de veículos com carga completa e janelas de tempo, (iv) roteirização de caminhões-tanque para entrega de combustíveis, e (v) programação de veículos de coleta de resíduos para a geração de energia. Estes problemas selecionados, que foram objeto das pesquisas desenvolvidas pelo autor, cobrem vários aspectos, bem como diferentes estratégias de solução.

Já o Capítulo 5 aborda o problema de configuração de redes logísticas, englobando os principais conceitos e modelos de localização, como também duas aplicações importantes, relacionadas à configuração de redes: o problema de configuração de rede para uma transportadora de carga parcelada e o problema de configuração de uma rede de produção e distribuição para uma editora de revistas.

O Capítulo 6 trata da modelagem matemática de problemas de planejamento e operação do transporte ferroviário de carga. Mais especificamente, são considerados os problemas de despacho de trens, de programação de locomotivas, e de blocagem de trens. São detalhadas estratégias de solução que permitem tratar e resolver instâncias de problemas reais encontrados na prática.

Por fim, o Capítulo 7 contém as considerações finais deste trabalho.

Capítulo

2

MODELAGEM E ALGORITMOS EXATOS PARA PROBLEMAS EM LOGÍSTICA E TRANSPORTES

Muitos dos problemas desafiadores que surgem em logística e transportes são de otimização discreta, ou seja, modelados de maneira mais realística como problemas de *programação linear inteira-mista* (“*mixed-integer linear program*”) ou simplesmente (PI), cuja formulação matemática apresenta a seguinte forma geral:

$$\begin{aligned} &\text{minimizar} && c^T x \\ &\text{sujeito a} && Ax = b \\ &&& l \leq x \leq u \\ &&& \text{alguns ou todos } x_j \in x \text{ inteiros} \end{aligned}$$

sendo A uma matriz ($m \times n$) denominada *matriz de restrições*, $x = (x_1, \dots, x_n)$ um vetor das *variáveis de decisão*, c um vetor dos coeficientes das variáveis de decisão na *função objetivo*, e l , u os vetores dos limitantes inferiores e superiores das variáveis de decisão, respectivamente. De outra forma, um problema de programação inteira corresponde a um problema de programação linear (PL) mais restrições adicionais de integralidade de algumas ou todas as variáveis. Se todas as variáveis forem inteiras, o problema é denominado de *programação inteira pura*; se todas as variáveis forem binárias, de *programação inteira binária*.

As restrições de integralidade tornam os problemas de PI difíceis de serem resolvidos, uma vez que a região viável deixa de ser convexa. Em outras palavras, isso significa que pode haver múltiplas regiões viáveis e, conseqüentemente, múltiplos pontos de ótimo locais, o que pode aumentar significativamente o esforço de busca para a obtenção da solução ótima global.

Não existe um algoritmo geral que resolva de maneira eficiente todos os problemas de programação inteira, como acontece na programação linear com o algoritmo Simplex. Adicionalmente, inúmeros problemas são muito difíceis de resolver; segundo Garey e Johnson (1979), a maioria deles possui ordem de complexidade exponencial, ou seja, o esforço computacional para a sua resolução cresce exponencialmente com o tamanho do problema.

O objetivo deste capítulo é apresentar uma visão geral do algoritmo “*branch-and-bound*”, uma das principais estratégias exatas para a solução de problemas de programação inteira, e também da relaxação Lagrangeana. A principal motivação é proporcionar uma visão geral de como os problemas de programação inteira são resolvidos, as dificuldades que tornam impossível a sua aplicação em inúmeros problemas, os principais avanços em termos de desempenho observados, e os recursos para uma modelagem mais efetiva.

2.1 ALGORITMO “*BRANCH-AND-BOUND*”

Também conhecido como busca em árvore, enumeração implícita, ou ainda algoritmo de bifurcação limite, o algoritmo “*Branch-and-Bound*” (B&B) é o método utilizado para a resolução de problemas de programação linear inteira.

Ao contrário de um método de enumeração completa para resolver um problema de otimização discreta, em que todas as possíveis soluções são “listadas” e a melhor escolhida, no B&B varre-se *implicitamente* (daí o nome) todo o espaço de solução, sendo uma grande parte deste simplesmente eliminada sem a necessidade de ser efetivamente investigada.

Note-se que o esforço para a enumeração completa de todo o espaço de solução pode ser absurdamente grande, uma vez que deve considerar todos os valores possíveis de todas as variáveis de decisão. Por exemplo, para um problema com n variáveis binárias, a idéia é iterativamente quebrar o problema em dois, como mostrado na Figura 2.1; na primeira iteração considera-se separadamente os casos em que $x_1 = 1$ e $x_1 = 0$; em seguida, para cada valor de x_1 , a variável x_2 pode assumir os valores 0 e 1, e assim sucessivamente até x_n .

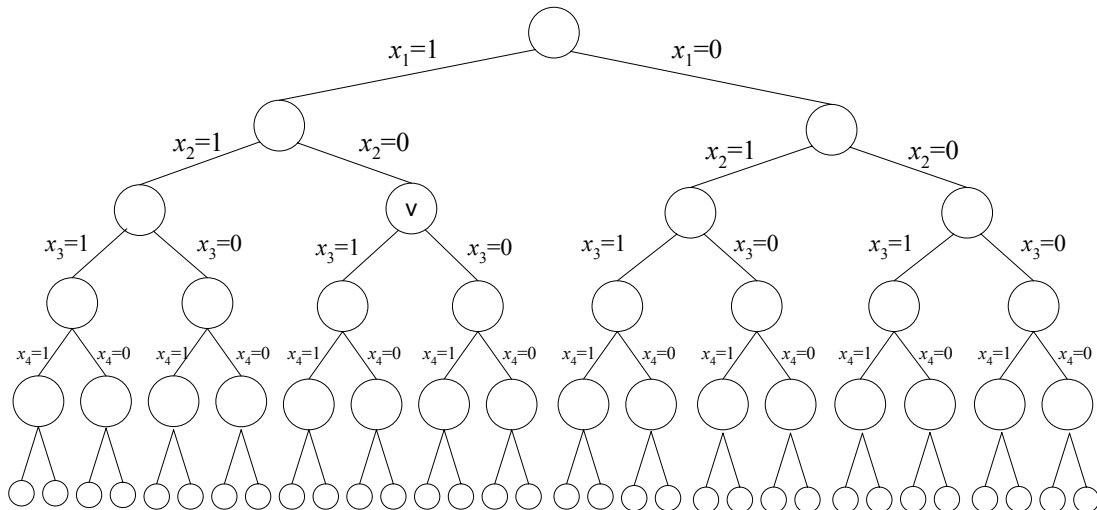


Figura 2.1: Árvore de Enumeração

O esforço de enumeração cresce exponencialmente com o tamanho do problema, já que existem 2^n diferentes soluções. Para $n = 60$, por exemplo, o número de possíveis combinações é $2^{60} \approx 1,1529 \times 10^{18}$. Supondo ser possível avaliar um trilhão de soluções (10^9) por segundo, o tempo total de processamento para este problema de dimensões modestas seria de aproximadamente 3,7 anos, caso todas as soluções tivessem que ser avaliadas!

O conceito básico utilizado pelo método B&B é “dividir para conquistar” pois, a cada nó da árvore gerado, o problema tratado é subdividido em problemas menores, mais fáceis de serem resolvidos. Como visto acima, essa característica é particularmente clara nos casos de problemas de programação inteira binária. Nesses casos, a expansão de um nó da árvore fixa o valor de uma determinada variável, gerando dois subproblemas, um com a referida variável igual a 1, e o segundo com a variável igual a 0, conforme ilustrado acima na Figura 2.1.

Assim, o algoritmo B&B consiste em um procedimento sistemático de busca no espaço de soluções. Esse processo dá-se por meio da montagem de uma árvore de busca, na qual os ramos da árvore representam valores alternativos que podem ser assumidos por cada variável de decisão do problema. Na extremidade de cada ramo localiza-se um nó, que representa uma solução parcial do problema, correspondente ao conjunto de nós percorridos a partir do nó em questão até a raiz da árvore.

Proposto inicialmente por Land e Doig (1960), a estratégia implícita no algoritmo B&B é a de buscar a eliminação de partes do espaço de soluções que certamente não resultarão em nós da solução ótima. Isso é feito através da eliminação dos ramos da árvore que gerem soluções

inviáveis, ou que gerem soluções viáveis, porém dominadas, isto é, soluções com valores superiores ao menor valor da função objetivo de uma solução viável encontrada até aquele momento (no caso de minimização). A determinação de uma função limitante inferior da função objetivo é geralmente obtida resolvendo-se o problema de programação linear (PL) correspondente, em que as restrições de integralidade das variáveis de decisão são relaxadas.

A título de ilustração, considere-se o seguinte problema de programação inteira binária:

$$\text{maximizar} \quad 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6 \quad (2.1)$$

$$\text{sujeito a} \quad 5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14 \quad (2.2)$$

$$0 \leq x_i \leq 1 \text{ e inteiro para } i = 1, \dots, 6 \quad (2.3)$$

Esse problema corresponde ao chamado *problema da mochila binária* (“*binary knapsack problem*”): dados seis objetos, para os quais são conhecidos os respectivos lucros (16, 22, 12, 8, 11, 19) e pesos (5, 7, 4, 3, 4, 6), deve-se selecionar quais objetos devem ser inseridos na mochila de modo que o lucro total seja maximizado (função objetivo 2.1) e o peso total não ultrapassasse a capacidade da mochila (restrição 2.2 que restringe o peso da mochila a 14).

O problema é resolvido da seguinte forma através do algoritmo B&B: inicialmente é resolvida a relaxação linear do problema original (2.1-2.3), obtendo-se a solução $X^1 = (1 \ 0.43 \ 0 \ 0 \ 0 \ 1)$, resultando lucro total de 44.43 e peso total de 14. Essa não é uma solução viável para o problema original, pois $x_2 = 0.43$ não atende à restrição de integralidade das variáveis de decisão. No entanto, o lucro obtido (44.43) é um limitante superior do valor da função objetivo; em outras palavras, a solução do problema com as restrições de integralidade das variáveis de decisão não pode ser maior que esse valor. Essa solução relaxada corresponde ao nó 1 (inicial) da árvore B&B indicada na Figura 2.2a.

Em seguida, fixa-se a variável $x_1 = 0$ e resolve-se a relaxação linear, obtendo-se a seguinte solução $X^2 = (0 \ 1 \ 0.25 \ 0 \ 0 \ 1)$, resultando lucro total de 44 e peso total de 14. Essa solução corresponde ao nó 2 da árvore B&B, conforme também mostrado na Figura 2.2a. Na seqüência gera-se o nó 3, através da adição da restrição $x_1 = 1$, e resolvendo-se a relaxação linear, resultando $X^3 = (1 \ 0.43 \ 0 \ 0 \ 0 \ 1)$, com lucro e peso iguais a 44.43 e 14, respectivamente (Figura 2.2b). Nenhuma das duas soluções geradas (nós 2 e 3) corresponde a uma solução viável para o problema, uma vez que ambas não atendem às restrições de integralidade das variáveis de decisão.

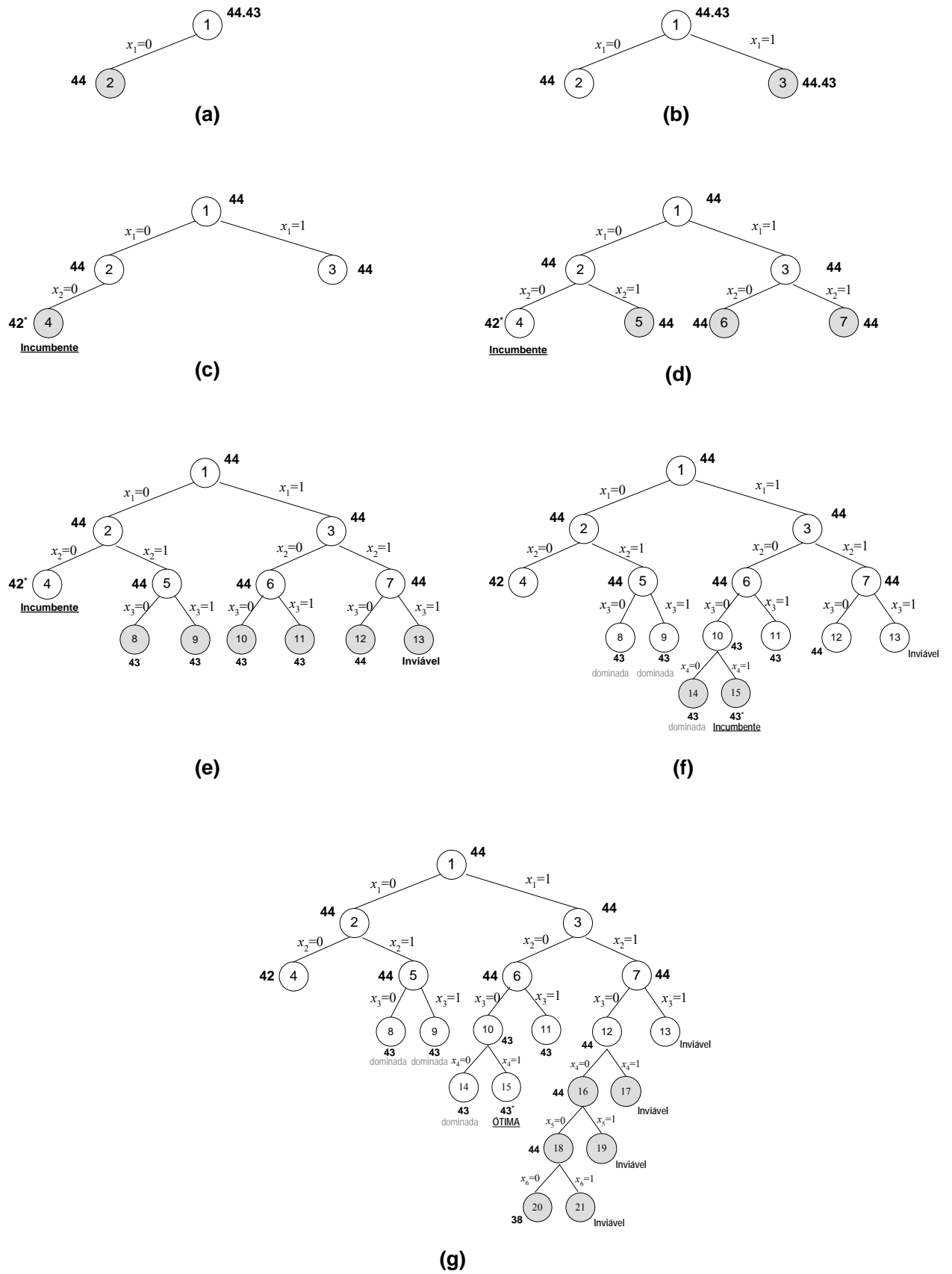


Figura 2.2: Árvores B&B para o problema de programação binária

Note-se que o valor da função objetivo obtida em cada nó da árvore (que corresponde a uma relaxação linear) pode ser truncado para o maior número inteiro inferior ao valor obtido, já que as variáveis de decisão são binárias e os respectivos coeficientes na função objetivo correspondem a valores inteiros. Assim, os valores nos nós 1 e 3 podem ser truncados para 44.

A partir do nó 2, isto é, com $x_1 = 0$, o nó 4 é gerado adicionando-se a restrição $x_2 = 0$ e resolvendo-se a relaxação linear, o que resulta na solução $X^4 = (0 \ 0 \ 1 \ 0 \ 1 \ 1)$, com lucro total de 42 e peso total de 14. Essa é uma solução viável para o problema, pois todas as variáveis resultam inteiras. Portanto, 42 é o melhor valor obtido até o momento para a função objetivo do problema, e $X^4 = (0 \ 0 \ 1 \ 0 \ 1 \ 1)$ é denominada solução incumbente, isto é, a melhor solução viável encontrada até o momento. Não é necessário descer mais a partir do nó 4, pois a solução obtida relaxando-se as restrições de integralidade já resulta inteira.

O melhor valor de lucro obtido até o momento é chamado de limitante inferior, ou “*lower-bound*” ($LB = 42$) e também serve para eliminar partes da árvore B&B: se algum outro nó gerado posteriormente resultar lucro menor que 42, mesmo que seja uma solução não inteira obtida através da relaxação linear, aquele tramo da árvore poderá ser simplesmente descartado, uma vez tratar-se de uma solução dominada pela melhor solução corrente obtida (incumbente), daí o nome de limitante inferior, pois a solução ótima está limitada inferiormente por esse valor.

Na seqüência geram-se os nós 5, 6 e 7, conforme mostrado na Figura 2.2c e cujas soluções são listadas na Tabela 2.1. Note-se que nenhum dos nós corresponde a uma solução inteira; entretanto, nenhum deles pode ser eliminado, pois os respectivos valores de função objetivo não são menores que os da melhor solução inteira obtida. Em outras palavras, se algum dos nós resultasse função objetivo com valor inferior a 42, todo o tramo da árvore abaixo desse nó poderia ser simplesmente desconsiderado.

Na seqüência são gerados os nós 8 a 13 conforme mostrado na Figura 2.2e e cujas soluções respectivas também se encontram na Tabela 2.1. Nenhum deles resulta em solução inteira; entretanto, nenhum deles pode ser eliminado, pois as soluções da relaxação linear apresentam valores superiores à melhor solução obtida até o momento.

Opta-se, então, por expandir a árvore abaixo do nó 10, obtendo-se, uma nova solução inteira no nó 15, $X^{15} = (1 \ 0 \ 0 \ 1 \ 0 \ 1)$, que se torna a incumbente (melhor solução), pois apresenta lucro total de 43, melhor que as soluções anteriores, conforme mostrado na Figura 2.2f.

Tabela 2.1: Soluções dos nós da árvore B&B

<i>objeto i</i>	1	2	3	4	5	6			
<i>lucro l_i</i>	16	22	12	8	11	19			
<i>peso p_i</i>	5	7	4	3	4	6			
nó	x_1	x_2	x_3	x_4	x_5	x_6	peso total	lucro total	obs
1	1	0.43	0	0	0	1	14	44.43	
2	0	1	0.25	0	0	1	14	44	
3	1	0.43	0	0	0	1	14	44.43	
4	0	0	1	0	1	1	14	42	incumbente 1
5	0	1	0.25	0	0	1	14	44	
6	1	0	0.75	0	0	1	14	44	
7	1	1	0	0	0	0.33	14	44.33	dominada
8	0	1	0	0	0.25	1	14	43.75	dominada
9	0	1	1	0	0	0.5	14	43.5	
10	1	0	0	0	0.75	1	14	43.25	
11	1	0	1	0	0	0.83	14	43.83	
12	1	1	0	0	0	0.33	14	44.33	
13									inviável
14	1	0	0	0	0.75	1	14	43.25	
15	1	0	0	1	0	1	14	43	incumbente 2
16	1	1	0	0	0	0.33	14	44.33	
17									inviável
18	1	1	0	0	0	0.33	14	44.33	
19									inviável
20	1	1	0	0	0	0	12	38	dominada
21									inviável

Não há necessidade de expandir mais a árvore a partir do nó 14, uma vez que esse ramo da árvore é claramente dominado pela nova solução incumbente; ou seja, a solução é ainda fracionária e o valor da função objetivo dessa relaxação é 43, valor esse que tende a diminuir quando forem adicionadas novas restrições, mas nunca superar os 43 da melhor solução encontrada até o momento. O mesmo vale para as sub-árvores abaixo dos nós 8 e 9, que não necessitam ser expandidas.

Resta apenas o ramo abaixo do nó 16, que é expandido até o final, conforme mostrado na Figura 2.2g, sendo a solução inteira encontrada, representada pelo nó 20, dominada. Assim, a busca está encerrada, pois não há mais nós a serem investigados e a solução ótima para o problema é $X^* = (1 \ 0 \ 0 \ 1 \ 0 \ 1)$, encontrada no nó 15. Note-se que foram explorados apenas 21 nós, de um total de 64 que corresponderiam à enumeração exaustiva.

Problemas de programação inteira mista, em que variáveis de decisão podem assumir valores inteiros, não limitados apenas a 0 ou 1, também podem utilizar o método B&B para sua resolução. Nesses casos, a expansão de um nó da árvore de busca não pode ser feita como no caso de programação inteira binária, fixando-se valores para as variáveis de decisão, pois

estas podem assumir um número muito grande de valores. A expansão dos nós é feita subdividindo-se o problema segundo faixas de valores para uma variável de decisão, o que representa uma subdivisão do espaço de solução em duas sub-regiões.

Por exemplo, seja x_j a variável (inteira) de decisão, correspondente a um nó em expansão da árvore de busca de um problema inteiro misto, e seja x_j^* o valor fracionário correspondente a x_j na solução ótima obtida com a relaxação das restrições de integralidade. A expansão do nó da árvore de busca é feita criando-se dois subproblemas, segundo os valores que x_j pode assumir. Se $\lceil x_j^* \rceil$ representa o maior número inteiro menor ou igual a x_j^* , então as faixas de valores dos dois subproblemas gerados correspondem a $x_j \leq \lceil x_j^* \rceil$ e $x_j \geq \lceil x_j^* \rceil + 1$. Exemplificando numericamente, se o valor de x_j^* obtido na solução ótima do problema relaxado for igual a $3\frac{1}{4}$, então $\lceil x_j^* \rceil = 3$ e o nó em questão é expandido em dois ramos, segundo as faixas de valores $x_j \leq 3$ e $x_j \geq 4$, como mostrado na Figura 2.3a.

O processo é repetido, chegando-se, eventualmente, à situação em que as faixas de valor, juntamente com as demais restrições do problema, conseguem fixar a variável de decisão x_j . Isto é mostrado na Figura 2.3b, quando as faixas determinadas pela solução do problema relaxado corresponderiam a $x_j \leq 0$ e $x_j \geq 1$, o que conjuntamente com a restrição de não negatividade das variáveis de decisão (i.e., $x_j \geq 0$) acabam por permitir a fixação da variável x_j , gerando dois subproblemas, com $x_j = 0$ e $x_j = 1$.

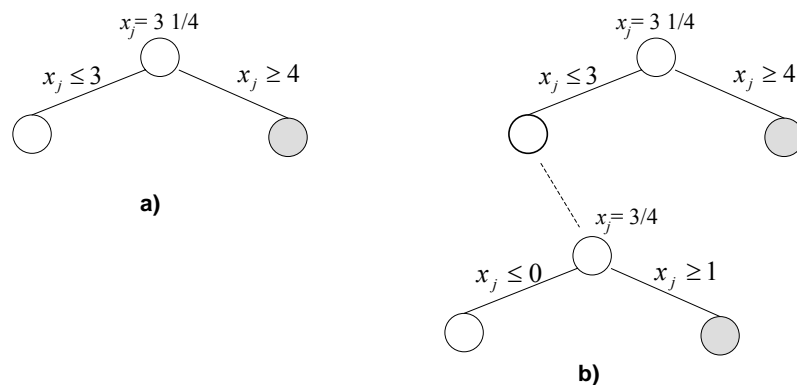


Figura 2.3: Exemplo de *branching* em problemas com variáveis inteiras não binárias

A título de ilustração, a Figura 2.4 mostra a árvore “*branch-and-bound*” completa para o seguinte problema de programação linear inteira mista:

$$\text{maximizar} \quad 8x_1 + 5x_2 \quad (2.4)$$

$$\text{sujeito a} \quad x_1 + x_2 \leq 16 \quad (2.5)$$

$$9x_1 + 5x_2 \leq 45 \quad (2.6)$$

$$x_1, x_2 \geq 0; x_1, x_2 \text{ inteiro} \quad (2.7)$$

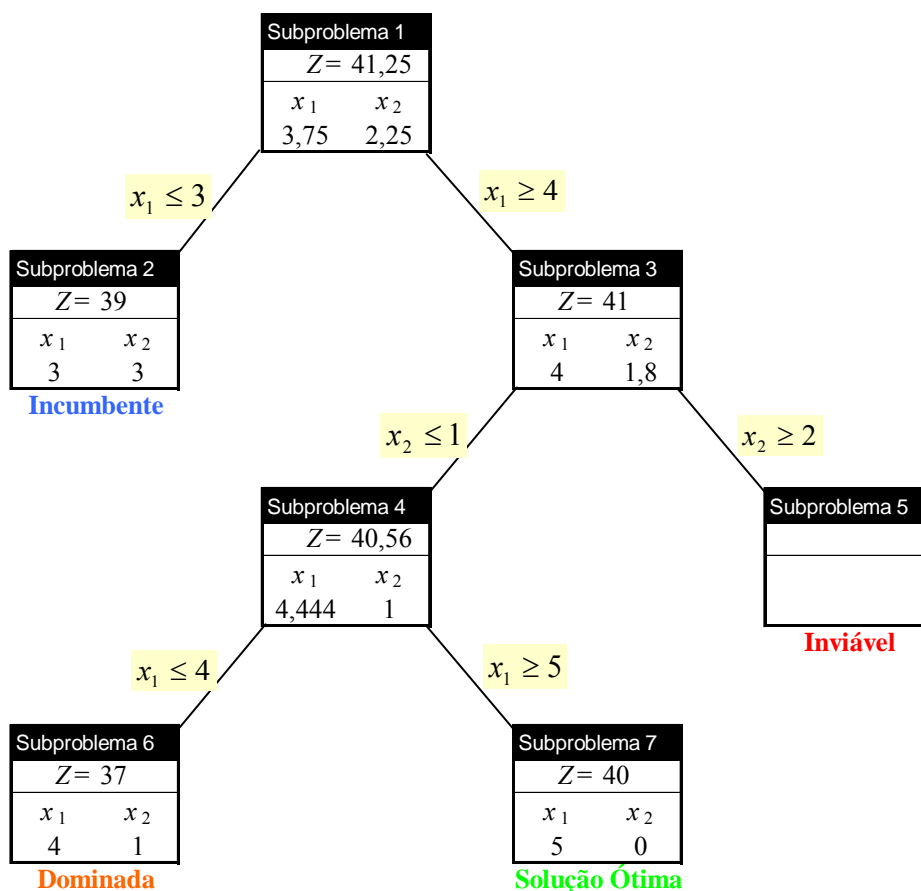


Figura 2.4: Árvore “*branch-and-bound*” para o exemplo de programação inteira mista

O “*branch-and-bound*” é iniciado resolvendo-se a relaxação linear do problema (2.4)-(2.6), que corresponde ao *subproblema 1*. Como ambas as variáveis de decisão resultam fracionárias, seleciona-se, por exemplo, a variável x_1 , impondo-se as restrições $x_1 \leq 3$ e $x_1 \geq 4$, já que $x_1 = 3.75$, resultando nos *subproblemas 2* e *3*, respectivamente. A relaxação linear do subproblema 2 corresponde a uma solução viável para o problema (2.4)-(2.6), já que ambas as variáveis de decisão resultam inteiras; essa solução é incumbente (i.e. melhor solução inteira encontrada até o momento, candidata a solução ótima). Já no subproblema 3, a variável x_2

resulta fracionária, levando à criação dos subproblemas 4 e 5, com a adição das restrições $x_2 \leq 1$ e $x_2 \geq 2$, respectivamente, já que $x_2 = 1.8$.

A relaxação linear do subproblema 5 resulta inviável. A variável x_1 do subproblema 4 resulta fracionária ($x_1 = 4.44$), levando à necessidade de criação de dois novos subproblemas, 6 e 7, com a adição das restrições $x_1 \leq 4$ e $x_1 \geq 5$, respectivamente. Não é possível parar a busca uma vez que o valor da função objetivo da relaxação linear ($Z = 40.56$) é superior à da melhor solução inteira conhecida ($Z = 39$). O subproblema 6 corresponde a uma solução inteira, porém dominada pela solução correspondente ao subproblema 2, uma vez que o valor da função objetivo ($Z = 37$) é menor que o melhor valor conhecido ($Z = 39$). Também o subproblema 7 resulta em solução inteira, que vem a ser a solução ótima para o problema (2.4)-(2.6), uma vez que o seu valor de função objetivo ($Z = 40$) é maior que a solução incumbente anterior, e não há mais nós a expandir (i.e., com solução não inteira). Note-se que, caso houvesse outros nós a expandir, correspondentes a soluções com variáveis fracionárias, poderiam ser eliminados, sem necessidade de expansão, todos os nós cujos respectivos valores de função objetivo da relaxação fossem iguais ou inferiores a 41.

Assim, tendo em vista que, a cada novo nó gerado é adicionada uma nova restrição e resolvida a relaxação linear do novo problema, a utilização do algoritmo dual simplex parece ser a alternativa mais lógica para a solução de cada problema relaxado, uma vez que a nova restrição se transforma em uma nova variável no problema dual, permitindo que se possa partir da solução ótima do problema anterior, sem ter que resolver o problema desde o seu início.

Em síntese, o algoritmo B&B baseado em relaxação linear é a estratégia geral para resolver problemas de programação inteira, não requerendo nenhum tipo de estrutura especial, exceto linearidade, para a sua aplicação. Seu funcionamento pode ser sumarizado da seguinte forma: seja $PI(0)$ a formulação original de PI e seja $PI(k)$ o problema correspondente ao nó k da árvore, com as restrições adicionadas em decorrência da bifurcação (“*branching*”). A relaxação $LP(k)$ nesse nó k da árvore é resolvida, sendo o valor da função objetivo resultante dado por $z(k)$. Se a solução de $LP(k)$ satisfizer as restrições de integralidade, então $x(k)$ é ótima em $PI(k)$, e caso $z(k) < z_{best}$, este último é atualizado (no caso de se estar minimizando); caso contrário, isto é $z(k) \geq z_{best}$, pode-se descartar o nó k , pois a solução é dominada, e a árvore abaixo de $LP(k)$ não necessita ser investigada, mesmo que $x(k)$ não satisfaça as restrições de integralidade. Por outro lado, se $z(k) < z_{best}$ e $x(k)$ não satisfizer as restrições de

integralidade, $PI(k)$ requer exploração adicional, e os descendentes são gerados, como explicado acima, tanto para variáveis binárias quanto inteiras.

Denomina-se “*branch-and-price*” quando o B&B é combinado com um método de geração de colunas, sendo utilizado para resolver problemas de programação inteira em que o número de variáveis ou de restrições é muito grande para serem representadas explicitamente. Assim, novas colunas são geradas conforme necessário durante a solução de problemas de programação linear. Por exemplo, imagine-se resolver um problema de roteirização de veículos enumerando-se todos os possíveis roteiros, sendo que para cada roteiro é possível saber o respectivo custo. Através da resolução do problema de partição (ou particionamento) de conjuntos (“*set partitioning*”) resultante, busca-se determinar o subconjunto de colunas (roteiros) que minimize o custo total, assegurando que cada ponto a ser atendido esteja em apenas um único roteiro. Obviamente, o número de colunas é potencialmente muito elevado para serem todas enumeradas explicitamente. Então, novas colunas vão sendo geradas, com base nos valores dos multiplicadores duais da solução relaxada do problema de particionamento. Boas referências sobre a aplicação do método de geração de colunas à roteirização de veículos com janelas de tempo podem ser encontradas em Desrosiers *et al.* (1984) e Desrochers *et al.* (1992).

Métodos baseados em geração de colunas são específicos para cada problema. De acordo com Barnhart *et al.* (1998), formulações baseadas em geração de colunas são motivadas por inúmeros fatores: (i) uma formulação matemática mais compacta do problema pode resultar em uma relaxação linear fraca, e a obtenção de relaxações melhores pode envolver um número elevado de variáveis; (ii) uma formulação matemática mais compacta do problema pode apresentar uma estrutura simétrica, o que leva o B&B a ter um desempenho ruim, uma vez que o problema muda muito pouco após uma bifurcação (“*branching*”); (iii) a geração de colunas possibilita uma decomposição do problema em mestre e subproblemas, o que pode permitir a consideração de restrições adicionais importantes; (iv) a formulação com um número elevado de variáveis pode ser a única alternativa.

Bramel e Simchi-Levi (1997) demonstraram que a formulação do problema de roteirização de veículos com janelas de tempo como um problema de particionamento de conjuntos produz um limitante inferior de boa qualidade. Para um aprofundamento sobre o assunto sugere-se o artigo de Barnhart *et al.* (1998).

2.1.1 Formas de Pesquisa do Espaço de Soluções pelo Algoritmo B&B

O algoritmo B&B pesquisa o espaço de soluções de maneira sistemática, gerando nós que correspondem a soluções (estados) parciais do problema. Cada nó gerado torna-se um nó ativo, sendo que um nó ativo cujos descendentes estão sendo gerados é chamado de nó em expansão. Uma característica do algoritmo B&B é que o nó em expansão permanece o mesmo até que todos os seus descendentes tenham sido gerados, o que, segundo Horowitz e Sahni (1978), implica a utilização de estruturas de dados (fila, pilha ou “*heap*”) para o armazenamento dos nós ativos. A cada nó gerado é calculada a estimativa de limite inferior (ou superior) do custo da solução parcial correspondente ao nó. São essas estimativas que determinam a eventual eliminação de ramos da árvore de busca no nó gerado, que então se torna um nó inativo e não será mais expandido.

A busca no espaço de soluções pode ser feita de várias maneiras, de acordo com a estrutura de dados utilizada para armazenar os nós gerados porém ainda não expandidos (i.e., para os quais não foram gerados os descendentes). Se a estrutura de dados usada for uma fila (definindo uma disciplina conhecida como FIFO — “*First-In-First-Out*”, em que o primeiro elemento a ser inserido é o primeiro a ser removido), a árvore será percorrida em largura; já se a estrutura for uma pilha (conhecida como LIFO — “*Last-In-First-Out*”, em que o último elemento a ser inserido é o primeiro a ser removido), a árvore será percorrida em profundidade. Alternativamente, pode-se ainda realizar a chamada busca com base no mínimo custo (“*least cost search*”). Nesse caso os nós ainda não expandidos são armazenados em uma estrutura de dados do tipo “*min-heap*” como sugerem Horowitz e Sahni (1978), ou outra que permita obter o nó de mínimo custo acumulado para ser expandido. Esse tipo de estrutura de dados permite a recuperação rápida do nó com menor custo estimado de solução até então, direcionando a busca para o ramo da árvore mais promissor.

Pode-se imaginar ainda um critério de seleção do nó seguinte a ser expandido que leve em consideração não apenas a ordem em que os nós foram gerados, mas também a qualidade das soluções. Esse tipo de critério prevê uma chave dupla, sendo a chave primária a profundidade do nó sendo expandido em relação à árvore, e a chave secundária o custo da estimativa da solução. Nesse caso, entre diferentes nós candidatos a expansão, isto é, para os quais devem ser gerados os descendentes, será escolhido aquele nó situado em uma posição mais profunda na árvore de busca (que corresponde a uma solução parcial com maior número de variáveis já determinadas) e, conseqüentemente, com maior grau de certeza e conhecimento; no caso de

nós candidatos situados no mesmo nível de profundidade da árvore, será escolhido para expansão o nó correspondente à melhor estimativa da função objetivo (ou seja, menor custo, no caso de minimização).

O termo “*backtracking*” designa uma forma específica de percorrer a árvore de busca em profundidade. Uma vez gerado um descendente de um nó, ele imediatamente se torna o nó ativo, antes mesmo que os demais descendentes sejam gerados. Uma vez percorrido todo um ramo da árvore em profundidade, ou encontrada uma solução não viável, retrocede-se ao nó do nível imediatamente acima (daí o termo “*backtracking*”), gerando-se o outro descendente ainda não gerado de um nó (se houver), e seguindo em profundidade da mesma forma por esse novo ramo a partir daí.

As principais virtudes do “*backtracking*” são a possibilidade de obtenção de uma solução viável mais rapidamente e a economia de memória, uma vez que não é necessário armazenar todos os descendentes gerados; por definição, assim que um descendente é gerado ele imediatamente se torna ativo, antes que os demais descendentes sejam gerados. Em geral, o algoritmo “*backtracking*” tem um desempenho inferior ao B&B, embora seja bastante eficiente para resolver alguns problemas específicos. Uma boa descrição do algoritmo pode ser encontrada em Horowitz e Sahni (1978).

Para ilustrar seu funcionamento, considerando-se a árvore de busca binária com quinze nós mostrada na Figura 2.5, a árvore seria percorrida na seguinte ordem: 1-2-4-8-9-5-10-11-3-6-12-13-7-14-15. Note-se que essa ordem é diferente daquela em que a árvore seria percorrida caso se usasse uma estrutura de dados correspondente a uma pilha: 1-2-3-6-7-14-15-12-13-4-5-10-11-8-9.

A título de ilustração, se fosse considerada uma fila, a árvore seria percorrida em largura na seguinte ordem: 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15.

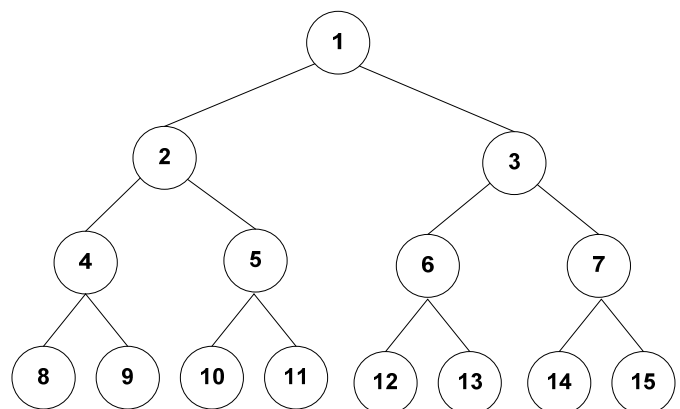


Figura 2.5: Exemplo de árvore de busca

De qualquer maneira, independentemente da forma de percorrer a árvore, deve-se ter em mente evitar que a lista de nós candidatos armazenados fique muito grande, dando preferência a nós que estão mais embaixo na árvore.

Outro aspecto a considerar é a ordem das variáveis. Algumas regras práticas indicam que (i) vale a pena bifurcar (“*branching*”) primeiro nas variáveis mais importantes (por exemplo, num problema de localização, primeiro nas variáveis de localização das instalações e depois nas variáveis de alocação dos pontos às instalações selecionadas); (ii) quando parecer evidente que $x_j = 1$ na solução ótima, é vantajoso bifurcar em $x_j = 0$ e $x_j = 1$, na esperança de que o ramo $x_j = 0$ possa ser eliminado.

Segundo Johnson *et al.* (2000), uma regra simples consiste em escolher a variável cujo valor fracionário esteja mais próximo de $\frac{1}{2}$, ou seja, que apresente a maior inviabilidade em termos de integralidade. Regras mais sofisticadas, muitas delas usadas em softwares comerciais, tentam escolher a variável que cause uma deterioração rápida no valor da função objetivo da relaxação linear, permitindo eliminar mais cedo partes do espaço de solução, sem que muitos descendentes tenham que ser gerados; tal estratégia pode ser crítica para manter a árvore pequena. Dado que pode ser muito custoso calcular as variações na função objetivo para todas as variáveis candidatas em cada nó em expansão, estimativas da taxa de variação (chamada pseudo-custo) são calculadas a partir das informações do dual.

Embora a relaxação linear seja a alternativa mais comum no B&B como forma de percorrer implicitamente a árvore de soluções e de obter estimativas da qualidade da solução sendo gerada, outras possibilidades podem ser exploradas. Por exemplo, em um problema de caixeiro viajante, ao invés de se resolver o problema relaxando-se a integralidade das variáveis de decisão binárias, pode-se, alternativamente, resolver o problema inteiro, porém sem as restrições de eliminação de sub-roteiros, resultando em um problema de atribuição que pode ser resolvido com certa facilidade através de métodos baseados em fluxo em rede; ou alternativamente, pode-se resolver o problema de árvore de cobertura mínima.

De qualquer maneira, quanto menor a diferença (“*gap*”) entre o resultado da função limitante e o valor correspondente da solução inteira, melhor. Entretanto, há claramente um “*trade-off*” entre a qualidade do limitante obtido e o tempo para a sua obtenção.

Grandes vantagens podem surgir se o problema puder ser formulado de tal forma que os subproblemas lineares possam ser modelados como problemas de fluxo em rede (“*network flow models*”), para os quais existem algoritmos específicos e muito eficientes, com desempenho significativamente superior aos dos métodos simplex e dual simplex, usualmente utilizados nos algoritmos B&B (Ahuja *et al.*, 1993).

Uma boa solução que possa ser obtida através de alguma heurística rápida também pode auxiliar na eliminação de partes da árvore de solução, principalmente em conjunto com uma boa função limitante.

Outro recurso utilizado para melhorar o desempenho do algoritmo B&B são os chamados planos de corte (“*cutting planes*”). A idéia é adicionar novas restrições que eliminem (cortem) soluções fracionárias sem prejudicar nenhuma solução inteira, como mostrado na Figura 2.6.

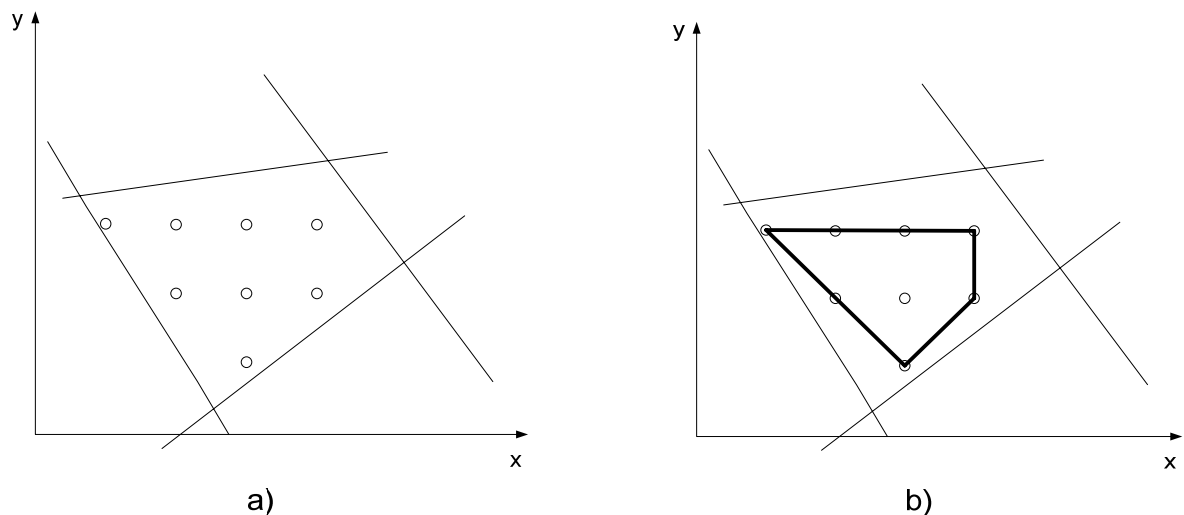


Figura 2.6: Exemplo da introdução de planos de corte (“*cutting planes*”)

Se forem adicionadas as restrições corretas, cada vértice do problema linear corresponderá a uma solução inteira, permitindo determinar a solução inteira através da resolução do problema linear, isto é, sem as restrições de integralidade das variáveis de decisão.

Por exemplo, na formulação do problema da mochila apresentado anteriormente (2.1-2.3), a relaxação linear (correspondente ao nó 0 da árvore B&B) resultou $X = (1 \ 0.43 \ 0 \ 0 \ 0 \ 1)$, com lucro total de 44.43 e peso total de 14. Portanto, a restrição de peso resultante $5x_1 + 7x_2 + 6x_6 \leq 14$ (uma vez que as variáveis x_3 , x_4 e x_5 resultam nulas) pode dar origem à

restrição $x_1 + x_2 + x_6 \leq 2$, uma vez que não é possível ter-se simultaneamente $x_1 = x_2 = x_6 = 1$, mas apenas no máximo duas delas.

Resolvendo-se a relaxação linear com essa restrição adicional obtém-se $X = (0 \ 1 \ 0,25 \ 0 \ 0 \ 1)$, com lucro total de 44 e peso total de 14. A nova restrição de peso $7x_2 + 4x_3 + 6x_6 \leq 14$ permite adicionar uma nova restrição $x_2 + x_3 + x_6 \leq 2$, uma vez que não é possível ter-se simultaneamente $x_2 = x_3 = x_6 = 1$. Resolvendo-se novamente o problema relaxação com mais essa restrição obtém-se a nova solução $X = (1/3 \ 1 \ 1/3 \ 0 \ 0 \ 2/3)$, com lucro total de 44 e peso total de 14. A restrição de peso decorrente $5x_1 + 7x_2 + 4x_3 + 6x_6 \leq 14$ (já que x_4 e x_5 são nulas) dá origem à restrição $x_1 + x_2 + x_3 + x_6 \leq 2$, uma vez que apenas duas das quatro variáveis não nulas na relaxação linear (x_1, x_2, x_3 e x_6) podem ser simultaneamente não nulas; note-se que essa restrição domina as duas anteriores.

Entretanto, para problemas de grande porte, essas restrições podem ser difíceis de determinar (Nemhauser e Wolsey, 1988). Assim, ao invés de se particionar toda a região viável do problema original, o que se faz normalmente é tratar cada relaxação linear separadamente, adicionando restrições iterativamente até que uma solução inteira possa ser determinada.

Já os cortes de Gomory (“*Gomory cuts*”) correspondem a um método para gerar cortes a partir dos “*tableaus*” simplex da programação linear, resultantes da relaxação linear. Por exemplo, considere-se uma restrição resultante em um problema de programação linear: $1.6x_1 + 4.2x_2 + 3x_3 + 0.4x_4 = 9.8$; ou seja, isolando-se as frações teremos $0.6x_1 + 0.2x_2 + 0.4x_4 = \text{Inteiro} + 0.8$, o que permite impor $0.6x_1 + 0.2x_2 + 0.4x_4 \geq 0.8$, ou colocando em forma de frações, $3/5x_1 + 1/5x_2 + 2/5x_4 \geq 4/5$, o que resulta $3x_1 + x_2 + 2x_4 \geq 4$, restrição esta que pode ser adicionada ao problema. Em linhas gerais, o método de cortes de Gomory consiste em encontrar uma variável básica cujo valor resulta fracionário; adicionar uma restrição de desigualdade (note que os únicos coeficientes fracionários na restrição correspondem aos das variáveis não básicas); adicionar a nova restrição ao problema e resolvendo a relaxação linear novamente. Uma boa referência sobre o assunto é o artigo de Marchand e Wolsey (2001).

Embora o algoritmo de Gomory convirja para uma solução ótima em um número finito de iterações, essa convergência pode ser extremamente lenta devido ao fato de que esses cortes derivados algebricamente são geralmente “fracos”, no sentido de que eles freqüentemente

nem definem hiperplanos ao polígono convexo de pontos viáveis, e não podem ser tornados mais “fortes” sem eliminar algumas soluções viáveis inteira do problema. Considerável esforço de pesquisa vem sendo dedicado à identificação de parte dessas inequações lineares (ou todas) para problemas combinatórios específicos. Tendo em vista que para muitos problemas inteiros o número de inequações que descrevem esse poliedro pode crescer exponencialmente com o número de variáveis, há de se indagar se tal abordagem algum dia virá a ser computacionalmente eficiente.

Existem inúmeros outros tipos de cortes que vem sendo adicionados aos métodos de enumeração implícita dos *softwares* comerciais, incluindo de cortes de clique (“*clique cuts*”), de cobertura (“*cover cuts*”), disjuntivos (“*disjunctive*”), entre outros. Quando o método B&B utiliza cortes, é usualmente chamado de “*branch-and-cut*”. Para um aprofundamento maior sobre o assunto, recomendam-se os trabalhos de Hoffman e Padberg (1985), Nemhauser e Wolsey (1988). e Bertsimas e Weismantel (2005).

2.1.2 Avanços na Resolução de Problemas de Programação Inteira

De acordo com Bixby *et al.* (1999), embora o potencial da modelagem de problemas de otimização como problemas de programação inteira já tenha sido constatado entre 1950 e 1960 (tendo sido feitas inúmeras tentativas para modelar e resolver problemas reais), naquela época os computadores e softwares disponíveis estavam muito aquém das necessidades, trazendo, como resultado, decepção e desilusão, cujos efeitos se fazem sentir até o presente. Muitos analistas de PO ainda acreditam que a programação inteira mista é um assunto interessante para discutir, porém tem limitada aplicabilidade prática. Entretanto, como destacam os autores, esta situação está mudando muito rapidamente; atualmente é possível resolver muitos problemas difíceis, interessantes e práticos de PI utilizando pacotes de otimização comerciais de prateleira (“*off-the-shelf*”).

Segundo os mesmos, até meados da década de 90, muito pouco das descobertas e dos avanços científicos haviam sido incorporados aos principais pacotes comerciais para resolução de problemas de PI, tais como o CPLEX, o LINGO e o XPRESS-MP. Atualmente, a maioria deles já incorpora planos de corte e muitos outros recursos resultantes dos desenvolvimentos teóricos alcançados. Como sugerem os autores, o “*gap*” entre a teoria e prática está se fechando.

Não são apenas os computadores que evoluíram em termos de velocidade de processamento e capacidade de memória, embora um microcomputador no início da década de 2000 possuía uma velocidade de processamento muitas vezes maior que os melhores supercomputadores disponíveis 10 anos antes (Bixby, 2002). Além de velocidade, aumentos significativos de capacidade de memória permitiram não só resolver problemas maiores, como também conceber e implementar estratégias de solução totalmente diferentes. Segundo Bixby (2002), muitos dos algoritmos que são fundamentais para os pacotes mais modernos de PL simplesmente não poderiam ser implementados caso não houvesse tamanha abundância de memória. Adicionalmente, os avanços das linguagens de programação e os recursos dos softwares tornaram muito mais simples desenvolver, testar e validar códigos e sistemas complexos.

Um dos passos importantes nessa direção é o impressionante aumento de eficiência dos algoritmos de programação linear, principalmente com a introdução dos algoritmos dual simplex e de barreiras (“*barrier*”). O aumento de desempenho desses algoritmos de PL é absolutamente essencial para o B&B, uma vez que os mesmos são aplicados inúmeras vezes, já que a relaxação linear é resolvida a cada novo nó da árvore gerado. Segundo Bixby (2002), embora o algoritmo dual simplex tenha sido introduzido em 1954, a primeira versão comercial data de 1987. O desempenho do algoritmo dual é muito superior ao do primal simplex. Os algoritmos primal-dual com barreira também se beneficiaram muito das idéias de algoritmos de pontos interiores proposta por Karmarkar (1984), conforme descrevem Lustig *et al.* (1994).

A título de ilustração desse incrível ganho de desempenho, Bixby (2002) cita o exemplo de um conjunto de problemas de programação linear ligados à logística militar, conhecidos como PDS (“*patient-distribution system*”). A instância *pds30* compreende 177.628 variáveis, 49.994 restrições e 393.657 coeficientes não nulos. Utilizando o CPLEX 1.0 (de 1988), o tempo para a obtenção da solução ótima foi de 15.891 segundos, o CPLEX 5.0 (1996) levou 1.155 segundos, e o CPLEX 7.1 (2000) apenas 75 segundos (usando o algoritmo primal simplex), e 39 segundos (usando o dual simplex). Ou seja, a versão 7.1 foi mais de 400 vezes mais rápida, sendo que todos os testes foram feitos utilizando o mesmo computador. A versão maior dessa classe de problemas, *pds100*, não podia ser resolvida utilizando o CPLEX 1.0, levava 50.413 seg no CPLEX 5.0, 2.415 seg no CPLEX 7.1 primal e apenas 256,3 seg no CPLEX 7.1 dual. As análises realizadas por Bixby indicam que o CPLEX 1.0 apresenta um comportamento quadrático com o tamanho do problema, enquanto o CPLEX 7.1 aparenta crescer linearmente.

O autor também reporta um problema de programação linear com 401 mil restrições e 1,584 milhões de variáveis de decisão, rodando numa máquina PIV 2.0GHz, que levou 29,8 dias para ser resolvido através do CPLEX 1.0, e apenas 59 segundos usando o CPLEX 9.0, recém lançado. Em síntese, um modelo que levaria cerca de um ano para ser resolvido no início da década de 80, pode atualmente ser resolvido em menos de 30 segundos. Outros experimentos realizados indicam que o algoritmo dual simplex é cerca de duas vezes mais rápido que o primal simplex.

Já para problemas de PI, diferentes técnicas foram desenvolvidas e incorporadas aos pacotes comerciais para serem aplicadas a cada modelo (Bixby *et al.*, 1999), embora o paradigma fundamental de enumeração com relaxação linear não tenha efetivamente mudado. A diferença é que os novos avanços permitem eliminar significativamente mais nós da árvore, reduzindo o tamanho da busca, embora às vezes o tempo de processamento de um nó tenha aumentado. Segundo os autores, a partir da versão 6.5 do CPLEX foram incorporadas seis diferentes tipos de rotinas de planos de corte (“*cutting planes*”), cada uma com seus parâmetros *default* que definem quando, e com que frequência devem ser aplicadas: (i) coberturas de mochila (“*knapsack covers*”), (ii) coberturas GUB (“*GUB covers*”), (iii) coberturas de fluxo (“*flow covers*”), (iv) cliques, (v) limitantes implícitos (“*implied bounds*”), e (vi) cortes inteiro-mistos de Gomory (“*Gomory Mixed-Integer Cuts*”). Os autores apresentam uma breve descrição de cada um deles, bem como as respectivas referências mais importantes. Outra fonte importante é a obra de Nemhauser e Wolsey (1988).

Outro recurso disponível corresponde às heurísticas de nó (“*node heuristics*”), cuja finalidade é forçar a obtenção de uma solução inteira em um nó através de operações locais no nó, ao invés de aguardar o B&B forçar a integralidade das variáveis. A partir da solução linear obtida, mergulha-se na árvore a fim de se fixar um número cada vez maior de variáveis, até que se obtenha uma solução inteira melhor (incumbente), ou inviável ou dominada. Várias técnicas diferentes são descritas sucintamente por Bixby *et al.* (1999).

Utiliza-se ainda o pré-processamento, que compreende operações elementares realizadas com a finalidade de melhorar ou simplificar a formulação através de limitantes melhores para as variáveis, fixando variáveis, etc., além de detectar inviabilidade (Nemhauser e Wolsey, 1988). Um limitante de uma variável pode ser melhorado caso se possa reconhecer que a restrição se torna inviável se uma variável for definida para aquele valor limitante. Por exemplo, a

restrição: $5x_1 + 3x_2 \geq 4$ torna-se inviável quando x_1 é definido como 0. Assim, conclui-se que x_1 deve ser 1.

O pré-processamento pode ser visto como uma fase intermediária entre a formulação e a resolução de um problema. Ele pode significativamente aprimorar a velocidade de um algoritmo sofisticado que não seja capaz de reconhecer que alguma variável possa ser fixada e, portanto, eliminada do modelo, ou então que determinadas restrições redundantes possam ser eliminadas (por exemplo, a restrição $3x_1 + 2x_2 \leq 6$ pode ser eliminada se as variáveis x_1 e x_2 forem binárias). Tais técnicas também são conhecidas como “*node presolve*” ou “*root reductions*” (Bixby *et al.*, 1999), pois se referem a tudo que pode ser feito antes da busca B&B começar.

Testes com outros conjuntos de problemas de grande porte reportados por Bixby (2002) indicam que o “*presolve*” consegue reduzir significativamente o tamanho de inúmeros problemas diferentes.

Outros detalhes sobre avanços nos algoritmos baseados em programação linear para problemas de programação inteira podem ser encontrados no excelente artigo de Johnson *et al.* (2000).

2.1.3 Formulações Matemáticas mais Robustas

De acordo com Johnson *et al.* (2000), diferentes formulações de um mesmo problema de programação inteira podem levar a diferentes limitantes inferiores (no caso de minimização), em função de produzirem diferentes relaxações de PL. Assim, boas formulações são caracterizadas por relaxações de PL que proporcionam bons limitantes.

Os autores descrevem um exemplo que vale a pena ser reproduzido, pois ilustra como restrições modificadas podem levar a uma formulação melhor. Considere-se o problema do buraco do pombo (“*pigeon hole*”), em que há $n + 1$ pombos e n buracos; deseja-se colocar todos os pombos nos buracos, de tal forma que dois pombos nunca ocupem o mesmo buraco. Obviamente, esse problema não tem solução, pois há mais pombos que buracos.

Esse problema pode ser modelado da seguinte forma. Seja $x_{ij} = 1(0)$ se o pombo i estiver (não estiver) no buraco j . Busca-se alocar cada pombo a um buraco, levando à seguinte restrição:

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n + 1 \quad (2.8)$$

Entretanto, dois pombos quaisquer i e k não podem ocupar o mesmo buraco j , o que leva à seguinte restrição ligada aos buracos:

$$x_{ij} + x_{kj} \leq 1 \quad i = 1, \dots, n, \quad k = 1, \dots, n + 1, \quad j = 1, \dots, n \quad (2.9)$$

Esse sistema é obviamente inviável quando as variáveis de decisão x_{ij} são binárias. Entretanto, quando as restrições de integralidade são relaxadas, $x_{ij} = (1/n)$ corresponde a uma solução válida para todo i e j . Adicionalmente, após muitas bifurcações (“*branchings*”), a relaxação linear continua viável, de tal modo que a quase totalidade do espaço de solução tem que ser enumerado, e a relaxação linear acaba não tendo utilidade prática.

Por outro lado, observa-se que a restrição dos buracos pode ser escrita como:

$$\sum_{i=1}^{n+1} x_{ij} \leq 1 \quad j = 1, \dots, n \quad (2.10)$$

isto é, no máximo um pombo por buraco. A relaxação de programação linear com essas n restrições, cada uma delas contendo $n + 1$ variáveis (em substituição às n^3 restrições, cada uma com duas variáveis) é muito mais apertada (“*tight*”). De fato, segundo os autores, fica fácil mostrar que o problema é inviável. Esse segundo conjunto de restrições é conhecido como restrições de clique; mesmo que não estejam presentes na formulação original, elas podem ser derivadas na fase de pré-processamento, dependendo do algoritmo utilizado.

Desagregação de restrições é outro princípio geral para se obter boas formulações (Johnson *et al.*, 2000). Ao invés de usar $x_1 + x_2 + \dots + x_n \leq my$ para representar que $y = 0$ implica que qualquer x_j seja igual a 0, para todo $j = 1, \dots, n$, é melhor usar a forma desagregada $x_j \leq y$ $j = 1, \dots, n$, que fornece uma relaxação mais apertada (“*tighter*”). Note-se que isso vai contra a idéia intuitiva de que quanto menos restrições melhor.

Considere-se agora um problema em que é importante a decisão correta das variáveis de decisão. Seja um modelo de produção em lotes não capacitado, com períodos de tempo discretos e horizonte finito. Há uma demanda d_t a ser atendida em cada período $t = 1, \dots, T$, utilizando o que foi produzido no período atual t e nos anteriores, isto é, $k = 1, \dots, t$. Em cada período t são conhecidos o custo unitário de produção, o custo unitário de estoque e um custo

de *setup* que só incide se houver produção no período. O problema é não capacitado, na medida em que não há limite na quantidade produzida y_t no período t . Entretanto, há um limitante implícito:

$$y_t \leq \sum_{k=t}^T d_k = D_t \quad t = 1, \dots, T \quad (2.11)$$

A continuidade de estoque (que não pode ser negativo, ou seja, sem “backlog”) é dada pelas variáveis de estoque:

$$s_t = s_{t-1} + y_t - d_t \quad t = 1, \dots, T \quad (2.12)$$

onde se assume que o estoque inicial $s_0 = 0$.

Os custos de “*setup*” são modelados através de variáveis binárias x_t que definem se há produção no período t . Assim, as restrições de limitante superior (capacidade) ficam:

$$y_t \leq D_t x_t \quad t = 1, \dots, T \quad (2.13)$$

Assumindo-se que os custos de “*setup*” sejam positivos, relaxando-se as variáveis binárias x_t , isto é, $0 \leq x_t \leq 1$, obtém-se:

$$x_t = y_t / D_t \quad t = 1, \dots, T \quad (2.14)$$

Assim, se y_t for positivo e estiver no seu limite superior, então x_t resultará fracionário. Tendo em vista que não resultaria o ótimo produzir toda a demanda corrente e futura em mais períodos que o mínimo necessário, devido aos custos de “*setup*”, todos exceto um dos valores x_t serão fracionários, o que leva a concluir que a relaxação linear não é muito boa.

É possível eliminar-se as variáveis de estoque usando variáveis y_{tk} que representam a quantidade produzida no período t para atender a demanda no período k , onde $k \geq t$. Agora as restrições de demanda podem ser escritas como:

$$\sum_{t=1}^k y_{tk} = d_t \quad k = 1, \dots, T \quad (2.15)$$

e as restrições de limitante superior tornam-se

$$y_{tk} \leq d_k x_t \quad k = 1, \dots, T \quad t = 1, \dots, k \quad (2.16)$$

Essas restrições de limitante superior são muito melhores para a relaxação linear. Em particular, segundo Johnson *et al.* (2000), pode-se demonstrar que existem soluções ótimas para o problema de programação inteira, e para a relaxação linear, tais que y_{tk} implica $y_{tk} = d_k$. Esse resultado implica que, ao se resolver a relaxação linear, resolve-se também o problema de programação inteira.

2.2 RELAXAÇÃO LAGRANGEANA

Segundo Fisher (1985), os problemas de otimização combinatória podem ser divididos em dois grupos. Há um número reduzido de problemas fáceis, que podem ser resolvidos em tempo limitado por uma função polinomial. Por outro lado, há um número elevado de problemas ditos difíceis, para os quais a ordem de complexidade dos algoritmos exatos conhecidos é exponencial.

Uma das idéias mais úteis do ponto de vista da implementação e eficiência computacional, segundo o autor, é a observação de que muitos problemas de programação inteira ditos difíceis podem ser visualizados como problemas fáceis, que se tornam complicados quando é adicionado um conjunto reduzido de restrições.

A Relaxação Lagrangeana é uma técnica de decomposição que pode ser aplicada a um problema de otimização em que as restrições complicadas são substituídas por um ou mais termos com penalidades introduzidos na função objetivo.

Seja (P) um problema de programação inteira, cuja formulação matemática pode ser escrita como:

$$Z = \min cx \quad (P) \quad (2.17)$$

sujeito a:

$$Ax = b \quad (2.18)$$

$$Dx \leq f \quad (2.19)$$

$$x \geq 0 \text{ e inteiro.} \quad (2.20)$$

onde x é um vetor de dimensões $(n \times 1)$, b tem dimensões $(m \times 1)$, f é do tipo $(k \times 1)$ e todas as demais matrizes têm as dimensões adequadas decorrentes.

Assume-se que as restrições de (P) tenham sido separadas em dois conjuntos $(Ax = b$ e $Dx \leq f)$ e que o problema (P) possa ser resolvido com facilidade caso o conjunto de restrições ditas “difíceis” $Ax = b$ seja removido.

Seja $\mu = \{\mu_1, \mu_2, \dots, \mu_m\}$ um vetor $(m \times 1)$ de multiplicadores de Lagrange. O problema Lagrangeano é obtido removendo-se as restrições $Ax = b$, e adicionando-se o termo $(Ax - b)$ à função objetivo de (P) , devidamente penalizado pelos multiplicadores μ , o que resulta no seguinte problema relaxado (RL_μ) :

$$Z_D(\mu) = \min cx + \mu(Ax - b) \quad (RL_\mu) \quad (2.21)$$

sujeito a:

$$Dx \leq f \quad (2.22)$$

$$x \geq 0 \text{ e inteiro.} \quad (2.23)$$

Assumindo-se que (P) seja viável e que o conjunto $X = \{x : Dx \leq f, x \geq 0 \text{ e inteiro}\}$ de soluções viáveis de (RL_μ) seja finito, então $Z_D(\mu)$ é finito para todo μ .

Ao se removerem as restrições $Ax = b$, o valor ótimo $Z_D(\mu)$ corresponde a um limitante inferior de Z , pois o problema (RL_μ) torna-se menos restrito; portanto $Z_D(\mu) \leq Z$. Seja x^* o vetor correspondente à solução ótima de (P) . Conseqüentemente, $Z_D(\mu) \leq cx^* + \mu(Ax^* - b) \leq Z$.

Caso $Ax = b$ seja substituído por $Ax \leq b$, é necessário impor $\mu \geq 0$ para que $Z_D(\mu) \leq cx^* + \mu(Ax^* - b) \leq Z$. Analogamente, se $Ax \geq b$ então $\mu \leq 0$ para que $Z_D(\mu) \leq Z$.

Métodos baseados em relaxação Lagrangeana podem contribuir para o desenvolvimento de heurísticas eficientes para a solução de problemas complexos de programação inteira e de otimização combinatória.

Uma das virtudes da relaxação Lagrangeana é a resolução do subproblema como um problema independente. Conseqüentemente, podem ser explorados quaisquer algoritmos ou métodos para a sua resolução.

Dependendo do problema e da restrição que é relaxada, o problema Lagrangeano pode ser simples de resolver e fornecer um bom limitante inferior (no caso de minimização) para a solução ótima do problema original. Sendo assim, esta técnica pode ser utilizada no lugar de

relaxações de programação linear, a fim de prover funções limitantes em um algoritmo "branch-and-bound" (Fisher, 1985).

Os conceitos de relaxação Lagrangeana, como são conhecidos hoje, devem-se a Held e Karp (1971), que formularam um problema Lagrangeano, baseado em árvores de cobertura mínima, como parte de um algoritmo eficiente para o problema do caixeiro viajante. Desde então, a técnica tem sido utilizada em inúmeros problemas de otimização combinatória, possibilitando a solução de problemas práticos reais (Fisher, 1985).

2.2.1 Solução do Problema Relaxado

Uma questão importante que surge é como selecionar um valor adequado para o vetor de multiplicadores μ , ou ainda, para que valores de μ se obtém uma solução $Z_D(\mu)$ igual ou muito próxima de Z .

De acordo com Fisher (1981), a melhor escolha para μ estaria na solução ótima do problema dual de otimização:

$$Z_D^* = \max_{\mu} Z_D(\mu) \quad (2.24)$$

Sabe-se ainda que:

$$Z_D(\mu) \leq Z_D^* \leq Z^* \leq cx \quad (2.25)$$

Com base nessas condições, pode-se garantir para que condições um vetor μ é ótimo para o problema dos multiplicadores de Lagrange ou uma solução viável x do problema original (P):

- se μ é um vetor de multiplicadores de Lagrange e x uma solução viável para o problema (P) que satisfaz a condição $Z_D(\mu) = cx = Z$, então $Z_D(\mu)$ corresponde à solução ótima para o problema relaxado (RL_{μ}), e $x = x^*$ é a solução ótima para o problema (P).
- se para alguma escolha do vetor de multiplicadores de Lagrange μ , a solução x^* da relaxação Lagrangeana (RL_{μ}) for viável para o problema (P), então x^* é a solução ótima para o problema (P) e μ a solução ótima para problema relaxado (RL_{μ}).

Em geral, não é possível assegurar que se possa encontrar sempre um vetor de multiplicadores μ tal que $Z_D(\mu) = Z$, embora seja possível determiná-lo para muitas instâncias particulares de problemas.

Os multiplicadores μ podem ser determinados através de um método baseado em otimização por subgradiente. De acordo com Fisher (1985), esse método é um dos mais populares e consiste em uma adaptação do método do gradiente. Dado um valor inicial μ_0 para o vetor de multiplicadores, a seqüência μ_k de multiplicadores, correspondentes às iterações $k=1, 2, \dots$ pode ser gerada através da seguinte expressão:

$$\mu^{k+1} = \mu^k + t_k (Ax^k - b) \quad (2.26)$$

onde x^k é a solução ótima da k -ésima iteração do problema relaxado e t_k é um escalar positivo que define quão longe se deseja mover na direção do gradiente na iteração k , sendo calculado da seguinte forma:

$$t_k = \frac{\lambda_k (Z_D^* - Z_D(\mu^k))}{\|Ax^k - b\|^2} \quad (2.27)$$

Tendo em vista que normalmente não se conhece o valor ótimo Z_D^* do problema de otimização dos multiplicadores de Lagrange (afinal de contas, é o que se está tentando buscar), utiliza-se a seguinte heurística para determinar o tamanho do passo:

$$t_k = \frac{\lambda_k (UB - Z_D(\mu^k))}{\|Ax^k - b\|^2} \quad (2.28)$$

onde UB corresponde a um limitante superior de Z^* para o problema (P) (e conseqüentemente um também do valor ótimo Z_D^* para o problema (RL_μ) e λ_k um escalar escolhido entre 0 e 2.

Inicialmente UB pode ser qualquer valor correspondente a uma solução conhecida para (P) , sendo atualizado à medida em que a relaxação Lagrangeana prossegue e uma nova solução viável (e melhor) é obtida para o problema (P) . Recomenda-se que o escalar λ_k inicie com o valor $\lambda_0 = 2$, sendo reduzido pela metade sempre que um novo valor melhor para a função objetivo do problema relaxado deixe de ser obtido após um certo número de iterações. Usualmente o critério de parada é definido como um número máximo de iterações.

Fisher (1985) e Ahuja *et al.* (1993) destacam a importância do tamanho do passo t_k : se for escolhido um valor muito pequeno, o algoritmo pode ficar preso em um ponto corrente e não convergir; ao contrário, caso se escolha um valor muito grande, t_k pode levar a que a solução ótima seja ultrapassada e talvez o algoritmo fique oscilando entre duas soluções não ótimas. Uma boa alternativa para o valor de t_k é escolher valores de modo a garantir que $t_k \rightarrow 0$ e $\sum_{j=1}^k t_k \rightarrow \infty$. Por exemplo, a escolha de $t_k = 1/k$ satisfaz esses requisitos.

A seguir apresentam-se três exemplos que ilustram a aplicação da Relaxação Lagrangeana em diferentes contextos.

2.2.2 O Problema de Caminho Mínimo Restrito

Este problema foi apresentado por Ahuja *et al.* (1993) para ilustrar os conceitos e a aplicação da Relaxação Lagrangeana.

Considere-se a rede (ou grafo) $G=(N, A)$ mostrada na Figura 2.7, em que cada arco (i, j) possui dois atributos: um custo c_{ij} e um tempo de viagem (ou travessia) t_{ij} .

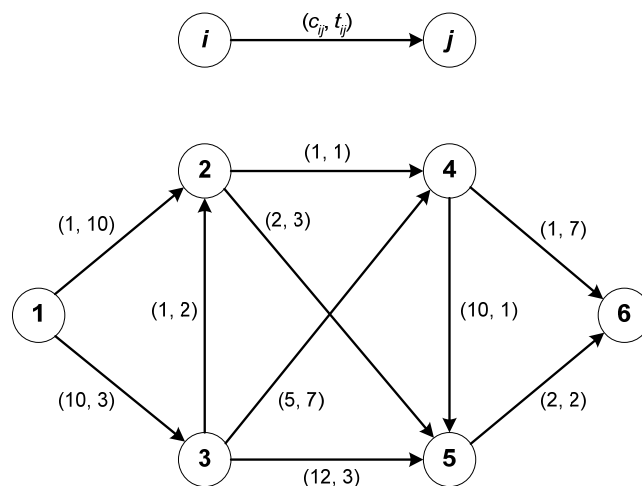


Figura 2.7: Problema de caminho mínimo com restrição de tempo

Fonte: Ahuja *et al.* (1993)

Deseja-se encontrar o caminho mínimo entre o nó de origem (1) e o nó de destino (6) dessa rede G , porém assegurando que o tempo total de viagem não ultrapasse $T=10$ unidades de tempo. Esse tipo de aplicação de caminho mínimo restrito surge com frequência na prática,

uma vez que em muitos contextos uma empresa (por exemplo, uma empresa de encomendas expressas) deseja prover serviços ao menor custo possível, porém assegurando um certo nível de serviço aos seus clientes (refletido na restrição de tempo total T).

O problema de caminho mínimo restrito do nó 1 ao nó n pode ser formulado como o seguinte problema de programação linear inteira:

$$\text{minimizar } \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.29)$$

sujeito a

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ji} = \begin{cases} 1 & \text{para } i = 1 \\ 0 & \text{para } i \in N - \{1, n\} \\ -1 & \text{para } i = n \end{cases} \quad (2.30)$$

$$\sum_{(i,j) \in A} t_{ij} x_{ij} \leq T \quad (2.31)$$

$$x_{ij} \in \{0, 1\} \quad \text{para todo } (i, j) \in A \quad (2.32)$$

Este problema corresponde a um problema de programação linear inteira, que seria normalmente resolvido através do algoritmo B&B. Entretanto, esse problema corresponde a um problema de caminho mínimo com restrição adicional de tempo (2.30).

Nesse contexto, a relaxação Lagrangeana consiste em mover a restrição (2.30) para a função objetivo, devidamente penalizada por um multiplicador $\mu \geq 0$. Assim, a função objetivo do problema relaxado pode ser escrita como:

$$\text{minimizar } \sum_{(i,j) \in A} (c_{ij} + \mu t_{ij}) x_{ij} - \mu T \quad (2.33)$$

sujeito às restrições (2.29) e (2.31). Note-se que neste caso é necessário apenas um multiplicador μ e não um vetor de multiplicadores, uma vez que se está movendo apenas uma única restrição para a função objetivo.

Esse problema relaxado pode ser visto como um problema de caminho mínimo em um grafo cujos custos nos arcos (i, j) são afetados por uma parcela de penalidade (μt_{ij}) que é proporcional ao tempo para transpor cada arco. Para um dado valor de μ , o problema relaxado pode ser resolvido de maneira muito eficiente e rápida através de algum algoritmo de caminho mínimo tal como, por exemplo, o algoritmo de Dijkstra (Ahuja *et al.*, 1993).

Naturalmente, o valor da função objetivo para o problema relaxado é um limitante inferior do problema original para qualquer valor de multiplicador $\mu \geq 0$.

Seja P um caminho viável para o problema original (2.28-2.31). Definem-se ainda o custo total $c_P = \sum_{(i,j) \in P} c_{ij}$ e o tempo total $t_P = \sum_{(i,j) \in P} t_{ij}$ nesse caminho P . Se P é um caminho viável, então naturalmente $t_P \leq T$.

Para o problema relaxado, o custo total em um caminho é dado por $(c_P + \mu t_P)$, e a função objetivo é dada por $(c_P + \mu t_P - \mu T)$, que pode ser reescrita como $[c_P + \mu (t_P - T)]$; obviamente $[c_P + \mu (t_P - T)] \leq c_P$, pois $(t_P \leq T)$, o que faz com que o segundo termo da função objetivo do problema relaxado seja menor ou igual a zero, já que $\mu \geq 0$.

Por exemplo, para $\mu = 1$ o caminho mínimo para o problema relaxado é $P = (1-2-5-6)$, com $c_P = 5$ e $t_P = 15$, e o valor da função objetivo do problema relaxado é igual a 10; entretanto, essa solução não é viável para o problema original, pois $t_P > T$.

Já para $\mu = 2$, o caminho mínimo no problema relaxado é $P = (1-3-2-5-6)$, com $c_P = 15$ e $t_P = 10$; o valor da função objetivo do problema relaxado é igual a 15. Note-se que esse caminho mínimo para o problema relaxado (quando $\mu = 2$) corresponde à solução ótima para o problema original de caminho mínimo restrito, uma vez que atende às condições enunciadas acima, ou seja (i) esse caminho é viável no problema original, uma vez que $(t_P \leq T)$ e (ii) $Z_D(\mu) = Z$.

Para ilustrar o método de subgradiente aplicado à relaxação Lagrangeana desse problema de caminho mínimo restrito, considere-se outro valor de tempo de viagem máximo, isto é, $T = 14$ ao invés do valor original ($T = 10$), uma vez que a solução ótima para $T = 10$ já foi obtida.

Inicialmente definem-se $\mu^0 = 0$, $\lambda_0 = 0.8$ e $UB = 24$ (que corresponde à solução viável 1-3-5-6). O escalar λ_k é dividido pela metade sempre que não ocorra melhoria na função objetivo $Z_D(\mu)$ do problema relaxado após 3 iterações consecutivas.

Na iteração inicial $k = 0$, a solução x^0 corresponde ao caminho $P = 1-2-4-6$, com a função objetivo do problema relaxado resultando $Z_D(0) = 3$. O termo do subgradiente $(Ax^0 - b)$ corresponde a $(t_P - 14) = 18 - 14 = 4$. Assim, para a primeira iteração calcula-se:

$$t_0 = \frac{0,8(24 - 3)}{4^2} = 1.05 \quad \Rightarrow \quad \mu^1 = [0 + 1.05(4)]^+ = 4.2$$

Para esse valor de multiplicador de Lagrange ($\mu^1 = 4.2$), o caminho mínimo resultante no problema relaxado é $P = 1-3-2-5-6$, com $Z_D(\mu=4,2) = 15 + 4.2(10) - 4.2(14) = -1.8$; o subgradiente $Ax^1 - b$ é $(t_P - 14) = 10 - 14 = -4$. Tendo em vista que 1-3-2-5-6 é um caminho viável para o problema original, e o custo $c_P = 15$ é menor que UB , o valor de UB é alterado para 15. Assim, calculam-se em seguida os valores:

$$t_1 = \frac{0,8(15 - (-1.8))}{(-4)^2} = 0.84 \quad \Rightarrow \quad \mu^2 = [4.2 + 0.84(-4)]^+ = 0.84$$

Na

Tabela 2.2 são mostrados os resultados de 33 iterações do método do subgradiente. Da iteração 2 à iteração 5 a solução de caminho mínimo alterna entre os caminhos (1-2-4-6) e (1-3-2-5-6). Ao final da quinta iteração, o valor λ_k é reduzido pela metade ($\lambda_5 = 0,4$), uma vez que não foi possível melhorar por três iterações consecutivas o melhor valor de função objetivo encontrado (=6.36). Nas sete iterações seguintes, os caminhos mínimos encontrados são (1-2-5-6), (1-3-5-6), (1-3-2-5-6), (1-3-2-5-6), (1-2-5-6), (1-3-5-6) e (1-3-2-5-6). Mais uma vez, após as três últimas iterações não houve melhoria no valor da função objetivo relaxada, então λ_k é novamente reduzido pela metade, ou seja, $\lambda_{12} = 0.2$. Desse ponto em diante, alternam-se os caminhos (1-3-2-5-6) e (1-2-5-6) como caminhos mínimos em cada iteração.

Como pode-se observar, o valor da função objetivo do problema relaxado está convergindo para o valor ótimo $Z_D^* = 7$, e o valor do multiplicador de Lagrange está convergindo para o seu valor ótimo de $\mu^* = 2$.

Esse ponto ótimo é ilustrado na Figura 2.8 e corresponde ao ponto de máximo da função $Z_D(\mu)$; nessa figura estão também representados os valores de custo penalizado $(c_P + \mu t_P - \mu T)$ para todos os possíveis caminhos entre os nós 1 e 6 em função do valor de μ .

Tabela 2.2: Otimização por subgradiente para o problema de caminho mínimo restrito

k	μ^k	t_{P-T}	$Z_D(\mu^k)$	λ_k	t_k
0	0.0000	4	3.0000	0.80000	1.0500
1	4.2000	-4	-1.8000	0.80000	0.8400
2	0.8400	4	6.3600	0.80000	0.4320
3	2.5680	-4	4.7280	0.80000	0.5136
4	0.5136	4	5.0544	0.80000	0.4973
5	2.5027	-4	4.9891	0.40000	0.2503
6	1.5016	1	6.5016	0.40000	3.3993
7	4.9010	-6	-5.4059	0.40000	0.2267
8	3.5406	-4	0.8376	0.40000	0.3541
9	2.1244	-4	6.5026	0.40000	0.2124
10	1.2746	1	6.2746	0.40000	3.4902
11	4.7648	-6	-4.5886	0.40000	0.2177
12	3.4589	-4	1.1646	0.20000	0.1729
13	2.7671	-4	3.9316	0.20000	0.1384
14	2.2137	-4	6.1453	0.20000	0.1107
15	1.7709	1	6.7709	0.20000	1.6458
16	3.4167	-4	1.3330	0.20000	0.1708
17	2.7334	-4	4.0664	0.20000	0.1367
18	2.1867	-4	6.2531	0.10000	0.0547
19	1.9680	1	6.9680	0.10000	0.8032
20	2.7712	-4	3.9150	0.10000	0.0693
21	2.4941	-4	5.0235	0.10000	0.0624
22	2.2447	-4	6.0212	0.05000	0.0281
23	2.1325	-4	6.4701	0.05000	0.0267
24	2.0258	-4	6.8966	0.05000	0.0253
25	1.9246	1	6.9246	0.00250	0.0202
26	1.9447	1	6.9447	0.00250	0.0201
27	1.9649	1	6.9649	0.00250	0.0201
28	1.9850	1	6.9850	0.00250	0.0200
29	2.0050	-4	6.9800	0.00250	0.0013
30	2.0000	-4	7.0000	0.00250	0.0012
31	1.9950	1	6.9950	0.00250	0.0200
32	2.0150	-4	6.9400	0.00250	0.0013
33	2.0100	-4	6.9601	0.00125	0.0006

Fonte: Ahuja *et al.* (1993)

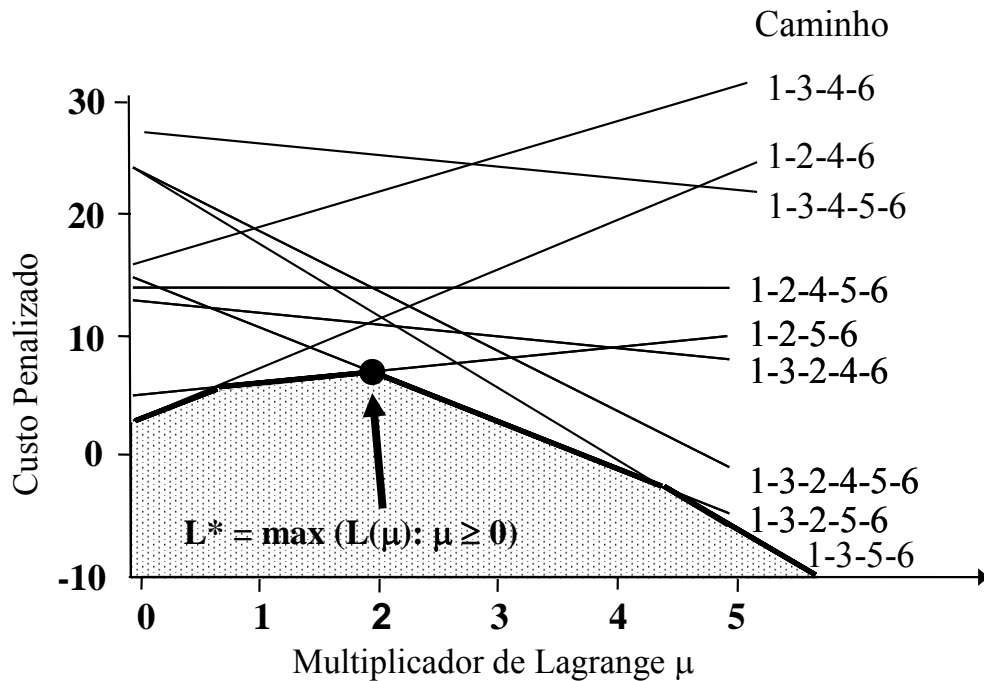


Figura 2.8: Função Lagrangeana $Z_D(\mu)$ para $T=14$

Fonte: adaptado de Ahuja *et al.* (1993)

Por exemplo, para $\mu = 1$ o caminho é 1-2-5-6, com comprimento $5 + \mu = 5 + 1 = 6$; para $\mu = 3$ o caminho mínimo é 1-3-5-6, com comprimento $24 - 6\mu = 24 - 6(3) = 6$; para $\mu = 2$ existem dois caminhos, 1-2-5-6 e 1-3-2-5-6, com comprimentos $5 + \mu = 5 + 2 = 7$ e $15 - 4\mu = 15 - 4(2) = 7$. Na Tabela 2.3 são listados todos os possíveis caminhos para o problema.

Tabela 2.3: Custos e tempos em todos os caminhos para o problema de caminho mínimo restrito

Caminho P	Custo no Caminho c_P	Tempo no Caminho t_P	Custo composto $c_P + \mu(t_P - T)$	Custo composto no caminho					
				$\mu=0$	$\mu=1$	$\mu=2$	$\mu=3$	$\mu=4$	$\mu=5$
1-2-4-6	3	18	$3 + 4\mu$	<u>3</u>	7	11	15	19	23
1-2-5-6	5	15	$5 + \mu$	5	<u>6</u>	<u>7</u>	8	9	10
1-2-4-5-6	14	14	14	14	14	14	14	14	14
1-3-2-4-6	13	13	$13 - \mu$	13	12	11	10	9	8
1-3-2-5-6	15	10	$15 - 4\mu$	15	11	<u>7</u>	<u>3</u>	<u>-1</u>	-5
1-3-2-4-5-6	24	9	$24 - 5\mu$	24	19	14	9	4	-1
1-3-4-6	16	17	$16 + 3\mu$	16	19	22	25	28	31
1-3-4-5-6	27	13	$27 - \mu$	27	26	25	24	23	22
1-3-5-6	24	8	$24 - 6\mu$	24	18	12	6	0	<u>-6</u>

Fonte: adaptado de Ahuja *et al.* (1993)

Note-se que nesse exemplo o valor da função objetivo para o problema relaxado $Z_D^* = 7$ é estritamente menor que o comprimento total do caminho mínimo restrito, que é de 13. Nesses casos, diz-se que a relaxação Lagrangeana tem um “*duality (ou relaxation) gap*”. Para encontrar a solução ótima de um problema com “*duality gap*”, é necessário aplicar algum método de enumeração (por exemplo, “*branch-and-bound*”), utilizando o limitante inferior da relaxação Lagrangeana a fim de reduzir o esforço de busca.

2.2.3 O Problema das Múltiplas Mochilas Binárias com K Restrições (K-MKP)

No problema das múltiplas mochilas binárias com K restrições (“*K-constraint multiple knapsack problem*”) existem n objetos e m mochilas. Cada objeto $j \in \{1, \dots, n\}$ tem associado um lucro p_j e K dimensões de peso w_{jk} , $k \in \{1, \dots, K\}$; cada mochila $i \in \{1, \dots, m\}$ possui K diferentes tipos de capacidade c_{ik} , $k \in \{1, \dots, K\}$ (ao invés de uma só capacidade, como no problema das múltiplas mochilas binárias). Busca-se selecionar quais objetos inserir em quais mochilas, de modo que o lucro total dos itens selecionados seja máximo e as K restrições de capacidade em cada uma das m mochilas sejam respeitadas.

Seja $x_{ij} = 1$ se o objeto j é inserido na mochila i e $x_{ij} = 0$ caso contrário. A formulação matemática do problema das múltiplas mochilas binárias com K restrições (K-MKP) pode ser escrita como (Ahuja e Cunha, 2005):

$$\text{maximizar } z = \sum_{i=1}^m \sum_{j=1}^n p_j x_{ij} \quad (2.34)$$

sujeito a

$$\sum_{j=1}^n w_{jk} x_{ij} \leq c_{ik} \quad i = 1, \dots, m \text{ e } k = 1, \dots, K \quad (2.35)$$

$$\sum_{i=1}^m x_{ij} \leq 1 \quad j = 1, \dots, n \quad (2.36)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, m, \quad j = 1, \dots, n \quad (2.37)$$

As K capacidades de cada uma das m mochilas são asseguradas pela restrição (2.34), enquanto a restrição (2.35) impõe que cada objeto seja inserido em, no máximo, uma mochila.

Esse problema corresponde a uma generalização do problema das múltiplas mochilas binárias (“*multiple knapsack problem*”), e foi estudado por Ahuja e Cunha (2005). Como o problema das múltiplas mochilas binárias é NP-difícil, a versão mais geral com K restrições de capacidade também apresenta ordem de complexidade exponencial. Para resolvê-lo, os autores propuseram uma estratégia baseada em busca em vizinhança de grande porte, que é detalhada no próximo capítulo (seção 3.5.6).

Para tanto, era necessário obter uma solução inicial viável. Duas diferentes abordagens foram testadas: uma através de uma heurística gulosa, em que os objetos foram ordenados segundo uma ordem decrescente de quociente entre lucro e peso total (p_j / w_j), este último considerando as K dimensões de peso, isto é $w_j = \sum_{k=1}^K w_{jk}$; analogamente, as mochilas foram ordenadas segundo uma ordem ascendente de capacidade total $c_i = \sum_{k=1}^K c_{ik}$.

Dessa forma, na heurística gulosa, os objetos (segundo a ordem definida) são inseridos nas mochilas em ordem. Assim, a primeira mochila $i = 1$ (na ordem dada) é preenchida seqüencialmente com todos os objetos j que couberem dentro dela (na ordem dada). Quando não for possível colocar mais nenhum objeto na mochila $i = 1$, então a próxima mochila $i = 2$ é enchida, considerando os objetos remanescentes (em ordem) que não tenham sido inseridos na mochila $i = 1$. O procedimento é repetido até que todas as m mochilas tenham sido totalmente preenchidas e não seja mais possível inserir nenhum objeto remanescente em nenhuma mochila.

A segunda heurística para a determinação de uma solução inicial baseia-se na relaxação Lagrangeana da restrição de capacidade (2.34) da formulação matemática do problema (2.33-2.36).

Dado um conjunto de multiplicadores não negativos μ_{ik} (um para cada restrição k de cada mochila i), a relaxação Lagrangeana do problema das múltiplas mochilas binárias com K restrições ($L(K\text{-MKP}, \mu)$) pode ser escrita como:

$$\text{maximizar } z(L(K\text{-MKP}, \mu)) = \sum_{i=1}^m \sum_{j=1}^n p_j x_{ij} + \sum_{i=1}^m \sum_{k=1}^K \mu_{ik} \left(c_{ik} - \sum_{j=1}^n w_{jk} x_{ij} \right) \quad (2.38)$$

sujeito às restrições (2.35) e (2.36).

Rearranjando-se os termos da função objetivo (2.37), tem-se:

$$z(L(K-MKP, \mu)) = \sum_{i=1}^m \sum_{j=1}^n \left(p_j - \sum_{k=1}^K \mu_{ik} w_{jk} \right) x_{ij} + \sum_{i=1}^m \sum_{k=1}^K \mu_{ik} c_{ik} \quad (2.39)$$

Considerando-se que o segundo termo da função objetivo (2.38) $\left(\sum_{i=1}^m \sum_{k=1}^K \mu_{ik} c_{ik} \right)$ é uma constante, para um dado conjunto de multiplicadores μ_{ik} a solução ótima do problema relaxado $L(K-MKP, \mu)$ pode ser facilmente determinada em tempo $O(n)$, da seguinte forma: cada objeto j é inserido na mochila i^* que corresponde ao máximo valor positivo de lucro modificado \tilde{p}_{ij} dentre todas as mochilas $i \in \{1, \dots, m\}$, onde \tilde{p}_{ij} é dado pela seguinte expressão: $\tilde{p}_{ij} = p_j - \sum_{k=1}^K \mu_{ik} c_{ik}$.

Em outras palavras, $\tilde{x}_{ij} = 1$ para $i = i^*$ tal que \tilde{p}_{ij} seja o maior valor positivo dentre todos os $i \in \{1, \dots, m\}$, e $\tilde{x}_{ij} = 0$ para as demais mochilas $i \in \{1, \dots, m\} \setminus \{i^*\}$. Caso todos $\tilde{p}_{ij} \leq 0$, então o objeto j não é inserido em nenhuma mochila, i.e. $\tilde{x}_{ij} = 0$ para todo $i \in \{1, \dots, m\}$.

Assim, uma solução inicial para o problema K-MKP pode ser obtida através da seguinte heurística baseada em relaxação Lagrangeana (K-MKP LR):

procedimento *K-MKP LR*;
início
 inicializar multiplicadores de Lagrange μ_{ik}^0 ;
 fazer $x^* = 0$;
enquanto critério de parada não atingido **faça**
 determinar a solução ótima do problema relaxado problem $L(K-MKP, \mu)$ usando μ_{ik}^t ;
 se $L(K-MKP, \mu)$ não for viável para o problema K-MKP **então**
 encontrar uma solução viável x para K-MKP baseada na solução $L(K-MKP, \mu)$;
 if o valor da função objetivo $Z(x) > Z(x^*)$ **então** faça $x^* = x$;
 atualizar os multiplicadores de Lagrange μ_{ik}^{t+1} para a iteração t ;
fim;
fim;

Figura 2.9: Heurística baseada em relaxação Lagrangeana para o K-MKP

Fonte: Ahuja e Cunha (2005)

Os multiplicadores de Lagrange são inicializados com os valores dos respectivos custos reduzidos das restrições de capacidade para a relaxação linear do problema K-MKP, obtidos através do CPLEX 8.1.

Para um dado conjunto de multiplicadores μ'_{ik} , se a solução ótima do problema relaxado $L(K\text{-MKP}, \mu)$ não for viável para o problema $K\text{-MKP}$, então uma solução viável é obtida da seguinte forma: objetos são sucessivamente removidos das mochilas, de n até 1 (com base na ordenação definida anteriormente) até que todas as restrições de capacidade tenham sido atendidas. Em seguida, a heurística gulosa de inserção é aplicada, considerando as capacidades residuais remanescentes das mochilas e os objetos não inseridos. Os multiplicadores de Lagrange são atualizados usando o método do subgradiente e considerando o valor ótimo da função objetivo da relaxação linear do $K\text{-MKP}$ como o limitante superior. Como critério de parada considerou-se um número máximo de iterações (100), ou 20 iterações consecutivas sem melhoria da solução relaxada.

Extensivos experimentos computacionais foram realizados, cujos resultados indicaram que a relaxação Lagrangeana permite obter de maneira consistente melhores soluções iniciais que a heurística gulosa, sendo isso importante para a obtenção de melhores resultados usando o algoritmo de busca em vizinhança de grande porte sem reinício de soluções. Maiores detalhes sobre a busca em vizinhança de grande porte são encontrados na Seção 3.5.6, e sobre o problema como um todo e a estratégia geral de solução em Ahuja e Cunha (2005).

Uma outra possibilidade não explorada seria utilizar essas soluções obtidas através de relaxação Lagrangeana como limitantes inferiores para um algoritmo de enumeração implícita.

2.2.4 Relaxação Lagrangeana Aplicada ao Problema de Roteirização de Veículos

Cunha (1997) tratou do problema de roteirização de veículos com restrições de janela de tempo, duração máxima da jornada, e frota heterogênea com NV veículos. O problema, denominado (PRPJT), pode ser formulado como:

$$\text{minimizar } \sum_{v=1}^{NV} \sum_{i=0}^N \sum_{j=1}^{N+1} C_d^v d_{ij} x_{ij}^v + \sum_{v=1}^{NV} \sum_{j=1}^{N+1} (T_i + s_i + t_{i,N+1}) C_h^v x_{i,N+1}^v + \sum_{v=1}^{NV} \sum_{j=1}^N C_f^v x_{oj}^v \quad (2.40)$$

sujeito a:

$$\sum_{v=1}^{NV} \sum_{j=1}^{N+1} x_{ij}^v = 1 \quad i = 0, 1, 2, \dots, N; i \neq j \quad (2.41)$$

$$\sum_{j=1}^{N+1} x_{0j}^v = 1 \quad v = 1, 2, 3, \dots, NV \quad (2.42)$$

$$\sum_{i=0}^N x_{ij}^v - \sum_{i=1}^{N+1} x_{ji}^v = 0 \quad j = 1, 2, 3, \dots, N; \quad v = 1, 2, 3, \dots, NV \quad (2.43)$$

$$\sum_{i=0}^{N+1} x_{i,N+1}^v = 1 \quad v = 1, 2, 3, \dots, NV \quad (2.44)$$

$$a_i \leq T_i \leq b_i \quad i = 1, 2, 3, \dots, N \quad (2.45)$$

$$T_i + s_i + t_{ij} - T_j \leq (1 - x_{ij}^v)M \quad i = 1, \dots, N \quad j = 1, \dots, N \quad j \neq i \quad v = 1, 2, 3, \dots, NV \quad (2.46)$$

$$\sum_{i=0}^N \sum_{j=1}^N q_j x_{ij}^v \leq K^v \quad v = 1, 2, 3, \dots, NV \quad (2.47)$$

$$(T_j + s_j + t_{j,N+1})x_{j,N+1}^v - (T_i - t_{0i})x_{0i}^v \leq H^v \quad i = 1, 2, 3, \dots, N \quad ; \quad j = 1, 2, 3, \dots, N \quad (2.48)$$

$$x_{ij}^v \in \{0,1\} \quad i = 0, 1, 2, 3, \dots, N+1; \quad j = 0, 1, 2, 3, \dots, N+1; \quad v = 1, 2, 3, \dots, NV \quad (2.49)$$

A função objetivo (2.40) representa o custo total a ser minimizado. As restrições (2.40) impõem que cada um dos pontos seja visitado uma única vez e por um único veículo; já as restrições (2.41) a (2.43) descrevem o fluxo no caminho que o veículo v utilizar. Caso o veículo não seja utilizado, ele segue o caminho do arco direto ligando a base de partida (nó 0) à base de chegada (nó $N+1$). As restrições (2.44) impõem que o horário de início de atendimento de cada nó ocorra dentro da sua respectiva janela de tempo; já as restrições (2.45) definem a continuidade e a compatibilidade temporal dos horários de início de atendimento ao longo das rotas. As restrições de capacidade de carga e de duração máxima da jornada dos veículos são dadas pelas restrições (2.46) e (2.47), respectivamente. Por fim, as restrições (2.48) asseguram a integralidade da solução.

Foram propostas estratégias de solução heurísticas baseadas na relaxação Lagrangeana das restrições relacionadas à obrigatoriedade de atendimento de todos os pontos uma única vez e por um único veículo (Cunha, 1997). Mais especificamente, relaxando-se as restrições (2.40), o problema resultante pode ser visto como um problema particular de fluxo em rede, em que NV unidades de fluxo devem seguir do nó que corresponde à base de origem (0) ao nó da base de destino ($N+1$).

Em particular, se $NV = 1$, o problema resultante corresponde a um problema de caminho mínimo com restrições adicionais de janela de tempo, de duração da jornada e de viabilidade da capacidade do veículo ao longo do roteiro. Para este caso particular existe um algoritmo de

etiquetamento permanente (AEP), baseado em programação dinâmica, bastante eficiente (Cunha e Swait, 2000; Desrochers e Soumis, 1988).

Uma estratégia de solução baseada no método da relaxação Lagrangeana pressupõe encontrar um algoritmo bastante eficiente para a solução do problema relaxado. Tendo em vista a dificuldade remanescente do problema relaxado para frota superior a um veículo, a estratégia geral de solução do PRPJT é heurística, e utiliza, como sub-rotina, o AEP para o problema de caminho mínimo com restrições de janela de tempo, duração da jornada e capacidade dos veículos.

Para instâncias do PRPJT em que a frota é homogênea são propostas duas heurísticas: a Heurística de Alocação Seqüencial e a Heurística de Alocação Paralela. Ambas procuram tirar proveito da característica do problema em que não há decisão quanto ao tipo/tamanho de veículo mais adequado/econômico a cada roteiro.

A heurística de *alocação seqüencial* consiste em utilizar, de maneira seqüencial, o algoritmo para a resolução do subproblema de caminho mínimo com janela de tempo. A estratégia básica desta heurística é considerar todos os clientes como candidatos a serem inseridos em um mesmo veículo.

Após um certo número de iterações, caso não tenha sido possível inserir todos os clientes no mesmo veículo (em decorrência das restrições de janela de tempo, de duração da jornada e de capacidade), mantém-se a melhor solução para o veículo. Os clientes atendidos pelo veículo são excluídos da lista de candidatos a serem programados nos veículos subseqüentes. Em seguida, o procedimento é novamente repetido, considerando apenas os clientes remanescentes, isto é, ainda não atendidos, e assim sucessivamente. Como a frota é homogênea, ou seja, composta de veículos idênticos não é necessário definir uma ordem de utilização dos mesmos, o que permite esta alocação seqüencial dos clientes aos veículos.

A *heurística de alocação paralela* visa a determinar uma programação que possibilite a utilização da frota total disponível, com uma distribuição mais uniforme dos clientes nos roteiros e, ao mesmo tempo, a minimização das parcelas de custo com a distância e/ou com o tempo de viagem. A principal diferença desta heurística com relação à anterior reside na maneira como são processadas as iterações da relaxação Lagrangeana. Ao invés de tentar inserir todos os clientes não atendidos num mesmo veículo, neste caso a construção dos

roteiros dos NV veículos é realizada em paralelo. Assim, em uma iteração da relaxação Lagrangeana são obtidos NV caminhos mínimos com penalidades.

Para instâncias que correspondem ao caso mais geral do PRPJT, em que a frota pode ser heterogênea, foi proposta a heurística de *agrupamento e alocação seqüencial*. Esta heurística é derivada da Heurística de Alocação Seqüencial, para problemas com frota homogênea. A diferença básica é a introdução de um procedimento de agrupamento de clientes. Ao invés de se tentar inserir todos os clientes ainda não atendidos em um mesmo veículo, antes é realizado um agrupamento e uma seleção dos clientes que devem estar potencialmente juntos, para então se tentar determinar um roteiro que passe por todos os pontos agrupados. O procedimento da relaxação Lagrangeana é utilizado para resolver instâncias de problemas cujos nós devem potencialmente poder ser atendidos pelo veículo.

Os clientes vão sendo agrupados, baseados em um critério de *proximidade*, que leva em conta a distância entre eles e também a proximidade “temporal”, isto é, definida pelas janelas de tempo. O agrupamento é interrompido quando as restrições de capacidade do veículo ou de duração da jornada tenham sido violadas. Adicionalmente, o procedimento de agrupamento visa racionalizar a alocação dos clientes aos veículos e também reduzir o número de nós candidatos a serem inseridos e roteirizados em um veículo, e não determinar exatamente quais os clientes a serem atendidos pelo veículo.

Tendo em vista a complexidade introduzida pelas janelas de tempo, que dificulta a adoção de métodos baseados em critérios espaciais de varredura, bem como a distribuição mais uniforme dos clientes aos veículos, o procedimento de agrupamento baseou-se no problema da árvore de cobertura mínima (“*minimum spanning tree*”) (Cunha, 1997). Uma árvore de cobertura consiste em um conjunto de nós e arcos tais que não pode haver ciclos; em outras palavras, saindo de qualquer nó i pertencente à árvore e percorrendo os arcos incluídos na árvore, é impossível retornar ao nó i . Para a decisão quanto ao tipo de veículo que é utilizado em cada agrupamento foi desenvolvido um procedimento que define a ordem de prioridade na utilização dos veículos da frota segundo uma ordem crescente de custo fixo por unidade de capacidade.

Maiores detalhes sobre a relaxação Lagrangeana aplicada a esse problema, bem como os resultados obtidos, podem ser encontrados em Cunha (1997). Ahuja *et al.* (1993) discutem

diferentes relaxações Lagrangeanas para o problema de roteirização de veículos com frota homogênea e sem restrições de janela de tempo.

2.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram apresentados os principais aspectos do algoritmo “*branch-and-bound*”, utilizado para a resolução de problemas de programação linear inteira, bem como os avanços mais recentes, que permitem resolver problemas de porte muito maior com tempos de processamento reduzidos.

Foram também apresentados os principais conceitos da relaxação Lagrangeana, uma técnica que permite não só obter bons limitantes para o B&B (no lugar das relaxações lineares), como também ser usada em procedimentos heurísticos.

Além da relaxação Lagrangeana, pretende-se explorar com mais detalhes em futuras pesquisas a técnica de geração de colunas. Uma outra abordagem interessante para problemas com estrutura em colunas é atribuir valores específicos a algumas variáveis, tornando o problema resultante fácil de resolver. Mais especificamente, no algoritmo de decomposição de Benders, fixam-se os valores de algumas variáveis e resolve-se o problema resultante iterativamente (Benders, 1962).

Apesar do progresso observado principalmente ao longo dos últimos 10-15 anos, resolver inúmeros problemas de programação inteira permanece um desafio, já que mesmo os algoritmos mais modernos ainda não conseguem obter boas soluções em tempo de processamento razoáveis para muitos problemas, principalmente quando (i) as soluções relaxadas na parte superior da árvore são ruins como limitantes, (ii) melhoram muito vagarosamente, ou (iii) há muitas soluções viáveis semelhantes, como ilustrado mais adiante, nos problemas reais descritos a partir do capítulo 4, muitos deles de natureza combinatória e, conseqüentemente, ordem de complexidade não polinomial.

Nesses casos, uma estratégia exata implica esforço similar a percorrer todo o espaço de solução, o que pode ser proibitivo, levando à necessidade do uso de heurísticas para a obtenção de boas soluções em tempos de processamento reduzidos.

Nesse contexto, destacam-se as chamadas metaheurísticas, que correspondem a processos inteligente e gerais, que guiam as heurísticas, dada a sua importância no contexto atual para a solução de problemas reais de natureza combinatória. As principais metaheurísticas são objeto do próximo capítulo.

Capítulo

3

ESTRATÉGIAS DE SOLUÇÃO HEURÍSTICAS

Conforme visto no capítulo anterior, muitos dos problemas encontrados nas áreas de logística e transporte são de programação linear inteira e do tipo NP-difícil (do inglês “NP-hard”), o que significa que possuem ordem de complexidade exponencial. Em outras palavras, o esforço computacional para a sua resolução cresce exponencialmente com o tamanho do problema e, portanto, os algoritmos “*branch-and-bound*”, apesar de todos os avanços, não permitem obter soluções ótimas, mesmo em tempos de processamento elevados.

A complexidade matemática desses problemas de natureza combinatória, para os quais não é possível determinar a solução ótima para instâncias reais, tem levado a um significativo e continuado interesse e esforço de pesquisa em busca de novas estratégias de solução. Em muitos casos, pequenos ganhos podem representar economias expressivas.

Por outro lado, em termos práticos, tal complexidade acarreta que os métodos de solução desenvolvidos se baseiem em procedimentos heurísticos, isto é, que não asseguram a obtenção de soluções ótimas do ponto de vista matemático. O termo *heurística* usualmente refere-se a um procedimento de busca iterativo, baseado em alguma estrutura de controle repetitivo.

Estratégias de solução heurísticas apóiam-se, em geral, em alguma abordagem intuitiva, na qual a estrutura particular do problema pode ser considerada e explorada de forma inteligente, a fim de se obter uma solução satisfatória, e considerando-se o compromisso qualidade *versus* esforço computacional para obtê-la (Silver, 2004). Conseqüentemente, em muitos casos, procedimentos heurísticos são específicos de cada contexto, carecendo de robustez. Em outras palavras, geralmente não produzem boas soluções para problemas similares, porém com

características, condicionantes ou restrições às vezes um pouco diferentes daquelas para as quais foram desenvolvidos. Isso significa que, para muitos problemas, uma heurística de solução aplicável a um determinado tipo de problema e respectivos dados pode não produzir resultados adequados para outro problema similar, nem tampouco apresentar desempenho computacional satisfatório (Teixeira e Cunha, 2002).

De acordo com Silver (2004), heurísticas são utilizadas para resolver modelos matemáticos tendo em vista:

- a sua facilidade de implementação;
- por permitirem obter melhorias em relação a práticas correntes, geralmente manuais ou muito simplificadas;
- por fornecerem rapidamente resultados de boa qualidade;
- por proporcionarem maior robustez em comparação a soluções de modelos matemáticos no tocante a variações nos dados de entrada;
- a sua utilidade quando utilizadas em conjunto com algoritmos de otimização, permitindo obter soluções iniciais de boa qualidade, provendo limitantes que permitem eliminar partes substanciais do espaço de soluções, e guiando o processo de busca.

No passado recente, em particular nos último dez a quinze anos, um importante e crescente esforço de pesquisa vem sendo direcionado à investigação e ao desenvolvimento (e também à aplicação a problemas reais) das chamadas metaheurísticas.

O termo *metaheurística*, introduzido pela primeira vez por Glover (1986), é a união de duas palavras gregas: *heurística*, derivada do verbo *heuriskein* (*εὕρισκειν*), que significa “encontrar”, e o sufixo *meta*, que significa “além, acima” (no sentido de superior). A título de curiosidade, heurística possui a mesma raiz da palavra grega “eureka”. Assim, as metaheurísticas podem ser definidas como as estratégias e técnicas mais recentes e avançadas, que guiam outras heurísticas a fim de se encontrar soluções melhores, ultrapassando o ponto de parada das heurísticas tradicionais. De acordo com Reeves (1993), antes que o termo ficasse consagrado, as metaheurísticas eram chamadas de heurísticas modernas.

Entre as metaheurísticas mais conhecidas e amplamente utilizadas com sucesso pode-se citar, entre outras, algoritmos genéticos, busca tabu, têmpera simulada (*simulated annealing*), busca

em vizinhança de grande porte (do inglês “*Very Large-Scale Neighborhood Search*”), GRASP (“*Greedy Randomized Adaptive Search Procedure*”) e “*scatter search*”.

Segundo Osman (2002), uma metaheurística pode ser definida como um procedimento mestre iterativo que guia e modifica a seqüência de operações de heurísticas subordinadas, a fim de produzir soluções de alta qualidade de maneira eficiente. Pode combinar inteligentemente diferentes idéias para explorar o espaço de solução usando estratégias adaptativas com aprendizado durante o processo. Busca-se evitar ficar preso em um ótimo local (para problemas que apresentam múltiplos ótimos locais) e/ou reduzir o espaço de busca, de maneira inteligente.

Glover e Kochenbeger (2003) definem metaheurísticas como métodos de solução que orquestram uma interação entre procedimentos de busca local e estratégias de nível superior para criar um processo capaz de escapar de ótimos locais e executar uma busca robusta de um espaço de soluções. Ao longo do tempo, esses métodos passaram a incluir quaisquer procedimentos que empregam estratégias para superar a armadilha de ótimos locais em espaços de solução complexos, particularmente aqueles utilizam uma ou mais estruturas de vizinhança como um meio de definir movimentos admissíveis para se deslocar de uma solução para outra, ou criar e destruir soluções em processos construtivos e destrutivos.

De acordo com Glover e Kochenbeger (2003), o grau em que tais estruturas são investigadas varia de acordo com o tipo de procedimento. No caso de certas heurísticas baseadas em população, tais como os algoritmos genéticos, as vizinhanças são implicitamente representadas, sendo exploradas através da substituição de componentes de uma solução por outros de outra solução, utilizando regras de troca popularmente batizadas de “cruzamento”. Por outro lado, em metaheurísticas que não manipulam múltiplas soluções simultaneamente, o espaço de soluções é manipulado e explorado de diversas maneiras, deterministicamente ou randomicamente, dependendo de fatores como a fase da busca ou (no caso de métodos baseados em memória) o histórico do processo de solução.

Genericamente, a idéia básica por trás das metaheurísticas é explorar, de maneira mais inteligente, as regiões mais promissoras do espaço de soluções, indo além do ponto em que as heurísticas geralmente param, a fim de buscar resultados melhores, em termos de qualidade da solução, com desempenho computacional adequado (Reeves, 1996).

No entanto, segundo Laporte *et al.* (2000), embora a qualidade das soluções obtidas através de metaheurísticas seja, em geral, superior a das heurísticas convencionais, os tempos computacionais ainda são, em muitos casos, muito elevados, o que dificulta a sua aplicação e incorporação em sistemas comerciais aplicáveis em problemas reais. Adicionalmente, de acordo com os mesmos autores, as metaheurísticas são muito dependentes do contexto de cada problema e requerem ajuste minucioso e preciso de inúmeros parâmetros caso a caso, o que também inviabiliza, por enquanto, a sua utilização em caráter mais abrangente em muitos sistemas computacionais destinados a usuários não especializados. Ainda segundo os autores, um dos desafios futuros que deve direcionar as pesquisas nessa área é o desenvolvimento de metaheurísticas mais simples, mais rápidas e mais robustas, mesmo com alguma diminuição da qualidade das soluções obtidas, mas que permitam a sua aplicação em problemas reais.

Nas próximas seções são descritas as principais metaheurísticas utilizadas para resolver problemas de otimização combinatória em logística e transportes: algoritmos genéticos, busca tabu, GRASP, “*simulated annealing*”, e busca em vizinhança de grande porte, com ênfase para aspectos da sua implementação computacional, fundamentados na prática deste autor.

3.1 ALGORITMOS GENÉTICOS

De acordo com Reeves (2003), o termo *Algoritmo Genético* (do inglês “*Genetic Algorithm*”), geralmente abreviado com AG (ou GA), é uma meta-heurística proposta por John Holland (1975), cujo livro intitulado *Adaptation in Natural and Artificial Systems* foi fundamental para o desenvolvimento de um campo de pesquisa e aplicação que vai muito além das idéias originalmente apresentadas. Muitos adotam os termos “algoritmo evolucionário” (“*evolutionary algorithm*”) e “computação evolucionária” (“*evolutionary computing*”), a fim de cobrir os desenvolvimentos ocorridos na última década. Segundo Reeves (2003), a otimização ocupou um pequeno espaço na obra de Holland, que tratou de sistemas adaptativos, embora a maioria da literatura sobre algoritmos genéticos tenda a assumir que essa foi a sua principal intenção.

Algoritmos genéticos inspiram-se na teoria de evolução darwiniana, que considera que quanto melhor um indivíduo se adapta ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes.

Os algoritmos genéticos fazem parte da classe de algoritmos denominados *algoritmos populacionais*. Segundo Beasley (1999), pertencem a uma classe geral de algoritmos de busca probabilística “inteligente”, que se baseiam no processo evolucionário de organismos biológicos na natureza, ou seja, no tratamento de uma população de soluções, e na combinação destas entre si, de tal forma a gerar novas soluções. Em linhas gerais, algoritmos genéticos trabalham com um grupo (ou população) de soluções. Cada indivíduo é representado por um cromossomo, que armazena as informações genéticas. Os cromossomos são compostos por genes, os quais são responsáveis pelas características dos seres e são trocados ou transmitidos durante o processo de reprodução. A cada iteração, indivíduos da população são selecionados para reprodução, sendo os mais aptos transmitidos para a geração seguinte e os demais menos aptos simplesmente eliminados de acordo com o princípio darwiniano de seleção natural e sobrevivência do mais forte. Ainda segundo Beasley (1999), algoritmos populacionais produzem, de um modo geral, resultados melhores para problemas de natureza combinatória quando comparados com heurísticas baseadas na melhoria de uma única solução.

Dessa forma, algoritmos genéticos compreendem um ferramental que tenta reproduzir matematicamente o processo genético de seleção natural, evolução de populações e sobrevivência dos mais aptos com a finalidade de buscar uma boa solução para um problema de otimização. É uma heurística iterativa, em que uma população inicial de soluções vai evoluindo de forma iterativa, através da combinação de elementos e da sobrevivência dos melhores elementos, até que algum critério de parada seja atingido.

A Figura 3.1 ilustra o esquema genérico de funcionamento do Algoritmo Genético. Uma iteração qualquer de um algoritmo genético pode ser descrita da seguinte forma: dada uma população de soluções $P(t)$ de uma iteração t , cada solução é avaliada a fim de se obter alguma medida da sua qualidade (“*fitness*”). Uma nova população $P(t+1)$ correspondente à iteração $t+1$ é então gerada, em duas etapas. Inicialmente indivíduos de $P(t)$ são selecionados para agir como pais que vão produzir a geração seguinte $P(t+1)$. Em seguida, os pais passam por um cruzamento (acasalamento) para produzir os filhos que formarão a nova população $P(t+1)$. Uma mutação em algum indivíduo também pode ser realizada com uma certa probabilidade, geralmente muito baixa, com a finalidade de gerar diversidade ou variabilidade de características genéticas.

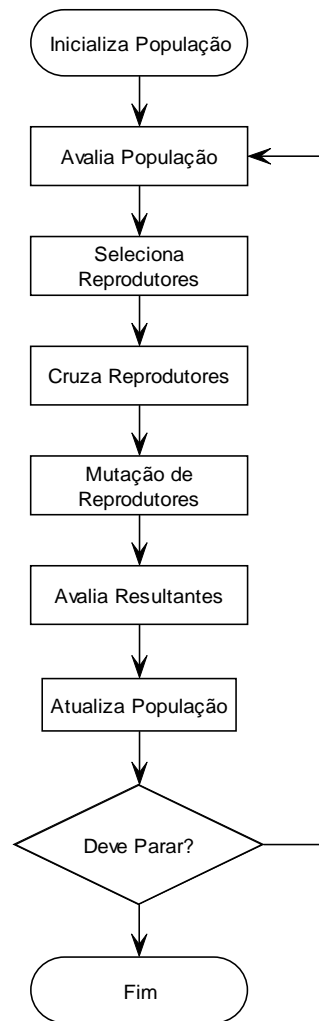


Figura 3.1: Esquema geral de funcionamento do Algoritmo Genético

O desenvolvimento de uma estratégia baseada em algoritmo genético envolve os seguintes componentes:

- codificação de indivíduos de forma a representar soluções do problema;
- geração da população inicial;
- função de avaliação da aptidão (“*fitness*”) de cada indivíduo da população;
- seleção de indivíduos para reprodução;
- operador genético de reprodução (ou cruzamento);
- operador genético de mutação;
- atualização da população.

Boas referências sobre Algoritmos Genéticos podem ser encontradas em Goldberg (1989), Davis (1991), Michalewicz (1994), Gen e Cheng (1997), Beasley (1999), entre outros.

3.1.1 Codificação de Indivíduos

Uma possível solução para um problema pode ser representada através de um vetor de genes que forma um cromossomo, ou seja, um indivíduo da população. Em um problema com n variáveis de decisão binárias, uma representação natural corresponde a um cromossomo com n genes do tipo 0-1. Tome-se, por exemplo, um problema da mochila binária (Martello e Toth, 1990), em que se deve selecionar um subconjunto de $n=10$ elementos de forma a maximizar o lucro dos objetos selecionados, e respeitando-se a restrição de capacidade da mochila. Uma possível solução, em que seriam selecionados os objetos 1, 3, 4, 7, 8 e 10, seria representada através do seguinte cromossomo:

1	2	3	4	5	6	7	8	9	10
1	0	1	1	0	0	1	1	0	1

Para o problema das múltiplas mochilas binárias (Martello e Toth, 1990), em que se têm m mochilas onde podem ser inseridos os objetos, cada gene pode assumir um valor no intervalo $[0, m]$, onde 0 indica que o objeto não foi selecionado e algum valor entre 1 e m indica a mochila em que o objeto foi inserido. Assim, para o problema com $n=10$ elementos e $m=3$ mochilas uma possível solução pode ser representada como:

1	2	3	4	5	6	7	8	9	10
3	1	2	0	0	2	1	0	3	1

ou seja, o objeto 1 é inserido na mochila 3, o objeto 2 na mochila 1, o objeto 3 na mochila 2, o objeto 4 não é inserido em nenhuma mochila, e assim por diante.

Para um problema de caixeiro viajante com $n=10$ cidades, numeradas de 1 a 10, uma representação possível corresponde a uma seqüência viável de cidades visitadas:

5	3	2	1	7	8	4	10	9	6
----------	----------	----------	----------	----------	----------	----------	-----------	----------	----------

que indica a seqüência de cidades visitadas: 5-3-2-1-7-8-4-10-9-6-5.

3.1.2 Geração da População Inicial

A população inicial pode ser simplesmente gerada de forma aleatória; entretanto, recomenda-se incluir pelo menos uma solução obtida através de uma boa heurística (construtiva). Outra possibilidade é assegurar que a população inicial esteja bem distribuída ao longo do espaço de soluções.

Pode-se pensar ainda na combinação de soluções obtidas através de heurísticas e soluções geradas aleatoriamente, embora isso possa levar a um aumento do risco de convergência prematura do algoritmo pela eventual pouca diversidade da população. Outra alternativa consiste em gerar um número muito maior de indivíduos que o tamanho da população e selecionar os melhores.

Por exemplo, para a geração de cada indivíduo da população inicial para o problema da mochila binária, George e Cunha (2002) inicialmente consideraram varrer seqüencialmente a lista de objetos ordenada segundo a ordem decrescente de taxa (lucro/peso) e ir decidindo aleatoriamente se cada um dos objetos devia ser inserido ou não na mochila (com uma probabilidade equivalente ao quociente entre a capacidade da mochila e o peso total de todos os objetos), sempre iniciando a partir do melhor objeto, até que a capacidade da mochila tivesse sido atingida. Entretanto, percebeu-se que a população não apresentava suficiente diversidade, e o algoritmo genético convergia prematuramente para um ótimo local. A fim de gerar indivíduos com diversidade suficiente, os autores passaram a iniciar não sempre do primeiro objeto, mas de um objeto sorteado aleatoriamente, o que permitiu gerar uma população com suficiente diversidade.

3.1.3 Função de Avaliação da Aptidão

A função de avaliação da aptidão (“*fitness*”) dos indivíduos tem por objetivo medir (ou avaliar) a aptidão ou qualidade de cada um dos indivíduos que compõem a população. Essa medida é utilizada para a seleção dos melhores indivíduos que sobrevivem em cada geração, bem como para a seleção dos indivíduos para cruzamento. Em outras palavras, os indivíduos com melhor aptidão possuem maiores chances de serem repassados para as próximas gerações por meio de reprodução ou elitismo

A função aptidão é composta pela função objetivo do problema, mais uma função penalidade para os indivíduos não viáveis. Por exemplo, no caso do problema da mochila binária, George e Cunha (2002) consideraram uma redução do lucro total como penalidade para os indivíduos que correspondiam a soluções que violavam a capacidade da mochila. Essa penalidade era proporcional à violação (i.e., diferença positiva entre o peso total e a capacidade da mochila), considerando-se uma redução do lucro por unidade de capacidade violada; foram considerados dois valores para essa penalidade: relação (lucro/peso) do melhor objeto (acarretando uma penalidade maior), e (lucro/peso) médio do conjunto de n objetos candidatos a serem inseridos na mochila.

3.1.4 Seleção de Indivíduos para Reprodução

Existem diversos métodos para seleção dos indivíduos para cruzamento (ou reprodução). Todos levam em consideração a aptidão de cada um dos N indivíduos da população. Os principais deles são:

- torneio (“*tournament*”): sorteiam-se aleatoriamente k indivíduos e seleciona-se o melhor, com base na função de aptidão. Muitas vezes a competição é realizada entre dois indivíduos, porém uma generalização é possível para qualquer tamanho arbitrário do grupo k . Esse critério de seleção pode ser implementado de maneira eficiente, sendo muito rápido, com ordem de complexidade $O(N)$, uma vez que não requer nenhuma ordenação prévia da população.
- seleção truncada: a seleção truncada consiste em definir somente uma fração T de melhores indivíduos que podem ser selecionados, e todos eles possuem a mesma probabilidade de seleção. Como é necessária uma ordenação prévia da população, este método possui ordem de complexidade $O(N \log N)$.
- roleta (“*roulette wheel*”): recebe essa denominação em função da sua analogia com uma roleta, ou seja, cada indivíduo é representado por uma seção da roleta. Os indivíduos são selecionados para cruzamento segundo uma probabilidade proporcional à sua aptidão; a seção ocupada pelo indivíduo na roleta é proporcional à sua aptidão. Roda-se o ponteiro e seleciona-se o indivíduo apontado pelo mesmo. Em termos práticos, a roleta pode ser vista como um segmento. Os indivíduos da população são

distribuídos em segmentos contíguos, sendo o tamanho (ou comprimento) de cada segmento calculado em função da aptidão do respectivo indivíduo, conforme mostrado na Figura 3.2 para um problema de maximização.

Indivíduo	1	2	3	4	5	6	7	8	9	10
Aptidão	2,0	1,8	1,6	1,4	1,2	1,0	0,8	0,6	0,4	0,2
Probabilidade de seleção	0,18	0,16	0,15	0,13	0,11	0,09	0,07	0,06	0,03	0,02

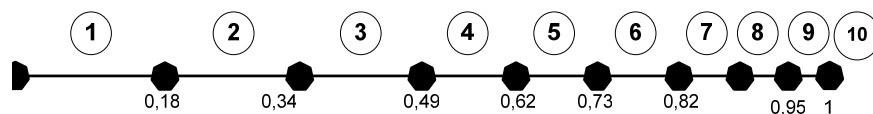


Figura 3.2: Seleção por Roleta

Assim, a probabilidade de escolha do indivíduo 1 (0,18) é calculada através do quociente entre a sua aptidão (2,0) e a soma de todas as aptidões (2,0 + 1,8 + ... + 0,4 + 0,2). Se a função objetivo fosse de minimização, o melhor indivíduo seria o 10; assim, tomar-se-iam os inversos dos valores das aptidões de forma a privilegiar o menor valor e penalizar o maior valor.

- *seleção por ranking linear*: consiste em inicialmente ordenar os indivíduos de acordo com a sua aptidão, sendo que o melhor indivíduo encontra-se na N -ésima posição, isto é, $r(1) = N$, enquanto que o pior indivíduo está na primeira posição $r(N)=1$. A probabilidade de cada indivíduo é determinada pelo quociente entre a sua posição no ranking e a soma de todas as posições; assim, o melhor indivíduo da população de tamanho N tem probabilidade $N/\sum_{i=1}^N r(i)$ de ser escolhido, o pior indivíduo tem probabilidade $1/\sum_{i=1}^N r(i)$ de ser escolhido e os demais $r(i)/\sum_{i=1}^N r(i)$, conforme ilustrado na Figura 3.3, em que $\sum_{i=1}^{10} r(i) = 55$. É importante salientar que, mesmo indivíduos com o mesmo valor de aptidão, possuirão diferentes valores de probabilidade de seleção, dado que estes possuem diferentes posições no ranking. Este método possui tempo de processamento da ordem $O(N \log N)$. A principal virtude desse método em relação à roleta é evitar que um indivíduo muito superior aos demais (em termos de aptidão) seja excessivamente utilizado para cruzamento, levando a uma

população com pouca diversidade e, conseqüentemente, a uma convergência prematura do algoritmo genético.

Indivíduo	1	2	3	4	5	6	7	8	9	10
Aptidão	2,0	1,8	1,6	1,4	1,2	1,0	0,8	0,6	0,4	0,2
Posição	10	9	8	7	6	5	4	3	2	1
Probabilidade de seleção	10/55 0.18	9/55 0.16	8/55 0.15	7/55 0.13	6/55 0.11	5/55 0.09	4/55 0.07	3/55 0.05	2/55 0.04	1/55 0.02

Figura 3.3: Seleção por Ranking Linear

- seleção por ranking exponencial: a seleção por ranking exponencial difere da seleção por ranking linear na medida em que a probabilidade de um indivíduo "rankeado" é ponderada exponencialmente. A base do expoente é o parâmetro c do método ($0 < c < 1$). Quanto mais próximo c estiver do valor 1, mais baixa é a "exponencialidade" do método de seleção. Este método possui também ordem de complexidade $O(M \log N)$, e é semelhante à seleção por ranking linear, exceto pela forma de cálculo da probabilidade de seleção.
- seleção por amostragem universal estocástica ("stochastic universal sampling"): semelhante ao método da roleta. Para selecionar k indivíduos, utilizam-se k ponteiros igualmente espaçados, girando-os em conjunto uma só vez, e selecionando-se os indivíduos marcados pelos k ponteiros, como ilustrado na Figura 3.4, considerando-se os mesmos indivíduos e probabilidades da Figura 3.3.

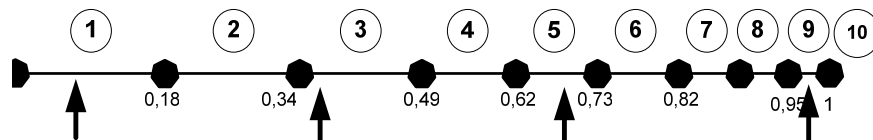


Figura 3.4: Seleção por Amostragem Universal Estocástica

3.1.5 Operador Genético de Cruzamento

O operador de cruzamento ou "*crossover*", também chamado de reprodução, tem por finalidade combinar as informações genéticas de dois indivíduos (pais) para gerar novos

indivíduos (filhos). É o principal operador de algoritmo genético aplicado aos indivíduos de uma população, a fim de gerar os novos indivíduos diferentes (melhores ou piores) que formam a descendência.

O método mais comum para combinar dois indivíduos é o *cruzamento em um único ponto*. Uma posição para cruzamento k do vetor indivíduo é selecionada aleatoriamente e os bits de cada indivíduo, após este ponto k , são trocados, produzindo dois novos descendentes, conforme mostrado na Figura 3.5, para o ponto de cruzamento $k = 4$.

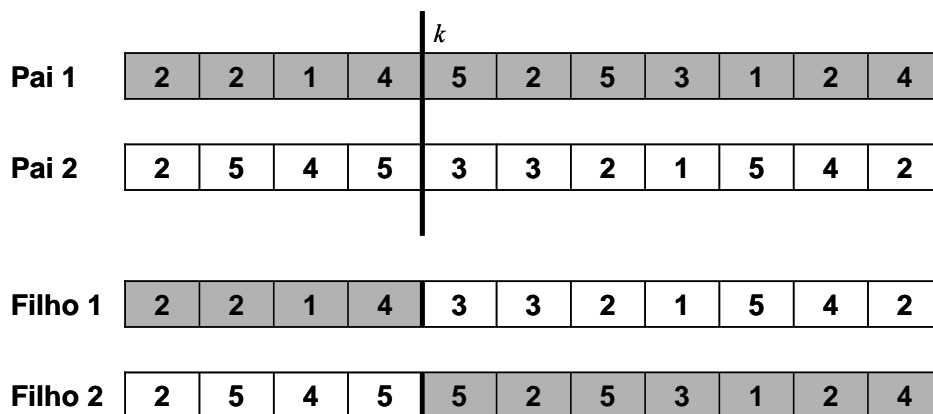


Figura 3.5: Cruzamento em um único ponto

Já no *cruzamento em múltiplos pontos*, m diferentes posições de cruzamento k são escolhidas aleatoriamente e ordenadas de forma ascendente. Assim, os genes (ou variáveis) entre pontos de cruzamento sucessivos são trocados entre os dois pais para produzir dois novos descendentes, conforme mostrado na Figura 3.6 para dois pontos de cruzamento ($m=2$) nas posições 3 e 9.

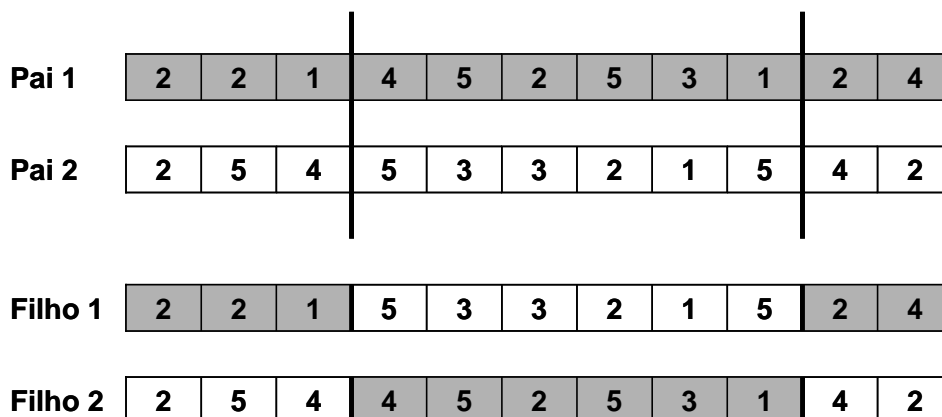


Figura 3.6: Cruzamento em múltiplos pontos

Já no *cruzamento uniforme*, não se utilizam pontos de corte: cada indivíduo é gerado gene a gene, com uma probabilidade de 50% de herdar o gene de um dos pais, como ilustrado na Figura 3.7, em que o 1º e o 2º genes do filho 1 são herdados do pai 1, enquanto o 3º e o 4º genes vêm do pai 2, o 5º gene vem do pai 1, o 6º e o 7º do pai 2, e assim sucessivamente. O mesmo procedimento é aplicado para gerar o filho 2.

Pai 1	2	2	1	4	5	2	5	3	1	2	4
Pai 2	2	5	4	5	3	3	2	1	5	4	2
Filho 1	2	2	4	5	5	3	2	3	1	4	2
Filho 2	2	5	1	4	3	2	5	1	5	2	4

Figura 3.7: Cruzamento uniforme

Dependendo do problema considerado, e da forma de codificação das soluções, os operadores acima podem não ser aplicáveis, uma vez que podem gerar indivíduos inconsistentes. Tome-se, por exemplo, o problema do caixeiro viajante, em que um cromossomo de solução é representado por uma seqüência ordenada de cidades a serem visitadas. Sejam A e B dois pais selecionados para cruzamento, sendo:

$$A = 3 \ 5 \ 7 \ 1 \ 2 \ 4 \ 8 \ 6 \ 9$$

$$B = 1 \ 9 \ 2 \ 3 \ 4 \ 6 \ 8 \ 7 \ 5$$

Em outras palavras, A corresponde à seguinte seqüência de cidades: 3-5-7-1-2-4-8-6-9-3. Suponha que seja aplicado um operador de cruzamento em dois pontos, após a 2ª e a 6ª posições, gerando os seguintes filhos:

$$3 \ 5 \ | \ 2 \ 3 \ 4 \ 6 \ | \ 8 \ 6 \ 9$$

$$1 \ 9 \ | \ 7 \ 1 \ 2 \ 4 \ | \ 8 \ 7 \ 5$$

Ambas as soluções acima não representam roteiros válidos para o problema do caixeiro viajante, uma vez que algumas cidades aparecem duas vezes, enquanto outras cidades não são visitadas (por exemplo, as cidades 3 e 6 aparecem duas vezes no primeiro filho, enquanto as cidades 1 e 7 não fazem parte do roteiro).

A fim de evitar esse tipo de problema, existem operadores específicos para tratar cromossomos que representam uma permutação (como, por exemplo, no problema do caixeiro viajante acima, ou no seqüenciamento de tarefas em uma máquina).

O operador *PMX* (“*partially matched crossover*”) é uma extensão do operador de cruzamento em dois pontos que busca preservar a ordem e a posição de alguns genes do cromossomo. Compreende um procedimento de reparo para evitar a ocorrência de elementos repetidos e funciona da seguinte maneira: (i) escolhem-se aleatoriamente dois pontos de corte aplicados a ambos os pais; (ii) geram-se dois pré-filhos, trocando-se os segmentos centrais entre os pais; (iii) alguns elementos podem estar duplicados e outros faltando; através dos segmentos centrais estabelece-se uma relação de mapeamento (ou equivalência de substituição), corrigindo-se os pré-filhos e obtendo os dois filhos, conforme ilustrado na Figura 3.8. Os pares de genes 3-6, 9-4, 2-5 e 1-6 ocupam as mesmas posições relativas, definindo a relação de mapeamento. Assim, o gene 5 repetido que ocupa a 1ª posição do filho 2 é substituído pelo gene 2, seu correspondente na relação de mapeamento, o gene 4 pelo 9, o gene 3 pelo 1, e assim sucessivamente, para cada um dos filhos gerados.

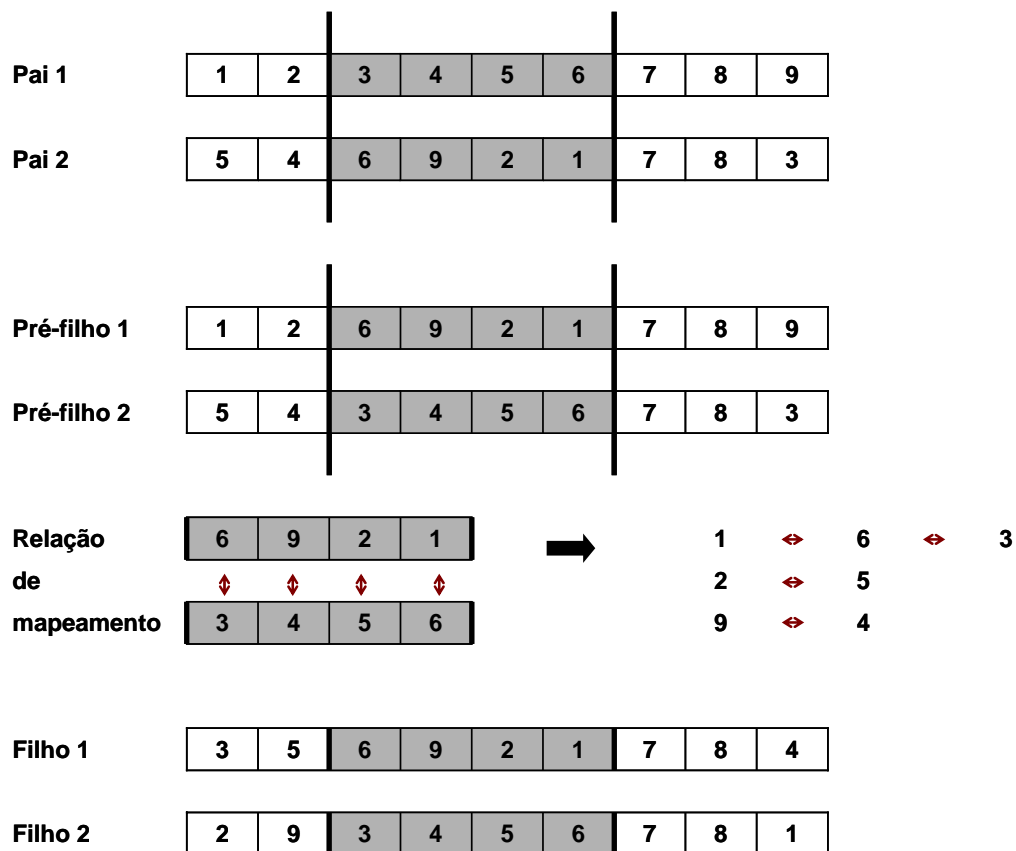


Figura 3.8: Operador de cruzamento *PMX*

Fonte: Gen e Cheng (1997)

O operador *OX* (“*order crossover*”) é uma variação do operador *PMX* com um procedimento de reparo diferente: escolhem-se aleatoriamente dois pontos de corte que irão determinar uma sub-rota; (ii) um pré-filho é criado copiando-se a sub-rota na posição correspondente do pai 1; (iii) os elementos da sub-rota são excluídos do pai 2; (iv) a partir do segundo ponto de corte, as posições vazias são preenchidas com os elementos não excluídos do pai 2; (v) quando o fim do cromossomo for atingido, continuar a partir da primeira posição deste; (vi) o segundo filho é construído seguindo-se o mesmo procedimento.

No exemplo da Figura 3.9, o pré-filho 1 é gerado mantendo-se os genes centrais [3 4 5 6] do pai 1. Esses genes são eliminados do pai 2 e os demais utilizados para preencher as demais posições do pré-filho 1, na ordem, a partir do 2º ponto de corte. Assim, o gene 2 do pai 2 preenche a sétima posição, o gene 8 a 8ª posição, o gene 7 (próximo da seqüência) a 9ª, o gene 9 a 1ª posição, e assim sucessivamente.

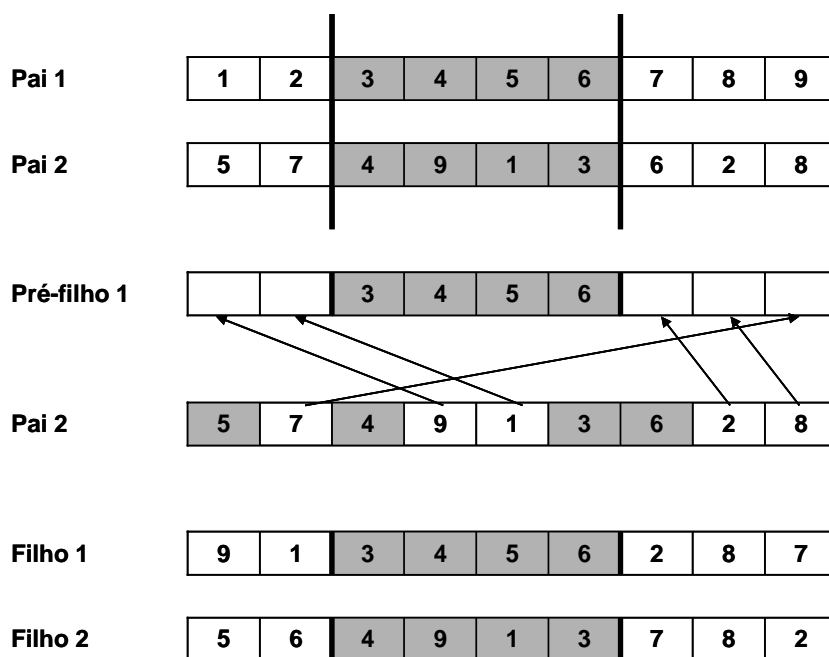


Figura 3.9: Operador de cruzamento *OX*

Fonte: Gen e Cheng (1997)

O operador *CX* (“*cycle operator*”) toma alguns pontos de um dos pais e seleciona os pontos remanescentes do outro pai. Os pontos do primeiro pai não são selecionados aleatoriamente, e sim aqueles que ocupam o mesmo conjunto de posições nos dois cromossomos pais, como ilustrado na Figura 3.10: 1-5, 5-2, 2-4, 4-9, 9-1, fechando o ciclo. Assim mantêm-se os genes 1, 5, 2, 4, 9 do pai 1, gerando o filho 1. As demais posições são preenchidas com os demais

genes faltantes na seqüência em que eles aparecem no pai 2 (6, 3, 7, 8). O mesmo procedimento é utilizado para gerar o filho 2.

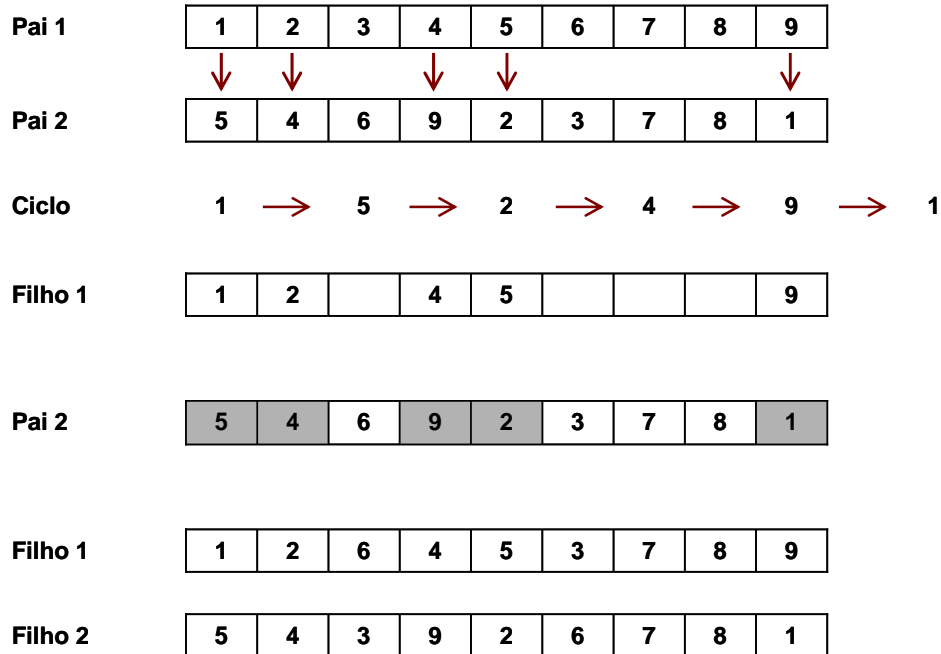


Figura 3.10: Operador de cruzamento *CX*

Fonte: Gen e Cheng (1997)

3.1.6 Operador Genético de Mutação

O operador de mutação tem por finalidade manter a variedade genética das populações geradas. A idéia é que cada gene dos descendentes gerados pode sofrer mutação com uma pequena probabilidade, de modo a evitar que o processo de busca torne-se puramente aleatório. Busca-se, dessa forma, recuperar boas características eventualmente perdidas nos processos de seleção e cruzamento, e também evitar ficar presos em determinadas regiões do espaço de busca, ou em pontos de ótimo locais.

O método de mutação depende do tipo de representação adotada. Para cromossomos com representação binária, pode-se mutar algum gene de 1 para 0 ou vice versa. Uma maneira de selecionar quais genes de quais indivíduos serão mutados consiste em percorrer todos os genes de todos os indivíduos, sorteando-se, para cada um deles, um número aleatório que, comparado com uma dada probabilidade de mutação (em geral baixa, de 1 a 5%), define se o gene deve ser mutado ou não. Entretanto, esse procedimento consome muito tempo de

processamento, pois requer que todos os genes de todos os indivíduos sejam examinados, podendo ser substituído por um mais expedito; por exemplo, se cada indivíduo possui 100 genes e a probabilidade de mutação é de 2%, dois genes são escolhidos aleatoriamente em cada indivíduo para mutação. Uma possibilidade não usual é definir uma taxa de mutação variável, que aumenta quando a solução deixa de convergir.

Já para cromossomos que tem sua representação baseada na ordem (i.e., permutação, como no problema do caixeiro viajante) existem operadores de mutação específicos, a semelhança do que ocorre no cruzamento.

Na mutação por inversão, dois pontos de corte são selecionados aleatoriamente e a seqüência entre essas duas posições do cromossomo é invertida, como mostrado na Figura 3.11.

Indivíduo	1	2	3	4	5	6	7	8	9
Indivíduo mutado	1	2	6	5	4	3	7	8	9

Figura 3.11: Mutação por inversão

Fonte: Gen e Cheng (1997)

A mutação por substituição consiste em selecionar uma seqüência ao acaso e inseri-la em uma posição aleatória, como mostrado na Figura 3.12. Uma variação para esse operador de mutação consiste na inversão da seqüência quando inserida na posição aleatoriamente escolhida.

Indivíduo	1	2	3	4	5	6	7	8	9
Indivíduo mutado	1	2	7	8	3	4	5	6	9

Figura 3.12: Mutação por substituição

Fonte: Gen e Cheng (1997)

A mutação por inserção é um caso particular da mutação por substituição em que a seqüência selecionada para inserção em uma posição aleatória contém apenas um gene.

Na mutação de troca recíproca selecionam-se aleatoriamente duas posições no mesmo cromossomo e os respectivos elementos são trocados, como mostrado na Figura 3.13.

Indivíduo	1	2	3	4	5	6	7	8	9
Indivíduo mutado	1	2	6	4	5	3	7	8	9

Figura 3.13: Mutaç o de troca rec proca

Fonte: Gen e Cheng (1997)

A *muta o heur stica* consiste em escolher λ genes ao acaso e gerar as solu es vizinhas considerando-se todas as poss veis permuta es dos λ genes selecionados, escolhendo-se a melhor, como ilustrado na Figura 3.14, em que foram escolhidos $\lambda = 3$ genes (3, 6 e 8) e testadas todas as possibilidades de permuta o desses 3 genes, devendo ser escolhida a melhor.

Cromossomo inicial	1	2	3	4	5	6	7	8	9
Vizinhan�a	1	2	3	4	5	8	7	6	9
	1	2	8	4	5	3	7	6	9
	1	2	8	4	5	6	7	3	9
	1	2	6	4	5	8	7	3	9
	1	2	6	4	5	3	7	8	9

Figura 3.14: Muta o heur stica

Fonte: Gen e Cheng (1997)

3.1.7 Atualiza o da Popula o

Uma vez gerados todos os descendentes de uma popula o e efetuadas as muta es,   necess rio atualizar a popula o, eliminando-se os indiv duos menos aptos e preservando-se os melhores indiv duos. Isso visa a manter a popula o sempre do mesmo tamanho, evitando-se os inconvenientes e a complica o computacional de lidar com uma popula o cada vez maior.

A maneira mais simples e direta de fazer isso   simplesmente substituir toda a popula o antiga pela nova, independentemente da sua aptid o. Para tanto,   necess rio preservar a informa o gen tica do melhor indiv duo dentre os ascendentes, repassando-o para a pr xima gera o, opera o esta chamada de *elitismo*. Entretanto, h  perda do material gen tico de bons

indivíduos, mesmo utilizando a estratégia elitista, uma vez que outros bons indivíduos dentre os pais são eliminados.

Outra alternativa é a chamada substituição de geração com sobreposição da população. Consideram-se todos os indivíduos, tanto os descendentes quanto seus pais, ordenando-os com base nos respectivos valores de função aptidão. Os “piores” indivíduos (menos aptos) são simplesmente eliminados, de modo a manter sempre a população do mesmo tamanho. Uma variante consiste em substituir o pior ascendente pelo melhor descendente e prosseguir assim até que os ascendentes remanescentes sejam melhores que os descendentes remanescentes. Pode-se considerar ainda um “fator de pressão” na comparação dos descendentes com os ascendentes; em outras palavras, a substituição dos ascendentes só ocorre quando o valor da função aptidão dos descendentes, multiplicado por um fator inferior a 1, superar a aptidão dos ascendentes.

Um aspecto fundamental é evitar a ocorrência de indivíduos repetidos na população que sobrevive (permanece), a fim de assegurar a diversidade genética necessária para prevenir o algoritmo genético de convergir prematuramente para um ponto de ótimo local. Entretanto, isso nem sempre é simples de ser verificado, uma vez que muitas vezes não basta simplesmente comparar os valores de função aptidão, já que dois indivíduos distintos (em termos de seus cromossomos) podem apresentar a mesma aptidão. Comparar os genes um a um pode ser muito trabalhoso e demorado. Uma alternativa seria comparar outros atributos globais da solução para verificar se os indivíduos são efetivamente idênticos. Por exemplo, no problema da mochila binária citado anteriormente, pode-se comparar os pesos totais de dois indivíduos, além dos respectivos valores de aptidão (lucro total) para definir se são ou não idênticos.

Outra possível alternativa, quando a comparação de indivíduos pode ser difícil, é a geração de poucos indivíduos novos, no máximo 25% da população, a cada geração. Dada a dificuldade para comparar indivíduos a fim de verificar se são idênticos, com impacto também no tempo de processamento, Galvão (2004) adotou uma estratégia diferente: essa verificação foi simplesmente eliminada; quando o algoritmo genético deixava de convergir para uma solução melhor após um certo número de iterações, simplesmente se reiniciava o processamento, gerando uma nova população de soluções, porém preservando alguns dos melhores indivíduos obtidos até então como parte de população inicial do novo processamento.

3.1.8 Parâmetros para o Algoritmo Genético

Os principais parâmetros para o Algoritmo Genético são:

- Tamanho da população, ou seja, número de indivíduos em cada geração. É um parâmetro crítico para o desempenho do AG, uma vez que uma população muito pequena não permite cobrir de maneira eficiente o espaço de busca e leva à convergência prematura do AG; por outro lado, quanto maior o número de indivíduos, maior o tempo total de processamento. Como regra geral, sugere-se uma população de tamanho entre N e $2N$, onde N é o tamanho do cromossomo (número de genes).
- Taxa de cruzamento: valores altos podem levar à introdução de novos indivíduos mais rapidamente na população, o que pode ser favorável, mas, ao mesmo tempo podem favorecer a exclusão de bons cromossomos de forma mais rápida; por outro lado, baixos valores podem tornar o algoritmo muito lento. Em geral sugere-se adotar uma taxa de probabilidade entre 60 e 90%.
- Taxa de mutação: para representações binárias, a maioria dos autores sugere uma taxa de mutação entre 0,1 e 1% para cada gene, podendo ser maior no caso do algoritmo não encontrar uma solução melhor após muitas iterações. Outros autores recomendam probabilidades baseadas no tamanho do cromossomo ($1/N$) e no tamanho da população. Para a representação baseada na ordem, a taxa de mutação deve ser aplicada a cada cromossomo e não a cada gene.
- Critério de parada: pode ser tempo total de processamento, número máximo de gerações, número máximo de novos indivíduos gerados, ou número máximo de gerações consecutivas sem melhoria da melhor solução.

Segundo Harik e Lobo (1999), a escolha do conjunto de parâmetros adequados ao problema a ser tratado pode, muitas vezes, se tornar uma tarefa árdua. Em alguns problemas, mais simples, os autores observaram que um AG com parâmetros convencionais pode ter melhor desempenho. Adicionalmente, segundo os autores, não se conhece os parâmetros ótimos para um problema qualquer do mundo real.

Assim, para facilitar o uso dos AGs por usuários não especialistas na técnica e sem perder a característica de robustez dos mesmos, os autores propuseram um AG com redução de parâmetros. A primeira simplificação é a desconsideração da mutação. Os autores

recomendam populações pequenas (N); quando a população convergir, o seu tamanho deve ser duplicado. Os autores verificaram que (i) a seleção para cruzamento por torneio, com taxa de seleção igual a 4 (i.e., quatro indivíduos são escolhido e o melhor selecionado), e (ii) a probabilidade de cruzamento igual a 50% permitem o contínuo repasse de boas características da população ao longo das gerações.

3.1.9 Novos avanços

Mais recentemente, vêm-se explorando a aplicação dos Algoritmos Genéticos em conjunto com outras heurísticas e metaheurísticas, a fim de buscar atenuar seus pontos fracos, em particular tempos de processamento muitos longos para a obtenção de boas soluções, e convergência prematura para um ótimo local. A idéia geral dos algoritmos genéticos híbridos é utilizar os AGs para uma exploração mais global (“*exploration*”) de uma população, enquanto outras heurísticas são usadas para realizar uma exploração local (“*exploitation*”) em torno dos cromossomos. Em outras palavras, os AGs podem ser combinados com diferentes técnicas de busca específicas para o problema a ser tratado, formando um híbrido que explora a perspectiva global do AG e a convergência da técnica específica do problema. Com a combinação das propriedades de AGs e heurísticas convencionais, o método híbrido pode apresentar desempenho superior em relação a outros métodos aplicados isoladamente.

A combinação de algoritmo genético com uma heurística de busca local (que é aplicada a cada novo indivíduo gerado) denota um algoritmo genético híbrido, também conhecido na literatura como *algoritmo memético* (Moscato, 1989), embora nem todos os autores façam essa distinção de nomenclatura (Beasley e Chu, 1996; Chu e Beasley, 1997). Uma estratégia híbrida envolvendo AG e “*simulated annealing*” foi proposta por Cunha e Silva (2006), e aplicada com sucesso a um problema de configuração de redes do tipo “*hub-and-spoke*”.

Por fim, outra estratégia para aumentar a eficiência dos algoritmos genéticos é o *paralelismo*, que envolve a divisão de tarefas entre os processadores de um computador paralelo.

Um algoritmo genético paralelo usa um processador para controlar a seleção, o cruzamento e a mutação, e outros processadores são utilizados para realizar a avaliação da função aptidão, uma vez que costuma ser uma das tarefas que mais consome tempo de processamento.

Já o algoritmo genético paralelo com granularidade baixa, também conhecido como modelo ilha, tenta imitar a separação geográfica de sub-populações existente no mundo real. Dessa forma, várias sub-populações isoladas evoluem simultaneamente em processadores paralelos, e periodicamente trocam informações através da migração dos seus melhores indivíduos para sub-populações vizinhas. A migração também pode ser feita disseminando-se o melhor indivíduo de cada geração entre todas as sub-populações, ou ainda, copiando-se subconjuntos de soluções escolhidos aleatoriamente entre as sub-populações em algumas gerações, bem espaçadas.

O algoritmo genético paralelo com granularidade alta, também conhecido como modelo da vizinhança, aloca cada indivíduo da população a um processador de uma grade de processadores. O processo de seleção e cruzamento é aplicado somente entre indivíduos vizinhos na grade, ou seja, só é permitido aos indivíduos reproduzir com seus vizinhos mais próximos.

Maiores detalhes sobre essa interessante e promissora área de paralelização em algoritmos genéticos podem ser encontrados em Cantú-Paz (2001); para uma abordagem mais genérica aplicada a metaheurísticas recomenda-se Crainic e Toulouse (2003).

3.2 BUSCA TABU

O termo *tabu* (ou *taboo*) vem do Tongan, uma língua falada na Polinésia, usada pelos aborígenes para indicar coisas que não podem ser tocadas por serem sagradas (Glover e Laguna, 1997). Na maioria dos dicionários de língua inglesa e portuguesa, *tabu* é definido como “proibição imposta pelo costume social como medida de proteção, algo banido devido ao risco, ou ainda restrição de natureza ritual e religiosa, que determina que certos objetos, indivíduos, lugares ou atos, por serem considerados sagrados ou especialmente impuros e perigosos, sejam evitados, e que, como instituição social está associada a fortes sanções e à crença de que sua violação traz castigo sobrenatural”.

Da mesma forma, na metaheurística Busca Tabu (“*Tabu Search*”), proposta por Glover (1986), o risco a ser evitado é seguir um curso contra-producente, que pode levar a busca a ficar presa, sem chance de escapar. Em outras palavras, embora não se considerem aspectos

sobrenaturais ou morais, uma das características-chave da busca tabu é a exclusão de regiões do espaço de busca classificadas como “proibidas”, evitando-se o risco de visitar repetidamente um mesmo conjunto de soluções e ficar preso em ótimos locais.

Por outro lado, assim como em um contexto social mais abrangente, em que proibições com a finalidade de proteção podem ser deixadas de lado quando a situação assim o demandar, os “tabus” da busca tabu não só podem, como devem ser violados, quando houver evidência clara de que há uma melhor alternativa a ser seguida.

A mais importante associação com o uso tradicional da palavra vem do fato de que proibições são transmitidas por meio de uma memória social, sujeita a modificações no tempo; ou seja, o que é proibido hoje pode não ser proibido amanhã, e vice-versa. Isso cria a ligação fundamental do sentido do termo *tabu* na busca tabu. O conjunto de elementos proibidos da busca tabu faz parte de uma memória evolutiva, o que possibilita sua alteração de acordo com o tempo e a circunstância.

Mais especificamente, a busca tabu baseia-se na premissa de que o método de solução de um problema, para ser qualificado como inteligente, deve incorporar *memória adaptativa* e *exploração sensível ao contexto* (“*responsive*”). A memória adaptativa da busca tabu permite guiar a busca do espaço de solução de maneira econômica e efetiva. Tendo em vista que as escolhas feitas localmente são guiadas pelas informações coletadas ao longo do processo, a busca tabu contrasta com outros métodos “sem memória”, que dependem de processos semi-randômicos baseados em uma forma de amostragem. A memória adaptativa também contrasta com os esquemas de memória rígidos encontrados em estratégias do tipo “*branch-and-bound*”.

A ênfase na exploração responsiva (“*responsive*”), ou seja, sensível ao contexto, tanto determinística quanto probabilística, deriva da suposição de que uma má escolha estratégica pode fornecer mais informação que uma boa escolha aleatória. Em um sistema que usa memória, uma má escolha baseada em uma estratégia pode fornecer pistas úteis sobre como a estratégia pode ser modificada ou aprimorada. A exploração sensível ao contexto integra os princípios básicos da busca inteligente, isto é, examinar boas características de solução, e simultaneamente explorar novas regiões promissoras.

A busca tabu pode ser caracterizada como uma forma de busca em vizinhança. Na busca em vizinhança, cada solução $x \in X$ (onde X denota o espaço de soluções viáveis) tem associada

um conjunto de vizinhos, $N(x) \subset X$ denominado *vizinhança* de x . Cada solução $x' \in N(x)$ pode ser obtida diretamente a partir de x através de uma operação denominada *movimento*.

Assim, a idéia central da busca tabu consiste na utilização de estruturas adaptativas de memória para guiar uma heurística de busca local, direcionando a busca a regiões mais promissoras e evitando que o processo de busca fique preso a um mínimo local. Partindo-se de uma solução inicial, essa heurística de busca local baseia-se na exploração da vizinhança de uma solução, vizinhança essa definida pelos movimentos considerados em cada implementação específica. Em problemas de roteirização, por exemplo, pode-se imaginar um movimento de troca de dois pontos que pertencem a rotas diferentes, ou ainda realocação de um ponto, que troca sua posição dentro da própria rota, ou é movimentado para uma determinada posição em outra rota. Cada movimento transforma uma solução intermediária em outra, e a cada iteração um movimento da vizinhança é selecionado, sendo o processo repetido até se atingir um critério de parada.

3.2.1 Uso de Memória

De acordo com Glover e Laguna (1997), as estruturas de memória da busca tabu baseiam-se em quatro dimensões principais: antecedência recente (“*recency*”), frequência, qualidade e influência.

A dimensão de *antecedência recente* (“*recency*”) tem por finalidade memorizar os atributos que mudaram durante o passado recente nas soluções encontradas. Atributos alterados recentemente são rotulados como “tabu-ativos”. Soluções que contém atributos tabu-ativos tornam-se tabu, evitando serem re-visitadas.

A dimensão de *frequência* visa a memorizar a frequência dos atributos durante o passado das soluções. Com base nas frequências observadas, pode-se dar pesos maiores ou menores para cada atributo no prosseguimento da busca.

Memórias baseadas em frequência e antecedência são complementares e possuem importantes características que serão detalhadas mais adiante.

A dimensão de *qualidade* refere-se à habilidade de diferenciar o mérito das soluções visitadas durante a busca. Nesse contexto, a memória pode ser usada para identificar elementos que são

comuns às boas soluções ou a caminhos que levam a tais soluções. Operacionalmente, a qualidade torna-se um fundamento para o aprendizado baseado em incentivo, onde estímulos são oferecidos para reforçar ações que levam a boas soluções, e penalidades são aplicadas para desencorajar ações que levam a soluções ruins.

A quarta dimensão, *influência*, considera o impacto das escolhas feitas durante a busca, não apenas quanto à qualidade mas também quanto à estrutura. Armazenar informação sobre a influência de escolhas nos elementos de uma solução em especial incorpora um nível adicional de aprendizado, diferentemente do “*branch-and-bound*”, onde as regras de escolha e direção são pré-estabelecidas e permanecem fixas.

A memória usada na busca tabu é explícita e por atributos. A memória explícita permite armazenar soluções completas, tipicamente soluções elite visitadas durante a busca. Adicionalmente, pode-se estender essa memória para armazenar vizinhos de soluções de elite muito atraentes, porém não explorados. As soluções de elite armazenadas e seus vizinhos podem ser utilizados para expandir a busca local. Já a memória de atributos é utilizada para guardar informações sobre atributos que mudam de uma solução para outra. Por exemplo, em um grafo os atributos podem ser os nós e arcos que são adicionados, retirados ou re-posicionados através de mecanismos de movimento.

3.2.2 Intensificação e Diversificação

Dois importantes componentes da busca tabu são as estratégias de intensificação e diversificação. Estratégias de *intensificação* baseiam-se em modificar regras de escolha, a fim de encorajar combinações de movimentos e características de soluções historicamente verificadas como boas. Elas também podem deflagrar um retorno a regiões atraentes para uma busca mais cuidadosa. Tendo em vista que soluções de elite devem ser armazenadas para permitir a inspeção de sua vizinhança, estruturas de memória explícita são utilizadas na implementação das estratégias de diversificação e intensificação. A intensificação requer um meio de identificar soluções elite para servir de base para a busca. Para tanto, considera-se o uso de um gatilho (“*threshold*”) associado ao valor da função objetivo da melhor solução encontrada para a inclusão de um membro no conjunto de elite.

Já a *diversificação* estimula o processo de busca a examinar regiões não visitadas, e gerar soluções que diferem de várias maneiras daquelas encontradas anteriormente. Novamente, pode-se gerar componentes totalmente diferentes para produzir novas soluções completas, ou alterar a função de avaliação através de esquemas de penalização de atributos de soluções que já tenham sido vistas e incentivo para novos atributos ainda não explorados.

3.2.3 Memória de Curto Prazo

De acordo com Glover e Laguna (1997), a memória de curto prazo objetiva reduzir a vizinhança considerada de uma solução, proibindo (daí o termo *tabu*) a exploração de algumas soluções vizinhas $x' \in N(x)$. Este mecanismo é utilizado para prevenir a formação de ciclos durante o trajeto de busca, não permitindo a visitação das últimas soluções escolhidas no trajeto.

Este tipo de memória é um dos pilares da busca *tabu*, pois confere à busca a oportunidade de transpor mínimos locais, através da escolha de movimentos que não melhoram a solução, em conjunto com a modificação da vizinhança para as soluções seguintes.

A principal característica da memória de curto prazo é a manutenção de uma lista de soluções (ou de movimentos que levaram a ela) que são consideradas proibidas (*tabu*) por um determinado número de iterações, denominada *lista tabu*. Isto é, após visitar uma solução, o processo de busca não pode retornar a essa mesma solução por um determinado número de iterações, chamado *período tabu* (“*tabu tenure*”).

Assim, evita-se que a busca fique presa a um mínimo local, já que, essa solução, ao tornar-se *tabu*, força a que outras soluções vizinhas sejam exploradas. Associado ao conceito de *lista tabu*, que nada mais é que uma memória de curta duração da busca realizada, está o conceito de critério de aspiração. Esse critério permite relaxar a aplicação da *lista tabu*, permitindo que movimentos proibidos possam ser executados, desde que atendam ao critério de aspiração. Em sua forma mais simples, esse critério permite a execução de um movimento *tabu*, desde que ele gere uma solução de menor custo que a melhor solução encontrada até então no processo de busca. Outra alternativa é permitir um movimento *tabu* se a direção da busca não mudar, independentemente da nova solução ser melhor ou não.

A título de ilustração, alguns exemplos de movimentos que se tornam tabu (Glover e Laguna, 1997):

- Após um movimento que muda o valor de x_i de 0 para 1, pode-se desejar prevenir x_i de assumir o valor 0 pelas próximas *DurTabu* iterações. O atributo a armazenar é i , com a seguinte regra de ativação-tabu: o movimento ($x_i \leftarrow 0$) é tabu se i estiver tabu-ativo.
- Após um movimento que troca de posição os elementos i e j de uma seqüência (por exemplo um roteiro de caixeiro viajante ou a seqüência de tarefas numa máquina), pode-se desejar prevenir que os elementos i e j troquem de posição novamente nas próximas *DurTabu* iterações. Os atributos a armazenar são i e j , com a seguinte regra de ativação-tabu: o movimento ($i \leftrightarrow j$) é tabu se ambos i e j estiverem tabu-ativos.
- Após um movimento que remove o elemento i e adiciona o elemento j na solução corrente (como, por exemplo, no problema da mochila binária, em que o objeto i é retirado da mochila e o objeto j inserido na mochila), pode-se desejar prevenir que o elemento i seja adicionado à solução nas próximas *DurTabuAdd* iterações e que o elemento j seja removido da solução nas próximas *DurTabuDrop* iterações. Os atributos a armazenar são i e j , com as seguintes regras de ativação-tabu: o movimento (*Adicionar i*) é tabu se i estiver tabu-ativo; o movimento (*Remover j*) é tabu se j estiver tabu-ativo.

Portanto, somente movimentos podem ser tabu. Atributos nunca são tabu, eles podem apenas estar tabu-ativos. A classificação de um movimento (como tabu ou não tabu) é determinada pelas regras de ativação-tabu.

Utiliza-se uma estrutura de memória para registrar quando (número da iteração) termina o status tabu-ativo de um atributo. Assim, é feita a seguinte atualização após um movimento: $FimTabu(Atributo) = Iter + DurTabu$. O atributo é tabu-ativo se $Iter \leq FimTabu(Atributo)$.

A Figura 3.15 ilustra a árvore de decisões de movimentos tabu.

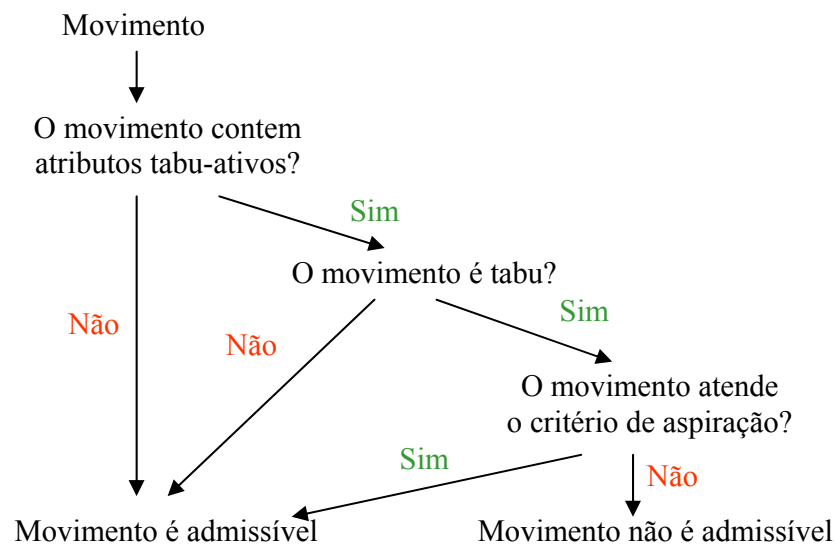


Figura 3.15: Árvore de decisão tabu

Fonte: adaptado de Glover e Laguna (1997)

Segundo Glover e Laguna (1997), durações menores do período tabu (*DurTabu*) geralmente induzem à exploração de soluções próximas a um mínimo local, ao passo que períodos mais prolongados permitem que a busca se desloque para uma região livre da influência de um mínimo local. Essa característica sugere a possível utilização de um período tabu (“*tabu tenure*”) variável que, segundo os autores, pode ser classificado em *sistemático* ou *dinâmico*. O primeiro caso consiste em utilizar, sistemática e seqüencialmente, valores de uma lista de períodos previamente definida. Já o segundo caso consiste em selecionar aleatoriamente um valor de período tabu dentre uma faixa definida por limites máximo e mínimo preestabelecidos. Esta última abordagem é bastante comum, podendo-se citar os trabalhos de Taillard (1993) e Gendreau *et al.* (1994) como exemplos bem sucedidos de aplicações de busca tabu a problemas de roteirização de veículos que utilizam período tabu com essa característica.

Embora em muitos casos as estruturas de memória de curto prazo sejam suficientes para se obter uma solução de boa qualidade, considera-se que a incorporação de estruturas de memória de longa duração complementa o funcionamento das estruturas de memória de curta duração, e geralmente torna o processo de busca mais poderoso. Nesse sentido, estruturas de longa duração geralmente estão associadas a estratégias de diversificação e/ou intensificação da busca. Estratégias de diversificação procuram evitar que regiões do espaço de busca permaneçam completamente inexploradas, ao passo que as estratégias de intensificação procuram focar a busca nas regiões mais promissoras do espaço de busca.

3.2.4 Memória de Longo Prazo

A utilização deste tipo de memória pode ser regida por estratégias de diversificação e intensificação. Conforme visto anteriormente, a diversificação tem o intuito de assegurar que a busca não ficará restrita a apenas uma parte do espaço de soluções do problema e a intensificação está ligada à exploração de uma região próxima às melhores soluções conhecidas.

Assim, a memória de longo prazo pode ser utilizada para alterar o critério de escolha da solução, bem como para incluir na vizinhança modificada, soluções não encontradas na vizinhança original. A primeira estratégia pode ser seguida, penalizando a escolha de movimentos muito executados, gerando uma certa diversificação à busca. A segunda estratégia é executada, por exemplo, armazenando características comuns em boas soluções encontradas no decorrer da busca e em determinados momentos combinar estas características, gerando novas soluções com grande potencial de exploração. Esta última aplicação também é conhecida como memória adaptativa (“*adaptive memory*”) e foi bastante explorada por Rochat e Taillard (1995) para problemas de roteirização. Apesar de não ser obrigatória, a utilização de memória de longo prazo em conjunto com a de curto prazo, em geral, confere uma maior robustez à busca e permite encontrar melhores soluções em um número reduzido de iterações (Glover e Laguna, 1997).

Essas estruturas baseiam-se no histórico da busca, e geralmente estão associadas à frequência da ocorrência de atributos das soluções ao longo do processo de busca. A informação assim coletada é utilizada para a implementação de estratégias de diversificação e/ou intensificação da busca.

Vale notar que estratégias de diversificação e/ou intensificação nem sempre se valem de estruturas de memória de longa duração para alcançar seus objetivos. Hertz *et al.* (1997) citam a modificação da vizinhança pela inclusão de movimentos mais complexos como uma forma de intensificação, e a utilização de reinícios aleatórios como uma forma de diversificação. No entanto, boa parte das estratégias de diversificação e/ou intensificação baseia-se na utilização de memória de longa duração, em particular na utilização de informação associada à frequência da ocorrência de atributos das soluções ao longo do processo de busca. Glover e Laguna (1993, 1997) ensinam que essas medidas de frequência são calculadas como razões, em que o numerador corresponde a uma medida de transição (“*transition measure*”), isto é

número de iterações em que um atributo foi mudado (i.e., adicionado ou removido de uma solução) ou medida de permanência (“*residence measure*”), isto é, número de iterações em que um atributo permaneceu em uma dada posição, e que os denominadores geralmente representam um valor tal como o total do número de ocorrências dos eventos representados nos numeradores, ou a média (ou soma, ou máximo) desses valores.

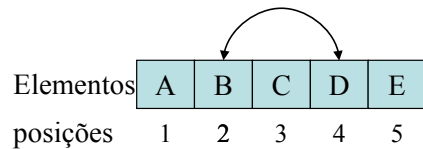
As medidas de transição medem a frequência com que atributos são alterados durante a busca, isto é, a frequência com que são adicionados ou eliminados durante esse processo. Já as medidas de permanência correspondem à frequência com que atributos pertencem às soluções visitadas durante o processo de busca.

3.2.5 Exploração da vizinhança

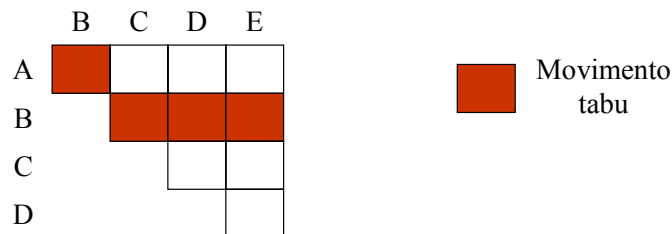
Estreitamente relacionada à avaliação dos movimentos, o gerenciamento da lista de movimentos candidatos é um dos fatores chave para uma eficiente implementação da *Busca Tabu*. Frequentemente os movimentos adotados definem uma vizinhança demasiado ampla, cuja completa exploração pode ser muito custosa do ponto de vista computacional.

A flexibilidade da busca, ou seja, o número de movimentos admissíveis na vizinhança da solução corrente depende do tipo de movimento; das regras de ativação-tabu, do período (ou duração) tabu (“*tabu tenure*”), e do critério de aspiração. Na Figura 3.16 são mostradas diferentes possíveis regras de ativação tabu relacionadas a um movimento de troca entre dois elementos de uma seqüência.

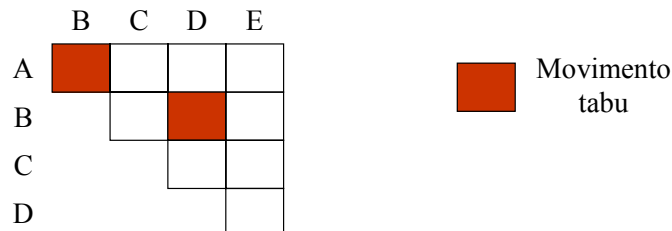
Nesse sentido, Glover e Laguna (1993) afirmam que amiúde é recomendável aplicar a busca tabu em conjunto com alguma estratégia que isole regiões da vizinhança que contenham movimentos com características desejáveis, inserindo-os na lista de movimentos candidatos. Diversas estratégias de gerenciamento da lista de movimentos que restringem o número de movimentos considerados são comentadas no trabalho, assim como em Glover e Laguna (1997), das quais pode-se destacar a estratégia de manutenção de candidatos de elite (“*elite candidate list*”) e a estratégia de filtros sucessivos (“*successive filter strategy*”), entre outras.



Regra de ativação tabu: movimento (B ↔ *) é tabu



Regra de ativação tabu : movimento (B ↔ *) é tabu
se B move para posição 2 ou menor



Regra de ativação tabu : movimento (B ↔ D) é tabu

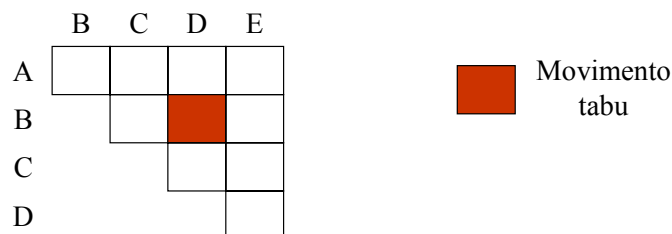


Figura 3.16: Diferentes regras de ativação tabu

Fonte: adaptado de Glover e Laguna (1997)

A primeira estratégia consiste em inicialmente construir a lista de candidatos de elite com todos os movimentos da vizinhança, que são então avaliados, mantendo-se apenas os melhores movimentos. A cada iteração subsequente, o melhor movimento da lista é selecionado, até que não seja possível alcançar um mínimo de qualidade preestabelecido, quando então a lista de candidatos de elite é recriada, repetindo-se todo o processo. Essa estratégia fundamenta-se na hipótese de que a qualidade de um movimento não se deteriora em um número reduzido de iterações. Ou seja, um bom movimento continua sendo atrativo (de elite), mesmo após a realização de algumas iterações.

A segunda estratégia baseia-se no fato de que um movimento, em geral, pode ser decomposto em operações mais simples, e que o conjunto de movimentos candidatos pode ser reduzido pela eliminação dos movimentos cujas operações não alcancem um determinado limite de qualidade. Está implícita a hipótese de que a avaliação aproximada dos movimentos por meio de suas operações mais simples, de menor esforço computacional, é suficiente para permitir a seleção dos melhores movimentos.

Não se pode deixar de comentar aqui a interessante estratégia de redução da lista de movimentos candidatos denominada *Busca Tabu Granular*, proposta por Toth e Vigo (2003). Os autores descrevem a aplicação dessa estratégia a um problema de roteirização de veículos, em que propõem a adoção de um limite de “granularidade” para a eliminação, a priori, dos arcos de maior custo do grafo associado ao problema, cuja probabilidade de pertencer a uma solução de boa qualidade é extremamente reduzida. Pode-se notar que essa abordagem incorpora elementos das estratégias comentadas anteriormente, pois reduz a lista de candidatos ao filtrar a vizinhança segundo características dos arcos formadores dos movimentos, e considera reconstruções periódicas da lista de candidatos, similar ao que ocorre na estratégia de candidatos de elite.

3.2.6 Estrutura Geral de Funcionamento da Busca Tabu

A Figura 3.17 sintetiza a estrutura geral de funcionamento da busca tabu. O critério de parada está usualmente relacionado a um número máximo de iterações, ou a um tempo máximo de processamento, ou ainda a um número máximo de iterações consecutivas sem melhorias. É importante destacar que, independente do critério escolhido, o número de iterações a serem executadas na busca tabu deve ser elevado, a fim de permitir que os mecanismos de memória, tanto de curto prazo quanto de longo prazo, tenham tempo de atuar. Muitas vezes a busca fica estagnada, isto é, não consegue encontrar uma solução melhor, porque os mecanismos de memória ainda não conseguiram deslocar a busca para uma região diferente.

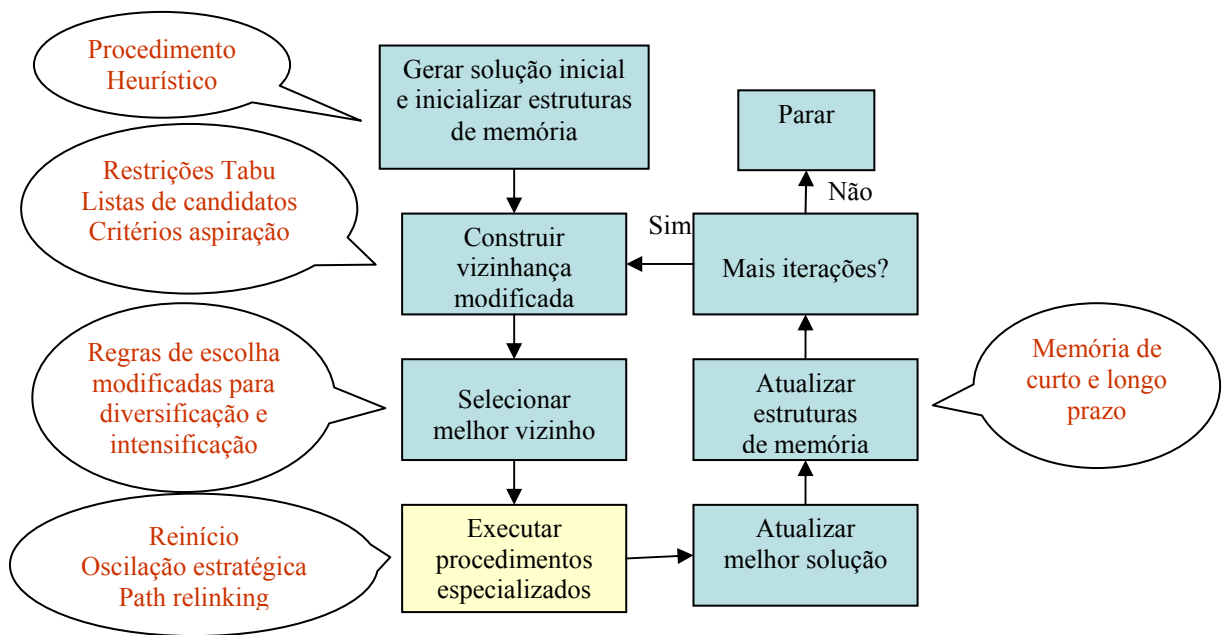


Figura 3.17: Estrutura Geral da Busca Tabu

Fonte: adaptado de Glover e Laguna (1997)

3.2.7 Outros Aspectos

A busca tabu é uma das metaheurísticas mais extensivamente utilizadas, com excelentes resultados para inúmeros problemas complexos de otimização combinatória, tendo se destacado no meio acadêmico como a metaheurística mais promissora para a resolução de problemas de roteirização de veículos (Laporte *et al.*, 2000).

Entretanto, sua implementação computacional é complexa e trabalhosa. O número de parâmetros da busca é elevado, o que pode requerer um esforço considerável para a sua adequada calibração, a fim de se obter melhores resultados. Dependendo da implementação e da complexidade da estrutura de vizinhança adotada, os tempos de processamento podem ser elevados (geralmente são), e não existe garantia de atingir o ótimo, o que pode dificultar a sua aplicação a problemas reais no contexto de uma solução integrada de *software*.

Deve-se destacar que o uso extensivo de recursos de memória requer estruturas de dados cuidadosamente projetadas para armazenar e acessar os dados armazenados de maneira eficiente. Adicionalmente, deve-se tomar cuidado para: evitar coletar mais dados do que se pode utilizar; entender claramente quais atributos de solução são cruciais, buscando limitar os

atributos de solução a serem memorizados; considerar padrões enganosos em regiões de ótimo local que podem ser muito distintos do ótimo global.

De acordo com Taillard *et al.* (2001), dentre todas as metaheurísticas descritas neste capítulo, a busca tabu é a única que foi explicitamente desenvolvida com um conjunto de memórias. De uma certa maneira, este método procura imitar o ser humano procurando por uma boa solução de um problema de otimização combinatória. Inicialmente, busca-se uma solução, mesmo que ruim. Em seguida, essa solução é iterativamente melhorada, através de modificações locais. Algumas dessas modificações não necessariamente levam a uma melhoria da solução em cada iteração, mas procuram direcionar a busca para um bom subconjunto de soluções.

As pesquisas mais recentes têm indicado ser recomendável que a busca tabu não se restrinja ao espaço de solução viável; portanto, em determinados momentos da busca, principalmente quando ocorrer uma estagnação da mesma, deve-se permitir movimentos que levem a soluções não viáveis. Isso decorre da observação de que em muitos casos a solução ótima está muito próxima da fronteira ou limite da região viável. Assim, a busca tabu pode ser vista como um método de busca em vizinhança dinâmica. Tal recurso, denominado oscilação estratégica (Kulturel-Konak *et al.*, 2004), visa alternar a busca entre a região viável e a inviável, deixando a solução ser inviável até certo ponto e a seguir, através de penalidades, voltar à região viável, como ilustrado na Figura 3.18.

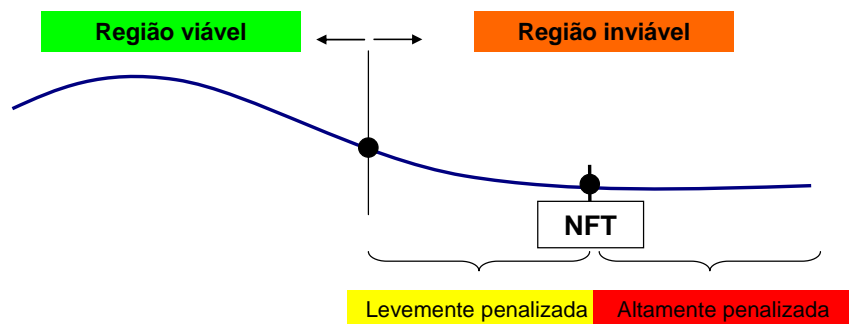


Figura 3.18: Oscilação Estratégica

Fonte: adaptado de Kulturel-Konak *et al.*, 2004

A *busca tabu reativa*, proposta por Battiti e Tecchiolli (1994), mantém registro de todas as soluções visitadas durante a busca. A duração tabu inicia em 1, aumentando cada vez que uma repetição é encontrada e diminuindo quando as repetições desaparecem. “*Hashing*” e árvores binárias são utilizadas para identificar repetições.

Mais recentemente, a busca tabu vem sendo utilizada em conjunto com uma técnica denominada “*Path Relinking*”, que permite gerar novas soluções através da exploração de trajetórias que conectam soluções de elite, como ilustrado na Figura 3.19. Regras de escolha são especialmente projetadas para incorporar atributos contido na solução alvo.

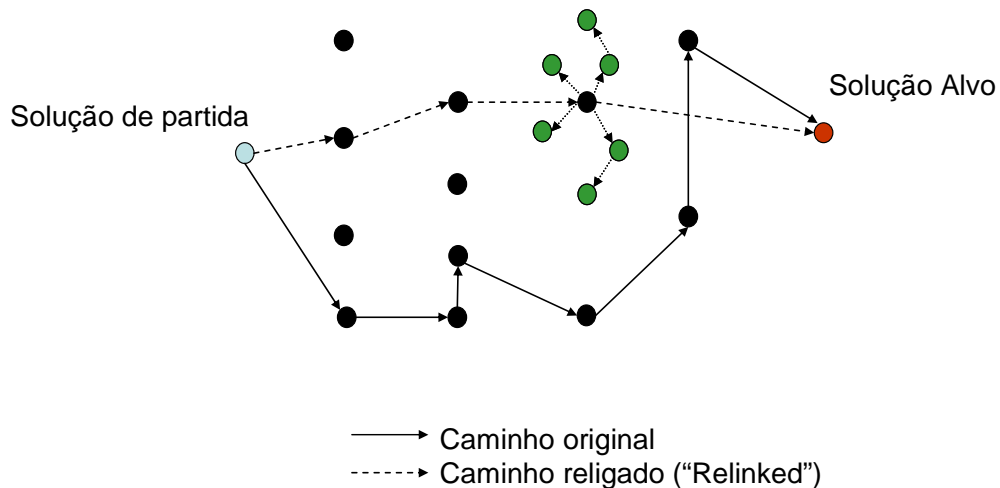


Figura 3.19: *Path Relinking*

Fonte: adaptado de Glover e Laguna (1997)

Cadeias de ejeção são mecanismos para combinar estratégias de diversificação e intensificação através de complexos movimentos compostos a partir de movimentos mais simples. Tais métodos têm possibilitado obter os melhores resultados para problemas difíceis, em especial aqueles que operam sob grafos. Para maiores detalhes sugere-se consultar os trabalhos de Glover (1996), Pesche e Glover (1997) e Rego e Roucariol (1996).

Greistorfer e Rego (2006) propuseram uma nova abordagem, denominada “*filter-and-fan*” que consiste em um mecanismo de busca em vizinhança com profundidade variável, sendo, segundo os autores, uma alternativa mais simples de implementar, além de mais eficiente que as cadeias de ejeção, tendo sido aplicada com sucesso a um problema de localização.

Outras características e mecanismos, além dos descritos acima, podem ser encontrados nas inúmeras aplicações de busca tabu descritas na literatura; entretanto, foge ao escopo do presente trabalho uma discussão aprofundada desses aspectos. Ao leitor interessado, sugerem-se os trabalhos de Hertz *et al.* (1997), Glover e Laguna (1993), e Glover e Laguna (1997) que apresentam, além da descrição do funcionamento da metaheurística, uma ampla ilustração do seu potencial, ao descrever problemas aos quais a busca tabu foi aplicada com sucesso, assim como, a edição especial do *Computers and Operations Research* (v.36, n.9, 2006).

3.3 GRASP

O GRASP (“*Greedy Randomized Adaptative Search Procedure*”), que pode ser traduzido como *procedimento de busca adaptativo randomizado guloso*, é uma metaheurística iterativa do tipo multi-início (“*multi-start*”) proposta por Feo e Resende (1989, 1995), em que cada iteração compreende duas etapas: construção e busca local. Na etapa de construção, determina-se uma solução viável elemento a elemento, cuja vizinhança é explorada até que um ponto de ótimo local seja encontrado durante a etapa de busca local.

O pseudo-código da Figura 3.20 ilustra o funcionamento geral do GRASP, em que um número de iterações igual a *Max_Iterações* são executadas e *Semente* é utilizado como a semente inicial para um gerador de números pseudo-aleatórios.

```

procedimento GRASP(Max_Iterações, Semente);
início
  Read_Input()
  para k=1 até Max_Iterações faça
    Solução ← Greedy_Randomized_Construction(Semente);
    Solução ← Local_Search(Solução);
    Atualiza_Solução(Solução, MelhorSolução);
  fim;
  Retorna com MelhorSolução
fim GRASP;

```

Figura 3.20: Pseudo-código da metaheurística GRASP

Fonte: adaptado de Resende e Ribeiro (2003)

A Figura 3.21 exhibe o pseudo-código da fase de construção. A cada iteração dessa fase, o conjunto de elementos candidatos é formado por todos os elementos que podem ser adicionados à solução parcial sendo gerada sem torná-la inviável. A seleção do próximo elemento a ser adicionado é determinada através da avaliação de todos os candidatos, segundo uma função de avaliação gulosa (“*greedy*”).

Essa função gulosa representa o acréscimo incremental na função de custo ocasionado pela incorporação de um elemento na solução em construção. A avaliação dos elementos através dessa função leva à criação de uma lista restrita de candidatos (LRC), formada pelos melhores elementos, isto é, aqueles cuja incorporação na solução parcial corrente resulta nos menores custos incrementais (este é o aspecto guloso do algoritmo, segundo Resende e Ribeiro, 2003).

```

procedimento Greedy_Randomized_Construction (Semente);
início
    Solução  $\leftarrow \phi$ ;
    Avaliar os custos incrementais dos elementos candidatos;
    enquanto Solução não completa faça
        Construir a lista restrita de candidatos (LRC);
        Selecionar aleatoriamente um elemento s da LRC
        Solução  $\leftarrow$  Solução  $\cup$  {s};
        Reavaliar os custos incrementais;
    fim;
    Retorna com Solução
fim Greedy_Randomized_Construction;

```

Figura 3.21: Pseudo-código da fase de construção

Fonte: adaptado de Resende e Ribeiro (2003)

O elemento a ser incorporado na solução parcial é selecionado randomicamente dentre aqueles na lista restrita de candidatos (este é o aspecto probabilístico da heurística). Uma vez que o elemento selecionado é adicionado à solução parcial, a lista de candidatos é atualizada e os custos incrementais recalculados (este é o aspecto adaptativo da heurística).

Segundo Resende e Ribeiro (2003), as soluções geradas pela fase de construção gulosa e randômica não são necessariamente ótimas, mesmo em relação a vizinhanças simples. A fase de busca local opera de maneira iterativa, substituindo a solução corrente por uma solução melhor encontrada na sua vizinhança. Ela termina quando nenhuma solução melhor puder ser encontrada na vizinhança da solução corrente. Os autores destacam que a efetividade de um procedimento de busca local depende de vários aspectos, entre os quais a estrutura de vizinhança, o método de busca na vizinhança, a rápida avaliação da função de custo para os vizinhos, e a própria solução inicial. A fase de construção exerce um papel importante em relação a esse último aspecto, permitindo obter soluções iniciais de alta qualidade para a busca local. A busca em vizinhança pode ser implementada considerando tanto uma estratégia de encontrar a melhor solução ou a primeira solução que melhore a solução corrente. No caso da melhor solução, toda a vizinhança da solução corrente é investigada, e a solução corrente é substituída pelo melhor vizinho encontrado; já na primeira solução, a troca é feita quando se encontrar o primeiro vizinho melhor que a solução corrente. De acordo com os autores, ambas as estratégias levam à mesma solução final, porém os tempos de processamento são normalmente menores quando a estratégia do primeiro melhor é usada. Os autores também observaram que a convergência prematura para um mínimo local é mais comum de ocorrer quando utilizada a estratégia do melhor vizinho.

3.3.1 Construção da Lista Restrita de Candidatos (LRC)

De acordo com Resende e Ribeiro (2003), uma das características atraentes do GRASP é a sua facilidade de implementação. Poucos parâmetros precisam ser definidos e ajustados. Dessa forma, seu desenvolvimento pode ser focado na implementação de estruturas de dados eficientes a fim de assegurar iterações rápidas. O GRASP possui dois parâmetros principais: um relacionado ao critério de parada (dado pelo número máximo de iterações) e outro relacionado à qualidade dos elementos da lista restrita de candidatos.

Para descrever como é a construção da lista restrita de candidatos, considera-se um problema de minimização, sem perda de generalidade, cujo conjunto de variáveis de decisão é dado por $E = \{1, \dots, n\}$; o conjunto de possíveis soluções viáveis por $F \subseteq 2^E$; e a função objetivo $f: 2^E \rightarrow \mathfrak{R}$. Denota-se $c(e)$ o custo incremental da incorporação do elemento $e \in E$ na solução em construção. Em qualquer iteração do GRASP, c^{\min} e c^{\max} representam o menor e o maior custo incremental, respectivamente.

A LRC é composta de elementos $e \in E$ que possuem os melhores (menores) custos incrementais $c(e)$. Essa lista pode ser limitada (restringida) pelo número de candidatos (ou seja, baseada na cardinalidade) ou pela qualidade dos mesmos. No primeiro caso, escolhem-se os p melhores elementos em termos de custos incrementais.

Segundo Resende e Ribeiro (2003), a LRC está relacionada a um parâmetro de gatilho (“*threshold*”) $\alpha \in \{0, 1\}$. Assim, a lista restrita de candidatos é composta por todos os elementos $e \in E$ que podem ser adicionados à solução parcial sendo construída sem torná-la inviável, e cuja qualidade seja superior ao valor gatilho, ou seja: $c(e) \in [c^{\min}, c^{\min} + \alpha(c^{\max} - c^{\min})]$. Para $\alpha = 0$, a heurística de construção torna-se puramente gulosa, enquanto que para $\alpha = 1$ a construção torna-se randômica. O pseudo-código mostrado na Figura 3.22 é um refinamento do pseudo-código da Figura 3.21. Ele explicita o controle do parâmetro α em relação ao grau de aleatoriedade do algoritmo.

```

procedimento Greedy_Randomized_Construction ( $\alpha$ , Semente);
início
  Solução  $\leftarrow \emptyset$ ;
  Inicializar o conjunto de candidatos:  $C \leftarrow E$ 
  Avaliar os custos incrementais  $c(e)$  para todos  $e \in C$ ;
  enquanto  $C \neq \emptyset$  faça
     $c^{min} \leftarrow \min\{c(e) \mid e \in C\}$ ;
     $c^{max} \leftarrow \max\{c(e) \mid e \in C\}$ ;
     $LRC \leftarrow \{e \in C \mid c(e) < c^{min} + \alpha(c^{max} - c^{min})\}$ 
    Selecionar aleatoriamente um elemento  $s$  da LRC
    Solução  $\leftarrow$  Solução  $\cup \{s\}$ ;
    Reavaliar os custos incrementais  $c(e)$  para todos  $e \in C$ ;
  fim;
  Retorna com Solução
fim Greedy_Randomized_Construction;

```

Figura 3.22: Pseudo-código refinado da fase de construção

Fonte: adaptado de Resende e Ribeiro (2003)

O GRASP pode ser visto como uma técnica de amostragem repetitiva. Cada iteração gera uma solução amostral de uma distribuição desconhecida, cuja média e variância são funções da natureza restritiva da LRC. Por exemplo, se a LRC se restringe a um único elemento, então a mesma solução é gerada em todas as iterações; a variância da distribuição vai ser zero e a média igual ao valor da solução gulosa. Se a LRC tiver mais elementos, então muitas diferentes soluções podem ser geradas, implicando uma maior variância. Tendo em vista a (pequena) influência da aleatoriedade, a solução média deve ser pior. Entretanto, o valor da melhor solução encontrada supera o valor médio e muitas vezes é ótimo. Os autores apresentam diversos resultados de experimentos conduzidos que comprovam essa tese.

Maiores detalhes sobre o GRASP podem ser encontrados em Feo e Resende (1989, 1995). Resende e Werneck (2004) descrevem a aplicação, ao problema das p -medianas, de uma heurística híbrida inspirada em GRASP, em que foram testadas vários métodos construtivos com características randômicas a fim de selecionar as p instalações, tendo sido obtidos ótimos resultados. Resende e Werneck (2005) tratam do problema de localização de instalações não capacitado de maneira similar ao problema das p -medianas. Encontram-se na literatura trabalhos que utilizam GRASP em conjunto com “*path relinking*”, uma estratégia de intensificação que explora trajetórias que ligam soluções de elite (Li, Pardalos e Resende, 1994; Resende e Ribeiro, 2005). Alguns conceitos do GRASP foram utilizados por Feriancic (2005), e Ahuja e Cunha (2005).

3.4 SIMULATED ANNEALING

O “*Simulated Annealing*” (SA), também conhecido como têmpera (ou recozimento) simulada, é uma metaheurística proposta por Kirkpatrick *et al.* (1983), inspirada no processo térmico de resfriamento de um material, com a virtude de permitir escapar de um ótimo local através da aceitação de movimentos que pioram a solução corrente. O termo “*simulated annealing*” deve-se à analogia com a termodinâmica, mais especificamente um processo físico, denominado recozimento (“*annealing*”), onde um material é aquecido até atingir um estado líquido e em seguida resfriado. Em altas temperaturas, os átomos tem liberdade de se moverem livremente. A medida em que a temperatura diminui, os átomos tendem a se cristalizar num sólido. Se a temperatura é reduzida muito rapidamente, o sólido cristalizado não apresenta forma. Se o resfriamento ocorre vagarosamente, maiores são as chances de se formar um cristal perfeito, ou seja, numa estrutura cristalina de energia mínima global. Para apresentar bons resultados, um resfriamento depende de uma temperatura inicial alta e de um resfriamento lento, a fim de evitar uma estrutura irregular e fraca, com alta energia em decorrência do esforço interno despendido.

O SA teve sua origem no algoritmo de Metropolis *et al.* (1953), que simula a evolução de um sólido em um banho térmico até que um equilíbrio térmico tenha sido atingido através da geração de uma seqüência de estados do sólido.

Partindo de uma solução inicial viável completa S , o SA gera aleatoriamente uma solução vizinha S' . Seja $\Delta = f(S') - f(S)$ a variação de valor da função objetivo ao se mover de S para S' . Considerando-se um problema de minimização, se $\Delta < 0$, ou seja S' é melhor que S , automaticamente a nova solução é aceita; caso contrário, i.e. $\Delta \geq 0$, a solução candidata S' poderá ser aceita com uma probabilidade $e^{-\Delta/T}$, onde T é o parâmetro do método, chamado de *temperatura*, e que regula a probabilidade de aceitação de soluções piores.

A temperatura T assume inicialmente um valor elevado T_0 , sendo gradualmente resfriada. A idéia é aceitar inicialmente soluções piores buscando escapar de ótimos locais; à medida em que T vai diminuindo, o comportamento do método aproxima-se de um método de descida (“*descent method*”), uma vez que diminui-se a probabilidade de aceitação de movimentos de piora. A temperatura inicial T_0 deve ser alta, de modo a permitir uma probabilidade alta de aceitação de movimentos; por outro lado não pode ser excessivamente alta a ponto de levar a

busca aleatória. Esse valor de T_0 pode ser determinado e ajustado testando-se o algoritmo por um certo número de iterações até que seja obtida uma taxa de aceitação de soluções piores entre 40 e 60% (alguns autores sugerem até 90%).

Usualmente a temperatura deve decrescer até zero, sendo esse o critério de parada usual do SA. Entretanto, tal critério pode levar a tempos de processamento muito elevados; na prática, basta reduzir a temperatura até um nível que a seja virtualmente impossível aceitar um movimento de piora.

Para resfriamento da temperatura, o esquema mais comum é utilizando a expressão: $T = \alpha * T$. A maioria dos autores sugere que α deve estar na faixa [0.8-0.99], com melhores resultados no extremo superior da faixa. Naturalmente, quanto maior o valor de α , maior é o número de iterações. Deve-se executar um número constante de iterações em cada temperatura T . Alternativamente, pode-se alterar dinamicamente o número de iterações ao longo do processamento; em temperaturas mais baixas, é importante que um maior número de iterações sejam permitidas de modo que a região de ótimo local possa ser completamente explorada; em temperaturas mais altas, o número de iterações pode ser menor.

Alguns autores sugerem um esquema de reaquecimento (“*reannealing*”), quando a busca deixa de convergir: $T = T/(1 - \gamma T)$, onde γ é um valor muito reduzido.

Um aspecto crítico do desempenho do SA é a função de avaliação, uma vez que a mesma é aplicada a cada iteração. Em muitos casos, é o componente mais demorado do algoritmo; segundo Johnson *et al.* (1991), o critério de aceitação pode consumir 1/3 do tempo de processamento. Assim, buscam-se esquemas que simplifiquem a estimativa dessa função. O autor experimentou com a seguinte função: $P(\Delta) = 1 - \Delta/T$. Essa função aproxima-se da exponencial e é muito mais rápida. Para incrementar ainda mais o desempenho, pode-se construir uma tabela de consulta (“*hash table*”) de valores Δ/T ; os valores Δ/T calculados podem ser arredondados para o inteiro mais próximo e então usados na consulta à tabela. Segundo Johnson (1991), isso melhora em cerca de 30% o desempenho do algoritmo sem perdas significativas de qualidade.

Entre as principais virtudes do “*simulated annealing*” estão a sua simplicidade de implementação, além de se demonstrar que converge para a solução ótima (Downsland, 1993). Por outro lado, dependendo da velocidade de resfriamento, que deve ser baixa, o

número de iterações para se atingir a solução ótima pode ser elevado, potencialmente exponencial; adicionalmente, é pouco inteligente, pois não utiliza informações do histórico da busca para guiar o algoritmo.

3.5 BUSCA EM VIZINHANÇA DE GRANDE PORTE

Algoritmos de busca em vizinhança (também conhecidos como algoritmos de busca local) pertencem a uma classe geral de algoritmos de melhoria (i.e, algoritmos que partem de uma solução inicial viável e tentam melhorá-la iterativamente) onde, a cada iteração varre-se a vizinhança da solução corrente em busca de uma solução melhor, como ilustrado na Figura 3.23.

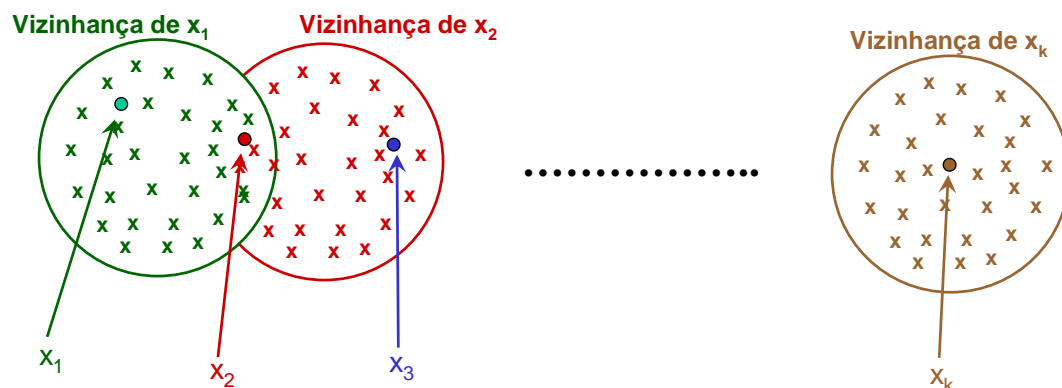


Figura 3.23: Busca em vizinhança

Fonte: adaptado de Ahuja *et al.* (2000)

Os chamados Métodos de Busca em Vizinhança de Grande Porte (BVGP) (do inglês “*Large Scale Neighborhood Search Methods*”) correspondem a algoritmos de busca em vizinhança em que o tamanho da vizinhança é muito grande, possivelmente exponencial, tornando virtualmente impossível a enumeração de todas as soluções vizinhas de uma solução corrente e a sua avaliação (Ahuja *et al.*, 2002).

Segundo Ahuja *et al.* (2002), alguns métodos de sucesso e amplamente utilizados na pesquisa operacional podem ser vistos como técnicas de busca em vizinhança de grande porte. Por exemplo, se o método Simplex da programação linear for visto como um algoritmo de busca em vizinhança, então a geração de colunas pode ser classificada como um método de busca

em vizinhança de grande porte. As técnicas de aumento (“*augmentation*”) usadas para resolver inúmeros problemas de fluxo em rede também podem ser classificadas como métodos de busca em vizinhança de grande porte. O algoritmo de cancelamento de ciclo negativo (“*negative cost cycle cancelling algorithm*”) para resolver o problema de fluxo de custo mínimo, e o algoritmo de aumento de caminho (“*augmenting path algorithm*”) para resolver problemas de emparelhamento (“*matching*”) são dois exemplos.

De acordo com Ahuja, Orlin e Sharma (2001), um aspecto crítico na concepção de uma estratégia de busca em vizinhança consiste na escolha da estrutura da vizinhança, isto é, na maneira como a vizinhança é definida. Essa escolha define, em grande parte, se a estratégia de busca em vizinhança permitirá se obter soluções de boa qualidade ou soluções apenas inferiores, correspondentes a ótimos locais. Em geral, quanto maior a vizinhança, melhor é a qualidade das soluções ótimas locais, e maior a exatidão do método. Por outro lado, quanto maior a dimensão da vizinhança, maior o tempo para percorrê-la e examiná-la, a cada iteração. Considerando-se que em geral são realizados vários processamentos de um algoritmo de busca em vizinhança, partindo-se de diferentes soluções iniciais, um maior tempo de processamento por iteração leva a um menor número de processamentos por unidade de tempo. Por essa razão, uma vizinhança maior não necessariamente significa uma heurística melhor, exceto se essa vizinhança maior puder ser explorada de uma maneira bastante eficiente.

3.5.1 Busca Local: Uma Visão Geral

Inicialmente formaliza-se um problema de otimização combinatória e o conceito de vizinhança. Há várias maneiras diferentes de se representar um problema combinatório, todas baseadas em algum método para representar o conjunto de soluções viáveis (Ahuja *et al.*, 2002). Neste contexto, sejam os *conjuntos de soluções viáveis* representados como subconjuntos de um conjunto finito $E = \{1, 2, \dots, m\}$. Seja $F \subseteq 2^E$ o conjunto de todas as *soluções viáveis*, onde 2^E indica todos os subconjuntos de E . Um problema de otimização combinatória pode ser representado como:

$$\text{minimizar } \{f(S) : S \in F\} \quad (3.1)$$

Assume-se não ser possível enumerar todas as possíveis soluções do conjunto F , mas sim representá-las numa forma compacta de tamanho polinomial em m . Uma função de

vizinhança é dada por $N : F \rightarrow 2^E$. Cada $S \in F$ tem um subconjunto associado $N(S)$ de E chamado de *vizinhança* da solução S ; assume-se sem perda de generalidade que $S \in N(S)$. Diz-se que uma solução $S^* \in F$ é *localmente ótima* em relação à função de vizinhança N se $f(S^*) \leq f(S)$ para todo $S \in N(S^*)$. A vizinhança $N(S)$ é dita exponencial se seu tamanho, dado por $|N(S)|$, cresce exponencialmente com o aumento de m .

Assim, a busca em vizinhança de grande porte considera vizinhanças que são muito grandes para serem varridas na prática, possivelmente, mas não necessariamente, exponenciais (por exemplo, uma vizinhança com m^3 elementos pode ser muito grande, dependendo do valor de m).

3.5.2 Problemas de Particionamento

Ahuja *et al.* (2002) descrevem os métodos de Busca em Vizinhança de Grande Porte através da sua aplicação a problemas de particionamento (“*partitioning problems*”).

Seja $A = \{a_1, a_2, a_3, \dots, a_n\}$ um conjunto com n elementos. Os subconjuntos $S_1, S_2, S_3, \dots, S_K$, para algum inteiro K , definem uma partição de A se os mesmos são disjuntos e a sua união resulta em A . Essa partição pode ser representada como $S = \{S_1, S_2, S_3, \dots, S_K\}$, conforme ilustrado na Figura 3.24, em que os 20 elementos são separados em quatro conjuntos. Seja $c_k(S_k)$ o custo mínimo de um subconjunto S_k .

Em algumas aplicações, a função de custo $c_k(S_k)$ é fácil de ser avaliada, enquanto que em outras aplicações o custo depende do arranjo de mínimo custo para os elementos no subconjunto S_k . Por exemplo, quando S_k denota um conjunto de atendimentos alocados ao veículo k em um problema de roteirização, então $c_k(S_k)$ denota o mínimo custo do roteiro do veículo k para a entrega de todas as encomendas em S_k , o que requer a determinação do roteiro, conforme ilustrado na Figura 3.25

Assim, o problema de particionamento pode ser formalizado da seguinte forma:

$$\text{Minimizar } c(S) = \sum_{k=1}^K c_k(S_k), \text{ tal que } S = (S_1, S_2, S_3, \dots, S_K) \text{ é uma partição de } A. \quad (3.2)$$

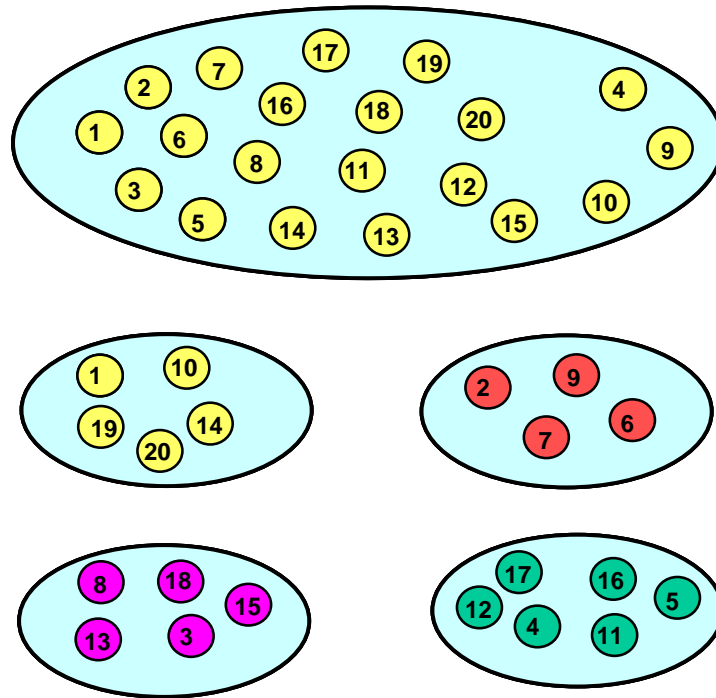


Figura 3.24: Problema de particionamento

Fonte: adaptado de Ahuja *et al.* (2000)

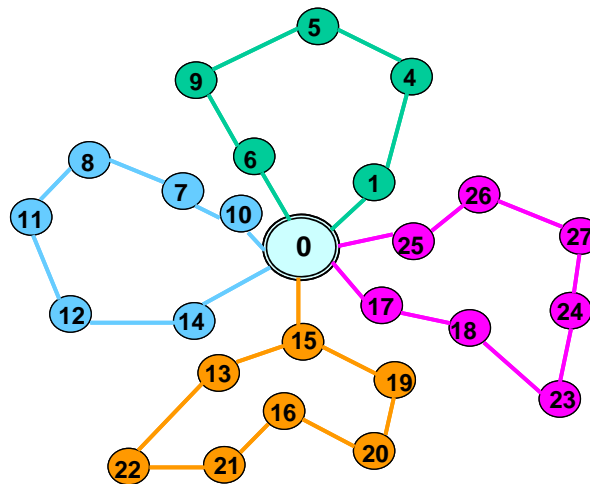


Figura 3.25: Problema de roteirização visto como um problema de particionamento

Fonte: adaptado de Ahuja *et al.* (2002)

Um problema de particionamento (3.2) é freqüentemente solucionado através de um processo de decisão em duas fases: inicialmente, os elementos de A devem ser alocados aos subconjuntos $S_1, S_2, S_3, \dots, S_K$; em seguida, cada subconjunto S_k deve ser configurado de maneira ótima em relação ao seu custo $c_k(S_k)$. Nesse problema de particionamento considera-se que a função objetivo possa ser calculada separadamente para cada subconjunto, isto é, o

custo $c_k(S_k)$ não depende da maneira como os demais elementos de S são alocados aos demais subconjuntos, nem tampouco depende de como os demais subconjuntos são configurados.

Problemas de particionamento estão entre os problemas de otimização combinatória mais difíceis de serem resolvidos até a otimalidade, abrangendo classes de problemas que não podem ser resolvidos quando o número de elementos é superior a 50. Problemas maiores são resolvidos utilizando heurísticas, em geral através de métodos de busca em vizinhança e suas variantes.

Um método de busca em vizinhança pode basear-se, por exemplo, em trocas aos pares, ou seja, dois elementos são trocados entre os dois subconjuntos aos quais pertencem, como mostrado na Figura 3.26. Essa vizinhança de trocas em pares tem tipicamente tamanho $\Theta(n^2)$, isto é, sua ordem de complexidade é $O(n^2)$. Podem ser consideradas múltiplas trocas, em que elementos de diversos conjuntos podem ser trocados.

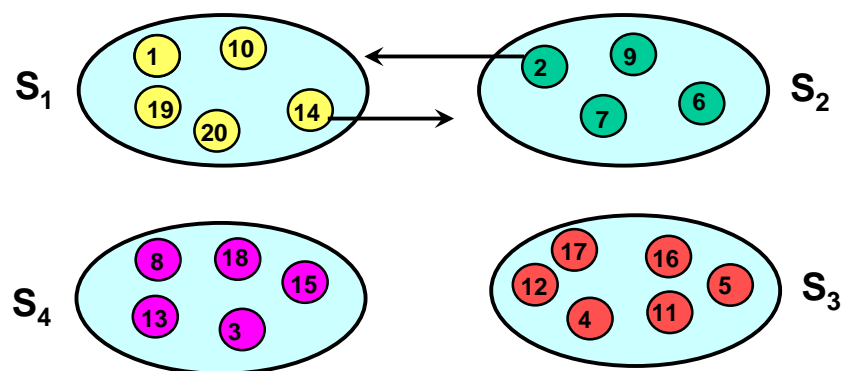


Figura 3.26: Vizinhança do tipo 2-troca

Fonte: adaptado de Ahuja *et al.* (2000)

3.5.3 Vizinhanças Definidas como Ciclos

Uma *troca cíclica* (“*cyclic exchange*”) pode ser definida por uma seqüência de elementos $a_1 - a_2 - a_3 - \dots - a_r - a_1$, onde os elementos $a_1, a_2, a_3, \dots, a_r$ pertencem a diferentes subconjuntos, como ilustrado na Figura 3.27, em que a troca cíclica corresponde ao ciclo 14-7-12-18-14, formado por elementos que pertencem a diferentes subconjuntos.

Seja $S[a_k]$ o conjunto ao qual pertence o elemento a_k . Dessa forma, a troca cíclica $a_1 - a_2 - a_3 - \dots - a_r - a_1$ implica as seguintes mudanças: o elemento a_1 é movido do

subconjunto $S[a_1]$ para o subconjunto $S[a_2]$, o elemento a_2 de $S[a_2]$ para $S[a_3]$, e assim por diante; finalmente, o elemento a_r é movido de $S[a_r]$ para $S[a_1]$, completando um ciclo de mudanças. No exemplo mostrado na Figura 3.27, o elemento 14 deixa o conjunto S_1 e vai para S_2 ; o elemento 7 vai de S_2 para S_3 ; o elemento 12 de S_3 para S_4 ; e o elemento 18 vai de S_4 para S_1 .

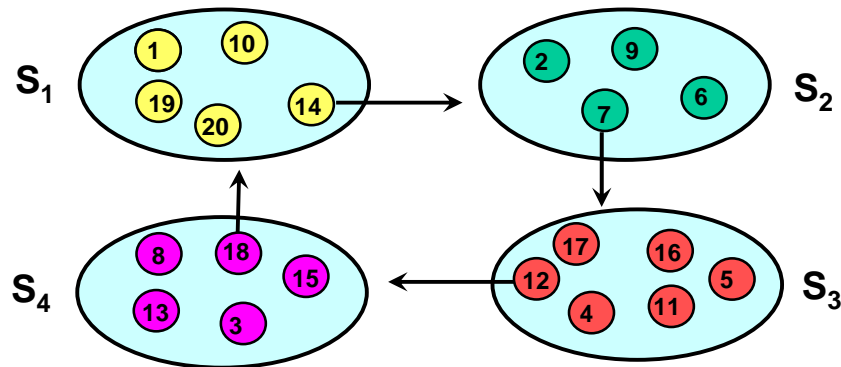


Figura 3.27: Troca cíclica (“*cyclic exchange*”)

Fonte: adaptado de Ahuja *et al.* (2000)

Uma *troca de caminho* (“*path exchange*”) é definida por uma sequência de nós $a_1 - a_2 - a_3 - \dots - a_r$, diferindo da troca cíclica pelo fato do último elemento a_r não ser movido de $S[a_r]$ para $S[a_1]$, como ilustrado na Figura 3.28, em que o elemento 12 vai de S_3 para S_4 e nenhum elemento é movido de S_4 . Deve-se notar que uma troca de caminho diminui a cardinalidade do subconjunto $S[a_1]$ e aumenta a cardinalidade do subconjunto $S[a_r]$, permanecendo inalteradas as cardinalidades dos demais subconjuntos. A troca cíclica, por outro lado, não altera a cardinalidade de nenhum subconjunto.

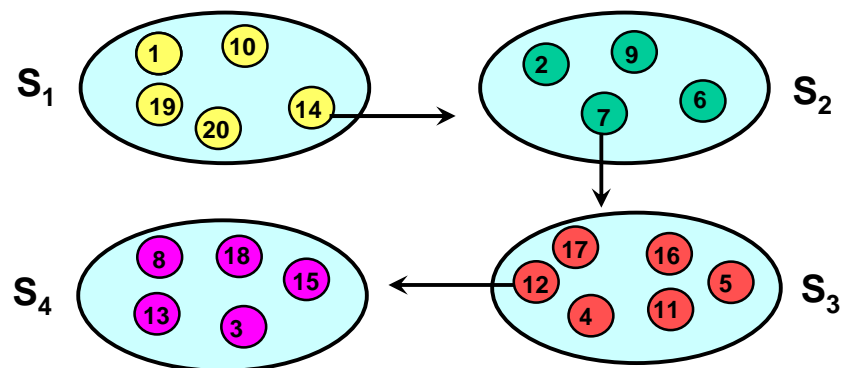


Figura 3.28: Troca de caminho (“*path exchange*”)

Fonte: adaptado de Ahuja *et al.* (2000)

Segundo Ahuja *et al.* (2000), pode-se transformar o problema de particionamento de tal modo que uma troca de caminho (“*path exchange*”) para o problema original seja transformada numa troca cíclica (“*cyclic exchange*”) para o problema modificado: basta criar um subconjunto fictício (“*dummy*”), assim como um nó fictício para cada subconjunto.

Para um dado valor de K , o tamanho da vizinhança definida através de múltiplas trocas corresponde a $O(n^K)$ vizinhos. A fim de ilustrar como a vizinhança da troca cíclica é significativamente maior que $O(n^2)$, considere-se, por exemplo, $n = 100$ e $K = 10$. Nesse contexto, a vizinhança correspondente à troca aos pares tem cerca de 9 mil elementos, ao passo que a vizinhança cíclica possui mais de 50 trilhões de elementos.

É de se esperar (e isso foi verificado experimentalmente para diversos problemas por Ahuja, Orlin e Sharma, 2001a) que soluções ótimas locais obtidas por meio de múltiplas trocas sejam, em média, superiores às soluções obtidas por trocas aos pares. Entretanto, uma vez que o tamanho da vizinhança em trocas múltiplas cresce exponencialmente com o tamanho do problema, torna-se necessário um método eficiente para encontrar um vizinho de menor custo (ou o vizinho de mínimo custo) na vizinhança. Thompson e Orlin (1989) e Thompson e Psaraftis (1993) desenvolveram um método que utiliza o conceito de *grafos de melhoria*.

3.5.4 Grafo de Melhoria

Um *grafo de melhoria* (“*improvement graph*”) para uma vizinhança com múltiplas trocas é definido para uma partição viável \mathcal{S} , e representado por $G(\mathcal{S})$. Como anteriormente, $S[a_j]$ indica o subconjunto que contém o elemento a_j .

O grafo $G(\mathcal{S})$ é um grafo direcionado com n nós, onde cada nó i corresponde a um elemento $a_i \in \mathcal{S}$. Um arco direcionado (i, j) em $G(\mathcal{S})$ representa o elemento a_i que deixa seu subconjunto atual $S[a_i]$ e é inserido no subconjunto que contém o elemento a_j , ou seja, $S[a_j]$; simultaneamente, o elemento a_j deixa $S[a_j]$, conforme ilustrado na Figura 3.29.

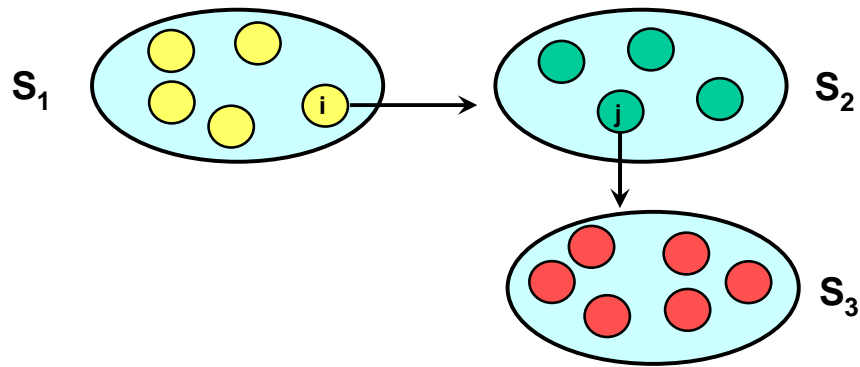


Figura 3.29: Arco (i, j) no grafo de melhoria

Fonte: adaptado de Ahuja *et al.* (2000)

Para construir $G(S)$, consideram-se todos os pares de elementos a_i e a_j em S . O arco (i, j) é adicionado a $G(S)$ se: (i) os elementos a_i e a_j pertencerem a diferentes subconjuntos, isto é $S[a_i] \neq S[a_j]$; e (ii) o novo subconjunto $\{a_i\} \cup S[a_j] \setminus \{a_j\}$ for viável. Define-se o custo c_{ij} no arco (i, j) como $c(\{a_i\} \cup S[a_j] \setminus \{a_j\}) - c(S[a_j])$, ou seja, a diferença entre o custo no subconjunto $S[a_j]$ após a substituição do elemento a_j por a_i e o custo anterior em $S[a_j]$.

No caso do arco (i, k) ligando um elemento a_i a um elemento fictício a_k de um outro subconjunto, o custo c_{ik} é igual a $c(\{a_i\} \cup S[a_k]) - c(S[a_k])$, ou seja, a diferença entre os custos no subconjunto $S[a_k]$ após e antes da inserção do elemento a_i , uma vez que o elemento fictício a_k (de cada subconjunto) tem a finalidade específica de permitir que um elemento a_i seja movido para um novo subconjunto $S[a_k]$ sem que nenhum elemento saia desse conjunto.

Analogamente, no caso do arco (k, i) ligando um elemento fictício a_k a um elemento a_i de um outro subconjunto, o custo c_{ki} é igual a $c(S[a_i] \setminus \{a_i\}) - c(S[a_i])$, ou seja, a diferença entre o custo no subconjunto $S[a_i]$ após a retirada do elemento a_i e o custo anterior em $S[a_i]$, uma vez que, neste caso, a inserção do elemento fictício a_k não ocasiona nenhuma mudança de custo no subconjunto $S[a_i]$.

Denomina-se W um ciclo direcionado no grafo de melhoria $G(S)$ se os elementos em S correspondentes aos nós em W pertencem a diferentes subconjuntos.

Thompson e Orlin (1989) apresentaram o seguinte resultado que demonstra a relação entre trocas cíclicas e o grafo de melhoria:

Lema 1: Há uma correspondência um-a-um entre trocas múltiplas em S e os ciclos direcionados (de subconjuntos desconexos) no grafo de melhoria $G(S)$, e ambos têm o mesmo custo.

Assim, uma troca múltipla que reduza o custo em S corresponde a um ciclo de subconjuntos desconexos com custo negativo em $G(S)$, como ilustrado na Figura 3.30. Embora identificar um ciclo negativo (não restrito) em um grafo seja possível em tempo polinomial (Ahuja, Magnanti e Orlin, 1993), o problema de encontrar um ciclo negativo em conjuntos desconexos é um problema NP-completo (Thompson e Orlin, 1989).

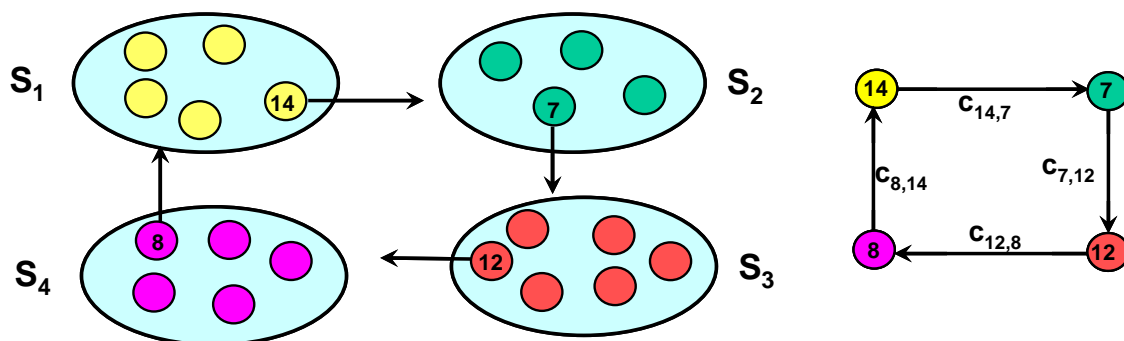


Figura 3.30: Ciclo direcionado no grafo de melhoria

Fonte: adaptado de Ahuja *et al.* (2000)

Define-se um *ciclo válido* (“*valid cycle*”) como um ciclo de custo negativo de um subconjunto desconexo em $G(S)$. Por exemplo, na Figura 3.30 o ciclo 14-7-12-8-14 é um ciclo válido se $(c_{14,7} + c_{7,12} + c_{12,8} + c_{8,14}) < 0$.

3.5.5 Algoritmo de busca em vizinhança de grande porte

Conforme apontam Ahuja *et al.* (2003b), um ciclo direcionado em $G(S)$ pode ser obtido a partir de um caminho direcionado, simplesmente ligando seus pontos extremos. Pode-se enumerar todos os ciclos direcionados em um grafo através da enumeração de todos os caminhos direcionados e ligar seus pontos extremos (desde que tais arcos existam). Entretanto, a desvantagem dessa abordagem é que operações redundantes podem ser realizadas, e um mesmo ciclo pode ser obtido através de diferentes caminhos. A fim de se evitar tais operações redundantes, considera-se o seguinte lema proposto por Lin and Kernighan (Ahuja *et al.*, 2003b):

Lema 2: Se $W = i_1-i_2--i_3-...-i_r-i_1$ é um ciclo direcionado de custo negativo, então existe um nó i_h nesse ciclo tal que cada caminho parcial direcionado (i_h-i_{h+1}) , $(i_h-i_{h+1}-i_{h+2})$, $(i_h-i_{h+1}-...-i_{h+k-1})$ (onde os índices são módulos de r) é um caminho direcionado de custo negativo.

A enumeração de todos os caminhos válidos tem ordem de complexidade exponencial $O(n/K)^K$ no pior caso, o que inviabiliza o uso de algoritmos exatos, e sugere que a atenção seja dedicada a heurísticas. Diversos algoritmos foram propostos para identificar ciclos válidos (Ahuja, Orlin e Sharma, 2001a, 2001b; Ahuja, Boland e Dumitrescu, 2001), entre os quais programação dinâmica, conforme proposto por Ahuja *et al.* (2003b). Primeiro obtém-se todos os caminhos válidos de comprimento unitário (isto é, contendo apenas um arco), que são usados para obter caminhos válidos de comprimento 2 (isto é, contendo exatamente dois arcos). Repete-se o processo até que tenham sido obtidos todos os caminhos válidos de comprimento R (onde $R \leq K$) para algum comprimento R ou até que um ciclo válido tenha sido encontrado. Maiores detalhes do algoritmo, incluindo como os caminhos válidos são enumerados implicitamente e os dominados eliminados, podem ser encontrados em Ahuja *et al.* (2003b).

Nos testes realizados, as melhores heurísticas são capazes de identificar ciclos válidos em subconjuntos desconexos, correspondentes a redes com milhares de arcos, em tempos de processamento inferiores a um segundo (Ahuja, Orlin e Sharma, 2001a).

Desta forma, pode-se definir a estratégia geral de solução para um problema de particionamento através de um algoritmo de busca em vizinhança de grande porte, conforme ilustrado na Figura 3.31.

<p>procedimento busca VLSN; início obter uma solução viável S para o problema; construir o grafo de melhoria $G(S)$; enquanto houver ciclo válido em $G(S)$ faça obter um ciclo válido W em $G(S)$; realizar as trocas múltiplas correspondente a W; atualizar S e $G(S)$; fim; fim;</p>

Figura 3.31: Algoritmo básico de busca VLSN

Fonte: adaptado de Ahuja *et al.* (2000)

Inicialmente obtém-se uma solução viável S para o problema de particionamento, utilizando alguma heurística de construção ou algum outro método específico. Em seguida, constrói-se um grafo de melhoria $G(S)$. Através de um algoritmo de detecção de ciclos, identifica-se um ciclo válido em $G(S)$. Esse ciclo leva a uma troca múltipla em relação a S , através da qual é obtida uma solução de menor custo S' para o problema de particionamento, ou seja, S é substituído por S' . O grafo de melhoria $G(S)$ é atualizado. O procedimento é repetido, até que $G(S)$ não contenha mais ciclos válidos; nesse ponto, a solução S corresponde a um ótimo local com relação a uma vizinhança de múltiplas trocas. Ahuja *et al.* (2002a) sugerem aplicar esse algoritmo diversas vezes, considerando diferentes soluções iniciais, de forma a se obterem soluções de boa qualidade.

3.5.6 Exemplos de Aplicação

O procedimento genérico de busca em vizinhança de grande porte apresenta algumas características que são específicas de cada problema, como a construção do grafo de melhoria, que difere para cada problema de particionamento.

Por exemplo, a determinação do custo de um arco num grafo de melhoria para o problema de roteirização de veículos requer a obtenção da solução de um problema de caixeiro viajante. No outro extremo, para o problema de atribuição generalizada (“*generalized assignment problem*”), os custos nos arcos do grafo de melhoria podem ser calculados em tempos $O(1)$ em média. Enquanto os procedimentos de cálculo dos custos nos arcos são específicos de cada problema, a busca de um ciclo válido pode ser a mesma para todos os problemas de particionamento. Assim, o método de busca em vizinhança pode ser aplicado em diferentes problemas de particionamento, com aproveitamento de alguns algoritmos básicos.

Deve-se destacar que, embora na construção do grafo de melhoria descrita acima se tenha estipulado que apenas um elemento é movimentado em cada subconjunto, o método pode facilmente ser modificado de tal forma a permitir a transferência simultânea de mais de um elemento de um subconjunto para outro, como mostrado na Figura 3.32 para o problema de árvore de cobertura mínima capacitada (“*capacitated minimum spanning tree*”), em que cada nó tem uma demanda a ser atendida a partir de um nó central 0, e existe uma capacidade máxima K nos arcos da rede. Esse é um problema muito comum na configuração de redes de comunicação (Ahuja *et al.*, 2000).

No exemplo da Figura 3.32, as sub-árvores formadas pelos nós (16, 21) e (18, 23, 24) são trocadas, uma vizinhança mais robusta e complexa que trocar apenas nós das árvores. As trocas cíclicas são genéricas e podem envolver mais de duas sub-árvores simultaneamente, como mostrado em Ahuja *et al.* (2000).

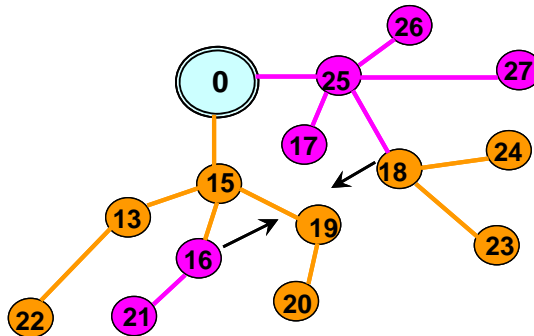


Figura 3.32: Troca cíclica de duas sub-árvores

Fonte: adaptado de Ahuja *et al.* (2000)

Entretanto, nem sempre essa idéia permite explorar de maneira eficiente a vizinhança de uma solução, uma vez que movimentos mais complexos podem ser requeridos e são difíceis de serem implementados, como é o caso do problema das múltiplas mochilas binárias com K restrições (“*K-constraint multiple knapsack problem*”), uma generalização do problema das múltiplas mochilas (“*multiple knapsack problem*”), tratado por Ahuja e Cunha (2005). Nesse problema, existem n objetos e m mochilas. Cada objeto tem associado um lucro e K tipos de peso; cada mochila apresenta K diferentes tipos de capacidade (ao invés de uma só capacidade, como no problema das múltiplas mochilas binárias). Busca-se selecionar quais objetos inserir em quais mochilas de modo que o lucro total dos itens selecionados seja máximo, e as K restrições de capacidade em cada uma das m mochilas sejam respeitadas.

Esse problema pode ser visto como um problema de particionamento, em que os n objetos devem ser particionados em $m+1$ conjuntos, correspondentes às m mochilas mais uma mochila especial fictícia que corresponde aos objetos não selecionados.

Diferentemente de outros problemas de particionamento, para este problema a contribuição (lucro) de um objeto j independe da partição (mochila) ao qual está alocado. Em outras palavras, um item selecionado contribui da mesma maneira para a função objetivo, independentemente da mochila ao qual pertence. Assim, um ciclo disjunto envolvendo

somente objetos inseridos em mochilas não é um ciclo válido, pois não representa uma melhoria na solução corrente, como mostrado na Figura 3.33a; para ser um ciclo válido é necessário que um objeto i não selecionado substitua um objeto l selecionado, desde que o lucro do objeto i seja maior que o lucro do objeto l ($p_i > p_l$), como mostrado na Figura 3.33b.

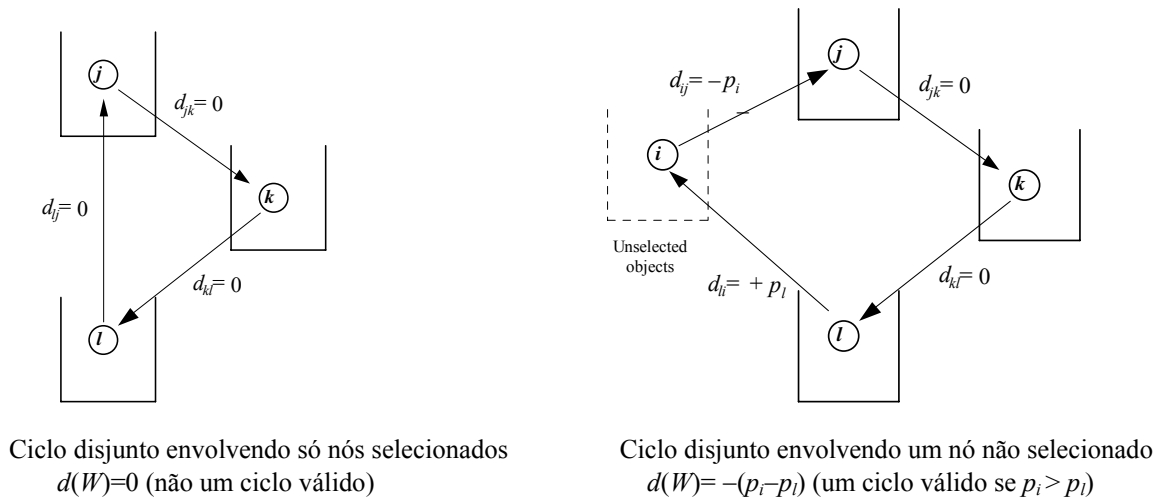


Figura 3.33: Dois possíveis ciclos para o problema das múltiplas mochilas

Fonte: Ahuja e Cunha (2005)

Assim, um item pode fazer parte da solução, isto é, ser um dos itens selecionados, mas estar inserido na partição (mochila) errada, o que impede que outros itens possam ser inseridos; isso requer que os itens sejam rearranjados nas mochilas, de forma a abrir espaço para que novos itens possam ser inseridos, mesmo que esses movimentos não representem nenhuma melhoria da solução.

Dada a complexidade desses movimentos, Ahuja e Cunha (2005) propuseram uma heurística de busca em vizinhança de grande porte que usa os conceitos de GRASP para gerar novas soluções iniciais a partir da melhor solução corrente. Mais especificamente, em cada iteração os primeiros r objetos são mantidos na solução corrente S (onde r é um número aleatório entre 0 and $n/2$) e os demais $(n - r)$ objetos removidos das suas partições correntes (mochilas).

Soluções iniciais foram obtidas tanto utilizando uma heurística gulosa simples, como através de relaxação Lagrangeana, como descrito anteriormente na Seção 2.2.3.

Os autores realizaram testes computacionais bastante abrangentes. O algoritmo proposto baseado em busca em vizinhança de grande porte é capaz de resolver de maneira bastante eficiente instâncias com até 500 itens e 5 tipos de peso, superando os resultados obtidos com o

software de otimização CPLEX 8.1, mesmo quando o CPLEX é processado por um período de tempo 6 a 12 vezes maior.

3.6 OUTRAS METAHEURÍSTICAS

Mais recentemente, outras duas metaheurísticas apareceram com certo destaque na literatura científica, colônia de formigas e “*scatter search*”. Embora ainda não tenham sido investigadas pelo autor, ambos o “*scatter search*” e a colônia de formigas estão entre as prioridades de pesquisa futura, em termos de aplicação a problemas reais de otimização nas áreas de logística e transportes, possivelmente de maneira combinada com outras metaheurísticas.

Cada uma delas é descrita sucintamente a seguir.

3.6.1 Colônia de Formigas

A metaheurística *colônia de formigas* (“*ant colonies*”), introduzida por Dorigo *et al.* (1996), é uma heurística populacional inspirada no comportamento de formigas na busca por comida. Indivíduos simples, organizados em colônias altamente estruturadas, são capazes de encontrar caminhos mínimos entre fontes de comida e a colônia, deixando trilhas de feromônio que induzem outras formigas. Mais especificamente, segundo Taillard *et al.* (2001), tão logo uma formiga encontra uma fonte de comida, ela avalia o potencial da fonte (quantidade e qualidade) e leva algum alimento ao ninho. Durante a viagem de volta, a formiga deposita no chão um trilha de feromônio químico, cuja quantidade depende da qualidade da fonte de alimento. O papel dessa trilha de feromônio é guiar outras formigas para essa fonte. Depois de um tempo, o caminho para uma boa fonte de comida é indicado por uma grande trilha de feromônio, uma vez que a trilha cresce quando mais formigas encontram a fonte. Tendo em vista que fontes mais próximas da colônia são visitadas mais frequentemente que aquelas mais afastadas, trilhas de feromônio levando a fontes mais próximas tendem a crescer mais rapidamente. Como resultado deste processo, as formigas são capazes de otimizar seu trabalho.

Ainda segundo os autores, a transposição desse comportamento para busca de alimento em um algoritmo para resolver problemas de otimização é obtida através de uma analogia entre: (i) a área de busca de formigas reais e o conjunto de soluções viáveis do problema combinatório; (ii) a quantidade de alimento associada com uma fonte e a função objetivo; (iii) a trilha de feromônio e uma memória adaptativa. Adicionalmente, caminhos alternativos são gerados; quanto maior a concentração de feromônio, melhor é a qualidade do caminho. Assim, a metaheurística colônia de formigas busca representar uma colônia cooperativa de indivíduos, em que cada formiga é capaz de produzir soluções viáveis; a cooperação permite obter soluções de boa qualidade.

Um algoritmo de colônia de formigas pode ser esquematicamente descrito da seguinte forma (Taillard *et al.*, 2001):

1. Inicializar a trilha de feromônio
2. Enquanto um critério de parada não tenha sido atingido faça:
 - 2.1 Para cada formiga, construa uma nova solução usando a trilha de feromônio corrente e uma avaliação da solução parcial sendo construída.
 - 2.2 Atualizar a trilha de feromônio.

O componente mais importante de um sistema de formigas é o gerenciamento das trilhas de feromônio. Em uma abordagem clássica, as trilhas de feromônio são usadas em conjunto com a função objetivo para guiar a construção de novas soluções. Uma vez que uma solução tenha sido obtida, as trilhas de feromônio são atualizadas da seguinte maneira: primeiramente todas as trilhas são enfraquecidas a fim de simular a evaporação de feromônio; a seguir, as trilhas que correspondem a componentes que foram usados para construir essa solução são reforçadas, com base na qualidade da mesma.

Mais especificamente, a chamada “comunicação estigmergética” é uma forma indireta de comunicação que permite modificar o estado das localizações visitadas e coletar informações locais. A trilha de feromônio e a comunicação estigmergética local estão associadas à obtenção de informações adicionais relacionadas a cada estado visitado, à percepção do ambiente e histórico da colônia, e à evaporação (estagnação). Maiores detalhes podem ser obtidos em Dorigo *et al.* (1996, 1999).

3.6.2 Scatter Search

Segundo Laguna e Martí (2003), a metaheurística “scatter search” (SS), à semelhança dos algoritmos genéticos, corresponde a um algoritmo evolucionário, na medida em que em ambas um conjunto de soluções (população) é gerado, mantido e evolui. O “scatter search” foi concebido como uma extensão de uma heurística na área de relaxação matemática desenhada para resolver problemas de programação inteira, denominada relaxação de restrições “surrogate” (“surrogate constraint relaxation”); as idéias que inspiraram esse método encontram-se no artigo de Glover (1977). Enquanto na relaxação Lagrangeana as restrições “complicadas” são movidas para a função objetivo, na relaxação de restrições “surrogate” novas restrições são geradas para substituir aquelas consideradas problemáticas.

O esquema geral de funcionamento do “scatter search” é apresentado na Figura 3.34.

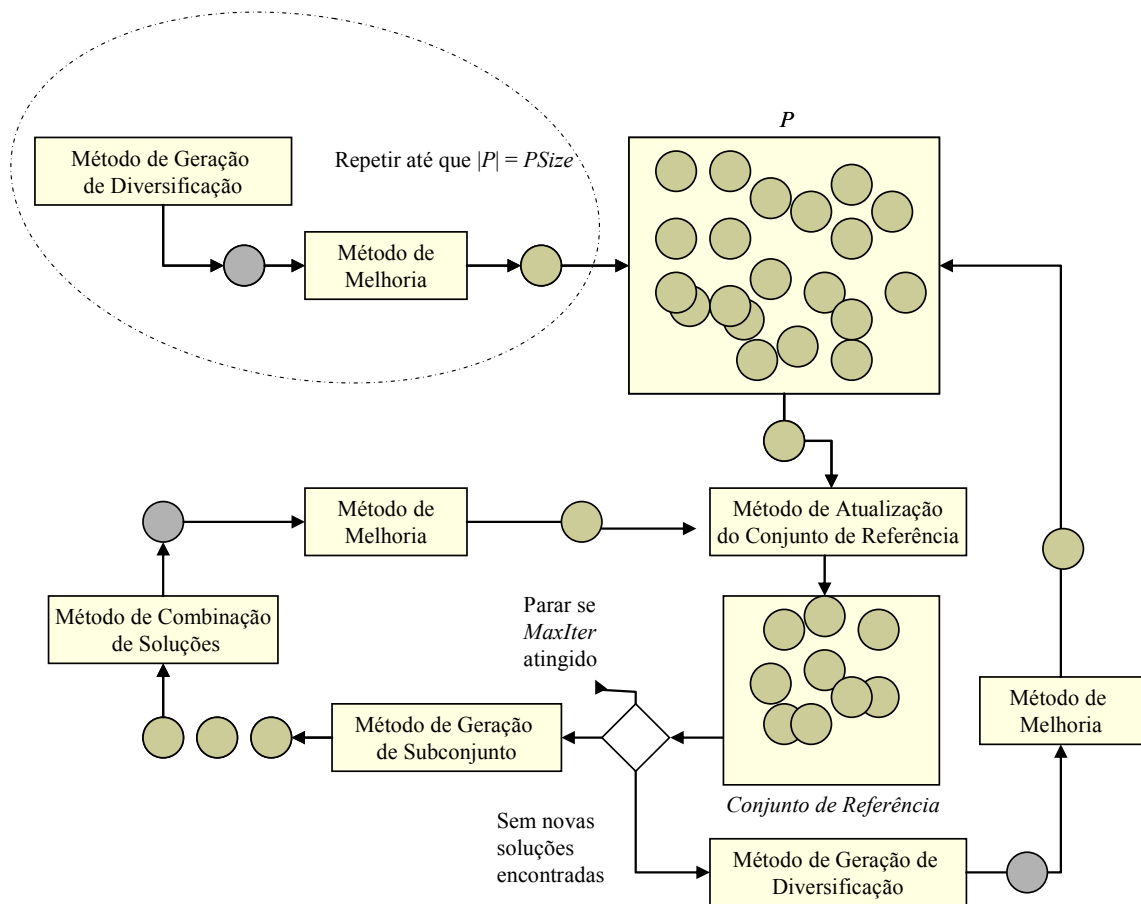


Figura 3.34: Esquema geral de funcionamento do “scatter search”

Fonte: adaptado de Laguna e Martí (2003)

O *scatter search* opera sobre um conjunto de soluções chamado conjunto de referência, combinando estas soluções para criar novas soluções, de modo a melhorar a solução original. Um gerador de permutações, cujo foco é a diversificação e não a qualidade das soluções geradas, é usado no início da busca para construir um conjunto P composto de $PSize$ soluções diversas. Um método de melhoria é então aplicado às soluções em P , a fim de se obter um conjunto de soluções viáveis de razoável qualidade, e também diversidade. O conjunto de referência consiste em uma coleção de soluções que são utilizadas para gerar novas soluções através de um método de combinação de soluções. Cerca de 50% das soluções do conjunto de referência são escolhidas dentre as melhores em P , e as outras 50% escolhidas entre as mais distantes das demais, de modo a proporcionar diversidade. Um procedimento de geração de subconjuntos do conjunto de referência é aplicado, considerando a combinação de todos os pares de soluções do conjunto de referência, a fim de gerar novas soluções, sendo o conjunto de referência atualizado com as melhores soluções obtidas, após a aplicação de um método de melhoria; dessa forma, o tamanho do conjunto de referência permanece sempre constante, porém o valor das soluções vai melhorando ao longo da busca. Quando não forem geradas novas soluções que se qualifiquem para fazerem parte do conjunto de referência, o conjunto de referência é então parcialmente reconstruído, gerando-se tanto soluções de alta qualidade quanto com características de diversidade. O procedimento pára quando um número máximo de iterações é atingido. Uma descrição completa do método pode ser encontrada em Laguna e Martí (2003).

De acordo com Taillard *et al.* (2001), o “*scatter search*” é bastante similar aos algoritmos genéticos, uma vez que o SS também se baseia em uma população de soluções, que evolui através de seleção, combinação linear, transformação do vetor inteiro e seleção para produzir uma nova população de soluções. Em comparação com o algoritmo genético padrão, o SS apresenta as seguintes particularidades: (i) vetores binários são substituídos por vetores inteiros; (ii) mais de dois pais podem ser combinados para produzir um novo vetor; (iii) o cruzamento é substituído pela combinação linear convexa ou não-convexa de dois ou mais vetores; (iv) a mutação é substituída por um procedimento que repara ou projeta o vetor recém-criado no espaço de soluções viáveis.

Tais particularidades podem ser vistas como generalizações do procedimento básico do algoritmo genético. O Quadro 3.1 sintetiza as principais diferenças do “*scatter search*” em relação aos algoritmos genéticos (Martí, Laguna e Campos, 2004).

Quadro 3.1: Comparação Algoritmo Genético *versus* “scatter search”

	AG	SS
População	Grande (~ 100)	Pequena (~10)
Reprodução	Seleção probabilística de pais	Seleção determinística do conjunto de referência
Combinação	Cruzamento e mutação	Combinações estruturadas
Evolução	Sobrevivência dos mais aptos	Atualização estratégica para preservar qualidade e diversidade
Busca Local	Recentemente adicionada como mutação	Parte integral do procedimento

Fonte: adaptado de Martí, Laguna e Campos (2004)

3.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram descritas sucintamente as principais metaheurísticas utilizadas para a resolução de complexos problemas de otimização. Procurou-se sumarizar os principais aspectos e conceitos que caracterizam cada uma delas, com base na experiência real do autor.

A partir do próximo capítulo são apresentados os principais desenvolvimentos metodológicos e os principais resultados obtidos nas principais pesquisas realizadas pelo autor desde o seu doutoramento em 1997. As pesquisas estão agrupadas por grandes temas; embora resultados individuais já tenham sido apresentados em artigos científicos publicados, a sistematização apresentada busca não só demonstrar claramente a evolução do conhecimento acumulado sobre o tema, como também, de maneira didática, a integração entre as pesquisas, além de apresentar uma avaliação crítica dos conhecimentos produzidos pelo autor e seus orientados nas considerações finais de cada trabalho e ao final de cada cap. O próximo capítulo descreve as pesquisas realizadas na área de roteirização e programação de veículos. Já o capítulo 5 trata da modelagem de redes logísticas. Finalmente, o capítulo 6 aborda a otimização ferroviária.

4

**ROTEIRIZAÇÃO E PROGRAMAÇÃO
DE VEÍCULOS****4.1 CARACTERIZAÇÃO DO PROBLEMA**

O problema de *roteamento* (ou *roteirização*) de veículos (do inglês americano “*routing*” ou britânico “*routeing*”) consiste em determinar o(s) roteiro(s) para atendimento de um conjunto de pontos, buscando minimizar o custo total e assegurando que cada ponto seja visitado exatamente uma vez e a demanda em qualquer roteiro não exceda a capacidade do respectivo veículo. Em outras palavras, quais atendimentos (coletas, entregas, ou visitas de serviço) devem ser alocados a cada veículo de uma frota disponível, e qual o roteiro (ou seqüência de paradas) de cada veículo, de forma a minimizar o custo total do serviço, geralmente composto da soma ponderada dos custos fixos e dos custos proporcionais à distância percorrida pelos veículos e ao tempo de viagem. Em poucas palavras, um problema simples de ser enunciado, embora muito difícil de ser resolvido até a otimalidade; uma das histórias de maior sucesso da Pesquisa Operacional, tendo em vista o impressionante número de artigos publicados na literatura científica ao longo de mais de quatro décadas (Assad, 1988).

Quando a definição dos roteiros também envolve aspectos temporais, tais como restrições de horários de atendimento (conhecidas como janelas de tempo, do inglês “*time windows*”) ou de duração máxima de roteiros, o problema é então denominado *roteirização e programação de veículos* (do inglês “*vehicle routing and scheduling*”).

Já o problema de *programação de veículos* pode ser visto como um caso particular do problema de roteirização e programação de veículos no qual os roteiros (ou seqüências de

viagem) são conhecidos a priori; nesse caso, busca-se determinar a alocação (ou programação) dos veículos que minimize a frota necessária para a realização dos serviços, tendo como um possível segundo objetivo a minimização da distância ou do tempo ocioso. Por exemplo, na programação de veículos no transporte coletivo por ônibus busca-se determinar o número mínimo de veículos necessários para realizar um dado conjunto de viagens programadas de uma ou mais linhas (para as quais são definidos a priori, os horários, os trajetos e os pontos extremos das viagens), assim como também definir a seqüência (ou programação) de viagens a ser executada por cada veículo da frota mínima, tal que o custo da operação, composto do custo da frota necessária mais o custo dos deslocamentos improdutivos, seja minimizado (Bodin *et al.*, 1983).

O primeiro problema dessa classe a ser estudado foi o Problema do Caixeiro Viajante (PCV), do inglês “*traveling salesman problem*” (ou TSP), que consiste em encontrar o roteiro ou seqüência de cidades a serem visitadas por um caixeiro viajante, sendo que cada cidade deve ser visitada exatamente uma vez e a distância total percorrida pelo caixeiro viajante seja minimizada; em outras palavras, determinar o itinerário de viagem, em termos da seqüência de cidades a ser visitada, de mínima distância (ou custo). Do ponto de vista matemático, isso equivale encontrar o ciclo hamiltoniano (i.e., ciclo passando por todos os vértices) de mínima distância total.

Segundo Cook (2005), as origens do PCV não estão bem definidas. Na década de 20, o problema foi introduzido pelo matemático e economista Karl Menger aos seus colegas em Viena. Nos anos 30, o problema reapareceu nos círculos matemáticos de Princeton; já nos anos 40, foi estudado por diversos autores, motivados por aplicações em agricultura, ganhando notoriedade como o protótipo de um problema complexo de otimização combinatória, uma vez que era impossível enumerar e examinar os possíveis roteiros um a um, dado o elevado número de possíveis combinações; nenhuma outra idéia de solução podia ser vislumbrada. Em 1962, a empresa americana *Proctor and Gamble (P&G)* do setor de higiene e limpeza, promoveu um desafio público que envolvia resolver um problema de caixeiro viajante para um conjunto de 33 cidades do território norte-americano e iniciando em Chicago, cujo pôster é reproduzido na Figura 4.1.

HELP! WE'RE LOST!

HELP "CAR 54"...AND WIN CASH
54...\$1,000 PRIZES
ONE...\$10,000 GRAND PRIZE

Help Teedy and Muldoon find the shortest round trip route to visit all 33 locations shown on the map. All you do is draw connecting straight lines from location to location to show the shortest round trip route.

HERE'S THE CORRECT START...
 Begin at Chicago, Illinois. From there, lines show correct route as far as Erie, Pennsylvania. Next, do you go to Carlisle, Pennsylvania or West Virginia? Check the easy instructions on back of this entry blank for details.

PROCTER & GAMBLE 1952

OFFICIAL RULES ON REVERSE SIDE

Figura 4.1: Pôster anunciando o desafio proposto pela P&G relacionado a um Problema de Caixeiro Viajante com 33 cidades

Fonte: Cook (2005)

Desde então, só tem aumentado o interesse em resolver esse problema e suas inúmeras variantes nos círculos acadêmicos de matemática, computação, engenharia e administração. Novos problemas e formulações vem sendo propostos, cada vez mais complexos, através da incorporação de novas restrições, incluindo capacidades dos veículos, horários de atendimento, duração máxima das rotas (tempo ou distância), tamanho e composição da frota, tipos de veículos que podem atender determinados clientes, precedência entre clientes, entre outros.

Assim, em linhas gerais, problemas de roteirização de veículos podem ser genericamente vistos como problemas de múltiplos caixeiros viajantes, em que se consideram restrições adicionais de capacidade, de horários de atendimento, de duração de roteiros, de precedência entre atendimentos, dependendo de cada aplicação. Ocorrem em um sem número de situações que envolvem a entrega ou a coleta de mercadorias e pessoas, serviços de atendimento, etc.

Problemas do tipo caixeiro viajante também são encontrados em outras aplicações que não transporte, coleta, distribuição, tais como em linhas de montagem de componentes eletrônicos, onde se busca encontrar, por exemplo, o roteiro de mínima distância para um equipamento cuja tarefa é soldar todos os componentes de uma placa eletrônica. O menor percurso total do equipamento para percorrer todos os pontos da placa está diretamente associado ao desempenho da linha (Souza, 1993).

Na sua forma padrão (clássica), os problemas de roteirização são formulados a partir de um grafo $G = (N, A)$, onde $N = \{1, 2, \dots, n\}$ representa o conjunto de nós (ou vértices) dos pontos a serem atendidos e da(s) base(s) onde se localiza(m) os veículos, enquanto A representa o conjunto dos arcos (ou arestas) com as ligações válidas (i, j) entre nós $i, j \in N$; o custo (ou distância) em cada arco $(i, j) \in A$ é dado por c_{ij} . A sua solução compreende a separação dos nós em subconjuntos alocados a cada um dos veículos e, para cada subconjunto, a determinação do ciclo hamiltoniano (i.e., a ordenação dos vértices de tal forma que cada vértice seja visitado exatamente uma vez) de mínimo custo.

A formulação matemática do problema do caixeiro viajante é simples e compacta. Seja $x_{ij} = 1$ se o trecho (ou arco) (i, j) entre as cidades i e j pertence ao roteiro, 0 caso contrário. O nó $i = 1$ representa a cidade de onde parte e para onde retorna o caixeiro.

$$\text{Minimizar } \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (4.1)$$

sujeito a

$$\sum_{(i,j) \in A} x_{ij} = 1 \quad j \in N \quad (4.2)$$

$$\sum_{(i,j) \in A} x_{ij} = 1 \quad i \in N \quad (4.3)$$

$$X = (x_{ij}) \in \mathcal{S} \quad i, j \in N \quad (4.4)$$

$$x_{ij} \in \{0,1\} \quad i, j \in N \quad (4.5)$$

A função objetivo (4.1) corresponde à minimização do custo (ou distância) total. As restrições (4.2) e (4.3) impõem que cada ponto seja visitado exatamente uma vez, embora não impeçam a ocorrência de “*subtours*”, o que é assegurado pela restrição (4.4). Possíveis escolhas para o conjunto S podem incluir:

$$S = \left\{ (x_{ij}) : \sum_{i \in Q} \sum_{j \notin Q} x_{ij} \geq 1 \right\} \text{ para todo subconjunto não vazio } Q \subset N \quad (4.6)$$

$$S = \left\{ (x_{ij}) : \sum_{i \in R} \sum_{j \in R} x_{ij} \leq |R| - 1 \right\} \text{ para todo subconjunto } R \subset \{2, 3, \dots, n\} \quad (4.7)$$

$$S = \left\{ (x_{ij}) : y_i - y_j + nx_{ij} \leq n - 1 \right\} \text{ para } 2 \leq i \neq j \leq n \text{ para alguns números reais } y_i \quad (4.8)$$

Deve-se notar que S contém aproximadamente 2^n restrições para eliminação de “*subtours*” em (4.6) e (4.7), mas somente $n^2 - 3n + 2$ restrições na formulação (4.8).

A título de ilustração, considere-se a solução para o problema de caixeiro viajante com 6 nós mostrada na Figura 4.2; essa solução é inviável, pois existem dois roteiros desconexos (1-2-3) e (4-5-6), embora satisfaça as restrições (4.2), (4.3) e (4.5), mas não (4.4). Na representação de S dada pela expressão (4.6), a violação ocorre para $Q = \{1, 2, 3\}$, uma vez que não há ligação entre Q e os demais pontos; já na representação de S dada por (4.7) a violação ocorre para $R = \{4, 5, 6\}$, uma vez que contém $|R|$ arcos e não no máximo $|R| - 1 = 2$. Já na representação de S dada por (4.8), se considerarmos os arcos (4,5), (5,6) e (6,4) obtemos $3n \leq 3(n - 1)$, uma contradição.

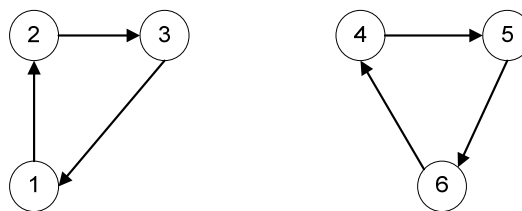


Figura 4.2: Dois “*subtours*” para o PCV com 6 cidades

Fonte: Bodin *et al.* (1983)

Uma formulação ligeiramente diferente para a eliminação de “*subtours*” foi proposta por Gavish e Graves (1978) apud Bodin *et al.* (1983). Sejam y_{ij} variáveis de fluxo para os arcos (i, j) que fazem parte do roteiro, considerando-se que $(n - 1)$ unidades de fluxo são fornecidas

a partir da origem (o nó 1, inicial do roteiro) e cada outro nó $\{2, 3, \dots, n\}$ demanda uma unidade. Assim, em torno de cada nó exceto a *base*, somadas as variáveis de fluxo y_{ij} que saem de cada nó e subtraídas as que chegam em cada nó (isto é $y_{ij} - y_{ji}$ para o nó i), o resultado do balanço deve ser igual a -1 . Deve-se impor ainda que $y_{ij} \leq (n-1)x_{ij}$ para todos os arcos $(i, j) \in A$. Assim, no caso da Figura 4.2 não há continuidade de fluxo para os arcos ligando os nós 4, 5 e 6.

Uma outra categoria distinta de problemas corresponde àqueles em que as demandas a serem atendidas estão localizadas ao longo dos arcos $(i, j) \in A$ do grafo G (ao invés de concentradas dos nós), derivados do chamado Problema do Carteiro Chinês (do inglês “*Chinese Postman Problem*” ou CPP). Uma curiosidade pouco conhecida é que a denominação exótica desse problema decorre do fato de ter sido no periódico *Chinese Mathematics*, em 1952, a primeira vez que esse problema foi abordado. Nessa classe de problemas, busca-se encontrar o(s) roteiro(s) de menor distância (ou custo), de modo a passar pelo menos uma vez em cada arco (i, j) do conjunto A . Matematicamente, isso corresponde a encontrar o ciclo euleriano, isto é o caminho que atravessa cada arco (ou aresta) do grafo exatamente uma vez, de mínimo custo.

A condição para a existência de um ciclo euleriano em grafos direcionados e não direcionados foi formulada pela primeira vez por Euler em 1736, motivado pelo famoso problema das pontes de Königsberg (Bodin *et al.*, 1983). Euler provou que existe um ciclo euleriano em um grafo não direcionado se e somente se o grau (número de arcos incidentes) de cada nó for um número par; ou seja, se um grafo contiver nós com grau ímpar, alguns arcos devem ser percorridos mais de uma vez para formar um ciclo que passe por todos os arcos.

Para a modelagem matemática do Problema do Carteiro Chinês não direcionado define-se x_{ij} igual ao número de vezes que o arco (i, j) é percorrido. A formulação matemática é dada por:

$$\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (4.9)$$

sujeito a

$$\sum_{k=1}^n x_{ki} - \sum_{k=1}^n x_{ik} = 0 \quad i = \{1, 2, \dots, n\} \quad (4.10)$$

$$x_{ij} - x_{ji} \geq 1 \quad \text{para todo } (i, j) \in A \quad (4.11)$$

$$x_{ij} \geq 0 \text{ e inteiro} \quad (4.12)$$

A função objetivo (4.9) busca minimizar a distância total percorrida. A restrição (4.10) assegura que o número de arcos entrando em um nó i seja igual ao número de arcos saindo do nó i (grau par), enquanto que a restrição (4.11) assegura que o roteiro passe por todos os arcos pelo menos uma vez.

Uma analogia interessante do Problema do Carteiro Chinês é o desafio dado às crianças de desenhar uma figura sem levantar o lápis do papel e sem repetir o mesmo traço.

Problemas de roteirização derivados do carteiro chinês podem ser encontrados na coleta de resíduos domiciliares, de entrega domiciliar de gás e de varrição de ruas, podendo considerar restrições de capacidade dos veículos, de duração máxima da jornada e de janelas de tempo associadas aos horários de circulação em certas vias.

4.1.1 Taxonomia dos Problemas de Roteirização

Uma das primeiras taxonomias para os problemas de roteirização de veículos foi proposta por Bodin *et al.* (1983), envolvendo condicionantes e aspectos operacionais tais como:

- (i) tamanho da frota (um ou múltiplos veículos), sua composição (homogênea ou heterogênea), sua localização (uma ou múltiplas bases) e a existência de capacidade dos veículos;
- (ii) natureza da demanda (determinística, estocástica, ou com permissão de atendimento parcial), sua localização (nos nós, nos arcos, ou em ambos);
- (iii) restrições temporais de atendimento dos clientes (janelas de tempo rígidas ou flexíveis com penalidade) e de duração dos roteiros (tempo, distância, ambos);
- (iv) tipo de operação (coletas, entregas, ambos);
- (v) tipo de grafo (direcionado ou não direcionado); distâncias euclidianas baseadas em coordenadas em que se assume a simetria triangular, isto é, $d_{ik} \leq d_{ij} + d_{jk}$;
- (vi) custos considerados (fixos, variáveis, de violação de restrições e de penalidades por não atendimento das demandas) e função objetivo a ser minimizada (distância ou duração total, número de veículos, custo total).

Já Ronen (1988) propôs uma classificação com foco no tipo de serviço oferecido: (i) transporte de passageiros, incluindo sistemas de transporte de idosos e deficientes (conhecidos como “*dial-a-ride*”), de transporte de escolares por ônibus, etc.; (ii) de serviços públicos (coleta de lixo, entrega postal, varrição de ruas, leitura de parquímetros) e de atendimento (equipes de reparos e instalação); de transporte de carga (coleta e distribuição).

Dentro de uma hierarquia de decisões, ou níveis de planejamento logístico, os problemas de roteirização são geralmente caracterizados como pertencentes ao nível operacional, ou seja, fazem parte das tarefas rotineiras de programação da frota, realizadas regularmente com periodicidade de curto prazo, em geral diária ou semanal. Além destes, são também encontrados na literatura problemas de roteirização de natureza mais tática ou estratégica do que operacional, tais como: problemas integrados de localização e roteirização; problemas integrados de estoque e roteirização, nos quais a programação dos atendimentos deve levar em consideração não só aspectos espaciais e os custos dos roteiros, como também questões como o nível de estoque; problemas de faturamento e roteirização, nos quais é preciso definir simultaneamente quem vai ser atendido a cada dia de um período de tempo pré-determinado; entre outros.

4.1.2 Principais Estratégias de Solução

Sob a ótica de otimização, os problemas de roteirização de veículos (incluindo o caso particular do caixeiro viajante) pertencem à categoria conhecida como NP-difícil (do inglês “*NP-hard*”), ou seja, possuem ordem de complexidade exponencial. Em outras palavras, o esforço computacional de solução cresce exponencialmente com o tamanho do problema (dado pelo número de pontos a serem atendidos). Em termos práticos, isto significa que não é possível resolver até a otimalidade problemas reais pertencentes à classe NP-difícil. Conseqüentemente, a quase totalidade dos artigos encontrados na literatura descreve métodos de solução heurísticos.

Conforme visto no capítulo anterior, uma característica importante das heurísticas é que as mesmas apóiam-se em alguma abordagem intuitiva, na qual a estrutura particular do problema possa ser considerada e explorada de forma inteligente, para a obtenção de uma solução adequada (Cunha, 1997). Conseqüentemente, na maioria dos casos, as heurísticas propostas são bastante específicas e particulares, e carecem de robustez, isto é, não conseguem obter

boas soluções para problemas com características, condicionantes ou restrições às vezes um pouco diferentes daquelas para as quais foram desenvolvidas. Em outras palavras, roteirização de veículos é uma área onde uma solução para um determinado tipo de problema (e dados) pode não ser adequada para outro problema similar, conforme apontado por Hall e Partyka (1997). Daí, em muitos casos, a necessidade de buscar soluções customizadas para cada problema.

Dada a diversidade de problemas de roteirização, bem como o grande interesse de pesquisa torna-se praticamente impossível abordar todas as estratégias de solução encontradas na literatura especializada. Assim, serão sucintamente descritas aqui as principais abordagens encontradas na literatura.

Bodin *et al.* (1983) apresentaram o primeiro trabalho abrangente que retratava o estado-da-arte da modelagem e dos algoritmos para problemas de roteirização e programação de veículos e tripulações. Ainda hoje é considerada uma das importantes referências sobre o assunto, pois são considerados inúmeros tipos de problemas e os principais algoritmos de solução. Segundo os autores, as estratégias de solução para problemas de roteirização de veículos podem ser classificadas como um dos seguintes tipos: (i) agrupa e roteiriza (“*cluster first-route second*”); (ii) roteiriza e agrupa (“*route first-cluster second*”); (iii) economias/inserção; (iv) melhorias/trocas; (v) baseados em programação matemática; (vi) procedimentos exatos.

Mais recentemente, Laporte *et al.* (2000) fizeram uma revisão dos principais métodos heurísticos para problemas de roteirização de veículos, incluindo as chamadas heurísticas clássicas tais como o método de economias de Clarke e Wright (1964), nas versões paralela e seqüencial; o método de varredura (“*sweep*”) de Gillet e Muller (1974) e suas extensões denominadas algoritmos petais; o método de atribuição generalizada (“*generalized assignment*”) de Fisher e Jaikumar (1981); e as heurísticas de melhorias através de busca local. Experimentos computacionais realizados pelos autores para as 14 instâncias de problemas propostos por Christofides *et al.* (1979) indicam que boas soluções podem ser obtidas em tempos de processamento reduzidos através dos métodos de economias e varredura. Entretanto, mesmo com melhorias algorítmicas para aprimorar o seu desempenho, os autores destacam que as heurísticas clássicas não conseguem superar as estratégias de solução baseadas em busca tabu, destacando-se: (i) o método *Taburoute* de Gendreau *et al.* (1994) que contém inovações na estrutura de vizinhança explorada e admite soluções

inviáveis devidamente penalizadas; (ii) o algoritmo de Taillard (1993), que considera algumas características do Taburoute, em particular durações tabu randômicas e diversificação, além de envolver uma decomposição espacial do problema; (iii) o método proposto por Xu e Kelly (1996), que incorpora uma estrutura de vizinhança mais sofisticada; (iv) o algoritmo de Rego e Roucairol (1996) que utiliza cadeias de ejeção para se mover de uma solução para a outra; (v) uma estrutura de tabu baseada em memória adaptativa proposta por Rochat e Taillard (1995), em que se mantém um conjunto de soluções de elite que são combinadas para gerar novas soluções de alta qualidade; (vi) o tabu granular proposto por Toth e Vigo (1998). Experimentos computacionais reportados pelos autores indicam que as estratégias baseadas em busca tabu permitem obter a solução ótima ou muito próxima da ótima para problemas de tamanho médio.

Já para o problema de roteirização de veículos com janelas de tempo existem inúmeras heurísticas específicas, podendo-se citar as de Solomon (1987), de Potvin e Rousseau (1995) e de Ioannou *et al.* (2001), sendo esta última superior às outras duas em termos de qualidade das soluções, entretanto os tempos de processamento são maiores. Heurísticas baseadas em busca tabu foram propostas por Rochat e Taillard (1995), Potvin *et al.* (1996), Chiang e Russel (1997), Taillard *et al.* (1997), Cordeau *et al.* (2001). Já estratégias baseadas em algoritmos genéticos foram propostas por Potvin e Bengio (1996) e Tan *et al.* (2001). Já Homberger e Gehring (1999) propuseram uma heurística evolucionária paralela inspirada em algoritmos genéticos com busca local. Bent e Hentenryck (2004) propuseram uma heurística híbrida, baseada em “*simulated annealing*” e busca local em vizinhança de grande porte para o problema de roteirização de veículos com janelas de tempo em que se busca primeiramente minimizar o número de veículos necessários e posteriormente a distância total percorrida, tendo sido obtidos os melhores resultados da literatura. Já Russel e Chiang (2004) desenvolveram uma heurística baseada em *scatter search* que também utiliza busca tabu para a melhoria de soluções obtidas.

Apresentam-se nas seções a seguir diversos problemas que envolvem a roteirização e a programação de veículos:

- Dimensionamento e roteirização de uma frota heterogênea de veículos, utilizando heurísticas de economias;
- Roteirização e programação do transporte de pessoas com deficiências, com restrições de janela de tempo, precedência e múltiplas garagens de onde partem os veículos;

- Roteirização de veículos com carga completa e janelas de tempo, utilizando busca tabu para a sua resolução;
- Roteirização de caminhões-tanque para entrega de combustíveis, utilizando heurística inspirada em GRASP;
- Programação de veículos de coleta de resíduos para a geração de energia através de algoritmos genéticos

Estes problemas selecionados, e que foram objeto das pesquisas desenvolvidas pelo autor, cobrem diversos aspectos, incluindo transporte de carga e passageiros, frota homogênea e heterogênea, um e múltiplos depósitos de onde partem os veículos, janelas de tempo. Englobam ainda diferentes estratégias de solução, tais como heurísticas de inserção e melhoria, busca tabu, algoritmos genéticos, GRASP, etc.

4.2 O PROBLEMA DE DIMENSIONAMENTO E ROTEIRIZAÇÃO DE UMA FROTA HETEROGÊNEA DE VEÍCULOS

O problema de dimensionamento e roteirização de uma frota heterogênea de veículos consiste em definir simultaneamente a composição de uma frota e as rotas, com a finalidade de atender a um conjunto de clientes a mínimo custo total, englobando os custos com as distâncias percorridas pelos veículos e os custos fixos dos veículos utilizados. Busca-se determinar qual a configuração ideal da frota, em termos de número de veículos por tipo e tamanho, bem como o roteiro de cada veículo, de forma que o custo total seja mínimo. Ao contrário de problemas com frota homogênea, onde se admite como razoável minimizar a frota total alocada e, em seguida, a distância total percorrida (Solomon, 1986), em problemas com frota heterogênea uma questão chave a ser analisada é se é melhor utilizar mais veículos de menor capacidade (o que implica maior custo fixo total da frota alocada), percorrendo uma distância total menor (incorrendo menor custo variável) ou vice-versa, ou seja, menos veículos de maior capacidade, percorrendo uma distância total maior.

Um ponto importante a ser destacado é que a tradicional heurística de economias proposta por Clarke e Wright (1964), tanto na sua versão paralela (em que todos os roteiros são formados ao mesmo tempo) quanto na seqüencial (em que apenas um roteiro é montado de cada vez),

não é adequada para problemas que envolvem frota heterogênea. Em ambas as versões, pontos vão sendo agrupados, formando roteiros parciais, seguindo uma ordem decrescente de economias (s_{ij}) decorrentes da sua união, calculadas a partir da seguinte expressão: $s_{ij} = d_{0i} + d_{0j} - d_{ij}$, onde d_{0i} e d_{0j} representam a distância da base (0) aos pontos i e j , respectivamente, e d_{ij} a distância entre eles.

Para ilustrar essa deficiência da heurística de economias para problemas com frota heterogênea, considere-se um problema simples, envolvendo apenas quatro pontos a serem roteirizados, podendo ser utilizados veículos de dois tamanhos, conforme mostrado na Figura 4.3. As economias ordenadas, calculadas segundo a heurística de Clarke e Wright (1964), encontram-se na Tabela 4.1.

Unindo-se os pontos de acordo com a heurística das economias (tanto na versão paralela quanto na seqüencial) obtêm-se, como resultado, dois roteiros que utilizam veículos grandes, conforme mostrado na Tabela 4.2. A solução ótima para este exemplo também se encontra na Tabela 4.2. Observa-se que, embora a heurística de economias produza uma solução de menor distância total percorrida (44km contra 48,4km), o custo total é maior, uma vez o agrupamento resultante requer dois veículos grandes (de 12t), com ociosidade de capacidade, uma vez que o total de carga a ser transportada é de 17t. Já na solução de mínimo custo é possível utilizar um veículo médio e um grande (ao invés de dois grandes), com melhor aproveitamento da capacidade dos mesmos.

A fim de buscar superar esse problema, Golden *et al.* (1984) propuseram heurísticas para o problema de dimensionamento e roteirização de uma frota heterogênea, e que se baseiam em generalizações da heurística de economias, conforme mostrado no Quadro 4.1. Os autores também apresentaram heurísticas baseadas no roteiro gigante, com particionamentos simples e múltiplo.

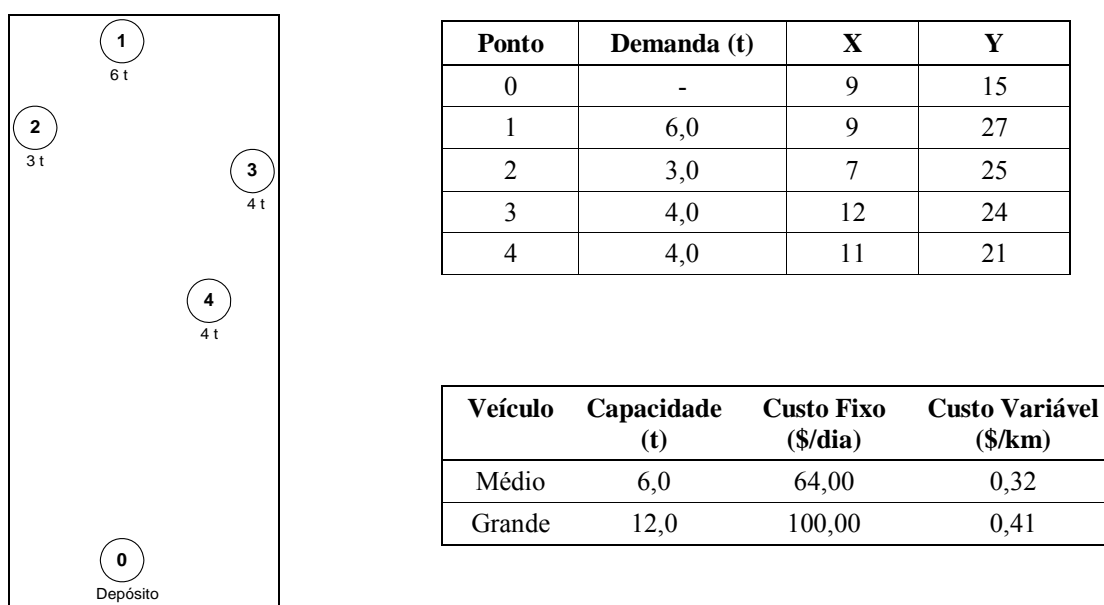


Figura 4.3: Dados do exemplo de problema de roteirização de frota heterogênea

Fonte: Teixeira e Cunha (2002)

Tabela 4.1: Economias ordenadas em ordem decrescente

i	j	d_{ij}	d_{0i}	d_{0j}	economia (s_{ij})
1	2	2,8	12,0	10,2	19,4
1	3	4,2	12,0	9,5	17,2
2	3	5,1	10,2	9,5	14,6
3	4	3,2	9,5	6,3	12,6
1	4	6,3	12,0	6,3	12,0
2	4	5,7	10,2	6,3	10,9

Fonte: Teixeira e Cunha (2002)

Tabela 4.2: Solução da heurística de economias e ótima para o problema da Figura 4.3

Rota	Solução Heurística de Economias				Solução Ótima			
	Veículo	Sequência	Distância	Custo	Veículo	Sequência	Distância	Custo
1	12t	0-2-1-0	25,0	110,25	12t	0-4-3-2-0	24,8	110,17
2	12t	0-4-3-0	19,0	107,79	6t	0-1-0	24,0	71,68
total			44,0	218,04			48,4	181,85

Fonte: Teixeira e Cunha (2002)

Quadro 4.1: Heurísticas de economia propostas por Golden *et al.* (1984) para problemas de roteirização com frota heterogênea

Heurística	Expressão
Clarke e Wright – CW	$s_{ij}^1 = c_{io} + c_{jo} - c_{ij}$
Economias Combinadas CS ¹	$s_{ij}^2 = s_{ij}^1 + F(Z_i) + F(Z_j) - F(Z_i + Z_j)$
Economias de Oportunidade Otimista OOS ²	$s_{ij}^3 = s_{ij}^2 + F(P(Z_i + Z_j) - Z_i - Z_j)$
Economias de Oportunidade Realista ROS ³	se $P(Z_i + Z_j) > \max(P(Z_i), P(Z_j))$ então $s_{ij}^4 = s_{ij}^3 + F(P(Z_i + Z_j) - Z_i - Z_j)$, caso contrário $s_{ij}^4 = s_{ij}^3$
Economias de Oportunidade Realista com parâmetro de forma γ (ROS- γ)	$s_{ij}^5 = s_{ij}^4 + (1 - \gamma)c_{ij}$

Fonte: Golden *et al.* (1984)

As distâncias d_{ij} entre dois pontos quaisquer são substituídas pelos custos variáveis c_{ij} , resultantes da multiplicação das distâncias por um custo variável unitário que, para Golden *et al.* (1984), não variam segundo o tipo ou tamanho do veículo.

Adicionalmente, a função $F(Z)$ calcula o custo fixo do menor veículo capaz de atender a demanda Z . Assim, para o exemplo da Figura 4.3 acima, $F(5t) = \$64,00$, já que o menor veículo capaz de atender a demanda de 5t é o veículo médio; analogamente $F(8t) = \$100,00$, correspondendo ao custo fixo do veículo grande. Já a função $P(z)$ define a capacidade do menor veículo capaz de atender a demanda z ; assim $P(5t) = 6t$, uma vez que o menor veículo que pode atender a demanda de 5t é o veículo médio, com capacidade 6t; da mesma forma $P(8t) = 12t$.

Conforme apontam Golden *et al.* (1984), uma das principais deficiências dos algoritmos de economias é o fato de que, quando um nó é inserido numa rota, ele permanecerá nela até o final da solução. Assim, a heurística de Economias de Oportunidade Realista com parâmetro de forma (ROS- γ) permite variar a economia gerada pela combinação de duas rotas, em função do parâmetro de forma γ .

Desrochers e Verhoog (1991) descrevem uma heurística derivada das heurísticas de economias para frotas heterogêneas de Golden *et al.* (1984). A principal diferença está na

¹ CS: do inglês “Combined Savings”.

²OOS: do inglês “Optimistic Opportunity Savings”.

³ ROS: do inglês “Realistic Opportunity Savings”.

forma de união dos pontos: ao invés de se unir os pontos seqüencialmente, em ordem decrescente de economias, resolve-se um problema de designação em grafo bipartido, em que se consideram todas as possibilidades de combinação de pontos e roteiros. A idéia é encontrar a solução do problema de designação que maximize a economia total, dada pela somatória de todas as economias das rotas unidas.

Nesse contexto, uma pesquisa de mestrado, orientada pelo autor e conduzida por Teixeira (2001), teve por objetivo a implementação computacional e a análise de algumas dessas heurísticas específicas para problemas de roteirização de veículos com frotas heterogêneas. Mais especificamente, consideraram-se a heurística proposta por Desrochers e Verhoog (1991) e uma variante desta.

4.2.1 Heurísticas para Problemas com Frota Heterogênea

A heurística proposta por Desrochers e Verhoog (1991), denominada HOK, corresponde a uma generalização do método de economias de Clarke e Wright (1964) para o caso da frota heterogênea, buscando evitar que as combinações sucessivas de pontos para formar roteiros, segundo uma ordem decrescente de economias, levem à união de pontos e à criação de roteiros que não aproveitem plenamente a capacidade dos veículos, como mostrado no exemplo acima.

Assim, ao invés de se unirem pontos seqüencialmente, segundo uma ordem decrescente de economias, resolve-se um problema de designação em grafo bipartido, com o objetivo de maximizar a economia total de todas as uniões possíveis, considerando-se todas as possibilidades de combinação de pontos extremos de roteiros parciais. De acordo com Desrochers e Verhoog (1991) devem ser unidas apenas as duas rotas parciais correspondentes ao arco de maior economia, dentre aqueles que formam a solução do problema de designação. Conseqüentemente, sucessivos problemas de designação devem ser resolvidos até que não haja mais possibilidade de união de rotas, por restrições de viabilidade ou por não haver mais economias.

Teixeira (2001) modelou o problema de designação como um problema de fluxo em redes, e solucionado através do algoritmo Out-of-Kilter (para detalhes, ver Ford e Fulkerson, 1962; Dantzig, 1993 ou Gualda, 1975). A idéia de usar o algoritmo Out-of-Kilter surgiu do interesse

em avaliar o desempenho desse algoritmo, que tem flexibilidade para resolver diferentes tipos de problemas de fluxo em rede (problema do transporte, do transbordo, de designação, de caminho mínimo, de fluxo máximo, etc.), tendo sido utilizado por Gualda (1975).

Tendo em vista que apenas os dois pontos extremos das duas rotas parciais correspondentes ao arco de maior economia são unidos em cada problema de designação, segundo a proposta de Desrochers e Verhoog (1991), o número de problemas a serem resolvidos pode potencialmente ser bastante elevado, levando a tempos de processamento muito longos. Assim, foi também proposta uma heurística modificada, denominada Heurística Híbrida (HH), que potencialmente permite reduzir o esforço computacional. Nessa heurística, agregam-se todas as rotas que fazem parte da solução de cada problema de designação que é resolvido, ao invés de apenas o par de rotas de maior economia. Espera-se que um número menor de instâncias do Problema de Circulação com Custo Mínimo tenham que ser resolvidas, levando a uma diminuição dos tempos de processamento.

Para efeito de comparação, foi também implementada a heurística de economias ROS- γ , proposta por Golden *et al.* (1984).

Diferentemente de Golden *et al.* (1984) e de Desrochers e Verhoog (1991), Teixeira (2001) considerou o custo com a distância variável por tipo de veículo. Adicionalmente, Desrochers e Verhoog (1991) não reportam nenhum experimento computacional.

4.2.2 Experimento Computacional

As heurísticas foram implementadas em linguagem Pascal Orientado a Objetos, utilizando a ferramenta Delphi da Borland. As heurísticas foram implementadas utilizando recursos de programação orientada a objetos, como herança e polimorfismo, o que proporcionou maior clareza ao código e menor tempo de implementação. Maiores detalhes podem ser encontrados em Teixeira (2001).

Os testes computacionais foram realizados em um microcomputador com processador Pentium III – 550 MHz e 128 MB de memória RAM. Consideraram-se problemas com 10 a 200 pontos de atendimento. As coordenadas dos pontos foram geradas aleatoriamente, considerando uma área quadrada de dimensão 50 km. Da mesma forma, a demanda em cada ponto foi gerada aleatoriamente, segundo uma distribuição uniforme entre os valores 1 e 27

toneladas (que corresponde à capacidade do maior veículo disponível). Em relação à frota disponível, foram considerados quatro tipos diferentes de veículos, com capacidades de 4, 7, 9 e 27t, sem limitação de frota e com diferentes custos fixos e variáveis.

Os resultados obtidos estão na Tabela 4.3. De certa forma surpreendente, a heurística híbrida (HH) apresentou, em grande parte das instâncias, resultados melhores que a heurística básica (HOK). Nas poucas instâncias em que a heurística HOK produziu melhores resultados, os resultados da heurística híbrida sempre foram muito próximos (no pior caso, apenas 0,12% superior). Por outro lado, observa-se que os tempos de processamento das heurísticas HOK e HH, que utilizam o resultado do problema de designação para definir o agrupamento das rotas, são significativamente mais altos que para a heurística de economias ROS- γ .

Tabela 4.3: Desvios de custo obtidos para as três heurísticas em relação ao melhor resultado obtido

Nós	ROS- γ		HOK		HH	
	desvio (%)	Tempo de CPU (seg)	desvio (%)	Tempo de CPU (seg)	desvio (%)	Tempo de CPU (seg)
10	<u>0,00</u>	0	<u>0,00</u>	0	<u>0,00</u>	0
20	8,62	0	<u>0,00</u>	0	<u>0,00</u>	0
30	11,98	0	<u>0,00</u>	2	0,05	2
40	9,43	0	<u>0,00</u>	8	0,17	9
50	11,81	0	<u>0,00</u>	25	<u>0,00</u>	25
60	7,63	0	2,90	66	<u>0,00</u>	67
70	9,87	0	1,21	144	<u>0,00</u>	145
80	12,79	0	<u>0,00</u>	299	<u>0,00</u>	305
90	8,95	0	<u>0,00</u>	539	0,12	545
100	9,63	0	3,28	877	<u>0,00</u>	895
110	9,43	0	1,65	1.546	<u>0,00</u>	1.567
120	9,05	0	<u>0,00</u>	2.309	0,02	2.319
130	9,69	1	<u>0,00</u>	3.466	0,01	3.566
140	10,67	1	1,26	5.210	<u>0,00</u>	5.287
150	11,39	1	2,29	7.441	<u>0,00</u>	7.476
160	10,87	2	3,26	10.250	<u>0,00</u>	10.564
170	11,01	2	4,92	15.409	<u>0,00</u>	15.806
180	10,75	2	4,05	20.551	<u>0,00</u>	20.436
190	10,65	3	4,25	26.441	<u>0,00</u>	26.493
200	9,99	3	1,43	53.412	0,00	34.235
Média	9,71		1,53		0,02	

Fonte: Teixeira (2001)

Por outro lado, embora a heurística híbrida tenha sido proposta com a motivação de reduzir o elevado tempo de processamento da heurística básica, verificou-se que, embora o número de iterações tenha reduzido significativamente, essa redução não se refletiu na diminuição do tempo de processamento.

Uma análise mais detalhada da composição dos tempos de processamento mostrou que a quase totalidade do tempo de CPU corresponde ao tempo para a obtenção da primeira solução do problema de designação pelo algoritmo Out-of-Kilter. Deve-se lembrar que o algoritmo Out-of-Kilter é uma generalização do método Primal-Dual, que permite que se inicie a resolução do problema com uma solução inviável, tanto na fase primal como na dual (Dantzig, 1993). A estratégia geral do algoritmo Out-of-Kilter consiste em iniciar o Problema de Circulação com Custo Mínimo com uma solução de fluxo nulo na rede, inviável, que vai sendo progressivamente melhorada. É justamente o primeiro problema de designação que requer muitas iterações até convergir para a solução ótima. Diferentemente do primeiro problema de designação, nos demais se aproveita a solução anterior como solução inicial, o que permite reduzir significativamente os tempos de processamento. Assim, as heurísticas propostas podem se tornar viáveis de serem aplicadas a problemas reais, caso o desempenho do algoritmo Out-of-Kilter possa ser aprimorado no tocante à solução inicial para o primeiro problema de designação a ser resolvido. Pode-se considerar ainda diminuir o esforço computacional através da não consideração, na resolução do problema de designação, de todas as possíveis combinações de rotas com economias, mas somente aquelas mais promissoras.

4.2.3 Considerações Finais

Tendo em vista que o tempo de resolução das inúmeras instâncias dos problemas de designação corresponde a uma parcela significativa do tempo total de processamento, recomenda-se a análise comparativa de outros algoritmos para a resolução do referido problema. Em particular, pretende-se no futuro investigar a utilização dos algoritmos específicos para problemas de fluxo em rede disponíveis no pacote CPLEX da ILOG, considerados dentre os mais eficientes e rápidos do mercado.

Uma continuidade desse trabalho envolverá também a avaliação do desempenho das heurísticas propostas para diversos conjuntos de problemas-teste disponíveis na literatura, avaliação essa não realizada por Desrochers e Verhoog (1991).

Adicionalmente, seria interessante avaliar o desempenho comparativo das três heurísticas para diferentes configurações espaciais em termos dos pontos de atendimento e da posição da base dos veículos (por exemplo, a base numa localização central ou num extremo da região), e em relação a diferentes distribuições (agrupamentos/dispersões) dos pontos na região.

Outro ponto a ser explorado consiste na avaliação da influência na qualidade das soluções do parâmetro de forma γ , relativo à heurística ROS- γ , utilizado no cálculo das economias.

4.3 ROTEIRIZAÇÃO E PROGRAMAÇÃO DO TRANSPORTE DE PESSOAS COM DEFICIÊNCIAS

Esta seção trata do problema de roteirização e programação de veículos para o transporte de pessoas com deficiências de seus locais de origem a seus destinos, e que foi objeto da pesquisa de mestrado de Znamensky (2000), sob orientação deste autor.

Embora comum há vários anos nos países mais desenvolvidos, em particular nas grandes cidades norte-americanas, esse importante serviço de utilidade pública que envolve o transporte de pessoas com deficiências está em operação no município de São Paulo desde 1996, por iniciativa da São Paulo Transportes S.A. (SPTrans), sucessora da tradicional CMTC (Companhia Municipal de Transportes Coletivos) desde 1995, e responsável pelo gerenciamento técnico e operacional do sistema de transporte urbano no município. O Serviço de Atendimento Especial (Atende), criado através do decreto 36.071, de 9 de maio de 1996, é uma modalidade de transporte porta-a-porta, gratuito, com regulamentação própria, oferecido pela Prefeitura do Município de São Paulo. Destina-se às pessoas portadoras de deficiência física com alto grau de severidade e dependência, impossibilitadas de utilizar outros meios de transporte público. Os veículos que operam neste serviço são devidamente adaptados para garantir conforto e segurança a seus usuários, e equipados com plataforma de elevação para embarque e desembarque. O planejamento, a organização, e a fiscalização do serviço são de responsabilidade da SPTrans, e sua operação compete às empresas de transporte coletivo do

município de São Paulo. O serviço destina-se prioritariamente à reabilitação, tratamento de saúde, educação e, caso haja oferta de veículos, a trabalho, esporte, lazer, cultura e outras atividades da vida diária. Para serem transportados, os usuários são previamente cadastrados e devem agendar a sua requisição de viagens sempre com antecedência de quinze dias.

Segundo informações da SPTrans, o Atende contava em 2005 com uma frota patrimonial de 252 carros, realizando mais de 100 mil viagens por mês para atender cerca de 6 mil usuários cadastrados, sendo a maior operação dessa natureza no Brasil.

Tendo em vista a relevância desse tipo de sistema para a sociedade, bem como a importância de uma programação eficiente dos veículos, de forma a assegurar a máxima capacidade de atendimento com nível de serviço adequado aos usuários, esse problema foi estudado por Znamensky (2000) e proposta uma heurística capaz de gerar, de forma automatizada, roteiros econômicos para esse tipo de problema, apoiando a necessidade de programação diária num cenário de demanda crescente por esses serviços.

4.3.1 Caracterização do Problema

Os problemas de roteirização e programação de veículos que envolvem o transporte de passageiros de seus locais de origem a seus destinos são conhecidos na literatura como problemas do tipo “*dial-a-ride*”. Tais problemas geralmente envolvem o transporte porta-a-porta de pessoas idosas e com deficiências. O termo “*dial-a-ride*” decorre da possibilidade dessas pessoas solicitarem esses serviços por telefone.

Ao contrário dos táxis, onde cada passageiro é atendido individualmente por uma viagem direta da sua origem para o seu destino, nesse tipo de serviço várias pessoas, com diferentes origens e destinos, podem ser transportadas ao mesmo tempo pelo veículo.

As restrições de atendimento podem ser de vários tipos. Note-se que o fato das tarefas serem divididas em coletas (embarques) e entregas (desembarques) define uma restrição de precedência, já que é impossível proceder à entrega de um passageiro que não foi embarcado. Tais restrições não estão presentes em problemas tradicionais de roteirização de veículos, dificultando ou até impedindo a aplicação de heurísticas consagradas na literatura, como por exemplo, o método de economias (Clarke e Wright, 1967),.

Há ainda limitações de capacidade nos veículos; mais especificamente em relação ao número máximo de passageiros que podem ser transportados. Finalmente, há também restrições temporais, que podem ser expressas na forma de janelas de tempo (isto é, janelas ou intervalos horários para atendimento dos passageiros) e também de tempo máximo de permanência de cada passageiro em trânsito dentro do veículo.

A frota é composta por veículos de diferentes tamanhos. Cada veículo possui uma capacidade, dada pelo número de assentos convencionais e pelo número de cadeiras de rodas. Analogamente, a cada solicitação de atendimento está associada uma ocupação em termos de quantidade de assentos convencionais (eventuais acompanhantes) e em quantidade de cadeiras de rodas, além dos endereços de origem e destino.

Os problemas do tipo “*dial-a-ride*” podem ser classificados em estático e dinâmico. No caso estático, como no problema em questão, os atendimentos a serem realizados, bem como a sua localização, são conhecidos *a priori*; dessa forma, não é possível a inclusão de novos atendimentos nos roteiros em andamento. Já no caso dinâmico, tais alterações são permitidas, sendo os roteiros modificados à medida que novas solicitações são recebidas e aceitas. Existem ainda diferenças quanto à frota, homogênea ou heterogênea, uma ou múltiplas garagens e quanto à função objetivo que se deseja minimizar.

Diversos autores propuseram algoritmos exatos para esse problema. Psaraftis (1980) desenvolveu um algoritmo exato de programação dinâmica para a resolução do problema de “*dial-a-ride*” para um único veículo, sem restrições de janelas de tempo, considerando ambos os casos estático e dinâmico do problema. Buscou-se minimizar uma ponderação entre o tempo total de operação do veículo e a “insatisfação” total dos usuários, que é medida por uma função linear dos tempos de espera e viagem de cada usuário. A extensão do problema para a inclusão de restrições de janelas de tempo também foi analisada por Psaraftis (1983). Uma melhoria na solução exata por meio de programação dinâmica foi proposta por Desrosiers *et al.* (1986) para o problema com um veículo e restrições de janelas de tempo, permitindo a resolução de problemas até quatro vezes maiores que os analisados por Psaraftis (1983). Dumas *et al.* (1991) desenvolveram um algoritmo exato, baseado no método de geração de colunas, para o problema de “*dial-a-ride*” com frota homogênea e restrições de janelas de tempo; o algoritmo foi aplicado com sucesso a problemas de 55 solicitações.

Jaw *et al.* (1986) descrevem uma heurística de inserção paralela para o problema de “*dial-a-ride*” com múltiplos veículos e restrições de janelas de tempo, que foi aplicada com sucesso a um problema com aproximadamente 2600 clientes cadastrados na sua base de dados (nem todos os clientes solicitavam atendimento no mesmo dia) e 20 veículos. Uma versão aprimorada dessa heurística foi proposta por Madsen *et al.* (1995), buscando priorizar a inserção de solicitações com restrições mais rígidas, quando os veículos ainda estão mais vazios, aumentando as possibilidades de inserção viáveis. Já Ioachim *et al.* (1995) desenvolveram um método que consiste na criação de “mini-clusters” (segmentos viáveis do itinerário em que o veículo não fica vazio), e na resolução do problema resultante dos *m*-caixeiros viajantes por meio do método de geração de colunas, tendo sido aplicado a um problema real com cerca de 2500 solicitações. Toth e Vigo (1996) descrevem uma heurística de inserção paralela para o problema com múltiplos veículos e restrições de janelas de tempo, que inclui um procedimento de busca local envolvendo trocas intra-rotas e trocas inter-rotas para a melhoria da solução, tendo sido a mesma aplicada com sucesso em um problema real com cerca de 1500 solicitações. Num trabalho subsequente, Toth e Vigo (1997) propuseram um aprimoramento do método de melhoria através de busca tabu.

4.3.2 Formulação Matemática

O conjunto de solicitações de atendimento é dado por k . Para cada solicitação de atendimento de passageiros $k= 1, 2, \dots, n$, associam-se o ponto de embarque (coleta) k e o ponto de desembarque (entrega) $n+k$. Definem-se $P^+ = \{1, 2, \dots, n\}$ o conjunto dos n pontos de embarque (coleta) e $P^- = \{n+1, n+2, \dots, 2n\}$ o conjunto dos n pontos de desembarque (entrega); $P = P^+ \cup P^-$ corresponde ao universo de pontos a serem visitados. Para cada ponto $i \in P$, é conhecida a janela de tempo $[a_i, b_i]$ dentro da qual o atendimento (seja embarque ou desembarque) deve ser iniciado ($a_i \leq b_i$), bem como o tempo $s_i \geq 0$ necessário para o atendimento (embarque ou desembarque). Não está excluída a possibilidade de existirem janelas de tempo com uma das extremidades em aberto, (como por exemplo $]-\infty, b_i]$ ou $[a_i, +\infty[$), ou não atuantes ($]-\infty, +\infty[$).

Seja $W = \{1, 2, \dots, |W|\}$ o conjunto de categorias de passageiros, tais como passageiros sentados, passageiros em cadeira de rodas, etc. D_i^w indica o número de passageiros da categoria $w \in W$ embarcados ou desembarcados no ponto $i \in S$, de tal forma que:

$$D_i^w \in Z \quad (4.13)$$

$$D_i^w \geq 0, \quad w \in W, i \in P^+ \quad (4.14)$$

$$D_i^w \leq 0, \quad w \in W, i \in P^- \quad (4.15)$$

$$D_i^w + D_j^w = 0, \quad w \in W, i \in P^+, j = i + n \quad (4.16)$$

$$D_i^w = 0, \quad w \in W, i \in G \quad (4.17)$$

$$\sum_{w \in W} D_i^w \neq 0, \quad i \in P \quad (4.18)$$

A expressão (4.13) impõe que o número de passageiros de uma determinada categoria, embarcados ou desembarcados em um dado ponto seja um número inteiro. As expressões (4.14) e (4.15) impõem que o número de passageiros da categoria w embarcados e desembarcados no ponto i , respectivamente, não seja negativo. A expressão (4.16) assegura que para uma dada solicitação de serviço, o número de passageiros da categoria w , embarcados no ponto de coleta correspondente a essa solicitação, é igual ao número de passageiros da categoria w desembarcados no ponto de entrega, enquanto que a expressão (4.17) impõe que não haja embarque ou desembarque de passageiros nos pontos correspondentes às garagens. Por fim, a expressão (4.18) exige que ao menos um passageiro seja embarcado ou desembarcado em um dado ponto a ser visitado.

Seja $G = \{1, 2, \dots, |G|\}$ o conjunto das garagens e $V = \{1, 2, \dots, |V|\}$ o conjunto dos veículos disponíveis. Define-se ainda $S = P \cup G$. A alocação dos veículos às garagens é conhecida a priori, dada por:

$$q_{vg} = \begin{cases} 1, & \text{se veículo } v \in V \text{ está alocado à garagem } g \in G \\ 0, & \text{caso contrário} \end{cases}$$

Para cada veículo $v \in V$ são conhecidos (i) o seu custo unitário variável com a distância C_d^v ; (ii) custo fixo diário total C_f^v ; e (iii) custo horário C_h^v . A capacidade do veículo v para cada categoria w de passageiros é dada K^{vw} .

O deslocamento de um veículo v de um ponto i até um ponto j requer um tempo de viagem t_{ij} , e a distância percorrida é dada por d_{ij} , $i \neq j$, $i \in S$, $j \in S$. Admite-se que os tempos de viagem t_{ij} independam do tipo de veículo considerado.

A formulação matemática do problema compreende as seguintes variáveis de decisão:

$$x_{ij}^v = \begin{cases} 1, & \text{se o ponto } j \in S \text{ é visitado após o ponto } i \in S \text{ pelo veículo } v \in V \\ 0, & \text{caso contrário} \end{cases}$$

$$T_i = \text{horário de início de atendimento do ponto } i \in P$$

$$Y_i^{vw} = \text{número total de passageiros do tipo } w \in W, \text{ no veículo } v \in V, \text{ após atendimento do ponto } i \in S$$

O problema de roteirização e programação de atendimento de pessoas com deficiências (“*Dial-a-Ride*”) com janelas de tempo e utilizando uma frota heterogênea de veículos localizada em múltiplas bases pode ser formulado como:

$$\text{Minimizar } \alpha \left(\sum_{v \in V} \sum_{i \in S} \sum_{j \in S} C_d^v d_{ij} x_{ij}^v \right) + \beta \left(\sum_{v \in V} \sum_{i \in G} \sum_{j \in P^+} C_f^v x_{ij}^v \right) \quad (4.19)$$

sujeito a:

$$\sum_{v \in V} \sum_{i \in P} x_{ij}^v = 1, \quad j \in P \quad (4.20)$$

$$\sum_{v \in V} \sum_{j \in P} x_{ij}^v = 1, \quad i \in P \quad (4.21)$$

$$\sum_{i \in S} x_{ip}^v - \sum_{j \in S} x_{pj}^v = 0, \quad v \in V, \quad p \in S \quad (4.22)$$

$$\sum_{j \in S} x_{gj}^v = q_{vg}, \quad v \in V, \quad g \in G \quad (4.23)$$

$$\sum_{j \in S} x_{jg}^v = q_{vg}, \quad v \in V, \quad g \in G \quad (4.24)$$

$$\sum_{j \in P} x_{ij}^v - \sum_{j \in P} x_{j,i+n}^v = 0, \quad v \in V, \quad i \in P^+ \quad (4.25)$$

$$x_{ij}^v = 1 \Rightarrow Y_j^{vw} = Y_i^{vw} + D_j^w, \quad v \in V, \quad i \in S, \quad j \in S, \quad i \neq j, \quad w \in W \quad (4.26)$$

$$0 \leq Y_i^{vw} \leq K^{vw}, \quad v \in V, \quad i \in P, \quad w \in W \quad (4.27)$$

$$Y_i^{vw} = 0, \quad v \in V, \quad i \in G, \quad w \in W \quad (4.28)$$

$$a_i \leq T_i \leq b_i, \quad i \in P \quad (4.29)$$

$$T_i + s_i + t_{ij} - T_j \leq (1 - x_{ij}^v)M, \quad v \in V, \quad i \in P, \quad j \in P, \quad i \neq j \quad (4.30)$$

$$T_i + s_i + t_{i,n+i} \leq T_{n+i}, \quad i \in P^+ \quad (4.31)$$

$$x_{ij}^v \in \{0, 1\}, \quad v \in V, \quad i \in S, \quad j \in S \quad (4.32)$$

A função objetivo (4.19) requer que o custo total de transporte, composto das parcelas correspondentes à distância total percorrida e aos veículos utilizados, ponderadas pelos fatores α e β , respectivamente, seja minimizado. As restrições (4.20)-(4.22) asseguram que cada requisição seja atendida uma única vez e por um único veículo; cabe ressaltar que uma delas é redundante, já que duas delas implicam a terceira. Já as restrições (4.23) e (4.24) impõem que cada veículo deve partir e retornar à garagem em que está alocado; o caso do veículo não ser utilizado é representado por uma viagem fictícia da garagem onde o veículo está alocado para ela mesma, evitando que o custo fixo desse veículo não utilizado seja computado na função objetivo. A restrição (4.25) garante que o embarque e o desembarque de um cliente sejam ambas realizadas pelo mesmo veículo, enquanto que a restrição (4.26) está relacionada à continuidade do número de passageiros de cada categoria w em cada veículo v , enquanto que a restrição (4.27) garante que, para cada veículo, a capacidade em passageiros da categoria w não será excedida. A restrição (4.28) impõe que todos os veículos deixem e retornem às respectivas garagens vazios. A restrição (4.29) garante que todos os atendimentos sejam realizados dentro das respectivas janelas de tempo, enquanto que a restrição (4.30) está relacionada à continuidade temporal dos roteiros, impedindo ainda a formação de sub-roteiros parciais. Já a restrição (4.31) impõe que o embarque de um cliente deve preceder ao desembarque do mesmo.

Cabe notar que a função objetivo proposta não inclui nenhuma parcela relativa à desutilidade ou inconveniência percebida pelo usuário em decorrência de um tempo de viagem excessivo em comparação com tempo de percurso direto. Pelo lado do usuário há um desejo de minimização da inconveniência, quantificada por tempo de viagem e tempo de espera, ao passo que, pelo lado do operador do serviço há um desejo de minimização dos custos e maior aproveitamento dos veículos. A alternativa adotada na presente formulação considera a questão de minimização da inconveniência do usuário na forma de restrições nas janelas de tempo; em outras palavras, as janelas de embarque e/ou desembarque são definidas de tal maneira a evitar que um usuário tenha um tempo de viagem excessivamente longo em relação ao seu percurso direto.

4.3.3 Estratégia de Solução

A estratégia de solução proposta consiste em uma adaptação da heurística de inserção paralela proposta por Madsen *et al.* (1995), seguida de uma heurística de melhoria, similar à de Toth e Vigo (1996).

Uma característica importante da heurística de inserção é a proposição de um critério de “dificuldade” para ordenar as solicitações de atendimento. Tal critério determinado com base em três parcelas de “dificuldade”, correspondentes à (i) largura da janela de tempo de menor duração, ao (ii) tempo máximo permitido de permanência no veículo, e à (iii) demanda de cada solicitação (número de assentos e cadeiras de rodas). A ponderação das parcelas de “dificuldade” é bastante flexível, e pode ser definida pelo operador do sistema, o qual pode priorizar a inserção de um determinado tipo de solicitação (solicitações com cadeiras de rodas, por exemplo).

Após a ordenação das solicitações, é calculado o custo de inserção da solicitação mais “difícil” em todas as rotas (considerando todos os veículos, já utilizados ou não), quando viável. Essa solicitação é então inserida na rota (veículo) que acarrete o menor acréscimo de custo, se tal rota existir. Caso contrário, a solicitação é inserida em uma lista de solicitações não atendidas. Repete-se o mesmo procedimento para cada solicitação, seguindo a ordem não decrescente de dificuldade, até que todas as solicitações tenham sido processadas. É interessante ressaltar que a heurística de inserção é de tal forma flexível que permite a utilização de frota com características operacionais e custos distintos para cada veículo.

Uma vez obtida uma solução inicial viável, aplica-se uma heurística de melhoria baseada em busca local, proposta por Toth e Vigo (1996). São considerados dois tipos de trocas inter-rotas:

- a) Re-inserção de Solicitações: remover a solicitação i de sua rota original, e re-inserir as paradas correspondentes nas melhores posições viáveis dentre as rotas disponíveis. Pode ocorrer o caso em que a melhor inserção seja na rota original, correspondendo ao caso de uma troca intra-rota.
- b) Troca de Solicitações: remover as solicitações i e j de suas rotas originais (necessariamente distintas), re-inserir as paradas correspondentes à solicitação i nas melhores posições viáveis na rota original da solicitação j , e analogamente re-inserir as

paradas correspondentes à solicitação j nas melhores posições viáveis na rota original da solicitação i .

Entende-se por posição viável a posição de inserção de uma parada (embarque ou desembarque) em uma rota, que não acarrete a violação das restrições de janela de tempo e de capacidade do veículo em nenhuma parada da referida rota. A melhor posição viável corresponde, pois, à posição viável que cause o menor acréscimo de custo, dentre todas as posições viáveis existentes.

4.3.4 Aplicação do Modelo

O modelo proposto foi aplicado a um caso real extraído da operação do Atende, serviço de transporte de deficientes gerido pela São Paulo Transporte S.A. Os dados utilizados correspondem a um dia típico do mês de novembro de 1998, perfazendo 349 solicitações. Existem 84 veículos disponíveis, localizados em 47 garagens no município de São Paulo.

Sem dúvida, a etapa mais trabalhosa foi a localização geográfica dos pontos de embarque e desembarque (coleta e entrega) e dos pontos referentes às garagens, cujas localizações são mostradas na Figura 4.4.

Duas importantes questões que aparecem nos problemas de roteirização são o cálculo das distâncias e qual velocidade adotar para o cálculo dos tempos de viagem entre pontos de atendimento. As distâncias utilizadas para o estudo de caso foram calculadas a partir das respectivas distâncias euclidianas corrigidas, entre os pontos de atendimento, segundo procedimento descrito por Novaes (1989), e apresentado em Znamensky (2000). Para a calibração consideraram-se cerca de 800 pontos no município de São Paulo, resultando 651.249 pares de pontos. Duas matrizes de distância foram criadas para esse conjunto de pontos: uma correspondente à distância medida sobre a rede viária, e a outra correspondente à distância euclidiana. A função obtida foi $D = 0,582 + 1,113 \cdot \overline{DE}$, onde D =distância real pela malha viária e \overline{DE} =distância euclidiana, ambas em km, com $R^2 = 0,99257$.

Uma análise semelhante feita por Novaes (1989) com apenas 57 pares de pontos havia resultado $D = 0,81 + 1,366 \cdot \overline{DE}$ com $R^2 = 0,95$.

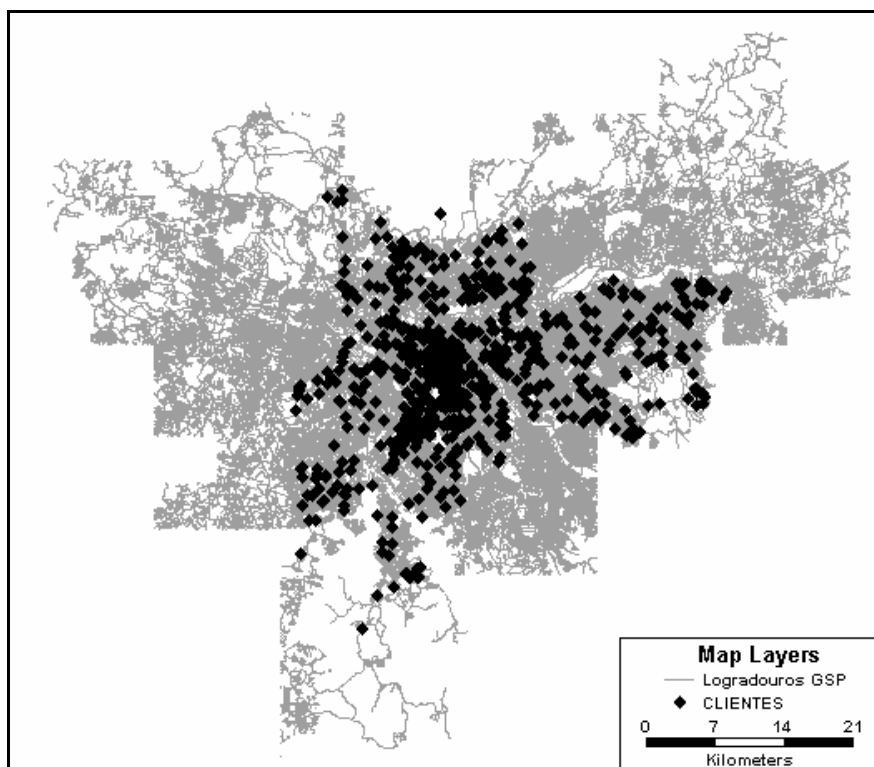


Figura 4.4: Localização dos pontos de atendimento

Fonte: Znamensky (2000)

Sabe-se que a velocidade média dos veículos varia conforme o trajeto e o horário da viagem, o dia da semana, etc. Por outro lado, o tratamento das velocidades médias como variáveis aleatórias acrescenta um elevado grau de complexidade ao problema, não sendo usualmente considerado nos trabalhos de roteirização encontrados na literatura. Foram adotadas, como é prática em vários *softwares* de roteirização, velocidades médias constantes para três faixas de distâncias. A Tabela 4.4 apresenta as faixas de distância e as respectivas velocidades médias adotadas.

Tabela 4.4: Velocidades médias e faixas de distâncias

Faixa de distância	Limites	Velocidade
Curtas distâncias	distância < 10 km	15 km/h
Médias distâncias	10 km ≤ distância < 20 km	30 km/h
Longas distâncias	20 km ≤ distância	45 km/h

Fonte: Znamensky (2000)

Buscou-se minimizar a distância total percorrida pelos veículos, dado que a frota é fixa. Alguns indicadores relativos à programação obtida pela aplicação da heurística de inserção paralela são apresentados na Tabela 4.5.

Tabela 4.5: Indicadores relativos à programação obtida pela heurística de inserção paralela

Indicador	Valor
Rotas programadas	59
% Utilização da frota	70,2 %
Percurso total	6.462,7 km
Percurso médio por rota	109,5 km
% Percurso vazio	35,8 %
Assentos * km (vazios)	12.579,4 Assentos * km
Assentos * km (total)	18.464,8 Assentos * km
% Assentos * km (vazios)	68,1 %
Cadeiras de Rodas * km (vazios)	6.739,8 Cad. Rodas * km
Cadeiras de Rodas * km (total)	10.155,6 Cad. Rodas * km
% Cadeiras de Rodas * km (vazios)	66,4 %

Fonte: Znamensky (2000)

A partir do resultado da heurística de inserção paralela, foi aplicada a heurística de melhoria descrito anteriormente. Vale observar que, durante o processo de re-inserção de solicitações, é possível ocorrer tanto o aumento quanto a redução do número de veículos utilizados. O aumento do número de veículos utilizados se dá com o aproveitamento de veículos disponíveis que estão ociosos, não utilizados na programação obtida pela heurística de inserção paralela.

O número de iterações foi estipulado em 10, sendo testadas as situações em que a utilização da frota ociosa é ou não permitida. Isto é, no primeiro caso pode haver um aumento do número de veículos utilizados, ao passo que no segundo caso, apenas a redução é possível. Os indicadores relativos à programação obtida após a aplicação do processo de melhoria são apresentados na Tabela 4.6. Houve uma melhoria significativa na distância total percorrida e uma redução do percurso ocioso (sem transportar passageiros).

Embora não tenha sido possível obter os dados relativos à programação efetivamente executada pela São Paulo Transporte S.A. para o dia correspondente ao estudo de caso, pode-se intuir a existência de ganhos em relação à prática manual, visto que se obteve uma programação que utiliza apenas 69% da frota disponível.

Nota-se ainda que, a utilização da frota remanescente no processo de melhoria não é particularmente vantajosa, já que, tanto em termos de distância total programada, quanto em número de veículos, os melhores resultados foram obtidos sem a utilização da frota ociosa. Observa-se que o elevado índice de ociosidade da oferta de assentos e cadeiras de rodas é

resultado do acúmulo de solicitações em determinados horários. Tais restrições de janelas de tempo fazem com que grande parte dos veículos acabe por transportar apenas um cliente nos horários de maior demanda, o que acaba por se refletir nos indicadores de ociosidade obtidos.

Tabela 4.6: Indicadores relativos à programação obtida após o processo de melhoria

Indicadores	Utiliza Frota Ociosa na Reinserção ?	
	Sim	Não
Rotas programadas	62	58
% utilização da frota	73,80%	69,00%
Percurso total (km)	5.732,8	5.728,9
Percurso médio por rota (km)	92,5	98,8
% percurso vazio	33,10%	33,60%
assentos * km vazio	10.592,3	10.585,3
assentos * km total	16.375,3	16.470,3
% assentos * km vazio	64,70%	64,30%
cad. rodas * km vazio	5.669,3	5.593,3
cad. rodas * km total	9.010,6	8.951,6
% cad. rodas * km vazio	62,90%	62,50%

Fonte: Znamensky (2000)

O modelo apresentado permitiu ainda a avaliação de diferentes alternativas operacionais, com utilização de veículos de maior capacidade, redistribuição da frota pelas garagens, e variação nas janelas de tempo adotadas para as solicitações (Znamensky, 2000). O desempenho computacional foi satisfatório, uma vez que os resultados puderam ser obtidos em poucos minutos, mesmo para um problema com número de atendimentos comparável aos maiores já avaliados na literatura. Adicionalmente, o problema analisado apresenta a complexidade dos veículos estarem localizados em 47 diferentes garagens.

4.3.5 Considerações Finais

Os resultados obtidos permitem identificar algumas medidas que resultariam em ganhos operacionais para o serviço oferecido pela SPTrans. Em primeiro lugar, observa-se que os horários desejados pelos usuários apresentam uma grande concentração em determinados períodos do dia. Uma melhor distribuição dos mesmos, se possível, reduziria o pico de atendimentos nos referidos períodos, permitindo um melhor aproveitamento dos veículos, com diminuição de ociosidade e eventualmente redução da frota necessária. Naturalmente, tal

iniciativa não pode ser unilateral, devendo ser feita em conjunto com as instituições de atendimento cadastradas (hospitais, centros de fisioterapia, etc.) e usuários credenciados.

Em segundo lugar, a adoção de uma frota mais diversificada poderia trazer benefícios operacionais, já que há rotas com elevado índice de ociosidade, e outras que poderiam se beneficiar da existência de veículos de maior capacidade. Essa questão está fortemente ligada às restrições de janelas de tempo, em especial com o problema comentado anteriormente, da concentração de solicitações em determinados períodos do dia.

Uma extensão mais imediata do trabalho é introduzir uma fase de pós-processamento para ajuste dos horários programados para cada rota. Embora tenham sido obtidos bons resultados com tempos de processamento reduzidos, uma extensão natural deste trabalho seria a incorporação, no procedimento de busca local, de um mecanismo de controle baseado em busca tabu, incluindo o mecanismo de “*tabu thresholding*”, conforme apontado por Toth e Vigo (1997).

Finalmente, pode-se propor a incorporação de outras restrições operacionais, tais como inclusão de horários de almoço para os motoristas, considerada por Madsen *et al.* (1995).

4.4 O PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS COM CARGA COMPLETA E JANELAS DE TEMPO

No problema de roteirização de veículos com carga completa (do inglês “*full truck load*”), objeto da pesquisa de mestrado de Mourad (2005) e orientada por este autor, a demanda a ser atendida é dada na forma de viagens de uma única origem para um único destino, sem paradas intermediárias para entregas parciais durante um percurso; ou seja, o veículo é totalmente carregado em um determinado local de coleta (origem) e descarregado completamente em um único local de entrega (destino). Cabe ressaltar que este termo não está restrito aos casos em que a capacidade total de carga do veículo é utilizada, mas ao fato da viagem não admitir paradas intermediárias para descargas parciais.

O transporte de cargas completas em geral envolve veículos de grande capacidade, usualmente um cavalo mecânico tracionando uma carreta semi-reboque e/ou reboque, que pode ser acoplada e desacoplada do mesmo. Nesse caso, a roteirização envolve a seqüência de

viagens realizadas por cada veículo trator, que não aguarda a carga e/ou a descarga da respectiva carreta nos pontos extremos de cada viagem. Isso se deve aos tempos normalmente elevados de espera, carga e descarga, e que muitas vezes não podem ser previstos: por exemplo, grandes filas para descarga em portos, ou a carreta acabar servindo como estoque, permanecendo parada por vários dias em terminais com problemas de capacidade. Tendo em vista serem os cavalos mecânicos ativos cujo investimento é elevado, é aconselhável que permaneçam parados o menos possível, embora existam situações em que o conjunto cavalo-carreta sempre roda acoplado.

Embora não muito explorado na literatura, esse problema de roteirização de veículos com carga completa pode ser encontrado em diversas situações reais. Um transportador rodoviário de cargas, por exemplo, recebe diversas solicitações de transporte com diferentes origens e destinos. Esta empresa, para prestar um bom atendimento a seus clientes e maximizar a produtividade de seus veículos, enfrenta, em seu cotidiano, decisões de como alocar sua frota de veículos à demanda de viagens. Adicionalmente, essas empresas, em geral, possuem alguns veículos próprios, e também utilizam os chamados “agregados”, isto é, motoristas autônomos que possuem seus próprios veículos e prestam serviço para transportadoras e embarcadores de carga, sendo contratados para a realização de viagens específicas e eventuais (“*spot*”).

Outra situação prática que envolve o transporte de cargas completas está relacionada, por exemplo, a um embarcador que transfere produtos de suas unidades fabris para pontos de armazenamento e centros avançados de distribuição. Estas transferências, bem como a distribuição dos produtos destes locais de armazenagem até grandes clientes, também caracterizam uma demanda de viagens com carga completa. Nos casos em que estas empresas possuem uma frota própria ou dedicada para a realização destas viagens, a contratação de um transportador para suprir o déficit de oferta de veículos representa a contratação de uma frota “*spot*”.

Pode-se citar ainda o transporte de contêineres. Para exportar uma carga, um contêiner vazio deve ser retirado em algum porto onde o armador proprietário do contêiner opere. Este contêiner é levado vazio até o ponto de carregamento, onde é estufado (mesma nomenclatura para carregado). Depois de carregado, outro transporte deve ser efetuado para levar o contêiner cheio até o porto de saída da carga. Nos casos em que o fluxo de exportações é

muito elevado, a racionalização dos custos de transporte rodoviário de contêineres (vazios e cheios) pode proporcionar significativas reduções de custos para as empresas.

Um dos primeiros trabalhos sobre o assunto é o de Ball *et al.* (1983), que trataram do problema de dimensionamento de frota para atendimento de viagens com carga completa na presença de uma opção de contratação de terceiros. É um dos poucos trabalhos encontrados que consideram diferentes formas de contratação da frota, uma dedicada e outra comum, geralmente remunerada por viagem. Os autores propuseram três heurísticas, buscando determinar a quantidade de veículos da frota dedicada, quais viagens devem ser alocadas aos veículos da frota dedicada e da comum, considerando restrições de duração máxima dos roteiros e um horizonte de planejamento de uma semana. Fisher *et al.* (1995) modelam o problema de roteirização de veículos com carga completa como um problema de fluxo em rede, gerando uma solução inviável, uma vez que corresponde a ciclos fechados que não incluem a garagem e eventualmente desrespeitam a duração máxima dos roteiros; em seguida, algumas heurísticas são aplicadas a esta solução visando torná-la viável. O problema tratado nesse trabalho é muito próximo ao encontrado em Ball *et al.* (1983), porém considera uma frota fixa que deve partir e retornar a uma mesma base; adicionalmente não considera a possibilidade de utilização de terceiros. Também Equi *et al.* (1997) consideram uma frota fixa para atendimento da demanda, e restrições de janela de tempo, sendo proposta uma solução baseada em decomposição e relaxação Lagrangeana. Jula *et al.* (2001) trataram do problema de movimentação de contêineres em áreas urbanas, que foi modelado como um problema de múltiplos caixeiros viajantes com janelas de tempo, sendo propostos dois métodos de solução, um exato, baseado em programação dinâmica, e outro híbrido, utilizando programação dinâmica e algoritmo genético. Ambos dividem o problema em duas fases, uma de enumeração de soluções viáveis e outra de escolha da melhor solução. O primeiro deles, apesar de exato, permite a resolução de problemas com no máximo 15 a 20 nós para problemas gerados aleatoriamente. Stringher (2004) trata do problema de designação de rotas para uma frota dedicada, aplicando uma metodologia similar à de Souza (1999), em que o problema é dividido em duas etapas, sendo a primeira para geração de roteiros e seus respectivos custos associados e a segunda etapa para a seleção dos melhores roteiros. O enfoque foi de planejamento tático, não sendo consideradas restrições de janelas de tempo, de frota disponível, nem de bases onde se localizam os veículos.

No problema em questão considera-se a possibilidade de utilização de transportadores autônomos ou “*spot*” de maneira complementar, uma prática comum no mercado de

transporte rodoviário brasileiro, e que não encontra similares em termos de escala em relação ao que ocorre em outros países desenvolvidos, onde o custo e a escassez da mão-de-obra, assim como a organização em sindicatos, torna essa alternativa menos atrativa e de uso mais restrito. Conseqüentemente, a grande maioria dos modelos de roteirização encontrados na literatura não contempla essa alternativa de uso de terceiros.

4.4.1 Formulação Matemática

Seja N_V o conjunto de viagens porta-a-porta que devem ser realizadas por uma dada frota de veículos para um dado horizonte de planejamento (por exemplo, uma semana). A cada viagem estão associados um local de origem e um local de destino, bem como uma data de entrega no destino. Cada um dos pontos extremos de uma viagem possui um horário de funcionamento e um tempo de atendimento (coleta/entrega). Não há flexibilidade para antecipação ou postergação de entregas; em outras palavras, caso o veículo se apresente para realizar a entrega antes do dia combinado, ele permanecerá aguardando até a data acordada.

Essa alternativa de representação permite considerar e representar inclusive eventuais serviços que o veículo necessite realizar em um dado local e numa certa data, tais como manutenção, ou descanso da tripulação; nesse caso, define-se uma viagem com o mesmo local de origem e destino e a data em que deve efetuar o serviço. O conjunto de paradas obrigatórias é dado por N_p .

A frota de veículos disponíveis para realizar as viagens é homogênea e pré-fixada, não necessariamente suficiente para realizar todas as viagens programadas; nesse caso, transportadores autônomos ou “spot” podem ser contratados para realizar as viagens não atendidas pela frota disponível. Todas as viagens especiais (paradas em pontos de apoio) são associadas a um veículo específico da frota e devem ser obrigatoriamente realizadas pelo mesmo. Para cada veículo v do conjunto V da frota dedicada, são conhecidos o local e o horário em que o mesmo estará disponível para iniciar uma rota. Esse local pode ser o ponto final de uma viagem anterior ou um ponto de apoio. No início do problema esse local será tratado como uma garagem em que o veículo estará disponível no horário indicado. O conjunto de garagens é dado por G .

Definem-se ainda os conjuntos $N = N_v \cup N_p$, representando todos os nós (correspondendo a viagens ou paradas obrigatórias) que devem ser atendidos e $S = G \cup N$.

Cada viagem é representada por um único nó $i \in N$, conforme ilustrado na Figura 4.5. As setas indicam solicitações de viagens; os triângulos representam as viagens especiais (isto é, alguma parada em um ponto de apoio), sendo que o número indica o veículo correspondente; os quadrados indicam as garagens de onde partem os veículos no início do período. Deve-se ressaltar que as distâncias d_{ij} e d_{ji} entre dois nós $i, j \in N$ são assimétricas, como mostrado na Figura 4.5, uma vez que, por exemplo, a distância d_{ij} representa o percurso entre o local de destino da viagem i e o local de origem da viagem j ,

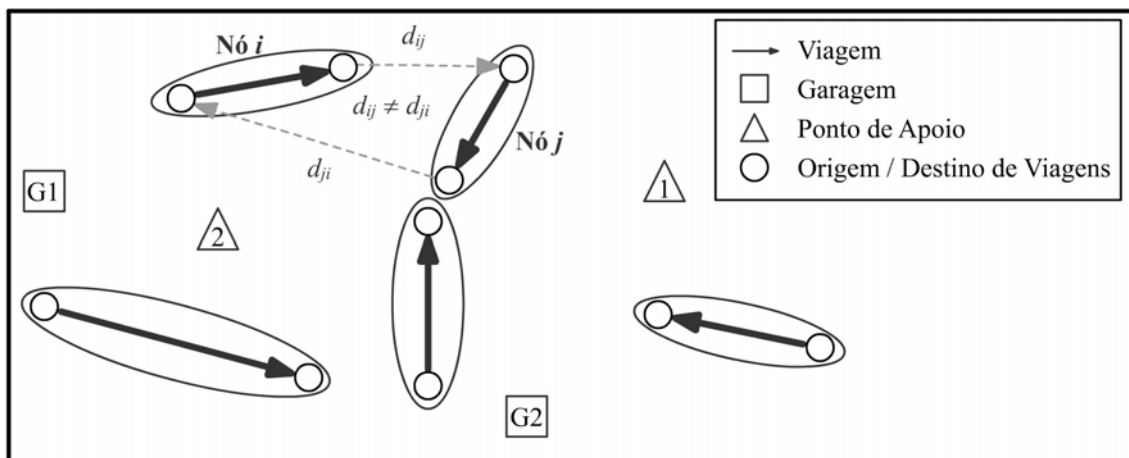


Figura 4.5: Representação do problema

Fonte: Mourad (2005)

Para cada nó $i \in S$ é conhecido um tempo de atendimento $s_i \geq 0$ que corresponde ao tempo total em que o veículo permanecerá no referido nó. Para os nós pertencentes ao conjunto das viagens, este tempo corresponde à soma dos tempos de coleta na origem, o tempo de traslado até o destino e o tempo de entrega no destino da carga. Já para os nós do conjunto de pontos de apoio, esse tempo corresponde ao tempo de parada para manutenção do veículo ou descanso da tripulação. Assume-se que este tempo seja nulo para os nós de garagem.

Atribui-se a cada nó $i \in S$ uma janela de tempo $[a_i, b_i]$ que define o período em que o atendimento pode ser iniciado ($a_i \leq b_i$). Para os nós referentes às garagens, os horários de início e fim da janela de tempo são iguais ao horário de disponibilidade do veículo correspondente. Nos nós referentes aos pontos de apoio, a janela de tempo é definida como o

horário de funcionamento do local de parada no dia em que esta deverá ser realizada. Para os nós do conjunto de viagens, o cálculo é mais complexo. Nesses nós, a janela deve refletir o horário em que a coleta na origem da viagem pode ser iniciada, de forma que o prazo de atendimento e os horários de funcionamento dos locais de origem e destino sejam respeitados. Maiores detalhes de como é determinado esse intervalo horário no local de origem da viagem podem ser encontrados em Mourad (2005).

Seja C_i o custo da viagem $i \in N_V$ com um veículo da frota dedicada e C_{dist} o custo variável unitário com a distância percorrida, utilizado no cálculo do custo dos percursos improdutivos (deslocamentos entre as viagens). Como a frota de cavalos mecânicos disponíveis é conhecida, e seu dimensionamento está fora do escopo deste trabalho, desconsiderou-se a parcela de custos fixos dessa frota. Por outro lado, seja CN_i o custo de atendimento de uma viagem $i \in N_V$ por um veículo da frota “spot”.

Definem-se ainda (i) $r_{vp} = 1$ se o veículo $v \in V$ tiver uma parada de serviço programada no local $p \in N_p$, 0 caso contrário; (ii) $q_{vg} = 1$ se o veículo $v \in V$ estiver alocado à garagem $g \in G$ no início do período, 0 caso contrário. Vale notar que, cada veículo pode ser associado a apenas uma garagem, porém pode realizar diversas paradas, que devem obrigatoriamente ser realizadas.

A formulação matemática do problema compreende as seguintes variáveis de decisão: (i) $x_{ij}^v = 1$ se a viagem $j \in S$ sucede a viagem $i \in S$ pelo veículo $v \in V$, 0 caso contrário; (ii) T_i = horário de início da viagem $i \in S$.

O problema consiste em gerar as rotas dos cavalos mecânicos, de modo que todas os nós (viagens) sejam atendidos, ou pela frota própria ou pelos veículos de terceiros, minimizando o custo total. Sua formulação matemática é apresentada a seguir:

$$\text{Minimizar } \sum_{j \in N_v} \left(C_j \sum_{i \in S} \sum_{v \in V} x_{ij}^v \right) + C_{dist} \sum_{v \in V} \sum_{i \in S} \sum_{j \in N} d_{ij} x_{ij}^v + \sum_{j \in N_v} \left(CN_j \left(1 - \sum_{i \in S} \sum_{v \in V} x_{ij}^v \right) \right) \quad (4.33)$$

sujeito a:

$$\sum_{j \in N} x_{gj}^v \leq q_{vg} \quad , \text{ para todo } v \in V, g \in G \quad (4.34)$$

$$\sum_{i \in S} x_{ip}^v = r_{vp} \quad , \text{ para todo } v \in V, p \in N_p \quad (4.35)$$

$$\sum_{v \in V} \sum_{i \in S} x_{ip}^v \leq 1 \quad , \text{ para todo } p \in N_v \quad (4.36)$$

$$\sum_{i \in S} x_{ip}^v - \sum_{j \in N} x_{pj}^v \geq 0 \quad , \text{ para todo } v \in V, p \in N \quad (4.37)$$

$$a_i \leq T_i \leq b_i \quad , \text{ para todo } i \in S \quad (4.38)$$

$$T_i + s_i + t_{ij} - T_j \leq (1 - x_{ij}^v)M \quad , \text{ para todo } v \in V, i \in S, j \in N, i \neq j \quad (4.39)$$

$$x_{ij}^v \in \{0, 1\} \quad , \text{ para todo } v \in V, i \in S, j \in N \quad (4.40)$$

A função objetivo (4.33) é composta por três parcelas: custo das viagens atendidas pela frota dedicada, custo dos deslocamentos improdutivos utilizando a frota dedicada, e o custo das viagens realizadas pela frota “spot”. A restrição (4.34) assegura que cada veículo parta da sua garagem, enquanto que a restrição (4.35) impõe que cada veículo realize as paradas operacionais a ele atribuídas. A restrição (4.36) assegura que cada viagem do conjunto N_v seja realizada por, no máximo, um veículo da frota dedicada; a não inclusão da viagem no roteiro de algum veículo da frota dedicada acarreta uma penalidade na função objetivo pela necessidade de contratação “spot”. A restrição (4.37) está associada à continuidade de fluxo dos veículos da frota dedicada. Já a restrição (4.38) impõe que o início de cada viagem por um veículo da frota dedicada ocorra dentro da sua respectiva janela de tempo, enquanto que a restrição (4.39) assegura a continuidade temporal das rotas, assim como impede a formação de subroteiros (“subtours”). Por fim, a restrição (4.40) impõe que as variáveis de decisão x_{ij}^v sejam binárias.

4.4.2 Estratégia de Solução Baseada em Busca Tabu

O problema de roteirização de veículos com carga completa, como os demais problemas de roteirização, apresenta natureza combinatória e complexidade exponencial, impedindo a sua solução através de pacotes de otimização disponíveis comercialmente. Assim, foram propostas heurísticas baseadas em Busca Tabu (Glover, 1990; Glover e Laguna, 1993, 1997) para a sua solução, tendo em vista ser a metaheurística de maior sucesso na resolução de problemas de roteirização de veículos (Laporte *et al.*, 2000).

Em linhas gerais a estratégia de solução proposta compreende uma heurística de construção simples, a fim de se obter uma solução inicial viável, e uma heurística de melhoria, baseada em busca local utilizando Busca Tabu.

Na primeira fase, de construção de uma solução inicial, implementou-se uma heurística de inserção rápida, inspirada nas idéias de Solomon (1987). Inicialmente são construídas tantas rotas quantos forem os veículos da frota dedicada; em seguida, são inseridos os demais atendimentos obrigatórios, ou seja, os que representam paradas em pontos de apoio. Busca-se inserir as viagens uma a uma na solução, procurando a melhor inserção possível na solução atual. Isso é feito através da verificação, dentre todas as viagens não programadas (ou seja, atendidas pela frota “*spot*”) a possibilidade de inserir cada uma delas em cada uma das rotas dos veículos da frota dedicada. É escolhida para inserção, dentre todas as possíveis inserções viáveis, a que trouxer o maior ganho em termos de redução de custo, até que não existam mais inserções viáveis com melhoria da solução. Deve-se destacar ser possível que alguma viagem não seja inserida no roteiro de nenhum veículo da frota dedicada, uma vez que o seu custo de atendimento seja maior que o da contratação de uma viagem “*spot*”.

Uma vez obtida uma solução inicial viável, aplica-se uma heurística de melhorias inspirada em Busca Tabu. Foram considerados quatro tipos de movimentos para a exploração da vizinhança de uma solução: inserção, exclusão, realocação e troca. O movimento de *inserção* consiste em alocar uma viagem não programada (i.e., atendida pela frota “*spot*”) a uma das rotas dos veículos da frota dedicada. Analogamente, o movimento de *exclusão* retira uma viagem atendida pela frota dedicada, que passa a ser atendida pela frota “*spot*”. A *realocação* de uma viagem pode ser vista como a aplicação consecutiva dos movimentos de exclusão e inserção, ou seja, retirar uma viagem de uma rota e inserí-la em outro veículo da frota dedicada, ou em outra posição na mesma rota, conforme ilustrado na Figura 4.6. Já no

movimento de *troca*, duas viagens diferentes têm suas posições trocadas, conforme mostrado na Figura 4.7; assim como no movimento de realocação, essas viagens podem ou não estar na mesma rota. A cada iteração são analisadas todas as possibilidades de movimentos, sendo selecionado o que proporciona o maior ganho.

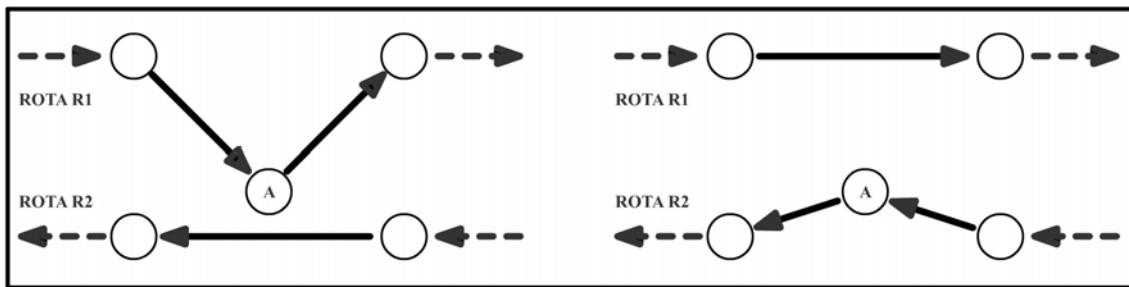


Figura 4.6: Movimento de realocação de uma viagem

Fonte: Mourad (2005)

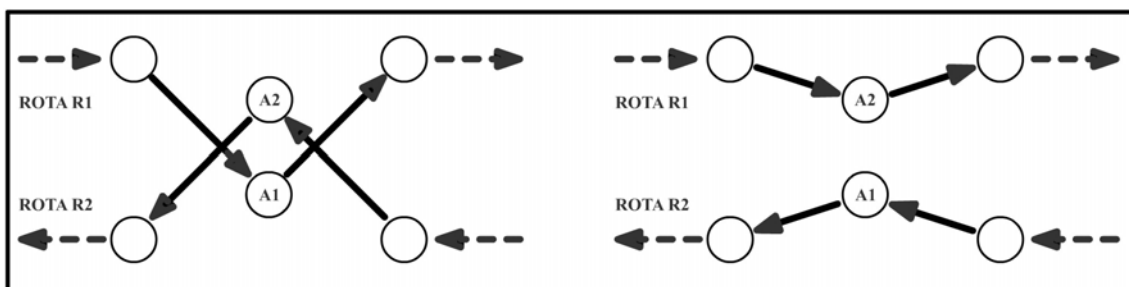


Figura 4.7: Movimento de troca entre viagens

Fonte: Mourad (2005)

Os movimentos de realocação e troca, são equivalentes ao movimento λ -*interchange* (com λ igual a 1) proposto por Osman (1991, 1993). São movimentos simples que já demonstraram bons resultados em implementações de busca tabu como em Taillard (1993). Entretanto, esses dois movimentos são aplicáveis apenas em atendimentos que estejam alocadas à frota dedicada. Para que um atendimento deixe de ser atendido pela frota dedicada ou pela frota “*spot*” são utilizados os movimentos de exclusão e inserção.

Foram implementadas e avaliadas três diferentes heurísticas de melhorias, que utilizam os mesmos movimentos e se diferenciam pelo período tabu (“*tabu tenure*”), pela estratégia de diversificação e pelo uso de uma vizinhança diferenciada, como mostrado no Quadro 4.2.

Em todas as heurísticas, a proibição consiste em impedir que uma viagem que fosse retirada de uma determinada rota retorne a ela durante o período em que o par viagem-rota estiver classificado como tabu. Este tipo de restrição é utilizado em implementações que obtiveram grandes sucessos na resolução de problemas de roteirização de veículos como o “*Taburoute*” de Gendreau *et al.* (1994, 1999) e as implementações de Taillard (1991, 1995)

Quadro 4.2: Características das Heurísticas de Melhorias Baseadas em Busca Tabu

	TS-2	TS-3	TS-4
Período Tabu	Aleatório em faixa fixa	Aleatório em faixa ajustável conforme busca	Aleatório em faixa ajustável conforme busca
Diversificação	Não possui	Aumenta faixa de período tabu até melhorar solução	Aumenta faixa de período tabu até melhorar solução ou atingir nível de piora
Vizinhança	Completa	Completa	Granular

Fonte: Mourad (2005)

Considerou-se ainda que todas as viagens não roteirizadas (i.e. atendidas pela frota “*spot*”) estão em uma mesma rota. Assim, se um movimento de inserção é realizado, esta viagem fica proibida de ser realocada à frota “*spot*” pelo período tabu. Considera-se ainda um critério de aspiração, que permite que um movimento com status tabu possa ser executado somente se levar a uma melhoria da melhor solução corrente.

Na heurística TS-2 o período tabu é escolhido aleatoriamente, em cada iteração, dentro de uma faixa pré-definida, tendo sido consideradas, nos experimentos computacionais realizados, as seguintes faixas: 5-15 iterações, 10-20, 20-40 e 30-60 iterações.

Já para a heurística TS-3, a faixa do período tabu pode aumentar quando a busca fica estagnada. Tal recurso, inspirado no trabalho de Pureza e França (2001) e conhecido como “*moving gaps*”, visa promover uma diversificação da vizinhança quando a busca fica estagnada, isto é, deixa de convergir para uma solução melhor. Assim, o aumento do número de iterações de um movimento que fica tabu torna a vizinhança da solução mais restrita. Com isso, há uma tendência de se realizar um movimento mais distante da solução atual e conduzir a busca para uma região no espaço de soluções ainda pouco explorada. Considerou-se um limite inferior de 5 iterações e uma amplitude de 10 iterações; assim, as faixas consideradas foram: 5-15, 15-25, 25-35 iterações, e assim sucessivamente. O critério para mudança de faixa é o número de iterações sem melhoria para uma mesma faixa, sendo esse valor sempre igual a duas vezes o limite superior da faixa, retornado à faixa inicial toda vez que uma nova solução

for encontrada. Dessa forma, depois de, por exemplo, trinta iterações sem melhoria, a faixa considerada para a determinação do período tabu numa dada iteração mudaria de 5-15 iterações para 15-25 iterações. Se passadas outras 50 iterações, não houver melhoria da solução, a faixa é aumentada novamente.

A heurística TS-4 incorpora duas modificações importantes em relação à heurística TS-3. O período tabu retorna à faixa inicial não apenas quando uma melhor solução for encontrada (como ocorre na TS-3), mas também quando a solução atingir um determinado nível de piora. A idéia é evitar que a vizinhança a ser explorada se torne excessivamente restrita, em decorrência de períodos tabu cada vez mais longos. Assim, uma vez atingido um certo nível de degradação das soluções (em decorrência de períodos tabu crescentemente mais longos), o que sugere que uma nova região esteja sendo explorada, retorna-se à faixa inicial em que o período tabu é escolhido. Adicionalmente, foi implementado um conceito de redução de vizinhança inspirado no Tabu Granular proposto por Toth e Vigo (2003), buscando reduzir drasticamente as vizinhanças a serem exploradas através da exclusão de movimentos que envolvam apenas elementos pouco prováveis de participar de boas soluções viáveis. Estes elementos, no problema de roteirização, são identificados como longos arcos. No caso abordado neste trabalho, são deslocamentos improdutivos de grandes distâncias realizados pela frota dedicada. Isso não significa dizer que nenhum arco longo fará parte da solução. Serão desconsiderados os movimentos que incluïrem somente arcos longos na solução. Para definir o que são arcos longos e arcos curtos, utilizou-se um filtro simples: o tamanho médio dos arcos (deslocamentos improdutivos da frota dedicada) da solução inicial multiplicado por um parâmetro de controle β . Se o arco for maior que este valor, ele é considerado longo, caso contrário, não sofrerá qualquer tipo de restrição. Assim, quanto menor o valor de β , mais restrita será a vizinhança granular. Foram considerados valores de β entre 1,0 e 4,0.

4.4.3 Experimentos Computacionais

A fim de se avaliarem os resultados das heurísticas propostas, foi necessário gerar algumas instâncias de problemas teste, uma vez que não foram encontrados na literatura problemas de *benchmark* com as mesmas características do problema considerado, ou seja, (i) a presença de uma frota “*spot*” em complemento à frota dedicada, (ii) a existência de janelas de tempo, (iii) de paradas especiais para execução de serviços, e a (iv) consideração de múltiplos depósitos.

Buscou-se gerar problemas com tamanho e características compatíveis com os problemas encontrados na prática, investigando também variações do problema estudado. Para isso, foram criados problemas com um horizonte de planejamento de 5 dias e viagens com duração média de 1 dia (considerando uma velocidade média de 50 km/h). Em cada instância, foram geradas 100 viagens no período (média de 20 por dia). Foram definidas janelas de tempo nos locais de origem e destino de cada viagem correspondentes a meio dia ou um dia inteiro; considerou-se dois níveis de restrição de janelas horárias para atendimento, dados pela parcela do número de locais que funcionam em apenas meio período: 20 e 50%.

Para a geração das viagens, inicialmente foram gerados os pontos de origem e de destino das mesmas, distribuídos aleatoriamente em uma região de formato quadrado com aresta de 1.200 km; em seguida definiu-se o dia em que a carga deve ser entregue em seu destino. As datas de entrega destas viagens foram programadas de forma a ficarem igualmente distribuídas no horizonte considerado. Definiram-se duas quantidades de viagens: uma que resultou em uma média diária de 50 viagens e outra com média diária de 100 viagens. Para atender essa demanda, uma frota dedicada de 50 veículos é disponibilizada no início do horizonte de planejamento, em uma garagem única, localizada no centro de gravidade de todas as origens das viagens. Julgou-se que essa frota de 50 veículos é suficiente para atender 50 viagens diárias (dado que a duração média das viagens é de 1 dia), porém, para os problemas com 100 viagens diárias, espera-se que esta oferta de veículos seja restritiva ao problema. Com o intuito de simular algumas paradas obrigatórias, 30% dos veículos devem obrigatoriamente voltar para a garagem no último dia. Assim, foram criadas quatro classes de problemas, definidas pelo número de viagens por dia e pelo nível de restrição das janelas de tempo. Para cada uma dessas classes, foram geradas três instâncias de problemas.

Instâncias de tamanho reduzido para cada uma das quatro classes tiveram que ser geradas, em termos do número de viagens por dia, porém mantendo os demais atributos, a fim de permitir a sua solução através de um pacote comercial de otimização (GAMS com o solver CPLEX 7.0) e os resultados poderem ser comparados com os obtidos com as três heurísticas propostas, permitindo avaliar a qualidade das soluções obtidas pelas heurísticas propostas.

Em geral as heurísticas geraram bons resultados, conforme mostrado na Tabela 4.7, em particular a TS-4, que encontrou as melhores soluções, todas elas menos de 2% acima da solução do CPLEX. Comparadas com os resultados do CPLEX com restrição de tempo de processamento, todas as heurísticas apresentaram melhores resultados em cinco das doze

instâncias criadas, em particular a TS-4 que encontrou as melhores soluções, todas elas menos de 2% acima da solução ótima, em tempos de processamento reduzidos, em geral inferiores a 2 minutos no caso das heurísticas TS-3 e TS-4, enquanto que, em alguns casos, a solução exata levou muito mais tempo para ser obtida. Os melhores resultados foram obtidos com a heurística TS-4, que apresentou um desvio médio de apenas 1,03% em relação às soluções ótimas e os menores tempos de processamento, seguindo-se a heurística TS-2 com desvio médio de 1,70% e a heurística TS-3 com desvio médio de 1,82%; entretanto, os tempos de processamento da heurística TS-2 foram significativamente maiores que os das heurísticas TS-3 e TS-4. Considerando-se um limite de tempo máximo de processamento de 5 minutos para o CPLEX, todas as heurísticas apresentaram melhores resultados em cinco das doze instâncias criadas, sendo que as heurísticas TS-3 e TS-4 em tempos de processamento bem menores que o limite de tempo imposto.

Tabela 4.7: Resultados para problemas reduzidos

Instância	CPLEX 5 Min	TS-2	TS-3	TS-4
P11 _r	0.64%	2.96%	3.33%	1.77%
P12 _r	---	0.56%	0.73%	0.32%
P13 _r	1.96%	1.57%	1.62%	1.31%
P21 _r	0.00%	1.55%	1.51%	1.41%
P22 _r	0.45%	2.14%	2.53%	1.80%
P23 _r	---	1.02%	1.15%	0.77%
P31 _r	---	1.98%	2.38%	1.29%
P32 _r	0.03%	1.86%	1.85%	0.66%
P33 _r	2.87%	1.98%	2.08%	0.66%
P41 _r	0.00%	1.43%	1.33%	0.81%
P42 _r	0.00%	1.30%	1.17%	0.50%
P43 _r	0.00%	1.92%	2.12%	1.09%
		1.70%	1.82%	1.03%

Fonte: Mourad (2005)

Também para os problemas completos, a heurística TS-4 proporcionou as melhores soluções e os menores tempos de processamento, com os parâmetros β de 4.0 e 1% de piora, embora tenha havido pouca variação dos resultados para outros valores de β e de nível de piora avaliados. Maiores detalhes sobre os experimentos computacionais podem ser encontrados em Mourad (2005).

4.4.4 Considerações Finais

Os resultados obtidos indicam a eficiência da Busca Tabu na resolução do problema de roteirização com carga completa e janelas de tempo. Em particular, os mecanismos de diversificação através de uma mudança controlada da faixa em que o período tabu é escolhido, evitando uma degradação excessiva da solução e gerando uma vizinhança muito restrita, bem como uma redução de vizinhança inspirada no tabu granular, mostraram-se bastante eficientes em termos tanto de qualidade das soluções quanto de tempos de processamento.

Investigações futuras podem incluir outros movimentos e estruturas mais complexas de exploração da vizinhança como, por exemplo, cadeias de ejeção. Além disso, estruturas específicas de memória de longo prazo para guiar estratégias de intensificação e diversificação, como a memória adaptativa, podem ser implementadas visando melhorias na qualidade da solução e de desempenho computacional. Espera-se ainda ser possível aplicar os métodos de solução propostos a problemas reais encontrados na prática das empresas no Brasil.

4.5 O PROBLEMA DE ROTEIRIZAÇÃO DE CAMINHÕES-TANQUE PARA A ENTREGA DE COMBUSTÍVEIS EM POSTOS

O problema de roteirização de caminhões-tanque para a entrega de combustíveis apresenta, em relação aos problemas de roteirização tradicionais, a complexidade adicional associada à compartimentalização dos veículos, conforme apontam Bechara e Galvão (1984). Mais especificamente, o caminhão-tanque é normalmente dividido em vários compartimentos, sendo que cada cliente deve receber uma quantidade do produto correspondente a um ou mais compartimentos integralmente preenchidos; tampouco é aceitável que o volume de um compartimento seja dividido entre mais de um cliente, uma vez que a confirmação da quantidade entregue pelo cliente no ato do recebimento se dá pela verificação de nível, ou seja, que o compartimento está completamente preenchido. Em outras palavras, tome-se, por exemplo, uma entrega de 5.000 litros de um dado combustível para um cliente. Essa entrega somente pode ser alocada a um compartimento de 5.000 litros, mas não a um compartimento maior (por exemplo, 10.000 litros).

Adicionalmente, caminhões do mesmo tamanho e capacidade total podem possuir diferentes configurações internas em termos de número e tamanho dos compartimentos, (por exemplo, um caminhão de 30.000 litros pode ter 6 compartimentos de 5.000 litros ou 4 compartimentos, sendo 2 de 10.000 litros e 2 de 5.000 litros), o que torna o problema ainda mais complexo.

Até um passado recente o problema era ainda mais complexo, uma vez que tanto os pedidos dos postos quanto os compartimentos dos veículos podiam ser de qualquer tamanho, desde que múltiplos de 1.000 litros. Atualmente, as companhias distribuidoras restringiram os pedidos a múltiplos de 5.000 litros, o que levou a uma padronização da frota em termos das capacidades dos compartimentos (5 ou 10 mil litros).

Trata-se de um problema cujas dimensões o tornam bastante relevante e complexo. Conforme indicado pela ANP - Agência Nacional de Petróleo (2001), os postos revendedores de combustíveis automotivos (postos de combustível) totalizavam, em 2000, 29.111 estabelecimentos em todo o território nacional, sendo 7.878 localizados apenas no Estado de São Paulo. A maior distribuidora do Brasil possuía, sozinha, 1.546 postos neste estado.

Poucos trabalhos na literatura trataram desse problema de distribuição de combustíveis. Bechara e Galvão (1984) propõem, como alternativa para modelar as restrições de compartimentos, a proibição de combinações de pedidos em veículos que sejam impossíveis de ocorrerem pelas configurações de compartimentos existentes em um mesmo veículo. Ou seja, toda ligação entre quaisquer dois clientes cujas demandas não correspondam a dois compartimentos existentes no mesmo veículo deve ser proibida. Uma alternativa proposta pelos autores é a substituição da capacidade dos veículos por restrições associadas a uma matriz que indique os compartimentos disponíveis em cada veículo. Para cada caminhão, então, consideram-se capacidades discretas (ao invés de contínuas).

Brown e Graves (1981) resolveram o problema de veículos que atendem um único cliente por viagem a partir de uma única base. Os autores propuseram um modelo de programação inteira no contexto de um sistema automatizado de despacho. A grande quantidade de dados e o volume de transações do banco de dados, porém, já consumiam a maior parte da capacidade de processamento do sistema, restando pouca capacidade para a execução da otimização em tempo real pelo método proposto. Os autores propuseram ainda uma heurística que, após diversos ajustes e testes, permitiu o módulo de automação da distribuição permitindo obter bons resultados.

Avella, Boccia e Sforza (2004) trataram de um problema de distribuição de combustível para postos, a partir de um único local, com uso de frota heterogênea de caminhões-tanque em termos de capacidades e perfil de compartimentos. O problema foi resolvido a partir de uma formulação baseada em “*set partitioning*”, sendo utilizado um algoritmo “*branch-and-price*” para a sua resolução. Para tanto, utilizou-se ainda uma heurística rápida a fim de limitar o número de colunas geradas.

Deve-se destacar que nenhum dos artigos encontrados na literatura contemplou as restrições consideradas neste trabalho, objeto da pesquisa de mestrado de Feriancic (2005), orientada por este autor, buscando representar a realidade brasileira da entrega de combustíveis em postos de abastecimento.

4.5.1 Modelo Matemático

Seja $I = \{1, 2, \dots, |I|\}$ o conjunto de clientes (postos de abastecimento) que podem efetuar pedidos de entrega de combustível. A cada cliente estão associados um ou mais pedidos, sendo o conjunto de pedidos dado por $P = \{1, 2, \dots, |P|\}$, sendo que cada pedido corresponde a um único tipo de combustível. Para cada pedido $p \in P$ é conhecido o respectivo cliente $i \in I$, dado por $Cliente_p$, e volume a ser entregue Vol_p , múltiplo de 5 mil litros. Um aspecto importante da modelagem são os pedidos que obrigatoriamente necessitam de um compartimento de 5 mil litros (ou seja, aqueles de 5, 15 ou 25 mil litros). Para tanto, define-se, para cada pedido $p \in P$, a constante $R5_p = 1$ se o pedido p necessita de um compartimento de 5 mil litros, 0 caso contrário.

Seja $V = \{1, 2, \dots, |V|\}$ o conjunto de veículos que compõem a frota disponível para distribuição. Para cada veículo $v \in V$ são conhecidos o custo unitário variável com a distância percorrida cv_v e o custo fixo cf_v diário; define-se ainda $Vmax_v$ como sendo a capacidade volumétrica máxima de cada veículo. Os compartimentos presentes em cada veículo $v \in V$ podem ser de 5.000 ou de 10.000 litros; o total deles é dado por nCp_v . Considerando-se que os volumes a serem entregues sejam múltiplos de 5.000 litros, define-se ainda o número de compartimentos de 5.000 litros de cada veículo v , dado por $V5_v$; busca-se, desta forma, garantir que um pedido de 5.000 litros não seja alocado a um compartimento de 10.000 litros.

Os veículos diferenciam-se ainda pelo padrão de pintura e pela propriedade (próprio ou de terceiros); o padrão de pintura pode interferir diretamente na possibilidade ou não de um determinado veículo atender um determinado cliente. Conseqüentemente, para cada cliente $i \in I$ define-se ainda o tipo de veículo que pode atendê-lo, dado não só pelo tamanho do veículo, mas também pelo padrão de pintura. Essa limitação de determinados veículos atender determinados clientes é dada por:

$$R_{iv} = \begin{cases} 1, & \text{se o cliente } i \text{ pode ser atendido pelo veículo } v \\ 0, & \text{caso contrário.} \end{cases}$$

O ponto 0 corresponde à base de distribuição de onde partem e para onde retornam os veículos.

Seja $G = (N, A)$ onde $N = I \cup \{0\}$ define o conjunto de nós do problema, e A o conjunto de arcos ou ligações entre esses nós. Para cada arco $(i, j) \in A$ é conhecida a distância d_{ij} entre os nós $i, j \in N$.

As variáveis de decisão são:

- $X_{ij}^v = 1$ se o nó $j \in N$ é visitado imediatamente após o nó $i \in N$ pelo veículo $v \in V$; 0 caso contrário;
- $Y_{pv} = 1$ se o pedido $p \in P$ é alocado ao veículo $v \in V$; 0 caso contrário.

Adicionalmente, define-se uma variável de decisão de fluxo contínua $Z_{ij}^v \geq 0$ para cada arco $(i, j) \in A$ e cada veículo $v \in V$, utilizada nas restrições que evitam a criação de sub-rotas, inspirada no modelo proposto por Gavish e Graves (1978) para o problema do caixeiro viajante.

O problema de roteirização de entrega de combustíveis pode ser formulado como:

$$\min \sum_{v \in V} \left(\sum_{j \in I} cf_v X_{0j}^v + \sum_{i \in N} \sum_{j \in N} cv_v d_{ij} X_{ij}^v \right) \quad (4.41)$$

sujeito a:

$$\sum_{v \in V} Y_{pv} = 1 \quad \forall p \in P \quad (4.42)$$

$$\sum_{i \in I} X_{ij}^v \geq Y_{pv} \quad \forall p \in P, \forall j = \text{Cliente}_p, \forall v \in V \quad (4.43)$$

$$\sum_{p \in P} (R5_p Y_{pv}) \leq V5_v \quad \forall v \in V \quad (4.44)$$

$$\sum_{p \in P} (Vol_p \times Y_{pv}) \leq Vmax_v \quad \forall v \in V \quad (4.45)$$

$$\sum_{i \in N} X_{ij}^v \leq R_{jv} \quad \forall j \in I, \forall v \in V \quad (4.46)$$

$$\sum_{j \in I} X_{0j}^v \leq 1 \quad \forall v \in V \quad (4.47)$$

$$\sum_{i \in N} X_{ij}^v - \sum_{k \in N} X_{jk}^v = 0 \quad \forall j \in N, \forall v \in V \quad (4.48)$$

$$\sum_{k \in N} Z_{ki}^v - \sum_{j \in N} Z_{ij}^v = \sum_{j \in N} X_{ij}^v \quad \forall i \in I, \forall v \in V \quad (4.49)$$

$$Z_{ij}^v \leq nCp_v X_{ij}^v \quad \forall v \in V, \forall i, j \in N \quad (4.50)$$

$$Z_{0j}^v = nCp_v \quad (4.51)$$

$$X_{ij}^v \in \{0,1\} \quad (4.52)$$

$$Y_{pv} \in \{0,1\} \quad (4.53)$$

A função objetivo (4.41) corresponde à minimização do custo total, composto pelas parcelas de custo fixo dos veículos e de custo variável proporcional com a distância total percorrida. A restrição (4.42) impõe que cada pedido $p \in P$ seja alocado a um único veículo, enquanto que a restrição (4.43) assegura que o pedido p faça parte do roteiro do veículo v ao qual foi alocado. Já a restrição (4.44) impede a alocação de um número de pedidos que necessitem de compartimentos de 5.000 litros maior que o número de compartimentos disponíveis no veículo v , ao passo que a restrição (4.45) assegura que a capacidade máxima $Vmax_v$, do veículo v também não pode ser excedida. A restrição (4.46) restringe os veículos que podem atender um dado cliente $j \in I$. As restrições (4.47) e (4.48) asseguram a continuidade de fluxo para cada veículo $v \in V$. As restrições (4.49), (4.50) e (4.51) impedem a formação de subrotas, inspiradas no trabalho de Gavish e Graves (1978). Por fim, as restrições (4.52) e (4.53) definem os domínios das variáveis de decisão binárias.

Com a finalidade de validar o modelo matemático proposto, foi criada uma instância de pequeno porte (10 pedidos, 4 clientes e 4 veículos), implementada em planilha eletrônica e resolvida utilizando o *Solver* do Excel. Os resultados foram comparados com a solução ótima obtida manualmente, permitindo comprovar a exatidão da modelagem proposta; no entanto, o *Solver* levou cerca de 8 horas para encontrar a solução ótima desse problema em um computador Pentium IV / 3.06 GHz.

Mesmo considerando que problemas de roteirização de veículos são de natureza combinatória e de complexidade exponencial, o que impede a solução de instâncias reais utilizando pacotes comerciais de programação linear inteira, buscou-se resolver de maneira exata o problema, utilizando-se o *Concert Technology* para *ILOG CPLEX* versão 9.0, um dos pacotes de otimização mais avançados disponíveis no mercado. A finalidade foi obter soluções exatas para instâncias de porte menor que as encontradas na prática, mas que permitissem validar a heurística proposta descrita mais adiante. Nesta segunda implementação, foram criadas e testadas as soluções de algumas diferentes instâncias, progressivamente maiores.

Entretanto, a complexidade do problema impediu a obtenção da solução ótima, mesmo para instâncias de porte bem reduzido (apenas 18 entregas) e bastante restritas em termos das possíveis combinações, conforme pode ser evidenciado na Tabela 4.8.

Tabela 4.8: Instâncias e Resultados do Processamento no ILOG CPLEX

Nº de Veículos	Nº de Pedidos	$ N $	Nº de Variáveis de Decisão	Nível de Restrição	Tempo CPU (seg)	“Duality Gap”
10	4	5	200	0%	6	0%
10	13	8	1.250	28,6%	153	0%
16	18	13	5.280	57,3%	10.978	11%
16	18	13	5.280	20,8%	3.062	n/d*
31	36	27	44.640	62,2%	2.718	n/d*

* Após o tempo de processamento indicado, nenhuma solução viável ainda havia sido encontrada e o processamento foi interrompido.

Fonte: Feriencic (2005)

Mais especificamente, verificou-se que, ao aumentar o número de pedidos e de veículos, o tempo de processamento aumentou drasticamente. Mesmo essas instâncias de porte muito reduzido e bastante restritos não puderam ser resolvidos até a otimalidade, sendo o *duality gap* ainda muito elevado após mais de uma hora de processamento. Percebeu-se também, que quanto menos restrito fosse o problema (em termos das possíveis combinações de pedidos/veículos), mais demorado era o processamento. Note-se que para os dois últimos problemas não foi possível sequer encontrar uma solução viável após um tempo de processamento bastante elevado, o que confirma a necessidade do uso de uma estratégia de solução heurística para ser possível resolver instâncias reais de problemas de roteirização para entrega de combustíveis.

4.5.2 Heurística de Solução

Inicialmente considerou-se propor uma heurística baseada em busca tabu para a solução do problema, que não se mostrou viável pela dificuldade do problema e suas restrições, que tornam a exploração de uma vizinhança uma tarefa complexa. Mais especificamente, os movimentos que permitem a evolução de uma solução corrente para uma solução vizinha ficam dificultados e são pouco eficientes, tendo em vista as restrições do problema que precisam ser verificadas. Em outras palavras, o esforço computacional para explorar a vizinhança de uma solução é bastante elevado e que requereria um mecanismo de busca demasiadamente complexo, além de potencialmente pouco eficiente.

Assim, optou-se por uma heurística rápida, que permite a obtenção de uma boa solução em tempo reduzido. Essa heurística, inspirada nos trabalhos de Toth e Vigo (1996) e Znamensky (2000), compreende, primeiramente, a classificação dos pedidos e dos caminhões conforme a facilidade de alocação e a utilidade para alocação, respectivamente. Em seguida, busca-se agrupar pedidos próximos, iniciando pelos de mais difícil alocação. O processo de alocação tem como objetivo principal definir quais pedidos serão atendidos por quais veículos. As alocações são realizadas segundo essa ordem de dificuldade, sem considerar o efeito posterior de uma alocação nos demais pedidos ainda não atendidos.

Para tanto, define-se o conceito de utilidade ou serventia de um veículo. Determina-se uma matriz binária de restrições, que indica, para cada veículo i e pedido j , se é possível realizar a alocação de j em i . Ela é atualizada a cada passo do algoritmo, uma vez que é responsável por nortear cada passo da alocação. A serventia de um veículo indica, durante o processo de alocação, quantos pedidos ainda podem ser alocados a ele; ou seja, a serventia de um veículo i é a contagem de quantos entre os pedidos j podem ser atendidos por ele. A facilidade de um pedido ser alocado indica durante o processo de alocação, por sua vez, quantos veículos podem atendê-lo; ou seja, a facilidade de um pedido j é a contagem de quantos entre os veículos i podem atendê-lo naquele momento. Na matriz binária de restrições, um valor “1” indica que é possível a alocação do pedido j no caminhão i , enquanto um valor “0” indica o contrário. A Tabela 4.9 ilustra o conceito do cálculo dos parâmetros de serventia de veículo e facilidade de pedido, que guiam o processo de priorização de pedidos para a heurística de inserção proposta.

Tabela 4.9: Exemplo de Cálculo de Serventia e Facilidade

Ped. \ Veíc.	0	1	2	3	4	5	6	7	8	9	10	Facilidade
0	1	1	0	1	1	0	1	0	0	1	1	7
1	1	1	1	1	1	1	1	1	1	1	1	11
2	1	1	1	1	1	1	1	1	1	1	1	11
3	1	1	1	1	1	1	1	1	1	1	1	11
4	1	1	1	1	1	1	1	1	1	1	1	11
5	1	1	1	1	1	1	1	1	1	1	1	11
6	1	1	1	1	1	1	1	1	1	1	1	11
7	1	1	0	1	1	0	1	0	0	1	1	7
8	0	0	1	0	0	1	0	1	1	0	0	4
9	1	1	0	0	0	0	0	0	0	0	0	2
10	1	1	1	0	0	0	0	0	0	0	0	3
Serventia	10	10	8	8	8	7	8	7	7	8	8	

Fonte: Feriancic (2005)

Exemplificando, na tabela acima, o pedido 7 pode ser alocado a sete diferentes veículos (0, 1, 3, 4, 6, 9 e 10), tendo como facilidade o valor 7. O pedido 8 pode ser alocado a quatro veículos (2, 5, 7 e 8), definindo sua facilidade como 4, portanto mais difícil que o pedido 7. Analogamente, o veículo 1 pode atender dez pedidos (0, 1, 2, 3, 4, 5, 6, 7, 9 e 10), tendo serventia 10, enquanto o veículo 2 pode atender a oito pedidos (1, 2, 3, 4, 5, 6, 8 e 10), tendo como serventia o valor 8.

A heurística rápida de alocação busca minimizar as rotas a serem percorridas pelos veículos, ou seja, a soma das distâncias totais percorridas deve ser a menor possível. Para tanto, foram definidas regras para a alocação dos pedidos aos veículos:

- o pedidos agrupados em um mesmo veículo devem estar o mais próximo possível entre si;
- o pedidos para um mesmo cliente preferencialmente estão agrupados;
- o melhor alocar antes os pedidos mais difíceis de serem alocados, ou seja, mais restritos;
- o um pedido muito difícil só consegue ser combinado com outros pedidos em caminhões mais úteis, ou seja, de maior serventia, uma medida de quantos pedidos ainda podem ser alocados a um caminhão.

Em outras palavras, a cada passo o algoritmo, apesar de míope, deve escolher pedidos e caminhões de forma a gerar uma boa solução. Para poder realizar isso sem o esforço

computacional de comparar diversas combinações, foram utilizados parâmetros que são atualizados a cada iteração. Os pedidos são alocados sequencialmente segundo uma ordenação que leva em conta a sua “dificuldade”. Pedidos mais difíceis, isto é, que podem ser alocados em poucos veículos disponíveis, têm prioridade de alocação. A cada nova alocação, as serventias dos veículos são atualizadas, de modo a guiar o processo.

A fim de superar as limitações dessa heurística rápida, porém míope (“*greedy*”), a mesma foi incorporada a um mecanismo geral de controle, inspirado no GRASP (*Greedy Randomized Adaptive Search Procedure*), proposto por Feo e Resende (1989). Para tanto, a heurística de inserção é reaplicada diversas vezes, até que uma condição de parada seja atingida. De forma simplificada, o processo adotado pode ser explicado como a desalocação aleatória de alguns pedidos, sendo os mesmos re-allocados através da heurística rápida de inserção sequencial. Uma vez removidos aleatoriamente alguns pedidos dos veículos aos quais estavam alocados, é atualizada a lista de dificuldade dos pedidos não atendidos (desalocados) e também a lista dos veículos segundo sua utilidade para alocação. Os pedidos são re-inseridos segundo essa ordem. Note-se que a desalocação aleatória de alguns pedidos dos veículos muda os valores de serventia e dificuldade dos pedidos, fazendo com que a ordem de re-alocação seja diferente em cada iteração e, conseqüentemente, as rotas geradas.

Existem duas diferenças básicas em relação ao GRASP: (i) não utilização de sorteio entre elementos candidatos na fase de construção, realizando-se a iteração de construção da solução através da inserção do elemento de melhor benefício; (ii) supressão de uma heurística específica de busca local, utilizando-se o próprio algoritmo de construção para gerar soluções na vizinhança da solução anterior. Como uma boa referência sobre GRASP sugere-se o trabalho de Feo e Resende (1995), que descreve várias aplicações, incluindo roteirização de veículos.

4.5.3 Experimentos Computacionais

Não se dispunha de um problema real para avaliação da heurística proposta, nem tampouco instâncias na literatura para *benchmark*. Assim, foram geradas instâncias de teste específicas, a semelhança do que foi feito por Solomon (1987), para aferir a aplicabilidade do método e que tornem possíveis futuras comparações com outras estratégias de solução que venham a ser desenvolvidas para problemas de mesma natureza. Buscou-se, na definição das instâncias

geradas, tamanhos compatíveis com a estimativa que se faz a respeito da grandeza de problemas reais. Foi proposta, então, a verificação do desempenho do algoritmo com problemas de ordem de grandeza entre 50 e 100 pedidos, atendendo entre 35 e 75 clientes em um único dia, o que parece razoável para um problema real. A quantidade de veículos disponível foi considerada metade em caminhões próprios e outra igual quantidade em veículos terceirizados.

Assim, foram geradas e testadas três classes de problema, com as seguintes características:

- Classe A: 50 pedidos, 50 caminhões e 35 clientes;
- Classe B: 75 pedidos, 75 caminhões e 53 clientes;
- Classe C: 100 pedidos, 100 caminhões e 75 clientes

Para cada classe foram geradas três instâncias: 20%, 40% e 60% de restrição (em termos das possíveis combinações de pedidos/veículos), totalizando nove instâncias para teste. As instâncias de problema dentro de cada classe foram geradas de forma aleatória, respeitando critérios que as tornassem próximas a problemas reais.

O principal aspecto a ser avaliado na heurística proposta é a calibração com relação ao parâmetro que influencia na variabilidade e na convergência do método proposto. A diferenciação das soluções produzidas em cada iteração depende do número de veículos desalocados em relação à solução anterior.

Como a heurística é sequencial, para um determinado conjunto de pedidos e veículos disponíveis, o resultado é sempre o mesmo. Então o número de veículos desalocado não pode nem ser muito grande, nem ser muito pequeno. Foram testados os valores de $\frac{1}{2}$ e $\frac{1}{4}$ de veículos da frota utilizada para cada instância considerada. Como cada uma das três classes foi gerada com três diferentes graus de restrição, temos na verdade, nove diferentes instâncias. Para cada um dos níveis de restrição, no entanto, foram mantidos os mesmos conjuntos de pedidos, clientes e caminhões. Cada uma dessas nove instâncias foi solucionada utilizando dois diferentes valores para o parâmetro de desalocação: $\frac{1}{2}$ e $\frac{1}{4}$.

Como o resultado de cada rodada é influenciado por fatores aleatórios, inerentes à estratégia de solução proposta, a heurística foi rodada cinco vezes para cada valor de parâmetro de taxa de desalocação ($\frac{1}{2}$ e $\frac{1}{4}$), totalizando dez rodadas para cada uma das nove instâncias, cujos resultados são sumarizados na Tabela 4.10.

Tabela 4.10: Resumo dos Resultados dos Testes Realizados

Instância	Resultado Inicial	Melhor Solução	Taxa	Melhor Resultado	Resultado Médio	Desvio Padrão %	Melhoria Média%	Tempo (s)
A - 20%	7.998,23	6.343,40	½	6.432,78 (+1,4%)	6.580,06 (+3,7%)	1,5%	17,70%	57
			¼	6.343,40 (+0,0%)	6.385,28 (+0,7%)	0,7%	20,20%	36
A - 40%	8.035,50	6.465,03	½	6.644,85 (+2,8%)	6.717,00 (+3,9%)	1,2%	16,40%	56
			¼	6.465,03 (+0,0%)	6.508,27 (+0,7%)	0,8%	19,00%	37
A - 60%	8.230,63	6.404,56	½	6.428,44 (+0,4%)	6.706,18 (+4,7%)	2,6%	18,50%	57
			¼	6.404,56 (+0,0%)	6.578,70 (+2,7%)	1,5%	20,10%	37
B - 20%	10.281,30	9.143,79	½	9.314,87 (+1,9%)	9.407,89 (+2,9%)	0,6%	8,50%	152
			¼	9.143,79 (+0,0%)	9.254,56 (+1,2%)	1,0%	10,00%	100
B - 40%	10.431,20	9.236,31	½	9.305,73 (+0,8%)	9.368,26 (+1,4%)	0,5%	10,20%	153
			¼	9.236,31 (+0,0%)	9.286,97 (+0,5%)	0,7%	11,00%	99
B - 60%	10.433,30	9.329,89	½	9.329,89 (+0,0%)	9.579,55 (+2,7%)	1,7%	8,20%	155
			¼	9.366,85 (+0,4%)	9.489,21 (+1,7%)	0,9%	9,00%	101
C - 20%	14.374,40	12.794,70	½	12.838,90 (+0,3%)	13.030,64 (+1,8%)	0,8%	9,30%	340
			¼	12.794,70 (+0,0%)	12.972,18 (+1,4%)	0,8%	9,80%	214
C - 40%	14.425,10	12.456,30	½	12.797,80 (+2,7%)	13.035,80 (+4,7%)	1,6%	9,60%	341
			¼	12.456,30 (+0,0%)	12.928,30 (+3,8%)	2,3%	10,40%	214
C - 60%	14.435,90	12.668,80	½	13.125,00 (+3,6%)	13.180,08 (+4,0%)	0,5%	8,70%	341
			¼	12.668,80 (+0,0%)	12.998,18 (+2,6%)	1,7%	10,00%	214

Fonte: Feriancic (2005)

Os experimentos computacionais realizados permitiram verificar que os ciclos de desalocação e re-alocação geraram melhorias entre 9% e 20% em relação à solução inicial. Adicionalmente, a taxa de desalocação de ¼ geralmente atinge melhores resultados que a taxa de desalocação de ½, além de ser mais eficiente em termos de tempo de processamento. Não houve diferença de desempenho significativo para alteração entre os níveis de restrições 20%, 40% e 60%. Observou-se ainda uma tendência de comportamento não exponencial do tempo de processamento em relação ao tamanho do problema.

4.5.4 Considerações Finais

O problema proposto configura-se como a roteirização de parte uma frota disponível heterogênea e veículos compartimentados com objetivo de minimizar o custo total para realizar a entrega de um conjunto de pedidos a diversos clientes. Embora seja um problema de roteirização de veículos, suas características e restrições, em particular as restrições de capacidade dos diferentes compartimentos de cada veículo, bem como as restrições de tamanho e tipo de veículo que pode atender cada cliente, tornam esse problema complexo e impedem a adoção das heurísticas clássicas para outros problemas de roteirização.

A heurística proposta permitiu obter soluções viáveis e de boa qualidade. Se a precisão do método em relação à solução ótima não pode ser verificada, foi comprovada a aplicabilidade prática do método desenvolvido no suporte à alocação feita hoje manualmente em diversos locais. A fim de aprimorar a avaliação das soluções obtidas, permanece o desafio por buscar referências. A primeira sugestão é realizar testes com dados reais e compará-los aos resultados obtidos manualmente pelas distribuidoras.

Outra sugestão é a formulação do problema como um problema de particionamento de conjuntos (“*set partitioning*”), que pode ser resolvido através da enumeração de todas as possíveis combinações de alocação para gerar colunas de entrada para um método de solução exata de programação de viagens. A enumeração de todas as combinações possíveis pode ser feita computacionalmente, porém o número de colunas também é exponencial em relação ao tamanho do problema, o que pode inviabilizar o método. Alternativamente pode-se pensar numa solução mais sofisticada baseada no método de geração de colunas, ou alguma heurística para selecionar colunas promissoras e eliminar combinações pouco atraentes.

Outra possibilidade de investigação é a introdução de um novo mecanismo de melhoria de resultados, que efetue ao final de cada iteração uma busca por trocas do tipo *2-opt* ou *3-opt*, possivelmente inserido num esquema de controle utilizando busca tabu. Ou ainda explorar uma vizinhança mais sofisticada, utilizando um método tal como busca VLSN. Entretanto, conforme discutido ao longo do trabalho, estima-se que poucas trocas sejam possíveis, visto que existem muitas restrições além da variação de volume dos pedidos.

4.6 PROGRAMAÇÃO DE VEÍCULOS DE COLETA DE RESÍDUOS PARA GERAÇÃO DE ENERGIA

Nesta seção trata-se do problema de programação da coleta de resíduos de biomassa de madeira visando a sua utilização como combustível para abastecimento de uma central produtora de energia. O mesmo foi objeto da pesquisa de mestrado de Galvão (2004), orientada por este autor.

O problema pode ser definido da seguinte forma: uma vez selecionado um conjunto de fornecedores de resíduos que minimize o custo global do fornecimento (aquisição mais transporte), busca-se determinar a programação de viagens diárias dos veículos ao longo de um horizonte de programação correspondente a vários dias de tal forma que a frota homogênea de veículos necessários para a realização das coletas seja mínima. Os veículos farão sempre viagens redondas (ida e volta) entre usina de geração de vapor e energia (central) e cada fornecedor, coletando carga correspondente à capacidade do veículo; em outras palavras, a cada viagem o veículo leva um contentor vazio e retorna com um contentor cheio de resíduos, à semelhança de outras situações de coleta de lixo ou de resíduos de construção civil em que são utilizadas caçambas. A cada fornecedor está associada uma frequência de coleta no período, em função da sua taxa de geração de resíduos.

Assim, mais especificamente, deseja-se determinar os dias de coleta em cada fornecedor, respeitando a sua frequência de coleta, bem como a programação de viagens diárias a serem realizadas por cada veículo, de tal forma a minimizar a frota total necessária ao longo do período, respeitando-se as restrições de duração da jornada diária de trabalho de cada veículo.

Embora existam inúmeros trabalhos na literatura relacionados à programação de veículos, em particular no contexto do transporte coletivo por ônibus (boas referências podem ser encontradas, por exemplo, em Bodin *et al.*, 1983; Carraresi e Gallo, 1984; Silva, 2001), nenhum trabalho encontrado na literatura trata da programação de veículos ao longo de um horizonte de tempo, buscando distribuir o esforço de coleta nos diferentes dias que compõem o período, de forma a reduzir a frota necessária. Weintraub *et al.* (1996) desenvolveram uma heurística para apoiar e otimizar a programação diária de veículos para coleta de madeira de

florestas, e transporte até pátios de seleção ou serrarias. A programação consiste em definir a próxima viagem de cada veículo que chega para descarregar madeira, de forma a atender à demanda de coleta e melhor distribuir as chegadas dos mesmos ao(s) ponto(s) de destino, diminuindo-se assim as filas para descarga, bem como as necessidades de equipamentos para essas operações. Van Vliet *et al.* (1992) trataram do problema de distribuição de açúcar a granel utilizando veículos em viagens com carga completa. Ao contrário do presente problema, havia restrições quanto aos horários de entrega; adicionalmente, a frota era heterogênea e os veículos podiam carregar em diferentes unidades produtivas, havendo restrições de capacidade de carregamento. Equi *et al.* (1997) estudaram o problema de transporte e programação de veículos para entrega de um único produto, produzido em diversas plantas para entrega em diferentes destinos. Os veículos são carregados nas origens e realizam uma única entrega com carga completa em cada viagem; assim, uma ou mais viagens podem ser realizadas numa mesma jornada diária de trabalho.

Em síntese, nenhum dos trabalhos encontrados na literatura é semelhante ao problema de programação de coleta de resíduos de madeira para um período de tempo.

4.6.1 Formulação Matemática

Seja $i = 1, 2, 3, \dots, N$ o conjunto de fornecedores de resíduos de biomassa de madeira. Para cada fornecedor i é conhecido o tempo total de coleta TC_i (viagem redonda), composto dos tempos de percurso de ida e volta, de carga no fornecedor e de descarga na usina. Define-se ainda o intervalo nd_i , em dias, entre duas coletas consecutivas no fornecedor i , definida com base na taxa média de geração de resíduos de cada geradora i . O horizonte de programação é dado por T . Assume-se ainda que nd_i é divisor de T para todo $i = 1, 2, 3, \dots, N$.

A fim de representar a possibilidade de que seja necessária mais de uma coleta por dia em um dado fornecedor, define-se ainda o número de coletas diárias nv_i em cada fornecedor i , com a condição que $nv_i > 1$ se e somente se $nd_i = 1$ (fornecedor que requer coletas em todos os dias do horizonte de programação T), e $nv_i = 1$ para $nd_i > 1$.

Seja $v = 1, 2, 3, \dots, V$ a frota de veículos homogêneos que podem realizar as coletas de resíduos. Note-se que V corresponde a um limitante superior do número de veículos potencialmente necessários para o atendimento da demanda que se busca minimizar (por

exemplo, $V=N$, ou seja, cada veículo faz apenas uma coleta em cada dia). A duração máxima da jornada diária de trabalho de cada veículo é dada por H .

O problema compreende as seguintes variáveis de decisão: (i) $y_v = 1$ se o veículo v é utilizado, 0 caso contrário; (ii) $x_{iv}^t =$ número de viagens do veículo v ao fornecedor i no dia t .

O problema de dimensionamento e programação da frota de coleta de resíduos de madeira pode ser formulado como:

$$[\min] \sum_{v=1}^V y_v \quad (4.54)$$

sujeito a:

$$\sum_{i=1}^N TC_i x_{iv}^t \leq H y_v \quad t = 1,2,3 \dots T \quad v = 1,2,3 \dots V \quad (4.55)$$

$$\sum_{v=1}^V \sum_{t=t_1}^{t_1+nd_i-1} x_{iv}^t = n v_i \quad i = 1,2,3 \dots N \quad t_1 = 1,2,3 \dots, (T - nd_i + 1) \quad (4.56)$$

$$y_v \in \{0,1\} \quad v = 1,2,3 \dots V \quad (4.57)$$

$$x_{iv}^t \in \mathbb{N}^+ \quad i = 1,2,3 \dots N \quad t = 1,2,3 \dots T \quad v = 1,2,3 \dots V \quad (4.58)$$

A função objetivo (4.54) corresponde à minimização da frota necessária. A restrição (4.55) assegura que o tempo total de viagens a serem feitas pelo veículo v no dia t não supere a duração máxima da jornada de trabalho H . A restrição (4.56) garante que sejam realizadas, para cada fornecedor i , as $n v_i$ viagens diárias necessárias e com a periodicidade nd_i . Por fim, as restrições (4.57) e (4.58) estão relacionadas aos domínios das variáveis de decisão.

O modelo matemático apresentado acima foi implementado em ambiente de planilha eletrônica Excel o utilizando o *XPRESS Solver Engine Version 5.5* da Frontline Systems Inc., destinado a aplicações de programação linear inteira mista e que permite tratar até 200 mil variáveis. Os tempos de processamento resultaram muito elevados, superiores a 36 horas, em um computador Pentium IV 1,7 GHz e com 768 Mb RAM.

Uma outra alternativa de modelagem matemática, além daquela considerada por Galvão (2004), pode ser formulada para o problema, mais genérica em termos de dias de coleta (não necessariamente igualmente espaçados em termos de intervalo entre coletas consecutivas).

Sejam $k = 1, 2, \dots, K_i$ as possíveis combinações de dias de coletas para o fornecedor i . Para cada combinação k define-se a_{ik}^t como sendo o número de coletas realizadas no fornecedor i no dia t na combinação k . Exemplificando, o vetor $(1\ 0\ 1\ 0\ 1\ 0)$ indica coletas nos dias 1, 3 e 5 para um dado fornecedor considerando-se um período de 6 dias; já o vetor $(2\ 1\ 2\ 1\ 2\ 0)$ indica duas coletas diárias nos dias 1, 3 e 5, uma coleta diária nos dias 2 e 4 e nenhuma coleta no dia 6 do período de planejamento.

Define-se a seguinte variável de decisão adicional: $u_{ik} = 1$ se a combinação k é selecionada para o fornecedor i , 0 caso contrário.

O novo modelo matemático pode ser escrito da seguinte forma:

$$[\min] \quad \sum_{v=1}^V y_v \quad (4.59)$$

sujeito a:

$$\sum_{k=1}^{K_i} u_{ik} = 1 \quad i = 1, 2, 3, \dots, N \quad (4.60)$$

$$\sum_{v=1}^V x_{iv}^t = a_{ik}^t u_{ik}^t \quad i = 1, 2, 3, \dots, N \quad k = 1, 2, 3, \dots, K_i \quad t = 1, 2, 3, \dots, T \quad (4.61)$$

$$\sum_{i=1}^N TC_i x_{iv}^t \leq Hy_v \quad t = 1, 2, 3, \dots, T \quad v = 1, 2, 3, \dots, V \quad (4.62)$$

$$y_v \in \{0,1\} \quad v = 1, 2, 3, \dots, V \quad (4.63)$$

$$x_{iv}^t \in \mathbb{N}^+ \quad i = 1, 2, 3, \dots, N \quad t = 1, 2, 3, \dots, T \quad v = 1, 2, 3, \dots, V \quad (4.64)$$

$$u_{ik} \in \{0,1\} \quad i = 1, 2, 3, \dots, N \quad k = 1, 2, 3, \dots, K_i \quad (4.65)$$

A função objetivo (4.59) busca a minimização da frota. A restrição (4.60) impõe que apenas uma combinação k de dias de coleta seja selecionada para cada fornecedor i . Já a restrição (4.61) assegura que sejam realizadas as viagens de coleta em cada dia de coleta t em cada fornecedor i de acordo com o número de coletas requeridas na combinação k selecionada. Já a restrição (4.62) o tempo total de viagens a serem feitas pelo veículo v no dia t não supere a duração máxima da jornada H de cada veículo. As restrições (4.63)-(4.65) estão relacionadas aos domínios das variáveis de decisão. Note-se que o modelo permite considerar, para cada fornecedor, um número de coletas variável em cada um dos dias do período de planejamento T , bem como em termo dos dias de coleta admissíveis e da sua periodicidade.

4.6.2 Estratégia de Solução

Propôs-se uma estratégia de solução híbrida, que decompõe o problema em dois, mantendo o ambiente de planilha eletrônica para a solução do problema:

- heurística baseada em Algoritmos Genéticos para a determinação dos melhores dias de coleta para cada um dos fornecedores que requer periodicidade de visita (nd_i) superior a 1, buscando minimizar a máxima frota necessária no horizonte de programação T ;
- modelo exato de programação linear inteira, utilizando pacote de otimização, para a programação da frota para cada um dos dias do período T , uma vez definidos os dias de coleta para cada fornecedor i .

Um aspecto importante e inovador da utilização do Algoritmo Genético para este problema é a representação cromossômica idealizada. O gene do modelo cromossômico é composto por um único alelo, caracterizado, para cada dia fornecedor i , pelo último dia $dmax_i$ em que pode ocorrer a primeira coleta no período T , considerando o intervalo nd_i entre dois dias de coleta consecutivas. Por exemplo, se $T=6$ e $nd_i=3$, então existem três possíveis alternativas de dias de coleta: (1,4), (2,5) e (3,6); nesse caso $dmax_i=3$, pois a primeira coleta pode ocorrer no máximo no dia 3, a fim de assegurar o intervalo nd_i de três dias entre coletas consecutivas. Assim, uma vez definido o dia da primeira coleta para um fornecedor i , ficam automaticamente determinados os demais dias de coleta no período T . Assim, um cromossomo qualquer é formado por um conjunto de valores que definem os primeiros dias de coleta para cada um dos fornecedores i no respectivo intervalo válido $[1, dmax_i]$. Com base nisso, é possível determinar, para um dado cromossomo, w_i^t que assume o valor 1 para cada dia t em que é efetuada uma coleta no fornecedor i e zero nos demais dias.

A função de aptidão, que define a qualidade de um indivíduo da população é calculada da seguinte forma:

$$FA = \max_{1 \leq t \leq T} \sum_{i=1}^N TC_i w_i^t \quad (4.66)$$

Essa função (4.66) avalia o tempo total de operação para cada um dos dias t e determina o maior valor no período T . O tempo total de operação em cada um dos dias t é determinado como a somatória dos tempos de ciclo TC_i dos fornecedores que são coletados no dia t , dado

por w_i^t . Dessa forma, o maior tempo de operação no período T é usado como uma estimativa da frota necessária para a operação de coleta. Em outras palavras, a função de aptidão FA busca o dia em que a soma dos tempos de ciclo veiculares é máximo no período T , como uma medida da frota necessária para a operação. É esse valor que se deseja minimizar, uma vez que representa a minimização da frota de veículos. A Figura 4.8 ilustra esse cálculo para quatro fornecedores com periodicidades diferentes. Na alternativa 1, o dia mais carregado é o dia 2, pois a soma dos tempos de ciclo dos fornecedores coletados nesse dia é 11 horas. Com a mudança nos dias de coleta para os fornecedores 2, 3 e 4 na Alternativa 2, o dia mais carregado passa a ser o dia 5, com uma redução do tempo de viagem total para 8 horas

Alternativa 1			dia (t)											
Fornecedor	Tempo de ciclo (horas)	Intervalo ndi	1	2	3	4	5	6	7	8	9	10	11	12
1	4	2	4		4		4		4		4		4	
2	3	3		3			3			3			3	
3	2	4		2				2				2		
4	6	6		6						6				
Tempo total			4	11	4	0	7	2	4	9	4	2	7	0

Alternativa 2			dia (t)											
Fornecedor	Tempo de ciclo (horas)	Intervalo ndi	1	2	3	4	5	6	7	8	9	10	11	12
1	4	2		4		4		4		4		4		4
2	3	3			3			3			3			3
3	2	4	2				2				2			
4	6	6					6						6	
Tempo total			2	4	3	4	8	7	0	4	5	4	6	7

Figura 4.8: Exemplo de cálculo do valor da aptidão

A Figura 4.9 ilustra como está estruturado o cromossomo de um indivíduo em planilha Excel, incluindo os dados de entrada (periodicidade nd_i , tempo de ciclo TC_i), bem como o cálculo do valor da aptidão respectivo (=24,50h no dia 10).

A população inicial é gerada aleatoriamente. Utiliza-se o ranqueamento, isto é, a classificação dos indivíduos com base na ordem não crescente das aptidões, para selecionar os pares para cruzamento, considerando um único ponto de cruzamento selecionado aleatoriamente. Os indivíduos gerados são posteriormente submetidos à mutação. O procedimento é reinicializado após um certo número de iterações sem melhoria da solução, preservando-se um conjunto elite de indivíduos da última geração. O algoritmo é interrompido quando um

número máximo de iterações for atingido. A heurística foi implementada em planilha eletrônica utilizando a linguagem *Visual Basic for Applications*. Maiores detalhes sobre a heurística podem ser encontrados em Galvão (2004).

Fornecedor	i	10	61	120	127	56	69	36	28	1	119	108
Tempo de Ciclo	TC_i	4,9	5,3	2,1	5,3	3,7	6,1	4,1	4,1	3,7	3,7	3,7
Intervalo de atendimento à geradora	nd_i	2	2	3	3	4	4	6	6	6	12	12
Número de viagens possíveis dentro do horizonte de planejamento T	NVP_i	6	6	4	4	3	3	2	2	2	1	1
Dia máximo possível para início do ciclo	$dmax_i$	2	2	3	3	4	4	6	6	6	12	12

	Aptidão	Dia de Início - din_i										
Valor da Aptidão / Cromossomo	24,50	1	2	3	1	2	2	5	4	5	1	12

Dia	Aptidão	Binário										
1	13,90	1	0	0	1	0	0	0	0	0	1	0
2	15,10	0	1	0	0	1	1	0	0	0	0	0
3	7,00	1	0	1	0	0	0	0	0	0	0	0
4	14,70	0	1	0	1	0	0	0	1	0	0	0
5	12,70	1	0	0	0	0	0	1	0	1	0	0
6	17,20	0	1	1	0	1	1	0	0	0	0	0
7	10,20	1	0	0	1	0	0	0	0	0	0	0
8	5,30	0	1	0	0	0	0	0	0	0	0	0
9	7,00	1	0	1	0	0	0	0	0	0	0	0
10	24,50	0	1	0	1	1	1	0	1	0	0	0
11	12,70	1	0	0	0	0	0	1	0	1	0	0
12	11,10	0	1	1	0	0	0	0	0	0	0	1

Figura 4.9: Estruturação do cromossomo e cálculo do valor da aptidão no Excel

Fonte: Galvão (2004)

Uma vez determinados os melhores dias de coleta para cada fornecedor, é necessário definir a programação dos veículos para cada um dos dias do período T . Busca-se determinar a alocação das coletas a serem realizadas por cada veículo em cada um dos dias individualmente, buscando minimizar a frota total necessária nesse dia.

Para tanto, é formulado um modelo de programação linear inteira mista, que corresponde a uma simplificação do modelo (4.54-4.58) para cada um dos dias individualmente, buscando minimizar a frota necessária naquele dia.

Definem-se as seguintes variáveis de decisão: (i) $y_v = 1$ se o veículo v é utilizado, 0 caso contrário; (ii) x_{iv} = número de viagens do veículo v ao fornecedor i .

O problema de programação da frota em cada dia do período T pode ser definido como:

$$[\min] \sum_{v=1}^V y_v \quad (4.67)$$

sujeito a:

$$\sum_{i=1}^N TC_i x_{iv} \leq H \cdot y_v \quad v = 1, 2, 3 \dots V \quad (4.68)$$

$$\sum_{v=1}^V x_{iv} = n v_i \quad i = 1, 2, 3 \dots N \quad (4.69)$$

$$y_v \in \{0, 1\} \quad v = 1, 2, 3 \dots V \quad (4.70)$$

$$x_{iv} \in \mathbb{N}^+ \quad i = 1, 2, 3 \dots N \quad v = 1, 2, 3 \dots V \quad (4.71)$$

A função objetivo (4.67) busca minimizar a frota necessária naquele dia. A restrição (4.68) assegura que a duração da jornada H seja respeitada pelos veículos efetivamente utilizados, enquanto a restrição (4.69) assegura que sejam realizadas as coletas necessárias.

Pode-se fazer uma analogia desse problema com o problema conhecido na literatura por *bin-packing*, que poderia ser genericamente enunciado como: dado um conjunto de *bins* (mochilas) com capacidade W e m itens com tamanhos v , o problema consiste em inserir todos os itens nas mochilas, buscando minimizar o número total de mochilas necessárias, sem exceder a capacidade de cada uma delas (tempo de ciclo total). Segundo Martello e Toth (1990), o problema *bin-packing*, apesar de sua formulação simples e compacta, é classificado como NP-difícil. Por outro lado, a decomposição do problema resultou em sub-problemas de porte significativamente menores para cada um dos dias, que puderam ser facilmente resolvidos utilizando um pacote de otimização comercial.

A estratégia de solução proposta foi totalmente implementada em planilha eletrônica e validada através da solução de um problema de menor porte com 33 fornecedores e período T de 6 dias, que também foi resolvido através do modelo de otimização (5.1-5.5) com 2189 variáveis de decisão. Para esse problema, a heurística conseguiu determinar a solução ótima do problema.

A heurística foi também aplicada com sucesso a um problema real envolvendo 62 geradoras e um horizonte de programação de 12 dias, resultando 17.135 variáveis de decisão inteiras. O problema foi resolvido em pouco menos de 30 minutos de tempo de processamento. Maiores detalhes podem ser encontrados em Galvão (2004).

4.6.3 Considerações Finais

A estratégia de decomposição proposta apresentou características inovadoras ao implementar uma heurística baseada em Algoritmos Genéticos em ambiente de planilha eletrônica, permitindo solucionar de maneira satisfatória um problema real de coleta de resíduos de madeira, mesmo considerando que a linguagem utilizada não proporciona bom desempenho em termos de velocidade de processamento.

Possíveis extensões desse trabalho podem englobar: (i) testes com o novo modelo matemático proposto (4.59)-(4.65); (ii) um aprimoramento dos algoritmos propostos, em particular do Algoritmo Genético, através, por exemplo, da eliminação dos indivíduos duplicados, a fim de assegurar a diversidade das populações ao longo das gerações; (iii) a sua implementação em linguagem C++, utilizando a biblioteca *Concert Technology* do ILOG/CPLEX para a solução dos problemas de otimização, permitindo dessa forma resolver problemas de maior porte; (iv) a generalização da estratégia de solução proposta, a fim de considerar, por exemplo, o caso mais geral de uma frota heterogênea, entre outros.

4.7 ASPECTOS PRÁTICOS DA APLICAÇÃO DE MODELOS DE ROTEIRIZAÇÃO DE VEÍCULOS A PROBLEMAS REAIS

Nas seções anteriores, apresentaram-se diferentes problemas de roteirização, com ênfase para as suas formulações matemáticas e seus algoritmos de solução. Eles retratam a importância da roteirização de veículos, bem como a diversidade de problemas encontrados no mundo real e a sua complexidade computacional, desafiando os pesquisadores no sentido de se buscarem novos algoritmos que permitam considerar problemas mais complexos de maneira realista, obter melhores resultados e resolver problemas de dimensões crescentes.

Por outro lado, conforme aponta Cunha (2000), pouco se tem discutido na literatura sobre os requisitos e condicionantes a serem considerados na escolha de *softwares* de roteirização disponíveis no mercado. Tal desconhecimento leva a resultados muitas vezes insatisfatórios, mesmo com a utilização de sofisticados pacotes de roteirização. A complexidade computacional dessa imensa gama de problemas classificados genericamente como de roteirização de veículos obriga a adoção de estratégias de solução heurísticas, que nem

sempre apresentam bom desempenho quando as restrições e condicionantes do problema diferem daquelas consideradas na sua concepção.

Tem-se observado em diversas aplicações, principalmente no caso brasileiro, que, embora a seleção e a implantação de softwares de roteirização tenha sido feita com cuidado, os benefícios obtidos com a sua utilização não raro resultam aquém das expectativas iniciais, mesmo em se tratando de “*softwares*” consagrados no mercado.

Nem sempre isso pode ser creditado exclusivamente à fragilidade dos algoritmos de solução incorporados nos softwares, na maioria das vezes extensivamente testados e validados, com inúmeras histórias de sucesso nos seus países de origem, mas principalmente a condicionantes locais e particularidades dos problemas que não podem ser considerados, assim como da baixa qualidade dos dados de entrada que alimentam os modelos.

Devem-se destacar ainda dificuldades na etapa de escolha do produto. O fato da maioria desses produtos serem verdadeiras caixas pretas em termos dos seus algoritmos de solução, conforme apontado por Hall e Partyka (1997) e Hall (2004), e o pouco conhecimento técnico especializado por parte dos representantes locais, acabam levando a escolhas que posteriormente se mostram equivocadas, uma vez que os softwares nem sempre conseguem atender às necessidades para os quais foram adquiridos. Conforme se procurou demonstrar nas seções anteriores, existem inúmeros diferentes problemas de roteirização de veículos, cada qual com características e condicionantes distintos, que requer soluções heurísticas customizadas.

Assim, pretende-se, a seguir, apresentar e discutir alguns aspectos e condicionantes que afetam o uso de roteirizadores, com ênfase para peculiaridades locais, que nem sempre estão ou podem ser incorporadas aos pacotes existentes no mercado, incluindo também as dificuldades na obtenção, na manutenção e na atualização dos dados de entrada para os modelos, que acabam interferindo na qualidade dos resultados obtidos.

4.7.1 A Função Objetivo a ser Otimizada

A definição dos roteiros em que é mais vantajoso o uso de frota própria ou é melhor utilizar serviços de terceiros, de modo a otimizar o custo total (da utilização da frota própria e do total de frete pago a terceiros) é um aspecto em que as especificidades da realidade brasileira na

contratação de terceiros, particularmente nas operações de coleta e distribuição urbanas, não conseguem ser representadas nos softwares de roteirização disponíveis no mercado.

Isso ocorre porque, no Brasil, os fretes pagos nem sempre são calculados com base na distância percorrida pelos veículos. Por exemplo, os contratos com terceiros para serviços de distribuição e coleta urbana, a prática mais comum é o pagamento segundo a quantidade total transportada pelo veículo (medida em peso, volume, número de caixas, etc.) e segundo o número de paradas do roteiro. Nas grandes cidades, usualmente a contratação pode prever ainda valores diferenciados de fretes unitários por região ou por área a ser atendida, de modo a considerar uma maior “dificuldade” espacial (loais mais distantes, na periferia, com sistema viário deficiente, com entregas mais dispersas, etc.). Há situações em que o frete de todas as entregas é calculado com base na entrega mais distante ou “difícil”.

Ocorre que nenhum software de roteirização disponível no mercado permite considerar esse tipo particular de estrutura de custo, desvinculada da distância efetivamente percorrida, o que pode levar a soluções em que uma medida aproximada da realidade, ou seja, custos com a distância percorrida e com a frota (própria) sejam minimizados. Tal resultado não necessariamente corresponde à solução de menor custo, em particular quando há terceiros realizando parte dos atendimentos.

Os pacotes comerciais também não consideram, na definição dos roteiros, o problema do arranjo da carga em cada veículo. Em entregas (e coletas) cujas cargas apresentem dimensões muito diversas (grandes e pequenos pesos e/ou volumes), como as encontradas, por exemplo, em entregas de lojas de departamento (por exemplo, geladeiras ao lado de batedeiras portáteis), o arranjo das cargas dentro do veículo pode ser decisivo para a otimização da distribuição. Em outras palavras, as seqüências de entrega que minimizam as distâncias podem ser inviáveis em termos do arranjo das cargas nos veículos, ou ainda a arranjos, por exemplo, em que cargas que estão na parte da frente de uma carroceria baú tenham que ser retiradas ou movimentadas para que outras cargas possam ser descarregadas, acarretando aumentos não previstos nos tempos de parada, podendo até inviabilizar o cumprimento da programação.

4.7.2 Atributos Espaciais para Problemas de Roteirização

Conforme visto anteriormente, para a modelagem de problemas de roteirização de veículos assume-se ser conhecido o grafo ou rede $G=(N, A)$, onde N é o conjunto de nós dos pontos a serem atendidos e a base onde se localizam os veículos, e A é o conjunto dos arcos representando as ligações entre os nós em N , para as quais são conhecidas as distâncias e os tempos de viagem.

Assim, o processamento de um algoritmo para um problema de roteirização deve ser precedido pela etapa de obtenção do grafo G . Isto envolve a localização geográfica ou espacial dos pontos de atendimento e a determinação das distâncias e dos tempos de viagem entre os mesmos. Este é um aspecto pouco discutido, mas de fundamental importância para a aplicação de modelos matemáticos a problemas reais de roteirização, uma vez que, em muitos casos, a forma como o grafo G é obtido e representado pode ser decisiva para a qualidade dos resultados obtidos e para a viabilidade de execução dos roteiros; às vezes tanto quanto a qualidade dos algoritmos de solução.

Alguns softwares adotam uma forma mais simplificada para a determinação do grafo G : os pontos de atendimento e a base onde se localizam os veículos são representados através de algum sistema de coordenadas, geralmente cartesianas ou georeferenciadas (latitude e longitude).

Nesse caso, as distâncias nos arcos são calculadas com base nas coordenadas dos pontos, segundo alguma métrica (distância Euclideana ou retangular), podendo ser ajustadas por fatores de correção, de forma a considerar o percurso adicional decorrente do sistema viário, conforme mostrado anteriormente na seção 4.3. Os tempos de viagem são calculados com base nas distâncias e em velocidades médias, que podem variar segundo o tipo de veículo, ou ainda segundo as zonas onde se localizam os pontos de origem e de destino e segundo a distância a ser percorrida.

Alguns softwares oferecem ainda o recurso adicional de cadastramento de barreiras geográficas, através de linhas ou polígonos, de modo a representar obstáculos naturais (tais como rios, montanhas, lagos, parques, etc.) ou artificiais (ferrovias, rodovias expressas, etc.) que não podem ser atravessados. Nesse caso, no cálculo das distâncias (em linha reta) é considerado o percurso adicional para contornar o obstáculo ou para a sua transposição através de pontos específicos (tais como pontes sobre rios).

Já outros softwares possuem interfaces com mapas digitais georeferenciados ou Sistemas de Informações Geográficas (SIG) para representar os pontos de atendimento e a malha viária por onde trafegam os veículos. Um SIG possibilita a localização automática de clientes e endereços. A distância e o tempo de viagem em cada um dos arcos do grafo G são obtidos através do processamento prévio de algum algoritmo de caminhos mínimos, aplicado à malha viária da região de interesse. Assim, um software de roteirização não opera diretamente sobre o banco de dados da malha viária, a qual pode conter até centenas de milhares de trechos de vias (muitas vezes quadras) cadastrados.

Mais informações sobre a utilização de SIG's associados a modelos de Pesquisa Operacional, incluindo aplicações a problemas de roteirização de veículos, podem ser encontradas em Koch (1999).

A decisão quanto à melhor forma de obtenção de um grafo, a partir de uma representação simplificada através de coordenadas ou de um banco de dados acoplado a um mapa digital depende de cada situação e também da natureza do ambiente de operação. No contexto espacial, pode-se dividir os problemas em duas categorias distintas: a dos roteiros urbanos e a dos interurbanos/regionais.

Na distribuição urbana, o problema é mais complexo, uma vez que o sistema viário é mais denso, o número de alternativas de roteiros é muito superior (é possível ir de algum ponto a qualquer outro) e as restrições de circulação são mais severas (restrições de circulação, mãos de direção, movimentos permitidos e proibidos, tais como conversões, retornos, etc.). Tal situação recomendaria uma representação mais realista da malha viária, através de mapas digitais.

Por outro lado, segundo Bodin (1990), criar e manter um SIG pode implicar um aumento significativo no custo de um sistema completo de roteirização e programação de veículos. A manutenção e a atualização de uma base de dados de informações viárias é particularmente crítica, principalmente em cidades maiores, nas quais há mudanças freqüentes de mão de direção e de restrições à circulação de veículos (conversões e outros movimentos proibidos).

Além de tudo isso, as bases geográficas armazenam um único valor de tempo ou velocidade em cada trecho da malha, que independe do horário do dia. Em regiões urbanas mais congestionadas, a consideração de velocidades médias ou de tempos de viagem que variam segundo o horário do dia, refletindo períodos de gargalo e congestionamento, são

fundamentais para o sucesso da roteirização, principalmente quando restrições temporais, como janelas de tempo, estão envolvidas.

Adicionalmente, no caso brasileiro, ainda não há mapas digitais da malha viária para a grande maioria dos municípios, com exceção de algumas capitais e cidades mais importantes. Mesmo onde disponíveis, tais bases nem sempre estão atualizadas, nem tampouco abrangem toda a malha viária, nem contêm informações de mãos de direção, de movimentos permitidos e proibidos e de velocidades e tempos de viagem suficientemente acurados.

Já na roteirização em nível regional, as distâncias entre pontos de atendimento (em geral, diferentes cidades) são geralmente mais longas, a malha muito menor em termos de extensão, assim como o número de trechos. Além disso, são menores as incertezas associadas às restrições e condicionantes de tráfego. Nesse contexto, não é difícil montar um cadastro rodoviário que contenha a matriz de distâncias e tempos de viagem entre cidades de interesse (onde se localizam clientes e/ou atendimentos), sem a necessidade de recorrer a coordenadas cartesianas ou à digitalização de mapas. Em geral esta matriz é suficiente para a obtenção de uma roteirização razoavelmente acurada, em termos de distâncias e seqüências de entregas.

Assim, fica nítido um “*trade-off*” a ser avaliado, caso a caso, entre a maior precisão e realismo proporcionados pela representação geográfica detalhada do grafo através de um SIG, ou mapa digital, que exige o processamento prévio dos caminhos mínimos para a montagem do grafo e o impacto que acarretam no desempenho computacional do sistema (maior tempo e recursos de processamento), em comparação com o cálculo simplificado de distâncias euclidianas (que podem considerar barreiras geográficas).

A questão da utilização de coordenadas cartesianas ou redes matemáticas para representar os pontos de atendimento depende primordialmente da necessidade de precisão quanto aos roteiros e programações de atendimentos e o esforço necessário para manter essas informações atualizadas.

Finalmente, um último aspecto a ser destacado refere-se aos problemas de localização dos pontos de atendimento, uma fonte de erros e imprecisões que afetam a qualidade dos roteiros obtidos. De nada adianta um algoritmo muito eficiente, se os pontos estão localizados incorretamente. Este problema é mais comum do que se imagina, e particularmente crítico em sistemas de atendimento em que os pontos mudam diariamente (como ocorre em serviços do tipo *courier*, serviços de entrega ou atendimento domiciliar, etc.). O recurso de localização

automática de endereços (conhecido como “*address matching*”) apresenta dificuldades no caso brasileiro, decorrentes tanto da imprecisão do cadastro (falta de CEP ou incorreto) quanto da existência de mais de um logradouro com o mesmo nome em regiões metropolitanas, ruas sem nome (rua A, B, etc.), de endereços que não constam do cadastro oficial (geralmente em áreas de periferia), entre outros.

Uma abordagem interessante baseada no CEP – Código de Endereçamento Postal - foi proposta por Fonseca (2002), na sua dissertação de mestrado, orientada pelo autor, que tratou do problema de programação dos veículos de coleta e entrega de encomendas expressas.

Há que se considerar ainda que no meio urbano, problemas com muitos atendimentos por veículo, geralmente em que os pontos estão muito próximos entre si, como por exemplo, na distribuição de bebidas, cigarros ou jornais, o detalhe mais microscópico da malha viária, das mãos de direção e dos movimentos permitidos e proibidos pode ser fundamental para que os roteiros programados possam ser efetivamente cumpridos na prática. Neste caso, até a própria localização geográfica dos pontos de entrega pode afetar os resultados. Alguns softwares localizam geograficamente os pontos de atendimento nos cruzamentos de ruas, outros no meio da quadra. Dependendo da proximidade entre os pontos e do peso/volume da carga a ser movimentada (como no caso de bebidas), uma localização deficiente pode inviabilizar o cumprimento da seqüência de entregas.

4.7.3 Requisitos de um Software para Roteirização de Veículos

Apresenta-se, a seguir, um conjunto de elementos para caracterização geral dos problemas de roteirização, que podem ser utilizados como uma lista de verificação (*check-list*) de apoio à especificação dos atributos e requisitos de um *software* a ser adquirido ou de um modelo de roteirização a ser desenvolvido; a mesma foi elaborada a partir do artigo de Assad (1988).

- Atendimentos (coletas; entregas; serviços de reparos, manutenção, instalação, etc.)
 - *Horários de atendimento* (janelas de tempo): uma ou mais de uma janela (por exemplo, 8-12h e 14-18h); rígida ou flexível (i.e., que pode ser violada mediante uma penalização);
 - *Tempos de atendimento*: fixo em cada ponto; variável com a quantidade a ser entregue/coletada; de acordo com o horário do dia;

- *Demandas*: conhecidas antecipadamente; vão surgindo ao longo do dia; conhecidas apenas no momento da entrega (pronta-entrega), como no caso, por exemplo, de distribuição de cigarros no varejo;
- *Tipo de Operação*: coletas, entregas ou ambos simultaneamente;
- *Precedência*: alguns pontos necessitam obrigatoriamente ser visitados antes de outros;
- *Tipos de veículo que podem atender*: tamanho máximo; necessidade de equipamento especial (e.g., plataforma elevatória) ou configuração para carga/descarga (por exemplo, *sider*, com movimentação só pela lateral);
- *Tipo de carga*: um ou mais de um (por exemplo, seca, térmica, refrigerada); com restrição ou incompatibilidade de transporte conjunto (por exemplo, alimentos e produtos de limpeza);
- *Tratamento de clientes*: com diferentes prioridades para atendimento (obrigatoriamente todos os clientes A e os B que for possível atender); clientes que não podem estar no mesmo roteiro (por exemplo, duas grandes redes de supermercado, como CBD e Carrefour);
- *Localização dos clientes*: fixos ou mudam todo dia (por exemplo, entrega de lojas de departamento);
- Frota
 - *Localização dos veículos*: uma ou múltiplas bases;
 - *Composição da frota*: homogênea ou heterogênea;
 - *Tamanho da frota*: fixo ou variável;
 - *Propriedade da frota*: própria ou de terceiros;
 - *Reaproveitamento dos veículos*: caso retorne cedo, um veículo pode ser reaproveitado para uma outra viagem;
 - *Duração das rotas*: distância e/ou duração máxima;
 - *Paradas para refeição e descanso*: em locais fixos;
 - *Horas-extra*: limite, análise se vale a pena;
 - *Distribuição homogênea da carga de trabalho entre os veículos*

- Rotas
 - *Fixas ou variáveis*: mudam todo dia?;
 - *Estáticas ou dinâmicas*, i.e, podem ser alteradas de modo a incluir novas solicitações que vão surgindo depois que os veículos saíam da base (como no caso do transporte de encomendas expressas por empresas como FedEx, UPS, etc.)
 - *Percurso urbano ou rodoviário*

O Quadro 4.3 sintetiza os principais condicionantes e requisitos desejáveis para um software comercial genérico para roteirização de veículos, segundo as visões de Assad (1988), Ronen (1988) e Bodin (1990). Essa relação de atributos pode ser utilizada como ponto de partida num processo de seleção, para definir uma lista de verificação dos condicionantes práticos que um software de roteirização deve poder lidar para uma dada aplicação prática.

Muitos dos pacotes disponíveis no mercado brasileiro são bastante sofisticados e poderosos em termos de recursos e de possibilidades de consideração de diversos tipos de restrições, e foram bem testados e validados em diferentes tipos de problemas. Por outro lado, sua implantação tem exigido, às vezes, investimentos e recursos significativos, além de tempo para a preparação de bases de dados e para treinamento para utilização, até que estejam em condições operacionais para a sua efetiva utilização no dia-a-dia das empresas.

Um aspecto importante a ser destacado é que, embora a maioria dos modelos se proponham a otimizar a roteirização, na prática nem sempre os algoritmos conseguem levar em consideração todas as parcelas dos custos de operação, que compreendem não só os custos variáveis com a distância percorrida, como também os custos fixos dos veículos e os custos horários da tripulação (incluindo a decisão de utilizar ou não horas extras da tripulação para reduzir a necessidade de frota e a quilometragem percorrida).

Em muitos casos, as heurísticas embutidas nos softwares produzem soluções que correspondem a algum tipo de sub-otimização, buscando prioritariamente minimizar a frota e, em seguida, a distância total percorrida. Isso decorre do fato de que as heurísticas clássicas em que se apóiam os softwares, tais como as heurísticas de economias (Clarke e Wright, 1964), de varredura (Wren e Holiday, 1972; Gillet e Miller, 1974) e outras do tipo agrupa-primeiro e roteiriza depois (Fisher e Jaikumar, 1981) baseiam-se em medidas de distâncias ou tempos de viagem e não consideram outras parcelas de custo.

Quadro 4.3: Requisitos e características de sistemas para roteirização de veículos

CARACTERÍSTICA	ASSAD (1988)	RONEN (1988)	BODIN (1990)
Uma ou múltiplas bases	Sim	Sim	Sim
Diferentes tipos de veículos	Sim	-	Sim
Coletas e entregas - coletas de retorno ("backhauls")	Sim	Sim	Sim
Janelas de tempo	Sim	Sim	Sim
Tempos de carga e descarga	Sim	-	-
Velocidades variáveis	Sim	-	-
Contratação de terceiros	Sim	Sim	-
Limite de peso e volume	Sim	Sim	-
Múltiplos compartimentos por veículo	-	Sim	-
Duração máxima do roteiro	Sim	Sim	Sim
Contabilização de horas extras	Sim	-	Sim
Horários de início e término de viagem	Sim	-	-
Roteiros com pernoite; troca de motoristas	Sim	Sim	-
Locais de parada fixos (e.g. almoço)	Sim	-	-
Restrições de tamanho de veículo e equipamentos para um cliente	Sim	-	Sim
Zonas de entregas e possibilidade de fracionamento de carga;	Sim	-	-
Barreiras físicas e restrições de circulação de veículos	Sim	Sim	-
Mais de um roteiro por veículo (quando veículo retorna cedo à base)	Sim	-	-

Um bom guia de referência sobre os principais softwares de roteirização disponíveis comercialmente e suas principais características pode ser encontrado em Hall (2004). O autor destaca que os avanços tecnológicos recentes são menores que os observados desde o primeiro ano em que a pesquisa foi realizada (Hall e Partyka, 1997). Os principais avanços compreendem softwares que funcionam em arquitetura de rede e internet, incluindo sistemas 3 camadas, a maior capacidade de integração com sistemas de posicionamento global GPS (*Global Positioning Systems*) para construir, alimentar e calibrar as bases de dados utilizadas nos modelos de roteirização. O autor observa que, embora os *softwares* sejam anunciados como de uso geral, os respectivos fornecedores especializam-se em determinados segmentos do mercado, oferecendo interfaces de entrada de dados e de resultados customizadas segundo os interesses de cada segmento.

4.8 CONCLUSÕES DO CAPÍTULO

Neste capítulo foram apresentados diversos problemas que envolvem a roteirização e programação de veículos, todos eles motivados por alguma aplicação encontrada na prática. Como foi visto anteriormente, a roteirização de veículos compreende não um problema, mas uma família de problemas, às vezes com características muito distintas entre si. Dada a natureza combinatória dos problemas de roteirização de veículos, o que impede o uso de pacotes comerciais de otimização para a sua solução, torna-se imperativo o desenvolvimento de estratégias de solução heurísticas, a fim de encontrar soluções para instâncias reais. Conforme se procurou mostrar ao longo dos diversos problemas descritos neste capítulo, objeto de pesquisas deste autor, dadas as características e condicionantes muito distintas desses problemas, os modelos matemáticos e as estratégias de solução bastante específicas para cada problema. Foram considerados problemas que envolvem o transporte de carga e de também passageiros, com frota homogênea e heterogênea, uma ou múltiplas garagens de onde partem os veículos, utilizando as principais estratégias de solução encontradas na literatura, desde heurísticas construtivas, passando por algoritmos genéticos, busca tabu, tabu granular, GRASP, entre outras.

Foram abordadas e discutidas restrições e características encontradas em problemas da realidade brasileira que não podem ser consideradas nos pacotes disponíveis no mercado, embora sejam fundamentais para a otimização. Entre elas, o sistema de cálculo de custos de fretes de terceiros e o problema de arranjo da carga no veículo constituem excelentes oportunidades de pesquisa de novas estratégias de solução mais adequadas à realidade brasileira.

Os problemas aqui apresentados não englobaram todas as pesquisas com as quais o autor se envolveu. Ficaram de fora a sua pesquisa de doutoramento, que tratou do problema de roteirização de veículos com janelas de tempo e frota heterogênea, sendo utilizada relaxação Lagrangeana (Cunha, 1997), que havia sido abordada anteriormente na seção 2.2.4; o problema de programação dos veículos de coleta e entrega de encomendas expressas (Fonseca, 2002); o problema de roteirização e programação de gerentes de banco (Cunha, 2003).

As pesquisas em andamento sob orientação deste autor englobam outros problemas, tais como o problema de roteirização periódica, em que se considera um período de planejamento

superior a um dia, e as decisões envolvem não só os roteiros para cada um dos dias do período, mas também os dias em que cada ponto será visitado, considerando que a cada ponto está associada uma frequência de atendimento (Wu e Cunha, 2005). Outro problema envolve ainda o problema de estoque-roteirização, em que se considera de maneira integrada integração as decisões de roteirização e de estocagem (Znamensky e Cunha, 2003).

Também serão considerados no futuro outros problemas que envolvem a programação de veículos, em uma extensão do trabalho de Galvão, além de problemas de programação de tripulações.

Capítulo**5****PROJETO DE
REDES LOGÍSTICAS**

O projeto (ou configuração) de redes logísticas, tanto de distribuição quanto de suprimento, ou até mesmo de transporte, é um dos problemas macro-logísticos mais relevantes enfrentados pelas empresas. Uma rede pode ser entendida como a representação físico-espacial e temporal das origens e dos destinos de cargas, produtos acabados e insumos, bem como dos seus fluxos e demais aspectos relevantes, desde o(s) ponto(s) de origem até o(s) ponto(s) de destinação final. Assim, genericamente uma rede logística pode ser composta por fornecedores, unidades produtivas, depósitos, centros de distribuição e pontos de demanda, como ilustrado na Figura 5.1, por onde podem fluir matérias primas, estoques em processo e produtos acabados. Pode envolver os agentes diretamente relacionados à empresa (fornecedores e clientes) ou toda uma cadeia, em diversos níveis, desde a origem da matéria prima mais básica até os clientes finais, geralmente passando por mais de um processo intermediário de produção ou manufatura, caracterizando, neste caso, uma cadeia de abastecimento (“*supply chain*”). Embora tenha se consagrado no Brasil o termo “cadeia de suprimentos” ao invés de cadeia de abastecimento, essa denominação não seja a mais adequada, por transmitir a idéia equivocada de cadeia restrita ao suprimento de matérias primas, e não ao abastecimento de insumos e produtos como um todo e de uma forma integrada ao longo de toda uma cadeia de empresas que se unem em um dado segmento do mercado.

A configuração da rede depende, necessariamente, das características operacionais de cada empresa, de seu negócio e envolve, entre outras, as seguintes decisões:

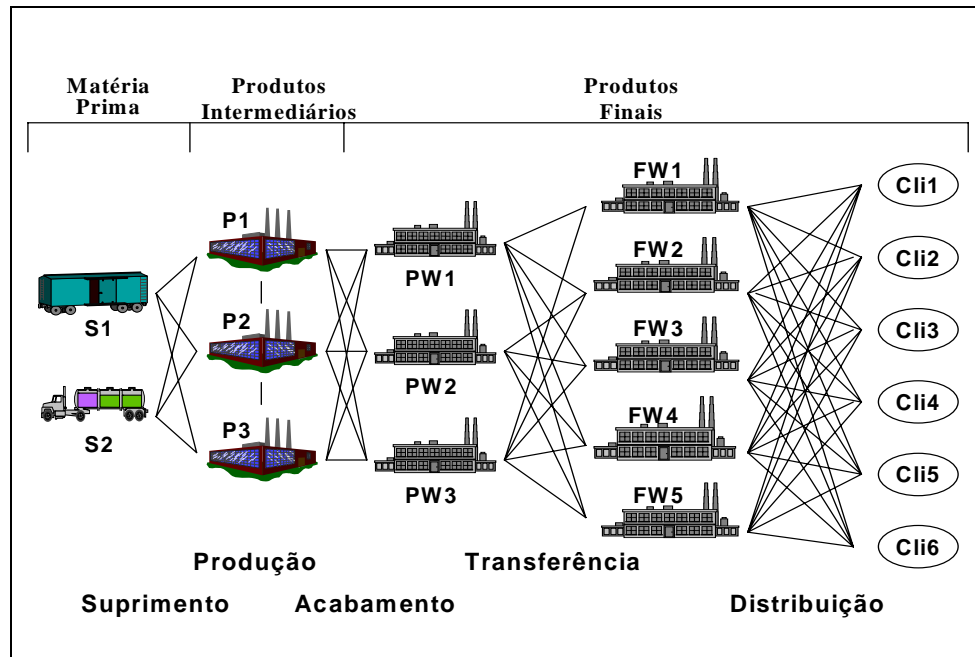


Figura 5.1: Exemplo de Rede Logística

- determinação do número de instalações;
- determinação da localização e do tamanho de cada instalação;
- determinação da alocação dos pontos de oferta e de demanda de cada instalação, isto é, dos clientes a serem atendidos por cada instalação e dos fornecedores que abastecem cada instalação;
- determinação dos fluxos entre as instalações logísticas e os pontos de oferta e demanda, podendo eventualmente incluir decisões quanto aos modais de transporte a serem utilizados.

As instalações passíveis de serem consideradas podem englobar unidades industriais de produção, armazéns, terminais e centros de distribuição, entre outras.

Em linhas gerais o objetivo básico é determinar a configuração que minimize os custos anuais do sistema, incluindo os custos de produção, de compras, de transporte, de manutenção de estoque e das instalações (custos fixos, de armazenagem e de movimentação), considerando restrições de nível de serviço.

Diversas são as situações enfrentadas pelas empresas que podem levar à necessidade de realização de um projeto de rede logística, entre as quais pode-se citar:

- empresa iniciando suas operações, necessitando definir a configuração de toda a sua rede logística;
- expansão de mercado, conquista de novos mercados, crescimento da demanda, novos requisitos de nível de serviço, redução de custos, tudo levando à necessidade de construir novas instalações e/ou ampliar as existentes;
- fusões ou aquisições de empresas, levando à necessidade de revisar toda a rede logística, definir pelo fechamento ou desativação de unidades existentes, assim como pela sua ampliação ou até mesmo construção de novas unidades;
- outras situações, tais como decorrentes da expansão e transformação urbana, tornando áreas outrora industriais periféricas em áreas nobres, com a conseqüente mudança no tipo de uso e intensidade de ocupação do solo, levando uma instalação existente a se tornar inadequada em termos da sua localização atual, prejudicando a sua operação, causando impactos indesejáveis, impedindo uma eventual ampliação, ou até mesmo levando a uma valorização imobiliária que recomende uma destinação mais nobre para a área.

Decisões de localização são consideradas estratégicas, uma vez que seus efeitos ocorrem no médio e longo prazo, além de usualmente implicarem investimentos vultosos. Uma nova instalação geralmente serve uma empresa por muitos anos, sendo o custo de desativação e/ou relocação bastante elevado. Por outro lado, uma rede logística otimizada permite a uma empresa operar com custos reduzidos e nível de serviço satisfatório; ao contrário, uma localização inadequada torna difícil para a empresa operar próxima da fronteira de eficiência e eficácia. Chopra e Meindl (2001) citam o exemplo da *Amazon*, empresa que comercializa livros, CD's de música e outros produtos através da internet, sem instalações físicas, que tinha dificuldades em cumprir prazos de entrega e assegurar baixos custos de distribuição operando a partir de um único armazém localizado em Seattle, nos EUA. Em decorrência, a empresa teve que abrir depósitos em outras partes do país a fim de assegurar uma distribuição eficiente em termos de custos (principalmente estoque e transporte) e eficaz em termos de nível de serviço (principalmente prazo de entrega).

Uma vez determinado o número e a localização das instalações, a decisão de quais clientes e fornecedores alocar a cada instalação tem um efeito significativo no desempenho da rede logística como um todo, uma vez que afeta custos de produção, de estoque e de transporte de modo a atender às demandas com o nível de serviço adequado. Em decorrência, essa alocação

deve ser revista periodicamente em função de mudanças e crescimento do mercado, bem como de novas exigências de nível de serviço.

O problema de configuração de redes tem dois aspectos importantes, um de natureza espacial e outro de natureza temporal. O aspecto espacial, ou geográfico, refere-se à localização dessas instalações e aos fluxos entre pontos, de forma a minimizar os custos relacionados ao atendimento da demanda. Já o aspecto temporal refere-se à disponibilidade de cargas, matérias primas e produtos no prazo adequado.

Este capítulo está organizado da seguinte forma: inicialmente são apresentados os conceitos básicos da teoria de localização de instalações e os principais modelos matemáticos, que formam a base para o projeto de redes logísticas. Em seguida, são detalhados dois problemas reais relacionados à configuração de redes: o problema de configuração de rede para uma transportadora de carga parcelada e o problema de configuração de uma rede de produção e distribuição para uma editora de revistas.

O primeiro deles envolve a configuração de redes do tipo "*hub-and-spoke*" para transportadoras de carga parcelada no Brasil. Mais especificamente, busca-se determinar quantos e quais são os terminais de consolidação de carga parcelada ("*hubs*") e quais pontos (ou filiais) são atendidos por cada "*hub*", de forma a minimizar a somatória dos custos fixos e variáveis. O modelo proposto se diferencia dos demais encontrados na literatura por permitir considerar fatores de economia de escala, para os custos de transporte, que variam de acordo com os respectivos volumes de carga entre instalações, como ocorre com as transportadoras de carga no Brasil.

Já o segundo problema trata da modelagem de redes para o caso de uma revista semanal de informações de grande tiragem. O problema consiste em determinar a quantidade e a localização de plantas industriais (*gráficas*), bem como a abrangência de cada uma delas e o seqüenciamento de produção, de forma a minimizar a soma dos custos de produção e de transporte. É proposto um modelo de programação linear inteira-mista para a sua solução, que considera aspectos temporais, de capacidade de produção das unidades industriais, de escolha do modal de transporte e de nível de serviço, que são essenciais para a competitividade e sobrevivência de editoras de revistas.

5.1 CONCEITOS BÁSICOS DE LOCALIZAÇÃO DE INSTALAÇÕES

Segundo Novaes (1989), o problema da localização de instalações tem sido tratado de modo bastante amplo na literatura, envolvendo desde problemas mais simples de localização de uma única instalação, até problemas bastante complexos, com várias instalações a serem localizadas, em diversos níveis de uma cadeia produtiva ou de serviços, com fluxos de naturezas distintas.

Podem englobar instalações produtivas tais como fábricas, portos, pontos de venda, armazéns, lojas de varejo e centros de serviço (Ballou, 1999), como também instalações de serviço urbano, incluindo serviços de rotina e de emergência, como postos de correio, pontos de incineração de lixo, instalações de bombeiros, postos de saúde e unidades de emergência médica, dentre outros (Larson e Odoni, 1981).

As decisões de localização, no contexto do planejamento logístico, correspondem à determinação do número, localização e tamanho das instalações a serem usadas (Ballou, 1999).

De acordo com Current, Daskin e Schilling (2002), os seres humanos têm analisado a eficácia das suas decisões de localização desde que habitaram a primeira caverna. De acordo com os autores, o longo e abrangente histórico da pesquisa sobre localização poder ser explicada pelos seguintes fatores:

- As decisões de localização geralmente são tomadas em todos os níveis das organizações humanas, desde indivíduos e famílias até empresas, agências governamentais e ainda agências internacionais.
- As decisões de localização são geralmente estratégicas por natureza, isto é, envolvem um elevado aporte de capital, e seus efeitos econômicos acarretam resultados no longo prazo. No setor privado, tais decisões têm uma influência maior na capacidade de uma empresa de competir no seu mercado. No setor público, elas influenciam na eficiência pelo qual instituições do poder público provêem serviços públicos e a habilidade destas instituições de atrair / agrupar famílias e outras atividades econômicas.

- As decisões de localização freqüentemente impõem exterioridades econômicas, tanto positivas como negativas. Tais exterioridades incluem desenvolvimento econômico, geração de empregos e de renda, poluição, congestionamento, entre outros.
- Modelos matemáticos de localização são, muitas vezes, extremamente complexos e difíceis de resolver.
- Modelos de localização correspondem a aplicações específicas, isto é, sua forma estrutural (a função objetivo, restrições e variáveis) é determinada de acordo com o problema particular em estudo. Conseqüentemente, não existe um único modelo geral de localização que seja apropriado para todas as aplicações potenciais ou existentes.

Current, Daskin e Schilling (2002) relacionam ainda 51 referências bibliográficas sobre o assunto, envolvendo inúmeras aplicações diferentes, destacando seu aspecto multidisciplinar.

Segundo Gualda (1995), no seu abrangente trabalho sobre a modelagem e a análise de terminais de transporte, o problema de localização pode ser definido como um problema de alocação espacial de recursos. A hipótese básica da teoria da localização é a de que cada empresa procura escolher a localização que leve à maximização dos lucros da sua atividade.

Silva Leme (1965) apud Gualda (1995) aponta que os fatores locais podem ser classificados em fatores aglomerativos, fatores desaglomerativos, e o fator transporte. Os fatores aglomerativos são os que contribuem para agrupar as atividades produtivas em um determinado ponto ou local, sendo que os desaglomerativos agem no sentido de desagrupar essas atividades, levando à localização das mesmas em mais de um ponto. O fator transporte pode agir tanto num sentido como no outro, dependendo do caso.

Nos problemas em que o fator transporte é predominante, isto é, tem grande peso nas decisões, a resolução de problemas de localização pode ser simplificada através da sua modelagem centrada no fator transporte, e as soluções assim obtidas analisadas com vistas aos demais fatores.

Ainda segundo Gualda (1995), os problemas de localização podem ser classificados em dois grupos, a saber:

- Métodos Indutivos, que se baseiam na análise de dados e informações estatísticos, históricos e provenientes de pesquisas de campo (questionários), através do que se

busca razões ou indicações quanto à melhor localização para uma dada indústria (ou terminal, no nosso caso);

- o Métodos Dedutivos, que consistem no estabelecimento de um modelo representativo da realidade, passível de tratamento matemático, para resolver o problema da localização; dados históricos ou estatísticos são usados para testar os resultados produzidos por esses modelos.

Há que se distinguir a macro-localização da micro-localização. A primeira precede a segunda e se refere à escolha de uma região para localização da instalação pretendida, enquanto que a segunda está associada à escolha de um sítio específico, dentro da região, para implantação da instalação.

Ainda segundo o autor, os problemas de macro-localização são mais adequados à aplicação de métodos do tipo dedutivo; os problemas de micro-localização envolvem um número muito grande de variáveis, e até mesmo fatores pessoais e políticos na tomada de decisões, o que limita, na maioria dos casos, à adoção de modelos matemáticos para a sua solução.

Os modelos matemáticos para solução do problema de localização baseiam-se na minimização dos custos envolvidos, admitindo-se, dessa forma, que não há variação significativa nas receitas associadas às possíveis soluções locacionais e que, portanto, a solução obtida satisfaz o objetivo de maximização de lucros. Essa é uma simplificação que pode não ser verdadeira em casos em que, por exemplo, a demanda pode apresentar variação importante com a localização escolhida. De qualquer maneira, o problema, mesmo com essa simplificação, tem natureza bastante complexa.

5.1.1 Modelos Matemáticos para a Configuração de Redes Logísticas

Segundo Ballou (2001), a teoria de localização de instalações é base para a modelagem matemática do problema de configuração de redes logísticas, também conhecido como projeto de cadeias de suprimentos. Do ponto de vista dos inúmeros artigos encontrados na literatura científica que tratam sobre o tema, esse problema se enquadra na categoria mais geral de problemas de localização de instalações (do inglês "*facility location problems*") ou problemas de localização de armazéns ("*warehouse location problems*"). Ou seja, definir a melhor localização de uma ou mais instalações, em termos espaciais ou geográficos.

De maneira geral, pode-se dizer que os modelos matemáticos para localização de instalações podem ser classificados em duas categorias: modelos para localização de uma única instalação e modelos para localização de múltiplas instalações (Magee, 1977; Ballou, 1985; Novaes, 1989; Gualda, 1995).

O problema de localização de uma única instalação pode ocorrer quando se pretende que haja apenas uma instalação, ou quando uma possível instalação estará tão isolada das demais que a demanda a ser atendida por ela pode ser considerada independente da demanda a ser atendida pelas demais instalações, possibilitando a decomposição do problema de localização de n instalações em n problemas independentes de localização de uma instalação, simplificando o problema.

A modelagem do problema de localização de mais de uma instalação é, certamente, mais complexa. Envolve, entre outras, considerações sobre a parcela da demanda a ser atendida por cada uma das instalações. Isso significa buscar respostas para questões relacionadas a quantas instalações implantar, onde implantá-las, o porte de cada uma delas e as respectivas áreas de influência, e os modos de transporte a serem utilizados para suprimento e distribuição. A função objetivo, nesse caso, está associada, em geral, à minimização da somatória dos custos associados a cada uma das instalações, sujeita a restrições quanto a porte mínimo ou máximo de cada instalação, distância entre elas, distâncias máximas de cada instalação aos pontos de demanda, além de limitações de capacidade de vias e dos veículos de transporte. A eventual existência de economias de escala aumenta a dificuldade na modelagem desse tipo de problema.

Os modelos locacionais podem diferir quanto à representação espacial da região considerada e da malha de transportes envolvida, podendo ser identificados dois tipos (Crainic, 1998):

- de localização no plano, o que implica a existência de infinitos pontos alternativos para a pretendida localização e a existência de uma malha de transportes bastante densa, cobrindo praticamente toda a região considerada. As distâncias, neste caso, são obtidas com base em métricas euclidianas ou retangulares;
- de localização discreta, isto é, num grafo ou rede, o que implica a representação matemática da malha de transportes existente e a consideração de um número finito de pontos alternativos para a pretendida localização, situados nos nós da rede. Modelos de programação linear, inteira e mista, assim como a teoria de grafos e de fluxos em

redes, formam a base dos modelos matemáticos utilizados neste segundo tipo de abordagem.

Problemas de localização envolvem a definição de algumas instalações, geralmente nos vértices de uma rede, de tal forma a facilitar a movimentação de produtos ou o fornecimento de serviços ao longo da rede. A demanda (“clientes”) também é geralmente representada nos vértices da rede. Estes podem estar nos mesmos (ou diferentes) vértices onde as instalações podem ser localizadas.

Brandeau e Chiu (1989) apresentam uma abrangente resenha contemplando 54 trabalhos sobre localização. O objetivo é localizar instalações (e, talvez, alocar clientes a pontos de prestação de serviços), de forma a otimizar um objetivo espacialmente dependente (implícita ou explicitamente dependente). Critérios típicos utilizados para tal incluem:

- minimização do tempo médio da viagem ou da distância média entre os pontos de oferta e de demanda, ponderados ou não pela demanda;
- minimização do tempo médio de resposta (tempo de viagem mais eventuais esperas para atendimento);
- minimização de uma função de custo da viagem ou do tempo de resposta;
- minimização do máximo tempo de viagem;
- maximização do mínimo tempo de viagem.

Apresentam-se a seguir as formulações matemáticas para alguns dos modelos matemáticos fundamentais mais representativos de localização de instalações. Para tanto, definem-se os seguintes conjuntos e dados de entrada:

I = conjunto de pontos de demanda (a serem atendidos), indexado por i ;

J = conjunto de locais candidatos a instalações, indexado por j ;

d_{ij} = distância do ponto de demanda i a partir do local candidato j ;

c_{ij} = custo de atendimento do ponto de demanda i a partir do local candidato j ;

q_i = quantidade demandada no ponto i ;

f_j = custo fixo da instalação j ;

b_j = capacidade da instalação j ;

Seja $G = (N, A)$ o grafo ou rede tal que N corresponde ao conjuntos dos nós ($N = I \cup J$) e A indica os arcos (i, j) entre nós dos conjuntos I e J , respectivamente.

Definem-se ainda as seguintes variáveis de decisão binárias: (i) $y_j = 1$ se a instalação j for selecionada; 0 caso contrário; (ii) $x_{ij} = 1$ se o ponto i é atendido a partir da instalação j ; zero caso contrário.

a) *Modelo de Localização de Cobertura de Conjuntos (Set Covering Location Model):*

Selecionar o menor número de instalações nos nós (ou vértices) do grafo G , tal que cada um dos pontos de demanda i seja atendido (coberto) por pelo menos uma instalação j^* selecionada, e assegurando que i esteja a uma distância máxima de cobertura D_c , isto é, $d_{ij^*} \leq D_c$. Assim, para cada nó i é definido um conjunto $N_i = \{j : d_{ij} \leq D_c\}$. A distância de cobertura D_c está geralmente relacionada ao caminho mais curto entre a instalação e cada um dos nós de demanda; pode ser a mesma para todos os vértices, ou pode depender de uma instalação específica e de pontos de demanda.

$$\text{Minimizar } \sum_{j \in J} y_j \quad (5.1)$$

sujeito a:

$$\sum_{j \in N_i} y_j \geq 1, \forall i \in I \quad (5.2)$$

$$y_j \in \{0,1\}, \forall j \in J \quad (5.3)$$

Na versão clássica, a função objetivo (5.1) corresponde à minimização do número de instalações necessárias para “cobrir” todos os pontos de demanda. Outras formulações encontradas na literatura podem considerar a minimização do custo total das instalações selecionadas (quando o custo de cada instalação puder variar), ou ainda considerar a situação de um orçamento fixo, em que então o objetivo pode ser maximizar a demanda coberta pelas instalações selecionadas. A restrição (5.2) assegura que todos os pontos de demanda estejam cobertos por no mínimo uma instalação selecionada.

Modelos de Cobertura de Conjunto são tipicamente associados à localização de instalações públicas, tais como centros de saúde, agências de correio, bibliotecas, e escolas.

b) *Problema dos p-centros:*

Localizar p instalações nos vértices de uma rede, de tal forma a minimizar D , que representa a distância máxima entre um ponto de demanda e à instalação ao qual o mesmo está alocado:

$$\text{Minimizar } D \quad (5.4)$$

sujeito a:

$$\sum_{j \in J} y_j = p, \quad (5.5)$$

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \quad (5.6)$$

$$x_{ij} - y_j \leq 0 \quad \forall i \in I, j \in J \quad (5.7)$$

$$D \geq \sum_{j \in J} d_{ij} x_{ij} \quad \forall i \in I \quad (5.8)$$

$$y_j \in \{0,1\} \quad \forall j \in J \quad (5.9)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in I, j \in J \quad (5.10)$$

Na formulação acima, a função objetivo (5.4) corresponde à minimização da distância máxima D . A restrição (5.5) impõe que p instalações devem ser escolhidas, enquanto a restrição (5.6) assegura que cada nó i esteja alocado a exatamente uma instalação escolhida. A restrição (5.7) restringe a alocação dos nós a instalações selecionadas, e a restrição (5.8) define o limitante inferior para o valor de D , que está sendo minimizado. As restrições (5.9) e (5.10) definem os respectivos domínios das variáveis de decisão.

Outras variações do problema de p -centro encontradas na literatura podem considerar as distâncias ponderadas pelas demandas dos nós, ao invés de tratar todos os nós igualmente como apresentado acima; também podem aceitar a localização das instalações em qualquer posição dos arcos, não se restringindo aos vértices. Problemas desse tipo muitas vezes surgem quando é necessário estabelecer a localização de instalações de emergência, tais como estações de bombeiros e resgate para emergências médicas.

c) *Problema das p-medias:*

Localizar p instalações nos vértices de uma rede e definir a alocação dos nós de demanda a essas instalações de tal forma a minimizar a distância total ponderada entre instalações e pontos de demanda. Esse problema pode ser formulado da seguinte maneira:

$$\text{Minimizar} \quad \sum_{i \in I} \sum_{j \in J} q_i d_{ij} x_{ij} \quad (5.11)$$

sujeito a:

$$\sum_{j \in J} y_j = p \quad (5.12)$$

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \quad (5.13)$$

$$x_{ij} - y_j \leq 0 \quad \forall i \in I, j \in J \quad (5.14)$$

$$y_j \in \{0,1\} \quad \forall j \in J \quad (5.15)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in I, j \in J \quad (5.16)$$

A função objetivo (5.11) busca minimizar a distância total ponderada pela distância. As restrições (5.12) a (5.14) são idênticas às restrições (5.5) a (5.7) do problema do p -centro; o mesmo ocorre com as restrições (5.15) e (5.16) em relação às restrições (5.9) e (5.10).

Assume-se, na formulação acima, que os locais candidatos se restringem aos nós da rede. Hakimi (1964) provou que mesmo relaxando o problema de modo a permitir a localização de instalações nos arcos da rede não permite a redução da distância total ponderada. Conseqüentemente, esta formulação permite obter a solução ótima mesmo se as instalações puderem se localizar em qualquer ponto dos arcos.

d) Problema de Localização de Instalações Capacitadas:

Algumas hipóteses do problema das p -medianas podem não ser apropriadas para algumas situações que envolvem a localização de instalações, em particular: (i) todas as instalações têm o mesmo custo fixo; (ii) as instalações não têm limite de capacidade; (iii) o número de instalações a serem selecionadas é definido a priori (igual a p). Assim, o problema de localização de instalações capacitadas pode ser visto como uma generalização do problema das p -medianas, de modo a considerar esses três aspectos, conforme pode ser visto na formulação matemática apresentada a seguir:

$$\text{Minimizar} \quad \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} q_i c_{ij} x_{ij} \quad (5.17)$$

sujeito a:

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \quad (5.18)$$

$$x_{ij} - y_j \leq 0 \quad \forall i \in I, j \in J \quad (5.19)$$

$$\sum_{i \in I} q_i x_{ij} \leq b_j y_j \quad \forall j \in J \quad (5.20)$$

$$y_j \in \{0,1\} \quad \forall j \in J \quad (5.21)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in I, j \in J \quad (5.22)$$

A função objetivo (5.17) corresponde à minimização das parcelas de custos fixos das instalações selecionadas e dos custos de transporte da demanda a ser atendida. A restrição (5.20) impede que a demanda total dos pontos alocados a uma instalação j supere a sua capacidade b_j . As demais restrições são as mesmas dos problemas anteriores de p -centros e p -medianas. Relaxando-se a restrição (5.22) permite-se que a demanda de um nó i possa ser atendida (parcialmente) por múltiplas instalações. Note-se que a restrição (5.19) poderia ser eliminada, uma vez que a restrição (5.20) assegura que os nós sejam alocados a instalações selecionadas; no entanto, a manutenção da mesma na formulação fortalece a relaxação linear do modelo. Por outro lado, sem a restrição (5.20) tem-se o problema de localização de instalações não capacitado.

e) *Problema de Localização com mais de uma camada*

Em todos os problemas anteriormente descritos havia apenas uma camada (do inglês “*layer*”) ou nível hierárquico, ou seja, pontos de demanda atendidos por instalações a serem selecionadas. Conforme visto anteriormente, podem existir situações em que é necessário determinar simultaneamente a quantidade e a localização de mais de um tipo de instalação (por exemplo, fábricas e depósitos), assim como a alocação dos depósitos às fábricas e também dos clientes aos depósitos, caracterizando diferentes camadas. Nesse caso, a formulação do problema de localização de instalações capacitado pode ser generalizada a fim de representar esse tipo de situação mais geral, conforme detalhado mais adiante. Para tanto, define-se o seguinte conjunto adicional:

K = conjunto de locais candidatos a instalações que abastecem as instalações $j \in J$, indexado por k ;

Em outras palavras, as instalações k podem representar, por exemplo, fábricas que abastecem depósitos j que, por sua vez, atendem clientes i . Definem-se ainda as seguintes variáveis de decisão: (i) $z_k = 1$ se a instalação k for selecionada; 0 caso contrário; (ii) w_{jk} que corresponde ao fluxo total para o depósito j a partir da fábrica k . Assim, o modelo matemático correspondente pode ser escrito como:

$$\text{Minimizar} \quad \sum_{k \in K} f_k z_k + \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{jk} w_{jk} + \sum_{i \in I} \sum_{j \in J} q_i c_{ij} x_{ij} \quad (5.23)$$

sujeito a:

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \quad (5.24)$$

$$x_{ij} - y_j \leq 0 \quad \forall i \in I, j \in J \quad (5.25)$$

$$\sum_{i \in I} q_i x_{ij} \leq b_j y_j \quad \forall j \in J \quad (5.26)$$

$$\sum_{j \in J} w_{jk} \leq b_k y_k \quad \forall k \in K \quad (5.27)$$

$$\sum_{k \in K} w_{jk} = \sum_{i \in I} q_i x_{ij} \quad \forall j \in J \quad (5.28)$$

$$y_j \in \{0,1\} \quad \forall j \in J \quad (5.29)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in I, j \in J \quad (5.30)$$

$$z_k \in \{0,1\} \quad \forall k \in K \quad (5.31)$$

A função objetivo (5.23) corresponde à minimização das parcelas de custo fixos dos dois tipos de instalações selecionadas (fábricas e depósitos), dos custos de transporte entre fábrica e depósito e também para atendimento dos clientes a partir dos depósitos, sendo que alguma das parcelas de custo de transporte pode também embutir algum custo unitário relativo às operações de manuseio e armazenagem no depósito. As restrições (5.24) e (5.25) são as mesmas que aparecem nos problemas anteriores. De maneira similar ao problema anterior, as restrições (5.26) e (5.27) impedem que a demanda total alocada a cada instalação supere a sua respectiva capacidade. Já a restrição (5.28) garante o balanço dos fluxos de entrada e de saída em cada depósito intermediário. As demais restrições são as equivalentes ao do problema anterior de localização de instalações capacitado.

Tal formulação pode ser estendida sem maiores dificuldades para considerar outras situações encontradas na prática, tais como uma rede logística com mais camadas, atendimento direto de fábricas para clientes (basta considerar um depósito junto à fábrica), ou ainda mais de uma categoria ou classe de produto. Entretanto, quanto mais complexo o modelo, maior o número de variáveis de decisão e maior a dificuldade para resolver instâncias reais através de pacotes comerciais de otimização, mesmo os mais avançados, uma vez que o problema de localização de instalações capacitado apresenta ordem de complexidade exponencial (Garey e Johnson, 1979).

Existem diversos outros modelos matemáticos relacionados a problemas mais específicos que envolvem a localização de instalações, não necessariamente voltados especificamente à configuração de redes logísticas. Boas referências gerais sobre a modelagem na localização de instalações podem ser encontradas nos trabalhos de Brandeau e Chiu (1989), Owen e Daskin (1998), Current *et al.* (2002), ReVelle e Eiselt (2005) e Ambrosino e Scutellà (2005).

5.1.2 Estratégias de Solução para Problemas de Localização

Conforme visto anteriormente, problemas de localização de instalações são geralmente formulados como modelos de programação linear inteira mista. Modelos de problemas reais podem facilmente ter dezenas e até centenas de milhares de variáveis de decisão e restrições, o que torna impossível sua resolução através de pacotes comerciais. A razão básica é que a maioria dos modelos de localização pertencem à categoria *NP-hard* (Garey e Johnson, 1979).

Conseqüentemente, é necessário utilizar heurísticas para a sua resolução. Muitas heurísticas baseiam-se em algoritmos gulosos (*greedy*) a fim de rapidamente se obter uma solução inicial viável, complementadas por um algoritmo de melhoria baseado em trocas ou em um processo de busca em vizinhança. Nessa linha, Ahuja *et al.* (2004b) propuseram uma heurística inovadora baseada em busca de vizinhança de grande porte que considera múltiplas trocas envolvendo vários clientes simultaneamente. Resende e Werneck (2004) desenvolveram uma meta-heurística híbrida que utiliza conceitos de algoritmos genéticos e busca tabu para o problema das *p*-medianas; mais especificamente, múltiplas soluções são geradas (*multistart*) e melhoradas através de *path-relinking*. A mesma foi posteriormente adaptada para resolver o problema de localização de instalações não capacitado (Resende e Werneck, 2005). Outra classe de heurísticas baseia-se no uso da relaxação Lagrangeana, que não só fornece limitantes inferiores ("*lower bounds*"), como também permite obter soluções viáveis (Barceló e Casanova, 1984; Pirkul, 1987; Hindi e Pienkosz, 1999).

Para finalizar, um aspecto importante, porém nem sempre devidamente considerado, está relacionado ao fato de que um problema típico de configuração de rede logística envolve muitos dados, incluindo (Simchi-Levi *et al.*, 2000):

- localização dos clientes, varejistas, instalações produtivas, fornecedores, depósitos e centros de distribuição;

- o todos os produtos, incluindo volumes, modos de transporte e requisitos de manuseio e acondicionamento (por exemplo, refrigerados);
- o demanda anual para cada produto por cliente;
- o custos de transporte e/ou fretes por modal e por par origem-destino;
- o custos fixos e variáveis de movimentação e armazenagem nos depósitos;
- o custos fixos e variáveis nas instalações produtivas;
- o metas e requisitos de nível de serviço aos clientes (por exemplo, prazo de entrega, tamanho do pedido, etc.)

Uma análise rápida da lista acima faz constatar a elevada quantidade de dados necessários um problema de otimização de rede logística. Assim, a fim de tornar possível a modelagem, é imperativa a agregação de dados utilizando os seguintes critérios (Simchi-Levi *et al.*, 2000; Ratliff e Nulty, 1996):

- a) Clientes devem ser agrupados com base em algum critério de proximidade geográfica, por exemplo, em nível de município, ou de zona de CEP (código de endereçamento postal) de três dígitos, e tratados como um único ponto de demanda, a fim de evitar a explosão do tamanho do modelo. Deve-se notar que o agrupamento de clientes deve considerar a classificação dos mesmos segundo requisitos de nível de serviço, de frequência de entrega, de tamanho mínimo de pedido, etc. Por exemplo, se houver duas categorias de cliente, grandes redes de supermercado e pequeno varejo, os clientes de cada categoria são agrupados independentemente, mesmo que estejam geograficamente próximos. Busca-se, dessa forma, reduzir a dimensão do problema, que geralmente pode envolver até dezenas de milhares de clientes de forma a permitir a solução do modelo matemático correspondente.
- b) Analogamente, produtos devem ser agrupados em categorias, que reflitam os requisitos logísticos, tais como acondicionamento, transporte, manuseio, armazenamento, bem como aspectos da demanda. Exemplificando, alimentos e produtos de limpeza devem compor dois grupos distintos, dadas as restrições de transporte e armazenamento conjunto.

Com relação ao cálculo dos custos (ou fretes) de transporte, tendo em vista o elevado número de potenciais pares origem-destino (cada instalação candidata pode atender todos ou uma grande parte dos clientes), idealmente busca-se uma função que calcule automaticamente

todos os valores, com base nas distâncias, função essa normalmente obtida através de regressão de dados históricos de fretes praticados. No caso brasileiro surge uma dificuldade, decorrente do fato de que nem sempre os fretes podem ser calculados de forma automática, através de regressão, uma vez que os mesmos podem apresentar variações significativas para uma mesma distância, em função de fatores de mercado e concorrência entre empresas, da possibilidade de se obter frete de retorno e de outros aspectos geográficos. Adicionalmente, pode haver as opções de usar frota própria ou de contratação de terceiros. Assim, na maioria dos casos os fretes precisam ser determinados individualmente para cada par origem-destino, o que aumenta significativamente o trabalho de preparação de dados.

Desta forma, um dos grandes desafios é desenvolver algoritmos de solução mais rápidos, eficientes e robustos, que permitam obter soluções de boa qualidade e possibilitem tratar problemas maiores e mais complexos em termos do número de produtos, de níveis ou camadas, de instalações candidatas e de pontos de demanda.

5.1.3 Modelagem da Localização de Instalações com Múltiplos Critérios

Conforme se procurou demonstrar nas seções anteriores, modelos de otimização são ferramentas importantes e vêm sendo amplamente utilizados para resolver problemas reais que envolvem a localização de instalações, apesar das dificuldades para representar adequadamente a realidade de tais problemas. Por outro lado, há que se considerar que a decisão sobre onde localizar, a qual é de natureza estratégica e geralmente representa investimentos vultosos, envolve a avaliação de inúmeros critérios, com diferentes níveis de influência no resultado final, incluindo, muitas vezes, a necessidade de se considerar aspectos não só quantitativos, como também qualitativos, tornando o problema ainda mais complexo.

Por outro lado, ao longo dos últimos anos, e com maior ênfase a partir dos anos 70, têm sido crescentes as propostas de utilização de novas ferramentas para soluções de problemas complexos, como alternativa aos modelos de otimização tradicionais.

Nesse sentido, o Método de Análise Hierárquica - MAH (do inglês "*Analytic Hierarchy Process*" - AHP), proposto por Thomas L. Saaty (Saaty, 1991), vem sendo adotado em inúmeras situações que envolvem decisões com múltiplos critérios, uma vez que permite tratar, simultaneamente, aspectos quantitativos e qualitativos.

O MAH busca refletir o método natural da mente humana ao comparar dois elementos entre si. Segundo Saaty (1991), em problemas de tomada de decisão normalmente são avaliados e considerados inúmeros elementos que, aparentemente, nem sempre são comparáveis entre si de forma direta. Entretanto, ao se estabelecer um procedimento que agregue tais elementos, segundo propriedades comuns, torna-se possível compará-los.

O MAH permite a avaliação de múltiplos critérios, com modelos complexos que incluem e medem todos os fatores importantes, qualitativa e quantitativamente, sejam eles tangíveis ou intangíveis.

A aplicação do MAH requer, inicialmente, a estruturação do problema de modo hierárquico. A estruturação de hierarquias tem sido muito discutida, mas ainda não se estabeleceu um conjunto de procedimentos exatos e rigorosos de forma a gerar objetivos, critérios e as respectivas atividades com relativa clareza e precisão. Assim, os problemas e os sistemas a eles associados teriam sua hierarquia estruturada com base na somatória de experiências dos integrantes do grupo de decisão que, de uma maneira organizada (por exemplo, em sessões de *brainstorming*), o que permitiria auxiliar na identificação dos conceitos, dos objetivos, das atividades e dos atores e, em decorrência, a estruturação hierárquica do problema.

A título de ilustração, a Figura 5.2 mostra uma estrutura de decisão hierárquica típica em 3 níveis, incluindo a forma de comparação paritária entre as alternativas e os critérios.

Os valores dos julgamentos devem ser organizados numa matriz, sendo que, ambos, os valores e a matriz, deverão seguir os conceitos definidos por Saaty (1994), para o que caberia lembrar a formulação da comparação paritária, a saber:

- ao se comparar um elemento com ele mesmo o valor é 1; isto é, na comparação do elemento A com o elemento A , o fator comparativo é igual a 1, pois um elemento comparado com si mesmo resulta igualmente importante;
- ao se comparar um elemento com outro e, em seguida, comparar o segundo com o primeiro, os fatores comparativos são inversos entre si; isto é, ao se comparar o elemento A com o elemento B , pode-se obter, por exemplo, o fator comparativo de 2; portanto, ao compararmos o elemento B com o elemento A , teremos, conseqüentemente, o fator comparativo inverso de 2, ou seja $1/2$.

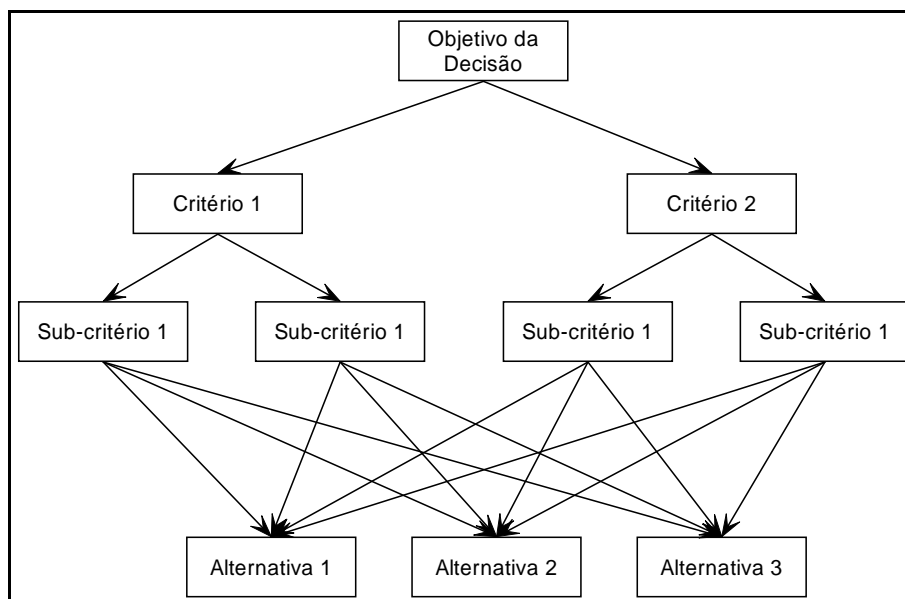


Figura 5.2: Estrutura de decisão hierárquica em 3 níveis

Lin e Tu (2000) argumentam que o problema de localização é complexo e o MAH pode, efetivamente, tratar os fatores qualitativos e quantitativos num ambiente de decisão de múltiplos critérios, configurando-se como uma importante ferramenta para tal, pois possui a habilidade de sintetizar cenários e produzir um diagnóstico que permita ao tomador de decisão entender o comportamento inter-relacional dos sub-sistemas que compõem a decisão.

A combinação do uso de modelos matemáticos com o MAH foi proposta por Mau-Crimming e Liberti (2002) para seleção de localização de reservas naturais. Esse estudo contemplou as ponderações dos acionistas da empresa e das demais partes envolvidas na decisão, para estruturação do problema no formato MAH. Os índices obtidos com o uso dessa metodologia foram utilizados como fatores a serem aplicados às variáveis de decisão que compõem a função objetivo e as respectivas restrições do modelo matemático formulado para o problema.

Galvão *et al.* (2003) aplicaram o MAH a um problema que envolvia a seleção de local para implantação do centro de distribuição de uma empresa do setor de bens de consumo não duráveis, cujos produtos requerem embalagem e manuseio diferenciado em função da sua fragilidade. O centro de distribuição envolverá um pequeno processo de industrialização para agregação de valor ao produto final, que nele será embalado, estocado e, a seguir, distribuído.

Uma etapa do processo envolveu a elaboração de uma lista de verificação (*check-list*), reproduzida no Quadro 5.1 e que contempla fatores relevantes a serem considerados e avaliados em problemas de localização. Essa lista almeja englobar diversos pontos de

verificação que podem influenciar a implantação de um empreendimento. A mesma é genérica, e deve ser adaptada ao tipo de instalação que será avaliada e implantada, com os necessários ajustes às respectivas características intrínsecas da instalação em análise, que pode ser um terminal de transbordo, um terminal portuário, uma unidade geradora de energia, um aeroporto, um centro de distribuição etc.

A estrutura hierárquica para o problema de localização do centro de distribuição, baseada na lista de verificação proposta no Quadro 5.1, foi implementada no *software* Expert Choice, conforme ilustrado na Figura 5.3.

A aplicação do MAH ao problema permitiu comprovar que o mesmo pode ser uma ferramenta útil para apoio à tomada de decisão sobre localização de instalações, principalmente na fase inicial do processo de avaliação, na qual há ausência de dados para uma modelagem mais sofisticada. Sua aplicação possibilitou a identificação da localização mais adequada para implantação do Centro de Distribuição, numa fase inicial do estudo. Essa ferramenta poderia também ser aplicada numa fase posterior do estudo, em que pudessem ser considerados, na estrutura hierárquica proposta, os resultados obtidos a partir de modelos de otimização aplicados ao problema.

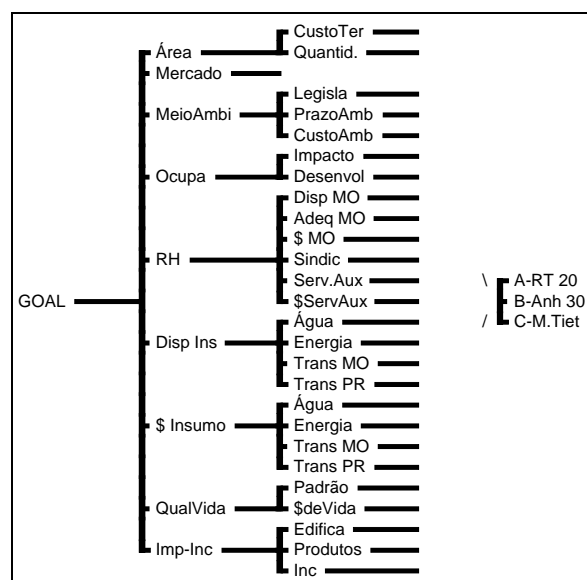


Figura 5.3: Estrutura hierárquica para o problema de localização do centro de distribuição

Fonte: Galvão *et al.* (2003)

Quadro 5.1: Lista de verificação proposta para seleção de localização

<ol style="list-style-type: none"> 1. Área para implantação <ol style="list-style-type: none"> a) custo da área b) disponibilidade de quantidade de área necessária para o projeto 2. Insumos <ol style="list-style-type: none"> a) Águas e Efluentes <ol style="list-style-type: none"> i) disponibilidade de fontes de abastecimento ii) disponibilidade de lançamento de efluentes iii) custos de fornecimento <ol style="list-style-type: none"> (1) disponibilidade em quantidade de água industrial e potável (2) distâncias das fontes (3) disponibilidade de corpo receptor para o lançamento de efluentes (4) sistemas de captação de água / adução / tratamento e armazenamento (5) tratamento da água (6) tratamento do efluente b) Gás Natural <ol style="list-style-type: none"> i) disponibilidade de gás natural ii) contrato de fornecimento de gás natural e sua confiabilidade iii) custo de fornecimento <ol style="list-style-type: none"> (1) distância do gasoduto (2) custos de acesso às redes de gás (3) custo do gás c) Energia Elétrica <ol style="list-style-type: none"> i) disponibilidade e fontes de energia elétrica ii) contrato de fornecimento de energia e sua confiabilidade iii) custo de fornecimento <ol style="list-style-type: none"> (1) distancias às redes elétricas (2) custo de acesso às redes de energia elétrica (3) integração na rede (4) reforços necessários para a saída da energia gerada d) Transporte <ol style="list-style-type: none"> i) Mão de Obra <ol style="list-style-type: none"> (1) Disponibilidade (2) Custo ii) Produto <ol style="list-style-type: none"> (1) Disponibilidade (2) Custo iii) Matéria Prima <ol style="list-style-type: none"> (1) Disponibilidade (2) Custo iv) Vias de Acesso <ol style="list-style-type: none"> (1) Rodoviário (2) Ferroviário (3) Hidroviário (4) Aeroviário e) Matéria-Prima <ol style="list-style-type: none"> i) disponibilidade e custo 	<ol style="list-style-type: none"> 3. Mercado <ol style="list-style-type: none"> a) Local b) Exportação para outros centros consumidores c) identificação de investimentos previstos na região d) concorrência 4. Aspectos Ambientais <ol style="list-style-type: none"> a) legislação ambiental estadual e municipal b) etapas e prazos para licenciamento ambiental c) impedimentos ambientais d) custo de atendimento à legislação 5. Vegetação, Fauna e Clima <ol style="list-style-type: none"> a) vegetação e seus dispositivos legais de proteção b) espécies de fauna em extinção e seus dispositivos legais de proteção no município c) clima e temperatura d) regime pluviométrico, enchentes e inundações e) ventos 6. Ocupação Urbana e Habitação <ol style="list-style-type: none"> a) cidades próximas b) população c) planos de desenvolvimento d) impacto social 7. Recursos Humanos <ol style="list-style-type: none"> a) legislação trabalhista b) sindicatos e sua atuação na região c) empresas de engenharia e de construção d) disponibilidade de serviços e de profissionais especializados e) disponibilidade de programas de treinamento educacional e vocacional f) disponibilidade de serviços de assistência médica e odontológica g) custos <ol style="list-style-type: none"> i) mão de obra direta ii) serviços auxiliares iii) mão de obra indireta e de suporte 8. Qualidade de Vida <ol style="list-style-type: none"> a) outras indústrias na região e qualidade de mão-de-obra e seu comportamento geral b) condições para moradia de trabalhadores na região c) moradia, segurança e infra-estrutura social d) disponibilidade para lazer e) custo de vida na região <ol style="list-style-type: none"> i) moradia ii) transporte iii) infra-estrutura 9. Impostos e Taxas <ol style="list-style-type: none"> a) área b) insumos c) mão-de-obra d) remessa de lucros e) outros 10. Incentivos Fiscais e Tributários <ol style="list-style-type: none"> a) estaduais b) municipais
---	--

5.2 CONFIGURAÇÃO DE REDES DO TIPO HUB-AND-SPOKE PARA O TRANSPORTE RODOVIÁRIO DE CARGA PARCELADA

O segmento de carga parcelada abrange, como o próprio nome já diz, os serviços de transporte de cargas cujo volume, para um cliente, uma origem e um destino, não é suficiente para lotar um veículo. Assim, cargas de diversos clientes são agregadas e transportadas conjuntamente. Esse tipo de serviço normalmente engloba as operações de coleta na origem, de consolidação com outras cargas de outros clientes para transferência de longa distância, de desconsolidação e de entrega no destino.

Conseqüentemente, as empresas de transporte rodoviário necessitam dispor de estruturas de apoio, ou seja, instalações geograficamente distribuídas, usualmente denominadas *filiais*, nas quais são consolidadas cargas de diferentes origens, para um ou mais destinos convergentes. Para uma empresa, o número e a localização dessas *filiais*, na quase totalidade das vezes, são definidas ao longo do tempo, em função da demanda de cargas, bem como pela prática e experiência do responsável pela operação da empresa.

Existem vários benefícios na consolidação de carga parcelada, e talvez o mais relevante é a possível redução dos custos de transporte nas transferências de longa distância, através da agregação de cargas de diferentes origens para o mesmo destino (ou destinos próximos). Assim, o planejamento da rede de transporte de uma transportadora, ou seja, a definição da localização e da área de atuação de cada um desses terminais de consolidação, é um componente crítico para o desempenho do sistema como um todo. Mais especificamente, o objetivo é determinar o número de terminais de consolidação, a localização de cada um deles, bem como sua respectiva área de atuação, de forma a minimizar a somatória dos custos envolvidos, isto é, custos variáveis de transporte e custos fixos das instalações ou terminais, atendendo requisitos de nível de serviço pré-definidos.

Muitas empresas de transporte de carga parcelada, em particular no segmento expresso, isto é, com prazos de entrega reduzidos e garantidos, têm desenvolvido sistemas de coleta, transferência e entregas usando redes do tipo “*hub-and-spoke*”. Ao invés de se estabelecerem ligações diretas entre todas as filiais, cujos volumes de carga nem sempre justificam a frequência que seria necessária para atender requisitos de nível de serviço (em particular prazos de entrega), algumas filiais funcionam como pontos concentradores de carga,

viabilizando um sistema de transferência eficiente, tanto em termos de custos quanto de qualidade de serviço, através da consolidação de cargas.

O termo “*hub-and-spoke*” é utilizado para descrever a estrutura da rede. Em uma tradução direta, temos “cubo e raio” fazendo uma analogia com as rodas de uma carroça, por exemplo. Segundo Aykin (1995), sistemas do tipo “*hub-and-spoke*” passaram a ser muito utilizados no transporte aéreo dos Estados Unidos, principalmente em decorrência do Ato de Desregulamentação do Transporte Aéreo de 1978, que permitiu às companhias aéreas escolherem as rotas que quisessem ofertar e as respectivas tarifas que iriam praticar. Tais mudanças tiveram efeitos profundos sobre muitos aspectos da operação aérea desse país, particularmente no que tange a tarifas, nível de serviço, qualidade e segurança. Entretanto, talvez o aspecto mais importante foi que estas empresas aéreas alteraram a estrutura de suas rotas, desenvolvendo redes do tipo “*hub-and-spoke*”.

A ilustração da Figura 5.4 representa um exemplo de uma rede do tipo “*hub-and-spoke*”, em que os nós k e l representam os terminais para consolidação de carga (“*hubs*”) e os nós i e j aqueles que são atendidos através dos terminais (“*spokes*”). Um fluxo de carga T_{ij} , com origem no nó i e destino em j , é atendido da seguinte forma: inicialmente a carga em i é enviada para o terminal de consolidação (“*hub*”) k , onde é consolidada com outras cargas recebidas e enviada para o “*hub*” l e daí distribuída para os pontos atendidos por l , incluindo o nó j .

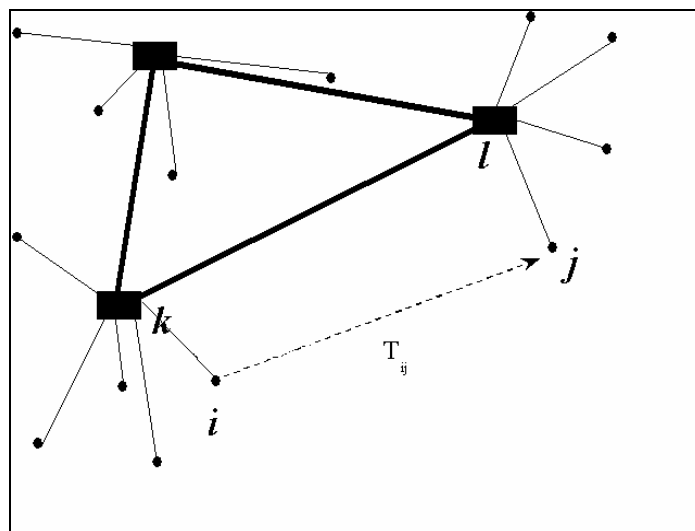


Figura 5.4: Exemplo de rede do tipo “*hub-and-spoke*”

Fonte: Silva (2004)

Redes do tipo “*hub-and-spoke*” também são extensivamente usadas no setor de encomendas expressas, de forma a minimizar custos e assegurar prazos de atendimento reduzidos. Na Figura 5.5 é mostrada a rede da empresa de encomendas expressas FedEx nos EUA com as ligações entre *hubs* em vermelho.

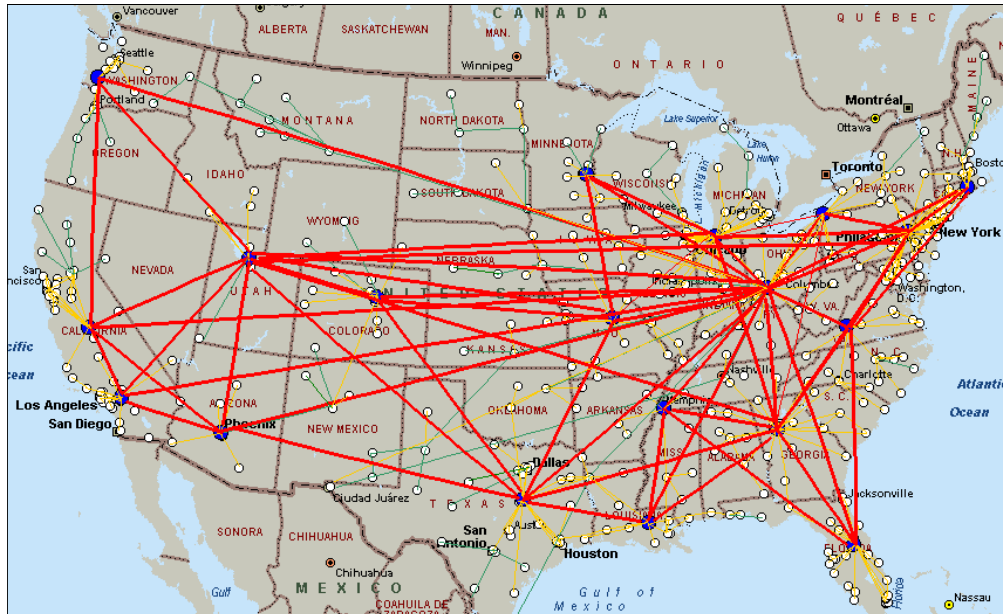


Figura 5.5: Rede “*hub-and-spoke*” da DHL nos EUA

Fonte: DHL

Uma pesquisa de mestrado, orientada pelo autor e conduzida por Silva (2004), explorou o problema de otimização da configuração de uma rede do tipo “*hub-and-spoke*” no contexto do transporte rodoviário de carga parcelada no Brasil. A modelagem diferenciou-se das demais encontradas na literatura por permitir considerar fatores de economia de escala para os custos de transporte que variam de acordo com os respectivos volumes de carga entre instalações, como ocorre com as transportadoras de carga no Brasil, tendo sido aplicada a um caso real para uma empresa que opera no segmento de transporte de carga parcelada pelo modal rodoviário no Brasil. Essa pesquisa resultou ainda três artigos científicos: Cunha e Silva (2006), Silva e Cunha (2003, 2004).

5.2.1 Problemas de Configuração de Redes “*Hub-and-Spoke*” na Literatura

O Problema de Localização de Terminais Não Capacitado com Alocação Única (PLTNC-AU) consiste em determinar o número e a localização de terminais de consolidação (“*hubs*”) e

quais pontos ou filiais (“*spokes*”) são alocados a cada terminal de consolidação (“*hub*”), de forma a minimizar a soma dos custos variáveis de transporte e dos custos fixos de operação desses terminais. Os terminais de consolidação são considerados não capacitados e cada filial só pode ser atendida por um único terminal de consolidação, daí o nome alocação única.

O PLTNC-AU foi estudado primeiramente por O’Kelly (1987), que formulou um modelo de programação inteira quadrática e propôs duas heurísticas, ambas fundamentadas na enumeração exaustiva de todas as alternativas de localização para “*hubs*”, enquanto que o critério para a designação dos nós não-“*hubs*” aos “*hubs*” baseou-se exclusivamente na distância ao “*hub*” mais próximo.

O PLTNC-AU está relacionado a diversos outros problemas, como por exemplo o p -“*hub*” *Location Problem*, em que o número de “*hubs*” é definido a priori. O’Kelly (1987) propôs a primeira formulação matemática para esse problema, como um problema de programação inteira quadrática. Segundo o autor, mesmo esse problema, em que o número de “*hubs*” é fixo, pertence à categoria conhecida como *NP*-difícil.

Uma heurística de troca (única e dupla) foi proposta por Klincewicz (1991), e também uma heurística de agrupamento de pontos (*cluster*) para o p -“*hub*” *location problem*. Klincewicz (1992) propôs duas outras heurísticas, baseadas em busca tabu e GRASP (*Greedy Randomized Adaptive Search Procedure*). Skorin-Kapov e Skorin-Kapov (1994) propuseram um algoritmo baseado em busca tabu.. O primeiro algoritmo exato a resolver este problema foi proposto por Skorin-Kapov *et al.* (1996), e se mostrou prático para redes de até 25 nós.

Sohn e Park (1998) consideraram tanto a variante com alocação única, como também a com alocação múltipla dos nós não “*hubs*” para o problema em que p é dado. Para o problema de alocação única, mais difícil do ponto de vista matemático, os autores propuseram um novo modelo matemático que permitiu reduzir pela metade o número de variáveis e restrições em relação à formulação proposta por Skorin-Kapov *et al.* (1996), possibilitando a obtenção de soluções ótimas para a maioria das instâncias analisadas pelos mesmos.

Klincewicz (1991) propôs uma heurística de troca e uma heurística de agrupamento para o PLTNC-AU, obtendo boas soluções para inúmeros testes realizados. Abdinnour-Helm e Venkataramanan (1995) desenvolveram uma heurística baseada em “*branch-and-bound*” e em algoritmos genéticos para resolver esse problema.

Skorin-Kapov e Skorin-Kapov (1994) propuseram um método de solução baseado em Busca Tabu, e para teste deste método de solução, foi utilizada a mesma base de dados que O’Kelly (1987) utilizou, sendo os dados do fluxo aéreo de passageiros entre as 25 maiores cidades dos Estados Unidos em 1970. Para todas os testes realizados, a busca tabu permitiu melhorar todas as soluções encontradas até a época. Skorin-Kapov *et al.* (1996) verificaram que através da busca tabu foi possível encontrar a solução ótima em todos os casos, sendo a mesma sugerida para resolver problemas de maior porte, como os problemas reais em redes de transporte de carga parcelada.

Abdinnour-Helm (1998) desenvolveu uma heurística híbrida para resolver o PLTNC-AU, obtendo a solução ótima em 98% dos casos (63 de um total de 64 problemas). Esta heurística foi baseada em algoritmo genético e busca tabu, sendo o primeiro utilizado para resolver a parte locacional do problema, ou seja, definir a quantidade e a localização de “hubs”, e a busca tabu para a parte alocaional do problema, ou seja, a alocação de nós não-“hub” aos nós “hubs”.

Estratégias de solução baseadas em algoritmos genéticos também foram aplicadas por Jaramillo *et al.* (2002) para a resolução de outros tipos de problemas de localização de instalações, entre os quais problemas de carga fixa capacitados e não-capacitados, o problema de cobertura máxima e modelos de localização competitiva, com resultados bastante promissores em relação a outros métodos disponíveis na literatura.

Entretanto, em nenhum desses trabalhos foi considerada uma questão prática e importante que ocorre no segmento de transportadoras de carga parcelada no Brasil: o fator de redução (economia de escala) do custo unitário de transporte entre dois “hubs” pode ser variável e função não só da distância e da localização dos terminais, como também do volume total de carga entre esses dois pontos. Dessa forma, procura-se levar em consideração que uma configuração da rede com muitos “hubs” não só representa maiores custos fixos, mas também custos de transporte mais elevados, uma vez que os volumes de carga entre “hubs” nem sempre podem justificar cargas fechadas em veículos grandes, nem tampouco é possível aguardar um volume de carga suficiente para o despacho de um veículo grande com carga completa, em função dos prazos de entrega acordados com os clientes e da prática do mercado.

5.2.2 Formulação Matemática

O problema proposto, denominado Problema de Localização de Terminais Não Capacitado com Alocação Única e Fatores de Redução Variáveis (PLTNC-AUFRV) consiste em determinar o número e a localização de terminais de consolidação (“hubs”) e a alocação dos pontos ou filiais (“spokes”) a cada terminal de consolidação (“hub”), de forma a minimizar o custo total de transporte e fixo das instalações. Neste problema o número de terminais de consolidação é uma variável de decisão, uma vez que, a cada instalação candidata está associado um custo fixo.

Seja $i=1, 2, \dots, N$ um conjunto de filiais de uma empresa transportadora. Para duas filiais $i, j \in N$, T_{ij} e C_{ij} representam, respectivamente, o fluxo e o custo unitário de transporte entre os nós i e j . Os valores O_i e D_i representam, respectivamente, os volumes totais de carga originados na filial i ($O_i = \sum_j T_{ij}$) e destinados à mesma ($D_i = \sum_j T_{ji}$), respectivamente; f_j indica o custo fixo de instalação e operação de um terminal de consolidação localizado no nó j . Conforme mencionado anteriormente, $\alpha_{km}(T_{km})$ corresponde a um fator de redução (economia de escala) na ligação entre os terminais de consolidação k e m , que é função do volume total de carga movimentada T_{km} entre esses terminais, não necessariamente restrito a funções lineares ou contínuas, nem simetria, i.e. $\alpha_{km}(T_{km}) \neq \alpha_{mk}(T_{mk})$.

A variável de decisão X_{ik} assume o valor 1 se o nó i é alocado ao “hub” k e igual a zero caso contrário. Cada “hub” é alocado a si próprio, ou seja, $X_{jj}=1$ se o nó j for um “hub”

A formulação matemática para o PLTNC-AUFRV pode ser escrita como:

$$\text{Minimizar: } \sum_i \sum_k X_{ik} C_{ik} (O_i + D_i) + \sum_j \sum_m X_{jm} \sum_i \sum_k X_{ik} (T_{ij} C_{km} \alpha_{km}(T_{km})) + \sum_j X_{jj} f_j \quad (5.32)$$

$$\text{sujeito a } X_{ij} \leq X_{jj} \quad \text{para todo } i, j \in N \quad (5.33)$$

$$\sum_j X_{ij} = 1 \quad \text{para todo } i \in N \quad (5.34)$$

$$X_{ij} \in \{0,1\} \quad \text{para todo } i, j \in N \quad (5.35)$$

A função objetivo (5.32) compreende a minimização das parcelas de custo: de transporte para as cargas com origem e destino em cada terminal, de transferência entre terminais e de custo

fixo total para os terminais selecionados. A restrição (5.33) impõe que uma filial i só possa ser atendida a partir de um “*hub*” selecionado. A restrição (5.34) assegura que cada filial seja designada a um e somente um “*hub*”. A restrição (5.35) está relacionada à integralidade das variáveis de decisão. Deve-se notar que esta formulação corresponde a um problema de programação inteira mista com função objetivo não linear e mais complexa que quadrática (que seria relativamente fácil de resolver), como no problema original, tendo em vista os fatores de economia de escala variáveis na função objetivo.

5.2.3 Estratégia de Solução Proposta

O Problema de Localização de Terminais Não Capacitado com Alocação Única (PLTNC-AU), em que o fator de redução dos custos nas ligações entre “*hubs*” é único e constante (isto é, $\alpha_{km}(T_{km}) = \alpha$), possui ordem de complexidade exponencial, o que impede a solução de instâncias reais através de algoritmos exatos disponíveis comercialmente. Já no caso do PLTNC-AUFRV, a função objetivo é não linear, o que reforça ainda mais a necessidade de uma heurística visando a resolução de instâncias reais.

A estratégia de solução proposta baseia-se em Algoritmo Genético. Mais especificamente, foram propostas três diferentes heurísticas para solução do PLTNC-AUFRV. Na primeira delas, denominada AG1, o algoritmo genético tem por finalidade buscar a configuração ótima em termos dos *hubs*. Uma heurística gulosa (*greedy*) é utilizada para a definição da alocação das filiais não-*hubs* ao respectivo “*hub*” mais próximo, tendo em vista que os “*hubs*” não apresentam restrição de capacidade. A grande virtude dessa heurística é a simplicidade e velocidade de processamento, pois permite calcular rapidamente a aptidão de cada indivíduo da população. Entretanto, nos testes computacionais realizados para validação da heurística, descritos em Silva (2004), verificou-se que, de um modo geral, embora fossem selecionados os melhores “*hubs*”, essa alocação das demais filiais ao respectivo “*hub*” mais próximo muitas vezes não era a ótima, uma vez que não considera as economias de escala nas ligações entre *hubs* (que pode levar um nó não-“*hub*” a ser alocado a outro “*hub*” que não o mais próximo).

Em decorrência, foi proposta uma nova heurística (AG2) que usa algoritmo genético, em duas fases seqüenciais: a primeira fase corresponde exatamente à heurística AG1, para a escolha

dos *hubs*. Uma vez selecionados os “*hubs*”, um segundo algoritmo genético é aplicado para a alocação dos nós não-“*hubs*”.

Inspirando-se no trabalho de Beasley (1999), que obteve resultados encorajadores ao incorporar ao algoritmo genético uma heurística de busca local, com a finalidade de buscar “melhorar” cada novo indivíduo gerado, foi desenvolvida uma terceira heurística de solução, denominada AG3. Essa heurística corresponde, em linha gerais, à heurística AG1 com a incorporação de uma heurística de busca local, que consistiu basicamente na tentativa de melhorar a aptidão de cada indivíduo gerado, permitindo novas realocações dos nós aos “*hubs*” encontrados. As etapas básicas desta heurística foram: gerar um indivíduo, fazer a alocação dos nós com base na distância (como no AG1) e, com base nesta solução, utilizar o mecanismo de busca local desenvolvido, testando novas alocações, visando melhorar a solução. Como mecanismo de busca local utilizou-se um método baseado no algoritmo Metropolis (Metropolis *et al.*, 1953), que inspirou o desenvolvimento do *simulated annealing*, sem o esquema de resfriamento de temperatura. São testados os movimentos de realocação dos nós não-“*hubs*”, com uma probabilidade de aceitação de um movimento que degrada a solução.

Um exemplo ilustrativo da representação cromossômica proposta é mostrado na Figura 5.6, para um problema com 10 nós (ou terminais) candidatos a “*hubs*”. O caso (a) representa um possível cromossomo de AG1 e AG3, e também da primeira fase de AG2, indicando duas instalações selecionadas como “*hubs*”, nos nós 4 e 6; pode-se também visualizar a alocação dos nós não-“*hubs*” 1, 2, 3, 5, 7, 8, 9 e 10 aos respectivos “*hubs*” mais próximos. Para o AG2, o caso (b) indica que os bits podem assumir os valores 1 ou 2, correspondentes aos nós 4 e 6, respectivamente. Um cromossomo resultante dessa segunda fase alocacional dos nós não-“*hubs*” é indicado em (c), onde o nó 1 é alocado ao “*hub*” que ocupa a posição 1 no vetor *hub*, que é a filial 4; analogamente o nó 2 é atendido pela filial 4 e assim sucessivamente.

A aptidão de um indivíduo é medida diretamente pelo valor da função objetivo do problema em estudo. Para os algoritmos AG1 e AG3, bem como a primeira fase do AG2, os vetores individuais na população são gerados aleatoriamente de tal forma que uma determinada posição neste vetor (*bit*) pode se tornar um “*hub*” (tomar o valor 1) com uma probabilidade maior que zero. Neste caso a probabilidade utilizada foi de 15%. Este valor de probabilidade foi obtido experimentalmente após alguns testes com diferentes probabilidades, sendo 15% o valor que convergiu mais rapidamente. Já para a segunda fase do AG2, cada gene é gerado

pelo sorteio aleatório de um determinado número entre 1 e K , onde K é o número de hubs determinado na primeira fase. Foi utilizado o método de seleção por torneio (“*tournament selection*”) para determinação dos pares para cruzamento, sendo adotado o o cruzamento em dois pontos (“*two-point crossover*”). A mutação foi aplicada com probabilidade de 1%.

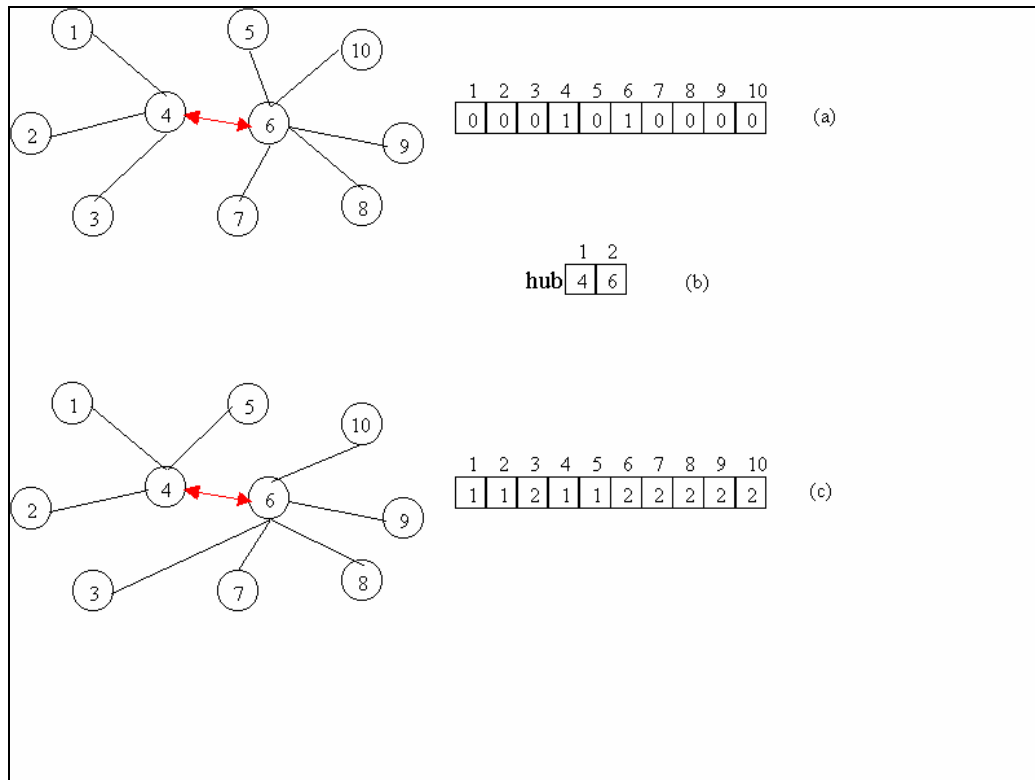


Figura 5.6: Codificação do Algoritmo Genético

Fonte: Silva (2004)

5.2.4 Experimentos Computacionais

As heurísticas propostas foram implementadas em C++, utilizando a biblioteca de Lobo (1997). Os testes foram realizados em um PC Intel Pentium IV de 1,7 GHz com 256MB de memória RAM e

Inicialmente foram realizados experimentos para validação das heurísticas implementadas (AG1, AG2 e AG3) para o PLTNC-AU. Foi utilizada para os testes a base de dados proposta por O’Kelly (1987), que vem sendo usada na literatura como *benchmark* para várias heurísticas para o problema de localização de “*hubs*”. Os dados são baseados no fluxo aéreo de passageiros entre as 25 maiores cidades dos Estados Unidos da América em 1970, como

avaliado pelo Departamento de Aviação Civil (*Civil Aeronautics Board*). Estas 25 cidades fazem parte de um conjunto maior de dados consistindo no fluxo de 100 cidades dos Estados Unidos. As distâncias e os fluxos entre as cidades são simétricos. Vale lembrar que, para comparação com outros trabalhos publicados na literatura, o parâmetro $\alpha_{km}(T_{km})$ foi mantido fixo, independente do volume de carga transportado entre “hubs”, sendo portanto para estes experimentos computacionais, chamado simplesmente de α .

Os parâmetros iniciais utilizados para o algoritmo genético foram definidos com base no estudo realizado por Schaffer *et al.* (1989) apud Abdinnour-Helm (1998), e Beasley (1999), sendo ajustados de forma experimental com base em experimentos computacionais realizados, sendo: tamanho da população de 100 indivíduos; probabilidade de cruzamento de 0,99; probabilidade de mutação de 0,01

O valor de temperatura no AG3 foi obtido experimentalmente e dependeu primeiramente da quantidade de nós do problema em avaliação. Alguns pesquisadores sugerem que o valor da temperatura deva ser tal que permita que movimentos desfavoráveis sejam aceitos na razão de 80% dos movimentos testados. Em testes preliminares realizados para o PLTNC-AU, esta taxa de aceitação de 80% se mostrou muito alta, sendo utilizado um valor médio de 40% de aceitação.

Os resultados obtidos foram comparados com os de Abdinnour-Helm (1998) e Abdinnour-Helm; Venkataramanan (1995) apud Abdinnour-Helm (1998), que desenvolveram, respectivamente, uma heurística híbrida baseada em algoritmos genéticos e busca tabu, e outra baseada puramente em algoritmos genéticos; sendo estas as duas únicas heurísticas desenvolvidas até o momento para resolver o PLTNC-AU (Abdinnour-Helm, 1998).

Foram considerados problemas com 10, 15, 20 e 25 cidades, conforme Abdinnour-Helm (1998), e para cada um deles quatro diferentes valores para o fator α : 1; 0,8; 0,6 e 0,4. Foram também considerados quatro valores para o custo fixo de operação de um terminal de consolidação no nó j (f_j): 250, 200, 150 e 100.

A qualidade das soluções obtidas foram avaliadas comparando com as soluções ótimas derivadas do estudo de Skorin-Kapov, D.; Skorin-Kapov, J.; O’Kelly (1996) indicadas por “Ótimo”. A estratégia de solução proposta por estes autores permitiu encontrar a solução

ótima para as instâncias do problema de *p-hubs* (*p-hub median problem*) para valores de *p* iguais a 2, 3 e 4.

A heurística AG3 foi a que apresentou o melhor desempenho, tendo encontrado a solução ótima em 89,1% dos problemas (57 de um total de 64 problemas), destacando-se frente às outras heurísticas devido ao desvio máximo da solução ótima que foi de apenas 0,35%, e com desvio médio da solução ótima de 0,13%. A heurística AG2 também teve um bom desempenho, obtendo a solução ótima em 85,9% dos problemas (55 de um total de 64 problemas), sendo que nos demais casos o desvio máximo da solução ótima foi de 1,3% em dois casos, tendo como desvio médio da solução ótima de 0,8%. Já a heurística AG1 obteve a solução ótima em 73,4% dos problemas (47 de 64), com desvio médio de 0,8% e desvio máximo de 2% (em um único problema). Resultados detalhados podem ser encontrados em Silva (2004).

Os experimentos computacionais realizados evidenciam a eficiência da estratégia de solução baseada em Algoritmo Genético para o problema de configuração de redes “*Hub-and-Spoke*”, mais especificamente o Problema de Localização de Terminais Não Capacitado com Alocação Única (PLTNC-AU).

5.2.5 Estudo de Caso

Considerou-se o problema real de uma grande empresa de transporte rodoviário de carga parcelada modal. Essa empresa opera com “*hubs*” que foram criados de forma empírica para atender os objetivos de expansão e crescimento da área de atuação da empresa. A mesma possui 46 filiais, atendendo mais de 6.000 municípios do país.

A empresa transporta em média 173 mil toneladas de carga por ano, tendo que o fluxo principal desta carga refere-se ao transporte cujas origens principais estão nas regiões sul e sudeste do país, e os respectivos destinos principais concentram-se nas regiões norte e nordeste. A receita operacional líquida é de cerca de R\$ 110 milhões por ano.

A frota é composta de veículos próprios e agregados, sendo a primeira (275 veículos) utilizada principalmente para as operações de coleta e entrega de carga, enquanto que os 800 veículos agregados estão dedicados principalmente às operações de transferência da carga entre filiais. Aproximadamente 55% da frota é rastreada via satélite.

Foi necessário um levantamento de dados dos fluxos de carga entre filiais e dos custos de transporte associados a estas operações para o período entre janeiro a agosto de 2002. Os custos de transferência entre filiais foram obtidos junto ao departamento de tráfego da empresa, que possui uma tabela de fretes pagos para transferências entre filiais para diversos tipos de veículos. É importante salientar que, ao contrário dos problemas encontrados na literatura e utilizados nos experimentos computacionais descritos acima, não há simetria nos custos de transporte de uma cidade x para outra y e vice versa.

Os valores assumidos pelo parâmetro de redução de escala $\alpha(T)$ seguiram o fluxo médio diário de carga entre filiais conforme apresentado na Tabela 5.1.

Tabela 5.1: Valores de $\alpha(T)$ de acordo com o fluxo médio diário de carga

Kg/dia	$\alpha_{km}(T_{km})$
Até 1000	1.00
1000 – 3000	0.95
3000 – 6000	0.90
6000 – 12000	0.80
Acima de 12000	0.70

Estimou-se em R\$ 2.600,00 o custo fixo mensal de operação de um “hub”, com base no custo da mão de obra incluindo salários adicionado aos encargos de uma equipe básica para descarga de um veículo em trânsito. Para complementar a análise, foi feita uma análise de sensibilidade considerando-se uma variação de +50% deste valor (\$3.900) e -50% (\$1.300), além da situação sem custo fixo.

As três heurísticas foram testadas para o problema, sendo que a AG3 foi a que obteve os melhores resultados (11% superior que a AG2). Assim, a mesma foi utilizada nos testes, considerando-se diferentes cenários em que se variaram os custos fixos, variáveis e fluxos entre pares de nós, conforme mostrado na Tabela 5.2 e na Tabela 5.3. Para cada entrada, a primeira linha indica os “hubs” selecionados, e a segunda a economia em relação à situação atual. Os resultados evidenciam que a configuração com dois “hubs” (1 e 45) é a de mínimo custo, levando a uma economia de 28,4% em relação à situação atual. Essa configuração (nós 1 e 45) é a mesma para a maioria dos cenários alternativos avaliados, com economias sempre superiores a 20%; a única exceção é quando os custos fixos são 50% menores (cenário pouco

provável) quando mais “hubs” são selecionados, entre os quais os nós 1 e 45. Deve-se notar que todas as 46 filiais eram candidatas, não se pré-definindo um número máximo de “hubs”.

Tabela 5.2: Hubs selecionados e economias para diferentes demandas e custos fixos

Custo Fixo	Demanda (T_{ij})			
	Base	+10%	+20%	-10%
Base	1, 45 +28.4%	1, 45 +27.3%	1, 45 +26.3%	1, 45 +29.6%
+20%	1, 45 +30.8%	1, 45 +29.6%	1, 45 +28.5%	1, 45 +32.2%
-20%	1, 45 +25.7%	1, 45 +24.8%	1, 45 +23.8%	1, 45 +26.8%
+50%	1, 45 +33.9%	1, 45 +32.7%	1, 45 +31.4%	1, 45 +35.5%
-50%	1, 35, 45 +21.1%	1, 45 +20.6%	1, 19, 42, 45 +20.3%	1, 45 +21.8%

Fonte: Cunha e Silva (2006)

Tabela 5.3: Hubs selecionados e economias para diferentes custos fixos e variáveis

Custo Fixo	Custo unitário de transporte (C_{ij})			
	Base	+10%	+20%	-10%
Base	1, 45 +28.4%	1, 45 +27.2%	1, 45 +26.1%	1, 45 +29.7%
+20%	1, 45 +30.8%	1, 45 +29.5%	1, 45 +28.4%	1, 45 +32.2%
-20%	1, 45 +25.7%	1, 45 +24.6%	1, 45 +23.7%	1, 45 +26.9%

Fonte: Cunha e Silva (2006)

Os tempos de processamento foram inferiores a 300 segundos de CPU, o que pode ser considerado bastante satisfatório dado o tamanho do problema (46 nós). Tempos de processamento superiores (até 1 hora de CPU) não permitiram obter resultados melhores.

5.2.6 Considerações Finais e Continuidade da Pesquisa

O transporte rodoviário de cargas, pela sua importância cada vez maior no contexto logístico brasileiro, requer a contínua busca da melhoria de desempenho, medido em termos de

eficiência econômica e de qualidade de serviço. Muitas transportadoras foram crescendo, se expandindo, abrindo novas filiais, sem um planejamento de suas redes logísticas. Assim, existem muitas oportunidades para redução de custos e melhoria de nível de serviço. Adicionalmente, muitas empresas de transporte de carga parcelada estruturaram seus sistemas de coleta, transferência e entregas usando redes do tipo “*hub-and-spoke*”.

A heurística proposta, baseada em Algoritmos Genéticos, é inédita em termos da literatura, e resultou em uma publicação em periódico internacional (Cunha e Silva, 2006), com destaque para a variação do parâmetro α (fator de redução no custo de transporte na ligação entre “*hubs*”) em função do fluxo médio diário de carga entre “*hubs*”, e a utilização dos Algoritmos Genéticos tanto para a parte locacional (definição da quantidade e localização de “*hubs*”), como para a parte alocaional do problema (definição da designação de um nó não-“*hub*” a um “*hub*” definido). Destaca-se ainda a aplicação da estratégia de solução proposta ao importante problema de configuração da rede para transportadoras de carga parcelada no Brasil.

Diversas são as oportunidades de continuidade e extensão desta pesquisa, que estão sendo perseguidas pelo autor, através da orientação de pesquisas de mestrado e doutorado:

- Investigação de novos mecanismos para a fase alocaional do Algoritmo Genético, isto é, buscar um mecanismo mais eficiente para alocação dos nós aos *hubs* do que o considerado na heurística AG1 (com base exclusivamente na distância), e que possibilite a redução do tempo de processamento, um dos gargalos da heurística AG3, principalmente para problemas de maior porte.
- Manter e explorar um *pool* de soluções de elite (alta qualidade) geradas pelo algoritmo genético (ao invés de apenas a melhor solução), possibilitando ao tomador de decisão analisar os *trade-offs*, por exemplo, de custo *versus* nível de serviço (difícil de incorporar na modelagem) e eventualmente selecionar uma outra solução que não a de menor custo, porém com nível de serviço melhor.
- Propor uma nova formulação matemática para o Problema de Localização de Terminais Não Capacitado com Alocação Única (PLTNC-AU) que permita a linearização da sua função objetivo (atualmente quadrática), de forma a permitir obter, utilizando algum pacote eficiente de programação inteira (por exemplo, o CPLEX da ILOG), a solução ótima para as instâncias da base de dados proposta por O’Kelly

(1987), que vêm sendo utilizadas na literatura como *benchmarks* para várias heurísticas para o problema de localização de “*hubs*”.

- Desenvolver e avaliar novas versões de heurísticas baseadas em algoritmos populacionais ou metaheurísticas híbridas para problemas do tipo “*hub-and-spoke*” não capacitados e também capacitados, que podem ocorrer no contexto de empresas de transporte de carga parcelada.
- Desenvolver e avaliar novas versões de heurísticas baseadas em Algoritmos Genéticos para problemas do tipo “*hub-and-spoke*” com múltiplos critérios, uma nova linha de pesquisa para problemas de localização de instalações e projeto de redes que vêm sendo explorada pelo Prof. Mark S. Daskin, da *Northwestern University*.

5.3 OTIMIZAÇÃO DA LOGÍSTICA DE PRODUÇÃO E DISTRIBUIÇÃO DE REVISTAS

Uma pesquisa de mestrado, orientada pelo autor e conduzida por Mutarelli (2004), tratou do problema de modelagem de uma rede de produção e distribuição de uma revista semanal de informações de grande tiragem.

O principal problema de distribuição em uma editora de revistas é a seleção das unidades de produção e a alocação de pontos de demanda a unidades gráficas. A complexidade do problema aumenta quando se trata de uma revista semanal, pois pode haver múltiplas unidades gráficas, dezenas de centros de distribuição e centenas de milhares de leitores que devem receber seus exemplares em prazo reduzido, situação essa agravada pela extensão do território nacional. As decisões de quais gráficas utilizar, a alocação das regiões de demanda às gráficas, o seqüenciamento de produção em cada unidade gráfica, e a escolha do modal de transporte, devem ser tomadas de maneira integrada, buscando otimizar a soma dos custos totais de produção e distribuição, e respeitando requisitos de nível de serviço e restrições de produção.

Não existem na literatura muitos trabalhos que tratam do assunto. Dillmann *et al.* (1996) abordam aspectos práticos que afetam a aplicação de modelos de otimização no planejamento dos roteiros de entregas de jornais e revistas para atacadistas na Alemanha. Nenhum modelo

ou algoritmo específico é proposto, porém são analisadas e discutidas as principais questões que influenciam a modelagem do problema, assim como características e restrições específicas de cada caso.

Song *et al.* (2002) tratam do problema de localização relacionado à distribuição de jornais e periódicos, porém sem a incorporação do problema de produção. Nesse trabalho, os autores apresentaram um método para encontrar a alocação ótima da alocação dos agentes de distribuição de jornal, que são os responsáveis por receber os exemplares das unidades industriais para posterior entrega domiciliar, bem como a roteirização dos veículos de abastecimento desses agentes, de forma a minimizar os custos de transporte. A aplicação deste método possibilitou uma economia de 15% nos custos de abastecimento destes agentes, bem como uma redução de 40% no tempo final da entrega dos jornais aos leitores em comparação à solução manual.

Koschat *et al.* (2003) discutem o recente processo de reformulação da distribuição de revistas realizado na *Times Inc.*, a maior editora de revistas dos EUA. O problema engloba a determinação do seqüenciamento de produção (impressão) e do número de cópias a serem impressas e despachadas, com base em estimativas de custos e receitas. Os autores descrevem sucintamente os modelos analíticos propostos, bem como discutem as dificuldades gerenciais para implementação das soluções.

Nenhum dos trabalhos relacionados à logística de distribuição de jornais e periódicos encontrados na literatura abordou o problema com o foco pretendido neste trabalho. Adicionalmente, embora problemas de localização de instalações, base para a configuração de redes de distribuição, sejam bastante estudados na literatura, não se encontrou nenhuma referência sobre problemas de modelagem de redes de distribuição aplicada ao caso de distribuição de revistas, que tratam simultaneamente de localização, produção e distribuição, principalmente de revistas e periódicos. Portanto trata-se de um assunto original e ainda pouco explorado.

5.3.1 Caracterização do Problema

Existem, no Brasil, quatro grande revistas semanais de informações de circulação nacional: Época, da Editora Globo, Veja, da Editora Abril; IstoÉ, da Editora Três; Carta Capital da Editora Confiança. A Veja e a Época estão entre as 10 maiores revistas semanais de informação do mundo em termos de tiragem e circulação, com cerca de um milhão e 500 mil exemplares distribuídos semanalmente, respectivamente.

Há basicamente dois canais de comercialização e distribuição para as revistas publicadas no Brasil: assinaturas e vendas avulsas em bancas de jornais e outros pontos de venda (livrarias, revistarias, supermercados, etc.).

No caso de assinaturas, o leitor opta por pagar antecipadamente os exemplares para um ano e receber os seus exemplares no endereço que desejar. No caso da venda avulsa, o leitor, quando desejar, vai até um ponto de comercialização, na maioria dos casos uma banca de jornais e revistas, para adquirir um exemplar.

Para cada canal de comercialização existe uma rede de distribuidores. Esses distribuidores, no caso de assinaturas, têm a função de receber o reparte da unidade gráfica, imprimir as etiquetas de identificação dos assinantes, ensacar e etiquetar as revistas e providenciar a entrega nos endereços indicados nestas etiquetas, até um horário pré-determinado. Os distribuidores de venda avulsa, isto é, para entrega em bancas de jornal, desempenham um papel diferente: recebem as revistas das unidades gráficas, separando-as por ponto de venda. No caso de revistas semanais de informações, as mesmas são entregues pelos distribuidores nas bancas e demais pontos de venda, tendo em vista a sua perecibilidade. Para outras revistas de tiragem mensal não há entrega em bancas: os “jornaleiros” retiram os exemplares nos distribuidores, conjuntamente com a devolução do chamado “encalhe”, ou seja, exemplares não vendidos da edição anterior.

Como os objetivos comerciais dos distribuidores são diferentes é comum ter dois distribuidores na mesma praça; um de assinaturas e o outro de venda avulsa.

Em termos de demanda, as revistas semanais de informações são consumidas, ou seja, lidas primordialmente aos finais de semana, sendo consideradas uma opção de lazer, e tendo como

como concorrentes a televisão e os jornais diários. As editoras buscam levar aos leitores as notícias mais completas e atualizadas. Como resultado, há muito pouco tempo, tipicamente poucas horas, entre o fechamento da redação de uma edição e o início de impressão e acabamento dos exemplares. Este intervalo de tempo engloba desde a transmissão dos arquivos digitais para as unidades industriais, passando pela geração de matrizes de impressão, até a impressão dos cadernos e capas, acabamento dos exemplares, paletização e expedição. Similarmente, é muito curto o tempo disponível para o abastecimento dos distribuidores regionais e entrega domiciliar para os leitores, no caso de assinaturas, e disponibilização para os pontos de vendas, no caso de venda avulsa.

A entrega domiciliar a partir dos distribuidores é realizada de maneira bastante pulverizada e rápida, o mesmo ocorrendo com a distribuição para as bancas. Dessa forma, em geral é suficiente garantir os prazos de entrega para os distribuidores a fim de assegurar que os leitores recebam as revistas em tempo adequado, sem a necessidade de considerar explicitamente a distribuição em nível local, a partir dos distribuidores.

Os distribuidores regionais, sejam de assinaturas ou de venda avulsa, recebem seus exemplares via modal aéreo ou rodoviário. Essa decisão é influenciada principalmente pelo tempo de abastecimento. No caso das revistas semanais de informação, o término do abastecimento da rede de distribuidores deve ocorrer, idealmente, no sábado, isto explica a utilização do modal aéreo em vários casos.

O mercado de revistas é muito competitivo, com margens muito reduzidas; portanto a eficiência operacional de uma editora de revistas pode determinar o sucesso de seus negócios. Esta eficiência pode ser traduzida, em linhas gerais, na definição da melhor configuração da rede de distribuição, que compreende desde a determinação da localização/seleção de unidades industriais, até a alocação da demanda, seqüenciamento de produção e escolha do modal de transporte para o abastecimento dos distribuidores regionais, onde o aspecto temporal é fundamental, tendo em vista ser um produto altamente perecível. Ao contrário dos jornais, as revistas têm uma abrangência nacional, tornando seu sistema logístico mais complexo, com a descentralização da produção, normalmente utilizando terceiros, tendo em vista a elevada capacidade de produção necessária por um período de tempo relativamente curto.

5.3.2 Modelagem Matemática

O problema a ser tratado pode ser genericamente descrito como: determinar quais gráficas serão utilizadas para a impressão e o acabamento das revistas, quais as respectivas praças atendidas por cada uma das gráficas, em que ordem produzir os exemplares destinados a cada praça e quais os modais de transporte a serem utilizados, de tal forma a minimizar o custo total, dado pela soma dos custos de produção e de transporte, respeitando-se as restrições de capacidade de produção das unidades gráficas e de nível de serviço associado a cada ponto de demanda.

Considere-se uma rede de distribuição de revistas composta de unidades industriais (gráficas), onde os exemplares são impressos e acabados, e um conjunto de locais de destino (pontos de demanda) que devem receber as revistas. Conforme explicitado acima, cada um desses locais não representa o destino final destas revistas, mas apenas uma praça de destino, a partir da qual a revista será distribuída localmente para os assinantes e bancas. Para cada canal de comercialização – assinaturas ou venda avulsa – existe um conjunto de locais de destino estrategicamente distribuídos pelo país. No caso de revistas com circulação semanal, o horário de chegada dos exemplares de uma determinada edição em cada um desses locais é crucial para assegurar o nível de serviço desejado, influenciando as decisões de onde produzir as revistas para cada destino, em que ordem produzi-las, e qual o modal de transporte utilizar.

Seja $t = 1, 2, \dots, TP$ os períodos de tempo em que o tempo total para produção e distribuição das revistas é discretizado.

Cada unidade industrial $i = 1, 2, \dots, NG$ possui uma capacidade de produção CP_{it} por período t , expressa como o número máximo de exemplares que pode ser produzido nesse período; essa capacidade pode variar ao longo do tempo; em geral, as capacidades são menores nos períodos iniciais de produção, devido à necessidade de *set-up* e ajustes nos equipamentos de impressão e acabamento dos exemplares. Define-se ainda um volume mínimo de produção PM_i , expresso em número de exemplares, caso a unidade industrial i seja selecionada. A cada unidade industrial i está associado um custo unitário de produção CUP_i por exemplar. Os serviços de impressão e acabamento podem ser terceirizados, portanto as unidades industriais são contratadas, ou seja, não se tratam de unidades próprias ou que devam ser construídas.

A cada ponto de demanda $j = 1, 2, \dots, ND$ estão associados uma demanda D_j , que corresponde ao volume de revistas a ser entregue em cada edição, e um horário máximo permitido de chegada das revistas ao mesmo, dado por H_j .

Há $m = 1, 2, \dots, NM$ modais para o transporte dos lotes de revista das gráficas i para os pontos de demanda j . Para cada modal de transporte m são conhecidos os custos de transporte CT_{ijm} de cada lote de uma gráfica i até um ponto de demanda j , bem como o respectivo tempo de trânsito T_{ijm} entre cada par de pontos. Os modais de transporte, (no caso, aéreo e rodoviário), são selecionados em função do tempo de deslocamento e do custo por viagem, ou seja, o modal aéreo, com custos mais altos quando comparados com o modal rodoviário, somente é selecionado quando o horário máximo permitido de chegada em um determinado ponto de demanda j , através do modal rodoviário, não puder ser respeitado.

Define-se ainda um número máximo K de unidades gráficas que podem ser selecionadas, tendo em vista diminuir a complexidade gerencial e operacional de produção, expedição e transporte das revistas a partir de um número muito grande de unidades gráficas.

Definem-se ainda as seguintes hipóteses:

- Para cada gráfica candidata i é conhecido o horário mais cedo em que a produção pode iniciar e o horário mais tarde em que pode terminar; para os períodos de produção t em que a gráfica não esteja disponível para produção as respectivas capacidades de produção CP_{it} são nulas.
- O número de períodos de produção (NP), por unidade gráfica, é suficiente para o atendimento de toda a demanda produzida em cada uma delas.
- Os períodos de produção t têm duração de uma hora; verificou-se que períodos menores não proporcionam benefícios operacionais e de nível de serviço, além de aumentarem o tamanho do modelo matemático e, conseqüentemente, sua dificuldade de resolução, expressa pelo tempo de processamento.
- A capacidade total de produção, isto é, considerando todas as gráficas candidatas, é capaz de atender toda a demanda, isto é, $\sum_{i=1}^{NG} \sum_{t=1}^{NP} CP_{it} > \sum_{j=1}^{ND} D_j$.

- Para cada gráfica i , a capacidade de produção total é maior do que a maior demanda dentre todos os pontos de demanda j , isto é, $\sum_{t=1}^{NP} CP_{it} > \max_{j=1, \dots, ND} D_j$.

A fim de permitir a modelagem do seqüenciamento de produção, bem como as restrições temporais de janela de entrega, define-se o parâmetro $a_{ijmt} = 1$ se as revistas produzidas na gráfica i , no período de produção t , podem atingir o ponto de demanda j em tempo (ou seja, até o instante H_j), utilizando o modal de transporte m , e 0 caso contrário. O parâmetro a_{ijmt} leva em consideração o tempo de trânsito total entre a unidade de industrial i e o ponto de demanda j , incluindo a disponibilidade e os horários de vôos no caso do transporte aéreo.

Definem-se as seguintes variáveis de decisão:

X_{ijmt} = número de exemplares produzidos na gráfica i para o ponto de demanda j no período t , utilizando o modal de transporte m .

Z_i = 1 se a unidade industrial i for selecionada;
= 0 caso contrário.

Y_{ijm} = 1 se a unidade industrial i atende o ponto de demanda j , através do modal m
= 0 caso contrário

A formulação matemática do problema de produção e distribuição de uma revista semanal de informações, como um problema de programação linear inteira-mista, pode ser escrita como:

$$\text{minimizar } \sum_{i=1}^{NG} \sum_{j=1}^{ND} \sum_{m=1}^{NM} \left(CT_{ijm} Y_{ijm} + CUP_i \sum_{t=1}^{NP} X_{ijmt} \right) \quad (5.36)$$

sujeito a :

$$\sum_{i=1}^{NG} \sum_{m=1}^{NM} Y_{ijm} = 1 \quad \text{para } j=1, 2, \dots, ND \quad (5.37)$$

$$\sum_{j=1}^{ND} \sum_{m=1}^{NM} \sum_{t=1}^{NP} X_{ijmt} \geq PM_i Z_i \quad \text{para } i=1, 2, \dots, NG \quad (5.38)$$

$$\sum_{i=1}^{NG} \sum_{m=1}^{NM} \sum_{t=1}^{NP} a_{ijmt} X_{ijmt} = D_j \quad \text{para } j=1, 2, \dots, ND \quad (5.39)$$

$$\sum_{j=1}^{ND} \sum_{m=1}^{NM} X_{ijmt} \leq CP_{it} \quad \text{para } i=1, 2, \dots, NG \text{ e } t=1, 2, \dots, NP \quad (5.40)$$

$$\sum_{t=1}^{NP} X_{ijmt} \leq M Y_{ijm} \quad \text{para } i=1, 2, \dots, NG; j=1, 2, \dots, ND \text{ e } m=1, \dots, NM \quad (5.41)$$

$$\sum_{j=1}^{ND} \sum_{m=1}^{NM} Y_{ijm} \leq M' Z_i \quad \text{para } i=1, 2, \dots, NG; \quad (5.42)$$

$$\sum_{i=1}^{NG} Z_i \leq K \quad \text{para } i=1, 2, \dots, NG \quad (5.43)$$

$$Z_i \in \{0,1\} \quad \text{para } i=1, 2, \dots, NG \quad (5.44)$$

$$Y_{ijm} \in \{0,1\} \quad \text{para } i=1, 2, \dots, NG; j=1, 2, \dots, ND \text{ e } m=1, \dots, NM \quad (5.45)$$

$$X_{ijmt} \geq 0 \quad \text{para todo } i, j, m \text{ e } t \quad (5.46)$$

A função objetivo (5.36) minimiza o custo total de produção e de transporte. A restrição (5.37) garante que cada ponto de demanda seja atendido por exatamente uma única unidade industrial e um único modal de transporte. A restrição (5.38) assegura que o volume mínimo de produção seja respeitado para cada unidade industrial selecionada. A restrição (5.39) garante que toda a demanda seja atendida, enquanto que a restrição (5.40) assegura que as capacidades de produção, nos períodos de produção t , sejam respeitadas. A restrição (5.41) garante que só pode haver fluxo de revistas produzidas em uma unidade industrial para unidades e modais selecionados, sendo que M é uma constante muito grande (igual, por exemplo, à máxima produção em um período). A restrição (5.42) assegura que os pontos de demanda somente possam ser atendidos por unidades industriais selecionadas, sendo M' uma constante suficientemente grande (igual, por exemplo, ao número total de pontos de demanda ND). A restrição (5.43) restringe o número máximo de unidades industriais selecionadas. As restrições (5.44) e (5.45) impõem que as variáveis de decisão Y_{ijm} e Z_i sejam binárias. Por último, a restrição (5.46) indica que as variáveis de decisão X_{ijmt} não podem assumir valores negativos.

5.3.3 Estruturação do Modelo em Ambiente de Planilha Eletrônica

O modelo matemático foi implementado em ambiente de planilha eletrônica Microsoft Excel, utilizando o pacote de otimização *What's Best* (WB), versão 7.0, desenvolvido pela Lindo Systems Inc.

Planilhas eletrônicas são ferramentas que estão se tornando cada vez mais populares nas empresas para modelagem e análise de problemas de natureza quantitativa (Conway e

Ragsdale, 1997). Assim, a motivação é disponibilizar uma ferramenta que possa ser utilizada por um usuário no seu ambiente de trabalho, sem necessidade de dominar uma linguagem de modelagem ou programação específica, permitindo ao mesmo alterar ou atualizar o modelo matemático, sem dificuldade e sem necessitar de um suporte específico.

Por outro lado, existem dificuldades para implementar e modelar um modelo matemático de maior porte (mais de 10.000 variáveis de decisão e 1.000 restrições) no ambiente não-estruturado das planilhas eletrônicas. Inicialmente foi realizado um experimento envolvendo um problema de dimensões reduzidas, de forma a auxiliar na definição do melhor *layout* e arranjo das células, assim como na depuração e validação do modelo matemático em termos das expressões numéricas e fórmulas, assegurando que o mesmo estivesse livre de erros, uma preocupação relevante na modelagem com planilhas (Panko e Sprague Jr., 1998; Teo e Tan, 1999).

Foram consideradas duas diferentes planilhas na pasta de trabalho, uma para os dados de entrada (que podem ser freqüentemente alterados ou atualizados pelo usuário) e outra para o modelo de otimização. Essa organização permitiu manter separados dados e fórmulas, assim como identificar eventuais erros nas fórmulas e nos intervalos (*ranges*), uma das principais causas de erros em planilhas, conforme apontado por Edwards *et al.* (2000). As Figura 5.7 a 5.9 ilustram como os dados de entrada foram organizados, ao passo que as Figura 5.10 a 5.12 fornecem detalhes de como o modelo matemático foi construído e estruturado na planilha.

Foram utilizados recursos de cor, sombreamento e borda, a fim de diferenciar os grupos de células contendo dados, variáveis de decisão, restrições e a função objetivo, assim como auxiliar o usuário a facilmente identificar a solução ótima obtida. Tomou-se cuidado de não embutir valores numéricos constantes nas fórmulas, mas sim armazená-los em células devidamente identificadas.

Conforme pode ser observado nas Figura 5.10 a 5.11, o modelo foi estruturado segundo uma orientação vertical, a fim de tornar a navegação mais fácil para o usuário. Os pontos de atendimentos foram inseridos verticalmente, isto é, segundo uma orientação por linha. Maiores detalhes sobre a estruturação em planilha podem ser encontrados em Cunha e Mutarelli (2006).

The screenshot shows an Excel spreadsheet with two main data tables. The first table, 'Primary Destinations', lists demand points with columns for Code, Primary Destination, State, ID, Weekly Demand, (kgs.), and Deadline for Delivery. The second table, 'Industrial Facilities', lists production units with columns for Code, Industrial Facility, State, ID, Start Time, Minimum Production Volume, Time Period, Time, and Production Capacity. Brackets on the right side of the spreadsheet group these tables under their respective labels.

Code	Primary Destination	State	ID	Weekly Demand	(kgs.)	Deadline for Delivery
j1	Salvador	BA	SSA	19.657	3.931	13,00
j2	Porto Alegre	RS	POA	14.095	2.819	10,00
j3	Fortaleza	CE	FDR	18.150	3.630	19,00
j5	Belo Horizonte	MG	BHZ	10.000	2.000	20,00
				61.902		

Code	Industrial Facility	State	ID	Start Time	Minimum Production Volume	Time Period	Time	Production Capacity
i1	Globo Cochrane	SP	GC	2,00	20.000	T1	2,250	4.375
						T2	2,500	8.750
						T3	2,750	8.750
						T4	3,000	8.750
i2	Quebecor World	PE	QW	3,00	5.000	T1	3,250	3.125
						T2	3,500	6.250
						T3	3,750	6.250
						T4	4,000	6.250
i3	Gráfica Plural	PE		3,00	10.000	T1	3,250	4.375
						T2	3,500	8.750
						T3	3,750	8.750
						T4	4,000	8.750

Figura 5.7: Dados de entrada do modelo matemático referentes aos pontos de demanda j e às unidades industriais i

Fonte: Cunha e Mutarelli (2006)

The screenshot shows an Excel spreadsheet with four data tables. 'Production Costs' shows costs for demand points j1-j5. 'Transportation Costs' shows costs for demand points j1-j5 across industrial facilities i1-i3. 'Total Costs' shows the sum of production and transportation costs. 'Transit Time' shows transit times in hours for demand points j1-j5 across industrial facilities i1-i3. Brackets on the right side of the spreadsheet group these tables under their respective labels.

Code	Primary Destination	i1	i2	i3
j1	Salvador	16.119	18.635	17.455
j2	Porto Alegre	11.558	13.362	12.516
j3	Fortaleza	14.883	17.206	16.117
j5	Belo Horizonte	34.607	40.009	37.477

Code	Primary Destination	i1		i2		i3	
		m1	m2	m1	m2	m1	m2
j1	Salvador	2.382	9.238	1.038	4.717	2.317	9.238
j2	Porto Alegre	1.428	4.764	3.503	12.911	1.131	4.764
j3	Fortaleza	3.521	11.398	993	6.352	3.456	11.398
j5	Belo Horizonte	964	10.466	2.750	17.303	891	10.466

Code	Primary Destination	i1		i2		i3	
		m1	m2	m1	m2	m1	m2
j1	Salvador	18.501	25.357	19.673	23.352	19.772	26.693
j2	Porto Alegre	12.986	16.322	16.865	26.273	13.647	17.280
j3	Fortaleza	18.404	26.281	18.199	23.558	19.573	27.515
j5	Belo Horizonte	35.571	45.073	42.759	57.312	38.368	47.943

Code	Primary Destination	i1		i2		i3	
		m1	m2	m1	m2	m1	m2
j1	Salvador	33,80	5,91	14,60	5,91	32,88	5,91
j2	Porto Alegre	20,17	6,25	60,02	9,25	19,25	6,25
j3	Fortaleza	50,06	9,25	13,97	5,00	49,14	9,25
j5	Belo Horizonte	11,17	4,75	33,77	8,75	10,25	4,75

Figura 5.8: Dados de entrada do modelo matemático referentes a custos e transit times.

Fonte: Cunha e Mutarelli (2006)

Microsoft Excel - WB Protótipo Maior 3003

Arquivo Editar Exibir Inserir Formatar Ferramentas Dados Janela WB! Ajuda

TRANSPORTATION MODE - LEVEL OF SERVICE							
		i1		i2		i3	
Code	Primary Destination	m1	m2	m1	m2	m1	m2
j1	Salvador	NO	YES	NO	YES	NO	YES
j2	Porto Alegre	NO	YES	NO	NO	NO	YES
j3	Fortaleza	NO	YES	YES	YES	NO	YES
j5	Belo Horizonte	YES	YES	NO	YES	YES	YES

Aijm = PARAMETER

		i1					i2					i3				
Code	Primary Destination	t1	t2	t3	t4	t5	t1	t2	t3	t4	t5	t1	t2	t3	t4	t5
j1	Salvador	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
j2	Porto Alegre	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
j3	Fortaleza	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
j5	Belo Horizonte	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Parameter Aijm

Figura 5.9: Dados de entrada do modelo matemático referentes aos pontos de demanda j e às unidades industriais i

Fonte: Cunha e Mutarelli (2006)

Microsoft Excel - WB Protótipo Maior 3003

Arquivo Editar Exibir Inserir Formatar Ferramentas Dados Janela WB! Ajuda

OPTIMIZATION MODEL - WHAT'S BEST

Code	Industrial Facility	Zi
i1	Globo Cochrane	1,0
i2	Ruebecor World	1,0
i3	Gráfica Pleral	1,0

Decision Variable Zi

Constraint (8)

		i1		i2		i3		lim(Yijm)
Code	Primary Destination	m1	m2	m1	m2	m1	m2	
j1	Salvador	-	1,0	-	-	-	-	1
j2	Porto Alegre	-	-	-	-	-	-	1
j3	Fortaleza	-	-	1,0	-	-	-	1
j5	Belo Horizonte	1,0	-	-	-	-	-	1

Constraint Set (2)

Constraint Set (7)

		i1					i2					i3				
Code	Primary Destination	t1	t2	t3	t4	t5	t1	t2	t3	t4	t5	t1	t2	t3	t4	t5
j1	Salvador	-	-	-	-	-	-	-	2.157	8.750	8.750	-	-	-	-	-
j2	Porto Alegre	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
j3	Fortaleza	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
j5	Belo Horizonte	-	3.407	6.593	-	-	-	-	-	-	-	-	-	-	-	-

Decision Variable Yijm

Decision Variable Xijm

		i1					i2					i3				
Code	Primary Destination	t1	t2	t3	t4	t5	t1	t2	t3	t4	t5	t1	t2	t3	t4	t5
j1	Salvador	-	-	-	-	-	-	-	2.157	8.750	8.750	-	-	-	-	-
j2	Porto Alegre	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
j3	Fortaleza	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
j5	Belo Horizonte	-	3.407	6.593	-	-	-	-	-	-	-	-	-	-	-	-

lim(Xijm)

lim(Yijm)

lim(Zi)

Figura 5.10: Parte da planilha do modelo correspondente às variáveis de decisão Z_i e Y_{ijm} e às restrições (5.37), (5.42) e (5.43)

Fonte: Cunha e Mutarelli (2006)

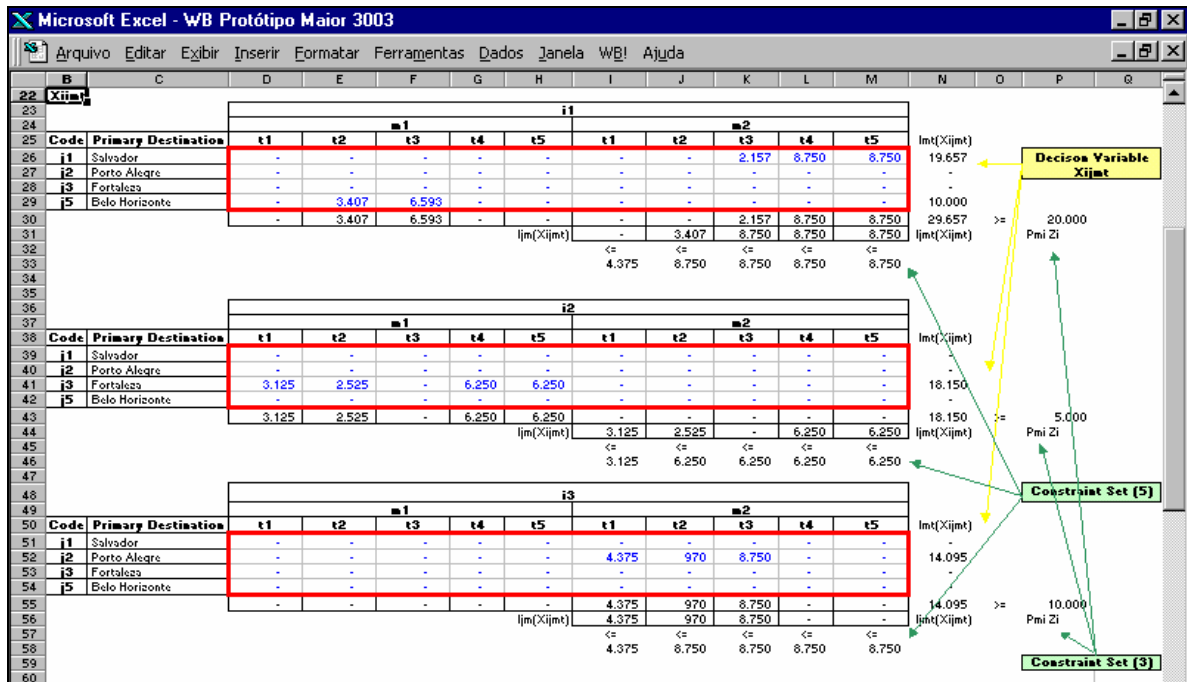


Figura 5.11: Variáveis de decisão X_{ijmt} e restrições (5.38) e (5.40)

Fonte: Cunha e Mutarelli (2006)

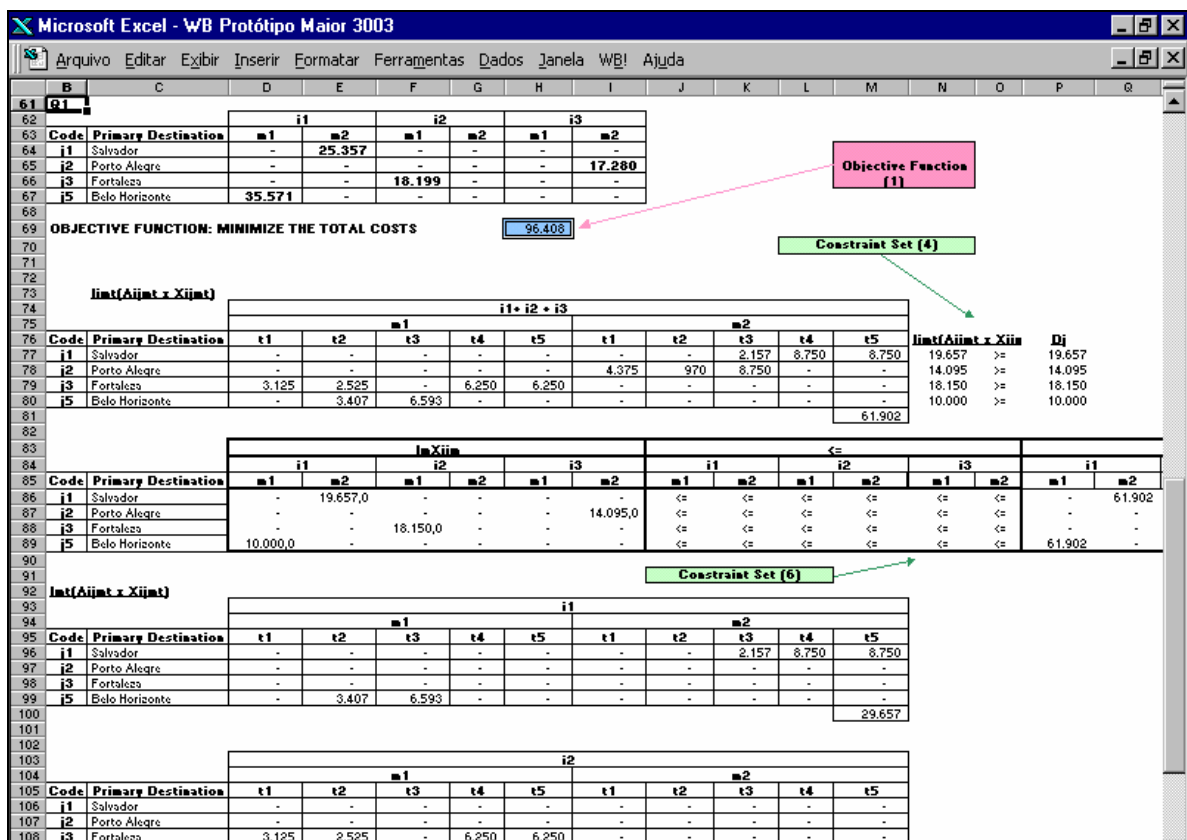


Figura 5.12: Função objetivo e restrições (5.39) e (5.41)

Fonte: Cunha e Mutarelli (2006)

5.3.4 Aplicação a um Problema Real

O problema considerado está relacionado à distribuição da *Época*, revista semanal de informações com circulação nacional e publicada pela Editora Globo.

Atualmente a empresa utiliza duas unidades industriais para a produção da tiragem de cada edição da revista, que é de 484 mil exemplares. A primeira gráfica, denominada Globo Cochrane, está localizada na cidade de Vinhedo, a 80 km da capital do Estado de São Paulo. A segunda, denominada Gráfica Plural, localiza-se na cidade de Santana de Parnaíba, a 22 km da capital do Estado de São Paulo. Cada uma delas produz, em média, metade desta tiragem, ou seja, aproximadamente 240 mil exemplares. Não há critério pré-estabelecido para a determinação da abrangência de cada uma delas.

Os pontos de demanda, conforme definido anteriormente, representam os locais de abastecimento da revista para posterior entrega aos leitores, no caso de assinaturas, e aos pontos de venda, no caso de venda avulsa. No Brasil, há 79 pontos de demanda, divididos entre pontos de assinaturas e pontos de venda avulsa, estrategicamente localizados. Para cada ponto de demanda estão associados um nível de serviço, que representa o horário máximo de chegada das revistas nas regionais de distribuição, e uma quantidade de exemplares que deve ser entregue aos leitores. O horário máximo permitido de chegada da revista nas regionais de distribuição foi determinado em função do horário, pré-determinado, de conclusão da entrega domiciliar para os assinantes, e do horário, pré-determinado, de conclusão do abastecimento de todos os pontos de venda de uma determinada praça. A quantidade de exemplares, por ponto de demanda, foi calculada pela média das 52 edições distribuídas em 2002.

Levantou-se a oferta de unidades gráficas em todo o território nacional para a prestação de serviços de impressão e acabamento de revistas. Cada unidade candidata foi minuciosamente avaliada, não apenas sob a ótica de custos de produção, mas também foram considerados aspectos qualitativos tais como a proximidade de aeroportos que operam com aeronaves cargueiras, o abastecimento de bobinas de papel, disponibilidade de mão-de-obra especializada, capacidade de expansão, dentre outros.

Além das duas gráficas atualmente utilizadas, foram identificadas mais quatro possíveis alternativas de produção: (i) Gráfica Minister, localizada na cidade do Rio de Janeiro; (ii) Gráfica Santa Marta, em João Pessoa (PB); (iii) Quebecor World, em Ipojuca (PE), a cerca de 80 km da capital do Estado; e (iv) Posigraf, em Curitiba (PR).

O modelo matemático proposto contempla 6 unidades gráficas candidatas, 79 praças de destino, 2 modais de transporte (rodoviário e aéreo) e 10 períodos de produção, totalizando 10.434 variáveis de decisão, sendo 954 variáveis binárias, e 1.178 restrições. Todos os dados utilizados estão detalhados em Mutarelli (2004).

Foram analisadas 4 alternativas, cujos resultados são apresentados na Tabela 5.4:

- *Alternativa 1*: otimização do esquema de produção atual, isto é, considerando apenas as duas unidades atualmente utilizadas para a produção, e considerando os dados atuais de produção, demanda, transporte e nível de serviço da revista;
- *Alternativa 2*: otimização do esquema de produção, considerando todas as gráficas candidatas, sem limitação no número máximo de gráficas selecionadas ($K = NG$) e dados atuais de demanda e nível de serviço;
- *Alternativa 3*: otimização do esquema de produção, considerando todas as gráficas candidatas, limitando a três o número máximo de gráficas selecionadas ($K = 3$) e dados atuais de demanda e nível de serviço;
- *Alternativa 4*: semelhante à Alternativa 3, porém limitando a dois o número máximo de gráficas selecionadas ($K = 2$);

A Alternativa 1 representa a otimização da situação atual, em que são utilizadas apenas as gráficas Globo Cochrane e Plural. Percebe-se, neste caso, que a Globo Cochrane teve um aumento de 43% em relação à produção atual, atendendo 59 praças, ou seja um aumento de 79% sobre o número atual de praças atendidas. Este aumento, tanto no número de exemplares quanto no número de praças atendidas, é explicado pelo menor custo unitário de produção na Globo Cochrane em relação à Gráfica Plural (4,9%). Os resultados são pouco representativos, em termos de redução do custo total de produção e de transporte, passando de R\$ 1,010 por exemplar para R\$ 0,994 por exemplar. Isso permite concluir que a configuração atual da rede de distribuição da revista *Época* está, de certa forma, racionalizada, se forem consideradas apenas as gráficas próprias.

Tabela 5.4: Resumo das Alternativas Propostas e da Situação Atual

Item	Indicador	Un	Situação Atual	Alternativas			
				1	2	3	4
				2 U.Inds.	-	$K \leq 3$	$K \leq 2$
1	Custo de Produção	R\$	407.365,36	403.528,04	409.223,64	409.223,64	403.277,88
2	Custo de Papel	R\$	232.342,18	232.342,08	235.233,60	235.233,60	232.342,08
3	Custo de Impressão	R\$	175.023,18	171.185,96	173.990,04	173.990,04	170.935,80
4	Custo de Transporte	R\$	79.101,29	77.844,76	42.707,26	42.707,26	76.870,70
5	Modal Aéreo	R\$	67.124,92	70.498,00	33.593,26	33.593,26	70.827,80
6	Modal Rodoviário	R\$	11.976,37	7.346,76	9.114,00	9.114,00	6.042,90
7	Custo Fixo de Produção	R\$	-	-	-	-	-
8	Custo Total (1+4)	R\$	486.466,65	481.372,80	451.930,90	451.930,90	480.148,58
9	Custo Total + Custo Fixo (1+4+7)	R\$	486.466,65	481.372,80	451.930,90	451.930,90	480.148,58
10	Custo Unitário por Exemplar (8/16)	R\$ / ex.	1,010	0,994	0,934	0,934	0,992
11	Nº Praças - Lançamento Sábado	un	43	43	43	43	43
12	Nº Praças - Lançamento Domingo	un	36	36	36	36	36
13	Número Total de Praças	un	79	79	79	79	79
14	Nº Exemplares - Lançamento Sábado	exs.	371.648	371.648	371.648	371.648	371.648
15	Nº Exemplares - Lançamento Domingo	exs.	112.398	112.398	112.398	112.398	112.398
16	Número de Exemplares Produzidos	exs.	484.046	484.046	484.046	484.046	484.046
17	Tempo de Processamento - What's Best	-	-	00h05m03s	01h29m29s	00h08m28s	02h32m51s

Na Alternativa 2, consideraram-se, além da Globo Cochrane e da Gráfica Plural, outras quatro localidades candidatas a sediar unidades gráficas. A solução obtida resulta em três unidades gráficas selecionadas: Globo Cochrane (Vinhedo, SP), Santa Marta (João Pessoa, PB) e Minister (Rio de Janeiro, RJ). O resultado merece alguns comentários quanto ao tempo de processamento do WB. Observou-se um tempo de processamento significativamente mais elevado, decorrente da provável dificuldade do algoritmo de solução de *branch-and-bound* do WB para eliminar parte do espaço de soluções a ser enumerado.

Na Alternativa 3, em que se considerou uma limitação do número de unidades gráficas na solução do problema, ou seja, K menor ou igual a 3, obteve-se a mesma solução que a obtida na Alternativa 2, ou seja, a solução ótima para o problema. Deve-se destacar que o tempo de processamento do What's Best foi sensivelmente reduzido em relação à Alternativa 2.

Já na quarta alternativa, definiu-se como 2 o número máximo de instalações selecionadas ($K \leq 2$). Percebe-se que o custo unitário total teve um acréscimo de 6,21%, quando comparado as Alternativa 2 e 3, principalmente pelo aumento da utilização do modal aéreo na Globo Cochrane; este aumento pôde ser compensado pelo custo unitário de produção, que é 4,88% menor do que na Gráfica Minister. Em comparação com a situação atual, nessa solução

é utilizada a Gráfica Minister no lugar da Plural.

Para o caso estudado, a solução ótima, ou seja, a alternativa mais indicada para a revisão da rede de produção e de distribuição desta revista semanal de informações, corresponde às Alternativas 2 e 3, em que a mesma solução foi obtida, proporcionando uma economia aproximada de 7,1% no custo total.

Os resultados apresentados ainda podem ser melhorados, principalmente do ponto de vista do nível de serviço, na medida em que a programação da produção seja reordenada, ou seja, considerar, na modelagem matemática, a antecipação da expedição das viagens quando possível. Deve-se salientar, entretanto, que as soluções obtidas são melhores, tanto do ponto de vista de custos quanto de nível de serviço, do que a solução atualmente praticada pela Editora Globo.

5.3.5 Considerações Finais

O modelo proposto, de programação linear inteiro-mista, apresenta uma característica original em relação aos modelos dessa natureza encontrados na literatura, pois permite considerar aspectos do seqüenciamento de produção de maneira integrada com as decisões de transporte e de distribuição. São considerados aspectos temporais, de capacidade de produção das unidades industriais e de escolha do modal de transporte, que são essenciais para a competitividade e sobrevivência de editoras de revistas. Busca-se preencher uma lacuna, que é a falta de ferramenta adequada para a abordagem deste tipo de problema pelas editoras no Brasil.

Seguindo tendências mais recentes, o modelo foi implementado em ambiente de planilha eletrônica, utilizando o software *What's Best*, e aplicado com sucesso ao caso real da Época; segunda maior revista semanal de informações no Brasil em termos de número de exemplares produzidos e distribuídos.

A revisão da literatura permitiu entender claramente a dimensão deste tipo de problema, suas abordagens e limitações. Embora este seja um problema logístico relevante, não foram encontradas, nesta revisão, referências sobre o tema. No Brasil, algumas revistas semanais de informações estão entre as dez maiores do mundo em termos de circulação, isto é, do número de exemplares produzidos e distribuídos semanalmente. A dimensão temporal, ou seja, o

reduzido intervalo de tempo entre o fechamento de uma edição da revista e a entrega dos exemplares às bancas e aos assinantes, forçou o tratamento simultâneo dos problemas de produção e de distribuição, o que também é inédito.

O modelo matemático permitiu resolver um problema real, de uma revista semanal de informações com circulação nacional, encontrando solução ótima global, e proporcionando uma economia de 7,1% nos custos de produção e de transporte, em uma situação já racionalizada e sem prejuízo no nível de serviço atual. O modelo permite ainda outros estudos e análises de forma a avaliar os impactos nos custos de alterações nos níveis de serviço atualmente praticados.

A implementação do modelo em ambiente de planilha eletrônica Microsoft Excel apresenta a grande virtude de permitir que o mesmo seja facilmente utilizado e assimilado, no dia-a-dia do ambiente empresarial, para análises e decisões diversas, por pessoal sem conhecimento específico e aprofundado de Pesquisa Operacional (PO), nem tampouco que precise dominar uma linguagem específica para operar um pacote especializado de otimização como o CPLEX ou o GAMS. Isso é particularmente importante no caso do Brasil, em que em geral as empresas não dispõem de pessoal especializado em PO nas suas equipes de Tecnologia da Informação, ao contrário do que ocorrem em outros países mais avançados.

Uma possível extensão do presente trabalho consistiria em considerar e aprimorar o tratamento dado à produção e expedição das revistas, tendo em vista o atendimento do nível de serviço oferecido aos clientes e as restrições de produção. Este aprimoramento, que não alteraria a solução ótima obtida nem tampouco a configuração da rede de distribuição, pode ser entendido como o melhor seqüenciamento da produção dos exemplares, de forma a expedir os repartes, para os distribuidores, o mais cedo possível. A antecipação da expedição dos exemplares, quando possível, permite que a conclusão da distribuição, para os leitores, ocorra mais cedo, um diferencial competitivo. Isso pode ser feito através de uma rotina que rearranje a seqüência de produção em cada unidade gráfica após o processamento do modelo matemático.

Outra extensão possível é avaliar a produção simultânea de duas ou mais revistas, com circulações diferentes, volumes diferentes e níveis de serviço diferentes, otimizando a produção e o transporte de abastecimento dos distribuidores regionais.

5.4 CONCLUSÕES DO CAPÍTULO

Este capítulo tratou do problema de projeto ou configuração de redes logísticas, um dos problemas estratégicos mais relevantes enfrentados pelas empresas, e que geralmente envolve investimentos vultosos e que apresenta conseqüências no médio e longo prazo, justificando a importância de se buscarem as melhores decisões em termos de custos e nível de serviço.

Inicialmente foram apresentados os conceitos básicos e os principais modelos matemáticos da teoria de localização de instalações, que formam a base para o projeto de redes logísticas. Em seguida, foram descritas duas pesquisas de mestrado orientadas pelo autor e relacionadas à configuração de redes logísticas: o problema de configuração de rede para uma transportadora de carga parcelada e o problema de configuração de uma rede de produção e distribuição para uma editora de revistas.

Além das possibilidades específicas de continuidade das duas pesquisas, descritas anteriormente, existem muitas outras oportunidades de pesquisas futuras relacionadas ao tema de projeto de redes logísticas. Dada a natureza combinatória do problema, um dos grandes desafios é buscar algoritmos de solução mais eficientes e rápidos, que permitam tratar problemas de maior porte e mais complexos em termos do número de produtos, de níveis ou camadas, de instalações candidatas e de pontos de demanda; em outras palavras, modelos mais realistas em termos da realidade que buscam representar. Em alguns casos é necessária uma significativa agregação de dados para permitir resolver algumas instâncias de problemas reais encontrados na prática.

Nesse sentido, e considerando mais especificamente a realidade brasileira, a modelagem de problemas de localização em geral, e de redes logísticas em particular, deve buscar incorporar aspectos fiscais e tributários que afetam decisivamente não só a localização de instalações produtivas, como também a configuração de redes logísticas. Nesse sentido, o trabalho de Yoshizaki (2002) é uma boa referência inicial sobre o assunto.

Um outro ponto importante não considerado na modelagem de redes logísticas é a questão do número de instalações *versus* os volumes de estoques de segurança necessários, um “*trade-off*” logístico clássico: sabe-se que quanto maior o número de instalações selecionadas (e, conseqüentemente, menores os volumes movimentados em cada instalação), maior é o estoque de segurança total necessário para assegurar um dado nível de serviço e vice-versa.

Portanto, o custo de estoque de segurança é um componente importante para a otimização do custo total, principalmente em problemas que envolvem produtos de alto valor agregado (por exemplo, eletrodomésticos, eletrônicos, farmacêuticos, etc.), embora quase nunca considerado nos modelos encontrados na literatura. Isso possivelmente decorra do fato de que não é muito simples modelar o custo de estoque de segurança, uma vez que não se trata de um custo unitário a mais a ser adicionado ao modelo matemático: em geral o estoque de segurança é modelado como uma função não linear do volume total movimentado na instalação; adicionalmente, requer dados históricos para a determinação dessa função. Nesse contexto, Robinson Jr. *et al.* (1993) descrevem uma interessante aplicação prática de modelagem de rede de distribuição relacionada à fusão de duas empresas em que se considera explicitamente o custo do estoque de segurança na modelagem.

Nos modelos de localização em geral assume-se que cada um dos pontos de demanda é atendido individualmente diretamente a partir de uma instalação. Existem situações em que essa hipótese não é adequada, sendo necessário levar em consideração aspectos relacionados à roteirização dos veículos que fazem as entregas (Current *et al.*, 2002), dando origem aos chamados problemas de localização-roteirização (do inglês “*location-routing problems*”). Tais problemas, ainda mais complexos e de difícil modelagem e solução, proporcionam uma excelente oportunidade para desenvolvimento de novas estratégias de solução inspiradas em técnicas como algoritmos genéticos, busca tabu, GRASP e busca em vizinhança de grande porte, entre outras. Modelagens mais realistas de redes logísticas podem vir a incorporar ainda mais detalhes sobre atendimento da demanda, tais como tamanhos e frequências de veículos, entre outros.

Variações e incertezas relacionada aos parâmetros de entrada para os modelos (demandas, custos de transporte e das instalações, etc.), bem como efeitos dinâmicos que podem provocar filas e congestionamentos, são outras das grandes questões que envolvem a modelagem de problemas de natureza estratégica, entre os quais os a localização de instalações, para as quais existem duas linhas básicas de abordagem (Current *et al.*, 2002): (i) através de modelos de localização dinâmicos, em que se considera um horizonte de tempo e as variações que ocorrem ao longo desse horizonte, sendo que as decisões envolvem não só onde e quais instalações abrir ou fechar, mas também quando fazê-lo; (ii) através de modelos estocásticos, que consideram explicitamente incertezas na modelagem. Em ambos os casos, a complexidade da modelagem dificulta a modelagem de problemas reais encontrados na

prática, abrindo oportunidades para pesquisa de novas alternativas de modelagem e novos métodos de solução.

Outra área de potencial interesse de pesquisa é a modelagem de problemas de localização e de configuração de redes com objetivos múltiplos (e às vezes, conflitantes) ou hierárquicos. Nesse sentido, o trabalho de Current *et al.* (2002) serve como uma boa referência inicial sobre o assunto. Estratégias de solução baseadas em população, tais como Algoritmos Genéticos, podem representar um caminho interessante nessa linha, uma vez que permitem obter um conjunto de soluções de elite (i.e., de boa qualidade) no tocante a mais de um critério, que podem ser exploradas a fim de determinar uma envoltória de soluções dominantes que podem vir a serem analisadas individualmente.

A integração de modelos de otimização com Sistemas de Informação Geográficas (SIG) é outra possibilidade a ser explorada para facilitar o uso de tal ferramental pelas empresas.

Capítulo**6****MODELAGEM DE PROBLEMAS
FERROVIÁRIOS**

Este capítulo trata da modelagem matemática de problemas de planejamento e operação do transporte ferroviário de carga. Seu conteúdo baseia-se particularmente nos trabalhos de Ahuja, Cunha e Sahin (2005) e Sahin, Ahuja e Cunha (2005). Ambos tiveram participação importante deste autor, no âmbito da pesquisa em nível de pós-doutorado desenvolvida pelo mesmo no *Department of Industrial and Systems Engineering* da Universidade da Flórida, com o Prof. Ravindra K. Ahuja, entre agosto de 2003 e abril de 2004.

6.1 RELEVÂNCIA E MOTIVAÇÃO

É inquestionável o importante papel que as ferrovias exercem no contexto da logística e do transporte de carga, movimentando grandes quantidades de cargas e provendo serviços bastante competitivos em termos de eficiência operacional e de custos. Dada a sua complexidade e abrangência, o transporte ferroviário também proporciona grandes desafios em termos de modelagem e de aplicação de técnicas de pesquisa operacional.

Um aspecto importante que nem sempre é considerado é que as ferrovias enfrentam uma diversidade de problemas e decisões relacionadas à oferta de transporte, nos diferentes níveis (estratégico, tático e operacional), que podem ser considerados muito mais complexos e abrangentes que os encontrados em outros modais, como, por exemplo, o rodoviário e o aéreo. Uma diferença capital é que, ao contrário de outros modais, as ferrovias são responsáveis por construir, operar e manter toda a infra-estrutura relacionada à sua operação:

as vias férreas por onde trafegam os trens; as estações e os pátios ferroviários onde vagões são carregados e descarregados, trens são formados e desmembrados e podem trocar vagões; a operação e controle do tráfego dos trens em tempo real, incluindo sinalização. Há um número muito grande de decisões envolvidas, usualmente inter-relacionadas e intrinsecamente complexas englobando, entre outras:

- *decisões estratégicas e de investimento de capital*, tais como (i) ampliação da malha ferroviária, implicando a construção de novos trechos, a duplicação e a melhoria de traçado de trechos existentes, o aumento de capacidade; (ii) ampliação, construção e desativação de estações e pátios de formação de trens e de cruzamento; (iii) aquisição e renovação de material rodante (locomotivas e vagões), bem como outros equipamentos; (iv) modernização de sinalização e controle de tráfego; (v) rastreamento de veículos, etc.
- *táticas*, relacionadas (i) à programação de trens e de serviços a serem oferecidos; (ii) à alocação de locomotivas e vagões; (iii) à programação da manutenção da via permanente; (iv) idem para o material rodante, em particular as locomotivas, etc.
- *operacionais*, envolvendo (i) o despacho de trens em tempo real; (ii) a operação dos pátios de formação e recepção de trens; (iii) a circulação de vagões vazios, (iv) a alocação de locomotivas e vagões; (v) a programação de tripulações e pessoal de operação de pátios, de manutenção, etc.

Outros modos de transporte, em particular o transporte aéreo, que tem se beneficiado significativamente do uso de modelos de otimização (Yu, 1998), não apresentam o mesmo nível de complexidade que as ferrovias em termos do seu ambiente operacional. Os problemas de planejamento e programação enfrentados pelas ferrovias são muito complexos. A título de ilustração, uma ferrovia de carga típica nos Estados Unidos possui mais de 16 mil quilômetros de vias por onde trafegam alguns milhares de trens semanalmente, sendo tracionados por 2 a 3 mil locomotivas, movimentando cerca de 50 mil vagões, e passando por 200 a 300 pátios. A malha ferroviária norte-americana compreende mais de 220 mil quilômetros de trilhos, como mostrado na Figura 6.1, e sete operadoras ferroviárias principais: *Burlington Northern Santa Fe* (BNSF), *Canadian National* (CN), *Canadian Pacific* (CP), *CSX*, *Ferrocarril Mexicano* (FXE), *Kansas City Southern* (KCS), *Norfolk Southern* (NS), *Transportacion Ferroviaria Mexicana* (TFM) e *Union Pacific* (UP).

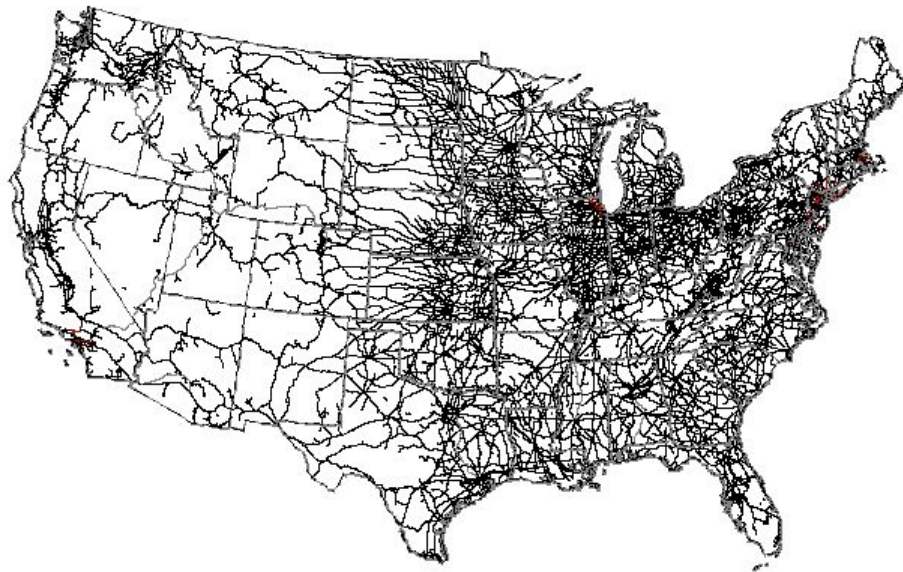


Figura 6.1: Malha ferroviária dos EUA

Fonte: Ahuja *et al.* (2005)

Já a malha ferroviária brasileira é cerca de oito vezes menor que a americana, totalizando cerca de 28.600 km para um país com dimensões territoriais semelhantes aos EUA. A malha foi privatizada na segunda metade da década de 90, sendo atualmente operada por 8 diferentes concessionárias, conforme ilustrado na Figura 6.2.

Outro aspecto complicador no caso de algumas ferrovias brasileiras são as diferenças de bitola, conforme pode ser observado também no Quadro 6.1, que prejudicam a conectividade da malha e requerem operações especiais, tais como a de troca de truques dos vagões e a substituição das locomotivas de tração.

Do lado da demanda, uma grande ferrovia americana possui cerca de 5 mil clientes, que podem gerar entre 4 a 7 mil cargas a serem transportadas semanalmente, envolvendo algo como mil origens e 2 mil destinos diferentes.

Algumas das ferrovias brasileiras apresentam características muito distintas das ferrovias americanas, na medida em que o transporte está concentrado em um número muito reduzido de produtos, quantidades elevadas, entre alguns poucos pares origem-destino.

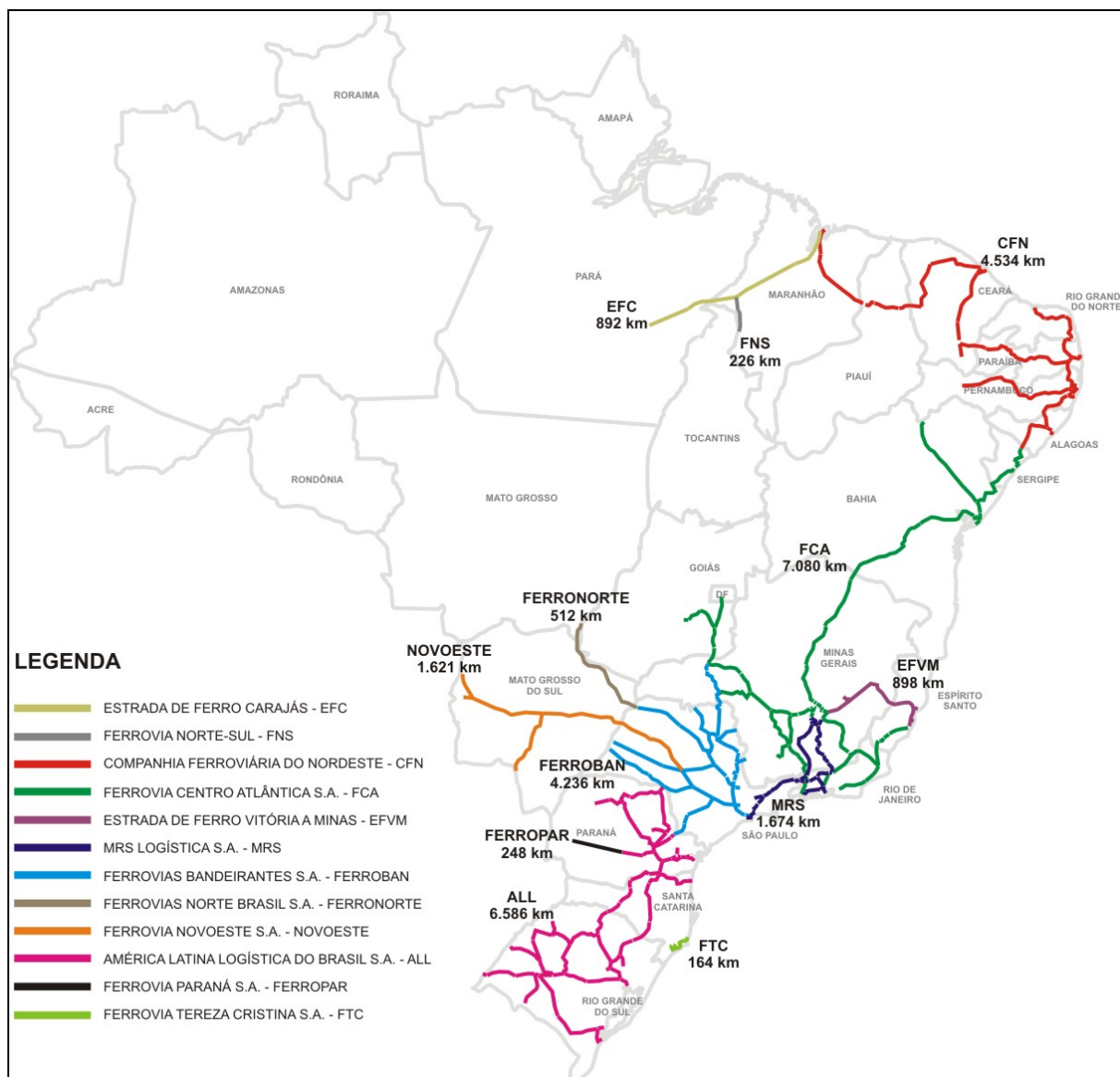


Figura 6.2: Malha ferroviária brasileira e concessionárias

Fonte: ANTF – Associação Nacional dos Transportadores Ferroviários

Por exemplo, as ferrovias de maior produtividade de transporte, como a Estrada de Ferro Carajás, a Estrada de Ferro Vitória-Minas (ambas pertencentes à CVRD – Companhia Vale do Rio Doce) e a MRS Logística, são altamente especializadas no transporte de minério de ferro, operando trens unitários muito longos (em alguns casos podendo atingir mais de 300 vagões), com alta frequência, numa operação regular em que se busca maximizar a quantidade total transportada e a utilização dos recursos.

Quadro 6.1: Concessionárias Ferroviárias no Brasil e respectivas extensões de malha

CONCESSIONÁRIA	BITOLA			
	Larga	Métrica	Mista	Total
FCA – Ferrovia Centro-Atlântica S. A.	182	6.898		7.080
ALL – América Latina Logística do Brasil S.A.	11	6.575		6.586
CFN – Companhia Ferroviária do Nordeste S.A.		4.516	18	4.534
FERROBAN – Ferrovias Bandeirantes S.A.	1.513	2.422	301	4.236
MRS – MRS Logística S.A.	1.632		42	1.674
NOVOESTE – Ferrovia Novoeste S. A.		1.621		1.621
EFVM – Estrada de Ferro Vitória a Minas		898		898
EFC – Estrada de Ferro Carajás	892			892
FERRONORTE – Ferrovias Norte do Brasil	512			512
FERROESTE – Estrada de Ferro Paraná Oeste		248		248
VALEC Engenharia, Construções e Ferrovias S.A.	226			226
FTC – Ferrovia Tereza Cristina S.A.		164		164
TOTAL	4.968	23.342	361	28.671

Fonte: ANTT – Agência Nacional de Transportes Terrestres

Mesmo as ferrovias menos especializadas, como por exemplo a Ferrobán e a ALL, têm como característica importante buscar cargas que permitam trens unitários, de uma única origem para um único destino. A demanda de transporte ferroviário é para cargas tipicamente ferroviárias, isto é, produtos de baixo valor agregado, grandes volumes a grandes distâncias; cargas mais segmentadas em termos de quantidades por cliente, muitas origens e destinos ainda são incipientes no transporte ferroviário no Brasil.

As mudanças observadas no cenário econômico produtivo mundial, com a globalização e a crescente competição entre empresas por mercados mundiais têm levado as empresas ferroviárias, de um modo geral, a buscar reduzir custos e prazos, bem como melhorar a confiabilidade dos serviços prestados e do nível de serviço. Medidas urgentes são necessárias para que as empresas ferroviárias não só consigam conquistar novos clientes, como também manter os clientes atuais, evitando perdê-los para as empresas de transporte rodoviário devido a serviços de baixa qualidade, tempos de viagens irregulares e fretes pouco competitivos, e ainda, buscar aumentar sua lucratividade. Mesmo no caso brasileiro, as ferrovias enxergam o potencial do transporte de carga geral e de contêineres, tendo em vista a maior rentabilidade desse transporte, e têm planos de ampliar suas operações para buscar atrair clientes nessas categorias.

Esse cenário geral em que estão inseridas as ferrovias pode ser considerado muito mais complexo que o de uma empresa de transporte rodoviário de carga ou de uma companhia aérea. Em decorrência, as ferrovias necessitam dramaticamente de ferramentas de apoio a decisões de planejamento e programação que permitam a elas buscar maior eficiência e rentabilidade.

Por outro lado, apesar do crescente número de artigos científicos que abordam técnicas de pesquisa operacional aplicadas a problemas reais nas áreas de transporte e de logística, o setor ferroviário, em particular o de transporte de carga, não vem se beneficiando desse avanço. A quase totalidade dos problemas que envolvem decisões de planejamento e programação enfrentados pelas ferrovias, os quais podem representar valores bastante significativos, vem sendo resolvida manualmente, sem o apoio de modelos quantitativos de apoio à decisão. A pesquisa avança vagarosamente; contribuições recentes encontradas na literatura propõem modelos simplificados, pouco realistas, ou aplicados a problemas de dimensões muito menores que as encontradas na prática. Um dos fatores que contribui para essa situação é a abrangência e complexidade desses problemas, que dificulta sobremaneira a sua modelagem matemática de forma a capturar adequadamente e de maneira realista todas as restrições envolvidas. Essa situação vem gradualmente sendo revertida. Recentemente, a ferrovia *Canadian Pacific Railways* foi escolhida como a vencedora do *2003 Franz Edelman Award for Achievement in Operations Research and Management Sciences* pelo trabalho intitulado “*Perfecting the Scheduled Railroad: Model-Driven Operating Plan Development*” (Ireland *et al.*, 2004). Nesse trabalho, foram desenvolvidos sistemas de apoio à decisão baseados em modelos de pesquisa operacional, a fim de que a ferrovia pudesse operar de maneira mais programada, e que possibilitou obter economias da ordem de 170 milhões de dólares entre 1999 e 2000.

Neste capítulo serão apresentados modelos matemáticos e algoritmos de solução que permitem tratar alguns problemas importantes enfrentados pelas ferrovias, apoiados no trabalho de Ahuja, Cunha e Sahin (2005). Mais especificamente serão considerados os problemas de despacho de trens, de programação de locomotivas e de blocagem de trens. São detalhadas estratégias de solução baseadas em técnicas avançadas que envolvem programação linear e inteira, modelos de fluxo em rede, otimização discreta, e sofisticadas heurísticas, que permitem tratar e resolver instâncias de problemas reais encontrados na prática. Esses problemas são formulados como modelos de programação inteira de grande porte, podendo conter um número muito elevado de variáveis de decisão (centenas de milhares a milhões).

6.2 O PROBLEMA DE DESPACHO DE TRENS

O problema de *despacho de trens* tem por objetivo determinar a trajetória detalhada de um conjunto de trens em uma malha ferroviária, em termos de seus deslocamentos e seus respectivos horários, levando em consideração várias restrições e condicionantes operacionais. O objetivo é minimizar os atrasos ou desvios na programação dos trens em decorrência de conflitos de cruzamento e ultrapassagem que ocorrem e que obrigam algum dos trens envolvidos a dar passagem para o outro trem.

Assume-se ser conhecida a programação de cada trem, isto é, as estações de origem e de destino e os respectivos horários de partida e de chegada, assim como os tempos de viagem nos trechos que compõem o seu percurso.

Esse problema é particularmente relevante em trechos da malha com via singela, isto é, vias com trens trafegando em ambos os sentidos, obrigando um dos trens a dar passagem para cruzamento com outro trem que trafega em sentido contrário, ou ultrapassagem para trens trafegando no mesmo sentido. Trata-se de um problema bastante relevante, uma vez que a parada de um dos trens envolvidos em algum conflito acaba ocasionando atrasos, o que afeta não só o *transit-time* das cargas, como a produtividade de locomotivas e vagões.

No caso mais geral, a malha de interesse pode ser composta de uma ou mais linhas férreas interligando estações principais. Cada linha pode englobar vários trechos de via singela e/ou via dupla entre estações intermediárias (terminais ou pontos de junção). Um trecho (ou seção) de via singela é definido como um segmento que pode acomodar um único trem de cada vez. Analogamente, um trecho de linha dupla pode acomodar dois trens simultaneamente, trafegando tanto no mesmo sentido quando em sentidos opostos. O trecho de via é delimitado, nos seus pontos extremos, por uma estação intermediária ou um pátio de cruzamento. Um pátio de cruzamento pode acomodar pelo menos um trem parado aguardando para cruzamento ou ultrapassagem. Uma estação intermediária pode servir como pátio de formação/decomposição de trens, bem como ponto de parada para cruzamento e ultrapassagem. A Figura 6.3 ilustra um exemplo de malha ferroviária, sendo que os pontos A, C, E e G correspondem a estações principais. Os trechos B-C, C-D e C-F são de via dupla e os demais trechos de via singela. Pátios de cruzamento estão localizados entre as estações A e B, D e E, e F e G.

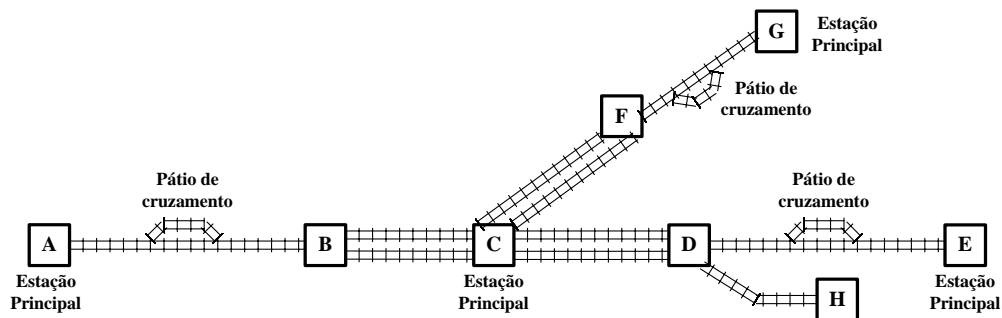


Figura 6.3: Parte de uma malha ferroviária com trechos de via singela e dupla

Fonte: Ahuja *et al.* (2005)

A programação detalhada de despacho dos trens consiste de um quadro horário com os horários de partida e de chegada de cada trem em cada trecho da sua rota, desde a estação de origem até a estação de destino; todas as restrições e condicionantes que regulam o tráfego de trens devem ser consideradas, em particular a capacidade dos trechos, de maneira a assegurar que todos os cruzamentos e ultrapassagens ocorram nos pátios de cruzamento ou em estações intermediárias com espaço suficiente para acomodação dos trens. Assim, a programação de despacho compreende detalhes de todos os trens que são atrasados devido a conflitos com outros trens, assim como os locais e horários de parada.

Além das restrições operacionais ligadas à circulação dos trens, outras restrições práticas podem ocorrer. Alguns trens podem não parar em algum pátio de cruzamento devido à extensão insuficiente de linha para acomodar o trem ou ao tipo de carga sendo transportada (por exemplo, um trem transportando produtos perigosos pode não ser autorizado a parar em um pátio localizado numa área com população na vizinhança); da mesma forma, alguns trens podem não parar em algum pátio, pelo fato do mesmo se situar em trecho de aclive e o trem não ter capacidade de tração para partir uma vez parado; ou ainda, a falta de infra-estrutura adequada para a tripulação pode limitar o tempo de parada de um trem em um pátio de cruzamento. Outra restrição importante é o atraso máximo admissível de cada trem (por exemplo, um trem de carga geral ou contêineres com garantia de hora marcada). Além disso, em algumas situações, a regularidade de partida e/ou chegada de trens em termos de intervalos entre trens consecutivos pode ser tão importante quanto os atrasos, de forma a evitar que trens sejam priorizados nos conflitos em detrimento de outros para depois ficarem aguardando em fila para carga ou descarga, quando poderiam ser parados ao longo do percurso sem maiores conseqüências para a produtividade da operação. Outra restrição encontrada na prática é que trens de um mesmo tipo de uma mesma origem para um mesmo

destino devem manter a seqüência, isto é, o trem que vem atrás não pode ultrapassar o trem que segue na frente.

O problema de despacho de trens pode ocorrer em diferentes contextos. No controle da operação em tempo real, busca-se auxiliar os controladores de tráfego a decidirem quais trens parar e onde, em decorrência de conflitos que vão surgindo à medida que os trens vão avançando e desvios nas programações vão ocorrendo. Em nível tático, o problema consiste em determinar a melhor grade ou programação dos trens, em termos de horários, a fim de minimizar os atrasos. Já no planejamento estratégico, esse problema está relacionado a decisões de investimentos em termos de ampliação da malha ferroviária, tais como onde construir novos pátios de cruzamento, quais ampliar para permitir acomodar trens maiores, que trechos de via duplicar, etc.

Este é um problema de otimização ferroviária bastante estudado na literatura, sendo também denominado por alguns autores como o problema de programação de trens (do inglês *train scheduling problem*). Entretanto, o verdadeiro problema de programação de trens corresponde ao problema mais geral de determinar quantos trens são necessários, bem como os respectivos tamanhos, freqüências, paradas intermediárias e horários, de forma a atender uma matriz de demanda conhecida.

Os primeiros trabalhos sobre o despacho de trens na literatura trataram de modelos analíticos de capacidade de linha, com a finalidade de estimar o atraso de cada trem causado por interferências na linha férrea em função de políticas de despacho, distribuição de tráfego e geometria de via; uma boa revisão desses trabalhos pode ser encontrada em Cordeau *et al.* (1998b). A partir de meados da década de 80, surgiram as primeiras ferramentas computadorizadas de apoio ao despacho de trens (Sauder e Westerman, 1983; Rivier e Tzieropoulos, 1984, 1987). Diversos trabalhos enfocaram explicitamente o problema de despacho e circulação de trens em ferrovias de passageiros na Europa (Carey e Lockwood, 1995; Adenso-Diaz *et al.*, 1999; Kroon e Peeters, 2003), em que predominam vias duplas e o problema de cruzamento e ultrapassagem quase não ocorre. Modelos matemáticos baseados em programação inteira, assim como estratégias de solução heurísticas, são propostas por Kraay e Harker (1994), Brännlund *et al.* (1996), Higgins *et al.* (1996, 1997) e Caprara *et al.* (2002). À medida em que aumenta o tamanho do problema, torna-se mais difícil resolvê-lo até a otimalidade; isso para não falar da necessidade de se obter uma solução em tempo reduzido,

o que tem levado à proposição de diversas heurísticas (para maiores detalhes sugere-se consultar os trabalhos de Cai *et al.*, 1998 e Şahin, 1999).

No entanto, os algoritmos propostos até o momento baseiam-se em simplificações do problema mais geral, a fim de possibilitar resolver instâncias mais gerais. Vários artigos tratam de problemas relacionados ao transporte de passageiros, em que a função objetivo e as restrições podem ser significativamente distintas do transporte de carga. A dificuldade em modelar os problemas de despacho de trens de carga, alguns formulados como problemas de otimização não-linear, tem levado a simplificações, com a finalidade de se obter uma solução viável que, muitas vezes, não consegue ser aplicada na prática. Em geral, os problemas consideram uma linha singela ligando dois pontos ou duas estações principais; nenhum deles contempla explicitamente uma malha ferroviária, composta de diferentes linhas, e as interações de trens com diferentes origens e diferentes destinos e que compartilham trechos distintos da malha ao longo das suas trajetórias. Estratégias de decomposição para tais redes podem levar a soluções de boa qualidade para cada linha individualmente, mas resultam soluções ruins do ponto de vista global, uma vez que não consideram o impacto de um trem que passa de uma linha para outra nos demais trens programados para cada linha. Restrições operacionais específicas tais como as mencionadas anteriormente e relacionadas aos tamanhos e tipos dos trens não são consideradas nas formulações nem tampouco nas soluções propostas.

6.2.1 Modelagem Matemática

Şahin, Ahuja e Cunha propõem uma nova abordagem para modelar o problema de despacho de trens, como um problema de fluxo multi-produto (“*multi-commodity flow problem*”) com restrições adicionais em um grafo (ou rede) espaço-tempo (“*space-time network*”). Cada trem t (de um conjunto T de trens a serem despachados) representa um produto (*commodity*) nessa rede espaço-tempo, que pode ser generalizada de forma a permitir representar malhas ferroviárias mais complexas, compostas de diversas linhas interligadas e com trechos em comum, conforme mostrado anteriormente na Figura 6.3. Inúmeras restrições encontradas na prática, que tornariam o modelo matemático ainda mais complexo e difícil (caso fossem adotadas as formulações matemáticas encontradas na literatura), podem ser consideradas na fase de montagem da rede, proporcionando, dessa forma, flexibilidade, sem necessidade de alteração da estrutura do modelo.

- **Representação da rede**

A fim de facilitar a compreensão de como a rede espaço-tempo é construída, considere-se a situação simplificada de uma linha férrea interligando duas estações principais, composta de trechos de linha singela e linha dupla; a generalização para uma malha mais complexa, envolvendo múltiplas linhas que se interconectam, é direta e não requer mudanças mais significativas.

A representação em rede proposta baseia-se no conceito de gráfico horário de trens, uma forma de representar a trajetória dos trens ao longo do tempo, incluindo suas paradas para cruzamento ou ultrapassagem. A Figura 6.4 exibe um exemplo simples de um gráfico horário para um trecho de via singela que interliga as estações A e F. Os pontos B a E indicam pátios de cruzamento intermediários. Os eixos vertical e horizontal denotam espaço e tempo, respectivamente. São também indicadas as trajetórias de dois trens que sem movem no sentido sul-norte (de baixo para cima), da estação A para a estação F, assim como um trem que se move no sentido contrário, de F para A; este último trem é parado duas vezes para permitir o cruzamento dos outros dois trens.

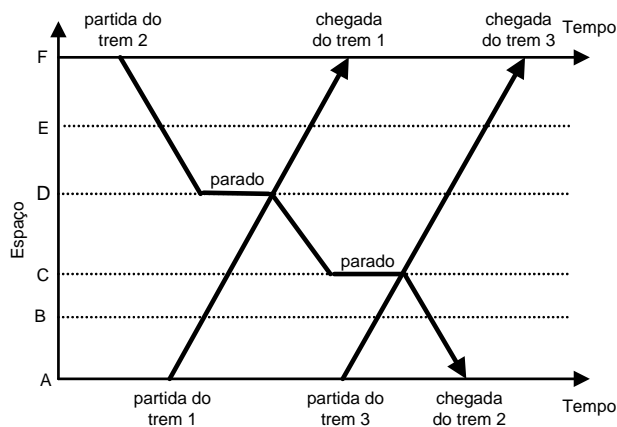


Figura 6.4: Exemplo de gráfico horário

Fonte: Ahuja *et al.* (2005)

Deve-se notar que, dependendo da solução que se dê para um conflito (por exemplo, entre os trens 1 e 2), mudam a localização e instante dos demais conflitos. Em outras palavras, mesmo para esse pequeno exemplo há diversas possibilidades de resolução dos conflitos em termos de quais trens parar e onde, sem mencionar eventuais mudanças nos horários de partida dos trens buscando minimizar atrasos. Portanto, trata-se de um problema de otimização

combinatória de grande porte, uma vez que muitos conflitos podem surgir quando houver vários trens programados. Cada conflito envolve dois trens se movimentando em sentidos opostos que se cruzam, ou no mesmo sentido, com o mais rápido ultrapassando o mais lento. Dependendo da solução dada para cada conflito envolvendo dois trens, podem mudar a localização e a hora de conflitos subseqüentes; novos conflitos anteriormente não existentes podem surgir e outros podem deixar de existir. Dessa forma, o número de possíveis soluções para o conjunto de conflitos pode ser bastante elevado.

Nessa linha interligando duas estações principais, seja $S = \{0, \dots, s\}$ o conjunto de estações e pátios intermediários de cruzamento, numerados segundo a ordem que eles aparecem ao longo da linha, tomando uma das extremidades como referência. O termo estação denota indistintamente uma estação ou um pátio de cruzamento, representando um local onde um trem pode ser parado e aguardar para cruzamento. Cada trem $t \in T$ pode partir de uma estação $s_1 \in S$ com destino a uma estação $s_2 \in S$ ($s_1 \neq s_2$). Sejam d^t e a^t as estações de origem e de destino para o trem t , respectivamente, e ed^t o seu horário de partida. Define-se $L = \{0, \dots, s-1\}$ como o conjunto de seções (segmentos de via) entre pontos de cruzamento consecutivos (estações ou pátios de cruzamento) s_i e s_{i+1} ; seja $IS(t) \subset S$ o conjunto de estações intermediárias (ou pátios) $s \in S$ onde o trem $t \in T$ pode aguardar para cruzamento ou ultrapassagem e $P(t) \subset L$ a seqüência ordenada de estações consecutivas ($d^t-s_1, s_1-s_2, \dots, s_k-a^t$) que compõem a trajetória do trem t . O tempo de viagem de cada trem $t \in T$ em cada seção i que compõe o seu trajeto é dado por f_i^t . Com base nesses dados, o horário programado de chegada do trem t ao seu destino sa^t (sem os eventuais atrasos por paradas devido aos conflitos), pode ser calculado como $sa^t = ed^t + \sum_{i \in P(t)} f_i^t$. Para cada trem t define-se ainda o atraso máximo admissível md^t , medido em relação ao horário programado de chegada sa^t , e que é definido com base em regras do negócio ou condicionantes operacionais; pode-se então calcular o horário máximo de partida do trem t $ld^t = ed^t + f_i^t$ e o seu horário máximo de chegada $la^t = sa^t + f_i^t$.

Denota-se $G = (N, A)$ a rede espaço-tempo, onde N e A indicam os conjuntos de nós e arcos, respectivamente. Definem-se três tipos de nós:

- i) *DepNodes*, que indica o conjunto dos nós fictícios, em que um fluxo de saída representa a partida de um trem da sua origem. Existe um nó de partida para cada

trem $t \in T$ dado por Dep^t . Portanto, a oferta em cada nó desse conjunto é igual a um.

- ii) *ArrNodes* indica o conjunto de nós fictícios em que um fluxo de chegada representa a chegada de um trem ao seu destino final. Analogamente, existe um nó de chegada para cada trem $t \in T$ dado por Arr^t e a sua demanda é igual a um.
- iii) *StatTimeNodes* é o conjunto dos nós intermediários da trajetória dos trens e que possuem dois atributos: local de cruzamento (i.e. pátio ou estação) e hora. O tempo é discretizado em intervalos de igual duração; assim, por exemplo, se a programação para o período de um dia (24 horas) for discretizada em intervalos de 10 minutos, então $q=144$ e o conjunto correspondente de instantes de tempo é dado por $Q = \{1, 2, \dots, q\}$. Em outras palavras, os nós do conjunto *StatTimeNodes* correspondem a cópias de cada estação/pátio $s \in S$ em cada instante de tempo $k \in Q$. Um nó pertencente a esse conjunto é representado por i_k , onde $i \in S$ e $k \in Q$.

O conjunto dos arcos A é composto de quatro subconjuntos, conforme mostrado na Figura 6.5. Os arcos do conjunto *OrigArcs* emanam dos nós em *DepNodes* e os do conjunto *DestArcs* se dirigem aos nós em *ArrNodes*. Para representar a partida de um trem t de sua estação d^t , são criados arcos de cada nó Dep^t para os nós d_k^t que indicam a primeira estação da sua trajetória em cada instante de tempo k , onde $ed^t \leq k \leq ld^t$. Analogamente, a chegada de um trem t na sua estação de destino a^t é representada por arcos dos nós a_k^t para o nó Arr^t para $sa^t \leq k \leq q'$, onde $q' = \min \{q, la^t\}$.

Os arcos da rede espaço-tempo são criados trem-a-trem, seguindo a seqüência de estações $s \in IS(t)$ e de trechos $l \in P(t)$ que compõem a trajetória do trem, a partir da sua estação inicial d^t até o seu destino final a^t . Para cada trem, os *arcos de percurso* (entre cada par de estações consecutivas da sua trajetória) são criados apenas para os instantes viáveis, considerando o horário mais cedo de partida e o atraso máximo admissível. Os *arcos de espera* representam a espera do trem t em cada uma das estações $s \in IS(t)$ onde o trem pode aguardar, considerando a máxima espera possível do trem t nessa estação; dessa forma, cada arco (i_k, i_{k+1}) representa o trem t parado na estação s entre instantes sucessivos k e $k+1$.

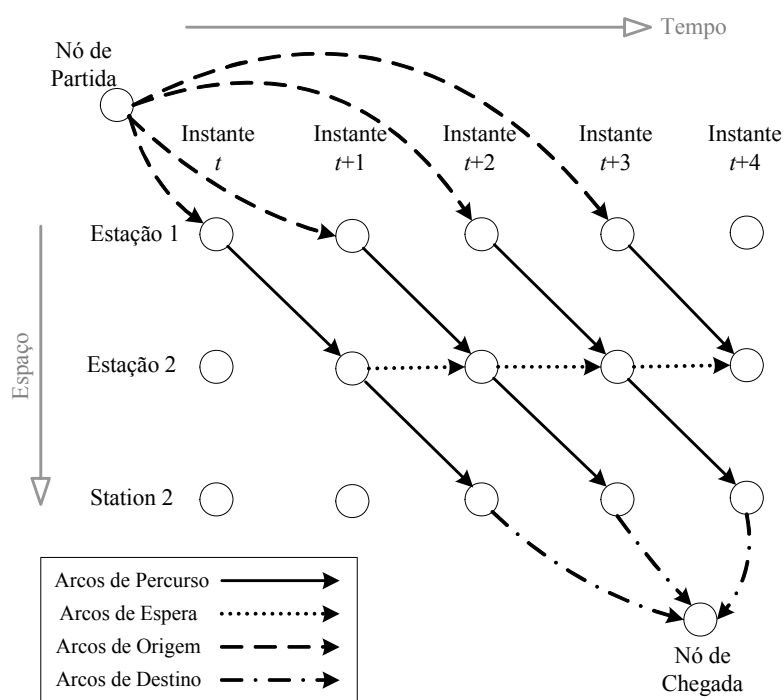


Figura 6.5: Parte de uma rede espaço-tempo para um trem

Fonte: Ahuja *et al.* (2005)

Restrições práticas podem ser consideradas e incorporadas ao modelo através da criação de novos arcos na rede ou da eliminação de certos arcos. Assim, se, por exemplo, um trem não puder aguardar num determinada estação ou pátio de cruzamento, os arcos de espera correspondente não são criados. Uma eventual outra restrição pode requerer que certos trens (basicamente trens de passageiros) permaneçam parados em determinadas estações durante períodos de tempo pré-determinados; para tal, criam-se arcos de espera correspondentes e eliminam-se os arcos de percurso que permitiriam ao trem partir ou chegar no período correspondente à janela na qual ele deve permanecer na estação. Da mesma forma, outras restrições práticas podem ser incorporadas na fase de construção da rede.

Por definição, atribui-se o valor um às capacidades de todos os arcos. O atributo de custo em cada arco de espera é igual à duração (em minutos) do intervalo em que o tempo é discretizado, tendo em vista que cada arco de espera representa um trem aguardando em um pátio do instante k até o instante $k+1$. Analogamente, os custos dos arcos de origem correspondem ao atraso inicial na partida de cada trem com relação ao seu horário programado. Os atrasos podem eventualmente ser ponderados de modo a refletir a prioridade relativa de determinados trens ou tipos de trens.

• **Formulação Matemática**

Definem-se as variáveis de decisão $x_{i_k j_l}^t$ binárias, indicando os fluxos dos trens $t \in T$ nos arcos $(i_k, j_l) \in A$. Deve-se notar que essas variáveis são definidas apenas para os movimentos viáveis de cada trem, correspondendo aos arcos dos quatro conjuntos acima descritos. Assim, o modelo matemático pode ser formulado como:

$$\text{Minimizar} \quad z = \sum_{t \in T} \sum_{(i,j) \in A} c_{ij} x_{ij}^t \quad (6.1)$$

sujeito a:

$$\sum_{ed^t \leq k \leq ld^t} x_{Dep^t d_k^t}^t = 1 \quad \forall t \in T \quad (6.2)$$

$$\sum_{sa^t \leq k \leq la^t} x_{a_k^t Arr^t}^t = 1 \quad \forall t \in T \quad (6.3)$$

$$\sum_{i \in StatTimeNodes} x_{ij}^t - \sum_{i \in StatTimeNodes} x_{ji}^t = 0 \quad \forall t \in T, \forall j \in StatTimeNodes \quad (6.4)$$

$$\sum_t x_{ij}^t \leq 1 \quad \forall (i, j) \in A \quad (6.5)$$

$$\sum_{t \in T^{out}} \sum_{l \leq k-1} \sum_{m \geq k} x_{i_l(i+1)_m}^t + \sum_{t \in T^{in}} \sum_{l \leq k-1} \sum_{m \geq k} x_{(i+1)_l i_m}^t \leq 1 \quad \forall i \in S, \forall k \in Q \quad (6.6)$$

$$x_{ij}^t \in \{0,1\} \quad \forall (i, j) \in A, \forall t \in T \quad (6.7)$$

A função objetivo (6.1) busca minimizar o atraso total dos trens. A restrição (6.2) impõe a partida de cada trem t , através de um fluxo unitário do respectivo nó de partida para algum nó do conjunto *StatTimeNode* que corresponde à estação de origem do trem, dentro da sua janela horária de partida; analogamente, a restrição (6.3) impõe a chegada de cada trem no seu destino dentro da sua janela (i.e. considerando o atraso máximo admissível). A restrição (6.4) corresponde à conservação de fluxo em cada nó do conjunto *StatTimeNodes*, enquanto que a restrição (6.5) assegura que o fluxo não ultrapasse a capacidade em cada arco. Já a restrição (6.6) impõe que cada seção de via seja ocupada por no máximo um trem simultaneamente. Deve-se notar que na formulação proposta, o fluxo total em um arco é a somatória dos fluxos para todos os trens, isto é, $x_{ij} = \sum_t x_{ij}^t$. O primeiro termo da restrição (6.6) representa a soma dos fluxos nos arcos na direção da estação i para a estação $i+1$ no instante k ; analogamente, o segundo termo representa o fluxo total nos arcos de $i+1$ para i no instante k . Essas restrições são impostas para todos os trechos em todos os instantes de tempo, isto é, totalizando $(|S| - 1) * |Q|$.

Para dar uma idéia da dimensão que um modelo matemático pode assumir, considere-se que um problema real pode ter 50 ou mais pátios ou estações onde os trens aguardam para cruzamento; a discretização do tempo requer intervalos de, no mínimo, 10 minutos para um período típico de 48 horas, o que resulta uma rede com 14.400 nós. Se houver cerca de 50 trens a serem programados, o número de variáveis de decisão binárias pode ser superior a 360 mil, tornando impossível a sua resolução através de pacotes comerciais de otimização.

6.2.2 Heurística de Solução

Motivados a proporem um algoritmo de solução que auxilie os despachadores a tomarem decisões em tempo real, o que requer que soluções efetivas possam ser obtidas em tempo reduzido (geralmente em menos que 5-10 minutos), Şahin, Ahuja e Cunha (2005) propuseram uma heurística baseada em programação inteira para o problema de despacho de trens, com base na observação do desempenho do CPLEX versão 8.1 para resolver diferentes instâncias do problema de despacho de trens. Mais especificamente, observou-se que para problemas de tamanho moderado, o CPLEX apresenta bom desempenho, atingindo a solução ótima em poucos minutos, tendo em vista a estrutura subjacente de problema de fluxo em rede. Deve-se destacar que quanto maiores os atrasos máximos admissíveis, maior o tamanho do problema devido ao aumento correspondente do número de arcos da rede, representando possíveis soluções alternativas para os conflitos. Assim, a diminuição dos atrasos máximos dos trens torna a formulação mais restrita, sem afetar a sua viabilidade. À medida em que o tamanho do problema aumenta e/ou maiores atrasos são permitidos, o CPLEX não consegue atingir a solução ótima rapidamente, embora uma solução viável possa ser encontrada rapidamente (em geral menos de um minuto).

Deve-se ressaltar que o controle de tráfego de trens em tempo real é essencialmente dinâmico; eventos inesperados e imprevisíveis ocorrem com frequência, trazendo perturbações que ocasionam atrasos dos trens a todo instante. A necessidade de otimização em tempo real sugere que um maior esforço, a fim de determinar uma solução ótima global para todos os trens englobando todo o horizonte de programação (geralmente variando entre 8-12 horas até 1-2 dias), pode não valer o esforço. Tais eventos imprevisíveis podem tornar muito improvável para alguns trens seguirem estritamente a sua programação original, particularmente para um horizonte mais longo. Por outro lado, não considerar um horizonte

mais longo pode levar a que decisões ótimas ou sub-ótimas obtidas considerando-se um horizonte de curto prazo possam vir a se mostrar inviáveis no longo prazo, isso sem considerar que mudanças freqüentes em toda a programação toda vez que algum trem atrase pode ser altamente indesejável do ponto de vista da sua aplicação prática. Assim, essas questões levaram ao desenvolvimento de uma heurística baseada em programação inteira, que utiliza o CPLEX (ou outros pacotes comerciais de otimização) para a determinação de soluções de boa qualidade.

A idéia básica consiste em ir gradualmente reduzindo os atrasos máximos (md^t) para todos os trens com base em novas soluções viáveis que são progressivamente obtidas através do CPLEX. Inicialmente divide-se o horizonte de planejamento q em dois períodos, o curto e o longo prazo. Seja q_s a duração do período de curto prazo, onde $q_s \leq q$; conseqüentemente $q_l = q - q_s$ define a duração do período remanescente de longo prazo. Definem-se δ_s e δ_l como os incrementos nos atrasos nos trens, obtidos a partir de soluções viáveis no curto e longo prazo, respectivamente. Suponha que se encontre uma solução viável na qual um trem t é atrasado d^t unidades de tempo ($d^t \leq md^t$). A idéia é tornar o modelo matemático progressivamente mais restrito, de modo a convergir mais rapidamente para uma solução de boa qualidade. Neste caso, o atraso máximo permissível para o trem t é reduzido de md^t para o mínimo entre $(d^t + \delta_s)$ e md^t para os instantes de tempo dentro do período de curto prazo. Analogamente, o atraso máximo é reduzido de md^t para o mínimo entre $(d^t + \delta_l)$ e md^t no longo prazo, onde $\delta_l \leq \delta_s$. Deve-se notar que o atraso total para todo o horizonte de programação ainda é menor que o atraso máximo para o curto prazo. Dessa forma, uma solução viável é sempre assegurada, e o problema torna-se mais restrito. O processo de otimização é então reiniciado, considerando-se o problema mais restrito, e prossegue por um período de tempo fixo σ . Assim, aumentam as chances de encontrar uma solução viável de melhor qualidade em um período de tempo menor. Repete-se o mesmo procedimento até que se tenha obtido uma solução ótima para o problema progressivamente mais restrito ou um tempo máximo de processamento tenha sido atingido.

6.2.3 Considerações Finais

Şahin, Ahuja e Cunha (2005) relatam resultados computacionais para uma ampla gama de problemas teste. Tais testes consideram ainda duas outras heurísticas propostas pelos autores:

uma heurística de construção baseada em simulação e uma heurística de enumeração gulosa. Os resultados evidenciam que a heurística descrita acima permite obter resultados de boa qualidade e em tempos de processamento reduzidos, mesmo em comparação com as demais heurísticas.

A heurística proposta permite tratar diversas restrições encontradas na prática sem tornar a formulação mais complexa. A estrutura da formulação não muda pela incorporação dessas novas restrições, uma vez que as variáveis baseiam-se numa representação de uma rede espaço-tempo ao invés de uma transformação de variáveis. Estruturas de malha mais complexas, tais como linhas duplas e linhas que se interconectam podem ser representadas sem dificuldade.

6.3 O PROBLEMA DE PROGRAMAÇÃO DE LOCOMOTIVAS

O problema de *programação de locomotivas* consiste em definir a alocação ótima de um conjunto de locomotivas aos trens programados que necessitam tração, incluindo os eventuais deslocamentos das mesmas para re-posicionamento, para um dado horizonte. A programação deve satisfazer restrições de tamanho da frota (em termos dos diversos tipos e modelos de locomotivas disponíveis), de abastecimento e manutenção das locomotivas, de necessidade de tração dos trens, e de compatibilidade. Os trens podem ter diferentes frequências: alguns trens podem ter partidas diárias, enquanto outros trens podem partir em dias específicos. A principal motivação é aumentar a utilização das locomotivas, buscando reduzir o número de locomotivas necessárias.

A programação de locomotivas pode ser analisada em dois níveis: de planejamento ou operacional. No primeiro, o objetivo é decidir quais locomotivas alocar a cada um dos vários trens programados, considerando os diferentes tipos disponíveis, e suas respectivas características como potência, capacidade de tração e custos. Algumas locomotivas podem viajar desligadas em trens, isto é, serem rebocadas com vagões de um trem; isso permite deslocar locomotivas onde não são necessárias e transferi-las para locais onde há trens necessitando tração. As locomotivas podem ainda viajar escoteiras, isto é, trafegarem sozinhas sem estarem tracionando um trem, com a mesma finalidade de re-posicionamento; nesse caso, a flexibilidade é maior, uma vez que o deslocamento das locomotivas não está

atrelado à programação dos trens, além de poderem trafegar em velocidade maior. Por outro lado, locomotivas trafegando escoteiras geram custos maiores, requererem tripulações específicas para essas operações, além de eventualmente interferirem na circulação de trens, podendo ocasionar atrasos indesejáveis, em função de conflitos de cruzamento que surgem.

Já em nível operacional, são considerados detalhes adicionais de qual unidade específica de cada tipo deve ser alocada a cada trem. Para tanto, é necessário levar em consideração os requisitos de abastecimento de cada locomotiva. Por exemplo, uma locomotiva de um determinado tipo, candidata a ser alocada a um trem pode ter combustível suficiente para percorrer mais 300 quilômetros, porém o ponto de abastecimento mais próximo para aquele trem dista 400 quilômetros, o que impede que aquela unidade específica seja alocada àquele trem. Cada locomotiva também necessita manutenção em intervalos regulares, por exemplo, uma revisão a cada 5 mil quilômetros e uma manutenção mais abrangente a cada 15 mil quilômetros; tais manutenções só podem ser realizadas em locais específicos. Portanto, uma dada locomotiva não pode ser alocada a um trem que a leve muito longe e não permita que essas manutenções possam ser realizadas conforme programado. Outro aspecto a ser considerado está relacionado aos atrasos e incertezas nos horários dos trens. Contrariamente ao nível de planejamento, em que se assume que todos os trens estejam nos seus respectivos horários, no nível operacional deve-se considerar que, na prática, os trens podem sofrer atrasos. Como levar em conta esses atrasos, e rever as alocações das locomotivas de acordo, minimizando os impactos nos custos, é uma questão importante que deve ser considerada no nível operacional.

Um trem que chega a um destino pode ter todo o seu conjunto de tração (i.e., todas as locomotivas) transferido para outro trem que está partindo (denominando-se essa transferência uma *conexão trem-a-trem*), ou alternativamente, as locomotivas podem ser separadas e alocadas a diferentes trens de partida (*quebra de tração*); nesse último caso, locomotivas originadas de diferentes trens podem compor o conjunto de tração para um trem que está partindo. Tal arranjo pode ser desvantajoso para as ferrovias, uma vez que requerem mais tempo e pessoal para manobras de reagrupamento, usualmente mais complexas, além de aumentar o risco de atraso em trens que estão partindo, uma vez que os mesmos recebem tração de mais de um trem, e se um deles atrasar, pode ocasionar atrasos em vários trens de partida. Assim, uma programação ideal deve buscar maximizar as conexões trem-a-trem e, conseqüentemente, minimizar as quebras de tração. A tração única (uma única locomotiva) também deve ser evitada pelo risco do trem ficar bloqueando a via caso a locomotiva quebre.

A programação de locomotivas é um dos problemas importantes enfrentados pelas ferrovias, uma vez que envolve um ativo de valor bastante elevado. Para ilustrar, a ferrovia CSX, uma das grandes ferrovias norte-americanas, possui cerca de 3.600 locomotivas, o que representa um capital investido superior a 7 bilhões de dólares americanos, acarretando custos operacionais e de manutenção da ordem de 700 milhões de dólares anuais. A utilização média das locomotivas (i.e., tracionando trens) está em 40%; no tempo restante elas estão ociosas, sendo deslocadas ou viajando escoteiras. Uma melhoria de apenas 5% na utilização média pode reduzir em 400 (de 3.600 para 3.200) o número de locomotivas necessárias para tracionar todos os trens programados, resultando economias potenciais superiores a 100 milhões de dólares por ano.

Dada a sua importância, são encontrados inúmeros artigos na literatura relacionados à programação de locomotivas. Cordeau *et al.* (1998a) apresentam uma abrangente revisão da literatura quanto aos modelos e algoritmos propostos para esse problema. A maioria dos modelos pressupõe haver múltiplos tipos de locomotiva disponíveis para utilização; entretanto, alguns modelos assumem que somente um tipo de locomotiva pode tracionar um trem. Esses modelos podem ser formulados como variantes de problemas de fluxo multi-produto com variáveis inteiras (“*integer multicommodity flow problems*”). Entre os artigos que tratam de problemas de alocação de um único tipo de locomotiva destacam-se Wright (1989), Forbes, Holt and Watts (1991), Booler (1995), e Fischetti and Toth (1997). Tais modelos podem ser adequados para algumas ferrovias européias, mas não o são para a realidade norte-americana, em que a tração de um trem pode ser composta de mais de um tipo de locomotiva. Já modelos de programação com múltiplas locomotivas foram considerados por Florian *et al.* (1976), Smith e Sheffi (1988), Chih *et al.* (1990), Nou *et al.* (1997), e Ziarati *et al.* (1997, 1999). A maioria dos modelos propostos está relacionada à alocação de locomotivas a trens de carga; foram encontrados poucos trabalhos com trens de passageiros (Ramani, 1981; Cordeau, Soumis e Desrosiers, 1998).

Embora existam atualmente alguns sistemas comerciais para a programação de locomotivas, os mesmos não são satisfatórios para as ferrovias norte-americanas (Shughart [1999]⁴), uma vez que não permitem obter boas programações, que possam ser implementadas. Entre as principais deficiências pode-se citar (i) a dificuldade de considerar os deslocamentos das

⁴ Comunicação particular

locomotivas escoteiras ou desligadas em trens e (ii) graus de preferência do usuário na alocação da tração de cada trem. Adicionalmente, tais modelos geram soluções com excessiva quebra de tração na alocação, acarretando atrasos e implicando custos que não estão refletidos na função objetivo desses modelos.

6.3.1 Formulação Matemática

No problema de programação de locomotivas em nível de planejamento, com um horizonte de programação semanal, busca-se a alocação ótima das locomotivas aos trens, de tal modo que cada trem receba potência suficiente ser tracionado na velocidade desejada. Atrasos, necessidades de abastecimento e de manutenção não são considerados. Busca-se ainda determinar as transferências de tração dos trens que chegam para os que partem, os deslocamentos das locomotivas escoteiras (i.e., não tracionando trens), de forma que o custo total, composto das parcelas de custo fixo, de deslocamento das locomotivas e de quebra de tração, seja minimizado.

É dada uma programação de trens semanal, em termos dos horários e locais de partida e chegada; L indica o conjunto dos trens. Assume-se que cada partida de um mesmo trem, em diferentes dias da semana, é representada por trens distintos no conjunto L . Sejam T_l e β_l a capacidade de tração (em peso) e a potência requerida para o trem l , e E_l a penalidade para usar tração composta de uma única locomotiva.

Seja K o conjunto de tipos de locomotiva disponíveis. Sejam h^k e λ^k a potência e o número de eixos para cada tipo de locomotiva k de um conjunto K de todos os tipos de locomotiva disponíveis. G^k e B^k indicam o custo fixo semanal de uma locomotiva do tipo k e a sua frota disponível, respectivamente; a capacidade de tração (em peso) de uma locomotiva do tipo k alocada a um trem l é dada por t_l^k .

O problema de programação de locomotivas é formulado como um problema de fluxo em rede multi-produto (“*multicommodity flow problem*”) do tipo linear inteiro misto, representando o fluxo de locomotivas em uma rede espaço-tempo semanal, com restrições adicionais. Nessa rede, os arcos indicam trens, os nós indicam eventos (i.e. chegadas ou partidas de trens e locomotivas), e cada tipo de locomotiva k define um produto (“*commodity*”).

Seja $G = (N, A)$ o grafo (ou rede) espaço-tempo semanal, onde N e A representam os conjuntos dos nós e dos arcos, respectivamente. Cria-se um *nó de chegada no pátio* (do inglês “*arrival ground node*”) para cada evento de chegada de trem; analogamente, para cada evento de partida é criado um *nó de partida do pátio* (do inglês “*departure ground node*”). Associam-se dois atributos a cada nó: horário e local. $DepNodes$ e $ArrNodes$ indicam, respectivamente, os conjuntos de nós de partida e de chegada de trens, enquanto $GrNodes$ indica os nós de pátio (“*ground nodes*”). Seja $AllNodes = DepNodes \cup ArrNodes \cup GrNodes$.

Essa rede possui quatro tipos de arcos, conforme mostrado na Figura 6.6, sendo construída como segue. Há um arco para cada trem programado, cujo conjunto é dado por ($TrArcs$); suas extremidades indicam os respectivos locais e horários dos eventos de partida e chegada do trem. Cada nó de chegada é conectado ao nó de chegada em um pátio correspondente através de um arco denominado *arco de conexão de chegada no pátio* (do inglês “*arrival-ground connection arc*”); analogamente, cada nó de partida de um pátio é conectado ao nó de partida do trem correspondente através de um *arco de conexão de partida do pátio* (do inglês “*departure-ground connection arc*”). O conjunto desses arcos de conexão é dado por ($CoArcs$). Em seguida, todos os nós de pátio ($GrNodes$) de cada estação são ordenados em ordem cronológica de tempo, conectando-se cada nó com o nó seguinte (de acordo com essa ordenação) através de arcos direcionados chamados arcos de pátio (do inglês “*ground arcs*”); esses arcos formam o conjunto $GrArcs$ (assume-se, sem perda de generalidade, que os nós de pátio em uma estação possuem atributos temporais únicos e diferentes entre si). Esses arcos de pátio têm a finalidade de permitir que as locomotivas de um trem chegando a uma estação possam aguardar para serem posteriormente alocadas aos trens de partida naquela mesma estação. A transferência de um conjunto de tração de um trem de chegada para um de partida é representada por um *arco de conexão trem-a-trem*; tais arcos também pertencem ao conjunto $CoArcs$.

Finalmente, permite-se a possibilidade das locomotivas se deslocarem escoteiras, i.e., sem estar tracionando um trem; para tanto, cria-se um *arco de escoteira* (do inglês *light arc*) ligando um nó de pátio (correspondendo a um local e um instante de tempo definidos) a outro nó de pátio de outra estação, sendo o conjunto de arcos de escoteira dado por $LiArcs$; F_l representa o custo fixo de enviar uma locomotiva escoteira, com tripulação, de um local a outro. Seja $AllArcs = TrArcs \cup CoArcs \cup GrArcs \cup LiArcs$. d_l^k indica o custo de uma locomotiva desligada do tipo k se deslocando no arco de trem $l \in TrArcs$; para os demais

arcos, d_l^k representa o custo de uma locomotiva do tipo k deslocando-se desligada no arco. Os conjuntos dos arcos que chegam e que emanam de em um nó $i \in N$ são dados por $I[i]$ e $O[i]$.

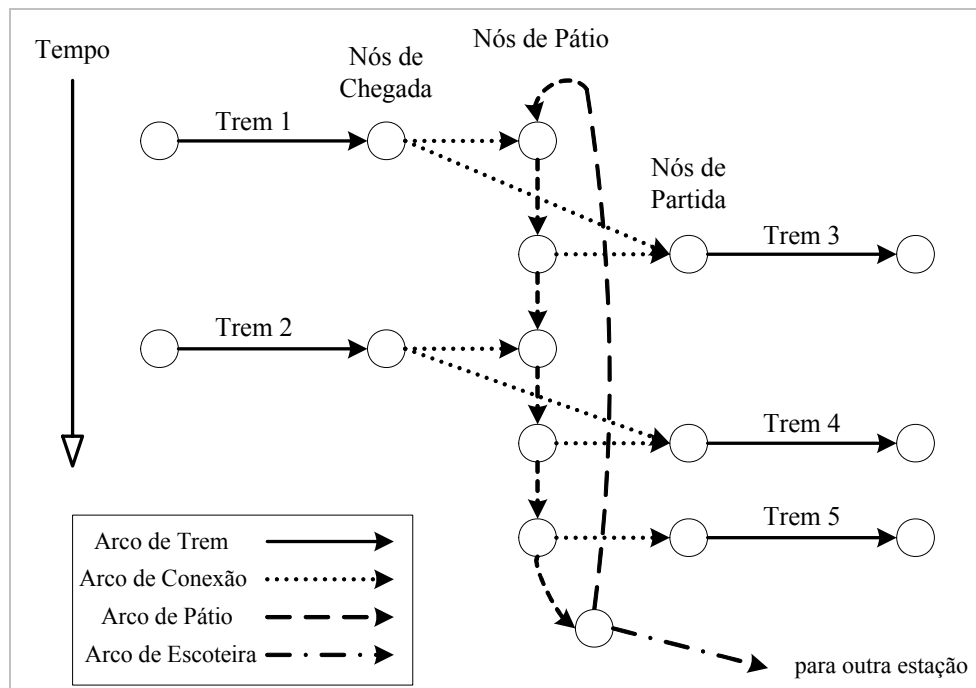


Figura 6.6: Parte de uma rede espaço-tempo para uma estação

Fonte: Ahuja *et al.* (2005)

Definem-se as seguintes variáveis de decisão: (i) x_l^k = número de locomotivas ativas do tipo k alocadas para tracionar o trem l ; (ii) y_l^k = número de locomotivas não ativas (i.e., rebocadas, escoteiras ou ociosas) do tipo k em todos os arcos; (iii) $z_l = 1$ se ao menos uma locomotiva passa pelo arco $l \in LiArcs \cup CoArcs$ e 0 caso contrário; (iv) $w_l = 1$ se uma única locomotiva é alocada ao trem l e 0 caso contrário; e, (v) $s^k =$ número de locomotivas do tipo k não utilizadas.

A formulação do problema de programação de locomotivas como um modelo de programação linear inteira mista é dada por:

$$\text{Minimizar } \sum_{l \in TrArcs} \sum_{k \in K} c_l^k x_l^k + \sum_{l \in AllArcs} \sum_{k \in K} d_l^k y_l^k + \sum_{l \in LiArcs} F_l z_l + \sum_{l \in CB} V z_l + \sum_{l \in TrArcs} E_l w_l - \sum_{k \in K} G^k s^k \quad (6.8)$$

sujeito a :

$$\sum_{k \in K} t_l^k x_l^k \geq T_l \quad \forall l \in TrArcs \quad (6.9)$$

$$\sum_{k \in K} h_l^k x_l^k \geq \beta_l T_l \quad \forall l \in TrArcs \quad (6.10)$$

$$\sum_{k \in K} \lambda_l^k x_l^k \leq 24 \quad \forall l \in TrArcs \quad (6.11)$$

$$\sum_{k \in K} (x_l^k + y_l^k) \leq 12 \quad \forall l \in TrArcs \cup LiArcs \quad (6.12)$$

$$\sum_{l \in I[i]} (x_l^k + y_l^k) = \sum_{l \in O[i]} (x_l^k + y_l^k) \quad \forall i \in AllNodes, \forall k \in K \quad (6.13)$$

$$\sum_{k \in K} y_l^k \leq 12 z_l \quad \forall l \in CoArcs \cup LiArcs \quad (6.14)$$

$$\sum_{l \in O[i]} z_l = 1 \quad \forall i \in ArrNodes \quad (6.15)$$

$$\sum_{l \in I[i]} z_l = 1 \quad \forall i \in DepNodes \quad (6.16)$$

$$\sum_{k \in K} (x_l^k + y_l^k) + w_l \geq 2 \quad \forall l \in TrArcs \quad (6.17)$$

$$\sum_{l \in S} (x_l^k + y_l^k) + s^k = B^k \quad \forall k \in K \quad (6.18)$$

$$x_l^k \in \mathbb{Z}^+ \quad \forall l \in TrArcs, \forall k \in K \quad (6.19)$$

$$y_l^k \in \mathbb{Z}^+ \quad \forall l \in TrArcs, \forall k \in K \quad (6.20)$$

$$z_l \in \{0,1\} \quad \forall l \in CoArcs \cup LiArcs \quad (6.21)$$

$$w_l \in \{0,1\} \quad \forall l \in TrArcs \quad (6.22)$$

A função objetivo (6.8) compreende seis termos: (i) os custos c_l^k das locomotivas tracionando trens; (ii) os custos das locomotivas sendo rebocadas d_l^k nos arcos de trens e nos arcos de escoteiras, assim como os custos das locomotivas ociosas; (iii) o custo fixo F_l das locomotivas escoteiras (deslocando-se sozinhas); (iv) o custo fixo V de quebra de tração de um trem que chega, para o conjunto CB de todos os arcos de conexão ligando nós de chegada a nós de pátio; (v) a penalidade associada a alocar apenas uma locomotiva a um trem; (vi) as economias das locomotivas não utilizadas. Deve-se notar que diferentes níveis de quebra de tração podem ser obtidos através da seleção de diferentes valores de V ; quanto maior o valor de V menos quebras de tração entre trens que chegam e trens que partem são admitidas.

As restrições (6.9) e (6.10) asseguram que as locomotivas alocadas a um trem proporcionem a capacidade de tração (em peso) e a potência necessária, respectivamente. A restrição (6.11) limita o número máximo de eixos do conjunto de tração alocado a um trem (24), enquanto que a restrição (6.12) impõe que sejam alocadas até 12 locomotivas a cada trem. A conservação de

fluxo em cada nó é garantida pela restrição (6.13). A restrição (6.14) captura a ocorrência de um fluxo positivo em cada arco de conexão ou de escoteira; também assegura que não mais do que 12 locomotivas possam viajar escoteiras em cada arco de escoteira. As restrições (6.15) e (6.16) impõem que as locomotivas que compõem a tração de um trem chegando sejam alocadas a um trem de saída, através de uma conexão trem-a-trem ou com quebra de tração. A possibilidade de alocação de uma única locomotiva a um trem é representada pela restrição (6.17). Finalmente, a restrição (6.18) contabiliza o número total de locomotivas utilizadas na programação semanal; esse número é determinado pela soma dos fluxos de locomotivas em todos os arcos que cruzam um determinado instante de tempo em que nenhum evento (chegada ou partida) ocorre, denominado *CheckTime*. O número de locomotivas não utilizadas (s^k) é calculado através da diferença entre o número de locomotivas disponíveis e o número de locomotivas efetivamente utilizadas.

O problema de programação de locomotivas corresponde a um problema combinatório de grande porte e do tipo NP-completo (Ahuja *et al.*, 2004). Para uma grande ferrovia pode haver centenas de trens programados, com diferentes frequências de partida, resultando em milhares de partidas por semana, sendo necessárias milhares de locomotivas de diferentes tipos, com diferentes custos, potências e capacidade de tração. Por exemplo, um problema real possui 538 trens (muitos dos quais com várias partidas semanais) e 5 tipos de locomotivas, resultando um modelo matemático com cerca de 200 mil variáveis de decisão inteiras e 67 mil restrições, dimensões essas muito superiores à capacidade de processamento dos pacotes comerciais de otimização.

6.3.2 Estratégia de Solução

Deve-se notar que a formulação matemática acima (6.8)-(6.22) não considera outras restrições encontradas na prática. Uma ferrovia pode desejar uma solução que seja consistente ao longo dos dias da semana em termos da alocação de locomotivas e das conexões trem-a-trem. Se um trem tem cinco partidas por semana, é desejável que a alocação de locomotivas seja a mesma em todos os dias, uma vez que permite definir regras de despacho que facilitem o trabalho dos despachadores de trens nos pátios. Outro ponto a ser considerado é que pode haver preferências na alocação de certos tipos de locomotivas a determinados trens.

Tendo em vista que essas restrições adicionais podem tornar a formulação matemática do problema ainda mais complexa em termos das variáveis e das restrições a serem adicionadas, sem mencionar que o tamanho da rede pode aumentar significativamente, Ahuja *et al.* (2004c) propõem uma eficiente heurística baseada em decomposição que permite obter soluções próximas da ótima para instâncias reais do problema de programação de locomotivas em tempos de processamento reduzidos. Essa heurística compreende dois estágios principais:

- (i) Modificar o problema original, de tal modo que todos os trens partam todos os sete dias da semana e resolver então o problema de programação diária de locomotivas, ao invés do problema de programação semanal.
- (ii) Ajustar a solução do estágio anterior (programação diária), de forma a obter a solução para o problema original de programação semanal em que alguns trens não têm frequência diária.

• ***Resolvendo o Problema de Programação Diária de Locomotivas (estágio 1)***

Neste primeiro estágio é considerada uma simplificação que reduz o problema semanal de locomotivas a um problema diário, reduzindo significativamente o tamanho do modelo de programação inteira mista, e também permitindo atender às restrições de consistência da alocação. Uma *rede espaço-tempo diária* é criada, de maneira idêntica à da rede semanal anteriormente descrita, e que é cerca de sete vezes menor. Assume-se que (i) todos os trens com partida em pelo menos p dias por semana partem todos os dias da semana e portanto são considerados na programação diária; (ii) todos os trens com partida em menos de p dias por semana não são considerados. Com isso, podem ser alocadas locomotivas a trens que não existem; analogamente, trens existentes podem não receber a tração que necessitam. Essa simplificação resulta da observação de dados reais, para os quais a grande maioria dos arcos de trens em uma rede espaço-tempo semanal corresponde a trens que correm pelo menos 5 dias por semana. São necessárias algumas modificações menores na formulação matemática a fim de ajustar a frota de locomotivas para o problema diário.

Embora a rede espaço-tempo diária seja significativamente menor, a mesma é ainda muito grande para que o problema possa ser resolvido utilizando o CPLEX. A título de ilustração, a rede espaço-tempo diária equivalente para o problema semanal descrito anteriormente possui 1.323 nós e 30.034 arcos. Embora a sua relaxação linear possa ser resolvida em poucos segundos, nos experimentos realizados não foi possível obter nenhuma solução inteira em 72

horas de processamento. Supõe-se que uma das principais razões seja a presença das variáveis z_i associadas aos custos fixos dos arcos de conexão e de fluxo de locomotivas escoteiras. Assim, propõe-se a eliminação dessas variáveis, obtendo-se soluções viáveis em tempos de processamento reduzidos, a partir da seguinte heurística:

Passo 1. Determinar as conexões trem-a-trem. Considera-se que as ferrovias normalmente especificam algumas conexões trem-a-trem que obrigatoriamente devem ser feitas. Essas conexões podem ser determinadas da seguinte forma: (i) adicionar todos as conexões candidatas; (ii) definir essas conexões trem-a-trem uma a uma e resolver a relaxação linear da formulação do problema diário a fim de avaliar o impacto dessa conexão; (iii) manter as conexões que impliquem impacto reduzido na função objetivo.

Passo 2. Determinar os deslocamentos das locomotivas escoteiras. A fim de permitir o deslocamento das locomotivas escoteiras de uma estação a outra em qualquer horário do dia é necessário inserir um grande número de arcos na rede espaço-tempo diária. Para evitar isso, é utilizado um procedimento heurístico para criar um conjunto reduzido de arcos potencialmente promissores para o deslocamento das locomotivas escoteiras. Para tanto: (i) adicionam-se todos os arcos (de escoteira) candidatos; (ii) em seguida, esses arcos são removidos um a um e o impacto de cada remoção é avaliado através da solução da relaxação linear do problema; (iii) removem-se os arcos com maior impacto no custo do fluxo de locomotivas escoteiras.

Passo 3. Determinar a alocação das locomotivas. Uma vez que se determinaram as variáveis de decisão z_i através dos dois passos anteriores, o problema de programação inteira resultante é resolvido, determinando-se as demais variáveis de fluxo nos arcos correspondentes às locomotivas ativas (tracionando) e às rebocadas. Embora o CPLEX possa convergir para uma boa solução em tempo reduzido (cerca de 15 minutos), o processamento não termina após mais de 48 horas de processamento. Por outro lado, essa solução inicial viável obtida rapidamente pode ser melhorada através de uma heurística baseada em vizinhança de grande porte (VLSN), conforme detalhado em Ahuja *et al.* (2004c).

Os passos acima permitem a obtenção de uma solução próxima da ótima para o problema diário de programação de locomotivas em tempos de processamento bastante reduzidos.

- ***Resolvendo o Problema de Programação Diária de Locomotivas (estágio 2)***

Conforme visto anteriormente, quando a alocação resultante da solução do problema diário de programação de locomotivas é aplicada ao problema semanal correspondente, alguns trens podem ficar sem tração (aqueles que correm menos que p dias na semana), e algumas locomotivas podem ser alocadas a trens que não existem (aqueles que não correm todos os dias). Para transformar a solução do problema diário em uma solução viável do problema semanal de programação de locomotivas, a idéia é retirar as locomotivas de trens inexistentes e alocá-las a trens que foram suprimidos do problema diário; deve-se notar que, nesse caso, podem ser necessárias locomotivas adicionais a fim de satisfazer as restrições do problema.

Constrói-se então uma rede espaço-tempo semanal modificada, de maneira similar à rede espaço-tempo diária, em que cada trem partindo em dias distintos é tratado separadamente como um trem. Entretanto, mantém-se os arcos das conexões trem-a-trem para os dias em que ambos os trens correm, eliminando-se as conexões para os dias em que algum dos dois trens não corre.

Tendo em vista o tamanho do modelo de programação linear inteira resultante para o problema semanal e, conseqüentemente, o tempo de processamento excessivo para a sua solução, Ahuja *et al.* (2004c) propõem uma estratégia de solução heurística, que determina a programação das locomotivas de cada tipo individualmente através da transformação do problema de fluxo em rede multi-produto (em que cada tipo de locomotiva define um produto) numa seqüência de problemas de fluxo em rede uni-produto com restrições adicionais, uma para cada tipo de locomotiva. O objetivo é determinar o fluxo ótimo para cada tipo de locomotiva sendo tratado, mantendo intactos os fluxos das demais locomotivas, de maneira similar à abordagem para o problema diário de programação de locomotivas. Os diferentes tipos de locomotivas são ordenados em ordem decrescente de disponibilidade de frota e tratados, um a um, nessa ordem. Após uma solução viável para o problema de programação semanal ser obtida, aplica-se uma heurística de melhoria baseada em busca em vizinhança de grande porte (VLSN) até que não seja possível melhorar a solução para nenhum tipo de locomotiva.

6.3.3 Experimentos Computacionais

A Tabela 6.1 apresenta a comparação dos resultados obtidos por Ahuja *et al.* (2004c) em comparação com aqueles obtidos através de ferramenta desenvolvida sob medida pela ferrovia norte-americana CSX. As heurísticas foram codificadas em linguagem C++ e testadas em um computador Pentium III com processador de velocidade 750 MHz. Os resultados apontam uma redução de quase 400 locomotivas em relação à prática atual da empresa, ou seja uma economia de cerca de 25% do total da frota de 1.610 locomotivas. A utilização das locomotivas aumentou 13.1%, enquanto que a ociosidade das locomotivas e os tempos viajando rebocadas diminuiram 2,4% e 9,5%, respectivamente. Os autores também reportam outros resultados em que foram obtidas melhorias semelhantes em outros problemas teste.

Tabela 6.1: Comparação da heurística proposta com as soluções da ferrovia CSX

Tipo de Locomotiva	Heurística Proposta	Solução CSX
SD40	249	498
SD50	160	171
C40	487	621
AC44	154	164
AC60	160	160
Total	1210	1614

Indicador	Heurística Proposta	Solução CSX
Tempo tracionando	44.4%	31.3%
Tempo sendo rebocada	8.1%	19.6%
Tempo ocioso	46.7%	49.1%
Tempo viajando escoteira	0.8%	0%
Quebra de tração	49.4%	85.0%

Fonte: Ahuja *et al.* (2004c)

6.3.4 Considerações Finais

Os resultados até então obtidos através da heurística proposta para o problema de programação de locomotivas na sua versão de planejamento representam soluções que atendem às restrições e regras do negócio consideradas pelas ferrovias, proporcionando economias significativas de custos, em particular a redução do número de locomotivas necessárias. As soluções podem ser obtidas em tempo de processamento reduzidos para o tipo do problema, em geral inferiores a uma hora. É a primeira abordagem de otimização a considerar explicitamente a quebra de tração, e locomotivas se deslocando escoteiras.

Por outro lado, permanecem ainda muitas oportunidades de pesquisa a serem exploradas, principalmente no que tange à programação operacional de locomotivas que possa ser utilizada no dia-a-dia das empresas ferroviárias. Para tanto, é necessário considerar restrições adicionais de abastecimento e manutenção das locomotivas, atrasos e outros desvios de programação dos trens e outras condicionantes da alocação de determinados tipos de locomotivas a certos trens, entre outras. Outra possível área de interesse é a programação de locomotivas para trens de passageiros, que pode apresentar características bastante distintas da programação para trens de carga em termos de objetivos e restrições; entretanto, as idéias e os algoritmos aqui apresentados podem ser úteis e potencialmente aplicáveis mediante modificações convenientes.

6.4 O PROBLEMA DE BLOCAGEM DE VAGÕES

O problema de *blocagem de vagões*, também conhecido como *blocagem ferroviária* (do inglês “*railroad blocking problem*”) ou *blocagem de trens*, é essencialmente um problema de consolidação: como consolidar um conjunto de cargas em grupos ou *blocos*, de maneira a reduzir a necessidade de manuseio individual de cada carga no percurso entre a sua origem e seu destino.

Esta é a forma como operam, por exemplo, os correios em inúmeros países. Uma agência postal localizada em um município qualquer, por exemplo, Piracicaba (SP) recebe diariamente correspondências destinadas a inúmeras outras localidades (talvez dezenas ou centenas) no estado de São Paulo, em outros estados e até mesmo outros países. É evidente não ser possível tratar individualmente e despachar cada correspondência diretamente para o seu destino, uma vez que o número de destinos geralmente é elevado, e o número de correspondências por destino muito reduzido (talvez, em alguns casos, apenas uma carta), o que implicaria grande ineficiência no processamento. Adicionalmente, o custo seria muito elevado para manusear e transportar cada malote por localidade de destino individualmente, para centenas ou milhares de destinos distintos, a partir de centenas ou milhares de origens. Alternativamente, o tempo necessário para acumular um conjunto de correspondências para uma mesma localidade que justificasse um transporte direto eficiente seria proibitivo diante dos prazos máximos aceitáveis. Assim, toda a correspondência recebida na tal agência de Piracicaba (SP) para

outras localidades pode ir para um centro regional de triagem e consolidação (por exemplo, em São Paulo) e daí ser separada e classificada por estado ou por região de destino, juntamente com correspondências recebidas de outras localidades para o mesmo estado ou região, e então transferida até um outro centro de triagem nesse estado ou região de destino, onde pode ser então classificada por destino final em nível de município, conjuntamente com outras correspondências recebidas de outras origens para o mesmo destino, para finalmente seguir até o seu destino final.

Esse processo de classificação e consolidação é adotado em situações semelhantes em que a demanda consiste de um número muito elevado de pequenas encomendas entre muitos pares origem-destino diferentes e que não justificam um transporte direto. Exemplos podem ser encontrados nas grandes empresas de encomendas expressas (FedEx, UPS, DHL, etc.); no transporte rodoviário de carga parcelada ou fracionada (em que a quantidade por cliente não lota um veículo, conforme visto anteriormente na seção 5.2); no transporte aéreo; em redes de telecomunicações.

No transporte ferroviário, esse problema está relacionado ao transporte de carga geral e contêineres. Ambos os serviços geralmente envolvem quantidades de carga que são insuficientes para o despacho de trens unitários (i.e., um trem de uma única origem a um único destino, sem paradas intermediárias para deixar e/ou receber vagões). Como normalmente se trata de produtos de alto valor agregado, os requisitos de nível de serviço são mais elevados em termos de tempo de trânsito, atrasos, ou outros desvios na programação.

O número de carregamentos de carga geral e containerizada numa ferrovia pode ser bastante elevado (cerca de 50 mil por mês para uma ferrovia típica americana); a grande maioria deles correspondendo a um ou alguns vagões para cada cliente, de uma origem para um destino. Daí a necessidade de consolidação desses fluxos a fim de permitir o seu transporte de maneira eficiente em termos de custo e prazo, evitando-se uma prática usual das ferrovias de adiar a partida de um trem até que se tenha atingido um número suficiente de vagões. Por outro lado, nesse processo de consolidação, os vagões podem ter que seguir um percurso que não o mais direto entre a sua origem e o seu destino, além de terem que sofrer uma ou mais reclassificações intermediárias, i.e, uma ou mais trocas de trens.

Assim, um bloco de vagões é composto de vários carregamentos que são agrupados, formando um conjunto de vagões que vai de uma origem a um destino, não necessariamente

coincidentes com a origem e/ou o destino de cada um dos vagões que o compõe. Uma vez que um vagão é inserido em um bloco, ele será re-classificado (ou separado) até que o bloco atinja o seu destino.

Dessa forma, o problema de blocagem de vagões consiste em determinar a configuração ótima de uma rede de blocos em termos dos arcos a serem construídos (sendo o que cada arco representa um bloco) e como os vagões são alocados e seqüenciados nos blocos, de modo que o custo total, dado pela soma ponderada dos custos de transporte e reclassificações intermediárias, seja minimizado.

A Figura 6.7 mostra um exemplo de uma rede de blocos onde podem ser identificados três tipos de nós: *origens* (nós 1-4) onde os carregamentos de vagões S_1 , S_2 , S_3 e S_4 se originam; *pátios* (nós 5-8) onde os vagões podem ser reclassificados ou trocarem de bloco; *destinos* (nós 9-12) para onde os carregamentos de vagões S_1 , S_2 , S_3 e S_4 se destinam. Cada arco na rede representa um bloco; por exemplo, o arco (1,5) representa um bloco que é formado na estação indicada pelo nó 1 com destino ao nó 5. Assim, o carregamento S_1 , que tem origem no nó 1 e destino em 9, pode seguir dois caminhos diferentes:

- 1-5-9, percorrendo os arcos (1,5) e (5,9), com uma reclassificação intermediária no pátio indicado pelo nó 5;
- 1-5-8-9, através dos arcos (1,5), (5,8) e (8,9), com duas reclassificações, nos nós 5 e 8.

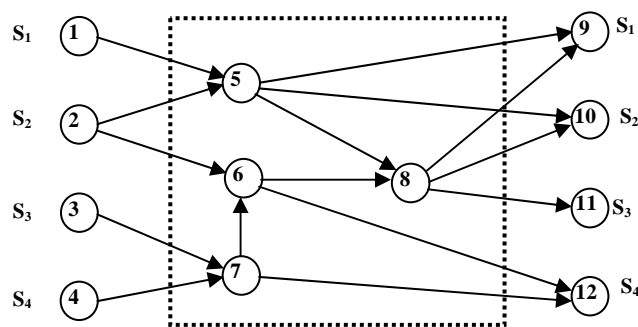


Figura 6.7: Exemplo de rede de blocos

Fonte: Ahuja *et al.* (2005)

Na prática, um pátio pode servir tanto como nó de origem, de destino, ou de reclassificação intermediária.

Determinar uma nova configuração otimizada pode representar ganhos significativos para uma ferrovia que transporta carga geral e containerizada. Por exemplo, no caso de uma ferrovia típica americana o custo de transporte é estimado em cerca de US\$ 0,50 por vagão-milha e o custo de reclassificação em US\$ 40 por vagão. Um carregamento típico percorre uma distância média de 500 milhas entre a sua origem e o seu destino final, podendo ser classificado 2,5 vezes em média. Assumindo-se haver cerca de 50 mil carregamentos por mês, com uma média de 10 vagões por carregamento, isso resulta 3 bilhões de vagões-milha e 9 milhões de reclassificações por ano, totalizando um custo anual 1,5 bilhões de dólares no deslocamento dos vagões e 1 bilhão de dólares em classificações de vagões. Se esses custos puderem ser reduzidos entre 2 e 5% através da utilização de um algoritmo de otimização, isso resultará uma economia anual potencial de dezenas de milhões de dólares.

6.4.1 Formulação Matemática

O problema de blocagem de vagões pode ser formulado como um problema de projeto de rede multi-produto (do inglês “*network design multicommodity flow problem*”). Esse problema consiste em determinar quais arcos (correspondentes a blocos) devem ser criados, e também como cada carregamento k (de um conjunto K), compreendendo v_k vagões com a mesma origem $o(k)$ e o mesmo destino $d(k)$, é movimentado através dos arcos da rede por diferentes trens que passam por pátios de classificação onde as operações de blocagem (i.e., o reagrupamento de vagões que chegam para conexão com os trens que partem) são realizadas.

Seja $G = (N, A)$ a rede de blocos, onde N é o conjunto de todos os nós que correspondem a estações onde os carregamentos se originam, se destinam ou trocam de trem, e A é o conjunto de todos os potenciais arcos que representam blocos a serem criados em $N \times N$; em outras palavras, $(i, j) \in A$ se um bloco pode ser criado do nó i para o nó j . O custo de classificar um vagão no nó i é dado por h_i , e o custo unitário de transporte de um vagão no bloco definido pelo arco (i, j) é dado por m_{ij} .

Definem-se dois conjuntos de variáveis de decisão para o problema: *variáveis de projeto* $y_{ij} \in \{0,1\}$, que definem quais *links* ou arcos $(i, j) \in A$ devem ser criados e *variáveis de fluxo* x_{ij}^k , que determinam como as “*commodities*” devem fluir pelos arcos da rede; em outras palavras, $x_{ij}^k = v_k$ se o carregamento k passa pelo arco $(i, j) \in A$ e 0 caso contrário. As

variáveis de fluxo dependem das variáveis de projeto, uma vez que os fluxos só podem passar por arcos que foram criados. $\delta^-(i)$ e $\delta^+(i)$ definem, respectivamente, os conjuntos de arcos em A chegando e saindo do nó i . Assim, o problema de blocagem de vagões pode ser formulado como o seguinte problema de programação linear inteira:

$$\text{Minimizar } \sum_{k \in K} \sum_{(i,j) \in A} m_{ij} x_{ij}^k + \sum_{i \in N} \sum_{k \in K} \sum_{(i,j) \in \delta^+(i)} h_i x_{ij}^k \quad (6.23)$$

sujeito a

$$\sum_{(i,j) \in \delta_i^+} x_{ij}^k - \sum_{(i,j) \in \delta_i^-} x_{ij}^k = \begin{cases} v_k & \text{if } i = o(k) \\ 0 & \text{if } i \neq o(k) \text{ or } d(k), \\ -v_k & \text{if } i = d(k) \end{cases} \quad \forall k \in K \quad (6.24)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i, j) \in A \quad (6.25)$$

$$\sum_{(i,j) \in \delta^+(i)} y_{ij} \leq b_i, \quad \forall i \in N \quad (6.26)$$

$$\sum_{k \in K} \sum_{(i,j) \in \delta^-(i)} x_{ij}^k \leq d_i, \quad \forall i \in N \quad (6.27)$$

$$x_{ij}^k \in \{0, v_k\} \quad \forall (i, j) \in A, \forall k \in K \quad (6.28)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (6.29)$$

Na formulação acima, a função objetivo (6.23) busca minimizar a soma dos custos de transporte nos arcos e de classificação nos nós. A restrição (6.24), em conjunto com a restrição (6.28), impõem que cada carregamento siga um único caminho na rede, enquanto que a restrição (6.25) assegura que só possa haver fluxo em arcos abertos, e que o fluxo total não ultrapasse a capacidade do arco. A restrição (6.26) restringe o número de blocos que podem ser criado em cada nó, enquanto a restrição (6.27) limita o número de vagões que podem ser classificados em cada nó. A restrição (6.29) impõe que as variáveis de projeto de arcos sejam binárias.

Esse é um problema de otimização de fluxo em rede de grande porte para problemas reais encontrados na prática, uma vez que podem atingir algo como 1.000 origens, 2.000 destinos e 300 pátios intermediários para reclassificação de blocos, por onde podem fluir 50 mil carregamentos por mês (entre 1 a 50 vagões). Um bloco pode ser criado de cada origem para qualquer pátio intermediário, entre qualquer par de pátios e destes para cada destino, resultando cerca de um milhão de variáveis de decisão y_{ij} ($1.000 \times 300 + 300 \times 300 +$

300×2.000); já o número de variáveis de fluxo x_{ij}^k pode chegar a ordem de grandeza de bilhão.

Foram feitas diversas tentativas no passado buscando modelar e resolver esse problema (Bodin *et al.*, 1980; Newton, 1996; Newton *et al.* 1998; Barnhart *et al.*, 2000). Entretanto, os maiores problemas resolvidos são bem menores que os encontrados na prática. Adicionalmente, as estratégias de solução baseadas em otimização não permitiram obter soluções de boa qualidade, ou não consideraram restrições importantes que possibilitam a sua aplicação a problemas reais.

6.4.2 Estratégia de Solução

Ahuja *et al.* (2004a) propuseram uma estratégia heurística baseada em busca em vizinhança de grande porte, ou busca VLSN (do inglês “*very large-scale neighborhood*”) a fim de permitir resolver instâncias reais desse problema.

A fim de resolver o problema de blocagem de vagões, considera-se uma simplificação: assume-se que deve ser respeitada apenas a restrição de capacidade em termos do número de blocos que podem ser formados em cada nó (b_i); as demais restrições de capacidade, i.e., número de vagões que podem ser classificados (d_i) e fluxos nos arcos (u_{ij}), podem ser ignoradas. Essa hipótese baseia-se no fato de que em problemas reais encontrados na prática, mesmo que essas duas restrições sejam desconsideradas, os valores obtidos estão dentro dos limites de viabilidade dessas restrições para a maioria dos nós e dos arcos. Dessa forma, uma vez decidido quais blocos serão criados, cada um dos k carregamentos pode seguir pelo seu caminho mínimo da sua origem até o seu destino na rede de blocos, sem violar nenhuma restrição. Conseqüentemente, as variáveis de decisão do problema se reduzem a quais blocos devem ser criados (i.e., quais y_{ij} assumem o valor 1), ficando as variáveis de fluxo x_{ij}^k automaticamente determinadas, uma vez que todos os carregamentos seguem pelos seus respectivos caminhos mínimos na rede de blocos que foi criada. As duas restrições eliminadas (i.e., número de vagões que podem ser classificados e fluxos nos arcos) são levadas em consideração através da incorporação de penalidades baseadas em relaxação Lagrangeana.

O algoritmo de solução proposto para o problema de blocagem de vagões é mostrado na Figura 6.8. Em linhas gerais, o procedimento consiste de dois passos principais: (i) obtenção

de uma solução inicial viável, (ii) que é iterativamente melhorada através da exploração da sua vizinhança, tornando-se a solução corrente, até que não seja mais possível melhorar essa solução corrente. Cada um deles é detalhado a seguir.

procedimento *VLSN-Blocking*;
início
 criar uma rede de blocos inicial e enviar todos os carregamentos através dos arcos que compõem seus respectivos caminhos mínimos nessa rede;
enquanto a solução corrente puder ser melhorada **faça**
 para cada nó $i \in N$ **faça** *{um passe}*
 re-otimizar os blocos emanando do nó i ;
 enviar todos os carregamentos ao longo dos respectivos caminhos mínimos na nova rede;
fim; *{one pass}*

Figura 6.8: Algoritmo de busca em vizinhança de grande porte (VLSN) para o problema de blocagem de vagões

Fonte: Ahuja *et al.* (2004a)

- ***Determinação de uma solução inicial viável***

Uma heurística simples e rápida é utilizada a fim de se obter uma solução inicial viável para o problema de blocagem de vagões. Essa solução deve assegurar que cada nó de origem de um carregamento seja conectado ao respectivo nó de destino através de um caminho que representa uma seqüência de arcos de blocos pelos quais o carregamento viaja. Para tanto, determina-se um ciclo direcionado que passa por todos os pátios uma única vez e retorna ao primeiro pátio (ou seja, um ciclo Hamiltoniano). Em seguida, cria-se um arco de bloco para cada nó de origem (de um carregamento), ligando-o ao pátio mais próximo deste; analogamente, cada nó de destino é conectado ao pátio mais próximo que ainda possui capacidade disponível em termos de blocos que podem ser formados. Depois que a rede de blocos for construída, cada carregamento é enviado pelo respectivo caminho mínimo nessa rede. Nos experimentos computacionais realizados, Ahuja, Jha e Liu (2004a) concluíram que o algoritmo proposto é muito pouco sensível à qualidade da solução inicial obtida, e portanto, não vale a pena o esforço adicional para a obtenção de uma solução inicial de melhor qualidade.

- ***Heurística de melhoria***

A heurística de melhoria consiste na exploração da vizinhança de uma solução viável, a qual é definida como todas as soluções que podem ser obtidas através da mudança dos blocos que

são criados considerando-se apenas um nó de cada vez e então re-seqüenciando todos os carregamentos através dos seus novos caminhos mínimos determinados nessa nova rede. Essa vizinhança ainda é muito grande, uma vez que há n possibilidades de selecionar um nó para o qual são alterados os blocos com origem no mesmo; adicionalmente, o número de diferentes maneiras que se pode selecionar k blocos que saem de um nó com p arcos que saem é dado por ${}^p C_k$, que cresce exponencialmente com p e k . Torna-se evidente ser impossível enumerar e avaliar todas as soluções vizinhas de um dado nó.

A formulação matemática desse problema de identificar o melhor vizinho resulta um modelo matemático de programação linear inteira de grande porte, que não pode ser resolvido até a otimalidade através de pacotes comerciais de otimização, mesmo após muitas horas de processamento. Acredita-se que esse problema seja NP-difícil e, portanto é necessária uma estratégia de solução heurística. Ahuja, Jha e Liu (2004a) propuseram uma estratégia de solução que permite determinar o melhor vizinho heurísticamente. Um passo desse algoritmo de melhoria para um dado nó i é composto dos seguintes passos:

Passo 1: apagar todos os arcos de bloco emanando do nó i ;

Passo 2: re-seqüenciar todo o tráfego que passa pelo nó i através dos novos caminhos mínimos;

Passo 3: criar novos arcos de blocos (um por vez) de tal maneira que resultem a maior economia no custo total;

Passo 4: re-seqüenciar os carregamentos que iniciam (ou passam) pelo nó i através dos novos caminhos mínimos considerando os novos arcos criados no passo anterior.

Nos passos 1 e 2 todos os blocos que passam pelo nó i são re-otimizados, sendo que os blocos para os demais nós não mudam. Para os passos 2 e 3 foi considerada seguinte regra de máxima economia: calcula-se a economia representada por cada novo arco (i.e., um bloco candidato) a ser criado a partir do nó i de forma a aproveitar esse arco e selecionar o arco que resulta a maior economia. No passo 4 são re-seqüenciados todos os carregamentos para os quais o custo total diminui com a criação do novo arco.

- ***Restrições adicionais encontradas na prática***

A heurística proposta tem a vantagem de permitir considerar outras restrições operacionais encontradas na prática, e que não podem ser facilmente incorporadas no modelo matemático, tais como: (i) *blocos que devem ser criados*, em virtude de razões históricas, de compromissos com clientes; (ii) *blocos que não devem ser criados*, uma vez que podem ser indesejáveis do

ponto de vista da ferrovia; (iii) carregamentos que devem ser inseridos em algum bloco pré-definido; (iv) não criar blocos com número reduzido de vagões; (v) dar preferência a alguns blocos, de modo a evitar mudanças maiores com frequência visando a atender a demanda; (vi) gerar um plano de blocos consistente com a programação dos trens, isto é, assegurar um limite no número de vezes que um bloco troca de trem; (vii) permitir mudanças incrementais na solução corrente, uma vez que as ferrovias podem querer fazer mudanças maiores tendo em vista os riscos associados em contraste com mudanças menores que podem ser facilmente implementadas; (viii) otimizar apenas parte de uma rede, ou seja, alguns nós ou um corredor.

Maiores detalhes sobre o algoritmo proposto podem ser encontrados em Ahuja *et al.* (2004a).

6.4.3 Experimentos Computacionais

A heurística proposta foi aplicada a problemas reais de três ferrovias americanas: *CSX Transportation*, *BNSF Railway* e *NS Corporation*. Os resultados obtidos foram comparados com as soluções praticadas pelas ferrovias, considerando dois parâmetros: distância média percorrida por vagão e número médio de reclassificações por vagão. Os processamentos foram feitos num computador equipado com processador Pentium IV de 2,4 GHz e memória RAM de 2 GB.

Na Tabela 6.2 são apresentadas as economias que podem ser potencialmente obtidas em comparação com a situação atual. Nota-se que o número de reclassificações intermediárias pode ser reduzido significativamente, o que representa um benefício adicional no tempo total de viagem (*transit time*), o que se traduz numa melhoria de nível de serviço para os clientes.

Tabela 6.2: Economias obtidas com novos planos de blocos otimizados

	CSX	BNSF	NS
Economias na distância percorrida	4,8%	0,1%	2,0%
Economias em reclassificações	31,6%	32,7%	41,9%
Tempo de processamento	2 horas	5 minutos	1 hora

Na Tabela 6.3 são apresentados os resultados para as ferrovias CSX e BSNF quando se permite até duas trocas de trem para cada bloco.

Tabela 6.3: Economias obtidas com limite de duas trocas de trem por bloco

	CSX	BNSF
Economias na distância percorrida	4,4%	5,2%
Economias em reclassificações	23,2%	19,8%
Economias em trocas de blocos	3,6%	4,7%
Tempo de processamento	2 hours	5 minutes

Outros experimentos considerando soluções incrementais são relatados em Ahuja *et al.* (2004a), sendo os resultados obtidos expressivos. Por exemplo, uma mudança em apenas 5% nos blocos da ferrovia CSX resulta economias de 3,2% na distância média percorrida por vagão e de 15,2% no número de reclassificações; esses valores são de 2,9% e 12,6%, respectivamente, se forem permitidas no máximo duas trocas de trem para cada bloco.

6.4.4 Considerações Finais

O problema de blocagem de vagões é um dos mais importantes enfrentados pelas ferrovias que têm na carga geral e no transporte intermodal, em particular contêineres, uma parcela importante do seu mercado. Dada a dificuldade desse problema do ponto de vista de modelagem matemática, as ferrovias resolvem esse problema manualmente. A heurística de solução baseada em busca em vizinhança de grande porte demonstrou ser possível resolver instâncias reais em tempos de processamento adequados, além de permitir incorporar restrições encontradas na prática e de difícil modelagem. Os resultados também mostraram que mudanças incrementais nas soluções utilizadas pelas ferrovias podem representar economias significativas sem o risco de uma mudança mais drástica, com reflexos em toda a programação de trens e tripulações da ferrovia.

Uma extensão interessante desse trabalho consiste em incorporar ao modelo proposto as decisões quanto à movimentação de vagões vazios, cuja principal diferença é a flexibilidade na definição dos pares origem-destino.

6.5 CONCLUSÕES DO CAPÍTULO

Neste capítulo foram apresentados três diferentes problemas relevantes no contexto das ferrovias, complexos do ponto de vista de modelagem matemática, e para os quais foram propostas sofisticadas estratégias de solução heurísticas que permitem não só resolver de maneira eficiente instâncias reais encontradas na prática, como também considerar restrições reais difíceis de incorporar em uma formulação matemática, obtendo soluções de boa qualidade e que possam ser efetivamente implementadas e utilizadas pelas ferrovias.

O modelo e a abordagem propostos para o problema de despacho de trens têm utilização direta e imediata nas ferrovias brasileiras, uma vez que os problemas encontrados aqui em nada diferem das hipóteses e condicionantes consideradas na modelagem; pelo contrário, a experiência e o conhecimento da realidade brasileira por este autor foram importantes na solução do problema, uma vez que muitas ferrovias americanas e européias contam com sofisticados sistemas de controle de tráfego em tempo real que já dispõem de recursos de auxílio ao despacho de trens.

Já para o problema de programação de locomotivas, a realidade das ferrovias brasileira após sua privatização levou a uma configuração em malhas, e conseqüentemente os problemas, são menores que os problemas típicos considerados na modelagem. Adicionalmente, conforme descrito anteriormente, algumas ferrovias brasileiras apresentam características operacionais bastante distintas, com fluxo de trens muito intenso de um único produto, entre um par origem-destino (como, por exemplo, minério de ferro na EFVM, na EFC e na malha MRS), que não requerem decisões quanto à alocação de tração, uma vez que os trens vão carregados e voltam vazios. Por outro lado, o recente interesse das ferrovias em aumentar sua participação no transporte de carga geral e containerizada, de maior valor agregado e com possibilidade de retorno maior, pode tornar útil o uso de modelos e algoritmos como os propostos aqui. Há de se considerar ainda outras práticas operacionais adotadas pelas ferrovias brasileiras, entre as quais o uso de uma locomotiva auxiliar (*helper*) em determinados trechos mais íngremes, mas que não fazem parte da tração designada ao trem ao longo do seu percurso.

As ferrovias brasileiras também podem se beneficiar dos avanços em termos do problema de blocagem de vagões, principalmente no contexto anteriormente exposto de incrementar a participação da carga geral e conteneurizada. Embora em escala muito menor que as ferrovias

americanas, a modelagem proposta pode ser aplicada no caso brasileiro, talvez de maneira combinada com a programação de trens, ao invés de separada e seqüencial como ocorre no caso americano (Ahuja *et al.*, 2005)

Outras questões, como mudança de bitola, podem necessitar serem consideradas em alguns contextos da realidade brasileira, em particular na malha sob responsabilidade do consórcio Brasil Ferrovias, composto pelas empresas Ferrobán, Ferronorte e Novoeste.

Capítulo

7

CONSIDERAÇÕES FINAIS

A modelagem matemática de problemas em logística e transportes continua atraindo grande interesse de pesquisa na comunidade científica e acadêmica. Uma consulta ao portal *Science Direct* (www.sciencedirect.com), que reúne alguns dos periódicos mais prestigiosos da editora Elsevier (incluindo alguns dos mais importantes periódicos da área, muitos delas com ênfase para métodos quantitativos e modelagem matemática, entre os quais *Transportation Research*, *Computers and Operations Research*, *European Journal of Operational Research*), em 23/12/2005, utilizando o termo “*logistics*”, retornou mais de 10 mil documentos; já com o termo “*transportation*” foram listados 8.640 artigos.

Conforme se procurou demonstrar ao longo deste texto, em que foram apresentados vários problemas de roteirização e programação de veículos, de projeto de redes de logística e de otimização ferroviária com os quais este autor se envolveu, existe uma grande diversidade de problemas práticos que envolvem a otimização em logística e transporte, com benefícios e impactos muito significativos em termos de redução de custos e investimentos, de melhoria de nível de serviço e de eficiência operacional. Adicionalmente, muitos desses problemas apresentam complexidade matemática e computacional que tornam a sua resolução impossível através de abordagens clássicas utilizando métodos exatos de otimização (apesar de todos os significativos avanços alcançados), atraindo elevado interesse para a pesquisa e o desenvolvimento de heurísticas que permitam tratar e resolver instâncias reais.

Para ilustrar esse interesse, o *Center for Research on Transportation* da Universidade de Montreal, no Canadá (<http://www.crt.umontreal.ca>), um dos maiores centros de pesquisa em nível mundial, e cuja missão é “*desenvolver pesquisas envolvendo métodos quantitativos e computacionais para problemas de planejamento, gestão/administração e operação de*

sistemas de transporte urbano, interurbano e regional de cargas e passageiros”, possui cerca de 48 docentes e pesquisadores permanentes no seu quadro, incluindo alguns dos nomes mais conhecidos mundialmente na área, tais como Gilbert Laporte, Michel Gendreau, Teodor Crainic, Michael Florian, entre outros, autores de centenas de artigos nos mais destacados periódicos da área.

Em termos de tendências, observou-se, em especial na última década, o surgimento e a consolidação de diversas estratégias de solução heurísticas, em particular as metaheurísticas, que permitiram resolver, de maneira eficiente e rápida, diferentes problemas combinatórios e de ordem de complexidade exponencial, entre as quais busca tabu, algoritmos genéticos, busca em vizinhança de grande porte, entre outros.

Embora tais algoritmos tenham possibilitado determinar os melhores resultados conhecidos para muitos problemas, alguns até então insolúveis, os tempos de processamento ainda são elevados, o que dificulta a sua incorporação a aplicações comerciais. Adicionalmente, as metaheurísticas são muito dependentes do contexto específico e requerem ajuste fino de parâmetros de processamento caso a caso, dificultando a sua incorporação aos *softwares* comerciais. Tome-se, por exemplo, o caso da busca tabu, uma das de maior sucesso para problemas de roteirização de veículos (Laporte *et al.*, 2000). Inúmeros são os parâmetros que controlam a busca tabu, como visto no Capítulo 3; os melhores resultados publicados na literatura foram obtidos para diferentes conjuntos de parâmetros, ajustados e calibrados caso a caso.

Assim, um dos grandes desafios é o desenvolvimento de estratégias de solução mais robustas, que resultem bom desempenho computacional e permitam obter bons resultados quando aplicadas a diferentes problemas. Só assim, muitos dos avanços científicos poderão ser efetivamente incorporados à realidade dos *softwares* e produtos utilizados pelas empresas e disponibilizados comercialmente. A título de ilustração, conforme aponta Cunha (2000), muitos dos pacotes de roteirização líderes de mercado apoiam-se em heurísticas clássicas, desenvolvidas há mais de três décadas, tais como a heurística de economias (Clarke e Wright, 1964) e de varredura (Wren e Holiday, 1972; Gillet e Miller, 1974).

Nesse contexto, parece muito promissor o caminho de explorar abordagens híbridas, aplicando simultaneamente e de maneira integrada mais de uma metaheurística, a fim de

compensar as deficiências de cada metaheurística e, ao mesmo tempo, evitar códigos muito complexos, específicos, difíceis de manter, cheio de parâmetros para calibrar.

Nesse sentido, um possível caminho a ser explorado em um futuro próximo é o uso de *scatter search* em conjunto com algoritmos genéticos, a fim de assegurar simultaneamente melhoria e maior diversidade da população. Da mesma forma, o uso de GRASP com heurísticas de melhorias, como busca tabu e busca em vizinhança de grande porte apresenta potencial a ser explorado. Ambas as idéias se inserem dentro de um contexto mais geral denominado hiper-*heurísticas* (“*hyper-heuristics*”), termo que vem sendo utilizado para descrever o processo em que metaheurísticas escolhem as metaheurísticas certas para resolver um problema; mais especificamente, diferentes metaheurísticas são utilizadas em diferentes fases da busca (Burke *et al.*, 2003). A principal motivação é que essa abordagem pode resultar maior simplicidade de implementação e maior facilidade de aplicação do que procedimentos especialmente desenhados para problemas específicos.

Além da questão da maior robustez e eficiência mencionada acima, outra oportunidade de pesquisa está na avaliação e comparação das heurísticas propostas, em termos da sua eficiência computacional. Muitos dos problemas-teste utilizados largamente como *benchmark* ainda estão muito distantes da realidade dos problemas encontrados na prática. A Figura 7.1 ilustra dois problemas teste dentre os propostos recentemente por Li, Golden e Wasil (2005) para testes de algoritmos de roteirização considerando-se instâncias de grande porte (até 1200 clientes).

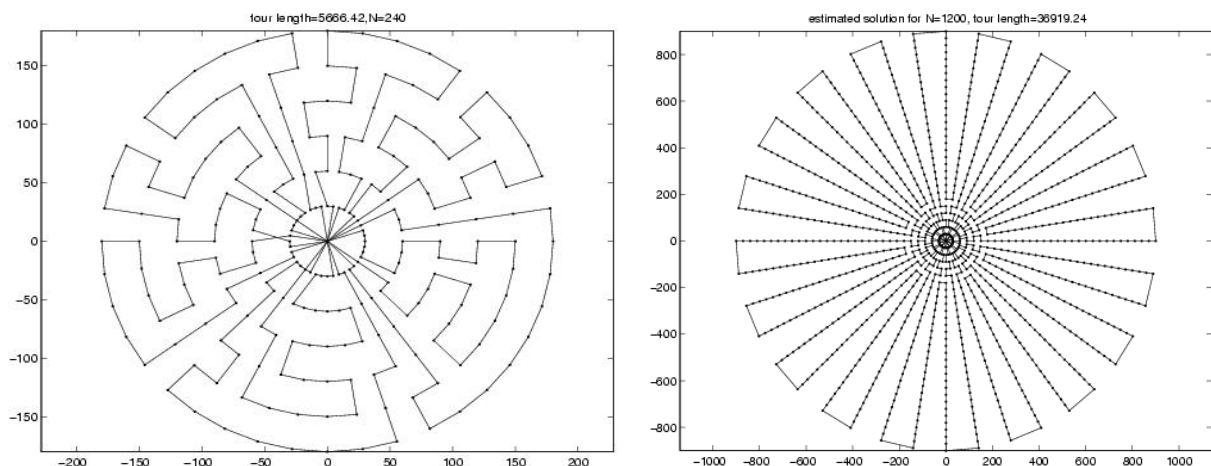


Figura 7.1: Novos problemas teste *benchmark* de grande porte com 240 e 1200 clientes

Fonte: Li, Golden e Wasil (2005)

Tais problemas, embora permitam conferir e avaliar quase que visualmente a qualidade das soluções obtidas pelas heurísticas testadas, estão muito distantes da realidade dos problemas encontrados na prática.

Portanto, avaliar e selecionar estratégias de solução heurísticas com base nos resultados obtidos para instâncias tão artificiais e distantes da realidade pode levar a conclusões equivocadas e induzir futuros insucessos e descrédito na aplicação dessas técnicas a problemas reais.

Adicionalmente, pode-se arriscar afirmar que existe atualmente uma verdadeira compulsão pela obtenção e publicação de resultados melhores, alguns implicando melhorias ínfimas, muitas vezes inferiores a 1%. Ignoram-se, além das questões da robustez e dos problemas-teste utilizados, aspectos do realismo da modelagem do problema (i.e., se o modelo matemático efetivamente representa o problema considerado), assim como a questão da qualidade dos dados de entrada. Em outras palavras, pode-se questionar qual o mérito ou virtude de alcançar melhoria de menos de 1% na distância total percorrida pelos veículos em um problema de roteirização, se a qualidade e a (im)precisão dos dados de entrada são muito superiores a isso.

Existem inúmeros aspectos quanto à forma de representação de dados de entrada e sua acurácia, conforme ilustra Cunha (2000) para problemas de roteirização: redes matemáticas que representam adequadamente restrições à circulação tais como mãos de direção, conversões permitidas e proibidas; um único valor de tempo (ou de velocidade) em cada trecho de via, independentemente do horário do dia; a impossibilidade de localização automática de endereços (“*address matching*”) no caso brasileiro, decorrente tanto da imprecisão do cadastro (falta de CEP ou incorreto), quanto da existência de mais de um logradouro com o mesmo nome em regiões metropolitanas, ruas sem nome (rua A, B, etc.), endereços que não constam do cadastro oficial (geralmente em áreas de periferia), entre outros.

Tais dificuldades com dados espaciais e geográficos afetam o grafo (ou rede) usualmente utilizado para representar e modelar não só problemas de roteirização de veículos, como também muitos outros problemas de modelagem em logística e transportes.

Em síntese, conforme procurou-se evidenciar ao longo do texto, muitos dos problemas encontrados na prática no Brasil apresentam características e condicionantes distintos

daqueles encontrados na literatura científica em nível mundial, abrindo oportunidades interessantes de pesquisa. Por exemplo, no caso da roteirização de veículos, uma questão nem sempre considerada é a do uso de frota própria *versus* contratação de terceiros. Isso ocorre porque, no Brasil, os fretes pagos nem sempre são calculados com base na distância percorrida pelos veículos. Nas contratações de terceiros para serviços de coletas e entregas urbanas, a prática mais comum é o pagamento segundo a quantidade total transportada pelo veículo (medida pelo peso, volume, número de caixas, etc.) e segundo o número de paradas do roteiro. Nas grandes cidades, usualmente a contratação pode prever ainda valores diferenciados de fretes unitários por região ou por área a ser atendida, de modo a considerar uma maior “dificuldade” espacial (loais mais distantes, na periferia, com sistema viário deficiente, com entregas mais dispersas, etc.). Há situações em que o frete de todas as entregas é calculado com base na entrega mais distante ou “difícil”. Os modelos encontrados na literatura não consideram esse tipo particular de estrutura de custo, desvinculada da distância efetivamente percorrida, levando a soluções onde os custos com a distância e com a frota própria sejam minimizados, o que não necessariamente corresponde à solução de menor custo quando há terceiros realizando parte dos atendimentos.

Também, por exemplo, na programação de veículos e tripulações e no transporte ferroviário de carga as restrições e condicionantes da realidade brasileira são muito diferentes das encontradas em outros países e abordados nos artigos científicos, abrindo excelentes oportunidades de pesquisa.

Uma área interessante de pesquisa é a utilização conjunta de modelos de otimização com modelos de simulação, como uma maneira de escapar da complexidade matemática decorrente de problemas de natureza estocástica, como apontam Glover e Kelly (1998).

Além de contribuir efetivamente para a pesquisa, o conhecimento e a aplicação de técnicas de Pesquisa Operacional, especialmente Modelagem Matemática, na solução de problemas típicos de sistemas logísticos e de transportes, espera-se que este trabalho tenha efetivamente contribuído para apresentar, de maneira introdutória, os principais algoritmos e metaheurísticas, como também para estabelecer a motivação para o uso de estratégias heurísticas, a partir de uma revisão do funcionamento dos algoritmos exatos, e sua dificuldade para resolver inúmeros problemas encontrados na prática. Evidentemente, não se buscou substituir os textos encontrados na literatura sobre cada uma das metaheurísticas apresentadas, mas sim apresentar uma visão global de todas elas e de aspectos da sua implementação

computacional, o que deve ser complementado por artigos específicos. Não houve a pretensão de ser exaustivo em nenhum dos seus aspectos, e muito menos, de esgotar os assuntos tratados.

Para finalizar, este trabalho não contemplou algumas pesquisas importantes em andamento com alunos de mestrado e doutorado, incluindo (i) o problema de roteirização periódica, uma generalização do problema de roteirização para um horizonte de planejamento superior a um dia, em que se deve decidir simultaneamente os melhores dias de coleta para cada ponto e os respectivos roteiros (Wu e Cunha, 2005); (ii) o problema de estoque-roteirização, em que os estoques do produto nos clientes são geridos pelo fornecedor (como, por exemplo, na entrega de gases industriais), e que consiste em definir simultaneamente quando atender cada cliente, quanto fornecer do produto quando o cliente é atendido, e os roteiros de entrega, de modo a minimizar o custo total de estoque e distribuição (Znamensky e Cunha, 2003); (iii) o problema de modelagem de redes capacitadas e com múltiplos critérios.

Apesar de todos os esforços, a Pesquisa Operacional, em particular os modelos matemáticos de otimização, ainda mal começaram a engatinhar no Brasil, trazendo desafios e abrindo perspectivas muito interessantes para a pesquisa e o desenvolvimento científico e tecnológico, solucionando problemas reais, e contribuindo para o progresso e a melhoria da qualidade de vida.

Assim, conforme apontam Ahuja, Cunha e Sahin (2005), se desejarmos que a Pesquisa Operacional seja efetivamente reconhecida como a *Ciência do Melhor* (do inglês “*Science of Better*”), e seja efetivamente utilizada para a tomada de decisão em problemas do mundo real e para adicionar valor às organizações, precisamos ir além dos paradigmas atuais que direcionam os problemas que escolhemos para resolver, a maneira como os modelamos, e como os resolvemos.

Em resumo, para desenvolver e oferecer ferramentas de PO que auxiliem as organizações a encontrar soluções quase-ótimas que possam ser aplicadas e utilizadas na prática, resultando ganhos substanciais, para não mencionar melhor qualidade, nível de serviço, satisfação dos clientes e confiabilidade, para citar alguns, devemos ser *inovadores* ao modelar e resolver esses problemas. Existem inúmeras oportunidades envolvendo problemas complexos em logística e transporte, que podem ser efetivamente resolvidos, conforme se procurou demonstrar ao longo deste texto.

REFERÊNCIAS BIBLIOGRÁFICAS

- Abdinnour-Helm, S. (1998) A hybrid heuristic for the uncapacitated hub location problem. *European Journal of Operations Research*, v.106, p.489-499.
- Abdinnour-Helm, S. e Venkataramanan, M.A. (1995). Solution approaches to hub location problems. *Annals of Operations Research*, 78, pp. 31-50
- Adenso-Diaz, B., Gonzalez, M.O. e Gonzalez-Torre, P. (1999). On-Line Timetable Re-Scheduling in Regional Train Services, *Transportation Research B* 33, p.387-398.
- Agência Nacional de Petróleo – ANP (2001). *Anuário Estatístico Brasileiro do Petróleo e do Gás Natural*.
- Ahuja, R. K., Magnanti, T.L. e Orlin, J.B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, NJ.
- Ahuja, R.K., Orlin, J.B. e Sharma, D. (2000) Very large scale neighborhood search. *International Transactions in Operations Research* v.7, 301-317.
- Ahuja, R.K., Orlin, J.B. e Sharma, D. (2001). Multi-exchange neighborhood search algorithms for the capacitated minimum spanning tree problem. *Mathematical Programming* 91, p.71-97.
- Ahuja, R.K., Ergun, O., Orlin, J.B. e Punnen, A.P. (2002). A survey of very large scale neighborhood search techniques. *Discrete Applied Mathematics* 123, p.75-102.
- Ahuja, R. K., Jha, K. C. e Liu, L. (2004a). Solving real-life railroad blocking problems. *Operations Research* (submetido para publicação).
- Ahuja, R. K., Orlin, J.B., Pallotino, S., Scaparra, M.P. e Scutellà, M. G. (2004b). A Multi-Exchange Heuristic for the Single-Source Capacitated Facility Location Problem. *Management Science* 50, p. 749-760.
- Ahuja, R. K., Liu, J, Mukherjee, A., Orlin, J. B., Sharma, D. e Shughart, L. A. (2004c). Solving real-life locomotive scheduling problems. Aceito para publicação no *Transportation Science*.
- Ahuja, R.K; Cunha, C. B. (2005). Very Large-Scale Neighborhood Search for the K-Constraint Multiple Knapsack Problem. *Journal of Heuristics*, v.11, n.5-6, p. 465-481.
- Ahuja, R.K., Cunha, C.B., Sahin, G. (2005). Network Models in Railroad Planning and Scheduling. In: *Tutorials in Operations Research 2005*, H.J. Greenberg, J. Cole Smith (eds), Informs, Hanover, MD, p. 54-101.
- Ambrosino, D. e Scutellà, M.G. (2005). Distribution network design: New problems and related models. *European Journal of Operational Research* 165, p.610-624.
- Assad, A. A. (1988). Modeling and implementation issues in vehicle routing. In: *Vehicle Routing: Methods and Studies*, B.L.Golden, A.A.Assad (eds), North Holland, Amsterdam, p. 7-46.
- Avella, P., Boccia, M. e Sforza, A. (2004). Solving a Fuel Delivery Problem by Heuristic and Exact Approaches. *European Journal of Operational Research*, vol. 152, pp. 170-179.
- Aykin, T. (1995). Network policies for hub-and-spoke systems with application to the air transportation system. *Transportation Science*, v. 29, n. 3.

- Ball, M. O., Golden, B. L., Assad, A. A. e Bodin, L. D. (1983) Planning for truck fleet size in the presence of a common-carrier option. *Decision Sciences*, v.14, n.1, p.103-120.
- Ballou, R.H. (2001). Unsolved Issues in Supply Chain Network Design. *Information Systems Frontiers* 3:4, p.417-426.
- Barceló, J. e Casanova J. (1984). A heuristic Lagrangean algorithm for the capacitated plant location problem. *European Journal of Operational Research* 15, p.212-226.
- Barnhart, C., Johnson, E., Nemhauser, G.L., Savelsbergh, M.W. e Vance, P. H. (1998). Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46(3), p.316-329.
- Barnhart, C., Jin, H. e Vance, P. H. (2000). Railroad blocking: A network design application. *Operations Research* 48, p.603-614.
- Battiti, R. e Tecchiolli, G. (1994). The reactive tabu search. *ORSA Journal on Computing* 6(2), p.126-140.
- Beasley, J. E. (1999). *Population Heuristics*. The Management School, Imperial College, London.
- Beasley, J.E., Chu, P.C. (1996). A Genetic Algorithm for the set covering problem. *European Journal of Operational Research* 94(2), 392-404.
- Bechara, J. J. B. e Galvão, R. D. (1984). O Uso de Sistemas Computacionais Interativos na Solução de Problemas de Roteamento de Veículos. *Anais do XVII Simpósio da SOBRAPO*, pp. 279-293, Rio de Janeiro.
- Benders, J.F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4, 238-252.
- Bent, R e Hentenryck, P. (2004). A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science* v.38, n.4, p. 515-530.
- Bertsimas D. e Weismantel, R. (2005). *Optimization over Integers*. Belmont, Massachusetts.
- Bisby, R.E. (2002). Solving real-world linear programs: a decade and more of progress. *Operations Research*, v. 50, n.1, p.3-15.
- Bixby, R.E., Fenelon, M., Gu, Z., Rothberg, E. e Wunderling, R. (1999) MIP: Theory and Practice: Closing the Gap. In: *System Modeling and Optimization: Methods, Theory and Applications*, M.J.D. Powell e S. Scholtes (eds.), Kluwer, Dordrech, p.19-49.
- Bodin, L.D., Golden, B.L., Schuster, A.D. e Romig, W.(1980). A model for the blocking of trains. *Transportation Research B*, 14 p.115-120.
- Bodin, L.D., Golden, B.L, Assad, A. e Ball, M. (1983). Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research* 10(2).
- Bodin, L.D. (1990). Twenty years of routing and scheduling. *Operations Research* 38(4), p.571-579.
- Bramel, J. e Simchi-Levi, D. (1998). *On the effectiveness of set covering formulations for the vehicle routing problem with time windows* Technical Report, Dept. of Industrial Engineering. Northwestern University.
- Brännlund, U., Lindberg, P. O., Nõu, A. e Nilsson, J-E. (1998). Railway timetabling using lagrangian relaxation. *Transportation Science* 32(4), p. 358-369.

- Brandeau, M.L. e Chiu, S.S. (1989). An overview of representative problems in location research. *Management Science*, p.645-667.
- Brown, G. G.; Graves, G. (1981). Real-time dispatch of petroleum tank trucks, *Management Science*, v. 27, pp.19-32.
- Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P. e Schulenburg, S. (2003). Hyper-heuristics: an emerging direction in modern search technology. In: *Handbook of Metaheuristics*, F. Glover e G.A. Kochenberger (eds). Kluwer, Boston, p.457-474.
- Cai, X., Goh, C. J. e Mees, A. I. (1998). Greedy heuristics for rapid scheduling of trains on a single track. *IIE Transactions* 30, p.481-493.
- Cantú-Paz, E. (2001). *Efficient and accurate parallel genetic algorithms*. Kluwer, New York.
- Caprara, A., Fischetti, M. e Toth, P (2002). Modeling and solving the train timetabling problem. *Operations Research* 50(5), p. 851-861.
- Carey, M. e Lockwood, D. (1995). A model, algorithms and strategy for train pathing. *Journal of the Operational Research Society* 46, p.988-1005.
- Carraresi, P. e Gallo, G. (1984). Network models for vehicle and crew scheduling. *European Journal of Operational Research*, v.16, p.139-151.
- Chih, K. C., Hornung, M. A., Rothenberg, M. S. e Kornhauser, A. L. (1990). Implementation of a real time locomotive distribution system. In: *T. K. S. Murthy, R. E. Rivier, G. F. List, and J. Mikolaj, editors, Computer Applications in Railway Planning and Management*. Computational Mechanics Publications, Southampton, UK ,1990.
- Chiang, W. C. e Russel, R.A. (1997). A reactive tabu search metaheuristic for the vehicle routing problem with time windows. *INFORMS Journal on Computing* 9, p.417-430.
- Christofides, N., Mingozzi, A., Toth, P. (1979). The vehicle routing problem. In: *Combinatorial Optimization*, N. Christofides, A. Mingozzi e P. Toth (eds), Wiley, Chichester, p.315-338.
- Chopra, S. e Meindl, P. (2001). *Gerenciamento da cadeia de suprimentos : estratégia, planejamento e operação*. 465 p. São Paulo : Pearson Education do Brasil: Prentice Hall, 2004.
- Chu, P.C., Beasley, J.E. (1997). A genetic algorithm for the generalized assignment problem. *Computers and Operations Research* 24(1), 17-23.
- Clarke, G. e J. W. Wright (1964), Scheduling of Vehicles From a Central Depot to a Number of Delivery Points. *Operations Research* 12, p.568-581.
- Conway, D.G. e Ragsdale, C.T. (1997). Modeling Optimization Problems in the Unstructured World of Spreadsheets. *Omega International Journal of Management Science* 25(3), p.313-322.
- Cook, W. (2005). The traveling salesman problem. Acessível em <http://www.tsp.gatech.edu/>
- Cordeau, J.-F., Soumis, F. e Desrosiers, J. (1998a) A benders decomposition approach for the locomotive and car assignment problem. *Technical Report G-98-35*, GERAD, École de Hautes Études Commerciales de Montréal, Canada.
- Cordeau, J.-F.; Toth, P. e Vigo, D. (1998b). A survey of optimization models for train routing and scheduling. *Transportation Science* 32, p.988-1005.

- Cordeau, J.-F., Laporte, G. e Mercier, A. (2001). A unified tabu search for vehicle routing problems with time windows. *Journal of Operational Research Society*, 53, p.513-522.
- Cordeau, J. F.; Gendreau, M; Laporte, G; Potvin, J. Y.; Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society* 53, p.512-522.
- Cormen, T.H., Leiserson, C. E., Rivest, Ronald L. e Stein, C. (2001). *Introduction to Algorithms, Second Edition*. Cambridge : MIT Press, 1180p.
- Crainic, T. G. e Toulouse, M. (2003). Parallel strategies for metaheuristics. In: *Handbook of Metaheuristics*, F. Glover e G.A. Kochenberger (eds). Kluwer, Boston, p.475-514.
- Cunha, C. B. (1997). *Uma contribuição para o problema de roteirização de veículos com restrições operacionais*. São Paulo. 222p. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de Transportes.
- Cunha, C. B. (2000). Aspectos práticos da aplicação de modelos de roteirização de veículos a problemas reais. *Transportes* 8(2), p.51-74.
- Cunha, C.B. e Swait, J.D. (2000). New Dominance Criteria for the Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows on Dense Graphs” *International Transactions in Operational Research*, v.7, p.139-157.
- Cunha, C. B. (2003). Um Modelo Matemático para o Problema de Sequenciamento e Programação de Visitas de Gerentes de Banco. *Gestão & Produção* no. 2, 10, 183-196. São Carlos.
- Cunha, C. B. e Mutarelli, F. (2006). A spreadsheet-based optimization model for the integrated problem of producing and distributing a major weekly newsmagazine. *Aceito para publicação no European Journal of Operational Research*.
- Cunha, C. B. e Silva, M. R. (2006). A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. *Aceito para publicação no European Journal of Operational Research*.
- Current, J., Daskin, M.; Schilling, D. (2002). Discrete network location models. In: Drezner, Z.; Hamacher, H. *Facility location theory: applications and methods*. Berlin: Springer-Verlag, 2002. p.81-118.
- Davis, L. (1991) *Handbook of Genetic Algorithms*. Van Nostrand. New York.
- Desrochers, M. e Soumis, F. (1988). A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFOR*, 26(3), p.191-212.
- Desrochers, M. e T.W. Verhoog (1991) A New Heuristic For The Fleet Size and Mix Vehicle Routing Problem. *Computers and Operations Research* 18(3), p.263-274.
- Desrochers, M., Desrosiers, J. e Solomon, M. M. (1992). A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. *Operations Research*, v.20, p.342-354.
- Desrosiers, J., Soumis, F. e Desrochers, M. (1984). Routing with Time Windows by Column Generation. *Networks*, v.14, p.545-565.
- Desrosiers, J., Dumas, Y. e Soumis, F. (1986). A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *American Journal of Mathematical and Management Sciences*, v.6, p.301-325.
- Dillmann, R; Becker, B; Beckefeld, V. (1996) Practical aspects of route planning for magazine and newspaper wholesalers. *European Journal of Operational Research*, n. 90, p.1-12.

- Dorigo M., Maniezzo, V. e Colorni, A. (1996). Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1), p.29-41.
- Dorigo M., Di Caro, G. e Gambardella L. M. (1999). Ant Algorithms for Discrete Optimization. *Artificial Life* 5(2), p.137-172.
- Downsland, K.A. (1993). Simulated Annealing. In: *Modern Heuristic Techniques for Combinatorial Problems*. Advanced Topics in Computer Science Series, Blackwell Scientific Publications, London, p.20-69.
- Dumas, Y., Desrosiers, J. e Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, v.54, p.7-22.
- Edwards, J.S., Finlay, P.N. e Wilson, J. M. (2000). The role of OR specialists in 'do it yourself' spreadsheet development. *European Journal of Operational Research* 127, p.14-27.
- Equi, L., Gallo, G., Marziale, S., Weintraub, A. (1997). A combined transportation and scheduling problem. *European Journal of Operational Research*, v.97, p. 94-104, 1997.
- Feo, T.A. e Resende, M.G. (1989). A probabilistic heuristic for a computationally difficult set covering problem, *Operations Research Letters* 8, pp. 67-71.
- Feo, T.A. e Resende, M.G.C. (1995). Greedy randomized adaptive search procedures, *J. of Global Optimization*, v.6, pp. 109-133.
- Feriancic, G. (2005). *Modelagem Matemática do Problema de Programação de Entregas de Derivados de Petróleo*. São Paulo. 104p. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, Programa de Pós-Graduação em Engenharia de Sistemas Logísticos.
- Fischetti, M. e Toth, P. (1997). A package for locomotive scheduling, *Technical Report DEIS-OR-97-16*, University of Bologna, Italy.
- Fisher, M.L. (1981). The lagrangian relaxation method for solving integer programming problems. *Management Science*, v.27, n.1, p.1-17.
- Fisher, M.L. (1985). An applications oriented guide to lagrangian relaxation. *Interfaces*, v.15, n.2, p.10-21.
- Fisher, M.L. e R. Jaikumar (1981) A generalized assignment heuristic for vehicle routing. *Networks*, v.11, p.109-124.
- Fisher, M. L., Tang, B. e Zheng, Z. (1995) A network flow based heuristic for bulk pickup and delivery routing. *Transportation Science*, v.29, n.1, p.45-55.
- Florian, M., Bushell, G., Ferland, J., Guerin, G. e Nastansky, L. (1976). The engine scheduling problem in a railway network. *INFOR* 14, p.121-138.
- Fonseca, C.H.F. (2002). *Alocação de Atendimentos de Coleta no Transporte de Remessas Expressas em Grandes Centros Urbanos*. Dissertação (Mestrado) Depto. de Eng. de Transportes, Escola Politécnica, Universidade de São Paulo. São Paulo
- Forbes, M. A., Holt, J. N. e Watts, A. M. (1991). Exact solution of locomotive scheduling problems. *Journal of Operational Research Society* 42, p.825-831.
- Galvão, F. A. (2004). *Otimização do sistema de coleta de resíduos de biomassa de madeira para fins energéticos*. São Paulo. 196p. Dissertação (Mestrado) - Escola Politécnica da

- Universidade de São Paulo, Programa de Mestrado em Engenharia de Sistemas Logísticos.
- Galvao, F. A., Cunha, C. B. (2003). Modelagem Matemática do Problema de Coleta de Resíduos de Biomassa de Madeira para Fins Energéticos. *Revista Transportes*, v.XI, n.1. 5-11. Rio de Janeiro. ANPET.
- Galvão, F. A.; Cunha, C. B.; Gualda, N. D. F. (2003). An Application of the Analytic Hierachy Process (AHP) for Locating a Distribution Center In *The 7th International Symposium on The Analytic Hierachy Process - ISAHP* The 7th International Symposium on The Analytic Hierachy Process - ISAHP Nusa Dua, Bali, Indonesia 2003.
- Garey, M.R. e Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman e Co, New York.
- Gavish, B. e Graves, S.C. (1978). The travelling salesman problem and related problems. *Working Paper OR-078-78*, Operations Research Center, MIT, Cambridge, MA.
- Gen, M. e Cheng, R. (1997). *Genetic Algorithms and Engineering Design*, Wiley, New York, 411p.
- Gendreau, M., Hertz, A., Laporte, G. (1994) A tabu search heuristic for the vehicle routing problem. *Management Science*, v.40, p.1276-1290.
- Gendreau, M., Laporte, G., Musaraganyi, C. e Taillard, É. D. (1999) A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, v.26, p.1153-1173.
- George, V e Cunha, C.B. (2002) Metaheurísticas e sua Aplicação em Problemas de Otimização Combinatória. *Produção em Iniciação Científica da Escola Politécnica da Universidade de São Paulo*, n.1.
- Gillet, B.E. e Miller, L.R. (1974) A heuristic algorithm for the vehicle dispatch problem. *Operations Research*, v.22, p.240-249.
- Glover, F. (1977) Heuristics for Integer Programming Using Surrogate Constraints, *Decision Sciences* 8, p.156-166.
- Glover, F. (1996). Ejection Chains, Reference Structures and Alternating Path Methods for Traveling Salesman Problems. *Discrete Applied Mathematics* 65, p.223-253.
- Glover, F. e Kelly, J. (1998). Combining Simulation and Optimization for Improved Business Decisions. *Colorado Business Review*, volume LXIV, n.4.
- Glover, F e Laguna, M. (1993). Tabu Search. In: *Modern Heuristic Techniques for Combinatorial Problems*. Advanced Topics in Computer Science Series, Blackwell Scientific Publications, London, p.70-141.
- Glover, F. e Laguna, M. (1998). *Tabu Search*. Kluwer Academic Press, Boston.
- Glover, F. e Kochenberger, G. A. (2003). *Handbook of Metaheuristics*. Kluwer Academic Publishers, Boston.
- Goldberg, D.E. (1989). *Genetic algorithm in search, optimization and machine learning*. Reading, MA: Addison-Wesley.
- Golden, B.; Assad; L. Levy e F. Gheysens (1984) The Fleet Size and Mix Vehicle Routing Problem, *Computers. & Operations Research* 11 (1), p.49-66.

- Gouvêa, M.T. (1992). *Algoritmos para a resolução de um problema geral de roteamento de múltiplos veículos*. São Paulo. 188p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo.
- Greistorfer, P. e Rego, C. (2006). A simple filter-and-fan approach to the facility location problem. *Computers and Operations Research*, v.33, n.9.
- Gualda, N.D.F. (1975). *The Out-of-kilter algorithm applied to the analysis of alternative limestone transport systems*. Austin, Texas. 78 p. Dissertação (Mestrado) – The University of Austin.
- Gualda, N.D.F. (1995). *Terminais de transportes: contribuição ao planejamento e ao dimensionamento operacional*. 277p. Tese (Livre Docência) – Escola Politécnica, Universidade de São Paulo. São Paulo.
- Hakimi, S. (1964). Optimal location of switching centers and the absolute centers and medians of a graph. *Operations Research* 12, p.450-559.
- Hall, R. (2004). On the road to recovery. *OR/MS Today*, p. june/04.
- Hall, R.W.; J.G. Partyka (1997). On the road to efficiency. *OR/MS Today*, p.38-47, jun/97.
- Hall, R.W. (2004). On the road to recovery. *OR/MS Today*, v.31, n.3.
- Harik, G. e Lobo, F. (1999). *A parameter-less genetic algorithm*. Illinois, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign. Report n° ILLIGAL 99009.
- Held, M., Karp, R.M. (1971). The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming* v.1, p.6-25.
- Hertz, A., Tailard, E. e de Werra, D. (1997). A tutorial on tabu search. In: *Local Search in Combinatorial Optimization*, E.H.L Aartz e J.K. Lenstra (eds). Wiley, New York, p.121-136.
- Higgins, A., Kozan, E. e Ferreira, L. (1996). Optimal Scheduling of Trains on a Single Line Track. *Transportation Research B* 30, p.147-161.
- Higgins, A., Kozan, E. e Ferreira, L. (1997). Heuristic Techniques for Single Line Train Scheduling. *Journal of Heuristics* 3, p.43-62.
- Hindi, H. e Pienkosz, K. (1999). Efficient solution of large-scale, single-source, capacitated plant location problems. *Journal of Operational Research Society* 50, p.268-274.
- Hoffman, K. e Padberg, M. (1985). LP-based Combinatorial Problem Solving. *Annals O.R.* 4, p.145-194.
- Holland, J.H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence*. University of Michigan Press.
- Homberger, J. e Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR* 37, p.297-318.
- Horowitz, E.; S. Sahni (1978). *Fundamentals of computer algorithms*. Rockville: Computer Science Press. 622p.
- Ioachim, I., J. Desrosiers, J., Y. Dumas, M.M.Solomon e D.Villeneuve (1995) A request clustering algorithm for door-to-door handicapped transportation. *Transportation Science*, v.29, n.1, p.63-78.

- Ioannou, G., Kritikos, M. e Pratascos, G. (2001). A greedy look-ahead heuristic for the vehicle routing problem with time windows. *Journal of Operational Research Society* 52, p. 523-537.
- Ireland, P., Case, R., Fallis, J., Van Dyke, C., Kuehn, J. e Meketon, M. (2004). The Canadian Pacific Railway transforms operations research by using models to develop its operating plans. *Interfaces*, 34(1), p.5-14.
- Jaramillo, J.H., Bhadury, J. e Batta, R. (2002) On the use of genetic algorithms to solve location problems. *Computer & Operations Research*, n.29, p.761-779.
- Jaw, J.J., A.R. Odoni, H.N.Psarafitis e N.H.M.Wilson (1986) A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research B*, v.20, n.3, p.243-257.
- Johnson, D. S., Aragon, C. R., McGeogh, L. A. e Schevon, C. (1991). Optimization by simulated annealing: an experimental evaluation. Part I, graph partitioning. *Operations Research* 37(6), p.865-892.
- Johnson, E. L., Nemhauser, G.L. e Savelsbergh, M.W.P. (2000). Progress in Linear Programming-Based Algorithms for Integer Programming: An Exposition. *INFORMS Journal on Computing*, v.12, n.1, p.2-23.
- Jula, H., Dessouky, M., Ioannou, P. e Chassiakos, A. (2001) *Container movement by trucks in metropolitan networks: modeling and optimization*. Los Angeles: METTRANS - University of Southern California. (Report no.2/11_20_2001)
- Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica* 4, p.373-395.
- Kirkpatrick, S., Gelatt, C.D. e Vecchi, M.P. (1983). Optimization by Simulated Annealing. *Science*, vol 220, n. 4598, pp 671-680.
- Klincewicz, J.G. (1991) Heuristics for the p-hub location problem. *European Journal of Operations Research*, n.53, p.25-37.
- Klincewicz, J.G. (1992) Avoiding local optima in the p-hub location problem using tabu search and GRASP. *Annals of Operations Research*, n.40, p.283-302.
- Kock, T. (1999) GIS: Mapping the OR/MS World. *OR/MS Today*, ago/99.
- Koschat, M.A., Berk, G.L., Blatt, J.A., Kunz, N.M. e LePore, M.H. (2003). Newsvendors tackle the newsvendor problem. *Interfaces*, 33(3), pp.72-84.
- Kraay, D. e Harker, P. (1994). Real-time scheduling of freight networks. *Transportation Research B* 29(3), p. 213-229.
- Kroon, L.G. e Peeters, L.W. (2003). A variable trip time model for cyclic railway timetabling. *Transportation Science* 37(2), p.198-212.
- Kulturel-Konak, S., Norman, B.A., Coit, D. W. e Smith, A. E. (2004). Exploiting Tabu Search Memory in Constrained Problems. *INFORMS Journal on Computing* 16(3), p.241-250.
- Land, H.A. e Doig, A.G. (1960). An Automatic Method for Solving Discrete Programming Problems. *Econometrica* 28, p. 497-520.
- Laguna, M. e Martí, R. (2003). *Scatter Search: Methodology and Implementations in C*, Kluwer Academic Publishers: Boston.

- Laporte, G. (1992). The vehicle routing problem: an overview of exact and approximate algorithms. *European Journal of Operational Research*, v.59, n.3, p.345-358.
- Laporte, G.; M. Gendreau; J.Y. Potvin e F. Semet (2000). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research* 7(4), p.285-300.
- Li, F., Golden, B. e Wasil, E. (2005). Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers and Operations Research* 32, p. 1165-1179.
- Li, Y., Pardalos, P.M. e Resende, M.G.C. (1994). A greedy randomized adaptive search procedure for the quadratic assignment problem. In: *Quadratic assignment and related problems*, P.M. Pardalos and H. Wolkowicz, eds., DIMACS Series on Discrete Mathematics and Theoretical Computer Science, vol. 16, pp. 237-261, 1994.
- Lin, C. e Tu, C. (2000). A scenario deployment of the analytic hierarchy process through application to the taiwan radwaste disposal site selection decision. In: *The 2000 Conference on Technology and Management*. p. 143 – 155.
- Lobo, F. (1997). *C++ Implementation for the simple genetic algorithm*. Disponível em: <http://w3.ualg.pt/~flobo/sga/cpp/>. Acesso em: 30 de jul. 2002.
- Lustig, I., Marsten, R. e Shanno, D. F. (1994). Interior point methods for linear programming: Computational state of the art. *ORSA Journal on Computing*, v.6, n.1, p.1-14.
- Madsen, O.B.G., Ravn, H.F., Rygaard, J.M. (1995). A heuristic for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research*, v.60, p.193-208.
- Marchand, H. e Wolsey, L.A. (2001). Agregation and Mixed Integer Roudings to Solve MIPS. *Operations Research*, v.49, n.3, p.363-371.
- Martello, S. e Toth P. (1988). A new algorithm for the 0 - 1 knapsack problem. *Management Science*. v.34, n. 5, p.633-644.
- Martí, R., Laguna, M., Campos, V. (2004). Scatter Search vs. Genetic Algorithms: An Experimental Evaluation with Permutation Problems, In: *Adaptive Memory and Evolution: Tabu Search and Scatter Search*, Rego, C., Alidaee, B. (eds.), Kluwer Academic Publishers.
- Mau-Crimmins, T e Liberti, L. (2002) Combined integer goal programming and analytic hierachy process approach for nature reserve design: A southern Arizona case study. In: *Symposium on Model and Systems in Forestry*. Chile.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E. (1953). Equation of state calculation by fast computer machines. *Journal of Chemical Physics*, n.20, p.1087-1092.
- Michalewicz, Z. (1994). *Genetic algorithms + data structures = evolution programs*. 2.ed. Springer-Verlag, USA.
- Moscato, P. 1989. *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*. Caltech Concurrent Computation Program, C3P Report 826.
- Mourad, F. A. (2005). *O problema de roteirização de veículos com carga completa e janelas de tempo*. 88p. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, Programa de Pós-Graduação em Engenharia de Sistemas Logísticos.

- Mutarelli, F. (2004) *Modelagem de Redes de Distribuição Aplicada ao Caso de uma Editora de Revistas*. São Paulo: EPUSP, Departamento de Engenharia de Transportes. 108p. Dissertação (Mestrado).
- Nazareth, J.L. (2005). The Art and Science of 'Better'. *OR/MS Today*, v.32, n.6, p.
- Nemhauser, G.L. e Wolsey, L.A. (1988). *Integer and Combinatorial Optimization*. Wiley, New York, NY.
- Newton, H. N. (1996). *Network Design under Budget Constraints with Application to the Railroad Blocking Problem*. PhD thesis, Auburn University, Industrial and Systems Engineering Department, Auburn, AL.
- Newton, H. N., Barnhart, C. e Vance, P. M. (1998). Constructing railroad blocking plans to minimize handling costs. *Transportation Science* 32 p.330-345.
- Nou, A., Desrosiers, J. e Soumis, F. (1997). Weekly locomotive scheduling at Swedish State Railways. *Technical report G-97-35*, GERAD, Ecole des Hautes Etudes Commerciales de Montreal, Canada.
- Novaes, A.G. (1989). *Sistemas logísticos: transporte, armazenagem, e distribuição física de produtos*. São Paulo: Edgard Blücher Ltda., 1989, 372p.
- O'Kelly, M.E. (1987) A quadratic integer program for the location of interacting hub facilities. *European Journal of Operations Research*, n. 32, p.393-404.
- Osman, I. H. (2002). Preface: focussed issue on applied metaheuristics. *Computers and Industrial Engineering* 44, p.205-207.
- Owen, S.H. e Daskin, M.S. (1998). Strategic facility location: A review. *European Journal of Operational Research* 111, p.423-447.
- Panko, R.R. e Sprague Jr., R.H. (1998). Hitting the wall: errors in developing and code inspecting a simple spreadsheet model. *Decision Support Systems* 22, p.337-353.
- Pesche, E. e Glover, F. (1997). TSP Ejection Chains. *Discrete Applied Mathematics*, p.165-181.
- Pirkull, H. (1987). Efficient algorithm for the capacitated concentrator location problem. *Computers and Operations Research* 14, p.197-208.
- Potvin, J. Y. e Bengio, S. (1996). The vehicle routing with time windows – Part II: Genetic search. *INFORMS Journal on Computing* 8, p.165-172.
- Psaraftis, H.N. (1980). A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, v.14, n.2, p.130-154.
- Psaraftis, H.N. (1983). k-Interchange procedures for local search in a precedence-constrained routing problem. *European Journal of Operational Research*, v.13, p.391-402.
- Pureza, V. e França, P. M.. (2001). Uma Abordagem Adaptativa de Busca Tabu Aplicada ao Problema de Roteamento de Veículos. *Transportes*, v. 9, n. 2, p. 28-47, 2001.
- Ramani, V. (1981). An information system for allocating coach stock on Indian Railways. *Interfaces* 11(3), p.44-51.
- Ratliff, H.D. e Nulty, W.G. (1996). Logistics Composite Modeling. *Technical White Paper Series*. The Logistics Institute at Georgia Tech.

- Reeves, C.R. (1993). *Modern heuristics techniques for combinatorial problems*. Oxford: Blackwell Scientific Publications.
- Reeves, C. R. (2003). Genetic Algorithms. In: *Handbook of Metaheuristics*, F. Glover e G.A. Kochenberger (eds). Kluwer, Boston, p.55-82.
- Rego, C., Roucariol, C. (1996) A parallel tabu search algorithm using ejection chains for the vehicle routing problem. In: *Meta-Heuristics: Theory and Applications*, I. H. Osman e J. P. Kelly (eds). Kluwer, Boston, p.661-675.
- Resende, M.G.C. e Ribeiro, C. C. (2003). Greedy Randomized Adaptive Search Procedures. In: *Handbook of Metaheuristics*, F. Glover e G.A. Kochenberger (eds). Kluwer, Boston, p.219-250.
- Resende, M.G.C. e Ribeiro, C. C. (2005) GRASP with path-relinking: Recent advances and applications. In: *Metaheuristics: Progress as Real Problem Solvers*, T. Ibaraki, K. Nonobe e M. Yagiura, (Eds.), Springer, p. 29-63, 2005,
- Resende, M.G.C. e Werneck, R.F. (2004). A Hybrid Heuristic for the p-Median Problem. *Journal of Heuristics*, 10(1), p.59-88.
- Resende, M.G.C. e Werneck, R.F. (2005). A Hybrid Multistart Heuristic for the Uncapacitated Facility Location Problem. Aceito para publicação no *European Journal of Operational Research*.
- ReVelle, C.S. e Eiselt, H.A. (2005). Location analysis: A synthesis and survey. *European Journal of Operational Research* 165, p.1-19.
- Rivier R. E. e Tzieropoulos, P. (1984). Interactive Graphic Models for Railway Operational Planning. In: *The Practice of Transportation Planning*, M. Florian (ed.), Elsevier Science Publishers, Amsterdam, p.245-259.
- Rivier R. E. e Tzieropoulos, P. (1987). Computer-Aided Planning of Railway Networks, Lines and Stations. In: *Computers in Railway Management*, T.K.S. Murthy, L. S Lawrence and R. E. Rivier (eds.), Computational Mechanics Publications, Berlin, p.3-16.
- Robinson Jr., E. P., Gao, L.L. e Muggenborg. S.D. (1993). Designing an Integrated Distribution System at Dow Brands, Inc. *Interfaces* 23(3), p.107-117.
- Rochat, Y., Taillard, É. D. (1995) Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, v.1, p.147-167.
- Ronen, D. (1988). Perspectives on practical aspects of truck routing and scheduling. *European Journal of Operational Research* 35(2), p.137-145.
- Russel, R. e Chiang, W. (2004). Scatter search for the vehicle routing problem with time windows. *European Journal of Operational Research*, v.169, n.2, p.606-622.
- Saaty, T. L.. (1991) *Método de análise hierárquica*., São Paulo: Makron. 367p.
- Saaty, T.L.. (1994) How to make a decision: The Analytic Hierarchy Process. *Interfaces*, v.24, n.6. p.19-43.
- Şahin, İ. (1999). Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Research B* 33(7), p. 511-534.
- Şahin, G., Ahuja, R. K. e Cunha, C. B. (2005). New approaches for the train dispatching problem. *submetido para publicação no Transportation Science*.

- Sauder, R.L. e Westerman, W. (1993). Computer Aided Train Dispatching: Decision Support Through Optimization. *Interfaces* 13(6), p.24-37.
- Shapiro, J.F. (1979). A survey of lagrangean techniques for discrete optimization. *Annals of Discrete Mathematics*, v.5, p.113-138.
- Silva, G. P. (2001). *Uma metodologia baseada na técnica de geração de arcos para o problema de programação de veículos*. 146p. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de Transportes.
- Silva, M. R. (2004). *Uma contribuição ao problema de localização de terminais de consolidação no transporte de carga parcelada*. 85p. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de Transportes.
- Silva, M. R. e Cunha, C. B. . (2003). Uma Aplicação de Algoritmos Genéticos para a Localização de Terminais de Carga Parcelada. In: *XVII Congresso de Pesquisa e Ensino em Transportes*, v. 2. p. 644-656.
- Silva, M. R. e Cunha, C. B. . (2004). Configuração de redes do tipo hub-and-spoke para o transporte rodoviário de carga parcelada no Brasil utilizando algoritmos genéticos. *Transportes*, v. XII, n. 1, p. 23-31.
- Silver, E.A. (2004). An overview of heuristic solution methods. *Journal of the Operational Research Society* 55(9), p.936-956.
- Simchi-Levy, D., Kaminsky, P. e Simchi-Levy, E. (2002). *Designing and managing the supply chain: concepts, strategies and case studies*. McGraw Hill, Irwin, CA.
- Skorin-Kapov, D. e Skorin-Kapov, J. (1994) On tabu search for the location of interacting hub facilities. *European Journal of Operations Research*, n.73, p.502-509.
- Skorin-Kapov, D., Skorin-Kapov, J. e O’Kelly, M. (1996) Tight linear programming relaxations of uncapacitated p-hub median problem. *European Journal of Operations Research*, n.94, p.582-593.
- Smith, S. e Sheffi, Y. (1988). Locomotive scheduling under uncertain demand. *Transportation Research Record* 1251, p.45-53.
- Solomon, M. M. (1986). On the worst-case performance of some heuristics for the vehicle routing and scheduling with time windows constraints. *Networks* 16, p.161-174.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v.35, n.2, p.254-265.
- Sohn, J. e Park, S. (1998) Efficient solution procedure and reduced size formulations for p-hub location problems. *European Journal of Operations Research*, n.108, p.118-126.
- Song, S.H.; Lee, K.S.; Kim, G.S. (2002). A practical approach to solving a newspaper logistics problem using a digital map. *Computers & Industrial Engineering*, n. 43, p. 315-330.
- Souza, P.S. (1993). *Asynchronous organizations for multi-algorithms problems*. Pittsburgh: Carnegie Mellow University, Department of Electrical and Computer Engineering. 139p. (Tese de Doutorado).
- Souza, E. C. (1999) *Modelagem e resolução de um problema de transporte do tipo: “carga única - coleta e entrega” com janelas de tempo*. São Paulo. 89p. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia Naval e Oceânica.

- Stringher, F. G. (2004) *Designação de rotas para frota dedicada em uma rede de distribuição de linha branca*. São Paulo. 90p. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de Transportes.
- Swait, J.D. (1991). *Fundamentos computacionais – Algoritmos e estrutura de dados*. São Paulo: Makron, McGraw-Hill. 295p.
- Taillard, É. D. (1991) Robust taboo search for the quadratic assignment problem. *Parallel Computing*, v.17, p.433-445.
- Taillard, É. D. (1993) Parallel iterative search methods for vehicle routing problems. *Networks*, v.23, p.661-673.
- Taillard, É. D., Badeau, P., Gendreau, M., Guertin, F. e Potvin, J. Y. (1997). A tabu search for the vehicle routing problem with soft time windows. *Transportation Science* 31, p. 170-186.
- Taillard, É. D., Gambardella, L. M., Gendreau, M. e Potvin, J.Y. (2001). Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research* v.135, p.1-16.
- Tan, K. C., Lee, L. H., Zhu, Q.L. e Ou, K. (2001). Heuristic methods for vehicle routing problems with time windows. *Artificial Intelligence in Engineering*, 15, p.281-295.
- Teixeira, R.G. (2001). *Heurísticas para o Problema de Dimensionamento e Roteirização de uma Frota Heterogênea utilizando o Algoritmo Out-of-kilter*. 118p. Dissertação (Mestrado) – Departamento de Engenharia de Transportes, Escola Politécnica, Universidade de São Paulo.
- Teixeira, R.G.; Cunha, C.B. (2002) Heurísticas para o Problema de Dimensionamento e Roteirização de uma Frota Heterogênea Utilizando o Algoritmo Out-of-Kilter. *Revista Transportes*, v.10, n.2, p.9-30.
- Teo, T.S.H. e Tan, M. (1999) Spreadsheet development and ‘what-if’ analysis: quantitative versus qualitative errors. *Accounting Management and Information Technologies* 9, p.141-160.
- Thompson, P.M. e Orlin, J.B. (1989). The theory of cyclic transfers. *MIT Operations Research Report* 200-89, Cambridge, MA.
- Thompson, P.M. e Psaraftis, H.N. (1993) Cyclic transfer algorithms for multivehicle routing and scheduling problems. *Operations Research* v.41, p.935-946.
- Toth, P., Vigo, D. (1996). Fast local search algorithms for the handicapped persons transportation problem. In: *Meta-Heuristics: Theory and Applications*, I. H. Osman, J. P. Kelly (eds.), Kluwer, Norwell, p.677-690.
- Toth, P. e D. Vigo, D. (1997). Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, v.31, n.1, p.60-71.
- Toth, P., Vigo, D. (2003). The granular tabu search and its application to the vehicle-routing problem. *Journal on Computing*, v.15, n.4, p.333-346.
- Van Vliet, A., Boender, C.G., Rinnooy Kan, A.H.G. (1992). Interactive optimization of bulk sugar deliveries. *Interfaces*, v.22, n.3, p. 4-14.
- Weintraub, A., Epstein, R., Morales, R., Seron, J., Traverso, P. (1996). A truck scheduling system improves efficiency in the forest industries. *Interfaces*, v.26, n.4, p.1-12.

- Wren, A. e Holliday, A. (1972) Computer scheduling of vehicles from one or more depots to a number of delivery points. *Operational Research Quarterly*, v.23, p.333-344.
- Wright, M. B. (1989). Applying stochastic algorithms to a locomotive scheduling problem. *Journal of Operational Research Society*, 40, p.187-192.
- Wu, L. e Cunha, C. B. (2005). Uma heurística baseada em algoritmos genéticos para o problema de roteirização periódica In: *Panorama Nacional da Pesquisa em Transportes - XIX Congresso de Pesquisa e Ensino em Transportes*, v.II, p.1556- 1567.
- Xu, J. e Kelly, J. P. (1996) A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science*, v.30, p.379-393.
- Yoshizaki, H.T.Y. (2002). *Projetos de Redes de Distribuição Física Considerando a Influência do Imposto de Circulação de Mercadorias e Serviços*. Tese de Livre-docência. Departamento de Engenharia de Produção, Escola Politécnica da Universidade de São Paulo. 144p.
- Yu, G. (1998). *Operations Research in the Airline Industry*. Kluwer Academic Publishers, Boston, MA.
- Ziarati, K., Soumis, F., Desrosiers, J., Gelinas, S. e Saintonge, A. (1997). Locomotive assignment with heterogeneous consists at CN North America. *European Journal of Operational Research* 97, p.281-292.
- Ziarati, K, F Soumis, F., Desrosiers, J e Solomon, M. M. (1999). A branch-first, cut-second approach for locomotive assignment, *Management Science* 45, p.1156-1168.
- Znamensky, A. (2000). *Um modelo para a roteirização e programação do transporte de deficientes*. São Paulo. 144p. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de Transportes.
- Znamesnki, A.; Cunha, C.B. (2001) Heurísticas para o problema de roteirização e programação do transporte de deficientes. In: *Transporte em Transformação V- Trabalhos Vencedores do Prêmio CNT Produção Acadêmica 2000. Problemas e soluções dos transportes no Brasil*. São Paulo: Makron Books do Brasil, São Paulo, SP, p.1-12.
- Znamesnki, A.; Cunha, C. B. (2003). O problema de estoque-roteirização com demanda determinística. *Revista Transportes*, v.XI, n.2, p.31-40.