

UNIVERSIDADE ESTADUAL DE CAMPINAS – UNICAMP

Faculdade de Engenharia Elétrica e de Computação

Departamento de Engenharia de Sistemas – DENSIS

Área de Concentração: Automação

**BUSCA TABU PARA O PROBLEMA DE ROTEAMENTO
DINÂMICO DE VEÍCULOS COM JANELAS DE TEMPO**

RODRIGO MORETTI BRANCHINI

Orientador: Prof. Vinícius Amaral Armentano

Comissão julgadora:

Cláudio Barbieri da Cunha	EPUSP – Departamento da Eng. de Transportes
Débora Pretti Ronconi	EPUSP – Departamento de Eng. de Produção
Paulo A. Valente Ferreira	UNICAMP – FEEC
Vinícius Amaral Armentano	UNICAMP – FEEC

Tese apresentada à Faculdade de
Engenharia Elétrica e de Computação da
Universidade Estadual de Campinas –
UNICAMP, como parte dos requisitos
exigidos para obtenção do título de
MESTRE EM ENGENHARIA ELÉTRICA.

Campinas, julho de 2005.

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Branchini, Rodrigo Moretti

B732b Busca tabu para o problema de roteamento dinâmico de
veículos com janelas de tempo / Rodrigo Moretti Branchini.
--Campinas, SP: [s.n.], 2005.

Orientador: Vinícius Amaral Armentano

Dissertação (Mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Caminhões - Rotas. 2. Pesquisa operacional. 3.
Heurística. I. Armentano, Vinícius Amaral. II. Universidade
Estadual de Campinas. Faculdade de Engenharia Elétrica e
de Computação. III. Título.

Título em Inglês: Tabu search for the dynamic vehicle routing problem with time
windows.

Palavras-chave em Inglês: Truck routes, Heuristic, Operational research.

Área de concentração: Automação.

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Cláudio Barbieri da Cunha, Débora Pretti Ronconi e Paulo A
Valente Ferreira.

Data da defesa: 05/07/2005

Resumo

O problema de roteamento dinâmico de veículos com janelas de tempo (PRDVJT) decorre do problema de roteamento estático de veículos com janelas de tempo (PRVJT). No problema estático, o objetivo é atender todos os clientes dentro de intervalos de tempo previamente estipulados, respeitando as restrições de capacidade dos veículos e minimizando o número de veículos e a distância percorrida. Neste problema, todos os dados necessários para a definição das rotas são conhecidos a priori. No PRDVJT, algumas informações, por exemplo, a localização e a demanda de alguns clientes, são reveladas dinamicamente ao tomador de decisão, ao mesmo tempo em que os motoristas percorrem seus roteiros. Este trabalho estuda e desenvolve métodos para os problemas PRVJT e PRDVJT. Para o PRVJT são testadas heurísticas construtivas, incluindo a proposta de uma heurística focada na redução do número de rotas, e métodos de busca local e tabu baseados na vizinhança granular. Posteriormente, as heurísticas são adaptadas e aplicadas ao PRDVJT, avaliando-se os benefícios de desvio do destino planejado e de inclusão de estratégias de espera e de posicionamento de veículos. Os testes computacionais do PRDVJT são executados num ambiente de simulação baseado em características práticas levantadas numa pesquisa a campo com empresas de transportes da região de Campinas e São Paulo.

Palavras-chave: roteamento de veículos com janelas de tempo, roteamento dinâmico, heurísticas, busca local granular, busca tabu.

Abstract

The dynamic vehicle routing problem with time windows (DVRPTW) originates from the static vehicle routing problem with time windows (VRPTW). The objective of the static problem is to serve a given set of clients within their time window constraints, subject to each vehicle capacity, minimizing the number of vehicles and the distance traveled. In this problem, all data needed to create routes is previously known. In the DVRPTW, some information, for instance, the positioning and demand of some clients, is dynamic revealed for the decision maker at the same time in which the drivers are executing their routes. This work studies and develops methods for the VRPTW and DVRPTW. Constructive heuristics, as well as a suggestion of an heuristic focused on route minimization, are discussed for the VRPTW. Also for the same problem, local and tabu search methods based on the granular neighborhood are tested. Afterward, these heuristics are adapted and applied for the DVRPTW, evaluating the benefits of diversion and waiting and positioning strategies. Computational experiments for the DVRPTW are performed with a simulation environment founded on practical attributes collected during interviews with transportation companies of the Campinas and São Paulo region.

Keywords: vehicle routing with time windows, dynamic routing, heuristics, granular local search, tabu search.

*À minha mãe Luiza, meu pai Bira,
meu irmãozinho Diogo e
à querida Josi, com todo o amor.*

Ao Guerreiro, com saudades.

Agradecimentos

Gostaria de agradecer a todos que contribuíram para este trabalho e ajudaram, de diferentes formas, a sua realização. Em especial:

- Ao Prof. Vinícius, pela orientação, crítica e discussões, por me introduzir à área de otimização e me mostrar que nem tudo é tão simples quanto parece, pela paciência e companheirismo nos momentos difíceis;
- Aos meus pais e meu irmão, por tudo que fizeram e ainda fazem por mim. Impossível retribuir;
- À Josi, por todo apoio, compreensão, carinho e amor, tornando tudo mais prazeroso, divertido e feliz;
- Ao Elias, pela amizade, companheirismo nos dias de PED e ajuda na implementação;
- Ao Marcelo Mitsumoto, pela contribuição nos programas em C++;
- Ao Leandro, pelas discussões e companhia em EA-954;
- Ao Anibal, por todas as dicas e apoio nesses últimos anos;
- Ao Vinícius Jacques, pela paciência e auxílio no mundo Linux;
- Ao Wilfredo, pela amizade e parceria em não-linear;
- Ao André e Olinto, por facilitar a introdução ao C++;
- Ao Sacchi, pela amizade e ajuda com probabilidade, MySQL e programação;
- Ao Prof. Hermano, pela agradável convivência, lições de didática e por me passar um pouco de sua preocupação com a qualidade do ensino;
- À Débora Ronconi, por me receber tão bem e me indicar a um professor excelente;
- À Márcia Fracaro, pela assistência nos dias de “correria”;
- Ao Washington da Coordenação de Graduação, pela ágil administração do programa PED;
- À turma da Unibanda que acrescentaram harmonia ao meu primeiro ano de Unicamp;
- A todos os professores da FEEC, pelo tanto que me ensinaram;
- A todos do laboratório DENSIS, pela ajuda e paciência em ceder computadores para a execução dos testes computacionais;
- À equipe técnica do DENSIS e do GRSD, pelo suporte e flexibilidade para uso de computadores pessoais no laboratório;
- Ao revisor(a) dos relatórios FAPESP, pelas críticas e contribuições nesta dissertação;
- Ao pessoal da segurança, Azevedo, Beth, Fernando e colegas, companheiros de tardes da noite e dos finais de semana;
- A todos os funcionários da Unicamp que possibilitam o desenvolvimento de nossas atividades;
- A todos que colaboraram com informações sobre operações de transporte, principalmente ao Dálton Vecchini (Transportadora Americana), Marco Aurélio e Leandro Santi (Mercúrio), Sérgio Ferreira (Atlas), Marcos Barnstein (Kieling) e representantes das empresas Transportadora Minuano e Correios;
- À comissão julgadora, pela instigante discussão e valiosos comentários;
- À FAPESP, pelo apoio financeiro.

Conteúdo

Capítulo 1. Introdução.....	1
1.1. Visão geral sobre logística e problemas de roteamento estáticos e dinâmicos	1
1.2. Entendimento dos problemas de roteamento no Brasil – pesquisa a campo.....	3
1.2.1. Descrição das empresas.....	3
1.2.2. Descrição das operações de coleta, estocagem e entrega.....	4
1.2.3. Atendimento aos clientes	8
1.2.4. Tecnologia de rastreamento / monitoramento	9
1.2.5. Ferramentas de roteamento	12
1.3. Objetivos desta dissertação e atividades desenvolvidas.....	18
Capítulo 2. Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT).....	19
2.1. Apresentação do PRVJT e formulação matemática	19
2.2. Revisão dos métodos heurísticos da literatura	21
2.2.1. Problema de Roteamento de Veículos (PRV)	21
2.2.2. Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT)	22
2.3. Heurísticas construtivas.....	24
2.3.1. Heurística construtiva de Clarke & Wright (1964)	24
2.3.2. Heurística construtiva de Solomon (1987).....	25
2.3.3. Heurística construtiva proposta – Redução de Rotas (RR).....	26
2.3.4. Experimentos computacionais.....	35
A) Considerações sobre a implementação: estrutura de dados e movimentos.....	35
B) Conjunto de problemas teste	36
C) Resultados dos experimentos computacionais	37
2.4. Busca local granular	41
2.4.1. Descrição da vizinhança granular de Toth e Vigo (2003).....	41
2.4.2. Busca local proposta.....	45
2.4.3. Resultados dos experimentos computacionais	46
2.5. Busca tabu granular	51
2.5.1. Vizinhanças e movimentos.....	51
2.5.2. Memória de curto prazo e atributo tabu	52

2.5.3. Critério de aspiração.....	55
2.5.4. Memória de longo prazo, intensificação e diversificação	55
2.5.5. Memorização de soluções visitadas – conjunto de elite.....	57
2.5.6. Reinício a partir do conjunto de elite – baseado em Rochat e Taillard (1995)	58
2.5.7. Critério de parada	59
2.5.8. Descrição dos métodos heurísticos baseados na busca tabu granular	59
A) <i>Tabu_simples</i>	60
B) <i>Tabu_LP</i>	60
C) <i>Tabu_reinício</i>	61
D) <i>Tabu_misto</i>	62
2.5.9. Resultados dos experimentos computacionais	63
2.6. Conclusões para o PRVJT.....	75
Capítulo 3. Problema de Roteamento Dinâmico de Veículos com Janelas de Tempo (PRDVJT) 77	
3.1. Descrição do PRDVJT	77
3.2. Revisão dos métodos heurísticos da literatura	78
3.3. Métodos heurísticos para o PRDVJT	87
3.4. Descrição do ambiente de teste para o PRDVJT.....	89
3.4.1. Informações sobre a empresa de transporte	89
3.4.2. Informações sobre os clientes	91
3.4.3. Informações sobre os veículos e contribuição ao lucro da operação de coleta	104
3.4.4. Desenvolvimento do ambiente de teste para o PRDVJT	107
3.5. Adaptações dos métodos do problema estático para o dinâmico Resultados dos experimentos computacionais.....	113
3.5.1. Adaptações gerais a todos os métodos	113
3.5.2. Adaptações da heurística construtiva	115
3.5.3. Adaptações da Busca Local Granular	119
3.5.4. Adaptações da Busca Tabu Granular	128
3.5.5. Inclusão das estratégias de espera e posicionamento	130
3.6. Resultados dos experimentos computacionais	134
3.6.1. <i>LS_BL</i> e <i>LS_BT</i>	135

3.6.2. Vizinho mais próximo	136
3.6.3. <i>Inserção</i>	139
3.6.4. <i>Inserção+</i> e <i>Busca Tabu</i>	140
3.7. Conclusões para o PRDVJT	145
Capítulo 4. Conclusões gerais e sugestões para pesquisa futura	147
Bibliografia	149
Anexos	157
A. Guia de entrevista para a pesquisa a campo	157
B. Heurística de Rochat e Taillard (1995) para o PRVJT	159
C. Estratégias para problemas dinâmicos – Bianchi (2000).....	160
D. Detalhamento sobre desvio	161
E. Busca tabu de Taillard et al. (1997).....	163
F. Busca tabu de Gendreau et al. (1999)	164
G. Movimentos da busca local granular.....	164
H. Vizinhança cadeias de ejeções de Glover (1991, 1996).....	183
I. Melhores resultados da literatura para o conjunto de problemas teste do PRVJT.....	185
J. Estimativa dos custos fixo e variável de um veículo	191
K. Pseudocódigos	194
K.1. Construção do grafo granular:	194
K.2. Busca local no grafo granular:.....	195
K.3. Regras de entrada de uma nova solução no conjunto de elite	196
L. Relatório gerado pelo sistema de rastreamento – roteiro de um veículo	198

Lista de Figuras

Figura 1.1. Esquema do <i>layout</i> do depósito das transportadoras	7
Figura 1.3. Detalhamento de uma rota	17
Figura 2.1. Ilustração de incompatibilidade positiva entre dois clientes	29
Figura 2.2. Ilustração de incompatibilidade negativa entre dois clientes.....	30
Figura 2.3. Análise da distância média e dispersão com Cliente A no centro do plano	32
Figura 2.4. Análise da distância média e dispersão com Cliente B próximo ao canto superior direito do plano	33
Figura 2.5. Exemplo de movimento da vizinhança troca-cross	35
Figura 2.6. Exemplo de <i>crossover</i>	42
Figura 2.7. Exemplo de inserção de um cliente	43
Figura 2.8. Exemplo de inserção de dois clientes	44
Figura 2.9. Exemplo de troca de um par de clientes	45
Figura 2.10. Percentual médio de rotas adicionais e tempo computacional por percentual do grafo granular	49
Figura 2.11. Desvio percentual médio da distância e tempo computacional por percentual do grafo granular	50
Figura 2.12. Evolução da qualidade dos resultados com a adição de componentes da busca tabu	69
Figura 3.1. As 5 regiões do espaço com diferentes probabilidades de ocorrência de pedidos – Problema Teste A	92
Figura 3.2. Disposição no espaço dos 1101 clientes do Problema Teste A	93
Figura 3.3. Número médio de clientes por rota das empresas de transporte.....	94
Figura 3.4. As 5 regiões do espaço com diferentes probabilidades de ocorrência de pedidos – Problema Teste B	95
Figura 3.5. As 5 regiões do espaço com diferentes probabilidades de ocorrência de pedidos – Problema Teste C	96
Figura 3.6. Disposição no espaço dos 1060 clientes do Problema Teste B	97
Figura 3.7. Distribuição do percentual de PDC^+ no período entre 8hs e 14hs.....	99
Figura 3.8. Distribuição do percentual de PDC^- no período entre 14hs e 18hs	100
Figura 3.9. Etapas de recebimento de pedidos e confirmação de atendimento.....	107

Figura 3.10. Etapas de movimentação dos veículos e rastreamento das coletas.....	109
Figura 3.11. Diagrama de rótulos dos veículos	111
Figura 3.12. Otimização em tempo real dos roteiros	112
Figura 3.13. Número de clientes atendidos por veículo, método inserção, problema teste A	115
Figura 3.14. Disposição dos clientes-semente – RR original – Problema teste A	116
Figura 3.15. Disposição dos clientes-semente – RR original – Problema teste B	117
Figura 3.16. Disposição dos clientes-semente – RR modificado – Problema teste A	118
Figura 3.17. Disposição dos clientes-semente – RR modificado – Problema teste B.....	119
Figura 3.18. Tempo para percorrer uma iteração da LS_BL – Problema teste A.....	120
Figura 3.19. Número de movimentos por faixa de ganho – Problema teste A	121
Figura 3.20. Número de movimentos por tamanho do grafo – Problema teste C.....	122
Figura 3.21. Acréscimo ao lucro por tamanho do grafo – Problema teste C	123
Figura 3.22. Variação de η no método <i>Inserção+</i> – Problema teste A.....	125
Figura 3.23. Variação do tempo para uma iteração do método <i>Inserção+</i> – Problema teste A	126
Figura 3.24. Número de iterações – método <i>Inserção+</i> , Problema teste A	127
Figura 3.25. Número de atualizações da incumbente – método <i>Inserção+</i> , Problema teste A	128
Figura 3.26. Veículos em espera e clientes recusados – método <i>Inserção+</i> , Problema teste B	131
Figura 3.27. Veículos em espera e clientes recusados – método <i>I+_e_p</i> , Problema teste B	133
Figura 3.28. Infra-estrutura utilizada na execução dos testes para o PRDVJT	134
Figura 3.29. Posição e espera dos veículos, <i>VMP_e</i> , problema teste B	138
Figura 3.30. Comparação dos métodos – Problema Teste A	142
Figura 3.31. Comparação dos métodos – Problema Teste B	142
Figura 3.32. Comparação dos métodos – Problema Teste C	143
Figura 3.33. Comparação dos custos (em R\$) dos métodos <i>VMP_e</i> e <i>I+_e_d_p</i> , Prob. Teste C	144
Figura 6.1. Exemplo de problema dinâmico de roteamento de veículos com janelas de tempo.....	162
Figura 6.2. Roteamento dinâmico sem e com possibilidade de desvio do destino atual	162
Figura 6.3. Movimento intra-rota troca-arcos – 1ª opção	165
Figura 6.4. Movimento intra-rota troca-arcos – 2ª opção	166
Figura 6.5. Movimento intra-rota troca-arcos com ‘a’ igual ao depósito – 1ª opção.....	167
Figura 6.6. Movimento intra-rota troca-arcos com ‘a’ igual ao depósito – 2ª opção.....	167
Figura 6.7. Movimento intra-rota inserção com $x = b$	169

Figura 6.8. Movimento intra-rota inserção com $x = 5$	169
Figura 6.9. Movimento inter-rota crossover – Opção 1	171
Figura 6.10. Movimento inter-rota <i>crossover</i> – Opção 2	172
Figura 6.11. Movimento inter-rota <i>crossover</i> – Opção 3	173
Figura 6.13. Movimento inter-rota inserção – Opção 1 com $x = b$	176
Figura 6.14. Movimento inter-rota inserção – Opção 1 com $x = 6$	176
Figura 6.15. Movimento inter-rota inserção – Opção 2 com $x = a$	178
Figura 6.16. Movimento inter-rota inserção – Opção 2 com $x = 1$	178
Figura 6.17. Movimento inter-rota troca – Opção 1 com $fima = Pa$ e $fimb = b$	180
Figura 6.18. Movimento inter-rota troca – Opção 1 com $fima = 2$ e $fimb = 6$	180
Figura 6.19. Movimento inter-rota troca – Opção 2 com $inicioa = Aa$ e $fimb = b$	182
Figura 6.20. Movimento inter-rota troca – Opção 2 com $inicioa = Aa$ e $fimb = 5$	182
Figura 6.21. Exemplo de cadeia de ejeções para $l = 2$	184

Lista de Tabelas

Tabela 1.1. Problemas de roteamento levantados na pesquisa a campo	5
Tabela 2.1. Número de problemas teste por tipo e número de clientes.....	37
Tabela 2.2. Testes das heurísticas construtivas	37
Tabela 2.3. Parâmetros dos testes da heurística RR.....	38
Tabela 2.4. Percentual de veículos adicionais sobre o melhor resultado da literatura – Heurísticas Construtivas para os 56 problemas teste do Solomon (1987)	38
Tabela 2.5. Desvio percentual médio da distância percorrida – Heurísticas Construtivas para os 56 problemas teste do Solomon (1987).....	39
Tabela 2.6. Tempo médio computacional em segundos – Heurísticas Construtivas para os 56 problemas teste do Solomon (1987).....	40
Tabela 2.7. Percentual de veículos adicionais sobre o melhor resultado da literatura – Busca local granular para os 56 problemas teste do Solomon (1987)	47
Tabela 2.8. Desvio percentual médio da distância percorrida – Busca local granular para os 56 problemas teste do Solomon (1987).....	47
Tabela 2.9. Tempo médio computacional em segundos – Busca local granular para os 56 problemas teste do Solomon (1987)	47
Tabela 2.10. Movimentos da busca tabu	51
Tabela 2.11. Hierarquia de movimentos executados pela busca tabu	52
Tabela 2.12. Ciclo dos valores das faixas das durações tabu ($perc_arestas_candidatas \geq 10\%$).....	54
Tabela 2.13. Ciclo dos valores das faixas das durações tabu ($perc_arestas_candidatas < 10\%$)	54
Tabela 2.14. Composição dos métodos baseados na busca tabu granular	59
Tabela 2.15. Versões da busca <i>Tabu_simples</i>	60
Tabela 2.16. Versões da busca <i>Tabu_LP</i>	61
Tabela 2.17. Versões da busca <i>Tabu_reinício</i>	62
Tabela 2.18. Versões da busca <i>Tabu_misto</i>	62
Tabela 2.19. Porcentagem de veículos adicionais – <i>Tabu_simples</i>	63
Tabela 2.20. Desvio % médio na distância – <i>Tabu_simples</i>	63
Tabela 2.21. Tempo médio de execução (segundos) – <i>Tabu_simples</i>	63
Tabela 2.22. Tempo médio até a melhor solução (segundos) – <i>Tabu_simples</i>	64

Tabela 2.23. Porcentagem de veículos adicionais – <i>Tabu_LP</i>	65
Tabela 2.24. Desvio % médio na distância– <i>Tabu_LP</i>	65
Tabela 2.25. Tempo médio de execução (segundos) – <i>Tabu_LP</i>	65
Tabela 2.26. Tempo médio até a melhor solução (segundos) – <i>Tabu_LP</i>	65
Tabela 2.27. Porcentagem de veículos adicionais – <i>Tabu_reinício</i>	66
Tabela 2.28. Desvio % médio na distância– <i>Tabu_reinício</i>	66
Tabela 2.29. Tempo médio de execução (segundos) – <i>Tabu_reinício</i>	66
Tabela 2.30. Tempo médio até a melhor solução (segundos) – <i>Tabu_reinício</i>	67
Tabela 2.31. Porcentagem de veículos adicionais – <i>Tabu_misto</i>	67
Tabela 2.32. Desvio % médio na distância – <i>Tabu_misto</i>	67
Tabela 2.33. Tempo médio de execução (segundos) – <i>Tabu_misto</i>	68
Tabela 2.34. Tempo médio até a melhor solução (segundos) – <i>Tabu_misto</i>	68
Tabela 2.35. Comparação entre o <i>Tabu_misto_4_fcnv_CE10</i> e o método de Cordeau <i>et al.</i> (2001)	70
Tabela 2.36. Porcentagem de veículos adicionais – Variantes por solução de partida e vizinhança ..	71
Tabela 2.37. Desvio % médio na distância – Variantes por solução de partida e vizinhança	72
Tabela 2.38. Tempo médio de execução (segundos) – Variantes por solução de partida e vizinhança	72
Tabela 2.39. Tempo médio até a melhor solução (segundos) – Variantes por solução de partida e vizinhança	72
Tabela 2.40. Comparação entre o <i>Tabu_misto_4_fcnv_CE10</i> e <i>benchmarks</i> da literatura – número médio de veículos para os problemas de 100 clientes.....	73
Tabela 2.41. Comparação entre o <i>Tabu_misto_4_fcnv_CE10</i> e <i>benchmarks</i> da literatura – distância média para os problemas de 100 clientes.	73
Tabela 2.42. Comparação entre o <i>Tabu_misto_4_fcnv_CE10</i> , Bräysy (2003) e Homberger (2001) – número médio de veículos para os problemas de 200 clientes.	73
Tabela 2.43. Comparação entre o <i>Tabu_misto_4_fcnv_CE10</i> , Bräysy (2003) e Homberger (2001) – distância média para os problemas de 200 clientes.....	74
Tabela 2.44. Tempo médio de execução (segundos) – <i>Tabu_misto_4_fcnv_CE10</i>	74
Tabela 2.45. Tempo médio até a melhor solução (segundos) – <i>Tabu_misto_4_fcnv_CE10</i>	74
Tabela 3.1. Evolução da pesquisa sobre problemas dinâmicos.....	86
Tabela 3.2. Testes para o PRDVJT	88
Tabela 3.3. Modalidades de turnos admitidos pelo ambiente de simulação	89

Tabela 3.4. Probabilidade de ocorrência de clientes por região – Problema Teste A	92
Tabela 3.5. Probabilidade de ocorrência de clientes por região – Problemas Teste B e C	96
Tabela 3.6. Percentual de pedidos por tipo por problema teste.....	99
Tabela 3.7. Distribuição de Poisson para uma taxa média de 0.067 clientes / segundo	101
Tabela 3.8. Distribuição da duração da janela de tempo ($dura\tilde{c}\tilde{a}\tilde{o}_i$).....	102
Tabela 3.9. Política de multas por $dura\tilde{c}\tilde{a}\tilde{o}_i$	103
Tabela 3.10. Distribuições da $receita_i$, d_i e s_i	103
Tabela 3.11. Custos fixo e variável de um veículo	105
Tabela 3.12. Adaptações ao tratar o problema dinâmico	113
Tabela 3.13. Resultados dos métodos LS_BL e LS_BT – Problema teste A	135
Tabela 3.14. Resultados dos métodos LS_BL e LS_BT – Problema teste B	135
Tabela 3.15. Resultados dos métodos LS_BL e LS_BT – Problema teste C	135
Tabela 3.16. Resultados dos métodos VMP , VMP_e e VMP_e_p – Problema teste A.....	136
Tabela 3.17. Resultados dos métodos VMP , VMP_e e VMP_e_p – Problema teste B	137
Tabela 3.18. Resultados dos métodos VMP , VMP_e e VMP_e_p – Problema teste C	137
Tabela 3.19. Resultados do método <i>Inserção</i> – Problema teste A	139
Tabela 3.20. Resultados do método <i>Inserção</i> – Problema teste B	139
Tabela 3.21. Resultados do método <i>Inserção</i> – Problema teste C	139
Tabela 3.22. Resultados dos métodos <i>Inserção+</i> e <i>Busca Tabu</i> – Problema teste A	140
Tabela 3.23. Resultados dos métodos <i>Inserção+</i> e <i>Busca Tabu</i> – Problema teste B.....	141
Tabela 3.24. Resultados dos métodos <i>Inserção+</i> e <i>Busca Tabu</i> – Problema teste C.....	141
Tabela 6.1. Intra-rota troca-arcos – 1 ^a opção	165
Tabela 6.2. Intra-rota troca-arcos – 2 ^a opção	166
Tabela 6.3. Intra-rota inserção.....	168
Tabela 6.4. Inter-rota <i>crossover</i> – 1 ^a opção.....	170
Tabela 6.5. Inter-rota <i>crossover</i> – 2 ^a opção.....	171
Tabela 6.6. Inter-rota <i>crossover</i> – 3 ^a opção.....	172
Tabela 6.7. Inter-rota <i>crossover</i> – 4 ^a opção.....	174
Tabela 6.8. Inter-rota inserção – Opção 1	175
Tabela 6.9. Inter-rota inserção – Opção 2	177
Tabela 6.10. Inter-rota troca – Opção 1	179

Tabela 6.11. Inter-rota troca – Opção 2	181
Tabela 6.12. – Melhores resultados por problema teste – objetivo principal é minimizar o número de veículos	185
Tabela 6.13. – Descrição das fontes da Tabela 6.8.1.	189
Tabela 6.14. Premissas da análise de custo fixo e variável.....	191
Tabela 6.15. Cálculo do custo fixo.....	192
Tabela 6.16. Estimativa da quilometragem média diária por veículo.....	192
Tabela 6.17. Cálculo do custo variável	193
Tabela 6.18. Resumo do custo de um veículo.....	193
Tabela 6.19. Contribuição ao lucro por número de clientes atendidos (em R\$).....	194

Capítulo 1. Introdução

1.1. Visão geral sobre logística e problemas de roteamento estáticos e dinâmicos

Logística está associada ao gerenciamento eficiente de fluxo e estoque de produtos e serviços. O Conselho de Profissionais da Administração da Cadeia de Suprimentos (*CSCMP – Council of Supply Chain Management Professionals*) define logística como “... o processo eficiente de planejamento, implementação e controle do fluxo e economicamente eficaz de matérias-primas, estoque em processo, produtos acabados e informações relativas desde o ponto de origem até o ponto de consumo, com o propósito de atender às exigências dos clientes”. Ballou (2001) coloca que “a missão da logística é dispor a mercadoria ou o serviço certo, no lugar certo, no tempo certo e nas condições desejadas, ao mesmo tempo em que fornece a maior contribuição à empresa”.

Segundo um artigo da Revista Tecnológica, maio 2001 intitulado “Quanto custa a logística no Brasil?” (http://www.cvlog.net/Arquivos/Custos_CustoBrasil.htm), não existe uma estimativa precisa sobre o custo logístico. A Associação Brasileira de Movimentação e Logística (ABML) estima que este custo pode equivaler a 19% do faturamento de uma empresa, mas a falta de confiabilidade neste número levou a ABML a priorizar a criação de um índice que pretende diagnosticar periodicamente e com precisão os custos do setor.

Ballou (2001) cita estatísticas do Fundo Monetário Internacional em que o custo logístico representa em média 12% do produto interno bruto mundial. Outras pesquisas demonstram que os custos logísticos podem variar entre 4% e 30% do faturamento das empresas norte-americanas.

Diversas aplicações reais nos Estados Unidos e Europa mostram que o uso de ferramentas computacionais leva a uma redução de 5% a 20% do custo de transporte total (Toth e Vigo, 2002). O sucesso das técnicas da Pesquisa Operacional deve-se à elaboração de modelos mais realistas, ao desenvolvimento de algoritmos de otimização capazes de tratar problemas de grande porte, e à crescente integração de sistemas de informação.

Dentre os diversos componentes dos sistemas logísticos mencionados pela CSCMP – “(...) serviços ao cliente, previsão de vendas, controle de estoque, manuseio de materiais, processamento de pedidos, peças de reposição e serviços de suporte, seleção do local da planta e armazenagem (análise de localização), compras, embalagem, manuseio de mercadorias devolvidas, recuperação e descarte de sucata, tráfego e transporte, e armazenagem e estocagem” – os problemas de roteamento de veículos, estáticos ou dinâmicos, estão inseridos na área de transportes / distribuição.

Nos problemas de roteamento estático e determinístico, todos os dados, tais como localização e demanda dos clientes, são conhecidos antes da construção das rotas e estes dados não se alteram. No caso dinâmico, todos ou uma fração dos pedidos de serviço ocorrem enquanto as rotas são executadas.

Os problemas dinâmicos de roteamento são aqueles em que as informações necessárias para a determinação de rotas eficientes são reveladas dinamicamente ao tomador de decisão. Desta maneira, a solução não pode ser definida a priori e precisa ser atualizada simultaneamente com o desdobramento das informações. Portanto, faz-se necessário reagir a eventos que ocorrem em tempo real, tais como, novos pedidos de serviço (novos clientes), atrasos não previstos e acidentes.

Como exemplos de roteamento dinâmico pode-se citar indústrias de bebidas, distribuição de gases e outros derivados do petróleo (Bell *et al.*, 1983; Brown *et al.*, 1987; Bausch *et al.*, 1995), serviços de *courier* (Gendreau *et al.*, 1999), serviços de emergência (Gendreau *et al.*, 2001), serviços de reparo em empresas de eletricidade, gás, água, esgoto, e transporte de passageiros com requisitos específicos (*dial-a-ride*) (Gendreau e Potvin, 1997). Existem também problemas dinâmicos sem roteamento, como serviços de polícia e ambulância (Gendreau *et al.*, 2001).

O interesse por problemas de roteamento dinâmico é muito recente. Gendreau *et al.* (1999) usam o serviço de *courier* como motivação para resolver o problema em tempo real através da adaptação de um algoritmo estático para roteamento de veículos, considerando uma penalização caso ocorra a violação das restrições de janelas de tempo. Em um artigo subsequente, Ichoua *et al.* (2000) analisaram o efeito de alterar a rota do veículo no momento que surgem novos pedidos. Em (Larsen *et al.*, 2002, Larsen, 2000) os problemas são classificados segundo seu nível de dinamismo (percentual da quantidade de pedidos dinâmicos e disponibilidade de

tempo para reagir aos mesmos) e principal objetivo do tomador de decisão (minimizar a distância percorrida vs. minimizar o tempo de resposta). Os autores utilizam o problema dinâmico de um agente de reparo e simulam várias políticas para minimizar o custo de roteamento.

1.2. Entendimento dos problemas de roteamento no Brasil – pesquisa a campo

Com o intuito de aprimorar o entendimento sobre problemas de roteamento, avaliar a aplicabilidade dos algoritmos desenvolvidos e incluir aspectos reais nesta dissertação de mestrado, foi feita uma pesquisa a campo com empresas de transporte de Campinas e da região de São Paulo.

No total, foram contatadas 6 empresas: 4 foram entrevistadas pessoalmente e 2 por telefone. O guia de entrevistas pode ser encontrado no Anexo A.

Exceto em casos específicos, foi feito um acordo durante as entrevistas em que as informações aqui reveladas não seriam associadas à respectiva empresa. As fontes das informações a serem apresentadas são: Empresa de Correios e Telégrafos, Transportadora Mercúrio, Transportadora Americana, Transportadora Atlas, Kieling Multimodal e Transportadora Minuano.

1.2.1. Descrição das empresas

As empresas entrevistadas apresentam distinções nas seguintes características:

- a) **Porte:** Consideramos empresas de grande porte aquelas que possuem mais do que 500 veículos e de médio porte caso contrário;
- b) **Modalidade:** Empresas multimodais oferecem transportes rodoviário, ferroviário, aéreo e marítimo de forma integrada, enquanto empresas monomodais oferecem somente um dos quatro tipos de transporte. A modalidade transporte rodoviário (inclui transporte urbano) por ser realizada através de caminhões, vans ou motos;
- c) **Serviços adicionais:** Além do transporte, há empresas que oferecem gestão de estoque com espaço físico próprio ou terceirizado, serviços de guinchos e empilhadeiras, intermediação de transações de importação/exportação (comércio exterior), contratação de

seguros, fretamento de aeronaves, soluções personalizadas¹ e transporte de cargas indivisíveis, por exemplo, uma turbina de hidrelétrica ou antena de telecomunicações;

- d) **Uso de tecnologia:** Foram encontrados três níveis de tecnologias nas empresas, sendo (1) empresas que utilizam sistemas somente para controles administrativos e controle de clientes e pedidos – via telefone e/ou Internet, (2) empresas que utilizam (1) e sistemas de rastreamento de veículos – posição do veículo em tempo real, controle de carregamento e descarregamento, monitoramento da segurança do motorista, e (3) empresas que utilizam (1) e (2), e ainda uma ferramenta que cria os roteiros dos veículos.

Foi possível observar dois tipos de empresas. De um lado estão as empresas que oferecem multimodalidade e serviços adicionais. Estas empresas comentam que às vezes grande parte do lucro de uma operação está na integração das modalidades e na execução dos serviços adicionais. De outro, estão as empresas de grande porte focadas principalmente no transporte rodoviário. Estas empresas, para serem competitivas, possuem grande quantidade de veículos, estão nos níveis tecnológicos (2) ou (3) e investem na eficiência de movimentação de mercadorias e em treinamento dos funcionários e motoristas.

1.2.2. Descrição das operações de coleta, estocagem e entrega

A operação das empresas de transporte consiste em dois tipos básicos de coleta e entrega:

- A) Coleta e entrega na mesma cidade;
- B) Coleta numa cidade e entrega em outra cidade.

Numa empresa entrevistada, a coleta é feita no mesmo dia se o pedido for feito antes das 12hs. Outras empresas já estendem este horário para até às 14hs. Pedidos recebidos depois das 12hs ou 14hs são coletados no dia seguinte.

¹ Exemplo de uma solução personalizada: coletar uma mala de produtos e documentos do cliente através de motoboy ou táxi e transportá-la ao aeroporto. No aeroporto, uma pessoa da empresa de transporte compra um bilhete de passagem e embarca com a mala como bagagem de mão até o destino. No destino, outro motoboy coleta a mala e a entrega ao cliente. Desta maneira, a coleta e entrega pode ser feita no mesmo dia para percursos interestaduais ou internacionais (Mercosul).

O prazo para a entrega depende do contrato com cada cliente e é função da urgência e da distância entre os pontos de coleta e entrega. Existem, por exemplo, contratos que garantem a entrega no mesmo dia, no dia seguinte, em até dois dias, etc. Conseqüentemente, estes contratos determinam a prioridade de cada entrega para cada veículo: pedidos de contratos de entrega no mesmo dia possuem prioridade sobre os pedidos de entrega de contratos de até um ou dois dias.

Existem dois tipos de carga: carga fechada, na qual toda a carga do veículo pertence a um único cliente, e carga fracionada, em que a carga do veículo pertence a vários clientes. A operação de carga fechada para um único destino de qualquer tipo de coleta e entrega (A ou B) não será analisada, pois esta operação não requer nenhum esforço para roteamento – basta enviar o veículo vazio para o ponto de coleta e em seguida enviá-lo ao ponto de entrega, sem passar por nenhum depósito nem exigir a definição de roteiro.

Considerando apenas cargas fracionadas, a Tabela 1.1 resume os problemas de roteamento que foram encontrados durante as entrevistas para os tipos de coleta / entrega (A ou B) e prazos:

Tabela 1.1. Problemas de roteamento levantados na pesquisa a campo

Configuração	Coleta e entrega na mesma cidade?	Prazo para entrega	Mercadoria passa pelo depósito	Problema de roteamento
1	Sim	Mesmo dia	Não	Coleta e entrega (similar a <i>dial-a-ride</i> , explicado a seguir) – fora do escopo desta dissertação.
2	Sim	Dia seguinte ou acima de dois dias	Sim	Problemas de roteamento independentes: somente coleta, somente entrega, ou entrega pela manhã e coleta durante à tarde.
3	Não	Mesmo dia	Não	Solução personalizada, pois provavelmente requer carga fechada – fora do escopo desta dissertação.
4	Não	Dia seguinte ou acima de dois dias	Sim	Problemas de roteamento independentes: somente coleta, somente entrega, ou entrega pela manhã e coleta durante à tarde

A partir destas configurações, percebe-se que o prazo para a entrega determina o uso do depósito. Se for necessário entregar para a mesma cidade no mesmo dia em que a mercadoria é coletada (configuração 1), o veículo não passará pelo depósito e o problema torna-se semelhante

aos problemas do serviço *dial-a-ride*. O serviço *dial-a-ride* é um conjunto de *vans* que transporta pessoas – em geral idosas ou com dificuldade de locomoção – de um local a outro de forma coletiva e personalizada, sendo o destino de cada pessoa distinto: supermercado, cinema, local de trabalho, etc. Quando for necessário entregar no mesmo dia para outra cidade – configuração 3 – o transporte provavelmente é feito por uma solução personalizada em função da dificuldade de se cumprir o prazo. Estas duas configurações estão fora do escopo desta dissertação.

Quando a mercadoria é estocada no depósito – configurações 2 e 4 – os problemas de coleta e entrega tornam-se independentes e resultam em problemas tradicionais de roteamento. Esta independência provém do fato de que as mercadorias coletadas num dia por um veículo são reorganizadas durante a noite no depósito para serem entregues nos dias seguintes por outro veículo. Segundo as entrevistas, este é o caso mais comum na prática e, como mostrado na Figura 1.1, é a maneira pela qual as transportadoras organizam seus depósitos.

Existe ainda o caso em que o mesmo veículo efetua entregas pela manhã e coletas durante à tarde – nem todas as empresas adotam esta prática. Observe que nesta situação a independência ainda é mantida, pois nenhum item é coletado e entregue no mesmo dia – todos os pedidos estiveram ou passarão pelo depósito. Esta configuração ocorre quando o roteiro de entrega do veículo não exige o dia inteiro para ser percorrido, ou quando um pedido realizado antes das 14hs é inserido no roteiro do veículo (problema dinâmico).

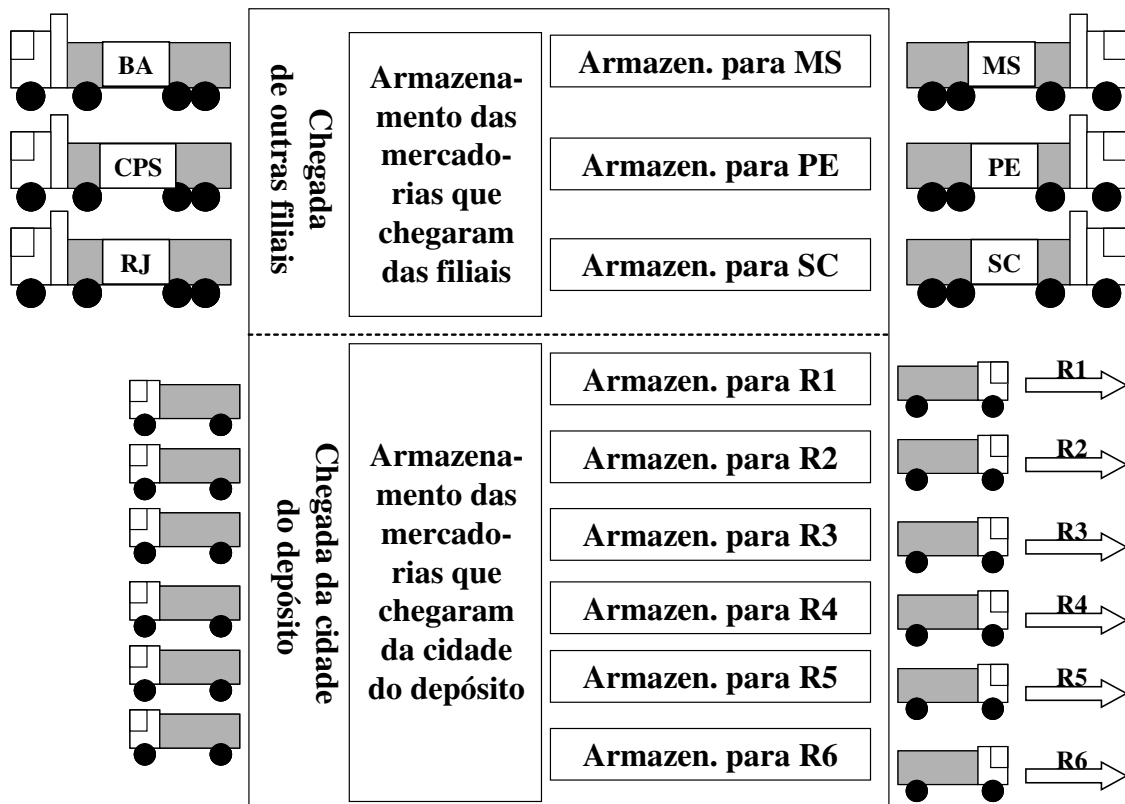


Figura 1.1. Esquema do layout do depósito das transportadoras

A chegada de mercadorias é feita pelo lado esquerdo da figura e é separada em duas seções: chegada de mercadorias das filiais da transportadora de outras cidades e/ou estados e chegada das mercadorias da cidade na qual o depósito está instalado. Observe que os veículos que chegam das filiais são em geral de maior porte.

O lado direito da figura representa a saída das mercadorias, também separada em filiais (MS, PE, SC) e regiões da cidade do depósito (R1 a R6). Uma região, por exemplo, R1, é geralmente composta por uma ou mais “cabeças” de CEP – os três primeiros dígitos do CEP – e corresponde a desde uma parte de um bairro a um conjunto de bairros de uma cidade.

Entre as docas de chegada e saída estão “corredores” que identificam a origem ou destino das mercadorias. Por exemplo, na direção da doca de MS, existe um espaço físico demarcado por um retângulo colorido pintado com tinta no chão do depósito no qual devem ser armazenadas as mercadorias destinadas a MS.

Após a chegada das mercadorias é feita a operação de *cross-docking*: as mercadorias que chegaram de diferentes veículos (de várias docas) são preparadas para serem carregadas em outro veículo (outra doca). Por exemplo, suponha que o depósito esteja localizado na capital São Paulo.

Podem existir mercadorias de um veículo da rota do bairro Perdizes e de outro da rota do bairro Tatuapé da capital que devem ser embarcadas para Santa Catarina. Ou então, podem haver mercadorias da Bahia e do Rio de Janeiro que devem ser embarcadas para o Centro de São Paulo.

1.2.3. Atendimento aos clientes

Cada filial das transportadoras é responsável pela coleta e entrega numa cidade. Para facilitar o manuseio das mercadorias no depósito, a geração dos roteiros dos motoristas e o aprendizado do motorista sobre os melhores caminhos dentro da cidade, a cidade é segmentada em regiões (conjunto de “cabeças” de CEP).

O tamanho das regiões é determinado de tal forma que, em geral, cada região demande no máximo três veículos. Os motoristas geralmente trabalham sempre na mesma região, pois com o maior conhecimento da região a quantidade de entregas por motorista tende a crescer.

A partir das definições das regiões, o processo de atendimento aos clientes, desde a chegada do pedido até a sua entrega, é feito da seguinte forma:

- 1) Cliente liga para a central de atendimento;
- 2) Atendente cadastra o pedido no banco de dados: endereço de coleta e de entrega, dimensões, peso e valor da mercadoria, etc. Em alguns casos, o pedido e a nota fiscal podem chegar à transportadora via *EDI (Eletronic Data Interchange)*, evitando digitações e erros;
- 3) O pedido é alocado a uma região através da “cabeça” do CEP;
 - 4.1) Se o pedido foi cadastrado antes das 12hs (ou antes das 14hs em algumas empresas), o pedido é inserido no roteiro de algum veículo de sua região para ser coletado no mesmo dia (problema dinâmico).

Observação: As pessoas responsáveis pela inserção do novo pedido de forma dinâmica trabalham no setor de monitoramento de veículos. Cada pessoa monitora no máximo 52 veículos e controla os roteiros, entregas e coletas concluídas e a serem realizadas, atrasos e outras questões a serem apresentadas posteriormente (item 1.2.4. Tecnologia de rastreamento);

- 4.2) Se o pedido foi cadastrado após às 12hs, o pedido será coletado no dia seguinte e fica armazenado no banco de dados para receber um roteiro no turno da noite (problema estático);
- 5) Após a coleta de todos os pedidos do dia, o veículo retorna ao depósito para efetuar o descarregamento. Durante a noite e a madrugada as mercadorias que chegaram são rearranjadas

para serem entregues no dia seguinte (*cross-docking*). Uma vez agrupadas todas as mercadorias de uma região de entrega, estas devem ser alocadas aos respectivos veículos de tal forma que as restrições de capacidade (peso e volume), janelas de tempo e condições contratuais e de seguradoras sejam atendidas. Na ausência de um software, existe uma pessoa que faz isso para cada região. Esta pessoa, ou o software, também pode alocar as coletas programadas no dia anterior para o final do roteiro dos veículos de entrega (pedidos do item 4.2);

6) Ao amanhecer, as cargas de cada veículo já estão separadas. Nas empresas que não possuem software de roteamento, o roteiro pode ser criado pelo responsável da região ou, mais frequentemente, pelo motorista. O carregamento reflete o roteiro, ou seja, as primeiras entregas devem estar mais próximas da porta do baú do caminhão do que as últimas entregas.

7) Os veículos partem do depósito para efetuar as entregas. Se o veículo é próprio, ou se algum pedido de coleta foi encaixado em seu roteiro, o veículo deve retornar ao depósito ou a outra garagem ao término de todas as entregas. Se o veículo é terceirizado, o veículo não precisa voltar ao depósito – caso mais comum. Perceba que quando o veículo não necessita retornar ao depósito o problema de roteamento se transforma num problema de seqüenciamento de tarefas em máquinas paralelas.

8) Durante o dia, as pessoas do setor de monitoramento podem incluir pedidos de coleta tanto no roteiro dos veículos de entrega quanto no roteiro dos veículos de coleta (“encaixe” de pedidos, problema dinâmico).

Note que, em geral, o problema de roteamento encontrado é estático. O problema é dinâmico quando existe coleta a ser realizada no mesmo dia em que o pedido foi feito – pedidos inseridos antes das 12hs ou 14hs. Estes pedidos são tratados separadamente e existem pessoas de monitoramento responsáveis pelo encaixe desses pedidos nos roteiros dos veículos que estão na rua. Estas pessoas podem, inclusive, alterar o destino atual do motorista para encaixar um novo pedido (vide Anexo D).

1.2.4. Tecnologia de rastreamento / monitoramento

A maioria das empresas de transportes utiliza pelo menos uma das três opções abaixo de software de rastreamento:

- 1) Control Loc: Software desenvolvido pela empresa Autofax Control;

- 2) OmniLink: Sistema da Omnilink Tecnologia Ltda. (<http://www.omnilink.com.br>)
- 3) Qtracs BR: Sistema comercializado pela empresa Autotrak (<http://www.autotrak.com.br>)

Estes aplicativos são ferramentas de rastreamento e controle dos veículos. Rastreamento porque é possível ver a posição praticamente exata do veículo num mapa digitalizado da cidade em tempo real. O posicionamento é realizado por um receptor *GPS*² (*Global Positioning System*) que converte sinais de satélites para calcular a latitude e longitude do local do receptor – precisão em torno de 100 metros na horizontal e 156 metros na vertical.

Os aplicativos também são ferramentas de controle porque:

- Possibilitam a comunicação rápida entre o motorista e a central através de mensagens de texto;
- Permitem a abertura e bloqueio das portas do baú do caminhão. Por exemplo, as portas do baú podem ser programadas para ficarem bloqueadas durante todo o percurso do veículo – nem mesmo o motorista consegue abri-las. Quando o motorista se aproxima de seu destino, por exemplo, num raio de 100m do cliente, as portas são desbloqueadas automaticamente. Outra possibilidade é manter sempre as portas bloqueadas. Quando o motorista chega ao destino este solicita através de um computador de bordo a abertura do baú. A central confere se a posição do motorista no mapa digitalizado é suficientemente próxima à posição do próximo cliente do roteiro. Caso afirmativo, a porta é liberada. Caso contrário, é feita uma comunicação via mensagens de texto, *trunking* ou telefonia convencional para esclarecimentos;
- Acionam automaticamente ou manualmente – pela central – o alarme do veículo (sirene, luzes, bloqueio de portas e de combustível) em situações de risco ou roubo. Exemplo: grande parte dos assaltos acontece quando o veículo está parando ou saindo. Estas situações são mais comuns em postos de abastecimento de combustível. E as empresas de seguro possuem cláusulas que determinam que não serão cobertos sinistros nos quais o

² O GPS foi desenvolvido pelo Departamento de Defesa dos Estados Unidos e é composto por um conjunto de 24 satélites que percorrem a órbita da Terra a cada 12 horas. Em qualquer ponto da Terra, um receptor GPS recebe sinais de 5 a 8 satélites, sendo necessários pelo menos 4 satélites para compor o posicionamento do receptor nas coordenadas X, Y, Z e Tempo.

- veículo tenha efetuado alguma parada num posto de abastecimento não autorizado. Desta maneira, os softwares de rastreamento são utilizados para disparar alarmes automaticamente se os motoristas efetuarem paradas em locais não autorizados;
- Disparam alerta se a porta do motorista foi aberta em local indevido. Para roubar o veículo, o motorista deve, em geral, descer do mesmo. Portanto, a central é notificada toda vez que a porta do motorista é aberta quando o veículo se encontra distante dos destinos – sabendo disso, os bandidos atualmente chegam até a remover o motorista pela janela ou vidro, sem abrir a porta;
 - Enviam, através de uma câmera digital, fotos a cada 10 segundos da cabine do veículo para a central. Isto é feito automaticamente quando os alarmes são acionados;
 - Notificam a central se o veículo está transitando em velocidade superior à máxima permitida;
 - Notificam a central se o veículo está se movendo ao destino por uma rota não aprovada. Por exemplo, a partir da rodovia Dutra, pode-se ir da capital São Paulo para Campinas por duas rodovias, dos Bandeirantes ou Fernão Dias. Caso o motorista opte pela rodovia não autorizada, a central pode ser notificada.

O controle sobre o motorista é muito forte porque o risco de roubo das mercadorias é alto. Parte desse investimento é recompensada porque as corretoras de seguro cobram taxas diferenciadas para veículos rastreados e não rastreados. Além disso, há maior preservação da vida dos motoristas.

Quando a mercadoria a ser transportada possui alto risco (pneus, alumínio, medicamentos e eletrônicos), pode-se ainda solicitar um veículo de passeio para escolta com dois seguranças armados.

Para efetuar a comunicação entre os motoristas e o setor de monitoramento da transportadora, os *softwares* (1) e (2) utilizam serviços das operadoras de telefonia celular baseados nas tecnologias CDMA, GSM ou GPRS enquanto o aplicativo (3) utiliza comunicação via satélite (BrasilSAT). Segundo as pessoas consultadas, as vantagens dos sistemas baseados em serviços das operadoras de telefonia celular são maior agilidade (mensagem ou comando é recebido geralmente em menos de três segundos) e menor custo (até sete vezes menor).

Apesar de possuir todas as informações em tempo real, a aplicação de rastreamento não faz nenhuma sugestão de roteiro. Entretanto, alguns destes sistemas podem ser conectados ao sistema de roteamento *Roadshow* (<http://www.descartes.com/solutions/roadshow.html>).

1.2.5. Ferramentas de roteamento

Durante as entrevistas foram mencionadas duas ferramentas de roteamento: *Roadshow* e *Truckstops* (<http://www.bestroutes.com/truckstops/index.html>). Das transportadoras entrevistadas, temos autorização para citar que a Transportadora Americana utiliza a ferramenta *Truckstops* para criar os roteiros dos veículos.

É importante observar que é mais comum encontrar ferramentas de roteamento em empresas de produção que possuem frota dedicada (própria ou terceirizada), como Souza Cruz, Kibon, Sadia, do que em empresas de transporte. O motivo apontado é que os pontos de coleta e entrega nestas empresas são razoavelmente constantes e periódicos, enquanto nas transportadoras são mais variados. A baixa variabilidade facilita o cadastro dos pontos de coleta e entrega no mapa digitalizado e a geração e análise dos roteiros. Um caso oposto, de alta variabilidade, foi mencionado durante a entrevista com os Correios: “*Eu teria mais pessoas cadastrando os pedidos no sistema do que nas ruas efetuando as entregas*”.

Nas empresas que não possuem uma ferramenta de roteamento, os pedidos são alocados a regiões de entrega e coleta, as quais são previamente definidas a partir de estudos de uma fase de planejamento. A alocação é feita com base em alguma informação do CEP do pedido. Após a alocação, o tamanho do problema é bastante reduzido, pois basta alocar os pedidos aos veículos dedicados à região. Uma vez feita a alocação dos pedidos aos veículos, o roteamento é delegado aos motoristas.

Nas empresas que possuem uma ferramenta de roteamento, os pedidos também são alocados a uma região com base no CEP. Note que esta etapa adiciona uma restrição que pode piorar a qualidade dos roteiros. Contudo, operacionalmente é vantajoso dividir o problema total em vários subproblemas de roteamento – por região – pelos seguintes motivos:

- Em cidades grandes é praticamente impossível para o motorista dirigir eficientemente em toda a cidade. Cada motorista é mais eficiente em determinadas regiões. Com o tempo, assumindo motoristas dedicados para certas regiões, esta habilidade é cada vez mais aprimorada;

- As ruas das cidades são, em alguns casos, planejadas para facilitar o fluxo dentro da região, sendo que existem poucas opções de conexão entre regiões. Por exemplo, dentro de um bairro podem existir várias vias (alta capilarização), mas o acesso ao bairro poderia ser feito através de uma única opção. Nestas situações é mais interessante enviar um único veículo ao bairro para realizar todas as coletas, do que enviar 5 veículos que passem pelo bairro para cada um efetuar poucas coletas. Outro exemplo é o centro da cidade. Em geral, é difícil para um veículo entrar e sair do centro da cidade. Isto sugere que o centro seja uma região com veículos dedicados;
- As regiões podem determinar o porte do veículo (restrições de peso e de dificuldade de mobilidade);
- A movimentação das mercadorias no depósito é facilitada: placas podem ser colocadas em cada doca com as “cabeças” de CEP para cada veículo. Desta maneira, os operadores de manuseio podem movimentar as mercadorias e organizar as notas fiscais sem precisar aguardar pela consolidação de pedidos, pela resposta de algum sistema ou pela tomada de decisão de um responsável.

Uma vez alocados os pedidos às regiões, o sistema de roteamento *Truckstops* gera as rotas. Uma rota nunca percorrerá mais de uma região e sempre respeitará as seguintes restrições:

- Capacidade: peso e volume das encomendas;
- Janelas de tempo e tempo de serviço: para tanto é mantido um histórico com médias e variâncias dos tempos de cada cliente. Por exemplo, o Carrefour recebe veículos somente até às 12hs e possui tempo de serviço de até 4 horas. Existem clientes que só recebem no horário de almoço, ou o contrário, que não recebem no horário de almoço. *Shopping Centers* também possuem horários específicos para recebimento de veículos;
- Dias de entrega: existem clientes que não recebem veículos em determinados dias da semana ou datas específicas (dias de fechamento, dias de balanço);
- Restrições de segurança: cláusulas das empresas de seguro estipulam valores máximos em reais que um veículo pode carregar. O valor pode ser determinado por tipo de produto (medicamentos e eletrônicos possuem valores menores enquanto detergente possui valores maiores), por região (RJ possui valores menores, Bauru possui valores

maiores), por tipo de veículo (Kombi vs. Caminhão baú), por equipamento de segurança (uma Kombi equipada com sistema de rastreamento pode carregar até R\$ 100.000 enquanto a mesma Kombi sem equipamentos de rastreamento carrega no máximo R\$ 40.000), etc. No total são 12 restrições impostas pelas empresas de seguro;

- Combinação de mercadorias: por exemplo, alimentos não podem ser carregados no mesmo veículo que carrega substâncias químicas;
- Restrições de posição: Cliente exige que suas mercadorias sejam as primeiras (ou últimas) do roteiro do veículo (menor risco de roubos, facilidade de descarregamento, restrições de movimentação de carga);
- Prioridade: para cada pedido é associada uma prioridade. Por exemplo, se o pedido deve, segundo contratos com o cliente, ser entregue obrigatoriamente naquele dia, ou o pedido pertence a um cliente estratégico, este pedido deve estar próximo do início do roteiro – menor probabilidade de atraso. Existem ainda outras prioridades específicas, como vacinas e remédios para hospitais;
- Restrições operacionais: por exemplo, a doca do cliente só recebe caminhões dos tipos A e B;
- Exigências de cargas fechadas (não fracionada): roteirizador deve agrupar pedidos de um período – uma semana, por exemplo – para entregar um caminhão dedicado ao cliente.

O sistema ainda permite a otimização de dois tipos de rotas: rotas tradicionais em formatos de pétalas (veículo parte do depósito, realiza o roteiro e retorna ao depósito) ou rotas longitudinais (veículo parte do depósito e realiza o roteiro – custo de retornar ao depósito não é considerado). Como a filial da empresa entrevistada trabalha com frota terceirizada, a otimização das entregas é feita com base em rotas longitudinais (problema de seqüenciamento de tarefas em máquinas paralelas).

Outras informações sobre o sistema *Truckstops*:

- O horizonte de otimização pode ser de 1, 2, 3 ou mais dias, dependendo da seleção do operador;

- O tempo de execução para gerar roteiros de 1.000 a 1.600 pedidos fica entre 40 e 60 minutos num computador pessoal de alto desempenho;
- Após a definição das rotas pelo sistema, existe uma fase na qual o operador manipula a solução para realizar manualmente os ajustes finais;
- O erro entre a distância percorrida planejada e realizada fica entre 5 e 8%;
- O custo de implementação do sistema está entre US\$ 15.000 e US\$ 20.000 com taxa de manutenção anual em torno de 10%.

As principais dificuldades de implementação foram:

- Criar e gerir a base de dados. Além dos dados básicos como endereços e pedidos dos clientes, é preciso atualizar as informações de janela de tempo, tempo de serviço, velocidade média e outras restrições específicas de cada cliente. Ainda, a coleta de dados deve ser sistematizada e as médias e variâncias recalculadas com certa frequência (por exemplo, semanalmente);

- Operacionalização com os motoristas terceirizados: Antes do uso do *Truckstops*, o próprio motorista definia o roteiro. Portanto, se o roteiro não fosse cumprido, o motorista não podia associar a falha a outra pessoa que não fosse a ele mesmo.

Quando o roteiro passou a ser informado ao motorista pelo sistema, o sistema passou a assumir a culpa das entregas não cumpridas (“*o roteiro do sistema é impossível de ser realizado...*”). Houve também queixas dos motoristas por perda de liberdade.

Ocorreram até “sabotagens” ao sistema, como, por exemplo, afirmar que o carregamento que o sistema definiu não cabia no caminhão. Ao observar mais detalhadamente o carregamento, eram observadas gaiolas de espaço vazio (empilhamento de caixas ocultando espaço vazio no seu centro).

Com o tempo estas dificuldades foram superadas e o funcionamento do sistema de roteamento se tornou rotineiro e integrado com a base de dados de informações sobre os clientes. As Figuras 1.2. e 1.3. ilustram as rotas criadas pelo sistema para a cidade de São Paulo.

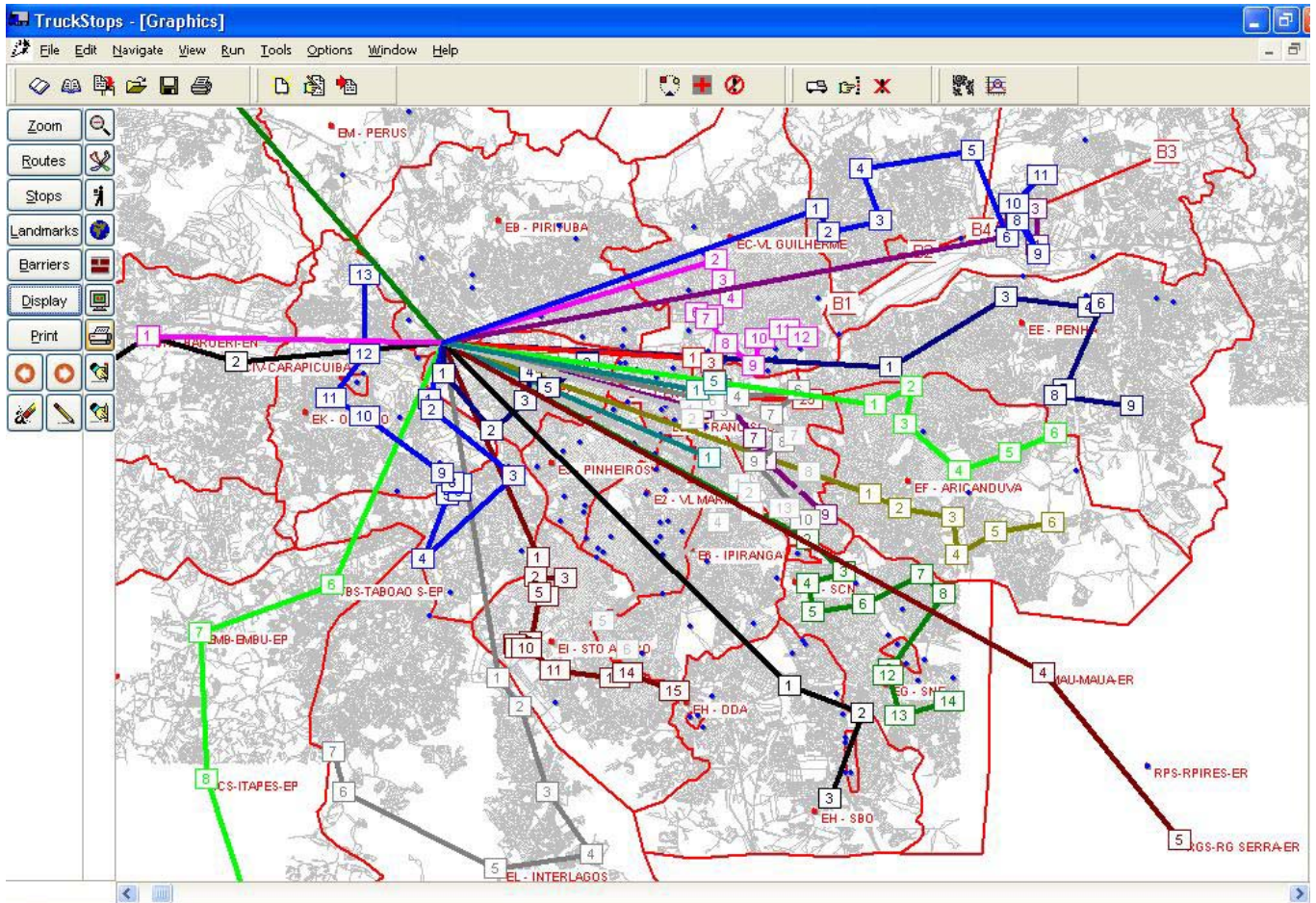


Figura 1.2. Visão geral dos roteiros de um dia da capital de São Paulo

fonte: Transportadora Americana

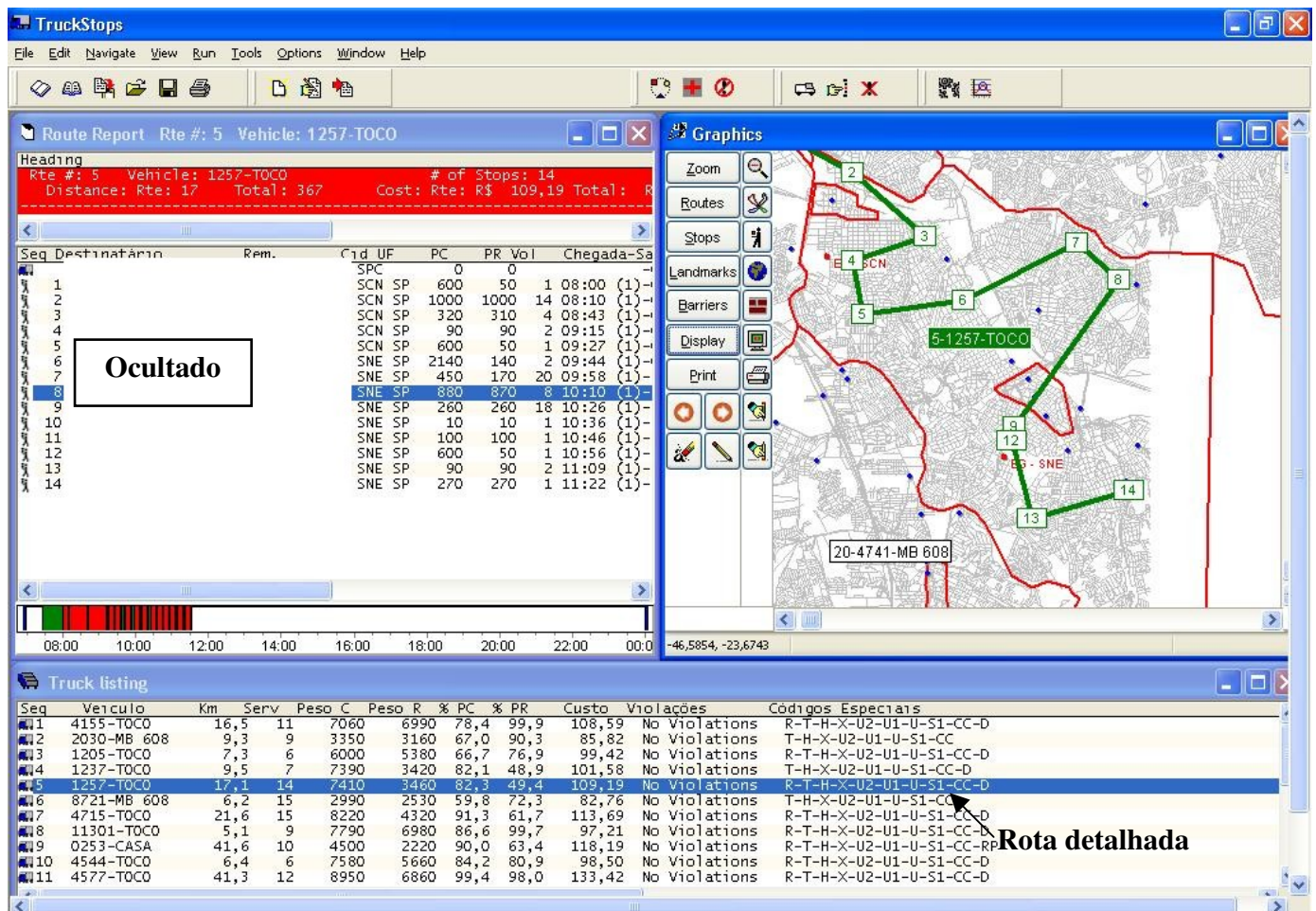


Figura 1.3. Detalhamento de uma rota

fonte: Transportadora Americana

1.3. Objetivos desta dissertação e atividades desenvolvidas

Este trabalho possui três objetivos: (i) desenvolver um método heurístico eficiente e com tempo computacional aceitável – baseado na metaheurística busca tabu de Glover e Laguna (1997) e na vizinhança granular de Toth e Vigo (2003) – para o problema estático de roteamento de veículos com janelas de tempo (PRVJT) e compará-lo a outros métodos da literatura; (ii) adaptar o método heurístico do problema estático para o problema dinâmico de veículos com janelas de tempo (PRDVJT), buscando tornar o tempo computacional compatível ao dinamismo do problema; (iii) desenvolver um ambiente de teste para os experimentos computacionais que inclua os aspectos práticos levantados na pesquisa a campo.

Para atingir estes objetivos as seguintes atividades foram desenvolvidas:

- Entendimento do PRVJT e revisão da literatura para o problema estático – **Capítulo 2, itens 2.1 e 2.2;**
- Implementação de heurísticas construtivas da literatura para o PRVJT e elaboração de uma nova heurística – **Capítulo 2, item 2.3;**
- Implementação da busca local e tabu granular para o PRVJT – **Capítulo 2, itens 2.4 e 2.5;**
- Conclusões sobre os testes do PRVJT – **Capítulo 2, item 2.6;**
- Entendimento do PRDVJT e revisão da literatura para o problema dinâmico – **Capítulo 3, itens 3.1 e 3.2;**
- Seleção dos métodos heurísticos a serem implementados para o PRDVJT – **Capítulo 3, item 3.3;**
- Projeto e implementação de um ambiente de teste para o PRDVJT – **Capítulo 3, item 3.4;**
- Adaptação do método heurístico do problema estático para o PRDVJT – **Capítulo 3, item 3.5;**
- Resultados dos testes para o PRDVJT – **Capítulo 3, item 3.6;**
- Conclusões sobre os testes do PRDVJT – **Capítulo 3, item 3.7;**
- Desenvolvimento das conclusões – **Capítulo 4;**

Observação: a lista das referências bibliográficas consultadas e os anexos se encontram após o **Capítulo 4**.

Capítulo 2. Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT)

2.1. Apresentação do PRVJT e formulação matemática

O Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT) é definido em um grafo completo $G = (V, A)$, em que $V = C \cup \{v_0, v_{n+1}\}$, $C = \{v_1, \dots, v_n\}$ é o conjunto de nós que representam os clientes, e v_0, v_{n+1} são nós que representam o depósito. O conjunto $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ corresponde aos arcos associados às conexões entre nós. Nenhum arco termina no nó v_0 e nenhum arco começa no nó v_{n+1} . Todas as rotas começam em v_0 terminam em v_{n+1} . Um custo c_{ij} e um tempo de viagem t_{ij} estão associados a cada arco $(v_i, v_j) \in A$. O tempo de viagem t_{ij} pode incluir o tempo de serviço s_i do cliente i . Um conjunto K de veículos idênticos está situado no depósito. Cada veículo tem capacidade q e cada cliente tem uma demanda d_i , $i \in C$.

Em cada cliente, o início do serviço deve estar dentro de uma janela de tempo $[e_i, l_i]$, $i \in C$. Os veículos devem deixar o depósito no intervalo $[e_0, l_0]$ e retornar durante o intervalo $[e_{n+1}, l_{n+1}]$. Um veículo pode chegar em um cliente antes da sua janela de tempo e esperar sem custo. Pode-se assumir, sem perda de generalidade, que $e_0 = l_0$, isto é, todas as rotas começam no instante 0. Assuma também que os parâmetros q , d_i , c_{ij} , t_{ij} , e_i , e l_i são inteiros não negativos. O PRVJT consiste em designar rotas para os veículos de forma que:

- i) cada rota inicia e termina no depósito;
- ii) cada cliente pertence somente a uma rota;
- iii) a demanda total de uma rota não excede a capacidade do veículo;
- iv) o serviço do cliente i começa no intervalo $[e_i, l_i]$, e cada veículo deixa o depósito no instante 0 e retorna ao depósito no intervalo $[e_{n+1}, l_{n+1}]$;
- v) o número de veículos e o custo total de viagens são minimizados.

Considere as seguintes variáveis:

$$X_{ij}^k = \begin{cases} 1 & \text{se o veículo } k \text{ percorre o arco } (i, j) \\ 0 & \text{caso contrário;} \end{cases} \quad \forall k \in K, \forall (i, j) \in A$$

$$S_i^k = \text{tempo em que o veículo } k \text{ começa a servir o cliente } i \quad \forall k \in K, \forall i \in C \cup \{v_{n+1}\}$$

O modelo matemático a seguir para o PRVJT foi proposto por Kohl *et al.* (1999).

$$(1) \min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} X_{ij}^k$$

s.a.

$$(2) \sum_{k \in K} \sum_{j \in C \cup \{v_{n+1}\}, j \neq i} X_{ij}^k = 1, \quad \forall i \in C$$

$$(3) \sum_{i \in C} d_i \sum_{j \in C \cup \{v_{n+1}\}, j \neq i} X_{ij}^k \leq q, \quad \forall k \in K$$

$$(4) \sum_{j \in C \cup \{v_{n+1}\}} X_{0j}^k = 1, \quad \forall k \in K$$

$$(5) \sum_{i \in C \cup \{v_0\}} X_{ih}^k - \sum_{j \in C \cup \{v_{n+1}\}} X_{hj}^k = 0, \quad \forall h \in C, \forall k \in K$$

$$(6) \sum_{i \in C} X_{i,n+1}^k = 1, \quad \forall k \in V$$

$$(7) X_{ij}^k (S_i^k + t_{ij} - S_j^k) \leq 0, \quad \forall (i, j) \in A, \forall k \in K$$

$$(8) e_i \leq S_i^k \leq l_i, \quad \forall i \in C \cup \{v_{n+1}\}, \forall k \in K$$

$$(9) X_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in A, \forall k \in K$$

A função objetivo (1) busca minimizar a somatória dos custos ou distâncias percorridas pelos veículos. As restrições (2) garantem que todos os clientes são visitados e somente por um veículo. As restrições (3) determinam que a soma das demandas da rota percorrida pelos veículos não seja superior à capacidade q . As restrições (4) forçam que todos os veículos deixem o depósito para algum cliente, ou, para o próprio depósito. As rotas artificiais (do depósito ao próprio depósito) são utilizadas se o número de veículos é uma variável. Neste caso, um custo c_v é atribuído a cada veículo usado, impondo-se $c_{0,n+1} = -c_v$. Se o valor de c_v é elevado, o modelo primeiramente minimiza o número de veículos, e em segundo lugar, minimiza o custo total. No

entanto, se o número de veículos for um parâmetro fixo, o arco $(0, n+1)$ deve ser removido. As restrições (5) garantem o balanço em cada nó: se um veículo chega ao nó h , um veículo deve deixar o nó h . As restrições (6) estabelecem que todos os veículos devem terminar no nó v_{n+1} (depósito). As restrições (7) definem que se um veículo chega ao nó v_j , o tempo de chegada do nó v_j deve ser superior ao tempo de chegada do nó anterior v_i somado ao tempo de viagem de i a j . As restrições (8) certificam que todo tempo de chegada está dentro do intervalo da janela de tempo – inclusive para a janela de tempo do nó v_{n+1} (depósito). Finalmente, as restrições (9) exigem que as condições de integralidade sejam satisfeitas.

Note que as restrições (7) podem ser transformadas em expressões lineares da seguinte forma:

$$(7') \quad S_i^k + t_{ij} \leq S_j^k + (1 - X_{ij}^k)M, \quad \forall (i, j) \in A, \quad \forall k \in K,$$

em que M é um número muito grande, maior ou igual ao máximo $(S_i^k + t_{ij})$, $\forall (i, j) \in A, \forall k \in K$.

2.2. Revisão dos métodos heurísticos da literatura

2.2.1. Problema de Roteamento de Veículos (PRV)

Antes de apresentar os métodos heurísticos do PRVJT, é feita uma breve introdução das heurísticas clássicas para a resolução do Problema de Roteamento de Veículos básico (PRV), considerando somente a restrição de capacidade dos veículos.

Laporte *et al.* (2000) expõem uma revisão dos métodos heurísticos para o PRV, incluindo as principais heurísticas construtivas tais como o algoritmo das economias de Clarke e Wright (1964), algoritmo *sweep* atribuído a Gillet and Miller (1974), algoritmos de pétalas, algoritmo cluster-first, route second de Fisher e Jaikumar (1981), entre outros. A comparação do desempenho das heurísticas construtivas aponta que bons resultados são obtidos rapidamente com o algoritmo das economias e o *sweep*. Os autores também apresentam as principais metaheurísticas, concluindo que em geral os algoritmos baseados em busca tabu, como o Taburoute de Gendreau *et al.* (1994) e a busca tabu de Taillard (1993), obtiveram ótimos resultados.

Cordeau *et al.* (2002) expandem a comparação entre heurísticas construtivas e metaheurísticas para os seguintes atributos: qualidade, velocidade, simplicidade e flexibilidade.

Dentre os métodos testados, a busca tabu unificada de Cordeau *et al.* (1997) foi considerada como uma das melhores metaheurísticas.

Recentemente, Toth e Vigo (2003) introduziram a busca tabu granular para o PRV. A idéia central das vizinhanças granulares é a redução do espaço de busca, focando apenas na avaliação de movimentos promissores. Por exemplo, para problemas de roteamento é razoável imaginar que arcos muito “longos” não estariam presentes nas melhores soluções, portanto, descartam-se movimentos que gerariam tais arcos.

2.2.2. Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT)

Existe uma grande variedade de métodos heurísticos para a resolução do PRVJT. Dentre as heurísticas construtivas, destacam-se as heurísticas de Solomon (1987), Potvin e Rousseau (1995) e Ioannou *et al.* (2001). Resultados comparativos no trabalho de Ioannou *et al.* (2001) mostram que a heurística proposta (IMPACT) é superior às outras duas heurísticas, embora gaste mais tempo computacional que a heurística I1 de Solomon (1987). Esta, por sua vez é, em geral, superior à heurística de Potvin e Rousseau (1995).

Balakrishnan (1993) introduz três heurísticas para o PRVJT com penalidade para a violação das restrições de janela de tempo: heurística do vizinho mais próximo, heurística da economia e heurística do espaço-tempo. Nas três heurísticas as rotas são criadas sequencialmente – uma nova rota somente é criada na impossibilidade de atendimento dos clientes com a rota corrente. As heurísticas diferem no critério de seleção do primeiro cliente a ser inserido na rota (o cliente-semente) e na regra de seleção dos clientes seguintes (ponderação de valores de distância e janela de tempo). O autor conclui que o número de veículos pode ser significativamente reduzido ao permitir a violação das restrições de janela de tempo, e que a heurística do vizinho mais próximo obteve, em geral, o melhor desempenho.

Diversas metaheurísticas foram sugeridas na literatura para o PRVJT, e destacamos a seguir as que, aparentemente, são mais eficientes.

Rochat e Taillard (1995) mantêm um conjunto de rotas extraídas das melhores soluções obtidas durante a busca e, a cada iteração da metaheurística, constrói-se uma solução a partir de rotas selecionadas probabilisticamente do conjunto. O resumo do método pode ser encontrado no Anexo B.

Cordeau *et al.* (2001) mencionam que as implementações de metaheurísticas mais comuns na literatura para o PRVJT são busca tabu e algoritmos genéticos. Algoritmos de busca tabu foram propostos em (Rochat e Taillard, 1995, Potvin *et al.*, 1996, Chiang e Russel, 1997, Taillard *et al.*, 1997, Cordeau *et al.*, 2001). Algoritmos genéticos com busca local foram sugeridos por Potvin e Bengio (1996) e Tan *et al.* (2001). Homberger e Gehring (1999) desenvolveram um algoritmo evolucionário, muito semelhante a algoritmos genéticos atuais (não requerem codificação binária, ver Michalewicz, 1996), com busca local que faz o papel de mutação.

A busca tabu implementada por Cordeau *et al.* (2001) considera uma função objetivo penalizada pela violação de restrições de capacidade, duração máxima das rotas e de janela de tempo. Clientes são movidos de uma rota a outra, sendo proibida a volta do cliente à rota original por um determinado número de iterações. Os autores aplicam uma estratégia de diversificação baseada numa matriz de frequência, matriz que armazena o número de vezes que o cliente foi inserido em cada rota. A busca tabu implementada pelos autores teve excelente desempenho, superando, na época, os melhores resultados da literatura em 20% dos problemas testados.

Recentemente, Bräysy (2003) tratou o problema de redução do número de veículos e de minimização da distância de forma separada, segmentando sua heurística em quatro fases:

- Fase 1: criação de soluções iniciais distintas para que, futuramente, diferentes mínimos locais possam ser encontrados;
- Fase 2: aplicação de um eficiente procedimento focado na eliminação de rotas, o qual foi baseado na vizinhança cadeia de ejeções de Glover (1991, 1996) – vide Anexo H;
- Fases 3 e 4: minimização da distância percorrida através de uma busca em vizinhança variada (*variable neighborhood search*).

Cada fase do método é projetada para buscar qualidade num objetivo específico – diversidade, redução do número de veículos e minimização da distância. Esta segmentação torna independente a avaliação do desempenho dos procedimentos em cada fase, facilitando, além da implementação, o teste de idéias para melhorar a qualidade de cada objetivo. O método se demonstrou extremamente bem sucedido.

Esta mesma vertente foi seguida por Bent e Hentenryck (2004^a). Primeiro o método minimiza o número de veículos utilizando *simulated annealing* e em seguida aplica uma busca local de vizinhança de grande porte para minimizar a distância. Este é atualmente o melhor método da literatura para o PRVJT, considerando ambos os objetivos. O *simulated annealing* da

primeira fase possui uma vizinhança com 5 tipos de movimentos e uma função objetivo lexicográfica, a qual analisa o número de rotas da solução, o número de clientes por rota (prioriza o desequilíbrio entre o número de clientes por rota) e a violação das restrições de janela de tempo. O método de busca local da segunda fase é complexo e contém um algoritmo de exploração baseado em *branch-and-bound*.

Finalmente, Russel e Chiang (2004) implementaram a metaheurística *scatter search* para o PRVJT, utilizando como algoritmo de melhoria de soluções uma busca tabu reativa. O método de combinação é baseado na conservação de arcos comuns de duas ou mais soluções, aplicando em seguida um método de recuperação de factibilidade.

Os testes computacionais destas metaheurísticas são geralmente executados no conjunto de 56 problemas teste propostos por Solomon (1987), todos com 100 nós³. Este conjunto contém problemas com posição dos nós gerada com distribuição uniforme, posição dos nós agrupadas e uma mistura destes dois tipos. Recentemente, Homberger (2000) estendeu os problemas de Solomon para 200, 400, 600, 800 e 1000 nós.

2.3. Heurísticas construtivas

2.3.1. Heurística construtiva de Clarke & Wright (1964)

A heurística construtiva de Clarke & Wright é uma heurística gulosa que tem como partida a solução de maior custo, em que um veículo é alocado para cada cliente. A cada passo da heurística é efetuada a união factível de duas rotas que proporcione a maior economia, a qual é composta por:

1. Redução do número de veículos em uma unidade;
2. Redução da distância segundo a expressão:

$$s_{ij} = c_{i0} + c_{0j} - c_{ij},$$

tal que:

$$c_{i0} = \text{distância entre o cliente } i \text{ e o depósito,}$$

³ As melhores soluções obtidas pelos algoritmos propostos estão disponíveis na página <http://web.cba.neu.edu/~msolomon/problems.htm>.

c_{0j} = distância entre o depósito e o cliente j ,

c_{ij} = distância entre o cliente i e o cliente j ,

s_{ij} = economia de distância ao unir i a j .

Existem duas versões da heurística de Clarke & Wright, uma paralela e outra seqüencial.

Abaixo está o resumo da versão paralela:

Passo 1: Criar uma rota para cada cliente (roteiro depósito-cliente-depósito);

Passo 2: Calcular a economia ao unir todas as combinações de pares de rotas;

Passo 3: Ordenar as economias em ordem decrescente;

Passo 4: Efetuar a união factível de maior economia;

Passo 5: Retornar ao passo 4 enquanto houver união factível.

A versão seqüencial difere da paralela a partir do passo 4. Ao invés de efetuar a união factível de melhor economia considerando todas as opções, a versão seqüencial fixa uma rota como semente e realiza as uniões factíveis que envolvem somente a rota semente. Quando não houver mais uniões factíveis para esta rota, seleciona-se outra semente.

Em geral, a versão paralela proporciona soluções de menor distância que a versão seqüencial. Entretanto, em alguns casos, a versão seqüencial consegue obter uma solução com um número menor de veículos.

Nossos testes mostraram que raramente a versão seqüencial obteve um número menor de veículos. Portanto, foi adotada a versão paralela.

2.3.2. Heurística construtiva de Solomon (1987)

Marius Solomon desenvolveu uma heurística construtiva que inclui a restrição de janelas de tempo nas decisões de seleção de sementes e de posição de inserção dos clientes. Dentre as diversas variantes propostas por Solomon, descreve-se aquela que proporcionou o melhor resultado para os problemas teste:

1. Defina dois conjuntos, um conjunto de clientes inseridos (I) e outro conjunto de clientes não inseridos (NI);
2. Coloque todos os clientes no conjunto NI ;

3. Faça $I = \text{vazio}$;
4. Ache o próximo cliente-semente ($c_semente$). Há duas opções:
 - a. Cliente mais distante do depósito: considera que clientes distantes do depósito são mais difíceis de serem inseridos;
 - b. Cliente com o menor tempo final da janela de tempo: prioriza clientes que devem ser atendidos com maior urgência.
5. Faça $I = I \cup c_semente$ e $NI = NI - c_semente$;
6. Procure em NI o cliente i que ao ser inserido na rota de $c_semente$ proporciona o menor acréscimo de custo, respeitando as restrições de capacidade e de janela de tempo. Para tanto, teste todas as posições de inserção do cliente e selecione a melhor posição de inserção;
7. Insira o cliente i na rota de $c_semente$;
8. Faça $I = I \cup i$ e $NI = NI - i$;
9. Retorne ao passo 6 enquanto houver clientes a serem inseridos na rota de $c_semente$;
10. Volte ao passo 4 enquanto $NI \neq \text{vazio}$.

2.3.3. Heurística construtiva proposta – Redução de Rotas (RR)

Uma nova heurística construtiva, denominada Redução de Rotas (RR), foi desenvolvida neste trabalho. Esta heurística tem como objetivo principal a redução do número de veículos e como objetivo secundário a redução da distância total percorrida. Esta heurística é descrita abaixo:

Fase I – Construção da Solução de Partida

1. Calcule o limitante inferior para o número de veículos como:

$$LI = \text{menor inteiro superior a } \frac{\sum_{i \in C} d_i}{q}$$

2. Faça $num_sementes = LI$;
3. Utilizando o procedimento *Encontrar_semente()*, ache um conjunto $C_semente$ de clientes-semente de cardinalidade $num_sementes$;
4. Faça $cliente_a_ser_inserido = Acha_próximo_cliente_a_ser_inserido()$;

5. Se *cliente_a_ser_inserido* $\neq$ vazio, insira o cliente na posição factível de menor acréscimo de distância, considerando todas as rotas dos clientes de *C_semente*. Se *cliente_a_ser_inserido* = vazio, ou se não houver uma posição de inserção factível, vá para a Fase II⁴.
6. Retorne ao passo 4.

Fase II – Uso de busca local para inserção dos clientes restantes

1. Armazene em *lista_clientes_não_inseridos* todos os clientes não inseridos porque não houve posição de inserção factível;
2. Realize um movimento de *Busca_local_move_trecho()*;
3. Percorra a *lista_clientes_não_inseridos* e procure inserções factíveis para todos os clientes. Efetue a inserção assim que a mesma tenha sido encontrada. Se for possível inserir um cliente, remova-o da *lista_clientes_não_inseridos*;
4. Retorne ao passo 2 enquanto houver movimentos de *Busca_local_move_trecho()* a serem realizados;
5. Se a *lista_clientes_não_inseridos* não estiver vazia, aloque um veículo a cada cliente não inserido da lista;
6. Defina a solução obtida como *solução_da_fase_II*;
7. Vá para a fase III.

Fase III – Teste da melhor solução e reinício

1. Se *solução_da_fase_II* for a primeira solução obtida, faça *melhor_solução* = *solução_da_fase_II*.

Senão, se a quantidade de rotas da *solução_da_fase_II* for menor que a quantidade de rotas da *melhor_solução*, faça *melhor_solução* = *solução_da_fase_II*.

Senão, se a quantidade de rotas da *solução_da_fase_II* for igual ao número de rotas da *melhor_solução* e a distância percorrida da *solução_da_fase_II* for menor que a distância percorrida da *melhor_solução*, faça *melhor_solução* = *solução_da_fase_II*.

⁴ Quando a busca local não é aplicada (Fase II não é executada), aloca-se um veículo a cada cliente não inserido, atribui-se a *solução_da_fase_II* a solução proveniente da Fase I e o método continua a partir da Fase III.

2. Se a quantidade de rotas de $melhor_solução = num_sementes \rightarrow$ FIM. Senão, vá para o passo 3.
3. Faça $num_sementes = num_sementes + 1$ e acrescente uma nova semente ao conjunto $C_semente$ utilizando o procedimento $Encontrar_semente()$ – note que as sementes antigas são mantidas;
4. Reinicie a solução removendo todas as alocações de clientes a rotas;
5. Retorne ao passo 4 da Fase I.

Procedimento Encontrar_semente()

Notação:

NI = conjunto dos clientes não inseridos em nenhuma rota;

$C_semente$ = conjunto de todos os clientes-semente;

Este procedimento é guiado pelo princípio de que os clientes-semente não devem pertencer à mesma rota. Desta maneira, o problema de encontrar sementes é encontrar os clientes que não podem pertencer à mesma rota e os clientes que provavelmente não pertenceriam à mesma rota nas boas soluções. Para tanto, utilizamos duas unidades de medida: incompatibilidade e tempo de espera.

Definimos incompatibilidade entre dois clientes i e j como:

$$incomp_{ij} = e_i + s_i + t_{ij} - l_j$$

Unidades de medida semelhantes à incompatibilidade foram utilizadas por Balakrishnan (1993), Kontoravdis e Bard (1995) e Bräysy (2003) em suas heurísticas.

Note que a parcela $e_i + s_i + t_{ij}$ expressa a menor hora possível no cliente j para um veículo que parte do cliente i . Portanto, se $e_i + s_i + t_{ij}$ for maior que l_j (incompatibilidade positiva), o arco (i,j) é infactível e os dois clientes não podem pertencer à mesma rota. Desta maneira, se $j \in C_semente$, o cliente i pode ser um candidato a se tornar semente. Este caso é ilustrado na Figura 2.1.

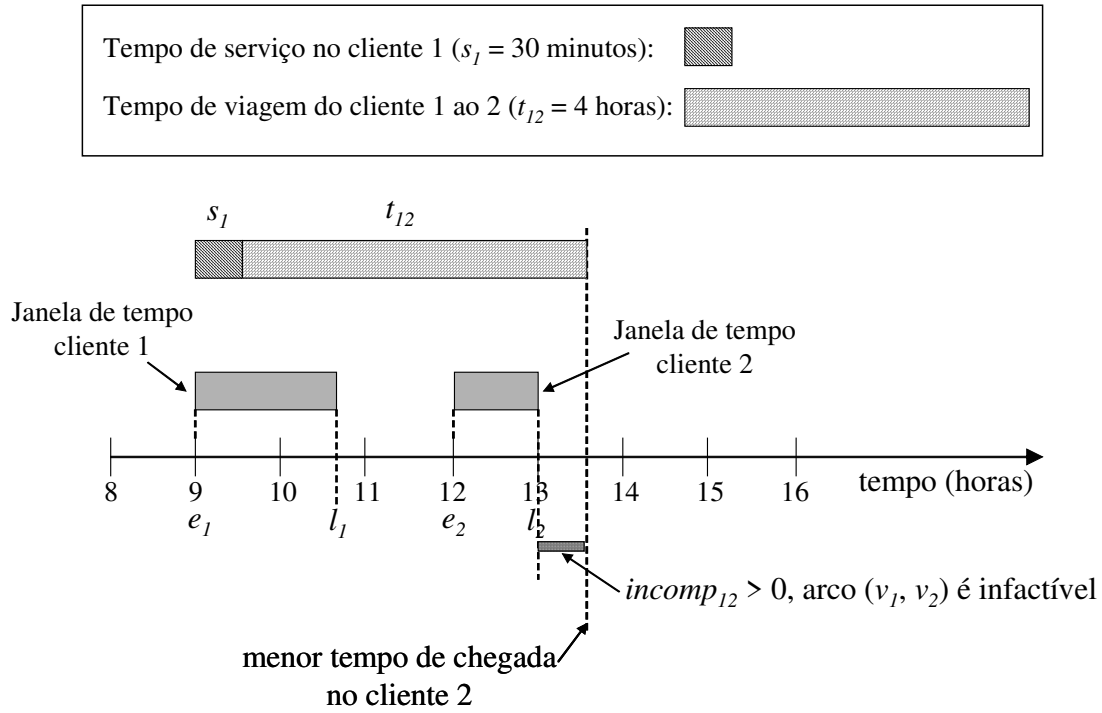


Figura 2.1. Ilustração de incompatibilidade positiva entre dois clientes

O exemplo da Figura 2.1 mostra que a menor hora de chegada no cliente 2 a partir do cliente 1 é 13:30. Como a hora final da janela de tempo do cliente 2 é 13:00, é infactível para qualquer veículo, em qualquer situação, atender o cliente 2 a partir de 1. Portanto, a $incomp_{12} > 0$.

A Figura 2.2 apresenta um caso de factibilidade, com $incomp_{12} < 0$.

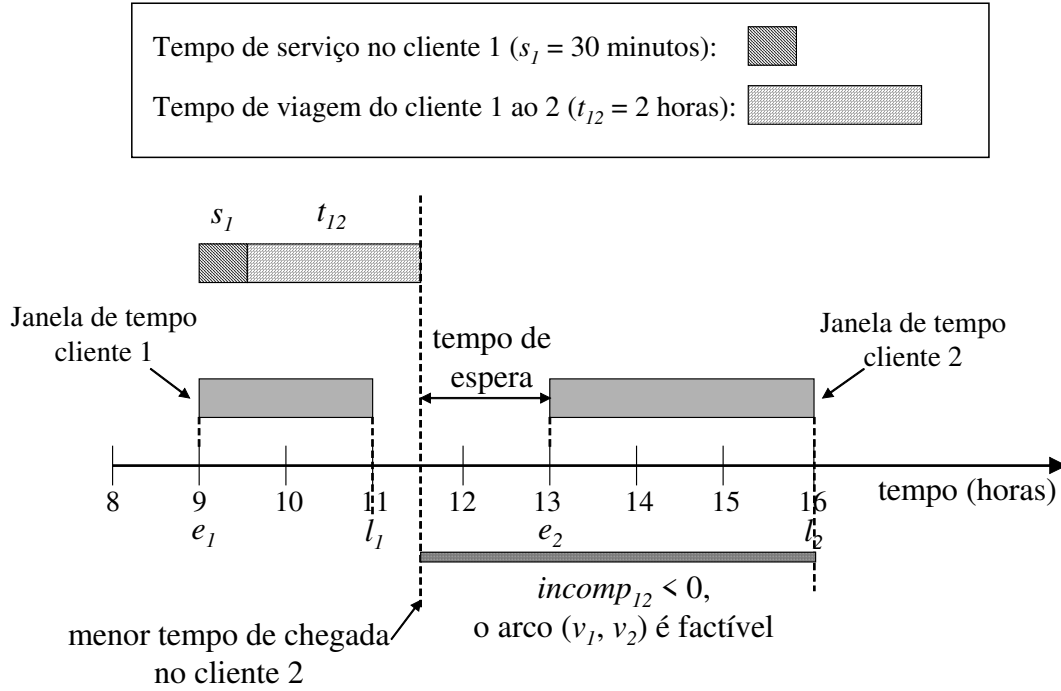


Figura 2.2. Ilustração de incompatibilidade negativa entre dois clientes

No exemplo da Figura 2.2, o menor momento de chegada no cliente 2 é 11:30. Como a janela de tempo do cliente 2 é $[13:00, 16:00]$, há tempo de espera⁵ de 1:30 antes do serviço no cliente 2. Observe que se a hora de início do serviço no cliente 1 for 11:00, no final de sua janela de tempo, a hora de chegada no cliente 2 seria 13:30, ainda dentro da janela de tempo do cliente 2. Nesta situação, não haveria tempo de espera.

Como a $incomp_{ij}$ pode assumir valores negativos (arcos factíveis) e preferimos trabalhar apenas com valores positivos tanto para incompatibilidade quanto para tempo de espera, definimos a normalização de $incomp_{ij}$ como:

$$incomp_norm_{ij} = incomp_{ij} - menor_incompatibilidade,$$

em que $menor_incompatibilidade = \text{Mínimo}(0, incomp_{ij}) \forall i, j \in NI$.

⁵ Consideramos que o tempo de espera não incorre em custos adicionais. Entretanto, é desejável que haja pouco tempo de espera para aumentar a utilização dos veículos. Esta motivação é tratada por outra medida a ser apresentada em seguida.

O tempo de espera é calculado de forma análoga à incompatibilidade:

$$t_espera_{ij} = \text{Máximo}(0, e_j - (e_i + s_i + t_{ij}))$$

Definimos agora um valor de separação entre um cliente i e uma semente j como a ponderação desses dois valores:

$$separação_{ij} = coef_inc \times incomp_norm_{ij} + coef_te \times t_espera_{ij},$$

em que $coef_inc \geq 0$, $coef_te \geq 0$ e $coef_inc + coef_te = 1$.

Assim, quanto maior a separação entre um cliente i e outro cliente-semente j , maior a probabilidade do cliente i não pertencer à rota do cliente-semente j . Desta maneira, se o cliente i possuir elevada separação entre todos os outros clientes-semente, o cliente i é um forte candidato a ser a próxima semente. Para medir a separação entre todos os clientes-semente calculamos a separação média e a dispersão:

$$separação_média_i = \frac{\sum_{j \in C_semente} separação_{ij}}{num_sementes}$$

$$dispersão_i = \sqrt{\frac{\sum_{j \in C_semente} (separação_{ij} - separação_média_i)^2}{num_sementes}}$$

Por último, procuramos o cliente i de maior $separação_final_i$, a qual combina informações de $separação_média_i$ e $dispersão_i$.

$$separação_final_i = separação_média_i - \frac{dispersão_i}{coef_dispersão}, \text{ tal que } coef_dispersão \geq 1.$$

É desejável alta $separação_média_i$ porque indica que o cliente i é, em geral, incompatível com os outros clientes-semente. A baixa dispersão é incentivada porque reduz a possibilidade de se selecionar um cliente i com elevada $separação_média_i$, mas baixa $separação_{ij}$ entre algum cliente j . O $coef_dispersão$ é um parâmetro e deseja-se que seja menor ou igual a 1 porque a $dispersão_i$ deve ser um objetivo secundário, atuando como critério de desempate entre todos os clientes i de elevada $separação_média_i$.

Desta forma, a próxima semente é o cliente i^* em que

$$separação_final_{i^*} \geq separação_final_i \quad \forall i \in NI.$$

Para facilitar o entendimento destes conceitos, as Figuras 2.3 e 2.4 ilustram, através de uma analogia entre as unidades de medida $separação_{ij}$ e $distância_{ij}$, situações em que a dispersão é uma medida decisiva na seleção do próximo cliente-semente.

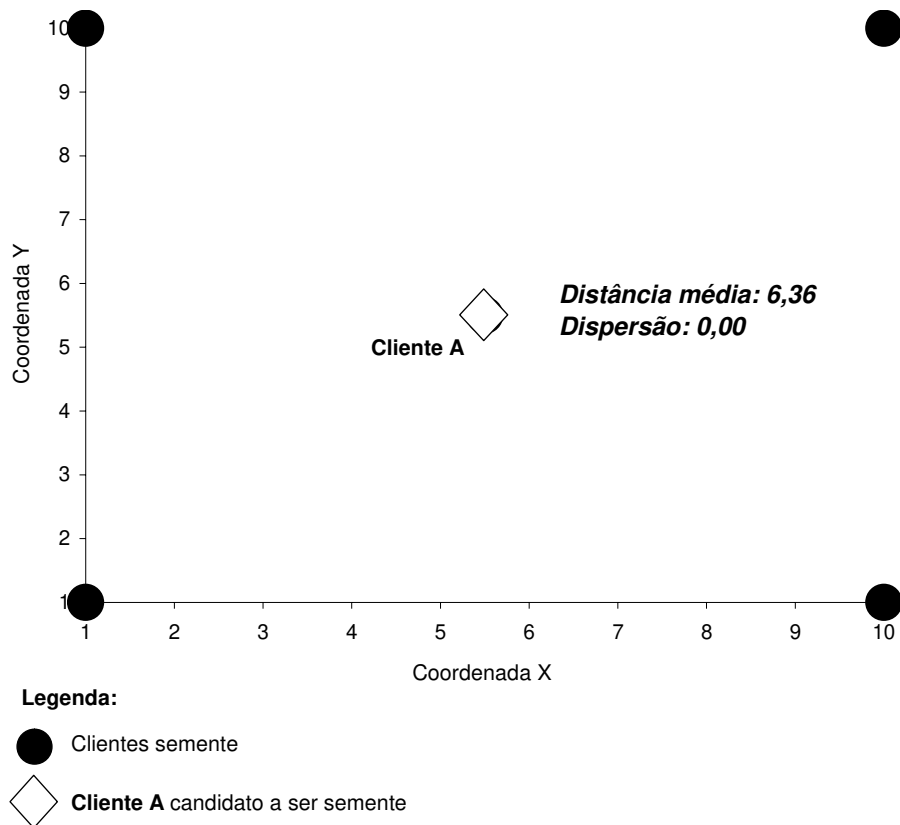


Figura 2.3. Análise da distância média e dispersão com Cliente A no centro do plano

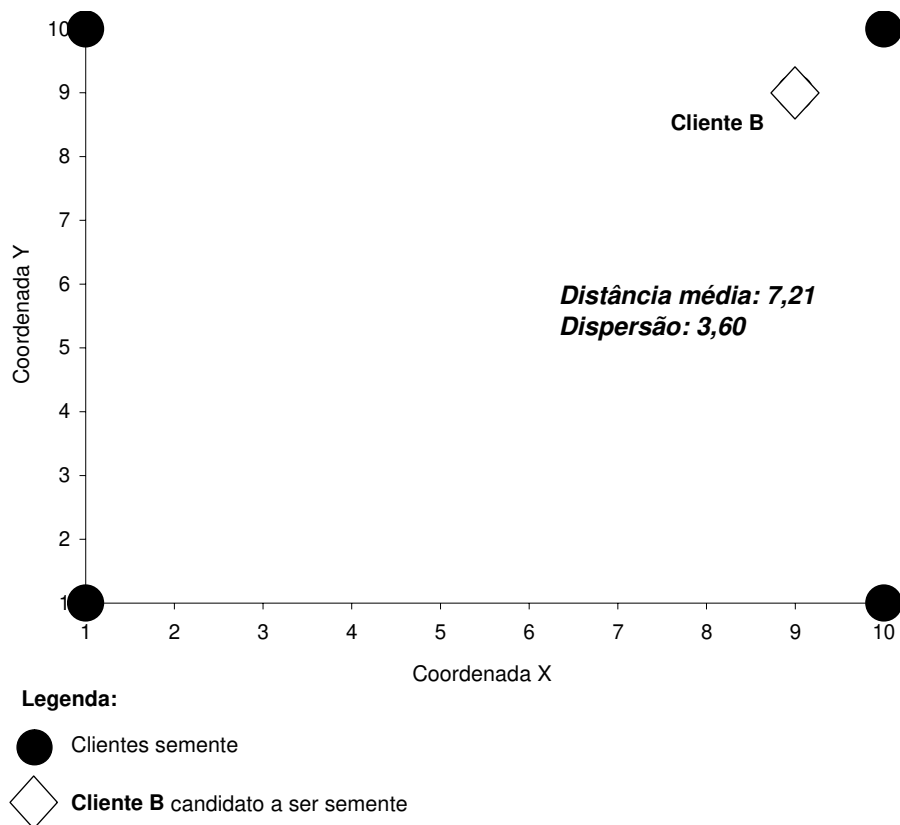


Figura 2.4. Análise da distância média e dispersão com Cliente B próximo ao canto superior direito do plano

Em ambas situações das Figuras 2.3 e 2.4 os clientes-semente estão situados nos extremos do plano. Na Figura 2.3, o cliente candidato (A) está localizado no centro e possui distância média com os clientes-semente de 6,36. Como as distâncias entre o cliente candidato (A) e qualquer cliente-semente são idênticas, a dispersão é igual a zero. Na Figura 2.4, outro cliente candidato (B) é situado próximo a um cliente-semente. A distância média desta situação é superior: 7,21. Porém, como as distâncias entre o cliente candidato (B) e os clientes-semente são diferentes, a dispersão é de 3,60.

Apesar da distância média do cliente (B) ser superior à de (A), visualmente percebe-se que (A) é um melhor candidato do que (B) – existe pouco sentido em iniciar uma rota em (B) uma vez que já existe outra rota iniciando no cliente-semente do canto superior direito. Conseqüentemente, utilizar como medida somente a distância média pode levar a decisões ruins. Ao considerar a dispersão, por exemplo, com $coef_dispersão = 1$, a distância final de (A) seria

$6,36 - (0 / 1) = 6,36$, enquanto a de (B) seria $7,21 - (3,6 / 1) = 3,61$. Portanto, (A) teria a maior distância final e seria selecionado.

Para encontrar a primeira semente utilizamos os mesmos dois critérios da heurística do Solomon (1987):

- a. Cliente mais distante do depósito;
- b. Cliente com o menor tempo final da janela de tempo.

Procedimento Acha_próximo_cliente_a_ser_inserido()

O objetivo deste procedimento é encontrar o cliente de maior urgência para ser inserido na solução. O conceito de urgência está relacionado ao tempo de folga que o cliente possui, sendo o tempo de folga de um cliente i numa rota j definido como:

$$\text{tempo_de_folga}_{(i, \text{rota } j)} = l_i - \text{menor_tempo_de_início_possível_de_atendimento}_{(i, \text{rota } j)}$$

O $\text{menor_tempo_de_início_possível_de_atendimento}_{(i, \text{rota } j)}$ é o tempo de início da primeira posição de inserção factível do cliente i na rota j , percorrendo a rota do primeiro ao último cliente visitado.

Em seguida, para cada cliente i é calculado o $\text{max_tempo_de_folga}_i$:

$$\text{max_tempo_de_folga}_i = \text{Máximo}(\text{tempo_de_folga}_{(i, \text{rota } j)}) \forall j \in R,$$

em que R é o conjunto de todas as rotas.

O próximo cliente i^* a ser inserido é aquele de menor $\text{max_tempo_de_folga}_i$:

$$\text{max_tempo_folga}_{i^*} \leq \text{max_tempo_folga}_i \forall i \in NI$$

Procedimento Busca_local_move_trecho()

Quando na fase I da heurística RR é impossível inserir todos os clientes nas rotas dos clientes-semente, a *Busca_local_move_trecho()* é aplicada na expectativa de se gerar novos pontos de inserção através da otimização da distância. Espera-se que rotas de menor distância possuam maior folga e, conseqüentemente, maior probabilidade de receber novos clientes.

Notação:

trecho_de_rota = qualquer conjunto de clientes em sequência de cardinalidade ≥ 1 ;

movimento = mover qualquer *trecho_de_rota* para a melhor posição de inserção entre todas as rotas. A melhor posição é aquela em que o *ganho* em distância é máximo, sendo $\text{ganho} = \text{máximo}(0, \text{distância_inicial} - \text{distância_final})$. Se o movimento resultar na economia de um veículo, $\text{ganho} = \text{distância_inicial} - \text{distância_final} + M$, em que M é um valor muito superior à ordem de grandeza de c_{ij} .

A *Busca_local_move_trecho()* realiza o *movimento* que proporciona o maior *ganho*, se $\text{ganho} > 0$.

2.3.4. Experimentos computacionais

A) Considerações sobre a implementação: estrutura de dados e movimentos

Após uma etapa inicial de testes computacionais, intuiu-se que uma maneira adequada de representar soluções de problemas de roteamento é tratá-las como a união de subconjuntos – ou trechos – de rotas. Constatou-se também que praticamente todos os movimentos já propostos pela literatura podem ser expressos como a união de até cinco trechos de rotas.

Para facilitar o entendimento, a Figura 2.5 ilustra um movimento de troca-cross, o qual requer até três trechos de rotas.

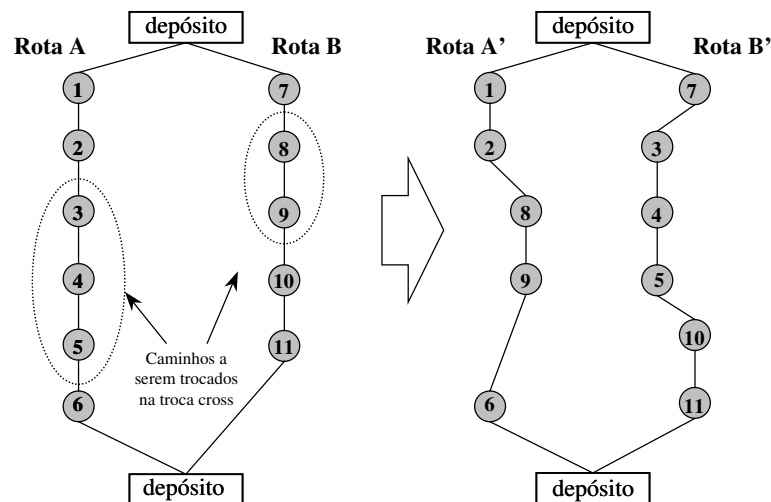


Figura 2.5. Exemplo de movimento da vizinhança troca-cross

Podemos verificar que a Rota A' é composta pela união dos trechos (1,2) da Rota A, com o trecho (8,9) da Rota B, e com o trecho (6,6) da Rota A. Da mesma forma, a Rota B' pode ser expressa como $(7,7) \cup (3,5) \cup (10,11)$.

Desta maneira, a implementação armazena as soluções como listas encadeadas de clientes, e os movimentos de qualquer vizinhança que alteram as soluções são executados através de um único procedimento, o qual recebe como parâmetros de entrada o cliente inicial e final de até cinco trechos. Por exemplo, para compor a rota B', a chamada do procedimento seria: *movimento(7,7, 3,5, 10,11)*.

Esta abordagem facilitou e tornou mais compreensível a implementação de todas as heurísticas e vizinhanças.

B) Conjunto de problemas teste

Solomon (1987) propôs seis tipos de problemas teste de 100 clientes, totalizando 56 problemas, classificados quanto à distribuição dos clientes no espaço e ao tamanho da janela de tempo:

- Distribuição dos clientes no espaço: Agrupada (C), Aleatória (R), Mista (RC);
- Tamanho da janela de tempo do depósito (apertada ou folgada) e relação entre a capacidade do veículo e as demandas dos clientes (também apertada ou folgada): quando ambas são apertadas, são admitidos poucos clientes por rota (problemas teste tipo 1), quando ambas são folgadas, são admitidos muitos clientes por rota (problemas teste tipo 2).

Homberger (2000) estendeu os problemas do Solomon para até 1000 clientes, e Kohl *et al.* (1999) trataram problemas reduzidos de 25 a 100 clientes. A Tabela 2.1 apresenta o conjunto de problemas teste foco deste estudo por tipo e número de clientes.

Tabela 2.1. Número de problemas teste por tipo e número de clientes

Tipo de Problema	Fonte e número de clientes		Total
	Solomon	Homberger	
	100	200	
C1	9	10	19
C2	8	10	18
R1	12	10	22
R2	11	10	21
RC1	8	10	18
RC2	8	10	18
Total	56	60	116

Todos os testes computacionais foram executados sobre os 56 problemas teste do Solomon (1987) para a seleção dos melhores métodos heurísticos. Uma vez encontrado os melhores métodos, executou-se testes para todos os 116 problemas teste.

No Anexo I está o melhor resultado obtido pela literatura para cada um dos problemas teste de até 100 clientes. O anexo foi baseado em Solomon (2004).

C) Resultados dos experimentos computacionais

Foram realizados testes computacionais para comparar o desempenho das três heurísticas construtivas descritas anteriormente: Clarke & Wright (1964), Solomon (1987) e RR. Apesar do objetivo desta seção ser realizar experimentos computacionais somente para heurísticas construtivas, foi necessário, em alguns testes, incorporar a *Busca_local_move_trecho()* nas heurísticas construtivas de Clarke & Wright (1964) e Solomon (1987) para que, além de tornar a comparação mais equilibrada, as heurísticas possam usufruir os mesmos benefícios da Fase II da RR.

Tabela 2.2. Testes das heurísticas construtivas

Nome do teste	Descrição
CWP	Heurística construtiva de Clarke & Wright versão paralela (com e sem busca local)
Solomon	Heurística de Solomon (com e sem busca local)
RR-A	Fases I e III da heurística RR
RR-B	Fases I, II e III da heurística RR

Os testes da heurística RR foram executados com os parâmetros da Tabela 2.3:

Tabela 2.3. Parâmetros dos testes da heurística RR

Teste	Escolha da primeira semente	Coefficiente Dispersão	Coefficiente incompatibilidade	Coefficiente tempo de espera
1_4_0.5_0.5	1 – Cliente mais distante	4	0.5	0.5
2_4_0.5_0.5	2 – Cliente mais urgente	4	0.5	0.5
1_1_0.75_0.25	1 – Cliente mais distante	1	0.75	0.25
2_1_0.75_0.25	2 – Cliente mais urgente	1	0.75	0.25

Para se chegar aos valores dos parâmetros da Tabela 2.3, que foram os que geraram os melhores resultados, testou-se a faixa de [1, 8] para o parâmetro *coef_dispersão* e todas as combinações do conjunto de valores {0; 0,25; 0,5; 0,75; 1} para os parâmetros *coef_inc* e *coef_te*.

Os testes foram executados em computadores Pentium IV 2.8 GHz, sistema operacional Linux Fedora Core 2. Um dos computadores utilizado foi adquirido com recursos da reserva técnica da bolsa de mestrado que apoiou esta dissertação, processo Fapesp nº 02/11907-2.

A Tabela 2.4 mostra o percentual de veículos adicionais, em relação ao melhor resultado da literatura, que as heurísticas encontraram para cada grupo de problemas teste. Quando aplicada, a *Busca_local_move_trecho()* considera todos os movimentos factíveis de todos os *trechos_de_rota*, ou seja, não há limitação para a cardinalidade dos conjuntos *trecho_de_rota*.

Tabela 2.4. Percentual de veículos adicionais sobre o melhor resultado da literatura – Heurísticas Construtivas para os 56 problemas teste do Solomon (1987)

Heurística	Teste	Tipo de problema							Class.*
		C1	C2	R1	R2	RC1	RC2	Todos	
RRA	1 4 0.5 0.5	11.1%	20.8%	20.8%	32.6%	26.4%	35.4%	24.4%	10
	2 4 0.5 0.5	8.9%	12.5%	20.8%	28.0%	26.0%	19.8%	19.7%	5
	1 1 0.75 0.25	10.0%	25.0%	21.1%	24.2%	28.1%	24.0%	21.9%	8
	2 1 0.75 0.25	7.8%	12.5%	18.3%	28.0%	25.0%	22.9%	19.3%	4
RRB	1 4 0.5 0.5	8.9%	20.8%	19.1%	28.0%	20.6%	22.9%	20.2%	6
	2 4 0.5 0.5	6.7%	4.2%	19.1%	16.7%	22.4%	15.6%	14.5%	1
	1 1 0.75 0.25	7.8%	20.8%	20.1%	19.7%	24.7%	7.3%	17.0%	3
	2 1 0.75 0.25	7.8%	4.2%	17.4%	18.9%	19.1%	18.8%	14.7%	2
Solomon	sem busca local	3.3%	25.0%	14.8%	43.2%	18.3%	34.4%	23.3%	9
	com busca local	2.2%	12.5%	14.8%	43.2%	18.3%	34.4%	21.3%	7
CWP	sem busca local	14.4%	58.3%	50.7%	297.7%	47.9%	261.5%	124.2%	12
	com busca local	6.7%	8.3%	40.7%	231.8%	38.3%	201.0%	90.7%	11

*A coluna Class. (Classificação) apresenta qual foi o 1º melhor método, 2º melhor, e assim por diante.

Observando os resultados, nota-se que, em geral, a heurística RR-B teve o melhor desempenho para a redução do número de veículos. A heurística de Solomon (1987), em ambas as versões, superou todas as outras heurísticas para os problemas teste do tipo 1 (C1, R1 e RC1) – problemas que admitem um menor número de clientes por rota. Para os problemas do tipo 2, Solomon (1987), mesmo com busca local, foi inferior à RR-A. Já a heurística de Clarke & Wright (1964) sempre gera soluções com um elevado número de veículos.

Na Tabela 2.5 temos os resultados para o desvio percentual médio entre a distância total percorrida e a melhor distância da literatura.

**Tabela 2.5. Desvio percentual médio da distância percorrida –
Heurísticas Construtivas para os 56 problemas teste do Solomon (1987)**

Heurística	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
RRA	1 4 0.5 0.5	77.6%	48.8%	28.1%	36.3%	24.8%	35.1%	41.1%	11
	2 4 0.5 0.5	72.8%	36.7%	28.3%	29.7%	26.9%	32.4%	37.3%	10
	1 1 0.75 0.25	71.8%	42.8%	26.2%	34.4%	35.4%	47.8%	41.9%	12
	2 1 0.75 0.25	71.0%	38.8%	25.0%	28.9%	28.8%	34.1%	37.0%	9
RRB	1 4 0.5 0.5	57.9%	23.3%	16.1%	13.3%	22.4%	18.5%	24.5%	7
	2 4 0.5 0.5	62.0%	15.7%	17.7%	12.0%	23.2%	15.0%	23.8%	5
	1 1 0.75 0.25	55.7%	18.6%	15.5%	13.5%	25.4%	21.6%	24.3%	6
	2 1 0.75 0.25	53.8%	20.6%	14.7%	9.9%	17.2%	14.1%	21.2%	4
Solomon	sem busca local	17.0%	27.4%	16.2%	37.2%	19.3%	40.0%	25.9%	8
	com busca local	3.6%	6.7%	6.7%	10.3%	8.0%	7.9%	7.3%	2
CWP	sem busca local	9.2%	19.5%	12.7%	10.5%	13.4%	13.5%	12.9%	3
	com busca local	3.3%	3.3%	8.1%	3.6%	8.6%	2.5%	5.0%	1

Os resultados revelam uma informação interessante: existe um balanço entre o número de rotas e a distância total percorrida para problemas de roteamento com janelas de tempo. Percebe-se que as heurísticas que geraram soluções com um número menor de veículos foram as mesmas que geraram soluções com a maior distância. A exceção é a heurística de Solomon (1987) para os problemas teste do tipo 1 – ótimos resultados tanto em número de veículos quanto em distância. Vale ressaltar o excelente desempenho da heurística de Clarke & Wright (1964) para o objetivo de minimização da distância.

A Tabela 2.6 apresenta agora os tempos computacionais.

**Tabela 2.6. Tempo médio computacional em segundos –
Heurísticas Construtivas para os 56 problemas teste do Solomon (1987)**

Heurística	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
RRA	1 4 0.5 0.5	0.3	0.4	0.9	0.5	0.8	0.6	0.6	6
	2 4 0.5 0.5	0.3	0.4	0.9	0.5	0.8	0.6	0.6	4
	1 1 0.75 0.25	0.3	0.5	0.9	0.4	0.9	0.6	0.6	5
	2 1 0.75 0.25	0.3	0.4	0.9	0.5	0.8	0.6	0.6	3
RRB	1 4 0.5 0.5	2.5	14.0	4.7	37.3	4.4	26.8	15.2	10
	2 4 0.5 0.5	1.9	11.5	4.1	35.6	3.8	22.2	13.5	9
	1 1 0.75 0.25	2.3	16.2	5.3	37.3	5.5	36.3	17.1	12
	2 1 0.75 0.25	2.3	15.3	5.1	42.0	4.3	27.7	16.5	11
Solomon	sem busca local	0.0	0.2	0.0	0.4	0.0	0.3	0.2	2
	com busca local	1.4	5.7	1.8	11.7	1.7	9.8	5.4	8
CWP	sem busca local	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
	com busca local	0.5	4.0	0.7	1.6	0.6	1.8	1.5	7

A heurística RR-B possui um elevado tempo computacional, gerado principalmente pelos reinícios e sucessivas buscas locais que auxiliam na redução do número de rotas. Já as heurísticas RR-A, Solomon e Clarke & Wright, possuem um tempo computacional pouco representativo – média inferior a um segundo por problema teste sem a aplicação da busca local.

Destacamos as seguintes conclusões:

- A heurística de Solomon (1987) é imbatível para os problemas teste do tipo 1;
- Existe um balanço entre reduzir o número de veículos e minimizar a distância total percorrida. Se o objetivo for minimizar a distância, a heurística Clarke & Wright é a mais indicada;
- A eficiência da aplicação da *Busca_local_move_trecho()* numa solução construída sem o objetivo de redução de rotas é baixa;
- Os problemas teste que admitem muitos clientes por rota – em virtude da folga nas restrições de capacidade e de janela de tempo – consomem muito tempo computacional na etapa da busca local (muitas possibilidades de inserção);
- Na média geral, a heurística construtiva de melhor resultado para a redução do número de veículos é RR-B_2_4_0.5_0.5, a qual pondera igualmente os conceitos de incompatibilidade e tempo de espera na seleção dos clientes-semente. Esta heurística exige um tempo computacional aproximadamente de 16 segundos por problema teste.

2.4. Busca local granular

Toth e Vigo (2003) introduziram a vizinhança granular como uma alternativa para métodos baseados em busca local. O benefício da vizinhança granular reside na maior eficiência da exploração do espaço da vizinhança, pois restringe a busca aos vizinhos com maior probabilidade de serem caminhos para as melhores soluções.

Os autores utilizaram a vizinhança granular no método busca tabu para o problema de roteamento de veículos capacitado (PRV). A seguir, é feita uma breve revisão da vizinhança empregada pelos autores para, posteriormente, apresentar a nossa implementação.

2.4.1. Descrição da vizinhança granular de Toth e Vigo (2003)

A vizinhança granular pode ser vista como uma implementação da estratégia de lista de candidatos de Glover e Laguna (1997). Seja $G = (V, A)$ um grafo completo em que $V = \{v_0, v_1, \dots, v_n\}$ representa o conjunto de nós e $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ o conjunto de arcos. Seja $G' = (V, A')$ um grafo reduzido em que $A' = \{(v_i, v_j) : v_i, v_j \in V, i \neq j, c_{ij} \leq \theta\} \cup I$. θ é o limiar granular, ou seja, o custo máximo que um arco pode ter para pertencer ao conjunto A' . I é o conjunto de arcos importantes, contendo todos os arcos que originam e terminam no depósito, mais os arcos que pertencem às melhores soluções.

Portanto, a estratégia é aplicar um método de busca que somente procure inserir arcos de A' na solução corrente. A inserção de cada arco de A' em uma solução exige a retirada de outros arcos para manter o conceito de rota ou ciclo. Existem diversas formas de retirar arcos, e cada forma induz tipos de movimento distintos, a saber, *crossover*, inserção de um ou dois clientes e troca de um par de clientes. A melhor combinação arco inserido-movimento é executada. Abaixo ilustramos cada um desses movimentos, adotando como notação o arco (a, b) como o elemento selecionado de A' .

1. *Crossover*: esta vizinhança foi originalmente proposta por Savelsbergh (1992) e por Potvin e Rousseau⁶ (1995) e consiste na troca de 2 pares de arcos gerando o cruzamento de duas rotas. No exemplo da Figura 2.6, o par $\{(a, 2), (5, b)\}$ é trocado pelo par $\{(a, b), (5, 2)\}$.

⁶ Potvin (1995) denotou esta vizinhança de 2-opt*.

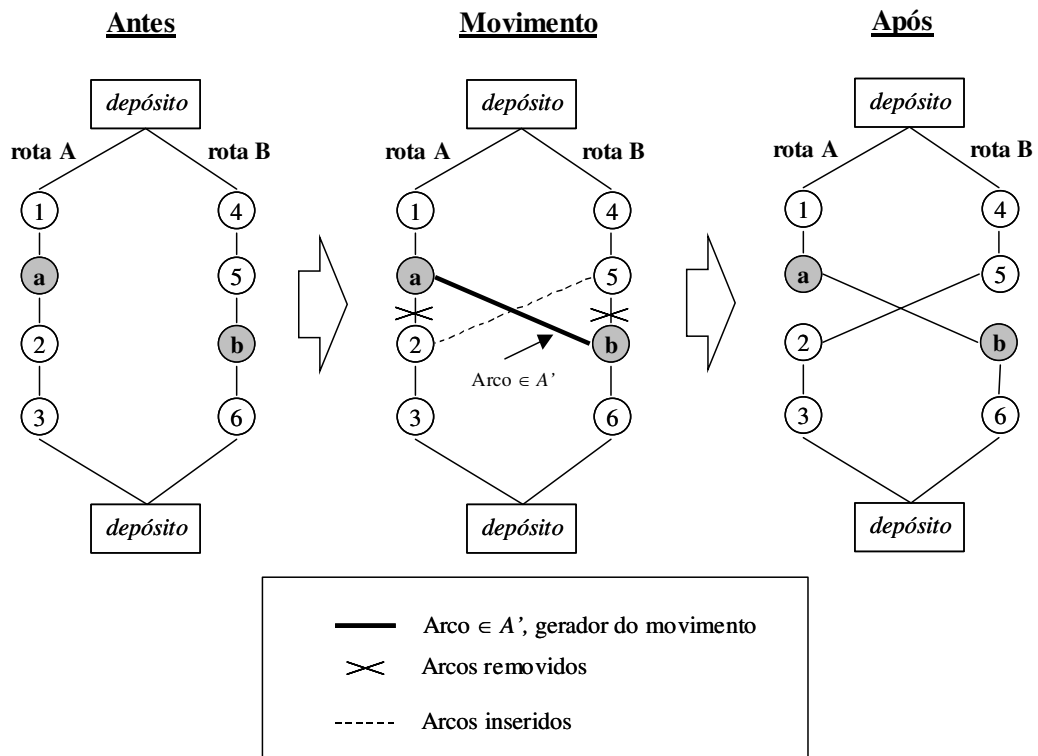


Figura 2.6. Exemplo de *crossover*

2. Inserção de um ou dois clientes: esta vizinhança pode ser vista como um caso particular da vizinhança proposta por Osman e Christofides (1989) denominada λ -troca (λ -interchange). Esta vizinhança troca subconjuntos de cardinalidade de até λ elementos de duas rotas distintas. Quando um destes subconjuntos é nulo, é feita a inserção. No exemplo da Figura 2.7, o cliente (a) foi inserido na rota B.

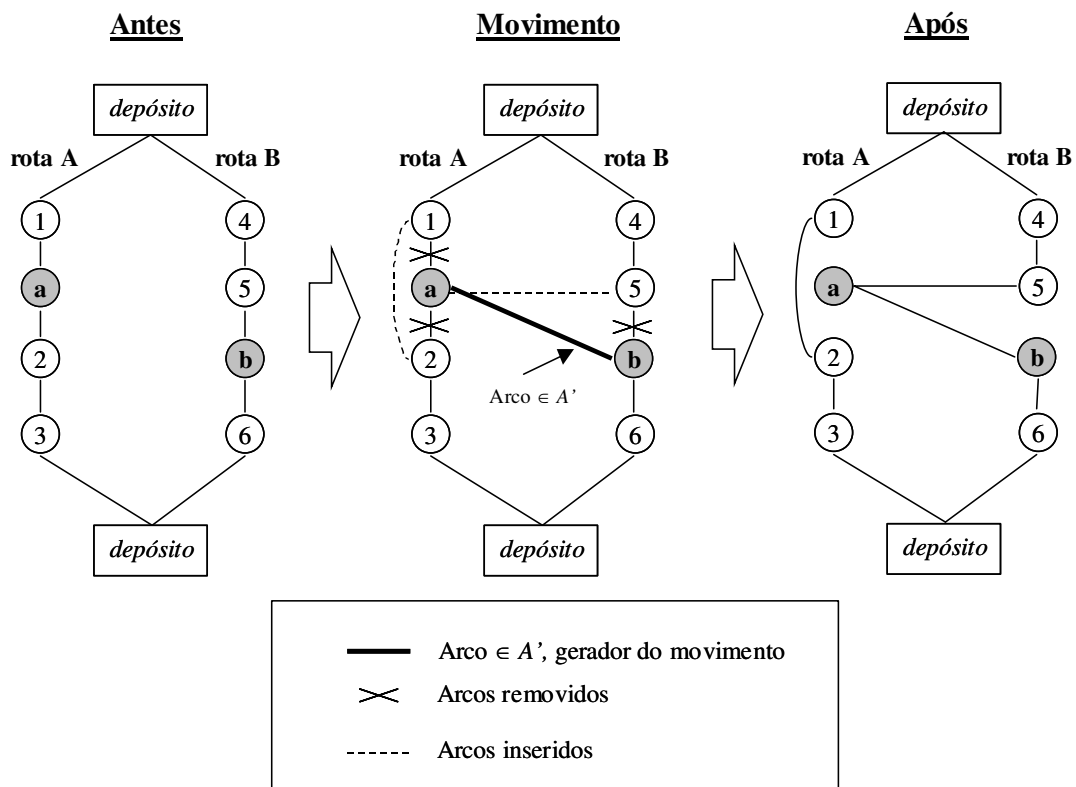


Figura 2.7. Exemplo de inserção de um cliente

Na Figura 2.8, os clientes (b) e (6) são inseridos na rota A através do acréscimo dos arcos $\{(a, b), (6, 2), (5, depósito)\}$ e remoção dos arcos $\{(5, b), (6, depósito), (a, 2)\}$.

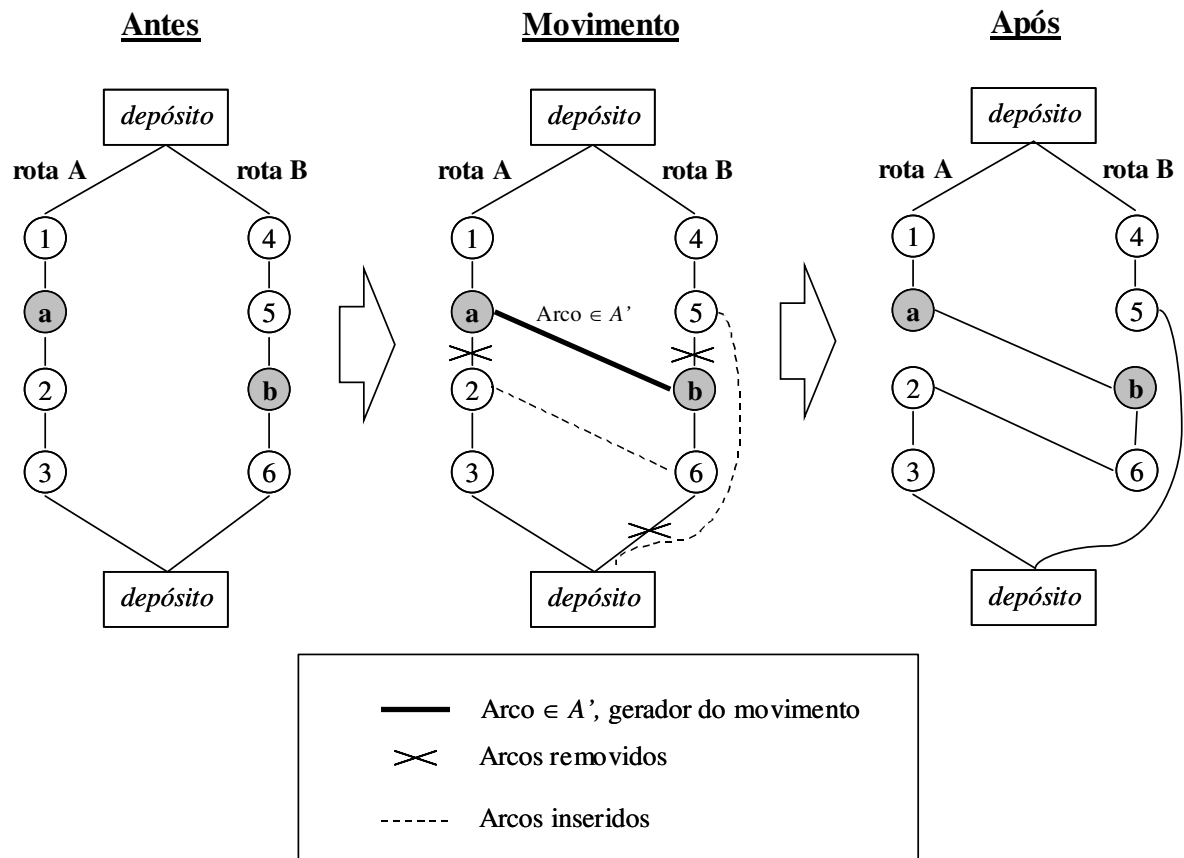


Figura 2.8. Exemplo de inserção de dois clientes

3. Troca de um par de clientes: vizinhança derivada do λ -troca, em que um cliente de cada rota é selecionado e trocado. Os clientes (2) e (b) foram trocados na Figura 2.9.

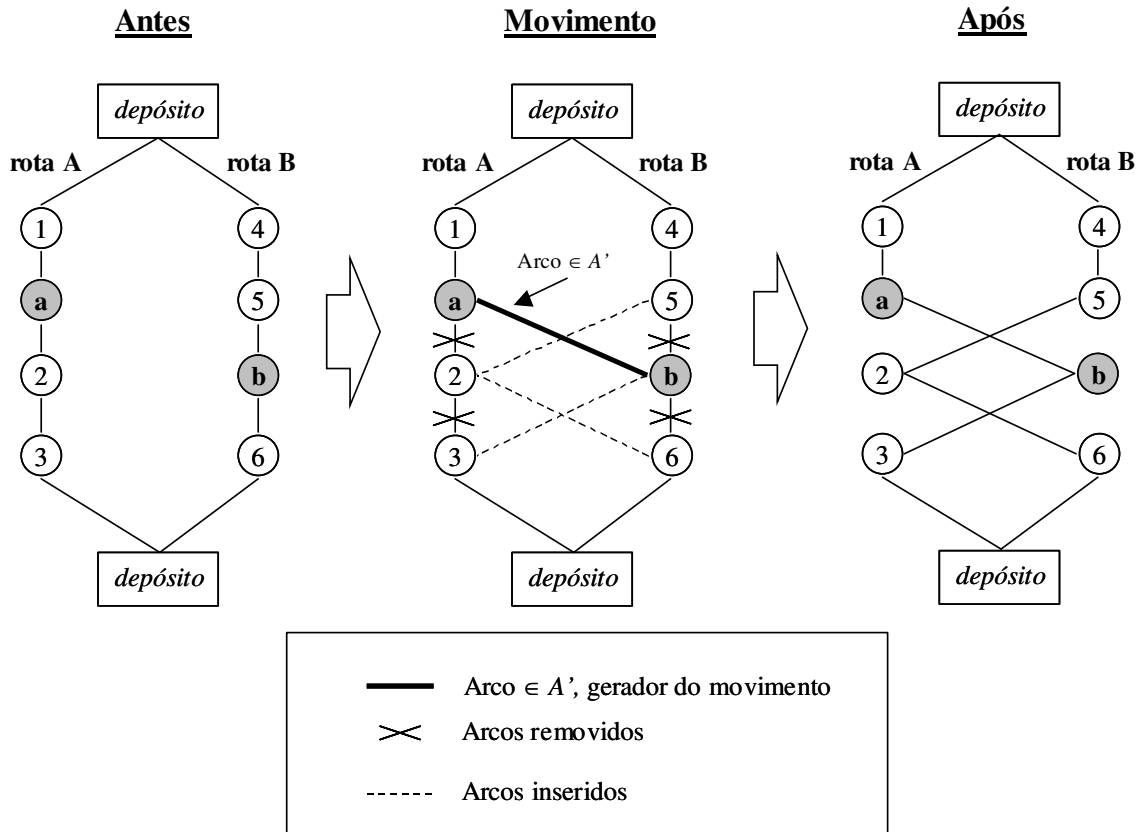


Figura 2.9. Exemplo de troca de um par de clientes

2.4.2. Busca local proposta

Nossa proposta para a busca local teve como base o trabalho de Toth e Vigo (2003) apresentado anteriormente.

Primeiramente o conjunto A' é construído da seguinte forma:

1. Calcula-se o número de arcos do grafo completo: $num_completo = n(n+1)$, em que n é o número de clientes;
2. Calcula-se o número de arcos que são originados ou terminados no depósito: $num_depósito = 2n$;

3. Calcula-se a quantidade de arcos potenciais a serem inseridos no grafo A' :

$$\text{num_potencial} = \text{num_completo} - \text{num_deposito} = n(n+1) - 2n = n^2 - n;$$
4. Dado um parâmetro η com a fração de arcos que serão adicionados ao grafo A' , seleciona-se os $\eta \times \text{num_potencial}$ menores arcos;
5. A quantidade final de arcos no grafo A' é: $|A'| = 2n + \eta \times \text{num_potencial}^*$;
6. Para facilitar a implementação, ao invés de se armazenar numa lista os $2n + \eta \times \text{num_potencial}$ arcos, guarda-se somente as $\frac{2n + \eta \times \text{num_potencial}}{2}$ arestas. As arestas são armazenadas no conjunto H .

Uma vez criado o grafo granular, para cada aresta (a, b) da lista são pesquisados os seguintes movimentos:

1. Movimento intra-rota troca-arcos (a, b) : troca de dois pares de arestas da mesma rota;
2. Movimento intra-rota inserção (a, b) : inserção de qualquer trecho de rota em outra posição da mesma rota;
3. Movimento inter-rota *crossover* (a, b) : troca de dois pares de arestas de rotas distintas;
4. Movimento inter-rota inserção (a, b) e (b, a) : inserção de qualquer trecho de uma rota em outra;
5. Movimento inter-rota troca (a, b) e (b, a) : troca de quaisquer trechos de rotas distintas.

A descrição de cada movimento pode ser encontrada no Anexo G.

Finalmente, o movimento que resulta no maior *ganho* é executado, enquanto $\text{ganho} > 0$. Lembre que se o movimento resultar na economia de um veículo, $\text{ganho} = \text{distância_inicial} - \text{distância_final} + M$, em que M é um valor muito superior à ordem de grandeza de c_{ij} . Logo, o valor de M confere prioridade aos movimentos que diminuem o número de veículos da solução.

2.4.3. Resultados dos experimentos computacionais

As Tabelas 2.7, 2.8 e 2.9 apresentam os resultados da busca local granular. Os testes foram executados com $\eta = 1$ e duas soluções de partida: o teste Gra (RR) utiliza como partida a solução

* Note que quando $\eta = 1$ a busca considera o grafo completo.

encontrada pela heurística RR-B_2_4_0.5_0.5 enquanto o teste Gra (Solomon) utiliza a solução de Solomon (1987) sem busca local.

Foram considerados também dois tipos de seleção do movimento: melhor movimento por iteração (*best improvement*) ou primeiro movimento que cause melhora da solução corrente (*first improvement*).

Tabela 2.7. Percentual de veículos adicionais sobre o melhor resultado da literatura – Busca local granular para os 56 problemas teste do Solomon (1987)

Heurística	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Gra (RR)	<i>first</i>	0.0%	0.0%	16.8%	16.7%	15.8%	15.6%	11.4%	1
	<i>best</i>	1.1%	0.0%	16.4%	16.7%	18.2%	15.6%	11.8%	2
Gra (Solomon)	<i>first</i>	1.1%	8.3%	13.0%	43.2%	17.3%	30.2%	19.4%	4
	<i>best</i>	0.0%	8.3%	13.9%	35.6%	14.3%	27.1%	17.1%	3

Tabela 2.8. Desvio percentual médio da distância percorrida – Busca local granular para os 56 problemas teste do Solomon (1987)

Heurística	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Gra (RR)	<i>first</i>	11.4%	3.2%	4.3%	8.3%	6.1%	8.9%	7.0%	4
	<i>best</i>	7.9%	2.2%	4.1%	7.3%	5.9%	9.2%	6.1%	3
Gra (Solomon)	<i>first</i>	2.2%	3.7%	3.2%	3.4%	3.4%	2.2%	3.1%	1
	<i>best</i>	2.2%	3.2%	3.6%	4.0%	2.5%	6.3%	3.6%	2

Tabela 2.9. Tempo médio computacional em segundos⁷ – Busca local granular para os 56 problemas teste do Solomon (1987)

Heurística	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Gra (RR)	<i>first</i>	15.8	49.5	8.7	76.1	10.3	62.2	36.8	1
	<i>best</i>	14.9	65.2	12.4	109.6	10.0	56.5	45.4	2
Gra (Solomon)	<i>first</i>	5.1	85.0	3.8	91.7	6.4	103.6	47.5	3
	<i>best</i>	9.3	141.7	9.9	366.5	10.1	173.2	122.0	4

Não foram encontradas diferenças significativas no desempenho das modalidades *first* e *best*. A modalidade *best* do teste Gra (Solomon) é ligeiramente superior à correspondente *first* no

⁷ Inclui o tempo computacional da heurística construtiva.

objetivo minimizar número de veículos. Contudo, seu tempo computacional foi quase o triplo, causado principalmente pela execução dos tipos de problema R2 e RC2⁸.

É interessante notar que a busca local granular tem melhor desempenho – tanto em qualidade quanto em tempo computacional – quanto lhe é entregue uma solução de partida de melhor qualidade.

A contribuição da busca local granular na redução do percentual médio do número de veículos foi de 14,5% para 11,4% (RR) e de 23,3% para 17,1% (Solomon). Esta pequena redução demonstra que a busca local granular não é eficiente para diminuir significativamente o número de veículos. Entretanto, a maior contribuição da busca local granular é na redução da distância total percorrida: de 23,8% para 6,1% (RR) e de 25,9% para 3,1% (Solomon).

Outra etapa de testes foi desenvolvida para quantificar a variação da qualidade da solução em relação ao número de arestas pertencentes ao conjunto H . Quanto maior o número de arestas do conjunto H , maior é o tamanho da vizinhança, o que tende a aumentar tanto a qualidade das soluções quanto o tempo computacional. Os testes foram executados com $\eta \in \{25\%, 50\%, 75\%, 100\%\}$.

As Figuras 2.10 e 2.11 mostram o efeito da redução do conjunto de arestas do grafo granular na qualidade e no tempo computacional para a busca local Gra (RR) na modalidade *best improvement*.

⁸ Este comportamento é uma acentuação do comportamento já percebido na heurística Solomon com busca local da Tabela 2.6.

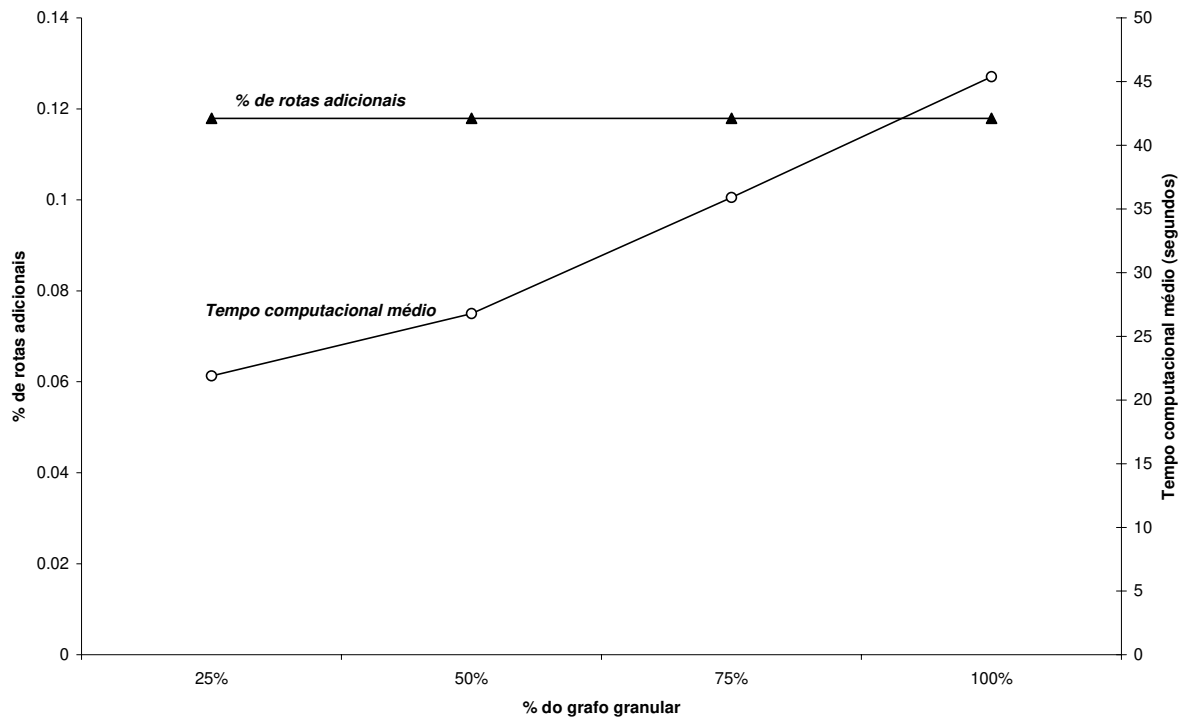


Figura 2.10. Percentual médio de rotas adicionais e tempo computacional por percentual do grafo granular

A diferença entre o tempo computacional de um grafo de 25% e de outro completo (100%) chega a ser mais do que o dobro (107%), não gerando redução de rota adicional.

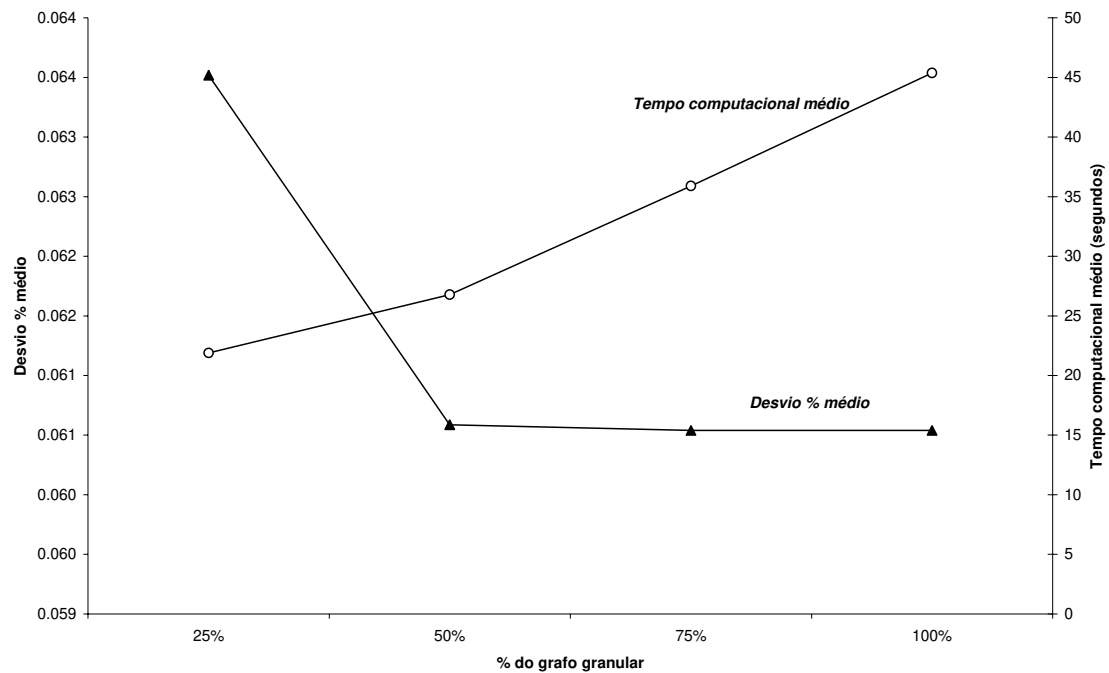


Figura 2.11. Desvio percentual médio da distância e tempo computacional por percentual do grafo granular

O gráfico da Figura 2.11 demonstra que não há ganhos significativos na distância para grafos acima de 50%. O tempo computacional possui o mesmo comportamento da Figura 2.10: o tempo para o grafo completo é 69% superior ao tempo do grafo com $\eta = 50\%$.

Logo, é razoável adotar $\eta \in [0.25, 0.50]$.

2.5. Busca tabu granular

2.5.1. Vizinhanças e movimentos

A busca tabu executa a cada iteração o movimento não tabu de menor degradação da distância total percorrida. No total há cinco movimentos possíveis – todos provenientes da vizinhança granular – mostrados na Tabela 2.10:

Tabela 2.10. Movimentos da busca tabu

Tipo de movimento	Movimento
Intra-rota	troca-arcos
	Inserção
Inter-rota	<i>Crossover</i>
	Inserção
	Troca

Quando não há movimento que proporcione uma melhora da solução atual, os movimentos intra-rota são aqueles que, geralmente, causam a menor degradação do valor da função objetivo por provocar pequenas alterações na solução. Logo, a busca tabu ficaria tendenciosa a executar movimentos intra-rota. Desta maneira, para evitar uma busca excessiva nas vizinhanças intra-rota, foi definida uma hierarquia de movimentos a serem executados, mostrada na Tabela 2.11.

Notação:

maior_ganho_intra: maior ganho em distância das vizinhanças intra-rota;

melhor_movimento_intra: o movimento correspondente ao *maior_ganho_intra*;

maior_ganho_inter: maior ganho em distância das vizinhanças inter-rota;

melhor_movimento_inter: o movimento correspondente ao *maior_ganho_inter*;

Tabela 2.11. Hierarquia de movimentos executados pela busca tabu

<i>Maior_ganho_intra</i>	<i>Maior_ganho_inter</i>	Desempate	Movimento executado
>0	>0	<i>maior_ganho_intra</i> > <i>maior_ganho_inter</i>	<i>melhor_movimento_intra</i>
>0	>0	<i>maior_ganho_intra</i> < <i>maior_ganho_inter</i>	<i>melhor_movimento_inter</i>
>0	≤ 0	-	<i>melhor_movimento_intra</i>
≤ 0	>0	-	<i>melhor_movimento_inter</i>
<0	<0	-	<i>melhor_movimento_inter</i>

Basicamente, essas regras garantem que durante a “descida” (melhoria da distância total percorrida) o movimento de maior ganho é executado. Durante a “subida” (degradação da distância total percorrida) sempre são executados movimentos inter-rotas. Esta hierarquia é fundamental para que os movimentos mais influentes (inter-rota) possam ser executados, viabilizando a busca em outras regiões do espaço de soluções.

2.5.2. Memória de curto prazo e atributo tabu

Durante a busca são coletadas informações da solução numa memória de curto prazo, a qual armazena dois tipos de informação:

1. Arestas inseridas na solução: todos os cinco movimentos inserem arestas na solução. Dentre as arestas inseridas, a aresta geradora (a, b) sempre será acrescentada. Como a aresta (a, b) pertence a um conjunto de “arestas promissoras”, é interessante que esta aresta permaneça na solução por algumas iterações. Portanto, a primeira parte da memória de curto prazo armazena qual foi a última iteração em que a aresta geradora foi inserida;
2. Arestas removidas da solução: para cada movimento foram analisados quais são as arestas removidas – vide Anexo G. A fim de evitar ciclagem, é conveniente impedir que arestas recém removidas sejam inseridas novamente. A segunda parte da memória de curto prazo armazena qual foi a última iteração em que cada aresta foi removida.

Foi associada uma faixa de duração tabu a cada informação da memória de curto prazo:

- Uma vez inserido uma aresta geradora (a, b), esta deve permanecer na solução por um número $iter_fica \in [iter_fica_min, iter_fica_max]$ de iterações;

- Uma vez removida uma aresta (x, y) , esta não pode voltar para a solução por um número $iter_não_volta \in [iter_não_volta_min, iter_não_volta_max]$ de iterações.

Os valores de $iter_fica$ e $iter_não_volta$ são sorteados dentro de seus respectivos intervalos a cada $passos_muda_duração_tabu$ iterações. Dependendo da versão da busca tabu, este parâmetro foi fixado em 4 ou 8 iterações.

É importante salientar que existe uma relação entre os valores de $iter_fica$ e $iter_não_volta$ com o número de arestas factíveis e não-tabu. Suponha um exemplo em que $n = 50$ e $\eta=25\%$. Logo $|A'| = 2 \times 50 + 0,25(50^2 - 50) = 712$ arcos distintos, ou, $|H| = 356$ arestas.

Assumindo que neste exemplo os clientes são espalhados uniformemente no espaço, em geral 30% a 40% das 356 arestas geraria pelo menos um movimento factível ao avaliar todos os cinco movimentos da Tabela 2.10. Caso a disposição dos clientes fosse agrupada, o número de movimentos factíveis seria menor – gerados a partir de 10% a 25% das 356 arestas.

Contudo, nem todas as arestas que geram movimentos factíveis podem ser selecionadas para ingressarem na solução, porque, após a verificação das condições tabu, alguns movimentos factíveis são proibidos. O número de arestas que geram movimentos factíveis, mas proibidos, fica em torno de 10% a 15% do total de 356 arestas. Então, se os valores de $iter_fica$ e $iter_não_volta$ são mantidos elevados por muito tempo, pode-se chegar numa situação em que há poucas arestas (ou nenhuma) que geram movimentos factíveis e não-tabu. E se não houver arestas que geram tais movimentos, a busca tabu não tem nenhum movimento a executar.

Para evitar este problema, a busca tabu conta o número de arestas pertencentes a H que gerariam movimentos factíveis e não-tabu e armazena este valor em $num_arestas_candidatas$. A partir de $num_arestas_candidatas$, o método calcula $perc_arestas_candidatas$:

$$perc_arestas_candidatas = num_arestas_candidatas / |H|$$

O objetivo é manter $perc_arestas_candidatas$ maior ou igual a 10%, reduzindo os valores de $iter_fica$ e $iter_não_volta$ quando $perc_arestas_candidatas$ assumir valores muito baixos.

Além deste conceito, a determinação das faixas $[iter_fica_min, iter_fica_max]$ e $[iter_não_volta_min, iter_não_volta_max]$ é feita através de ciclos (Glover e Laguna, 1997). As Tabelas 2.12. e 2.13. descrevem o ciclo da memória de curto prazo.

Tabela 2.12. Ciclo dos valores das faixas das durações tabu
(perc_arestas_candidatas $\geq$ 10%)

Etapa do ciclo	Objetivo da etapa	Número do bloco de 4 ou 8 iterações	Faixa do número de iterações tabu			
			<i>iter_fica_min</i>	<i>iter_fica_max</i>	<i>iter_não_volta_min</i>	<i>iter_não_volta_max</i>
A	- Buscar um mínimo local.	1	2	4	2	4
		2	2	4	2	4
		3	2	4	2	4
		4	2	4	2	4
B	- Efetuar pequena mudança na busca.	5	5	10	5	10
C	-Buscar um novo mínimo local.	6	2	4	2	4
		7	2	4	2	4
		8	2	4	2	4
D	- Efetuar grande mudança na busca.	9	10	15	10	15
		10	10	15	10	15

Tabela 2.13. Ciclo dos valores das faixas das durações tabu
(perc_arestas_candidatas < 10%)

Etapa do ciclo	Objetivo da etapa	Número do bloco de 4 ou 8 iterações	Faixa do número de iterações tabu			
			<i>iter_fica_min</i>	<i>iter_fica_max</i>	<i>iter_não_volta_min</i>	<i>iter_não_volta_max</i>
A	- Buscar um mínimo local.	1	1	4	1	4
		2	1	4	1	4
		3	1	4	1	4
		4	1	4	1	4
B	- Efetuar pequena mudança na busca.	5	2	5	2	5
C	-Buscar um novo mínimo local.	6	1	4	1	4
		7	1	4	1	4
		8	1	4	1	4
D	- Efetuar grande mudança na busca.	9	3	6	3	6
		10	3	6	3	6

Basicamente, o ciclo contém etapas com baixas durações tabu (A e C) que buscam por um mínimo local, seguidas por etapas de mudanças (B e D), sendo que há mudanças de baixa e alta intensidade.

A cada bloco de iterações completadas novas durações tabu são sorteadas. Ao término do bloco 10, o ciclo é recommençado e sorteiam-se durações tabu dentro da faixa do bloco 1.

2.5.3. Critério de aspiração

O critério de aspiração adotado foi o clássico de Glover e Laguna (1997): um movimento tabu é executado se a solução resultante for melhor do que a melhor solução encontrada até o momento – atualização da incumbente.

2.5.4. Memória de longo prazo, intensificação e diversificação

O número de iterações que cada aresta ficou presente na solução é armazenado numa matriz simétrica de residência.

Esta matriz é utilizada para o cálculo de penalidades, as quais motivam movimentos de intensificação ou diversificação. Os movimentos de intensificação procuram inserir na solução as arestas que mais estiveram presentes durante a trajetória da busca, enquanto os movimentos de diversificação procuram inserir as arestas que menos estiveram presentes.

O objetivo dos movimentos de intensificação é explorar com mais detalhe a região próxima de um mínimo local. Espera-se que as arestas inseridas possam combinar com novos movimentos que possibilitariam encontrar um novo mínimo local.

O objetivo dos movimentos de diversificação é escapar da região atratora do mínimo local, para que outras partes do espaço de solução possam ser exploradas. A aposta é que existam outros mínimos locais de melhor qualidade nessas regiões.

A relação entre a matriz de residência, as penalidades e os movimentos de intensificação e diversificação é feita da seguinte forma:

- A busca tabu sempre executa o movimento não-tabu de maior *ganho*;
- O valor do movimento, ou *ganho*, é calculado da seguinte forma: $ganho = \text{distância total da solução antes do movimento} - \text{a distância total da solução após o movimento}$. Se $ganho > 0$, o movimento promove um decréscimo no valor da função objetivo. Se $ganho < 0$, o movimento promove um acréscimo (degradação) no valor da função objetivo;

- Num movimento de intensificação, $ganho = \text{distância total da solução antes do movimento} - \text{distância total da solução após o movimento} + \text{penalidade}$;
- Num movimento de diversificação, $ganho = \text{distância total da solução antes do movimento} - \text{distância total da solução após o movimento} - \text{penalidade}$;
- Para definir a *penalidade* utilizaremos a seguinte notação (para cada movimento):
 - H_e = conjunto das arestas que estão entrando na solução;
 - H_s = conjunto das arestas que estão saindo da solução;
 - $matriz_residência(a)$ = quantidade de iterações em que a aresta a esteve presente em soluções durante a trajetória da busca;
 - $maior_residência = matriz_residência(a^*)$, tal que $matriz_residência(a^*) \geq matriz_residência(a)$, $\forall a \in \{a=(v_x, v_y) : v_x, v_y \in V, v_x \neq v_y\}$;
 - $maior_aresta = c_{ij}^*$, tal que $c_{ij}^* \geq c_{ij} \forall i, j$

A *penalidade* é definida como:

$$penalidade = \left(\frac{\sum_{a_1 \in H_e} matriz_residência(a_1) - \sum_{a_2 \in H_s} matriz_residência(a_2)}{maior_residência} \right) \times maior_aresta$$

A expressão da *penalidade* calcula, para as arestas envolvidas no movimento, o saldo do número de iterações em que as arestas permaneceram nas soluções durante a trajetória da busca. Este valor é normalizado pela *maior_residência* e em seguida multiplicado pela *maior_aresta*. Multiplica-se por *maior_aresta* para que a penalidade tenha a mesma ordem de grandeza que as distâncias da fórmula do *ganho* (c_{ij}). O resultado é uma distância positiva se o saldo da residência das arestas inseridas menos removidas for positivo (inserção de arestas muito frequentes), ou negativa caso contrário (inserção de arestas pouco frequentes). Este resultado é somado ou subtraído ao *ganho*, dependendo do tipo de movimento, direcionando a seleção dos movimentos da busca tabu.

2.5.5. Memorização de soluções visitadas – conjunto de elite

Durante toda a trajetória da busca, cada solução proveniente de qualquer movimento é considerada para ser armazenada num repositório de soluções visitadas de alta qualidade, denominado de conjunto de elite.

Este conjunto contém soluções boas e distintas. Para quantificar a diferença estrutural entre duas soluções, foi implementada uma medida de distância baseada em Mazzini (1998):

Notação:

a_1 = número de arestas da solução 1;

a_2 = número de arestas da solução 2;

$a_{1 \cap 2}$ = número de arestas coincidentes entre as soluções 1 e 2;

$$distância(solução_1, solução_2) = \frac{\frac{a_1 - a_{1 \cap 2}}{a_1} + \frac{a_2 - a_{1 \cap 2}}{a_2}}{2}$$

Observe que:

- $distância(solução_1, solução_2) = distância(solução_2, solução_1)$;
- $0 \leq distância(solução_1, solução_2) \leq 1$;
- Quando $distância(solução_1, solução_2) = 0$, $solução_1 = solução_2$;
- Quando $distância(solução_1, solução_2) = 1 \Rightarrow a_{1 \cap 2} = 0 \Rightarrow solução_1$ é completamente distinta da $solução_2$.

Considerando esta medida de distância, uma solução somente ingressa no conjunto de elite nos seguintes casos:

- 1) A solução é melhor do que a melhor solução do conjunto;
- 2) O conjunto não está totalmente preenchido e a solução é suficientemente distinta de todas as outras soluções do conjunto;
- 3) O conjunto está totalmente preenchido, porém a nova solução é melhor do que a pior solução do conjunto e seu ingresso aumentaria a menor distância entre todas as soluções e/ou a distância média entre todas as soluções.

Quando uma solução é acrescentada a um conjunto preenchido, a solução que deixa de pertencer ao conjunto é aquela cuja substituição pela nova solução gera o maior incremento da menor distância e/ou distância média.

O detalhe das regras de ingresso no conjunto de elite pode ser encontrado no Anexo K.3.

2.5.6. Reinício a partir do conjunto de elite – baseado em Rochat e Taillard (1995)

O conjunto de elite é utilizado para recomençar o processo de busca. O reinício é feito da seguinte forma:

1. Seja s_nova uma solução vazia que conterà a solução resultante do procedimento de reinício;
2. Crie um repositório de rotas que armazene todas as rotas das soluções de elite que tenham no mínimo 2 clientes. Seja tot_rotas o número de rotas contidas pelo repositório;
3. Associe a cada rota do repositório um rótulo igual ao valor da função objetivo da solução da qual a rota é proveniente;
4. Ordene as rotas do repositório por ordem decrescente de rótulo;
5. Associe um índice $i = 1$ para a primeira rota da lista ordenada, $i = 2$ para a segunda, e assim por diante, até $i = tot_rotas$;
6. Calcule $soma_índices = [tot_rotas \times (tot_rotas + 1)] / 2$;
7. Sorteie um número aleatório $sorte \in [0, 1]$;
8. Faça $acumulado = 1$ e $cont_posição = 1$;
9. Enquanto $acumulado / soma_índices < sorte$, faça:
 - 9.1. $cont_posição = cont_posição + 1$;
 - 9.2. $acumulado = acumulado + cont_posição$;
 - 9.3. Retorne ao passo 9.
10. Acrescente a rota do repositório que tem como índice $cont_posição$ à solução s_nova ;
11. Remova do repositório todas as rotas que contenham pelo menos um cliente da rota do passo anterior.
12. Se o repositório ainda contiver rotas, faça tot_rotas igual ao número de rotas restantes do repositório e vá para o passo 5. Caso contrário → FIM.

Após este procedimento, s_{nova} conterá um conjunto de rotas do repositório (prioridade para as rotas das soluções de melhor qualidade) e, em geral, um conjunto de clientes sem roteiro. Para estes clientes é criado um roteiro simples: o veículo parte do depósito, visita o cliente e em seguida retorna ao depósito.

Conseqüentemente, s_{nova} possivelmente conterá mais rotas do que necessário. Contudo, a tarefa de reduzir as rotas de s_{nova} é passada para a própria busca tabu, a qual a realiza logo nas primeiras iterações após o reinício.

2.5.7. Critério de parada

A busca tabu termina quando são realizadas $4 \times n$ iterações, em que n = número de clientes do problema teste.

Uma iteração corresponde a encontrar e executar o melhor movimento de todas as arestas do conjunto H .

2.5.8. Descrição dos métodos heurísticos baseados na busca tabu granular

Foram testadas quatro versões da busca tabu granular, cada uma constituída por diferentes combinações dos componentes apresentados anteriormente. A Tabela 2.14 mostra quais são os componentes presentes em cada versão.

Tabela 2.14. Composição dos métodos baseados na busca tabu granular

Componente	Método heurístico			
	<i>Tabu_simples</i>	<i>Tabu_LP</i>	<i>Tabu_reinício</i>	<i>Tabu_misto</i>
2.5.1. Vizinhanças e movimentos	✓	✓	✓	✓
2.5.2. Memória de curto prazo e atributo tabu	✓	✓	✓	✓
2.5.3. Critério de aspiração	✓	✓	✓	✓
2.5.4. Memória de longo prazo, intensificação e diversificação		✓		✓
2.5.5. Memorização de soluções visitadas – conjunto de elite			✓	✓
2.5.6. Reinício a partir do conjunto de elite – baseado em Rochat e Taillard (1995)			✓	✓
2.5.7. Critério de parada	✓	✓	✓	✓

Todos os métodos utilizam a mesma vizinhança, praticam os mesmos movimentos e obedecem aos mesmos critérios de aspiração e de parada. A seguir detalharemos as diferenças de cada um dos métodos.

A) *Tabu_simples*

A busca *Tabu_simples* utiliza somente a memória de curto prazo e altera as durações tabu segundo as Tabelas 2.12 e 2.13. A Tabela 2.15 mostra seis versões do método *Tabu_simples*, para cada atributo tabu:

Tabela 2.15. Versões da busca *Tabu_simples*

Número de iterações por bloco do ciclo (vide Tabelas 2.12 e 2.13)	Atributo tabu		
	Aresta geradora (<i>a</i> , <i>b</i>) inserido deve permanecer na solução por <i>iter_fica</i> iterações	Arestas removidas não podem retornar à solução por <i>iter_não_volta</i> iterações	Ambas condições devem ser satisfeitas
4	<i>Tabu_simples_4_fc</i>	<i>Tabu_simples_4_nv</i>	<i>Tabu_simples_4_fcnv</i>
8	<i>Tabu_simples_8_fc</i>	<i>Tabu_simples_8_nv</i>	<i>Tabu_simples_8_fcnv</i>

B) *Tabu_LP*

O *Tabu_LP* acrescenta aos métodos *Tabu_simples_4_fcnv* e *Tabu_simples_8_fcnv* a memória de longo prazo baseada na *matriz_residência*. Lembramos que *matriz_residência* é atualizada a cada movimento efetuado e nunca é zerada.

A partir da *matriz_residência*, calcula-se as penalidades associadas aos movimentos de intensificação e diversificação, os quais são aplicados em sincronia com o ciclo de duração tabu.

Após a busca ter percorrido 2 ciclos sem atualizar a incumbente, os movimentos de intensificação são aplicados no 5º bloco do próximo ciclo e os movimentos de diversificação são aplicados no 9º bloco também deste ciclo.

Depois da realização desses movimentos, a busca continua por mais um ciclo na sua forma tradicional. Se após este ciclo a incumbente ainda não foi atualizada, aplica-se novamente os movimentos de intensificação (5º bloco) e diversificação (9º bloco). Este processo é repetido até uma solução melhor que a incumbente ser encontrada.

Caso a incumbente seja atualizada, a busca continua por mais dois ciclos tradicionais antes de aplicar novamente movimentos de intensificação ou diversificação.

Foi fundamental sincronizar o momento de realização dos movimentos de intensificação e diversificação com o bloco do ciclo. Durante a intensificação, arestas que estiveram presentes por muito tempo – possivelmente “boas” arestas – são inseridas na solução. É interessante que as arestas inseridas permaneçam por várias iterações na expectativa de que a combinação com outros movimentos gere novas soluções boas. Durante a diversificação, arestas pouco presentes são inseridas. Para escapar de um mínimo local, estas arestas também devem permanecer na solução por um número suficiente de iterações. Portanto, a duração tabu das arestas, nos dois casos, deveria ser mais elevada. Dentro do ciclo, esta situação ocorre nos blocos 5, 9 e 10.

A opção por aplicar a intensificação antes da diversificação também tem sua lógica. Após a busca explorar uma região por várias iterações, é desejável aplicar movimentos de intensificação para analisar a região com mais detalhe. Se a intensificação não gerar frutos, então a diversificação é aplicada para que a busca mude para novas regiões.

O método *Tabu_LP* também foi testado em seis versões, como apresentado na Tabela 2.16.

Tabela 2.16. Versões da busca *Tabu_LP*

Número de iterações por bloco do ciclo (vide Tabelas 2.12 e 2.13)	Uso da memória de longo prazo		
	Somente movimentos de Intensificação	Somente movimentos de Diversificação	Ambos os movimentos
4	<i>Tabu_LP_4_fcnv_I</i>	<i>Tabu_LP_4_fcnv_D</i>	<i>Tabu_LP_4_fcnv_ID</i>
8	<i>Tabu_LP_8_fcnv_I</i>	<i>Tabu_LP_8_fcnv_D</i>	<i>Tabu_LP_8_fcnv_ID</i>

C) *Tabu_reinício*

O método *Tabu_reinício* utiliza a memória de curto prazo e, durante a busca, coleta soluções para criar um conjunto de soluções de elite.

Após um ciclo completado sem atualizar a incumbente, o método recomeça a busca no 9º bloco do próximo ciclo. O reinício foi definido no 9º bloco porque é desejado que a mudança gerada pela maior duração tabu nos blocos 9 e 10 do ciclo anterior possa surtir efeito.

O método também é baseado em *Tabu_simples_4_fcnv* e *Tabu_simples_8_fcnv*, modificados em seis versões, como na Tabela 2.17.

Tabela 2.17. Versões da busca *Tabu_reinício*

Número de iterações por bloco do ciclo (vide Tabelas 2.12 e 2.13)	Tamanho máximo do conjunto de elite (número de soluções)		
	5	10	15
4	<i>Tabu_reinício_4_fcnv_CE5</i>	<i>Tabu_reinício_4_fcnv_CE10</i>	<i>Tabu_reinício_4_fcnv_CE15</i>
8	<i>Tabu_reinício_8_fcnv_CE5</i>	<i>Tabu_reinício_8_fcnv_CE10</i>	<i>Tabu_reinício_8_fcnv_CE15</i>

D) *Tabu_misto*

Finalmente, o *Tabu_misto* contém todos os componentes da busca tabu implementada: memória de curto e longo prazo, conjunto de elite e reinício.

A coordenação dos componentes é feita da seguinte forma:

1. Num primeiro momento, a busca possui o comportamento do *Tabu_LP*: após dois ciclos sem melhorar a incumbente, são aplicados, no ciclo seguinte, movimentos de intensificação no 5º bloco e movimentos de diversificação no 9º bloco;
2. Se a incumbente não for atualizada no ciclo subsequente, o reinício é executado no 9º bloco;
3. Aguarda-se um ciclo para que a solução do reinício seja explorada;
4. Caso a incumbente ainda não seja melhorada, são aplicados os movimentos de intensificação no 5º bloco e movimentos de diversificação no 9º bloco. O método retorna ao passo 2;
5. Se houve melhora da incumbente, o método retorna ao passo 1.

Este método também foi testado em seis versões:

Tabela 2.18. Versões da busca *Tabu_misto*

Número de iterações por bloco do ciclo (vide Tabelas 2.12 e 2.13)	Tamanho máximo do conjunto de elite (número de soluções)		
	5	10	15
4	<i>Tabu_misto_4_fcnv_CE5</i>	<i>Tabu_misto_4_fcnv_CE10</i>	<i>Tabu_misto_4_fcnv_CE15</i>
8	<i>Tabu_misto_8_fcnv_CE5</i>	<i>Tabu_misto_8_fcnv_CE10</i>	<i>Tabu_misto_8_fcnv_CE15</i>

2.5.9. Resultados dos experimentos computacionais

Foram realizados testes computacionais para todos estes métodos – 24 no total – nos 56 problemas teste propostos por Solomon (1987). A solução de partida utilizada é sempre aquela obtida pela heurística construtiva RR-B_2_4_0.5_0.5.

As quatro tabelas seguintes apresentam os resultados do método *Tabu_simples*.

Tabela 2.19. Porcentagem de veículos adicionais – *Tabu_simples*

Metaheurística <i>Tabu_simples</i>	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	fica	0.0%	0.0%	15.4%	16.7%	15.8%	15.6%	11.1%	3
	não volta	0.0%	0.0%	15.4%	16.7%	15.8%	15.6%	11.1%	3
	fica + não volta	0.0%	0.0%	15.4%	16.7%	14.6%	15.6%	10.9%	1
Tamanho do bloco: 8 iterações	fica	0.0%	0.0%	15.4%	16.7%	15.8%	15.6%	11.1%	3
	não volta	0.0%	0.0%	15.4%	16.7%	17.0%	15.6%	11.2%	6
	fica + não volta	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	2

Tabela 2.20. Desvio % médio na distância – *Tabu_simples*

Metaheurística <i>Tabu_simples</i>	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	fica	2.6%	0.9%	0.5%	3.8%	1.5%	4.0%	2.2%	6
	não volta	0.7%	0.9%	0.7%	3.8%	1.3%	3.2%	1.8%	4
	fica + não volta	2.5%	0.8%	0.7%	3.7%	0.5%	2.0%	1.7%	3
Tamanho do bloco: 8 iterações	Fica	0.4%	0.8%	0.5%	4.5%	0.5%	2.4%	1.6%	1
	não volta	2.2%	0.9%	0.6%	4.1%	0.8%	3.4%	2.0%	5
	fica + não volta	0.3%	0.6%	0.4%	4.7%	1.6%	2.3%	1.7%	2

Tabela 2.21. Tempo médio de execução (segundos) – *Tabu_simples*

Metaheurística <i>Tabu_simples</i>	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	Fica	45.9	300.8	34.2	477.0	29.5	354.3	206.2	1
	não volta	53.8	341.1	41.0	581.0	34.4	422.6	245.6	4
	fica + não volta	45.4	316.0	34.0	493.0	29.6	355.0	211.5	3
Tamanho do bloco: 8 iterações	Fica	54.0	350.6	40.6	629.3	35.1	418.4	255.9	5
	não volta	45.8	289.6	34.6	500.6	29.3	348.2	208.4	2
	fica + não volta	53.8	465.8	43.1	641.4	29.4	356.1	265.5	6

Tabela 2.22. Tempo médio até a melhor solução (segundos) – *Tabu_simples*

Metaheurística <i>Tabu_simples</i>	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	fica	15.5	54.8	24.8	211.7	13.7	126.3	77.2	1
	não volta	16.9	56.8	28.4	382.9	23.1	198.0	123.7	5
	Fica + não volta	13.4	68.2	26.2	323.4	24.5	148.1	105.7	3
Tamanho do bloco: 8 iterações	fica	17.3	64.3	26.7	354.3	25.2	202.7	119.8	4
	não volta	13.8	40.3	22.8	281.1	20.0	226.8	103.3	2
	Fica + não volta	16.2	105.2	33.4	401.7	22.2	149.0	128.1	6

O tamanho do bloco não influencia na qualidade do objetivo de minimizar o número de veículos. Para o objetivo de minimização de distância, o tamanho do bloco de 8 iterações gerou os melhores resultados.

Dentre os três tipos de atributos tabu (*fica*, *não volta* e *fica+não volta*), *fica+não volta* gerou os melhores resultados para número de veículos e, em geral, distância. A combinação de manter arestas promissoras na solução e de evitar ciclagem através do impedimento da inserção das arestas recém removidas foi uma estratégia bem sucedida.

Comparando as versões *Tabu_simples_4_fcnv* e *Tabu_simples_8_fcnv* com os resultados da busca local Gra (RR) modalidade *first*, a redução do número de veículos foi pouco significativa, de 11,4% para 10,9%. O desvio da distância foi mais expressivo, diminuiu de 7,0% para 1,7%. Vale notar que os piores resultados para a redução do número de veículos são os dos problemas teste com distribuição aleatória ou mista de clientes (R ou RC).

O tempo computacional ficou em torno de 4 minutos, aproximadamente seis vezes o tempo da busca local. Todavia, na média a melhor solução é encontrada em até 2 minutos. Ainda, da mesma forma da busca local, o tempo computacional é sempre superior nos problemas teste do tipo 2.

A seguir temos o resultado do método *Tabu_LP*.

Tabela 2.23. Porcentagem de veículos adicionais – *Tabu_LP*

Metaheurística Tabu_LP	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	I	0.0%	0.0%	15.4%	16.7%	15.8%	15.6%	11.1%	6
	D	0.0%	0.0%	14.8%	16.7%	15.8%	15.6%	10.9%	5
	I + D	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	1
Tamanho do bloco: 8 iterações	I	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	1
	D	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	1
	I + D	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	1

Tabela 2.24. Desvio % médio na distância– *Tabu_LP*

Metaheurística Tabu_LP	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	I	0.3%	0.9%	0.5%	4.7%	1.7%	1.0%	1.6%	5
	D	2.5%	1.0%	0.1%	3.6%	1.1%	1.1%	1.6%	4
	I + D	2.5%	0.9%	0.0%	3.6%	0.2%	1.2%	1.4%	1
Tamanho do bloco: 8 iterações	I	0.3%	0.6%	0.4%	4.8%	1.6%	2.3%	1.7%	6
	D	0.3%	0.6%	0.0%	4.8%	1.2%	1.4%	1.5%	2
	I + D	0.3%	0.6%	0.2%	4.9%	1.2%	1.3%	1.5%	3

Tabela 2.25. Tempo médio de execução (segundos) – *Tabu_LP*

Metaheurística Tabu_LP	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	I	45.5	326.3	34.7	493.1	41.1	360.7	215.6	2
	D	53.6	371.2	40.6	592.9	35.0	418.0	251.5	5
	I + D	45.5	314.9	34.1	498.8	29.6	356.0	212.7	1
Tamanho do bloco: 8 iterações	I	45.7	309.6	34.3	517.6	29.4	359.7	216.2	3
	D	45.5	307.8	34.4	518.6	29.5	364.6	216.8	4
	I + D	53.8	354.8	40.8	609.7	35.0	420.3	252.9	6

Tabela 2.26. Tempo médio até a melhor solução (segundos) – *Tabu_LP*

Metaheurística Tabu_LP	Teste	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	I	15.4	38.7	26.9	235.7	23.0	196.9	91.5	1
	D	14.7	66.2	30.3	383.9	24.7	228.1	129.8	5
	I + D	13.6	54.0	30.8	362.7	23.4	225.5	123.3	4
Tamanho do bloco: 8 iterações	I	13.7	66.1	26.2	303.7	22.4	149.1	101.4	2
	D	13.7	66.5	27.8	318.7	22.4	259.8	120.6	3
	I + D	16.2	76.8	33.6	339.3	26.5	307.6	135.2	6

O uso da memória de longo prazo trouxe benefícios para o objetivo de minimizar distância. A melhor alternativa foi combinar intensificação e diversificação no mesmo método.

Em segundo lugar, a opção é aplicar somente movimentos de diversificação, e por último, somente movimentos de intensificação.

Com relação à versão *Tabu_simples*, a memória de longo prazo aprimorou o desvio da distância de 1,7% para 1,4%.

É interessante notar que o acréscimo da memória de longo prazo não causa aumento significativo nos tempos computacionais.

Abaixo estão os resultados do método *Tabu_reinício*.

Tabela 2.27. Porcentagem de veículos adicionais – *Tabu_reinício*

Metaheurística <i>Tabu_reinício</i>	Tamanho do conjunto	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	5	0.0%	0.0%	15.4%	16.7%	15.8%	15.6%	11.1%	5
	10	0.0%	0.0%	15.4%	16.7%	15.8%	15.6%	11.1%	5
	15	0.0%	0.0%	14.7%	16.7%	15.8%	15.6%	10.9%	4
Tamanho do bloco: 8 iterações	5	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	2
	10	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	2
	15	0.0%	0.0%	14.7%	16.7%	14.7%	15.6%	10.8%	1

Tabela 2.28. Desvio % médio na distância– *Tabu_reinício*

Metaheurística <i>Tabu_reinício</i>	Tamanho do conjunto	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	5	1.1%	0.8%	0.8%	5.0%	1.1%	1.4%	1.8%	2
	10	1.5%	0.8%	0.6%	4.7%	1.9%	2.1%	2.0%	3
	15	1.9%	0.8%	0.4%	3.9%	1.2%	1.7%	1.7%	1
Tamanho do bloco: 8 iterações	5	1.9%	0.6%	0.2%	5.3%	1.9%	2.1%	2.0%	4
	10	2.4%	0.6%	0.2%	5.3%	2.0%	2.3%	2.2%	6
	15	2.0%	0.6%	0.1%	5.4%	1.9%	2.2%	2.1%	5

Tabela 2.29. Tempo médio de execução (segundos) – *Tabu_reinício*

Metaheurística <i>Tabu_reinício</i>	Tamanho do conjunto	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	5	52.2	344.0	40.1	534.8	34.8	412.2	235.1	5
	10	52.6	344.2	40.3	517.2	34.7	379.9	227.1	4
	15	45.0	294.4	34.6	442.7	30.1	342.1	196.9	1
Tamanho do bloco: 8 iterações	5	45.3	299.6	34.3	493.3	29.4	348.3	208.3	2
	10	45.2	305.3	35.3	520.9	29.8	364.6	217.1	3
	15	54.4	350.0	52.8	608.4	35.8	395.6	251.2	6

Tabela 2.30. Tempo médio até a melhor solução (segundos) – *Tabu_reinício*

Metaheurística <i>Tabu_reinício</i>	Tamanho do conjunto	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	5	15.7	69.1	21.8	266.5	24.8	260.3	110.1	6
	10	19.3	45.6	28.2	308.8	19.0	167.4	102.9	5
	15	15.4	42.9	23.1	237.3	19.1	160.4	85.8	1
Tamanho do bloco: 8 iterações	5	12.2	66.0	23.9	238.2	22.3	214.4	97.1	3
	10	8.7	66.3	26.4	228.4	21.4	190.7	91.7	2
	15	15.3	76.9	38.4	232.0	27.9	206.2	100.7	4

No *Tabu_reinício*, blocos de 8 iterações geraram resultados melhores para o objetivo de redução do número de veículos. No entanto, o resultado em distância foi prejudicado – sendo inclusive pior do que os resultados do *Tabu_simples*. Aparentemente, é mais interessante aplicar um método que pesquise mais iterações numa região, do que aplicar outro que explore várias regiões por poucas iterações.

Percebemos também que o tamanho máximo do conjunto de elite teve pouca influência na qualidade das soluções.

Por último, são apresentados os resultados do método *Tabu_misto*.

Tabela 2.31. Porcentagem de veículos adicionais – *Tabu_misto*

Metaheurística <i>Tabu_misto</i>	Tamanho do conjunto	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	5	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	2
	10	0.0%	0.0%	14.2%	16.7%	14.7%	15.6%	10.7%	1
	15	0.0%	0.0%	15.4%	16.7%	15.8%	15.6%	11.1%	6
Tamanho do bloco: 8 iterações	5	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	2
	10	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	2
	15	0.0%	0.0%	15.4%	16.7%	14.7%	15.6%	10.9%	2

Tabela 2.32. Desvio % médio na distância – *Tabu_misto*

Metaheurística <i>Tabu_misto</i>	Tamanho do conjunto	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	5	2.5%	0.8%	0.1%	4.2%	0.3%	1.2%	1.6%	5
	10	2.5%	0.8%	0.0%	4.1%	0.3%	1.4%	1.6%	6
	15	2.1%	0.9%	0.0%	4.1%	0.5%	1.3%	1.5%	4
Tamanho do bloco: 8 iterações	5	0.3%	0.6%	0.2%	4.9%	1.2%	1.3%	1.5%	1
	10	0.3%	0.6%	0.2%	4.9%	1.2%	1.3%	1.5%	2
	15	0.3%	0.6%	0.2%	4.9%	1.2%	1.3%	1.5%	2

Tabela 2.33. Tempo médio de execução (segundos) – *Tabu_misto*

Metaheurística <i>Tabu_misto</i>	Tamanho do conjunto	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	5	53.5	361.5	40.5	564.5	35.1	413.0	243.8	4
	10	50.2	300.1	34.5	493.8	30.5	336.0	207.7	1
	15	53.9	355.2	41.4	561.7	36.1	393.1	239.9	3
Tamanho do bloco: 8 iterações	5	53.8	350.5	40.8	606.5	35.0	421.8	251.8	5
	10	45.5	303.1	34.6	518.7	29.6	359.0	215.4	2
	15	54.6	355.6	41.9	669.9	47.6	502.8	278.8	6

Tabela 2.34. Tempo médio até a melhor solução (segundos) – *Tabu_misto*

Metaheurística <i>Tabu_misto</i>	Tamanho do conjunto	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
Tamanho do bloco: 4 iterações	5	16.1	101.8	35.2	370.8	23.7	311.2	145.3	5
	10	14.8	59.9	31.2	317.8	23.7	205.4	112.8	1
	15	19.2	75.4	37.7	359.4	29.9	252.8	132.9	3
Tamanho do bloco: 8 iterações	5	16.2	76.7	32.3	370.0	26.5	309.1	141.1	4
	10	13.8	66.2	27.3	290.1	22.4	262.5	115.2	2
	15	16.5	77.2	33.0	371.4	36.5	369.2	151.7	6

Este foi o método que gerou os melhores resultados para o objetivo de reduzir o número de veículos (10,7%). Também foi o método que gerou bons resultados para o objetivo de minimizar distância em todas as suas versões com mínima variância entre elas (pior caso da média do desvio foi de 1,6%). A versão de melhor resultado foi *Tabu_misto_4_fcnv_CE10*.

Atribuímos o sucesso desta versão ao fato de que cada componente da busca tabu deve possuir competência em certos tipos de problemas. Como o *Tabu_misto* contém todos os componentes, a falha de um componente pode ser compensada pelo sucesso de outro. Isto pode ser percebido de maneira indireta através dos tempos médios até a busca encontrar a melhor solução (Tabela 2.34). O *Tabu_misto* é a versão que possui os maiores tempos médios para encontrar a melhor solução, indicando que foi necessário aplicar componentes dos últimos ciclos para melhorar as soluções de alguns problemas.

Da mesma forma das outras versões, o tempo computacional para os problemas do tipo 2 pode chegar a até dez vezes os tempos computacionais dos problemas do tipo 1.

Analisamos a seguir a evolução da qualidade dos resultados com a adição de componentes da busca tabu até compor o método *Tabu_misto_4_fcnv_CE10*: *Tabu_simples_4_fcnv*, *Tabu_LP_4_fcnv_I*, *Tabu_LP_4_fcnv_D*, *Tabu_LP_4_fcnv_ID*, *Tabu_misto_4_fcnv_CE10*.

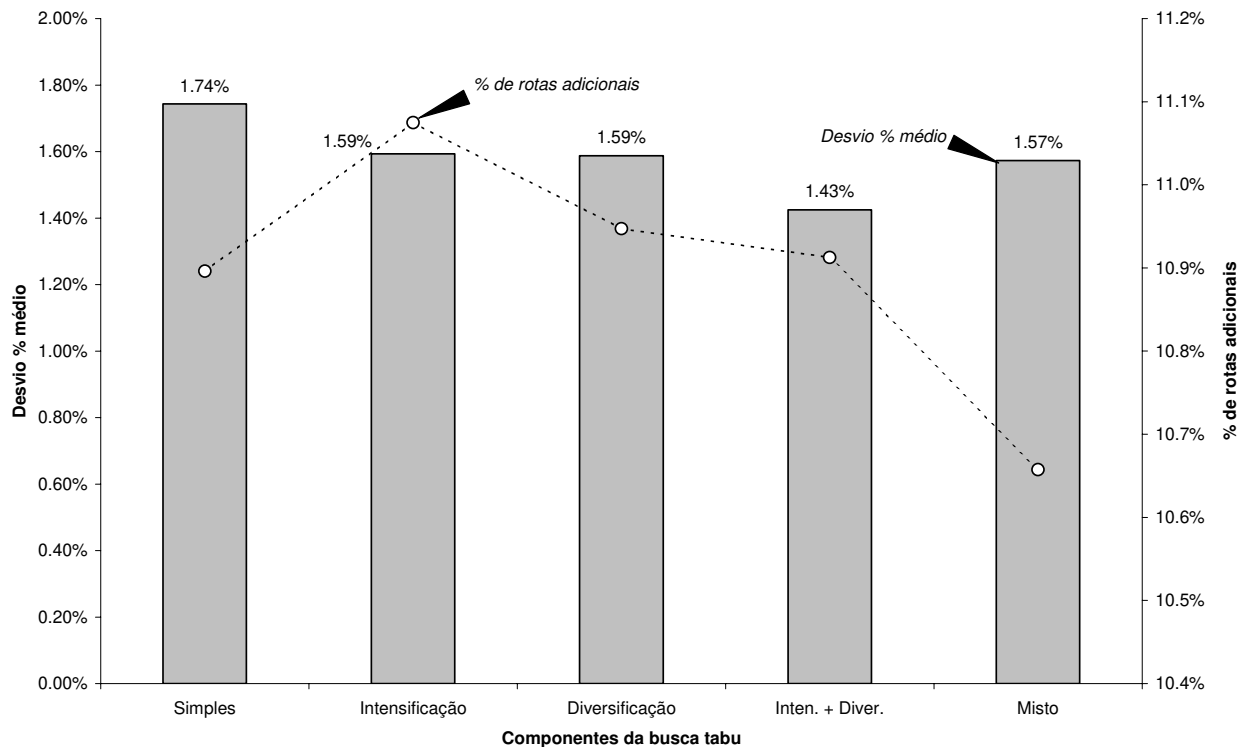


Figura 2.12. Evolução da qualidade dos resultados com a adição de componentes da busca tabu

A linha do gráfico apresenta o percentual de rotas adicionais enquanto as colunas apresentam o desvio percentual médio da distância. É observada uma queda do percentual de rotas adicionais com o acréscimo de componentes da busca tabu. O desvio percentual médio também decresce, exceto na versão final.

Apesar de existir melhoria na qualidade com o acréscimo de componentes, esta não ocorre em grandes saltos. Verificamos que, após um certo estágio de qualidade das soluções, é cada vez mais difícil se obter ganhos. E, ao comparar nossos métodos com um dos melhores métodos da literatura percebemos que ainda há muito a se melhorar. A Tabela 2.35. compara o *Tabu_misto_4_fcnv_CE10* com os resultados da busca tabu de Cordeau *et al.* (2001).

Tabela 2.35. Comparação entre o *Tabu_misto_4_fcnv_CE10* e o método de Cordeau *et al.* (2001)

Característica	Método	Tipo de problema							Class.
		C1	C2	R1	R2	RC1	RC2	Todos	
% de veículos adicionais	<i>Tabu_misto_4_fcnv_CE10</i>	0.0%	0.0%	14.2%	16.7%	14.7%	15.6%	10.7%	2
	Cordeau <i>et al.</i> (2001)	0.0%	0.0%	1.9%	0.0%	0.0%	0.0%	0.4%	1
Desvio % médio	<i>Tabu_misto_4_fcnv_CE10</i>	2.5%	0.8%	0.0%	4.1%	0.3%	1.4%	1.6%	2
	Cordeau <i>et al.</i> (2001)	0.2%	0.4%	0.0%	2.0%	0.4%	1.3%	0.7%	1

O método de Cordeau *et al.* (2001) possui desvio de menos de 1% tanto para o percentual de veículos adicionais quanto para o percentual de distância. Alguns motivos para a excelente qualidade do método de Cordeau *et al.* (2001) são:

- O método parte de uma solução inicial – não necessariamente factível – que contém m veículos, em que m é o menor número de veículos que outros autores publicaram na literatura. Se após um certo número de iterações a busca não encontrar nenhuma solução factível, o número de veículos é aumentado;
- É permitido que a busca explore soluções infactíveis, tanto para as restrições de capacidade quanto para as de janelas de tempo. Desta maneira, soluções infactíveis com menos rotas podem ser analisadas e eventualmente viabilizadas;
- Os esforços computacionais são concentrados somente na aplicação de movimentos muito influentes (inserções inter-rota), ao invés de movimentos intra-rota;
- Os resultados publicados provêm das melhores soluções encontradas a partir de várias rodadas com diferentes soluções de partida.

Quatro novos testes foram realizados para investigar duas questões:

1. Qual seria o comportamento do método *Tabu_misto_4_fcnv_CE10*, principalmente nos problemas teste tipo 1, se a solução de partida fosse gerada a partir da heurística construtiva Solomon (1987)?
2. Qual seria o comportamento do método *Tabu_misto_4_fcnv_CE10* ao restringir a vizinhança da busca para somente movimentos inter-rota?

Desta maneira, foram testadas as seguintes variantes do método *Tabu_misto_4_fcnv_CE10*:

- *Tabu_misto_4_fcnv_CE10 (inter)*: *Tabu_misto_4_fcnv_CE10* com vizinhança restrita a movimentos inter-rota (*crossover*, inserção e troca);
- *Tabu_misto_4_fcnv_CE10 (Solomon)*: *Tabu_misto_4_fcnv_CE10* com solução de partida obtida através de Solomon (1987);
- *Tabu_misto_4_fcnv_CE10 (Solomon)*: *Tabu_misto_4_fcnv_CE10* com solução de partida obtida através de Solomon (1987) e vizinhança restrita a movimentos inter-rota.

A seguir estão os resultados dos novos testes:

Tabela 2.36. Porcentagem de veículos adicionais – Variantes por solução de partida e vizinhança

Metaheurística	Tipo de problema							Class.
	C1	C2	R1	R2	RC1	RC2	Todos	
Tabu_misto_4_fcnv_CE10	0.0%	0.0%	14.2%	16.7%	14.7%	15.6%	10.7%	3
Tabu_misto_4_fcnv_CE10 (inter)	0.0%	0.0%	13.2%	16.7%	15.8%	15.6%	10.6%	2
Tabu_misto_4_fcnv_CE10 (Solomon)	0.0%	0.0%	13.0%	40.2%	11.0%	30.2%	16.6%	4
Tabu_misto_4_fcnv_CE10 (Solomon-inter)	0.0%	12.5%	13.0%	40.2%	11.0%	30.2%	18.3%	5
Cordeau et al. (2001)	0.0%	0.0%	1.9%	0.0%	0.0%	0.0%	0.4%	1

Restringir a vizinhança a somente movimentos inter-rota não produziu uma significativa redução no percentual de veículos adicionais. Quando a solução de partida provém de Solomon (1987), a restrição da vizinhança piora os resultados dos problemas teste C2. Aparentemente, a política de sempre utilizar movimentos inter-rota quando ocorre a degradação da função objetivo, política já adotada em *Tabu_misto_4_fcnv_CE10*, é suficiente.

Usar Solomon (1987) como heurística de partida reduziu o percentual de veículos adicionais para os problemas teste do tipo 1, principalmente para RC1: redução de 14,7% para 11,0%.

Tabela 2.37. Desvio % médio na distância – Variantes por solução de partida e vizinhança

Metaheurística	Tipo de problema							Class.
	C1	C2	R1	R2	RC1	RC2	Todos	
Tabu_misto_4_fcnv_CE10	2.5%	0.8%	0.0%	4.1%	0.3%	1.4%	1.6%	3
Tabu_misto_4_fcnv_CE10 (inter)	2.2%	3.4%	2.0%	6.7%	2.8%	7.7%	4.1%	4
Tabu_misto_4_fcnv_CE10 (Solomon)	0.5%	0.6%	0.3%	0.0%	0.4%	-1.0%	0.1%	1
Tabu_misto_4_fcnv_CE10 (Solomon-inter)	2.8%	8.3%	2.3%	6.4%	2.0%	5.0%	4.4%	5
Cordeau <i>et al.</i> (2001)	0.2%	0.4%	0.0%	2.0%	0.4%	1.3%	0.7%	2

Para o objetivo de minimizar a distância, restringir a vizinhança não é adequado. O desvio percentual sobe de 1,6% para aproximadamente 4,0%.

A variante *Tabu_misto_4_fcnv_CE10 (Solomon)* possui desvio percentual médio bastante pequeno, sendo que este chega até a ser negativo nos problemas teste do tipo RC2 (balanço entre redução do número de veículos e minimização de distância). Para os problemas teste do tipo 1, conclui-se que *Tabu_misto_4_fcnv_CE10 (Solomon)* é a melhor variante.

Tabela 2.38. Tempo médio de execução (segundos) – Variantes por solução de partida e vizinhança

Metaheurística	Tipo de problema							Class.
	C1	C2	R1	R2	RC1	RC2	Todos	
Tabu_misto_4_fcnv_CE10	50.2	300.1	34.5	493.8	30.5	336.0	207.7	4
Tabu_misto_4_fcnv_CE10 (inter)	43.7	268.7	33.7	470.0	28.9	352.5	199.4	2
Tabu_misto_4_fcnv_CE10 (Solomon)	43.6	276.9	30.3	419.4	26.3	317.5	184.6	1
Tabu_misto_4_fcnv_CE10 (Solomon-inter)	49.1	276.5	34.8	465.7	29.5	346.2	200.0	3

Tabela 2.39. Tempo médio até a melhor solução (segundos) – Variantes por solução de partida e vizinhança

Metaheurística	Tipo de problema							Class.
	C1	C2	R1	R2	RC1	RC2	Todos	
Tabu_misto_4_fcnv_CE10	14.8	59.9	31.2	317.8	23.7	205.4	112.8	2
Tabu_misto_4_fcnv_CE10 (inter)	23.2	64.1	28.9	296.3	20.9	166.9	104.1	1
Tabu_misto_4_fcnv_CE10 (Solomon)	8.0	59.3	20.4	325.8	20.5	225.2	113.2	3
Tabu_misto_4_fcnv_CE10 (Solomon-inter)	21.3	114.1	21.6	340.7	23.6	196.9	122.7	4

Os tempos médios de execução são equivalentes entre as variantes. As variantes que utilizam Solomon (1987) como solução de partida possuem, nos problemas teste do tipo 2, tempos superiores até a melhor solução.

Finalmente, o método *Tabu_misto_4_fcnv_CE10* foi comparado com os propostos por Bräysy (2003) e Homberger (1999) para todos os 116 problemas testes. Para os problemas de 100 clientes, foram comparados também os resultados de Bent e Hentenryck (2004^a).

Tabela 2.40. Comparação entre o *Tabu_misto_4_fcnv_CE10* e *benchmarks* da literatura – número médio de veículos para os problemas de 100 clientes.

Metaheurística	Tipo de problema							Class.
	C1	C2	R1	R2	RC1	RC2	Todos	
Tabu_misto	10.0	3.0	13.5	3.1	13.3	3.8	446.0	3
Homberger (1999)	10.0	3.0	11.9	2.7	11.6	3.3	406.0	2
Bräysy (2003)	10.0	3.0	11.9	2.7	11.5	3.3	405.0	1
Bent e Hentenryck (2004^a)	10.0	3.0	11.9	2.7	11.5	3.3	405.0	1

Tabela 2.41. Comparação entre o *Tabu_misto_4_fcnv_CE10* e *benchmarks* da literatura – distância média para os problemas de 100 clientes.

Metaheurística	Tipo de problema						Class.
	C1	C2	R1	R2	RC1	RC2	
Tabu_misto	846.9	592.6	1,211.2	987.9	1,383.9	1,126.9	3
Homberger (1999)	828.4	589.9	1,228.1	970.0	1,392.6	1,144.4	4
Bräysy (2003)	828.4	589.9	1,222.1	975.1	1,389.6	1,128.4	2
Bent e Hentenryck (2004^a)	828.4	589.9	1,213.3	966.4	1,334.2	1,141.2	1

Tabela 2.42. Comparação entre o *Tabu_misto_4_fcnv_CE10*, Bräysy (2003) e Homberger (2001) – número médio de veículos para os problemas de 200 clientes.

Metaheurística	Tipo de problema						Class.
	C1	C2	R1	R2	RC1	RC2	
Tabu_misto	19.6	6.4	18.9	4.2	19.3	5.0	3
Homberger (1999)	18.0	6.0	18.2	4.0	18.0	4.3	1
Bräysy (2003)	18.9	6.0	18.1	4.0	18.0	4.4	2

Tabela 2.43. Comparação entre o *Tabu_misto_4_fcnv_CE10*, Bräysy (2003) e Homberger (2001) – distância média para os problemas de 200 clientes.

Metaheurística	Tipo de problema						Class.
	C1	C2	R1	R2	RC1	RC2	
Tabu_misto	2,746.6	1,858.9	3,791.9	3,084.2	3,349.2	2,525.8	1
Homberger (1999)	2,782.0	1,846.0	3,705.0	3,055.0	3,511.0	2,658.0	2
Bräysy (2003)	2,778.8	1,842.4	3,821.4	3,045.3	3,508.1	2,628.4	3

Tabela 2.44. Tempo médio de execução (segundos) – *Tabu_misto_4_fcnv_CE10*

Tamanho do problema (número de clientes)	Tipo de problema						Todos
	C1	C2	R1	R2	RC1	RC2	
100	50.2	300.1	34.5	493.8	30.5	336.0	207.7
200	465.7	3,450.8	494.1	12,247.0	502.3	11,929.4	4,848.2

Tabela 2.45. Tempo médio até a melhor solução (segundos) – *Tabu_misto_4_fcnv_CE10*

Tamanho do problema (número de clientes)	Tipo de problema						Todos
	C1	C2	R1	R2	RC1	RC2	
100	14.8	59.9	31.2	317.8	23.7	205.4	112.8
200	220.0	1,474.1	423.5	7,927.2	419.1	9,846.0	3,385.0

Tanto o método de Bräysy (2003) quanto o de Homberger (1999) obtiveram excelentes resultados na redução do número de veículos e minimização da distância. Através da análise dos resultados dos problemas de 100 clientes, o método de Bent e Hentenryck (2004^a) pode ser considerado como o melhor método da literatura no momento.

Para problemas de 200 clientes, o método *Tabu_misto_4_fcnv_CE10* teve o melhor desempenho na minimização de distância com no máximo 1,6 veículos acima do menor número dos outros métodos.

Para a mesma quantidade de iterações, o tempo computacional dos problemas de 200 clientes é no mínimo 10 vezes o tempo computacional dos problemas de 100 clientes. Para os problemas do tipo 2, o acréscimo foi ainda superior, alcançando até 35 vezes o tempo dos problemas de 100 clientes (tipo RC2). O maior tempo médio foi de 3,4 horas por problema de 200 clientes (tipo R2).

Analisando os trabalhos de Bent e Hentenryck (2004^a) e Bräysy (2003), infere-se que uma das principais competências dos métodos foi a divisão do projeto da heurística em fases, tratando independentemente os objetivos de redução do número de veículos e de minimização de distância.

2.6. Conclusões para o PRVJT

Podemos salientar as seguintes conclusões para o PRVJT:

- Existe um balanço entre os objetivos de redução do número de veículos e minimização da distância;
- A heurística construtiva de Solomon (1987) é adequada para problemas teste do tipo 1;
- A heurística construtiva RR foi eficiente no objetivo de redução do número de veículos. Os conceitos adotados por esta heurística poderiam, possivelmente, ser utilizados para criar procedimentos mais complexos focados neste objetivo;
- A busca local granular é eficiente somente na minimização da distância;
- Caso ambos os objetivos sejam tratados de forma única, é importante permitir que a busca tabu explore soluções ineficazes com menos veículos. Se os objetivos forem tratados separadamente, deve-se implementar procedimentos eficientes e dedicados para cada objetivo;
- Todos os componentes da busca tabu contribuem para a qualidade das soluções geradas – o *Tabu_misto* foi o método de melhor desempenho. Para os problemas teste do tipo 1, de todas as variantes de *Tabu_misto*, a versão que produziu os melhores resultados foi *Tabu_misto_4_fcnv_CE10 (Solomon)*;
- Para a busca tabu, as vizinhanças intra-rota, de uma forma geral, pouco colaboram para que o método atinja bons mínimos locais. Consequentemente, é fundamental incentivar que a busca execute movimentos inter-rotas, que são mais influentes. Isto foi alcançado ao forçar sempre a execução de movimentos inter-rota quando ocorre a degradação do valor da função objetivo;
- É mais difícil encontrar soluções com um menor número de veículos para os problemas teste com distribuição aleatória de clientes no espaço;
- A respeito do tempo computacional, podemos afirmar:
 - Quanto maior o número de clientes por rota, maior será o tempo computacional das heurísticas (construtiva, busca local e tabu);
 - A redução do grafo granular é uma forma competente de se diminuir o tempo computacional com baixa perda de qualidade;
 - A busca local na modalidade *first* é mais rápida do que a modalidade *best*, apresentando qualidade de solução equivalente;

- Quando se dobra o número de clientes do problema, o tempo computacional é aumentado de 10 a 35 vezes.

Capítulo 3. Problema de Roteamento Dinâmico de Veículos com Janelas de Tempo (PRDVJT)

3.1. Descrição do PRDVJT

O problema de roteamento dinâmico de veículos pode ser definido como aquele em que as informações necessárias para a determinação de rotas eficientes são reveladas dinamicamente ao tomador de decisão. Para esta dissertação, a informação desconhecida é a posição e a demanda de alguns clientes.

Pode-se visualizar o PRDVJT como a composição de uma série de PRVJTs estáticos de menor porte. Por exemplo, seja SP_0 um sub-problema com as informações conhecidas a priori, antes do primeiro veículo iniciar a primeira rota. Este sub-problema pode conter os pedidos recebidos no dia anterior que foram agendados para o dia seguinte.

Decisões são tomadas sobre o SP_0 , o qual é um PRVJT de menor porte, criando roteiros preliminares para os veículos. Ao início do turno, os veículos partem do depósito e começam a percorrer seus roteiros.

Com o decorrer do tempo, pedidos são recebidos gerando novos sub-problemas intermediários, representados por SP_i , que contém todas as informações dos sub-problemas anteriores ($SP_0, SP_1, SP_2, \dots, SP_{i-3}, SP_{i-2}$ e SP_{i-1}). No entanto, somente é necessário tomar decisões para a parcela de SP_i que ainda não foi percorrida pelos veículos. Portanto, o problema de otimização ainda é resolver PRVJTs, cujo tamanho, em geral, não aumenta demasiadamente ao longo do tempo.

Finalmente, após o recebimento do último pedido do dia, tem-se o SP_f com as informações completas.

Analisando esta decomposição, poder-se-ia imaginar que a melhor solução para o problema é dada pela resolução exata do SP_f (PRVJT). Todavia, tal situação inexiste, pois assume a utopia em que nenhum veículo haveria iniciado a sua rota no momento em que o SP_f é conhecido.

Para o PRDVJT a medida da função objetivo será a contribuição ao lucro da operação de transporte. Desta maneira, a resolução do SP_f fornece um limitante superior ao lucro real.

Uma possível aproximação para buscar esta solução ideal é utilizar métodos de previsão para o SP_f . Então, ao invés de otimizar cada SP_i isoladamente, o grande desafio reside em resolver os sub-problemas intermediários com uma previsão sobre como será o SP_f . Alguns autores, a ser visto no item a seguir, utilizam componentes estocásticos em seus métodos para obter estes benefícios. É interessante apontar também que a resolução exata da sequência de sub-problemas $SP_0, SP_1, SP_2, \dots, SP_{i-3}, SP_{i-2}$ e SP_{i-1}, SP_i não leva necessariamente à melhor solução.

3.2. Revisão dos métodos heurísticos da literatura

Um importante estudo da década de 80 sobre problemas de roteamento dinâmico é o trabalho de Psaraftis (1988), no qual são delineadas 12 distinções deste problema em relação ao estático. Destacamos a seguir algumas dessas diferenças:

- Baixo tempo computacional é necessário;
- A função objetivo pode ser distinta e/ou mais complexa;
- Informações futuras podem ser imprecisas ou desconhecidas;
- Eventos de curto prazo possuem maior prioridade;
- Mecanismos de atualização da informação são fundamentais.

O autor define os problemas estáticos como aqueles em que todos os dados do problema não se alteram tanto durante a execução do método de resolução quanto durante a efetivação das soluções (veículos visitando os clientes). Nos problemas dinâmicos, parte ou todos os dados do problema são alterados ou revelados ao tomador de decisão durante a efetivação das soluções.

De uma forma geral, a literatura aborda e classifica os problemas dinâmicos de duas maneiras: problemas de roteamento dinâmico de veículos e problemas estocásticos de roteamento de veículos. Os métodos da primeira abordagem, foco principal desta dissertação, utilizam somente informações determinísticas do problema, tais como posição e demanda exatas dos clientes conhecidos e novos. Os roteiros desta abordagem podem – e geralmente são – alterados e reotimizados durante a efetivação da solução. Os métodos da abordagem estocástica utilizam informações probabilísticas para determinar, a priori, o roteiro completo de todos os veículos. E na maioria das vezes, uma vez iniciada a efetivação da solução, os roteiros não se alteram.

Contudo, formulações através de programação estocástica com múltiplos estágios ou de um processo de decisão de Markov podem contemplar processos de reotimização de soluções.

Gendreau *et al.* (1996) fazem uma revisão da literatura sobre os problemas estocásticos do caixeiro viajante e de roteamento de veículos. Nestes problemas há pelo menos um componente probabilístico, que pode ser a distância (ou tempo de viagem), demanda e/ou ocorrência de clientes. São apresentados métodos e referências bibliográficas para as seguintes variações de problemas estocásticos:

- Problemas de roteamento com demanda estocástica: esta é a variação de problemas estocásticos mais estudada, na qual o número e posição dos clientes são conhecidos a priori, porém a demanda somente é revelada quando o veículo chega ao cliente;
- Problemas de roteamento com clientes estocásticos: nesta variação as demandas são determinísticas, mas cada cliente possui uma probabilidade de efetuar o pedido;
- Problemas de roteamento com clientes e demandas estocásticos: combinação das duas variações anteriores.

Uma conclusão apresentada é que os problemas de roteamento com clientes e demandas estocásticos são muito mais difíceis de serem tratados. Para sua resolução, dois métodos são selecionados como promissores: métodos exatos baseados em Laporte *et al.* (1994) e a aplicação da metaheurística busca tabu.

Métodos heurísticos são discutidos em Gendreau e Potvin (1997) para a resolução das seguintes classes de problemas dinâmicos: *dial-a-ride* (transporte de idosos ou deficientes, agendado pelo cliente), serviços de reparo, *courier* e correio expresso. Apesar dos diversos métodos elaborados para os problemas dinâmicos, os autores concluem que ainda existem tópicos a serem estudados, destacando-se a inclusão de previsão das demandas, consideração de outros tipos de incertezas tais como cancelamentos e variações no tráfego, análise dos benefícios da utilização de desvio ou mudança da rota planejada (*diversion*⁹), quantificação sobre o quanto se perde por desconhecer todas as informações a priori, e avaliação das vantagens de implementações paralelas para este tipo de problema.

⁹ Um maior detalhamento sobre desvio pode ser encontrado no Anexo D.

Metaheurísticas, tais como busca tabu, simulated annealing, algoritmos genéticos e *Greedy Randomized Adaptive Search Procedures (GRASP)*, têm sido aplicadas com sucesso em uma grande variedade de problemas de otimização combinatória. Para o PRDVJT, uma das primeiras metaheurísticas desenvolvidas é a de Gendreau *et al.* (1999). Este algoritmo é essencialmente uma implementação em paralelo do algoritmo sugerido por Taillard *et al.* (1997) – vide Anexo E para detalhes sobre o algoritmo de Taillard.

Ao tratar o problema dinâmico, Gendreau *et al.* (1999) adotaram as seguintes premissas:

- Os pedidos novos devem ser recebidos com um mínimo de antecedência pré-definida para serem atendidos no mesmo dia. Caso contrário, os pedidos são programados para o dia seguinte;
- As incertezas consideradas no problema dinâmico são a ocorrência e a demanda dos novos pedidos. Não existem cancelamentos de pedidos, congestionamentos ou quebra de veículos e todas as distâncias são conhecidas;
- A comunicação entre a central de despacho e o motorista é realizada no momento em que o veículo chega ao local do cliente. Somente nesta ocasião o motorista toma conhecimento do próximo destino a ser servido;
- Se há previsão de tempo de espera – o motorista chegaria antes do momento de início da janela de tempo do próximo destino – o veículo fica aguardando na posição atual até o momento em que o tempo de chegada no próximo destino seja idêntico ao início da janela de tempo. Isto permite que o motorista atenda eventual pedido que apareça no caminho até o próximo destino;
- Não é permitido o desvio de rota;
- O número de veículos disponíveis no depósito é igual ao menor número de veículos encontrados na literatura para a resolução do respectivo problema estático.

A busca tabu foi adaptada a partir destas premissas e está detalhada no Anexo F. Os autores compararam o desempenho da busca com o de outros cinco métodos:

- i. Método de inserção: o novo cliente é inserido na posição que minimize o custo adicional;

- ii. Método de inserção+: Após cada inserção do método anterior, a busca local troca-cross (troca de dois subconjuntos de até k clientes em sequência de duas rotas distintas) é aplicada;
- iii. Método reconstrução: As rotas planejadas são reconstruídas a cada novo pedido, utilizando uma adaptação da heurística de inserção I1 do Solomon (1987);
- iv. Método reconstrução+: Aplica a busca local troca-cross a cada reconstrução;
- v. Método da descida adaptativa: Heurística de busca local com múltiplos reinícios. A busca local troca-cross é aplicada a diferentes soluções iniciais obtidas pela memória adaptativa.

Os autores concluem que a busca tabu teve desempenho superior ao dos outros cinco métodos. Ainda, quando comparado o desempenho da busca tabu no problema dinâmico com o desempenho da busca no problema estático, a busca tabu do problema dinâmico obteve resultados, em geral, apenas 2,5% a 4,1% superiores ao custo (distância percorrida) do problema estático.

Mais recentemente, Zhu e Ong (2000) desenvolveram um sistema de roteamento de veículos reativo, contendo uma central de controle que administra os roteiros e quantos veículos devem partir do depósito. A central de controle recebe informações sobre o posicionamento dos veículos e as condições de tráfego, e, através de um otimizador baseado em busca local, modifica os roteiros quando alguma restrição de janela de tempo será violada. Os autores implementaram um simulador discreto para a execução dos testes.

A possibilidade de se modificar o percurso de um veículo para atender um novo cliente é analisada em Ichoua *et al.* (2000). Esta alteração de destino está sendo viabilizada em função dos avanços tecnológicos em GPS e comunicação em tempo real (telefones celulares, *trunking* e computadores de bordo). Estratégias de desvio foram também propostas em Regan *et al.* (1994, 1995) no contexto de transportadoras de cargas.

Larsen *et al.* (2002) desenvolvem indicadores para classificar os problemas dinâmicos segundo o nível de dinamismo, o qual é medido pelo percentual da quantidade de pedidos dinâmicos e pela disponibilidade de tempo para reagir aos mesmos. Os autores destacam que quanto maior o nível de dinamismo, maior o custo das soluções. Foi verificado também que a estratégia do vizinho mais próximo gerou, em média, os melhores resultados para todos os níveis.

Bianchi (2000) agrupa os trabalhos sobre problemas dinâmicos em duas classes: a) métodos baseados em políticas ou estratégias, e b) métodos heurísticos. A autora conclui que um método baseado em políticas que gera bons resultados é o do vizinho mais próximo – vide Anexo C para uma descrição de outras políticas. Este método aloca os primeiros k clientes aos k veículos disponíveis. Após o término do atendimento, cada veículo parte para o cliente mais próximo. Para a classe de métodos heurísticos, a autora destaca os excelentes resultados da busca tabu de Gendreau *et al.* (1999), principalmente para problemas com baixo nível de dinamismo, em que há maior disponibilidade de tempo computacional para o método aprimorar a solução.

Uma extensa revisão sobre os diferentes tipos de problemas de roteamento dinâmico pode ser encontrada em Ghiani *et al.* (2003). Os autores listam sete classes de problemas dinâmicos e apresentam alguns métodos heurísticos para a resolução. Estratégias para computação paralela – inclusive de uma busca tabu – são apresentadas em detalhes.

Hvattum *et al.* (2004) realizaram um estudo de caso numa empresa de distribuição líder da Noruega. O problema consiste em criar roteiros para veículos atenderem clientes conhecidos e novos – total de 130 clientes – respeitando as restrições de janelas de tempo. Para sua resolução, foi implementada uma heurística baseada em cenários, os quais contém roteiros para três tipos de clientes: (i) clientes conhecidos antes do primeiro veículo partir do depósito, (ii) novos clientes que apareceram enquanto os veículos estão na rua, (iii) clientes fictícios, sorteados segundo distribuições de probabilidade baseadas em dados históricos. Os clientes fictícios atuam como guia para que sejam gerados roteiros considerando as regiões de maior probabilidade de ocorrência de novos pedidos. Os autores concluíram que a inclusão dos clientes fictícios pode gerar soluções com distância 15,6% inferior.

Bent e Hentenryck (2004^b) desenvolveram um trabalho semelhante ao de Hvattum *et al.* (2004). O método armazena um conjunto de rotas factíveis e associa a cada uma delas um valor de função objetivo, que pode ser a distância percorrida ou a função consenso. A função consenso é uma medida do grau de comprometimento causado pela seleção de uma rota. A motivação é privilegiar rotas que, quando escolhidas, mantêm um grande número de opções de rotas factíveis para seleções futuras. São testados dois tipos de geração de conjunto de rotas, considerando somente clientes conhecidos e considerando clientes conhecidos e fictícios. A conclusão é que utilizar a função consenso e considerar clientes fictícios melhoram significativamente a qualidade das soluções, principalmente para problemas com elevado nível de dinamismo.

Um problema dinâmico de coleta e entrega com janelas de tempo é analisado por Mitrović-Minić *et al.* (2004^a). Neste problema, a chegada de um novo cliente determina dois pontos no espaço, um de coleta e outro de entrega. O veículo deve necessariamente passar por estes dois pontos, respeitando a ordem de precedência e os intervalos definidos pelas janelas de tempo. O método proposto baseia-se na separação do problema, e da função objetivo, em dois horizontes: de curto e de longo prazo. A meta para o problema de curto prazo é minimizar a distância, enquanto para o problema de longo prazo é maximizar a folga das rotas, porque rotas folgadas possuem maior flexibilidade para a inserção de novos pedidos.

Mitrović-Minić e Laporte (2004^b) desenvolvem novos métodos incluindo estratégias de espera para o problema anterior. O benefício da espera é atender a um custo menor os novos clientes que poderiam surgir em locais próximos à posição atual do veículo.

A base do método é composta pela inserção mais barata dos novos pedidos, seguida, opcionalmente, por uma busca tabu simplificada que tem como vizinhança a cadeia de ejeções de Glover (1991, 1996) – vide Anexo H. Sobre esta base é aplicada uma das quatro estratégias de espera, a qual define o momento em que um veículo que terminou o atendimento do cliente deve partir para o próximo destino:

1. Estratégia partir-primeiro: veículo deve partir no tempo mais cedo possível, ou seja, logo após a conclusão do serviço no cliente visitado;
2. Estratégia esperar-primeiro: veículo deve partir no tempo mais tarde possível, tempo em que a visita para todos os clientes subsequentes do roteiro ainda seja viável;
3. Espera dinâmica: composição das duas estratégias anteriores. O veículo segue a estratégia partir-primeiro para um conjunto de clientes de seu roteiro que são próximos entre si. Os clientes são próximos entre si se pertencem à mesma zona de serviço, a qual é demarcada considerando tanto a distância entre os clientes quanto o intervalo de tempo para o atendimento de todos os clientes da zona. Após o término do atendimento dos clientes da zona de serviço, o veículo segue a estratégia esperar-primeiro até a próxima zona de serviço;
4. Espera dinâmica avançada: semelhante à espera dinâmica. A única modificação é uma atenuação da estratégia esperar-primeiro, limitando a espera para momentos anteriores ao mais tarde possível.

As quatro estratégias foram testadas e a espera dinâmica avançada foi a que produziu os melhores resultados.

Larsen *et al.* (2004) também aplicaram, com sucesso, estratégias de espera e de reposicionamento de veículos para o problema dinâmico do caixeiro viajante com janelas de tempo.

Estratégias de espera foram aplicadas para o PRDVJT em Ichoua *et al.* (2005). Estas foram acrescentadas à busca tabu de Gendreau *et al.* (1999), forçando os veículos a ficarem parados antes de partirem para o próximo destino. Os autores salientam a importância da definição do tempo de espera, pois se for muito elevado, causará atrasos nos destinos planejados, e, se for muito baixo, não produzirá ganhos. A decisão de espera é baseada na probabilidade de ocorrência de um novo pedido na região ao redor veículo, no próximo intervalo de tempo. Para evitar que vários veículos fiquem concentrados numa única região, os autores segmentaram o espaço em territórios e definiram um limitante superior para o número de veículos de cada território. Um simulador foi implementado para representar os principais aspectos encontrados na prática, e os experimentos computacionais demonstram que a heurística de espera e a política posicionamento geram ganhos da ordem de 10% sobre a soma do tempo de viagem com o atraso.

Gutenschwager *et al.* (2004) analisaram uma interessante aplicação para o problema dinâmico de coleta e entrega de um armazém, o qual utiliza um sistema transportador de cargas sobre trilhos elétricos. O objetivo do sistema é montar e despachar paletes contendo os pedidos de cada cliente. Para tanto, dado um pedido, o sistema coleta paletes de produtos em estantes de seis corredores, transporta-os até a mesa de embalagem na qual o palete do pedido será montado, e depois os retorna até as estantes. Como em geral um pedido é composto por diferentes produtos, para montar um palete de pedido é preciso transportar diversos paletes de produtos até as mesas de embalagem. Três objetivos são tratados: (i) minimizar o tempo de término da montagem do último palete de pedido (*makespan*), (ii) minimizar a fila de requisições de paletes de produtos e (iii) minimizar o tempo ocioso dos operadores de embalagem. São realizados testes computacionais com os métodos inserção mais barata, busca local, *simulated annealing* e busca tabu reativa, que foi a mais eficiente.

Taniguchi e Shimamoto (2004) trataram o problema em que o tempo de viagem é dinâmico, desenvolvendo métodos para problemas de roteamento utilizando sistemas inteligentes

de transporte¹⁰. O sistema inteligente de transporte fornece informação em tempo real sobre as condições de tráfego, além de permitir a visualização, também em tempo real, da posição dos veículos. A partir de dados históricos sobre o tráfego, da informação em tempo real e de previsões de congestionamento (eventos especiais como jogos esportivos, manutenção das vias, previsão de chuvas, etc.), o sistema sugere e altera os roteiros dos veículos. Foram implementadas duas heurísticas, uma que utiliza informações em tempo real e outra que não as utiliza. A utilização da informação em tempo real gera ganhos da ordem de 3,7% do custo total em ienes.

Fleischmann *et al.* (2004) analisam a combinação de dois aspectos dinâmicos, a ocorrência de clientes e condições de tráfego, num caso prático de coleta e entrega com 50 veículos e 981 clientes, dos quais aproximadamente 50% são dinâmicos. São testados algoritmos e regras de designação de clientes a veículos e métodos baseados em inserção mais barata e em busca local (considerando diferentes avaliações de custo). O método que apresentou os melhores resultados foi o algoritmo de designação de clientes a veículos. Este algoritmo determina, a cada novo pedido, qual seria o custo de atrasar os pedidos e de alocá-los aos veículos. Calcula também, no mesmo momento, qual seria o custo de manter cada veículo parado, esperando nova ordem. Em seguida, o algoritmo de designação LAPJV de Jonker e Volgenant (1987) é aplicado utilizando a matriz de custos. A decisão tomada é a solução ótima fornecida pelo algoritmo.

Percebe-se que nos últimos dois anos, principalmente em 2004, houve um avanço significativo no estudo de problemas dinâmicos. Há diversas metodologias sendo testadas e seria prematuro afirmar quais são as mais promissoras. Para organizar este conhecimento e facilitar o desenvolvimento de nossa proposta, descrita no próximo item, resumimos a evolução da pesquisa sobre problemas dinâmicos:

¹⁰ ITS (Intelligent Transportation System).

Tabela 3.1. Evolução da pesquisa sobre problemas dinâmicos

Período	Temas	Fontes estudadas nesta dissertação
Década de 80 e anos anteriores	• Caracterização do problema; • Desenvolvimento dos primeiros métodos e aplicações.	• Psaraftis (1988)
1990 a 1997	• Classificação da literatura para problemas estocásticos e dinâmicos; • Uso de métodos baseados em estratégias; • Avanço dos métodos heurísticos; • Novas aplicações práticas; • Inclusão da possibilidade de desvio.	• Rega <i>et al.</i> (1994, 1995) • Gendreau <i>et al.</i> (1996) • Gendreau e Potvin (1997)
1998 a 2000	• Aplicação de metaheurísticas (busca tabu); • Uso de métodos paralelos; • Novos estudos sobre desvio; • Classificação dos métodos em heurísticos e baseados em estratégias; • Tratamento da distância – de forma isolada – como fator dinâmico (condições de tráfego).	• Gendreau <i>et al.</i> (1999) • Ichoua <i>et al.</i> (2000) • Bianchi (2000) • Zhu e Ong (2000)
2001 a 2003	• Avaliação do nível de dinamismo dos problemas; • Revisão dos problemas e dos principais métodos para resolvê-los; • Início do estudo de problemas dinâmicos de coleta e entrega.	• Larsen <i>et al.</i> (2002) • Ghiani <i>et al.</i> (2003)
2004	• Avaliação de aplicações práticas complexas; • Inclusão de componentes estocásticos em métodos heurísticos; • Extenso estudo de problemas dinâmicos de coleta e entrega; • Divisão do horizonte de otimização avaliando diferentes objetivos; • Aplicação de outras funções objetivos (e.g. função consenso); • Inclusão de estratégias de espera nos métodos heurísticos; • Monitoramento da concentração de veículos no espaço; • Uso de políticas de posicionamento de veículos; • Implementação de simuladores baseados em dados reais; • Uso de informações em tempo real das condições de tráfego (ITS); • Combinação dos fatores dinâmicos de tráfego e de ocorrência de pedidos num único problema.	• Gutenschwager <i>et al.</i> (2004) • Hvattum <i>et al.</i> (2004) • Bent e Hentenryck (2004^b) • Mitrović-Minić <i>et al.</i> (2004^a) • Mitrović e Laporte (2004^b) • Larsen <i>et al.</i> (2004) • Ichoua <i>et al.</i> (2005) • Taniguchi e Shimamoto (2004) • Fleishmann <i>et al.</i> (2004)

3.3. Métodos heurísticos para o PRDVJT

Nesta dissertação foram aplicadas quatro heurísticas para o PRDVJT.

- Vizinho mais próximo: como apontado por Bianchi (2000) e Larsen *et al.* (2002), a heurística do vizinho mais próximo é a que produz os melhores resultados dentre os métodos baseados em políticas;
- Inserção: Cada novo pedido é inserido na melhor posição factível;
- Inserção+: Baseada em Gendreau *et al.* (1999), o método de Inserção+ é composto por uma etapa de melhor inserção do novo pedido (avaliando inserções em todos os roteiros), seguida pela busca local granular;
- *Tabu_LP_4_fcnv_ID_dinâmico*: Aplicação do método *Tabu_LP_4_fcnv_ID* do PRVJT para o PRDVJT¹¹. Inicialmente, o novo pedido é inserido na melhor posição factível. Em seguida, a busca tabu é iniciada para aprimorar a solução.

O método *Tabu_LP_4_fcnv_ID* foi preferido ao método *Tabu_misto_4_fcnv_CE10* porque há pouca disponibilidade de tempo de execução para se criar um repositório de soluções de qualidade. Ainda, a factibilização das soluções criadas a partir da união de rotas do repositório é dificultada pela impossibilidade de se modificar o roteiro dos clientes visitados.

São testadas ainda a possibilidade de desvio, estratégias básicas de espera e estratégias de posicionamento de veículos.

A estratégia de espera considera duas situações (assuma que o veículo terminou o serviço no cliente i e está pronto para partir para o cliente j):

1. Se o veículo chegaria em j antes de e_j , o veículo aguarda em i até o tempo de chegada em j ser igual a e_j , como em Gendreau *et al.* (1999);
2. Se j é igual ao depósito e existe a possibilidade de se atender mais um cliente (há folga na ocupação do veículo e o tempo de viagem é menor do que l_{n+1}), o veículo aguarda em i .

Quando ambas as políticas de desvio e espera são adotadas, ocorre a situação em que o veículo está em movimento em direção ao cliente j e este é trocado por outro cliente k . Se nesta

¹¹ Para ser aplicado ao contexto dinâmico, o método do PRVJT sofrerá algumas adaptações, as quais são detalhadas no item 3.4.5.

situação o veículo chegaria em k antes de e_k (ou l_{n+1} quando k igual ao depósito), o veículo aguarda na posição atual até o tempo de chegada em k for igual a e_k (ou l_{n+1}).

A estratégia de posicionamento busca evitar que veículos entrem em estado de espera em locais onde há pouca probabilidade de ocorrência de novos pedidos. Esta estratégia será detalhada na seção 3.5.5.

Os problemas teste do PRDVJT também são tratados como problemas estáticos, numa situação utópica em que todas as informações são conhecidas a priori, para que os métodos busca local granular e *Tabu_LP_4_fcnv_ID* do PRVJT possam ser aplicados. O resultado desses testes será utilizado como limitante superior do lucro.

A Tabela 3.2 apresenta os testes realizados.

Tabela 3.2. Testes para o PRDVJT

Nº	Nome do teste	Heurística	Estratégias de Espera	Desvio	Estratégias de Posicionamento
1	<i>LS_BL</i> (limitante superior)	Busca local granular do PRVJT	na*	na*	na*
2	<i>LS_BT</i> (limitante superior)	<i>Tabu_LP_4_fcnv_ID</i> do PRVJT	na*	na*	na*
3	<i>VMP</i>	Vizinho mais próximo			
4	<i>VMP_e</i>	Vizinho mais próximo	✓		
5	<i>VMP_e_p</i>	Vizinho mais próximo	✓		✓
6	<i>I</i>	Inserção			
7	<i>I_e</i>	Inserção	✓		
8	<i>I_d</i>	Inserção		✓	
9	<i>I_e_d</i>	Inserção	✓	✓	
10	<i>I_e_p</i>	Inserção	✓		✓
11	<i>I_e_d_p</i>	Inserção	✓	✓	✓
12	<i>I+_e_p</i>	Inserção+	✓		✓
13	<i>I+_e_d_p</i>	Inserção+	✓	✓	✓
14	<i>BT_e_p</i>	<i>Tabu_LP_4_fcnv_ID_dinâmico</i>	✓		✓
15	<i>BT_e_d_p</i>	<i>Tabu_LP_4_fcnv_ID_dinâmico</i>	✓	✓	✓

* Não se aplica.

3.4. Descrição do ambiente de teste para o PRDVJT

A pesquisa a campo forneceu informações importantes para a geração de problemas teste – total de três problemas – com caráter mais próximo da prática. Esta seção apresenta como o ambiente de simulação foi desenvolvido e como as características práticas foram introduzidas.

3.4.1. Informações sobre a empresa de transporte

O ambiente de teste considerou as seguintes informações para simular a operação da empresa de transporte:

- A empresa é de grande porte (frota acima de 500 veículos). Nas maiores filiais, por exemplo, a filial da região de Grande São Paulo, a empresa dispõe de 70 veículos. Este foi o número de veículos adotado para os problemas teste;
- O depósito abre às 8hs e todos os veículos devem retornar ao mesmo até às 17hs¹². Atrasos no tempo de retorno dos veículos ao depósito não são admitidos;
- Os motoristas são próprios* da empresa e contratados conforme regulamenta a CLT (Consolidação de Leis Trabalhistas), em que há basicamente três modalidades de turnos, descritas na Tabela 3.3;

Tabela 3.3. Modalidades de turnos admitidos pelo ambiente de simulação

Número de horas do turno	Minutos reservados para almoço	Tempo total do turno	Adicional Hora extra
Até quatro horas	0	4:00	50% a partir da quarta hora
De quatro a seis horas	15	4:15 a 6:15	50% a partir da sexta hora
Acima de seis horas	60	7:00 a 9:00	50% a partir da oitava hora

- São transportadas cargas fracionadas através da modalidade rodoviária, principalmente urbana – não são considerados serviços adicionais (seção 1.2.1, c);
- Todo o transporte é feito numa única cidade e esta não é segmentada* por regiões de veículos dedicados;

¹² Apesar dos veículos terem que retornar ao depósito até as 17hs, este não fecha neste horário. O turno da noite faz o trabalho de descarregamento dos veículos.

* Características que diferem das encontradas na pesquisa a campo.

- O nível tecnológico da empresa é (2) – vide seção 1.2.1 – com sistemas de controle de clientes/pedidos e de rastreamento de veículos. Sobre esta tecnologia serão aplicados os módulos de roteamento propostos nesta dissertação;
- Dentre as possibilidades de operações de coleta e entrega, descritas na configuração 2 da Tabela 1.1, somente a operação de coleta será analisada:
 - Como visto numa empresa da pesquisa a campo, as operações de coleta e entrega podem ser completamente independentes, inclusive a sua gerência. Os veículos são dedicados a somente entrega ou a somente coleta e não se admite que sejam feitos “encaixes” de pedidos de coleta nos roteiros dos veículos de entrega. Ainda, é praticamente impossível efetuar entregas de forma dinâmica numa empresa de transporte. O produto a ser entregue teve que ser previamente coletado, e, portanto, existe uma previsão para a data de entrega. Deste modo, somente a operação de coleta pode ter caráter dinâmico, foco deste trabalho;
 - Considerando a operação de coleta, serão tratados dois tipos de pedidos: (1) coleta de pedidos agendados (veículo parte vazio do depósito para coletar os pedidos agendados no dia anterior) e (2) coleta de pedidos novos (novos pedidos podem ser coletados por veículos que já estão realizando roteiros ou por veículos disponíveis no depósito).
- A seguinte política de atendimento de pedidos é adotada:
 - Os pedidos que foram agendados no dia anterior devem ser coletados (pedidos com alta prioridade);
 - A empresa assume o compromisso de coletar no mesmo dia todos os pedidos que receber antes das 14hs (pedidos com alta prioridade);
 - Pedidos recebidos após as 14hs podem* ser coletados no mesmo dia, e sempre serão quando viável (pedidos de baixa prioridade). Note que, atualmente, as empresas de transporte sempre coletam estes pedidos no dia seguinte.
- Os veículos podem ser desviados do destino planejado. Uma vez que a decisão de desvio foi tomada, são necessários 5 minutos para o que o veículo altere seu percurso. Este

* Características que diferem das encontradas na pesquisa a campo.

período corresponde aos tempos de comunicação entre a central e o motorista e de alteração do caminho (retornos e desvios);

- Há um intervalo de 15 minutos entre a decisão de se enviar um veículo para seu primeiro destino – partir do depósito – e o veículo iniciar o percurso. Este período contém os tempos de comunicação ao motorista e de liberação (fila) do veículo no depósito;
- Serão consideradas somente restrições de capacidade e janelas de tempo dos clientes e do depósito*;
- O horizonte da análise é de um dia.

3.4.2. Informações sobre os clientes

Não foi possível coletar uma base de dados com as informações dos clientes para servir como fonte dos problemas teste. Então, a pesquisa a campo foi utilizada para definir premissas – por exemplo, número de clientes atendidos por dia, número médio de clientes por rota, tamanho médio das janelas de tempo – nas quais serão baseados os problemas. As informações foram coletadas com o auxílio do guia de entrevistas do Anexo A.

A primeira premissa sobre o problema teste está relacionada à disposição dos clientes e do depósito no espaço. Existem empresas de transporte com depósito localizado próximo ao centro de massa da disposição dos pedidos (média das coordenadas dos pedidos), e outras com o depósito localizado distante do centro de massa. Na Figura 1.2 pode-se visualizar que o depósito da Transportadora Americana está localizado um pouco à esquerda do centro de massa. Espera-se que depósitos localizados próximo ao centro de massa levem a roteiros de menor comprimento. Contudo, sabe-se que a decisão sobre localização do depósito depende de uma coleção de fatores, tais como preço do imóvel, facilidade de acesso e regulamentações da prefeitura.

Para a geração dos problemas teste, dividiu-se o espaço numa matriz de 150 x 150 posições, totalizando 22.500 posições. Cada posição possui um par de coordenadas (x , y) e uma probabilidade de ocorrência de pedidos. A probabilidade é função da região em que a posição pertence. A Figura 3.1 ilustra as regiões do Problema Teste A.

* Características que diferem das encontradas na pesquisa a campo.

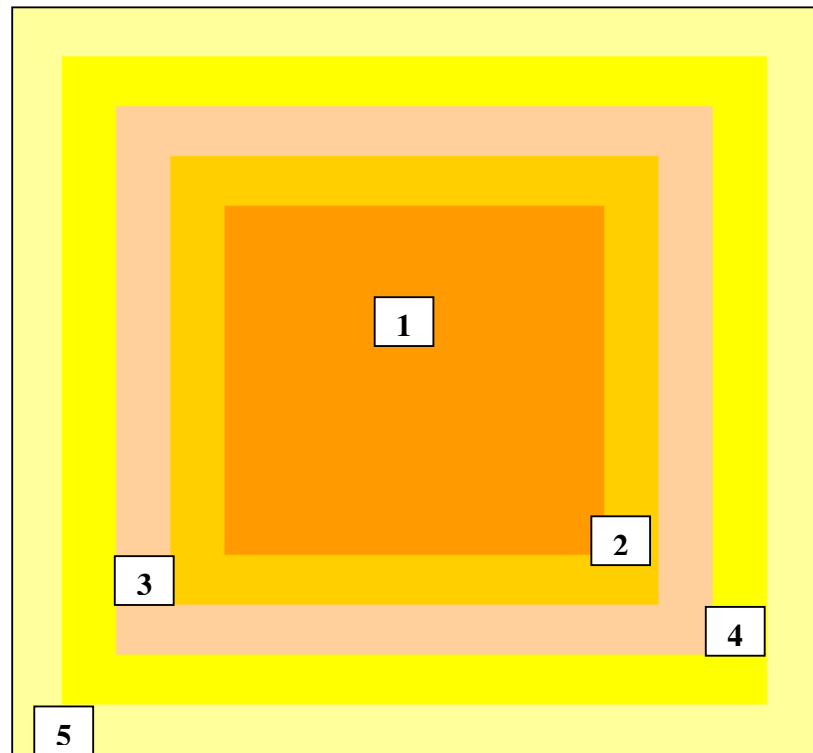


Figura 3.1. As 5 regiões do espaço com diferentes probabilidades de ocorrência de pedidos – Problema Teste A

Na figura, quanto mais escura a cor da região, maior será a probabilidade de ocorrência de pedidos e, conseqüentemente, a concentração de clientes. A probabilidade de ocorrência de pedidos para cada posição dentro de uma região é constante, como mostra a Tabela 3.4.

Tabela 3.4. Probabilidade de ocorrência de clientes por região – Problema Teste A

Região	Probabilidade de existir um cliente numa posição
1	0,128
2	0,064
3	0,032
4	0,016
5	0,008

Para cada posição da região é sorteado um número aleatório $\in [0, 1]$. Se o número sorteado for menor ou igual à probabilidade de um cliente pertencer à posição da região (Tabela 3.4), um cliente é alocado à posição, caso contrário, nenhum cliente é alocado.

Com a disposição das regiões segundo a Figura 3.1 e as probabilidades da Tabela 3.4, são, em geral, sorteados de 1.000 a 1.150 clientes, ilustrados na Figura 3.2. Note que há uma maior concentração de clientes no centro e uma menor concentração de clientes na periferia.

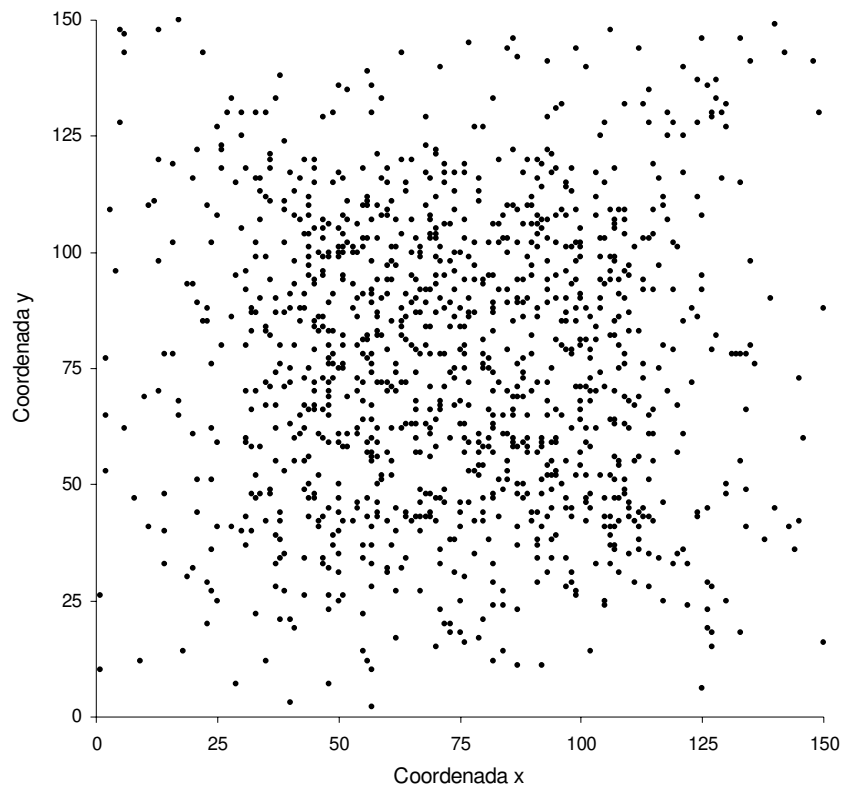


Figura 3.2. Disposição no espaço dos 1101 clientes do Problema Teste A

O Problema Teste A possui o depósito localizado no centro de massa das posições dos pedidos, centro da Figura 3.2, posição (75, 75).

Para uma empresa de transporte com 70 veículos no depósito, a faixa de [1000, 1150] para o número de clientes representa, em média, de 14,3 a 16,4 clientes por rota. Estes valores estão coerentes com os valores das empresas de transporte mais produtivas levantados nas entrevistas – Figura 3.3.

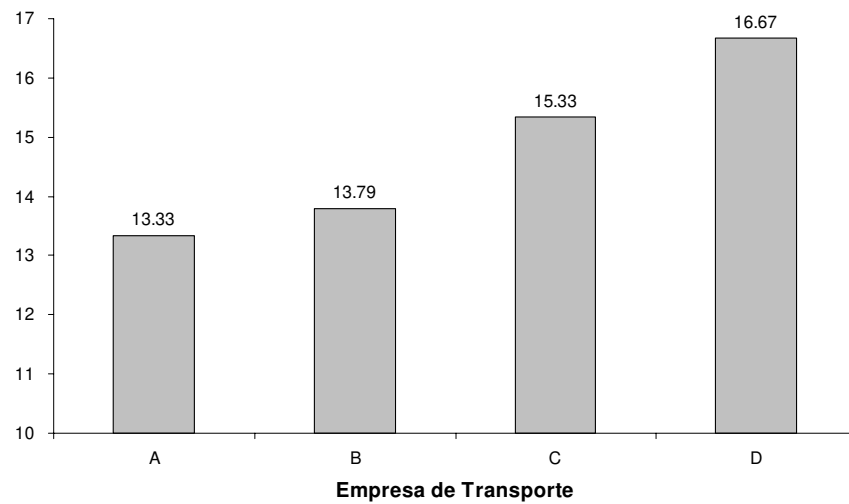
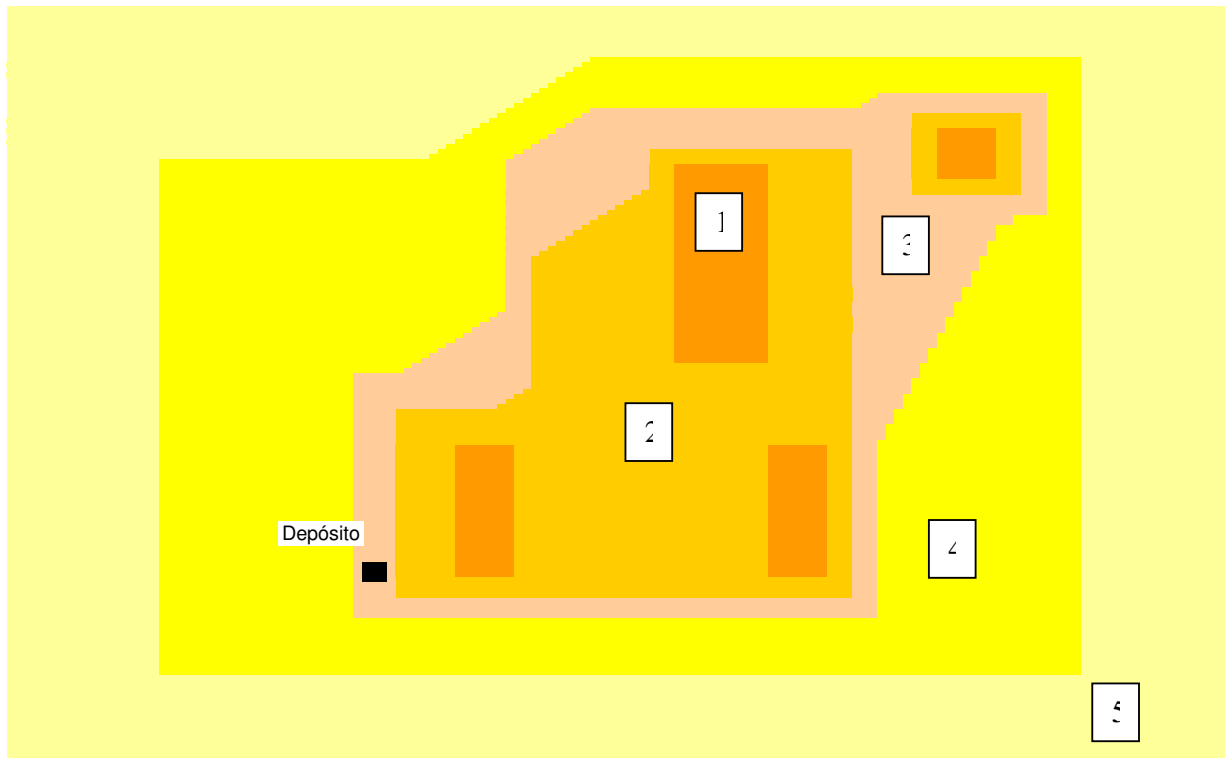


Figura 3.3. Número médio de clientes por rota das empresas de transporte

Para criar os problemas teste B e C, as premissas foram modificadas buscando simular a situação da cidade de São Paulo, baseada na Figura 1.2. O problema teste B localiza o depósito na região da zona sul, próximo à Marginal Pinheiros, enquanto o problema teste C é uma tentativa de se aproximar da realidade da Transportadora Americana.



**Figura 3.4. As 5 regiões do espaço com diferentes probabilidades de ocorrência de pedidos –
Problema Teste B**

O depósito está localizado na posição (50, 40) – quadrado preto da Figura 3.4.

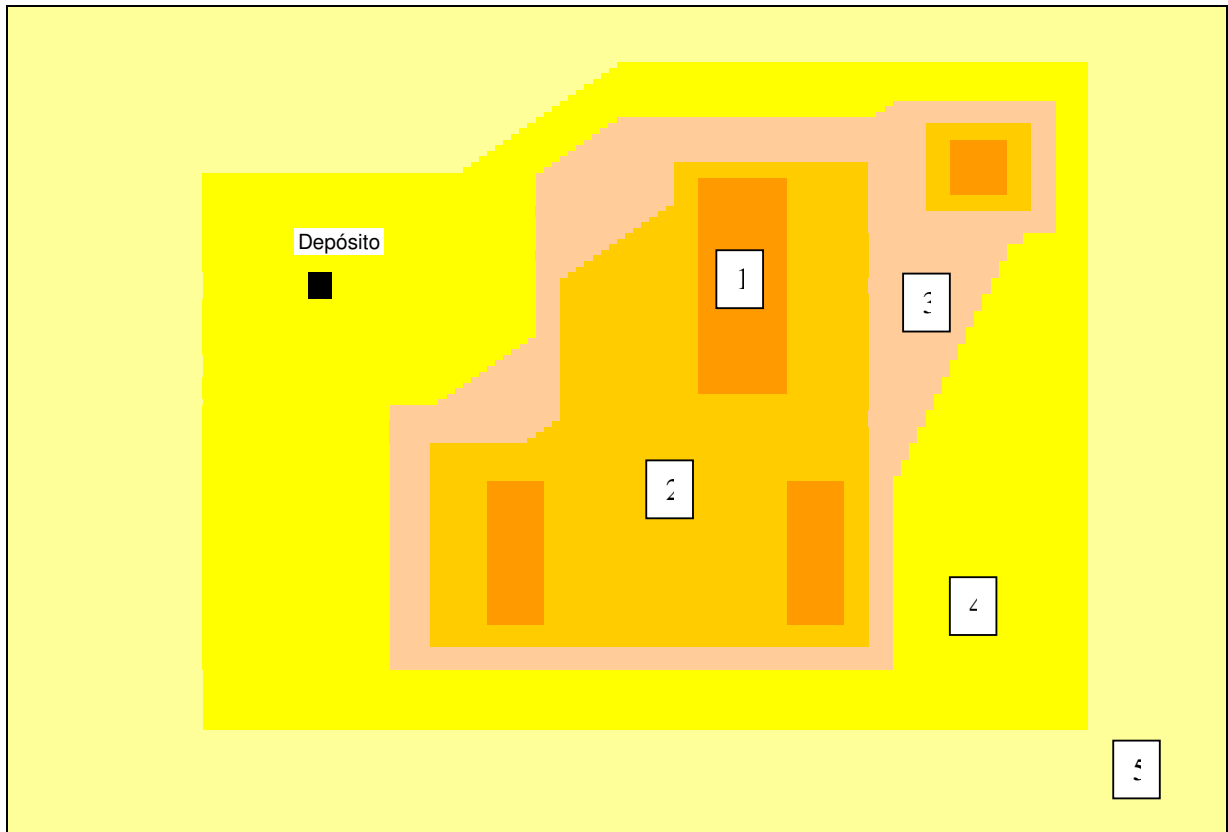


Figura 3.5. As 5 regiões do espaço com diferentes probabilidades de ocorrência de pedidos – Problema Teste C

O depósito está localizado na posição (30, 110) – quadrado preto da Figura 3.5.

Para manter o número de clientes entre 1.000 e 1.150, foi necessário modificar as probabilidades de cada região, como apresenta a Tabela 3.5.

Tabela 3.5. Probabilidade de ocorrência de clientes por região – Problemas Teste B e C

Região	Probabilidade de existir um cliente numa posição
1	0,396
2	0,132
3	0,044
4	0,022
5	0,011

A partir destas premissas, a localização dos clientes do Problema Teste B segue a Figura 3.6.

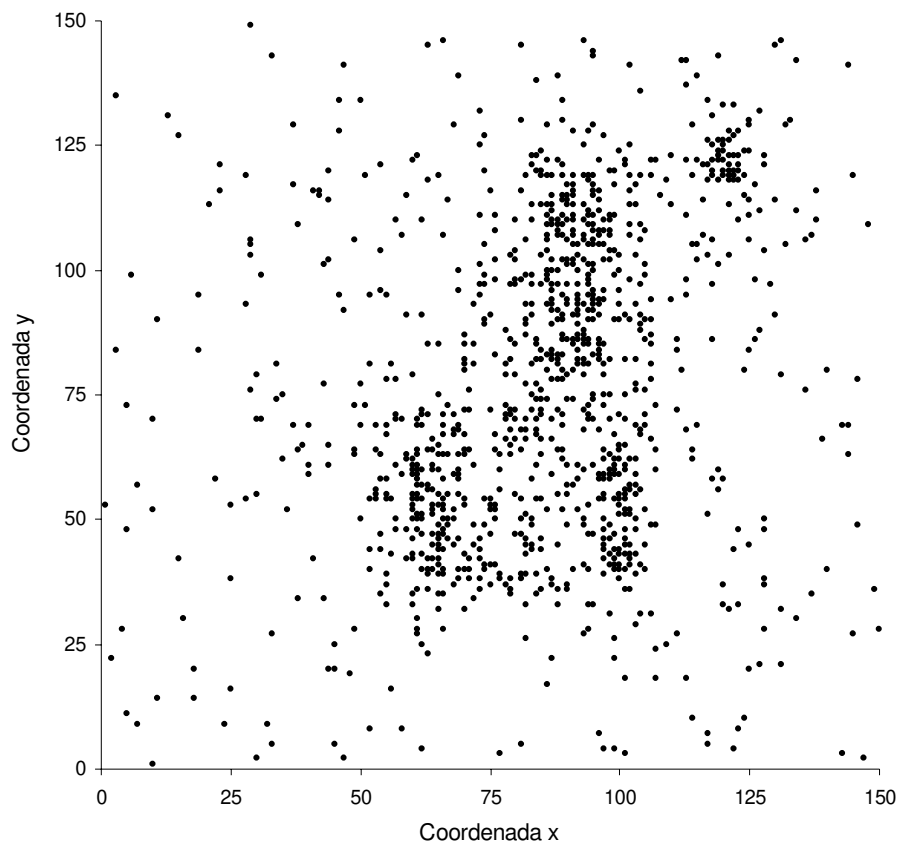


Figura 3.6. Disposição no espaço dos 1060 clientes do Problema Teste B

O Problema Teste C possui 1.080 clientes com disposição semelhante à da Figura 3.6.

A próxima premissa está relacionada ao momento de chegada do pedido. Primeiramente é necessário classificar os pedidos nos quatro tipos de situações encontradas na pesquisa a campo:

1. Pedidos estáticos de entrega (PEE^+): pedidos que estão no depósito e devem ser necessariamente carregados nos veículos e entregues no dia atual – fora do escopo desta dissertação;

2. Pedidos estáticos de coleta (PEC^+): pedidos que não foram coletados no dia anterior e devem ser necessariamente coletados no dia atual;
3. Pedidos dinâmicos de coleta, período manhã (PDC^+): Novos pedidos que chegaram antes das 14hs que devem ser coletados no dia atual;
4. Pedidos dinâmicos de coleta, período tarde (PDC^-): Novos pedidos que chegaram após as 14hs e podem ser coletados no dia atual. O prazo de coleta destes pedidos é até o fim do dia seguinte, respeitando as janelas de tempo.

Observações: a) os índices $^+$ ou $^-$ indicam a prioridade do pedido: pedidos com índice $^+$ devem ser atendidos no dia enquanto os de índice $^-$ podem ser atendidos no dia; b) perceba que se a empresa tem como política coletar todos os no PDC^- no dia seguinte, o número de PDC^- do dia atual é igual ao número de PEC^+ do dia seguinte.

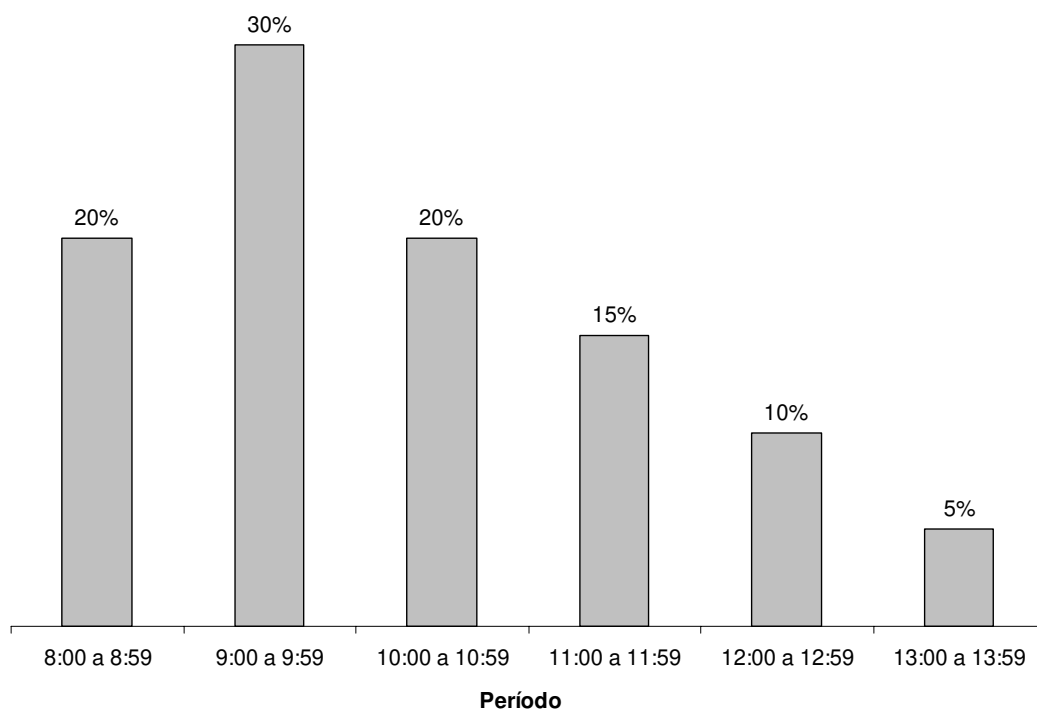
Para a operação de coleta, a empresa de transporte expôs a seguinte situação: de todos os pedidos coletados no dia, um terço foi agendado no dia anterior, um terço chega antes das 14hs, e o restante chega após as 14hs. Utilizaremos esta informação como premissa para os Problemas Teste B e C.

Para o Problema Teste A, aumentaremos o nível de dinamismo. Assumiremos que um quinto do total de pedidos foi agendado no dia anterior (PDC^-). Desta maneira, temos que o número de PEC^+ também é igual a um quinto do total de pedidos. Portanto, o número de PDC^+ é igual a 3/5 do total de pedidos. A Tabela 3.6 resume estas informações:

Tabela 3.6. Percentual de pedidos por tipo por problema teste

Tipo de pedido	Problema Teste		
	A	B	C
PEC ⁺	20%	33%	33%
PDC ⁺	60%	33%	33%
PDC ⁻	20%	33%	33%
Total	100%	100%	100%

A distribuição dos PDC⁺ no período entre 8hs e 14hs é mostrada na Figura 3.7.

**Figura 3.7. Distribuição do percentual de PDC⁺ no período entre 8hs e 14hs**

Como observado nas entrevistas, o pico de chegada de pedidos é entre as 9hs e 10hs, com redução gradual até às 14hs. Esta redução é mantida para os PDC⁻, como mostra a Figura 3.8.

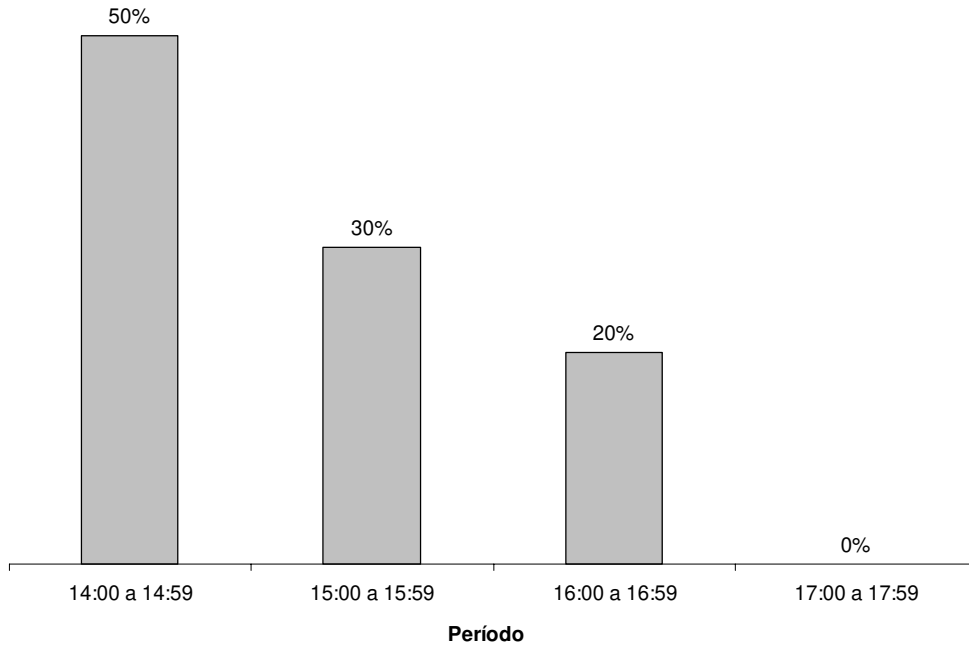


Figura 3.8. Distribuição do percentual de PDC⁻ no período entre 14hs e 18hs

Estas distribuições são mais realistas que uma distribuição uniforme, pois é razoável imaginar que se um cliente deseja ser atendido no mesmo dia, ele estaria entrando em contato com a empresa o quanto antes, preferencialmente pela manhã. Hvattum *et al.* (2004) apresentam um estudo de caso com comportamento semelhante.

Para que os problemas teste reflitam estas distribuições, são associadas taxas λ_t de número de novos pedidos por segundo para cada período t . A partir de λ_t , a probabilidade da ocorrência de k clientes em cada segundo do período t segue a distribuição de Poisson:

$$P(k, t) = \frac{\lambda_t^k e^{-\lambda_t}}{k!}$$

Por exemplo, suponha $\lambda_8 = 0,067$ novos clientes/segundo, equivalente a 4 novos clientes por minuto para o período entre 8:00 e 8:59. A partir desta taxa, a probabilidade de chegar nenhum cliente no momento 8:10:02 é 0,935507%. A probabilidade de chegar um cliente neste mesmo momento é 0,062367%. Como ilustração, a Tabela 3.7 mostra as probabilidades para $\lambda_8 = 0,067$.

Tabela 3.7. Distribuição de Poisson para uma taxa média de 0.067 clientes / segundo

Número de novos clientes	Probabilidade da ocorrência (%)
0	0,935507
1	0,062367
2	0,002079
3	4,62E-05

Seguindo o exemplo com $\lambda_s = 0,067$, para cada unidade de segundo entre o período 8:00:00 e 8:59:59 é sorteado um número aleatório $\in [0, 1]$. Se este número for menor ou igual a 0.935507, nenhum cliente é alocado. Se o número estiver entre 0.935507 e $0.935507 + 0.062367$, um cliente é alocado. Se o número estiver entre $0.935507 + 0.062367$ e $0.935507 + 0.062367 + 0,002079$, dois clientes são alocados, e assim por diante.

Resta ainda unir a informação do momento de chegada do cliente à do posicionamento do cliente no espaço, processo feito de forma aleatória. A seguir, mostra-se como isto é feito, por exemplo, para o Problema Teste A:

1. Crie uma lista E com as 1.101 coordenadas dos clientes segundo a disposição da Figura 3.2;
2. Sorteie aleatoriamente de E 219 clientes (20% do total). Designe aos clientes sorteados o rótulo de PEC^+ e o momento de chegada igual a 8:00:00. Remova estes clientes da lista E ;
3. Crie uma lista T_PDC^+ com 662 momentos de chegadas de clientes (662 é equivalente a 60% do total de clientes) segundo a distribuição da Figura 3.7 e a expressão da distribuição de Poisson – a criação desta lista requer a definição de λ_t para cada período entre 8hs e 14hs;
4. Faça $i_PDC^+ = 0$;
5. Enquanto $i_PDC^+ < 662$, faça:
 - 5.1. Sorteie aleatoriamente um cliente de E ;
 - 5.2. Aloque a este cliente o primeiro momento disponível da lista T_PDC^+ . Marque o momento alocado como não-disponível;
 - 5.3. Remova o cliente sorteado da lista E e faça $i_PDC^+ = i_PDC^+ + 1$. Retorne ao passo 5.

6. Crie uma lista T_PDC^- com 220 momentos de chegadas de clientes segundo a distribuição da Figura 3.8 e a expressão da distribuição de Poisson – a criação desta lista requer a definição de λ_t para cada período entre 14hs e 18hs;
7. Faça $i_PDC^- = 0$;
8. Enquanto $i_PDC^- < 220$, faça:
 - 8.1. Sorteie aleatoriamente um cliente de E ;
 - 8.2. Aloque a este cliente o primeiro momento disponível da lista T_PDC^- . Marque o momento alocado como não-disponível;
 - 8.3. Remova o cliente sorteado da lista E e faça $i_PDC^- = i_PDC^- + 1$. Retorne ao passo 8.
9. FIM.

Para definir a janela de tempo de cada cliente, assumimos e_i idêntico ao momento de chegada do pedido. Para os pedidos agendados, significa que a coleta pode ser feita a partir das 8hs, enquanto para os novos pedidos, significa que o cliente está preparado para receber o veículo de coleta a partir do momento em que o cliente entrou em contato com a empresa de transporte. Como assumimos que l_i é função de e_i (mostraremos a seguir que $l_i - e_i \in [1, 5]$), fazer e_i equivaler ao momento de chegada reduz o valor de l_i , aumentando o dinamismo do problema e requerendo resposta mais rápida do sistema de roteamento.

Para calcular l_i é necessário definir a $duração_i$ da janela de tempo, pois $l_i = e_i + duração_i$. A $duração_i$ para os PDC^+ e PDC^- é sorteada de acordo com a distribuição da Tabela 3.8:

Tabela 3.8. Distribuição da duração da janela de tempo ($duração_i$)

$duração_i$ (horas)	Probabilidade
1	0.1
2	0.2
3	0.4
4	0.2
5	0.1

As probabilidades da Tabela 3.8 indicam que a maioria dos clientes tem duração de janela de tempo de 3 horas – a pesquisa mostrou que quando existe janela de tempo, sua duração fica

entre 2hs e 4hs. Na prática, existem clientes sem janela de tempo, mas, nesta dissertação, foram associadas janelas de tempo a todos os clientes.

Para os PEC⁺ utilizou-se de uma duração fixa de 4hs. Portanto, todos os PEC⁺ possuem $e_i = 8:00:00$ e $l_i = 11:59:59$, representado que todos os pedidos agendados no dia anterior serão coletados até às 12hs do dia seguinte.

Caso ocorra atraso no atendimento, uma multa é aplicada. A multa está relacionada à receita que o cliente proporciona para a empresa de transporte. Por exemplo, se o atraso for muito grande, assumimos que o cliente não pagaria nada pelo transporte. Considerando que os clientes paguem em média R\$ 200,00 por coleta e entrega (valor obtido na pesquisa a campo), foi adotada a política de multas da Tabela 3.9:

Tabela 3.9. Política de multas por $duração_i$

$duração_i$	Multa por hora atrasada (R\$/hora)	Multa por segundo atrasado (R\$/segundo)
1	200,00	0,055556
2	100,00	0,027778
3	66,67	0,018519
4	50,00	0,013889
5	40,00	0,011111

Para evitar problemas de relacionamento com os clientes, o atraso máximo tolerado é de 2 horas.

Finalmente, foram consideradas as premissas da Tabela 3.10 para definir a receita por cliente ($receita_i$), a demanda d_i e tempo de serviço s_i , valores também baseados na pesquisa a campo.

Tabela 3.10. Distribuições da $receita_i$, d_i e s_i

Informação	Distribuição utilizada	Média	Desvio Padrão
$receita_i$	Normal	R\$ 200	R\$ 40
d_i	Normal	100 unidades de demanda	20 unidades de demanda
s_i	Normal	15 minutos	3 minutos

3.4.3. Informações sobre os veículos e contribuição ao lucro da operação de coleta

Para que o método possa maximizar a receita e minimizar os custos de transporte e de multas, foi necessário estimar o custo fixo e variável dos veículos.

Nas entrevistas, fomos informados de que um veículo para transporte urbano com capacidade de até 3 toneladas pode ser adquirido por R\$ 70.000. No site da divisão de transporte da Universidade Federal de Viçosa (<http://www.cpd.ufv.br/dtr/>), temos um valor de R\$ 75.000 para um utilitário urbano. Assumindo um investimento de R\$ 75.000 por veículo, acrescentando as despesas de licenciamento e seguro obrigatório, e considerando um valor residual de 10% após dez anos, temos um custo fixo diário de aproximadamente R\$ 42,00 – vide Anexo J para o detalhamento de todos os cálculos aqui descritos.

O cálculo do custo do motorista foi baseado num rendimento de 4 salários mínimos, com todos os encargos e benefícios regulamentados pela CLT. O custo do motorista foi estimado em R\$ 2.184,00 / mês, no limite superior da faixa de valores apontada nas entrevistas (de R\$ 1.800/mês a R\$ 2.200/mês). Isto representa um custo diário de aproximadamente R\$ 100/dia.motorista, pois foi considerada a média de 22 dias úteis por mês.

Portanto, estaremos assumindo um custo fixo por veículo de aproximadamente **R\$ 142,00** por dia.

O custo variável mensal, segundo a pesquisa, pode ser estimado de 1,5% a 2% do valor do investimento do veículo – de R\$ 1.125 a R\$ 1.500 por mês. Após desenvolver a planilha de cálculo do Anexo J, chega-se a um custo mensal de R\$ 1.829,07. Estaremos utilizando este valor, resultando num custo diário aproximado de **R\$ 83,00** por veículo.

A quilometragem média para um turno de 8 horas nos foi passada entre 150 e 200 quilômetros. O relatório do Anexo L mostra uma quilometragem média de 232,77 para um período de 10 horas e 11 minutos, equivalente a 182,86 km para o turno de 8 horas. Para refinar este cálculo, podemos ainda considerar um roteiro com 15 clientes atendidos, com um tempo de serviço de 15 minutos cada, totalizando um tempo de 3,75. Subtraindo de 8 (horas produtivas do turno) este valor, temos 4,25 horas de veículos em trânsito. Adotando como velocidade média 40 km/h – velocidade citada nas entrevistas e média dos valores das velocidades em trânsito do

Anexo L – o veículo poderia percorrer **170 km** num dia (40 vezes 4,25), que foi o valor adotado como quilometragem média diária.

Logo, os custos fixo e variável de um veículo são descritos na Tabela 3.11:

Tabela 3.11. Custos fixo e variável de um veículo¹³

Tipo de Custo	R\$/dia	R\$/km
Fixo	142,00	0,84
Variável	83,00	0,49

No *site* PME Link (http://www.pmlink.pt/pmlink_public/EC/0,1655,1005_5053-3_41104--View_429,00.html) e em outros de fóruns de discussão de motoristas e de profissionais de logística, pode-se encontrar também estimativas de custo variável de aproximadamente R\$ 0,40 / km.

Observação: A Tabela 6.19 do Anexo J mostra que são necessários, no mínimo, roteiros com dois clientes para que a contribuição ao lucro da rota seja positiva – desconsiderando custos administrativos e de movimentação e armazenagem.

Para calcular os valores do tempo de viagem t_{ij} e custo c_{ij} é necessário escalonar as distâncias da matriz de 150 x 150 posições, na qual foram baseadas as Figuras 3.2 e 3.6. Por exemplo, se o escalonamento não fosse feito, a distância entre o depósito localizado no centro e um cliente localizado numa extremidade seria:

$$distância_{centro,extremidade} = \sqrt{(0-75)^2 + (0-75)^2} = 106$$

Para um veículo partir do depósito, visitar este cliente localizado na extremidade e retornar ao depósito, é necessário percorrer uma distância total de $2 \times (distância_{centro,extremidade}) = 212$. Se a unidade de $distância_{centro,extremidade}$ for exatamente um quilômetro, temos que $distancia_em_km_{centro,extremidade} = distância_{centro,extremidade}$. Desta maneira, o tempo para um veículo com velocidade média de 40km/h percorrer 212 km é de aproximadamente 5,3 horas – tempo demasiadamente demorado e pouco realista. Então, mantendo as distâncias na forma euclidiana, foi definido o seguinte fator de correção:

¹³ As Tabelas 6.15 e 6.17 contém o cálculo dos custos em maior detalhe.

$$fator_correção_distância = \frac{distância_em_km_{i,j}}{distância_{i,j}} = 0,4$$

Aplicando o $fator_correção_distância$, o tempo para se visitar o cliente da extremidade e retornar ao depósito é de aproximadamente duas horas.

Logo, t_{ij} é calculado com a expressão:

$$t_{ij} = \frac{(0,4 \times distância_{i,j})km}{40km/h} = 0,01 \times distância_{i,j} \quad (em \text{ horas})$$

e c_{ij} com a expressão:

$$c_{ij} = 0,49 \text{ R\$/km} \times (0,4 \times distância_{i,j})km = 0,196 \times distância_{i,j} \quad (em \text{ R\$})$$

Conseqüentemente, o custo da solução de um problema teste é calculado da seguinte forma:

$$custo_solução_coleta = \sum_{k \in K} \sum_{(i,j) \in A} (c_{ij} + multa_i) X_{ij}^k + |K| \times 142,00,$$

tal que $multa_i$ representa a multiplicação do número de segundos de atraso no atendimento do cliente i pela respectiva multa da Tabela 3.9., e $|K|$ representa o número de veículos que partiram do depósito ($|K| \leq 70$).

Finalmente, o lucro da operação é definido como:

$$lucro_operação_coleta = \frac{\sum_{k \in K} \sum_{(i,j) \in A} receita_i X_{ij}^k}{2} - custo_solução_coleta$$

A parcela da receita é dividida por dois porque para cada cliente atendido são necessárias duas viagens: num dia um veículo é enviado para efetuar a coleta e em outro dia outro veículo é enviado para efetuar a entrega. Portanto, associamos metade da receita para a operação de coleta e metade para a de entrega, sendo esta fora do escopo da dissertação.

A capacidade do veículo foi estipulada em 1.800 unidades de demanda, comportando, em média, 18 pedidos de 100 unidades de demanda cada. Nestas condições é possível atender mais do que a média de 15 clientes, que considera somente a restrição de janela de tempo do depósito (turno de 8 horas). Esta folga de 3 clientes foi acrescentada propositalmente, pois permite que ambas as restrições possam ficar ativas: a) se um veículo atender clientes próximos entre si, possivelmente a restrição de capacidade ficará ativa, b) se um veículo atender clientes distantes, o

tempo de viagem será elevado e é provável que o veículo deva retornar ao depósito, mesmo com baixa taxa de ocupação.

3.4.4. Desenvolvimento do ambiente de teste para o PRDVJT

O ambiente de simulação consiste em módulos independentes desenvolvidos através da linguagem de programação C++. Cada módulo representa um departamento da empresa e troca informações com outros módulos através de um banco de dados MySql.

A apresentação de cada módulo é feita seguindo o processo de atendimento de um novo pedido, como foi descrito no item 1.2.3. A Figura 3.9 ilustra o processo de chegada e aceite do pedido.

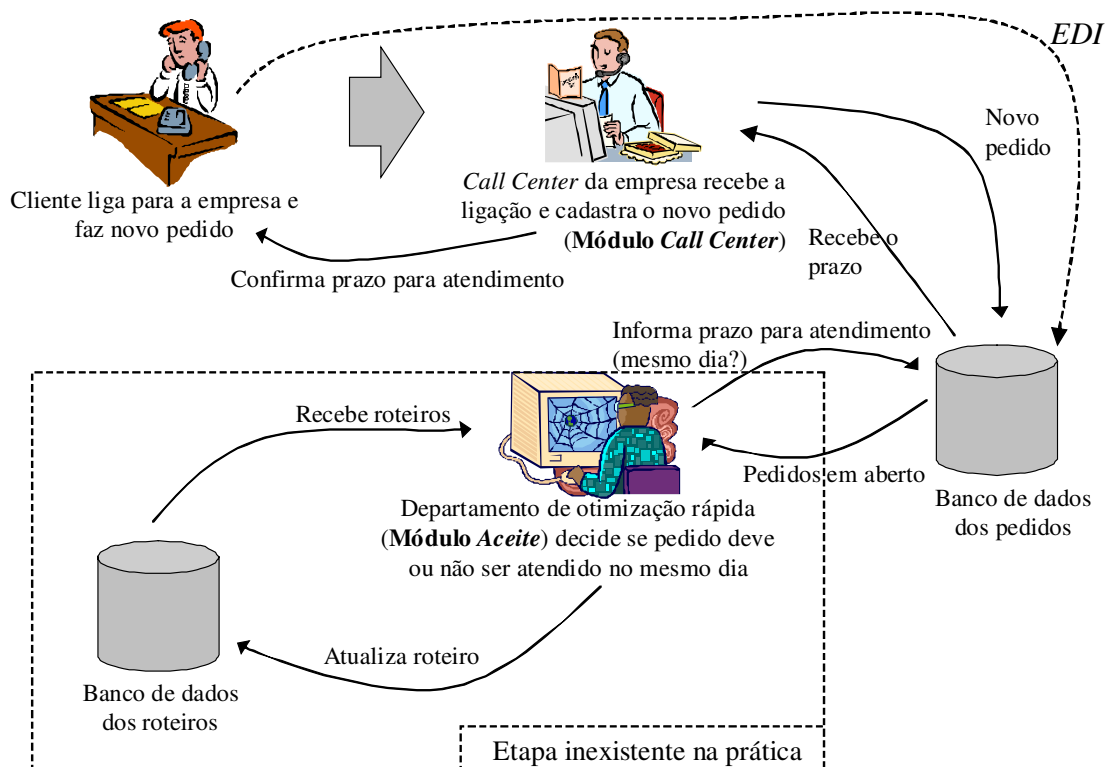


Figura 3.9. Etapas de recebimento de pedidos e confirmação de atendimento

Para estas etapas foram desenvolvidos dois módulos, um módulo denominado *Callcenter* e outro *Aceite*.

O módulo *Callcenter* simula o recebimento de ligações dos clientes. Numa fase inicial, este módulo cadastra todos os PEC^+ no banco de dados dos pedidos. Em seguida, faz a leitura de todos os PDC^+ e PDC^- do problema teste e fica em modo de espera. Toda vez em que a hora do sistema – tempo marcado pelo relógio do computador – for superior ao momento de chegada do pedido, este é cadastrado no banco de dados dos pedidos como pedido em aberto.

O módulo *Aceite* acrescenta uma etapa de otimização que atualmente não é encontrada na prática¹⁴. Este módulo consulta o banco de dados dos pedidos a cada dois segundos a procura de pedidos em aberto. Ao encontrar um pedido em aberto, seus dados são lidos, as rotas do banco de dados dos roteiros são consultadas e um processo de otimização rápida é iniciado. A otimização rápida busca a melhor posição de inserção factível do novo pedido nos roteiros. Caso uma posição de inserção tenha sido encontrada, o pedido é marcado como aceito para atendimento no mesmo dia. Caso contrário, o pedido é agendado para o dia seguinte.

Como o método do vizinho mais próximo não possui a fase de melhor inserção, o módulo *Aceite* será desativado para a realização de seus testes.

A Figura 3.10 ilustra a movimentação dos veículos, a comunicação dos motoristas com a empresa (e vice-versa), e a gestão dos roteiros e das coletas completadas – simulação da seção 1.2.4.

¹⁴ Segundo uma empresa entrevistada, segue-se, a qualquer custo, a política de atendimento: pedidos recebidos antes das 14hs devem ser atendidos no mesmo dia. Porém, numa discussão sobre o assunto, a inserção de uma etapa de otimização rápida, como a do módulo *Aceite*, foi vista como uma grande contribuição para se reduzir o número de prazos não cumpridos, além de ser uma alternativa para a flexibilização da política de atendimento.

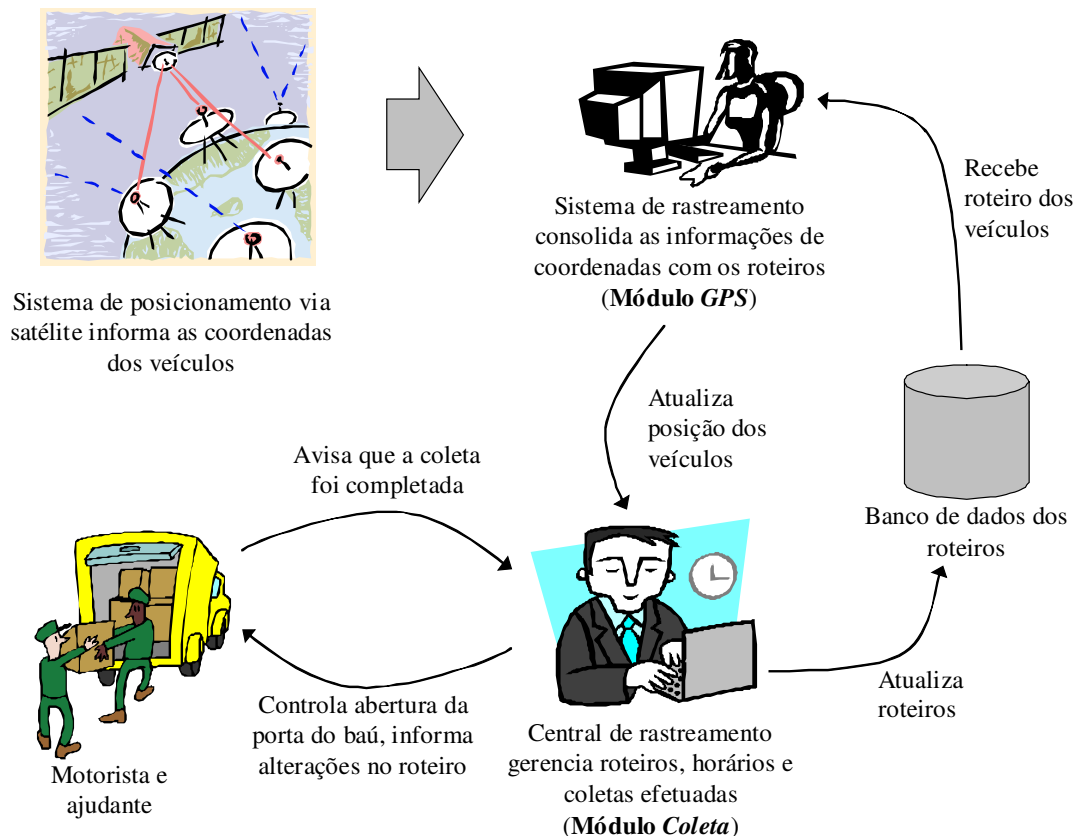


Figura 3.10. Etapas de movimentação dos veículos e rastreamento das coletas

Os dois módulos que simulam estas etapas são *GPS* e *Coleta*.

O módulo *GPS* controla e movimenta os veículos no espaço, armazenando numa tabela do banco de dados as seguintes informações sobre os veículos: coordenadas (x , y), momento da última atualização da posição, carga (demanda carregada), quilometragem e tempo de viagem até o momento.

Para atualizar estas informações, a cada dez segundos o módulo *GPS* seleciona todos os veículos que possuem como rótulo *parado* ou *transitando* e cujas posições não foram atualizadas há pelo menos 4 segundos. Para cada veículo selecionado é calculada a diferença de tempo entre a hora atual e o momento da última atualização. Com esta diferença de tempo, velocidade média do veículo (40 km/h), distância e coordenadas do próximo destino, o módulo *GPS* projeta as próximas coordenadas.

Se o veículo chegar a uma distância suficientemente próxima do destino (utilizamos 100 metros como distância suficientemente próxima), o rótulo do veículo é alterado para *coletando*,

permanecendo neste rótulo até receber novas instruções do módulo *Coleta*. Caso o próximo destino seja o depósito, o rótulo do veículo é alterado para *final_depósito* e o veículo não poderá mais ser utilizado para outras coletas.

Caso a simulação adote estratégia de espera, o rótulo do veículo é determinado para *esperando* quando o tempo de chegada no próximo destino for inferior a e_i . Seu rótulo é posteriormente mudado para *parado* e em seguida para *transitando* quando o tempo de chegada for maior ou igual a e_i . Além disso, se o veículo estiver em espera e o tempo atual for entre 11hs e 14hs – horário em que o motorista pode iniciar seu almoço – e o veículo ainda não efetuou a parada para almoço, o rótulo *esperando* é automaticamente alterado para *almoçando* (o motorista recebe uma mensagem para deixar o veículo e ir almoçar), ficando neste estado por uma hora.

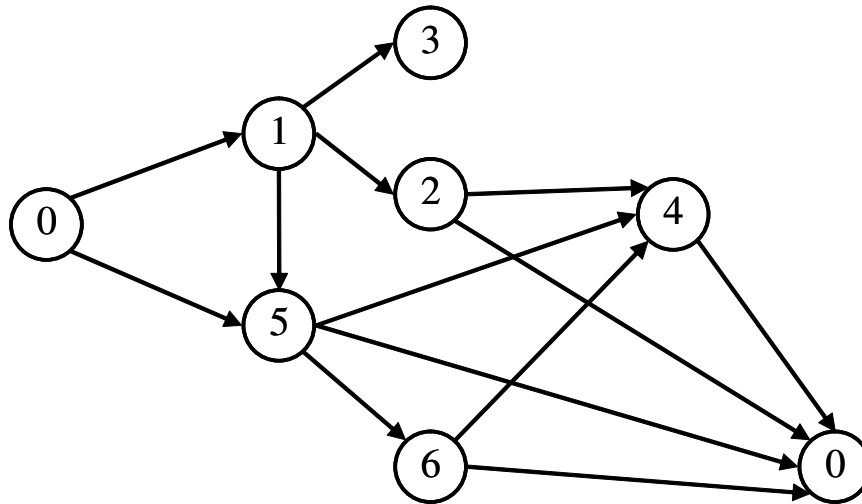
O módulo *GPS* também monitora todos os horários de almoço, alterando os rótulos de *almoçando* para *parado* assim que a duração do almoço tenha sido completada.

Quando é permitida a estratégia de posicionamento, os veículos em espera podem receber o rótulo *reposicionando* caso sua capacidade seja inferior a 1.650 e a diferença entre l_{n+1} e a soma do tempo da posição atual até um ponto estratégico com o tempo do ponto estratégico até o depósito seja superior a 30 minutos. Esta condição garante que o veículo a ser reposicionado poderá coletar um cliente adicional enquanto se movimenta até o ponto estratégico – a estratégia de posicionamento será detalhada na seção 3.5.5. Também, se o rótulo do veículo é *reposicionando* e o tempo atual está entre 11hs e 14hs e o veículo não parou para almoço, o veículo recebe o rótulo *almoçando*.

O módulo *Coleta* cria um lista de todos os veículos com rótulos *coletando*. Se o rótulo do veículo é *coletando*, o módulo *Coleta* acrescenta à carga do veículo a demanda do cliente visitado e soma o tempo de serviço do cliente ao tempo de viagem do veículo. Quando o tempo atual for superior ao tempo de viagem somado ao tempo de serviço, o módulo *Coleta* altera o rótulo para *parado*, ou, *almoçando*¹⁵.

¹⁵ A preferência é que os motoristas almocem próximo aos locais dos clientes. Portanto, a posição de almoço é , em geral, coincidente com a posição de atendimento de algum cliente do roteiro.

A Figura 3.11 mostra o diagrama de rótulos (estados) dos veículos.



Legenda dos rótulos: (0) *parado*, (1) *transitando*, veículo em movimento; (2) *coletando*, veículo chegou ao destino e está efetuando a coleta; (3) *final_depósito*, veículo terminou seu roteiro; (4) *almoçando*, veículo parado em horário de almoço, (5) *esperando*, veículo em estado de espera até novo comando, (6) *reposicionando*, veículo está se movendo a um ponto de espera estratégico.

Figura 3.11. Diagrama de rótulos dos veículos

A seguir, a Figura 3.12 ilustra o módulo de otimização em tempo real dos roteiros.

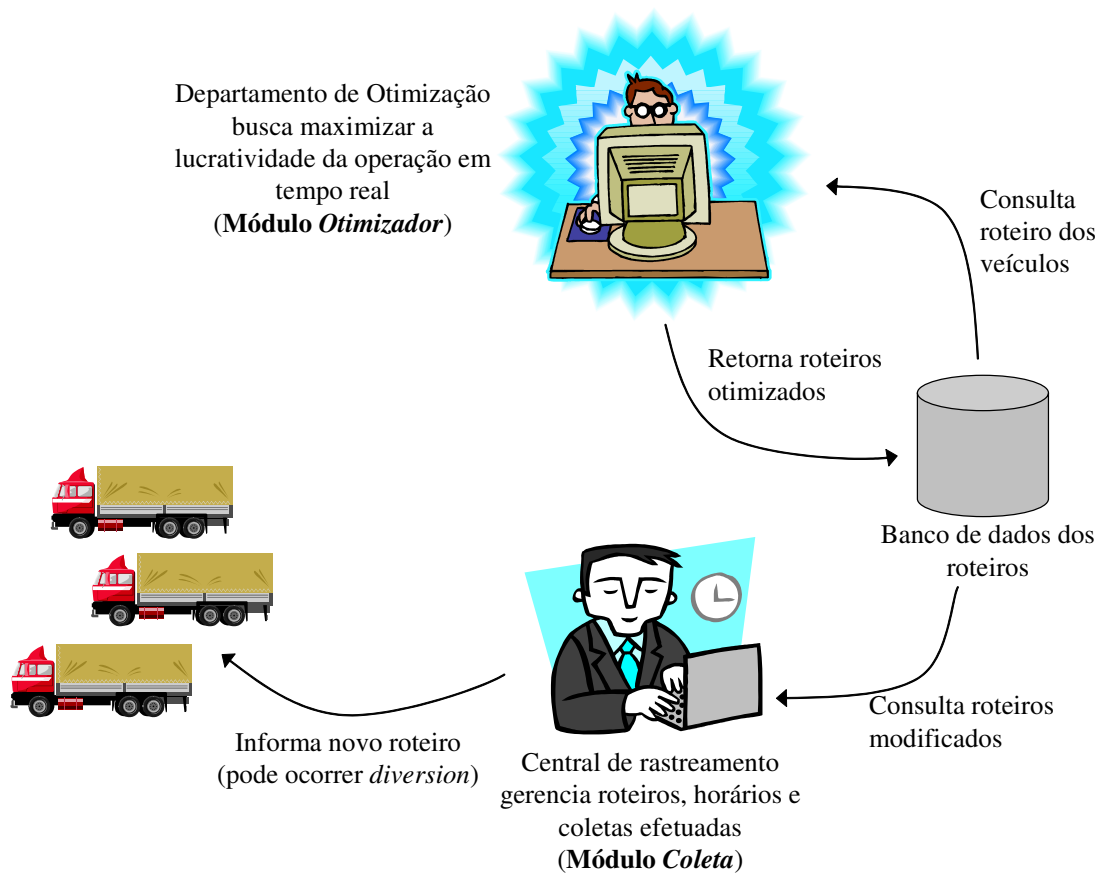


Figura 3.12. Otimização em tempo real dos roteiros

O módulo *Otimizador* consulta o banco de dados dos roteiros e as coordenadas atuais dos veículos. A partir dessas informações, o módulo projeta como seria a posição e os rótulos dos veículos para um período futuro de *tempo_projeção* (*tempo_projeção* é definido em 20 segundos).

O módulo projeta a situação atual alguns segundos no futuro para compensar o tempo de execução da heurística de otimização. De nada adiantaria tirar uma “foto” da situação atual, otimizá-la por 10 segundos e, no momento em que a solução otimizada fosse implementada, esta não é mais vantajosa ou factível (por exemplo, um veículo entrou em horário de almoço).

Note que a definição do *tempo_projeção* possui um balanço: de um lado, tempos curtos reduzem a probabilidade do método encontrar movimentos factíveis, de outro, tempos longos podem resultar em soluções pouco otimizadas.

Existem situações em que mesmo a solução projetada e otimizada não poderá ser aproveitada porque, durante o processo de otimização, ocorreu pelo menos um dos casos abaixo:

- Algum veículo chegou ao próximo destino;
- Um novo pedido chegou e este foi inserido no roteiro de algum veículo;
- O horário de almoço de algum veículo começou ou terminou;
- Algum veículo entrou em espera (ou saiu da espera);
- O roteiro de algum veículo foi alterado por algum módulo (*Callcenter*, *Aceite*, *GPS*, *Coleta*, *Otimizador*).

3.5. Adaptações dos métodos do problema estático para o dinâmico

Resultados dos experimentos computacionais

3.5.1. Adaptações gerais a todos os métodos

A Tabela 3.12 mostra as principais adaptações feitas nos métodos para a simulação do problema dinâmico:

Tabela 3.12. Adaptações ao tratar o problema dinâmico

Adaptação	Dificuldade	Implicações e Observações
• Implementação da estrutura de dados no formato de tabelas do MySQL.	Altíssima	• Aprendizado da linguagem SQL e sua integração com C++; • Replicação e gerenciamento de duas estruturas de dados, uma no banco de dados e outra em C++, para cada módulo.
• Acréscimo de controles no processo de otimização: tempo máximo do método heurístico, monitoramento de eventos que causam reinício da otimização.	Altíssima	• Ocorriam erros graves e esporádicos (difíceis de serem depurados) durante a simulação, os quais foram causados pelo atraso da atualização do banco de dados¹⁶ e falta de sincronismo dos módulos na leitura e escrita da solução armazenada no MySQL. Garantir a coerência entre as soluções do banco de dados e as soluções armazenadas em cada módulo foi um grande desafio.
• Modificação da função objetivo para considerar custos e receitas em dinheiro (R\$).	Alta	• Novo método para avaliar o lucro da solução; • Modificação dos métodos que procuram e avaliam movimentos (intra-rota troca-arcos e inserção, inter-rota <i>crossover</i>, inserção e troca).

¹⁶ O servidor do banco de dados é remoto e acessado através de uma rede local (*Local Area Network* - LAN).

Adaptação	Dificuldade	Implicações e Observações
Flexibilização das restrições de janela de tempo para permitir atrasos com a aplicação de multas.	Alta	Modificação dos métodos que procuram e avaliam movimentos (intra-rota troca-arcos e inserção, inter-rota <i>crossover</i>, inserção e troca).
Inclusão de paradas de até uma hora para almoço dos motoristas.	Alta	Modificação do método que avalia se um movimento é factível: primeiro é testado o movimento sem considerar a parada de almoço. Durante esta etapa, são armazenadas as k posições factíveis (clientes) para se alocar paradas de almoço. Se $k > 0$, o método é chamado k vezes, uma vez para cada posição. Finalmente é escolhido o roteiro factível de menor custo de atraso.
Desenvolvimento de métodos para a projeção de soluções.	Média	A posição, carga e roteiro dos veículos é função do tempo. Foi implementado um método que tira uma “fotografia” do banco de dados, armazena esta foto numa solução com estrutura em C++, e projeta esta solução por <i>delta_tempo</i> segundos no futuro.
Restrição de movimentos que envolvam clientes já visitados (roteiros passados não podem ser modificados).	Média	O problema dinâmico não admite que clientes já visitados sejam movidos de posição no roteiro. Foi necessário adaptar o método que verifica a factibilidade de movimentos.
Permitir / proibir movimentos que envolvam desvios	Média	Foi criado um método que identifica, para cada movimento, se o roteiro resultante acarreta em desvio. Quando ocorre o desvio, a distância entre dois clientes deixa de ser a distância euclidiana entre os mesmos, passando a ser a soma da distância percorrida até o momento, com a perda causada pelos 5 minutos necessários para o veículo mudar seu destino atual, com a distância entre a posição do veículo e a do novo destino.
Implementação da heurística vizinho mais próximo para comparar seu desempenho com o dos métodos adaptados	Baixa	A implementação da heurística vizinho mais próximo foi relativamente simples, indicando que é uma rápida alternativa para avaliar o desempenho de métodos mais sofisticados.

Um fator que modificou (e dificultou) significativamente a execução de testes dos métodos adaptados foi a duração de cada teste: 9 horas. Este tempo de simulação, além de inibir a

depuração do código passo-a-passo, aumenta o tempo para se descobrir se as alterações nos métodos produziram os efeitos desejados.

3.5.2. Adaptações da heurística construtiva

A heurística construtiva que fornece a solução de partida (solução com os roteiros dos pedidos estáticos) aos métodos *Tabu_LP_4_fcinv_ID_dinâmico*, *Inserção* e *Inserção+* é a *RR*. Escolheu-se a heurística construtiva *RR* porque esta possui maior flexibilidade para ser adaptada ao problema dinâmico.

Inicialmente foram feitos testes com a heurística construtiva do Solomon (1987) e a solução de partida, por exemplo, para o problema teste A, fica com 14 veículos. Cada um dos 14 veículos atende aproximadamente 19 clientes. Com a chegada de pedidos, novos veículos são ejetados do depósito para atendê-los. Porém, quanto mais tarde o veículo é ejetado, menos clientes são atendidos por veículo, resultando num baixo aproveitamento, como mostra a Figura 3.13.

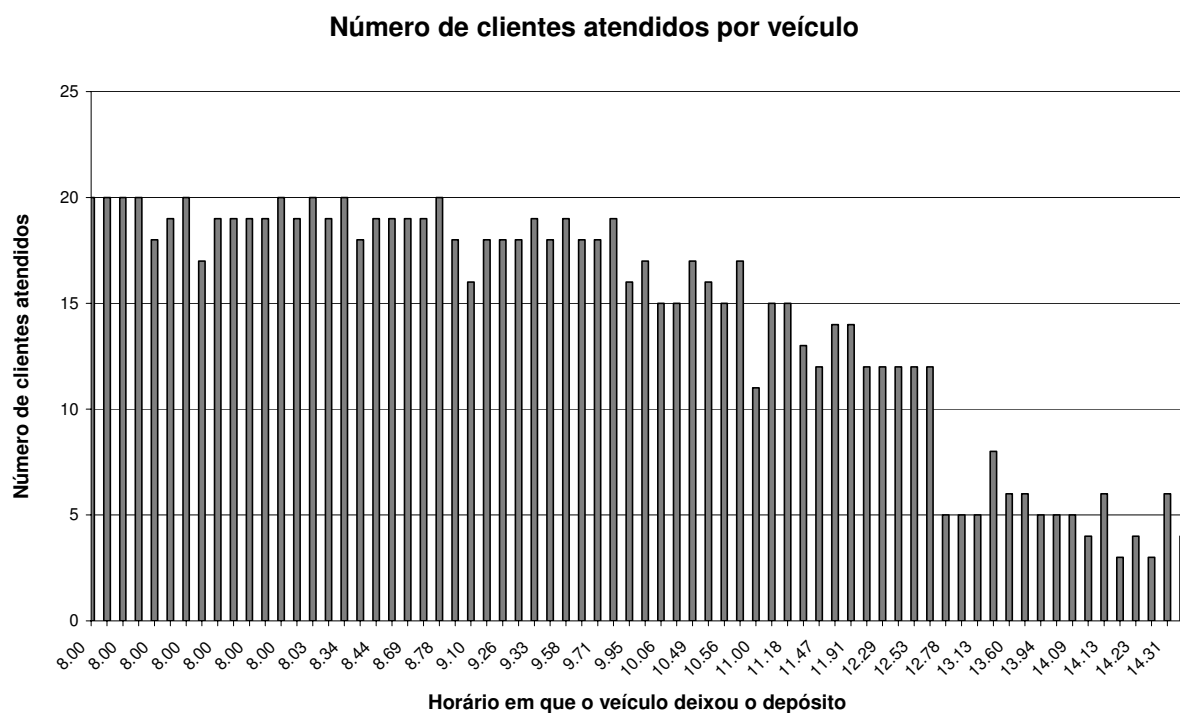


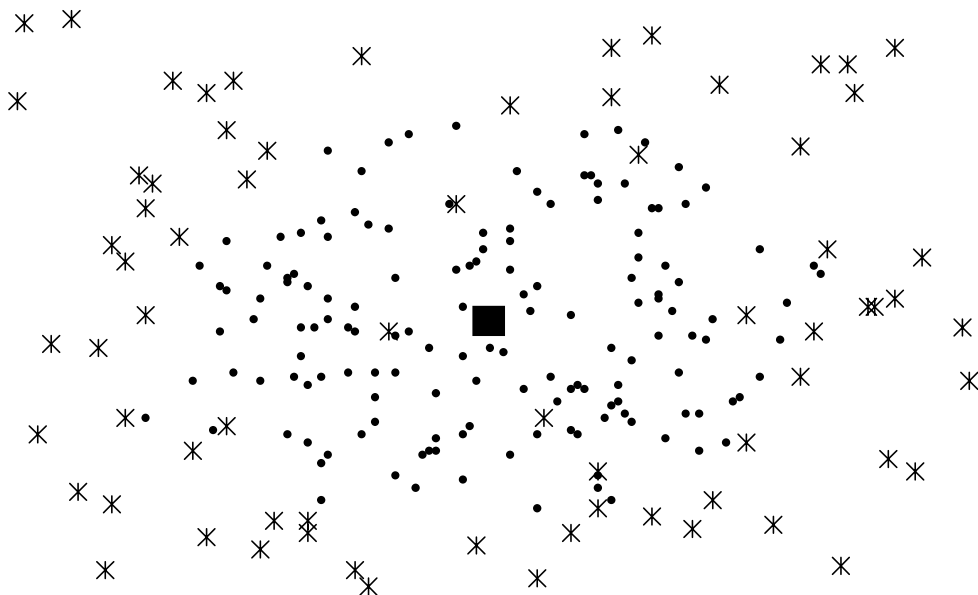
Figura 3.13. Número de clientes atendidos por veículo, método inserção, problema teste A

Como um veículo ejetado às 8hs possui aproximadamente o mesmo custo que outro ejetado às 15hs, concluiu-se que a melhor estratégia é calcular, numa etapa de planejamento, quantos

veículos são em média necessários para atender a demanda típica de um dia. Uma vez determinado este número, todos os veículos devem ser ejetados no momento em que o depósito abre (8hs). Os resultados dos testes preliminares mostraram que adotar esta estratégia pode ser até duas vezes mais lucrativa do que ejetar veículos ao longo do dia.

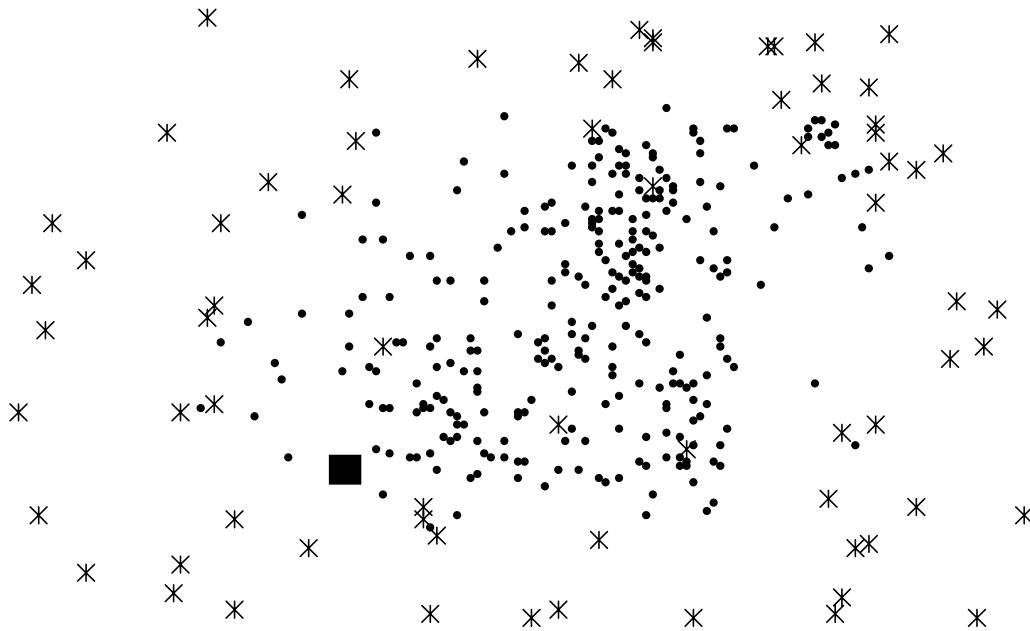
Para descobrir quantos veículos são necessários para atender toda a demanda, foram executados os testes do limitante superior (LS_{BL} e LS_{BT}). O resultado é que os 70 veículos disponíveis não são suficientes para atender todos os pedidos do dia, o que é razoável uma vez que, na prática, a empresa não é obrigada a atender os pedidos que chegaram após as 14hs. Logo, foi decidido por ejetar todos os 70 veículos às 8hs.

Então, a heurística construtiva RR teve que ser modificada para encontrar 70 sementes num conjunto de 200 a 300 clientes. Caso a heurística fosse aplicada da forma tradicional, os clientes-semente estariam localizados na periferia do espaço, como mostram as Figuras 3.14 e 3.15.



Legenda: ✕ clientes-semente, ● clientes a serem inseridos nas rotas sementes, ■ depósito

Figura 3.14. Disposição dos clientes-semente – RR original – Problema teste A



Legenda: * clientes-semente, ● clientes a serem inseridos nas rotas sementes, ■ depósito

Figura 3.15. Disposição dos clientes-semente – RR original – Problema teste B

Esta disposição dos clientes-semente não é a ideal para a solução de partida do problema dinâmico, pois há muitas sementes para poucos clientes (3 a 4 clientes por semente) e, principalmente para os problemas teste B e C, existem regiões com baixa densidade de clientes que contém muitas sementes. Por exemplo, observando o lado esquerdo da Figura 3.15, nota-se uma concentração de sementes (rotas) para poucos clientes na região.

Uma partida mais apropriada seria posicionar as sementes de forma espalhada e mais próxima das regiões de maior densidade de clientes, aumentando, de uma forma geral, a utilização dos veículos e conseqüentemente o número de clientes atendidos.

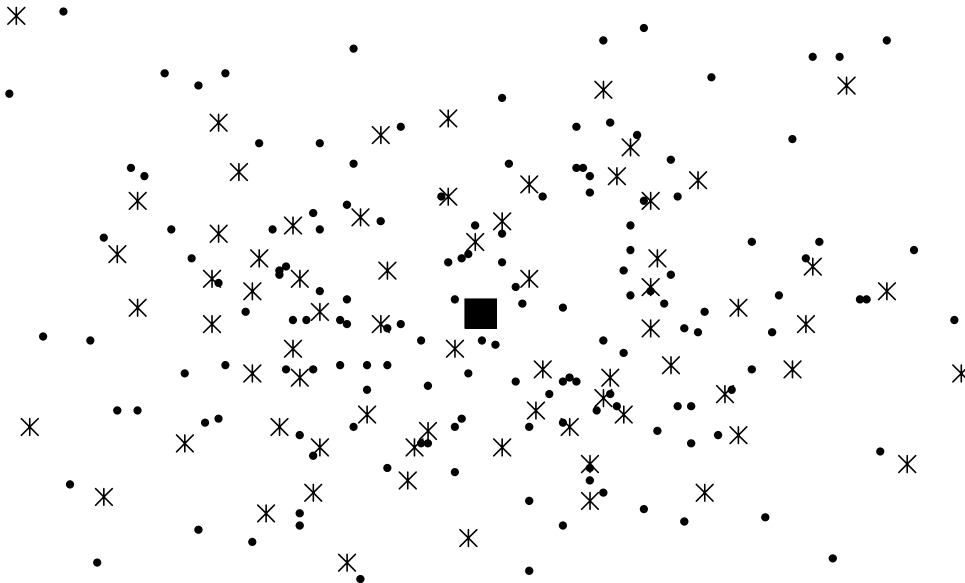
Para tanto, o procedimento *Encontrar_semente()* da heurística construtiva RR foi modificado. Ao invés de escolher como semente o cliente de maior *separação_final_i*, a seleção é feita da seguinte forma:

1. Ordene decrescentemente os clientes por *separação_final_i* e armazene-os numa lista;

2. Exclua da lista todos os clientes i que estão entre os l clientes mais próximos de qualquer semente já selecionada. Este critério busca manter um espalhamento mínimo entre a localização das sementes;
3. Selecione, dentre os m primeiros clientes da lista, o cliente k que possui a menor soma de distâncias entre k e os n clientes mais próximos de k . Note que o cliente com a menor soma de distâncias está, provavelmente, numa região de alta densidade de clientes;
4. Defina o cliente k como nova semente. Se houver outras sementes a serem selecionadas, recalcule $separação_final_i$ para cada cliente i que não é semente e retorne ao passo 1.

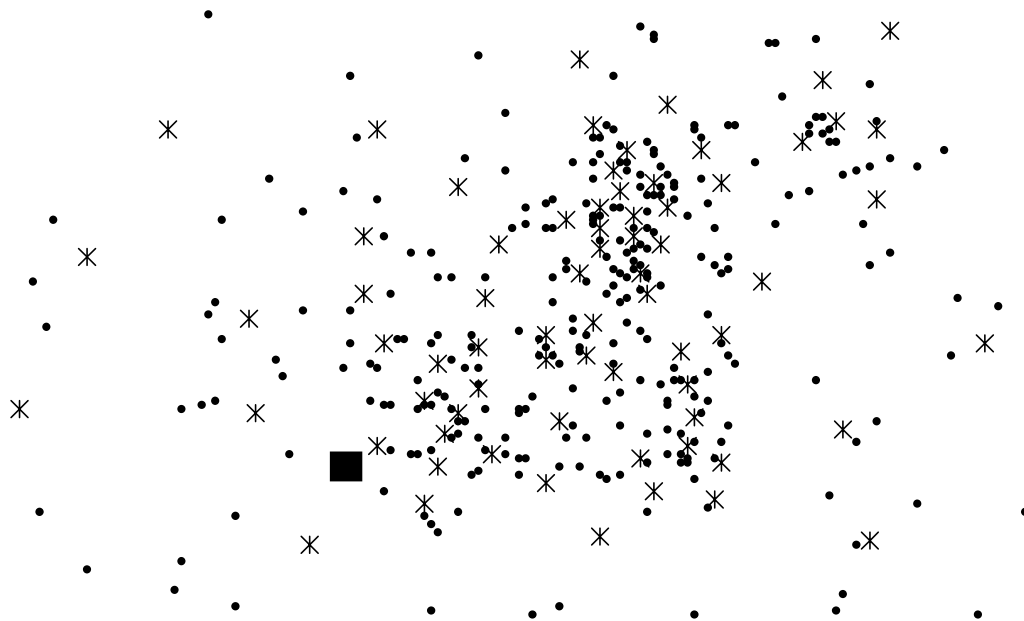
Após testes computacionais e análise visual das soluções obtidas pela heurística adaptada, os parâmetros l , m e n foram definidos em 4, 50 e 30 respectivamente.

Com estas modificações, a disposição dos clientes-semente fica como mostram as Figuras 3.16 e 3.17.



Legenda: * clientes-semente, ● clientes a serem inseridos nas rotas sementes, ■ depósito

Figura 3.16. Disposição dos clientes-semente – RR modificado – Problema teste A



Legenda: * clientes-semente, ● clientes a serem inseridos nas rotas sementes, ■ depósito

Figura 3.17. Disposição dos clientes-semente – RR modificado – Problema teste B

Uma vez encontradas as sementes, a heurística RR segue normalmente os procedimentos descritos na seção 2.3.3.

3.5.3. Adaptações da Busca Local Granular

Ao tratar o problema dinâmico, descobriu-se que a maior dimensão do problema (o tamanho do problema aumentou de 200 clientes no problema estático para 1.000 clientes no problema dinâmico) e o custo computacional dos controles para manter o sincronismo entre as soluções do banco de dados e as soluções em C++ elevam em muito o tempo computacional para se explorar a vizinhança granular.

Foi calculado o tempo de uma iteração do teste *LS_{BL}*, o qual considera os 1.101 clientes como conhecidos no momento em que o depósito abre, para diferentes tamanhos de grafo granular. O resultado é mostrado na Figura 3.18.



Figura 3.18. Tempo para percorrer uma iteração da LS_BL – Problema teste A

O tempo computacional de uma iteração segue proporcionalmente o tamanho do grafo, chegando a 16 e 32 minutos para um grafo com $\eta = 25,6\%$ e $\eta = 51,2\%$ respectivamente.

Outra constatação importante é que o valor do ganho de cada movimento da busca local é geralmente abaixo de R\$ 2,00, requerendo um grande número de iterações para que o método produza bons resultados. A Figura 3.19 apresenta o número de movimentos realizados por faixa de ganho até a busca local ($\eta = 25\%$) atingir o máximo local.

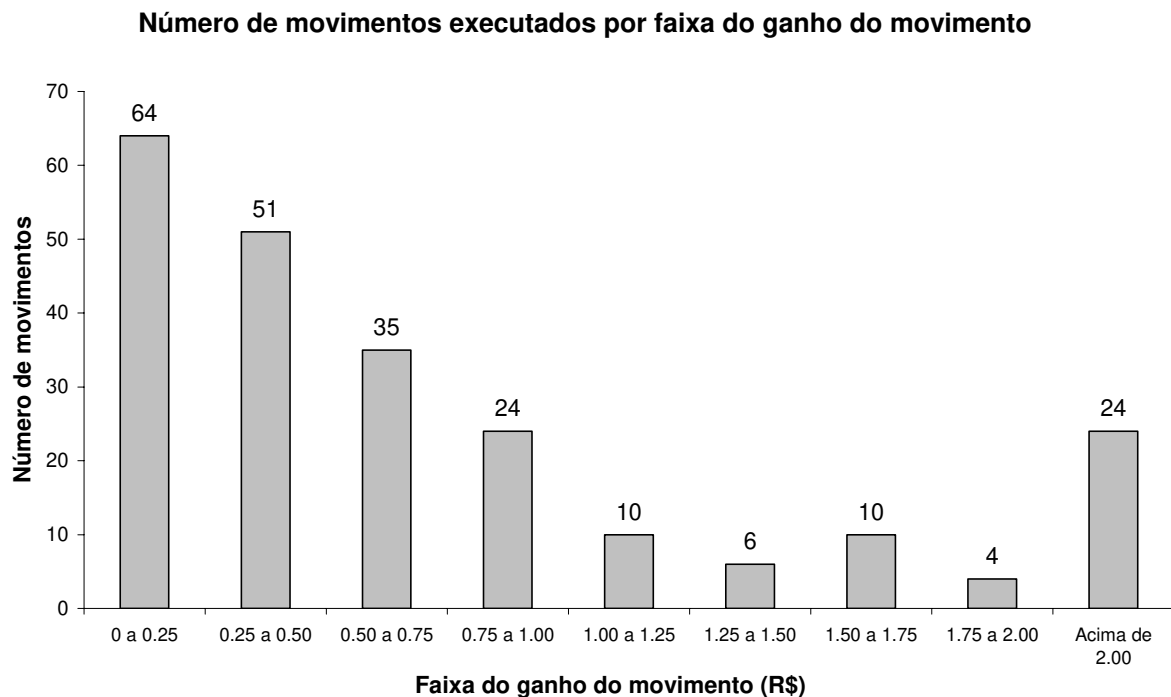


Figura 3.19. Número de movimentos por faixa de ganho – Problema teste A

A partir destas duas informações – tempo para a busca efetuar uma iteração e o valor médio dos ganhos – inferiu-se que fazer muitas iterações em grafos de grande dimensão não implica que ganhos de maior valor serão encontrados, ou que a busca atingirá mais rapidamente um máximo local de melhor qualidade. Então, foram feitos os seguintes testes para o problema A, considerando todos os 1.101 clientes conhecidos a priori (limitante superior do lucro):

- Busca local com tamanho do grafo fixo em $\eta = 25\%$: Máximo local de R\$ 94.881 atingido após 2,84 dias;
- Busca local com tamanho de grafo crescente: A busca inicia com $\eta = 0,5\%$ e somente realiza movimentos que geram ganhos superiores a R\$ 0,50. Quando nenhum movimento produz ganho superior a R\$ 0,50, o tamanho do grafo é dobrado até o limite de $\eta = 25\%$. Finalmente, quando o grafo estiver com $\eta = 25\%$ e nenhum movimento produzir ganhos superiores a R\$ 0,50, qualquer movimento que produza ganho não-negativo é permitido: Máximo local de R\$ 94.878 atingido após 1,74 dias.

A partir dos resultados, conclui-se que adotar uma busca com tamanho de grafo crescente possui tempo de execução mais rápido com perda de qualidade insignificante. Logo, a busca local

para o teste *LS_BL* foi adaptada para que o tamanho do grafo cresça gradualmente. As Figuras 3.20 e 3.21 mostram o comportamento dessa busca local adaptada para o problema teste C.

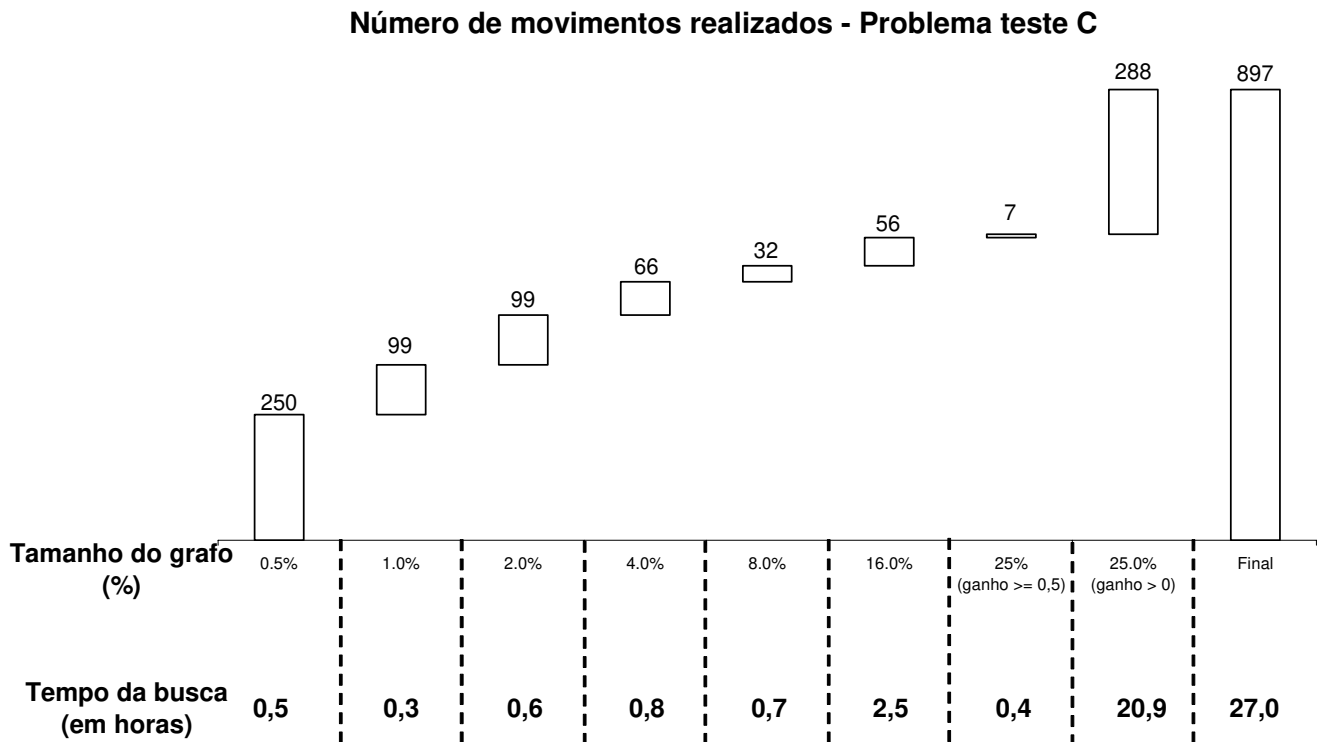


Figura 3.20. Número de movimentos por tamanho do grafo – Problema teste C

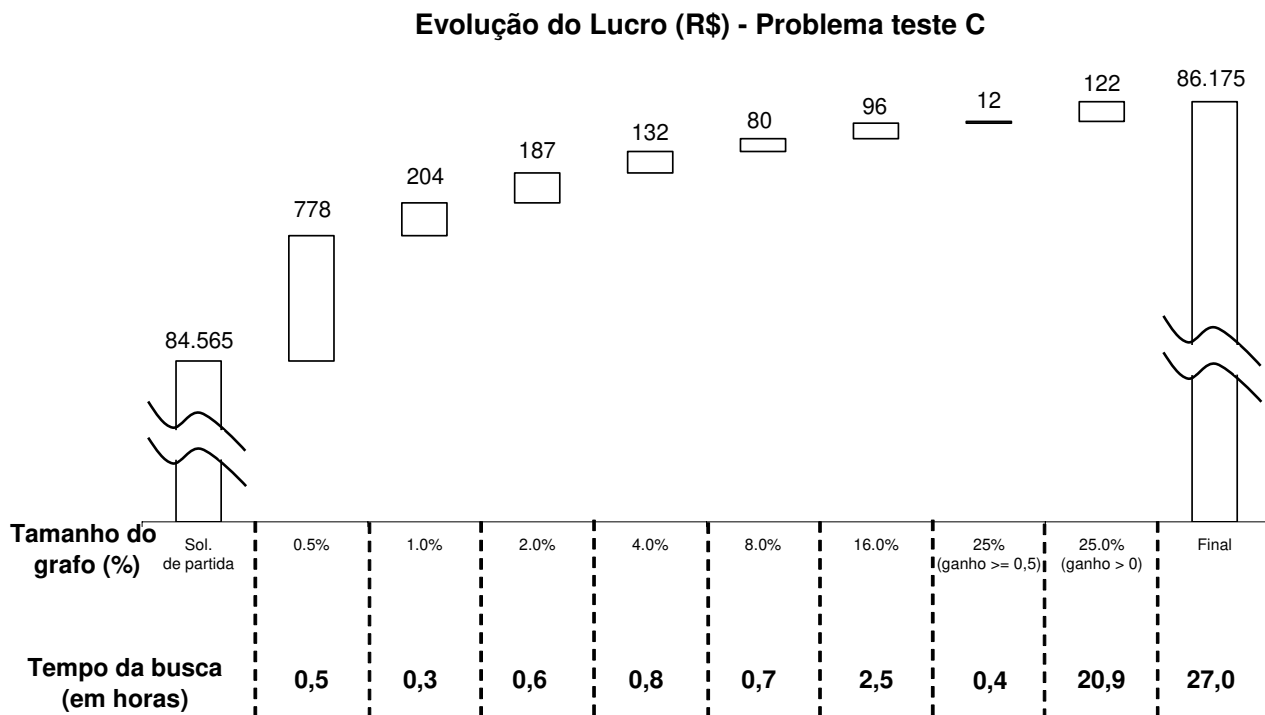


Figura 3.21. Acréscimo ao lucro por tamanho do grafo – Problema teste C

Observando ambos os gráficos, percebe-se que em menos de uma hora de execução, $\eta \in \{0,5\%, 1,0\%\}$, foi executado 39% do número total de movimentos gerando 61% do ganho total do lucro. Quando são somados os números de movimentos e ganhos até $\eta = 25\%$, considerando somente os movimentos cujos ganhos são superiores a R\$ 0,50, tem-se um tempo de execução de 5,7 horas (21% do total de 27 horas), durante o qual é executado 68% dos movimentos, produzindo 92% do ganho total da busca local.

Este comportamento possui importância ainda maior para o problema dinâmico, no qual, em geral, não há tempo suficiente para que a busca atinja o máximo local. Conseqüentemente, para o método *Inserção+*, a busca local foi modificada da seguinte forma:

1. Inicie a busca com $\eta = 0,5\%$;
2. Enquanto existirem veículos percorrendo roteiros, execute o método de busca local granular e altere o tamanho do grafo (η) segundo os critérios abaixo:

- 2.1. Se a busca foi impedida de executar um movimento de melhora por 3 vezes¹⁷ → faça $\eta = \text{Máx}\{0,5\%, \eta - 1\%\}$ e permita que somente sejam executados movimentos que produzam ganhos superiores a R\$ 0,50;
- 2.2. Se o máximo local foi atingido e a busca foi impedida de executar um movimento de melhora menos do que 3 vezes → faça $\eta = \text{Min}\{100\%, \eta + 1\%\}$;
- 2.3. Se o máximo local foi atingido e $\eta = 100\%$ e a busca foi impedida de executar um movimento de melhora menos do que 3 vezes → permita que sejam executados movimentos que produzam ganhos não-negativos.

A busca é iniciada com um grafo de pequeno tamanho, 0,5%, o qual é aumentado e reduzido segundo os critérios 2.1 e 2.2. Sempre que o método é impedido por 3 vezes de efetivar um movimento de melhora, o que implica no descarte do melhor movimento encontrado, o critério 2.1 reduz o grafo em 1%, diminuindo o tempo computacional para se encontrar o movimento de melhora na próxima iteração. Quando menos do que 3 movimentos foram perdidos e o máximo local foi atingido, o critério 2.2 aumenta o grafo em 1% para que, com a maior vizinhança, novas melhorias possam ser descobertas.

Já o critério 2.3 permite a execução de movimentos com ganhos inferiores a R\$ 0,50. Esta situação somente ocorre quando o grafo está no seu tamanho máximo e nenhum movimento com ganho superior a R\$ 0,50 foi encontrado.

Esta adaptação foi fundamental para que o método *Inserção+* produzisse bons resultados. Uma das maiores dificuldades encontradas no problema dinâmico é a variação do dinamismo do problema ao longo do dia e do tempo necessário para se buscar a vizinhança. No início do turno, o dinamismo é alto porque há uma maior taxa de chegada de novos pedidos e a vizinhança requer um tempo maior para ser explorada, pois existem muitas possibilidades de movimento (poucos clientes visitados). No fim da tarde, chegam poucos clientes novos e há um menor número de movimentos viáveis porque vários clientes já foram visitados. Desta maneira, é razoável imaginar que η deve ser menor no período da manhã e que pode ser maior no período da tarde. A Figura

¹⁷ Um movimento de melhora não é executado quando: (i) algum veículo chegou ao próximo destino, ou entrou em espera, ou saiu da espera, (ii) um novo pedido chegou e este foi inserido no roteiro de algum veículo, (iii) o horário de almoço de algum veículo começou ou terminou, (iv) o roteiro de algum veículo foi alterado por algum módulo (*Callcenter*, *Aceite*, *GPS*, *Coleta*, *Otimizador*).

3.22 ilustra a variação de η ao longo do dia para o método *Inserção+* aplicado ao problema teste A.

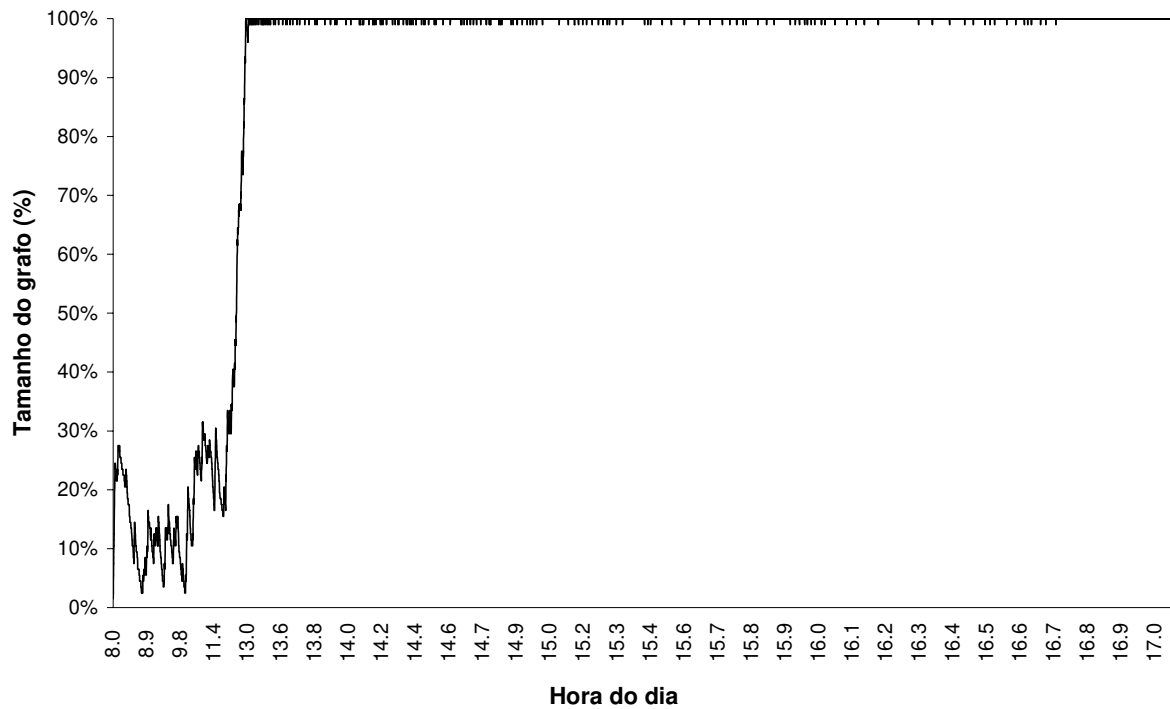


Figura 3.22. Variação de η no método *Inserção+* – Problema teste A

A Figura 3.22 mostra que até aproximadamente 12hs, η fica na faixa [0,5%, 30%], decorrente da alta taxa de novos pedidos. Neste período, o tempo de uma iteração da busca local é elevado, como mostra a Figura 3.23, uma vez que há poucos clientes visitados e conseqüentemente um maior número de movimentos factíveis a serem explorados.

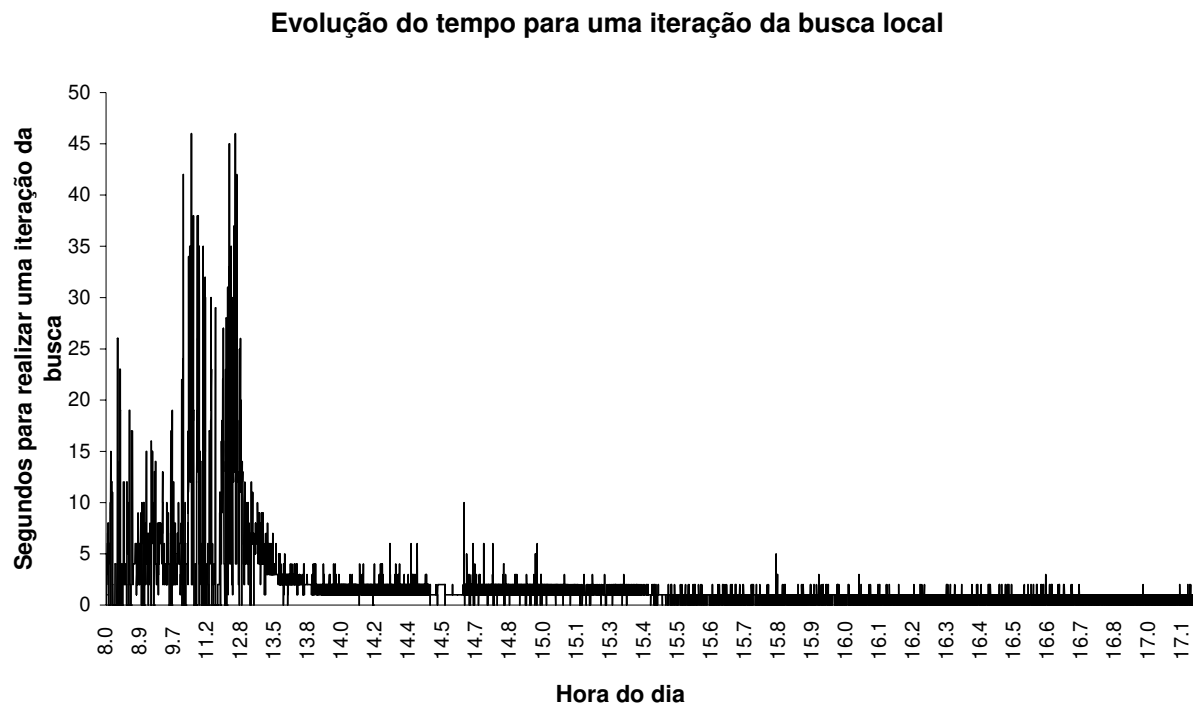


Figura 3.23. Variação do tempo para uma iteração do método *Inserção+* – Problema teste A

Neste exemplo, a vizinhança granular é explorada 13.324 vezes (iterações). O método realiza um maior número de iterações por minuto após às 13h30, como mostrado na Figura 3.24., de forma coerente com a análise das duas figuras anteriores.

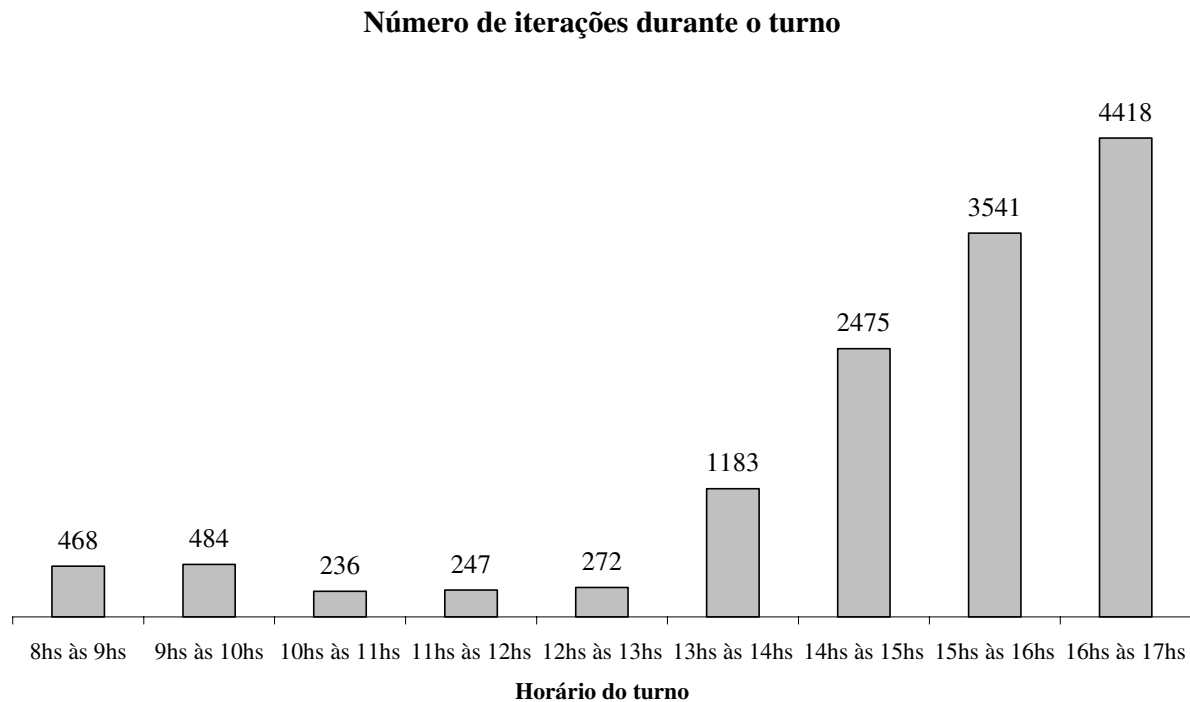


Figura 3.24. Número de iterações – método *Inserção+*, Problema teste A

Desse conjunto de iterações, o método obteve sucesso na atualização da incumbente em apenas 607 iterações, como mostra a Figura 3.25. O método não atualiza a incumbente em 12.718 iterações por dois motivos: (i) nenhum movimento de melhora foi encontrado, ou (ii) o movimento de melhora foi encontrado, porém a solução do banco de dados sofreu alguma alteração que impediu a atualização da incumbente. No início do turno predomina o segundo motivo, enquanto no final, o primeiro.

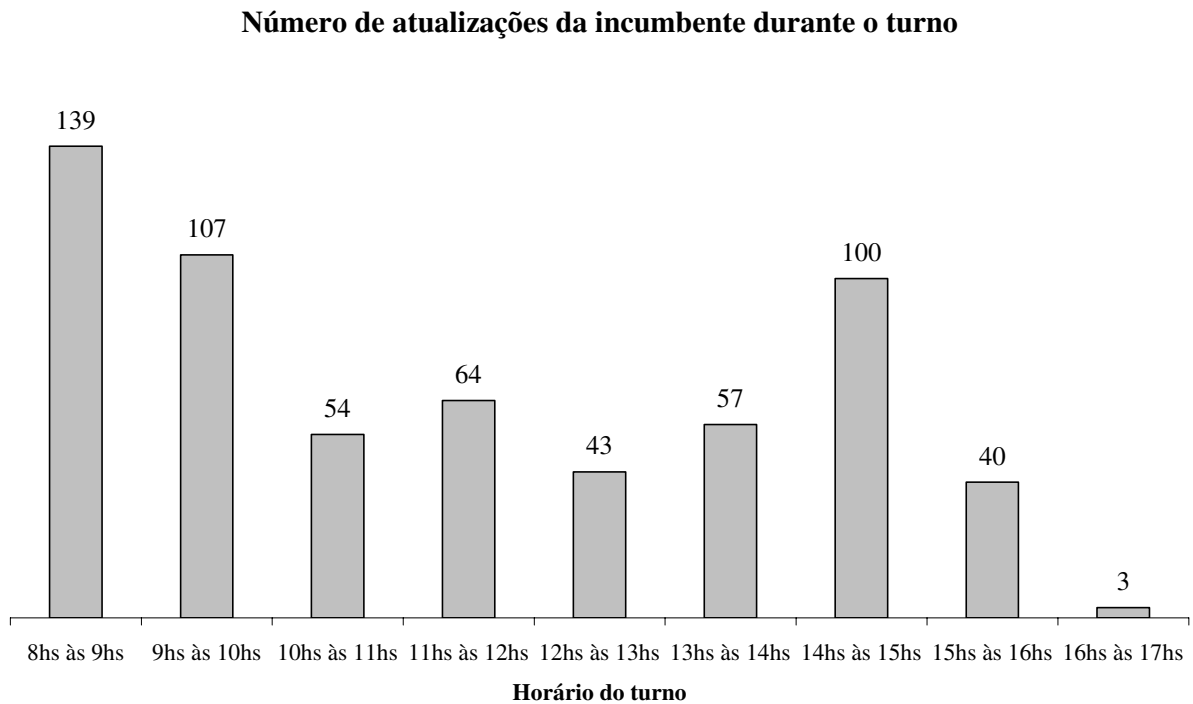


Figura 3.25. Número de atualizações da incumbente – método *Inserção+*, Problema teste A

Uma importante observação é que a facilidade de se ajustar, ao longo do turno, o tamanho da vizinhança através de um único parâmetro η torna a busca local (e tabu) granular atraente para o problema dinâmico.

3.5.4. Adaptações da Busca Tabu Granular

A busca tabu sofreu modificações semelhantes à busca local granular, como descrito a seguir:

1. Inicie a busca com $\eta = 1\%$;
2. Enquanto existirem veículos percorrendo roteiros, execute o método de busca tabu granular e altere o tamanho do grafo (η) segundo os critérios abaixo:
 - 2.1. Se a busca tabu foi impedida de realizar um movimento (melhora ou piora) em virtude de alguma alteração na situação da solução do banco de dados e foram realizados pelo menos *num_mov_superior* movimentos desde a última atualização de $\eta \rightarrow$ faça $\eta = \text{Min}\{100\%, \eta + 1\%\}$ e reinicie a busca;

2.2. Se a busca tabu foi impedida de realizar um movimento (melhora ou piora) em virtude de alguma alteração na situação da solução do banco de dados e foram realizados no máximo $num_mov_inferior$ movimentos desde a última atualização de $\eta \rightarrow$ faça $\eta = \text{Máx}\{1\%, \eta - 1\%\}$ e reinicie a busca;

2.3. Se a busca tabu realizou num_mov_max movimentos (melhora ou piora) desde a última atualização de $\eta \rightarrow$ faça $\eta = \text{Min}\{100\%, \eta + 1\%\}$ e reinicie a busca.

Após testes computacionais, os parâmetros $num_mov_superior$, $num_mov_inferior$ e num_mov_max foram definidos para 11, 4 e 56, respectivamente. Os valores de $num_mov_inferior$ e $num_mov_superior$ ajustam dinamicamente o tamanho do grafo para que o número de movimentos executados pela busca tabu fique, em geral, entre 5 e 10 movimentos. Isto é importante para evitar duas situações:

- a) Execução de muitos movimentos num grafo de tamanho pequeno, situação evitada pelo critério 2.1. Os testes demonstram que quanto maior o tamanho do grafo, maior a chance da busca tabu encontrar um movimento que atualize a incumbente;
- b) Execução de poucos movimentos num grafo de tamanho grande, situação evitada pelo critério 2.2. Se a busca tabu executa poucos movimentos entre os reinícios, um baixo número de movimentos de degradação da função objetivo é realizado, tornando o comportamento da busca tabu semelhante ao de uma busca local.

Finalmente, o critério 2.3 aumenta o tamanho do grafo sempre que pelo menos um ciclo foi completado (56 movimentos equivalem a percorrer um ciclo completo mais a etapa A do segundo ciclo – vide seção 2.5.2), com motivação semelhante à do critério 2.1.

Outras observações:

- Em qualquer reinício nenhuma das duas memórias – curto e longo prazo – são zeradas;
- O limitante inferior de 1% para η na busca tabu é maior que o limitante de 0,5% para a busca local porque um grafo de apenas 0,5% possui, em geral, poucos movimentos factíveis e não tabu.

3.5.5. Inclusão das estratégias de espera e posicionamento

Uma distinção fundamental da otimização de problemas dinâmicos para problemas estáticos é que somente a aplicação de métodos heurísticos nos problemas dinâmicos não garante que bons resultados serão atingidos. Nesta dissertação, os métodos heurísticos foram combinados com estratégias de espera e posicionamento.

A motivação da estratégia de espera é postergar ao máximo a decisão sobre o próximo cliente a ser visitado. Isto é interessante porque confere maior flexibilidade para a inserção de novos clientes, além de oferecer maior tempo computacional para os métodos de otimização.

A estratégia de posicionamento proposta é construída sobre a estratégia de espera. Se um veículo já visitou o último cliente de seu roteiro e está aguardando novas instruções antes de finalmente ir ao depósito, é desejável que este veículo aguarde numa região em que há maior probabilidade de novos pedidos aparecerem. Este tipo de estratégia é mais eficaz nos problemas teste B e C, nos quais existe uma diferença significativa entre o número de clientes por região. A Figura 3.26 mostra as posições, considerando todo o histórico do dia, em que os veículos ficaram em espera quando o método Inserção+ foi aplicado ao problema teste B, sem estratégia de posicionamento.

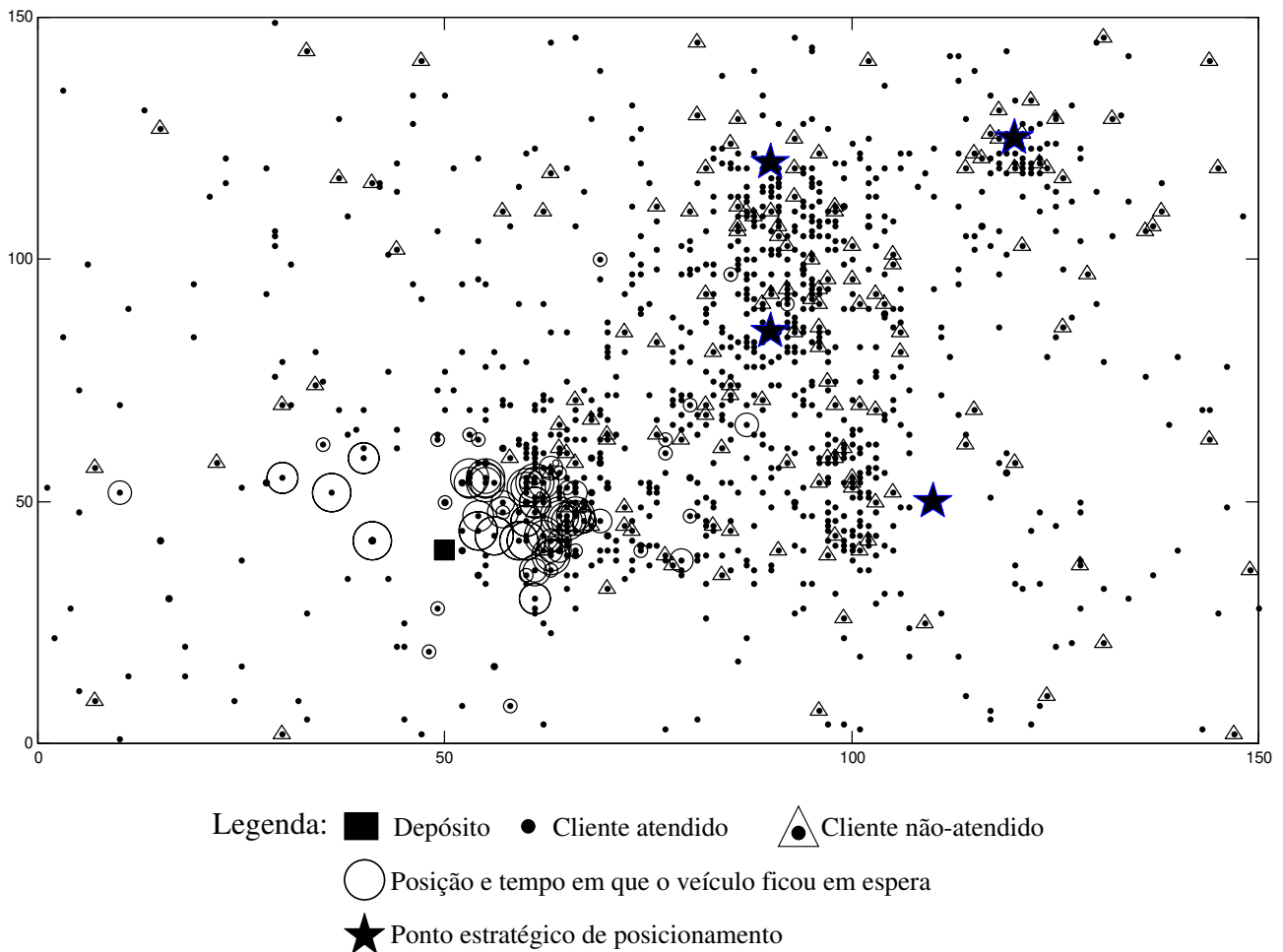


Figura 3.26. Veículos em espera e clientes recusados – método *Inserção+*, Problema teste B

Na figura, o local em que o veículo ficou em espera é o centro do círculo, sendo o tempo de espera proporcional ao seu diâmetro. A figura mostra uma concentração de veículos em espera próxima ao depósito, região em que há poucos clientes para muitos veículos. A figura também exibe vários clientes não atendidos nas regiões centrais e do lado direito. Neste caso, o objetivo da estratégia de posicionamento é mover os veículos da região próxima ao depósito para os quatro pontos marcados por estrelas, o que potencialmente aumenta o número de clientes atendidos.

Considerando que é possível para uma empresa de transporte em operação há alguns anos mapear as regiões de maior concentração de clientes, é razoável assumir que pelo menos quatro pontos estratégicos possam ser definidos. Além disso, a escolha dos pontos estratégicos é facilitada quando o espaço está segmentado em regiões com veículos dedicados, prática atual das empresas de transportes.

A estratégia de posicionamento pode ser interpretada como uma aposta. Por um lado, o veículo paga o custo de deslocamento do ponto atual até algum ponto estratégico. Por outro, há a esperança de coletar, durante o percurso ou no ponto estratégico, um novo cliente com receita superior ao custo de deslocamento.

A partir destas considerações, a estratégia de posicionamento é proposta da seguinte forma:

1. Crie uma lista com todos os veículos em espera, com ocupação inferior a 1.650 unidades de demanda e que possuam como próximo destino o depósito;
2. Selecione dentre os quatro pontos estratégicos aquele para o qual foi enviado o menor número de veículos (a estratégia de posicionamento sempre armazena o número de veículos reposicionados a cada ponto estratégico);
3. Calcule para cada veículo i da lista a $folga_i$:

$$folga_i = l_{n+1} - (hora_atual + tempo_{p_i, pe} + tempo_{pe, 0}), \quad \text{em que}$$

$$tempo_{i,j} = \text{tempo para ir do ponto } i \text{ ao ponto } j$$

$$p_i = \text{posição atual do veículo } i$$

$$pe = \text{posição do ponto estratégico escolhido}$$

$$0 = \text{posição do depósito}$$

4. Encontre o veículo i^* de maior $folga_i$;
5. Se $folga_{i^*} \geq 30$ minutos, então envie a instrução para o veículo i^* se mover de p_{i^*} para pe e remova o veículo i^* da lista. Se a lista estiver vazia $\rightarrow$ FIM. Senão, volte ao passo 2;
6. Se todos os pontos estratégicos já foram selecionados $\rightarrow$ FIM. Senão, selecione outro ponto estratégico e volte ao passo 3.

No passo 1 somente são selecionados veículos com ocupação inferior a 1.650 unidades de demanda para que pelo menos um novo cliente possa ser atendido sem violar a capacidade do veículo (a demanda média de um cliente é 100 unidades). O passo 2 busca balancear a distribuição do número de veículos pelos pontos estratégicos. O passo 3 é um critério para escolher o veículo mais adequado a ser enviado ao ponto estratégico. A medida $folga_i$ representa quanto tempo restaria se o veículo partisse do ponto atual, visitasse o ponto estratégico e retornasse imediatamente ao depósito. Se $folga_{i^*} < 0$, é inviável para o veículo i^* realizar este percurso. E quanto maior $folga_{i^*}$, maior a probabilidade do veículo i^* ser capaz de atender um

novo cliente, o qual pode surgir durante o percurso do veículo i^* , ou enquanto i^* espera no ponto estratégico. Foi colocada a restrição $folga_{i^*} \geq 30$ minutos para evitar o custo de deslocamento de veículos que, por possuírem pouca folga, têm menos flexibilidade para coletar novos clientes (lembramos que o tempo de serviço médio de um cliente é de 10 minutos).

A Figura 3.27 demonstra a trajetória dos veículos em direção aos pontos estratégicos para o mesmo caso da Figura 3.26.

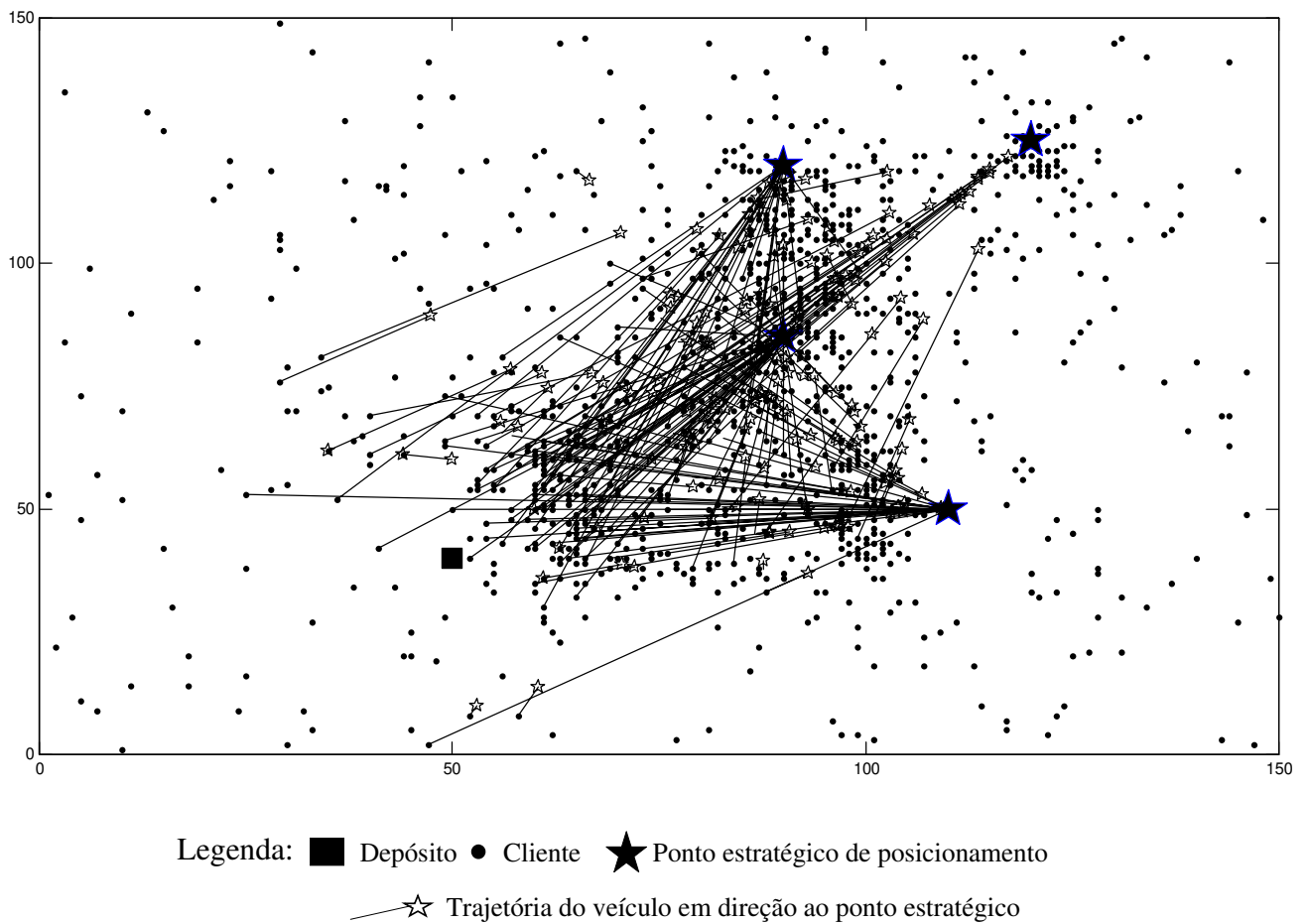


Figura 3.27. Veículos em espera e clientes recusados – método $I+_e_p$, Problema teste B

A Figura 3.27 mostra que vários veículos são deslocados do quadrante inferior esquerdo para os pontos estratégicos e que o número de veículos que chegam a cada ponto estratégico é balanceado. Há também trajetórias que são interrompidas antes do veículo atingir o ponto estratégico, por exemplo, algumas trajetórias em direção ao ponto do quadrante superior direito. Isto representa que um novo cliente foi alocado ao veículo durante seu percurso.

Aparentemente, há um elevado custo de deslocamento de veículos para os pontos estratégicos. Contudo, como será mostrado no item a seguir, este custo é compensado pela receita proveniente do atendimento de novos clientes.

3.6. Resultados dos experimentos computacionais

Os resultados dos testes da Tabela 3.2. são apresentados nesta seção. Os testes foram executados, na sua maioria, em computadores Pentium IV 2.8 GHz com 1 Gb de RAM (sistema operacional Linux Fedora Core 2) interligados por uma rede local, como mostra a Figura 3.28.

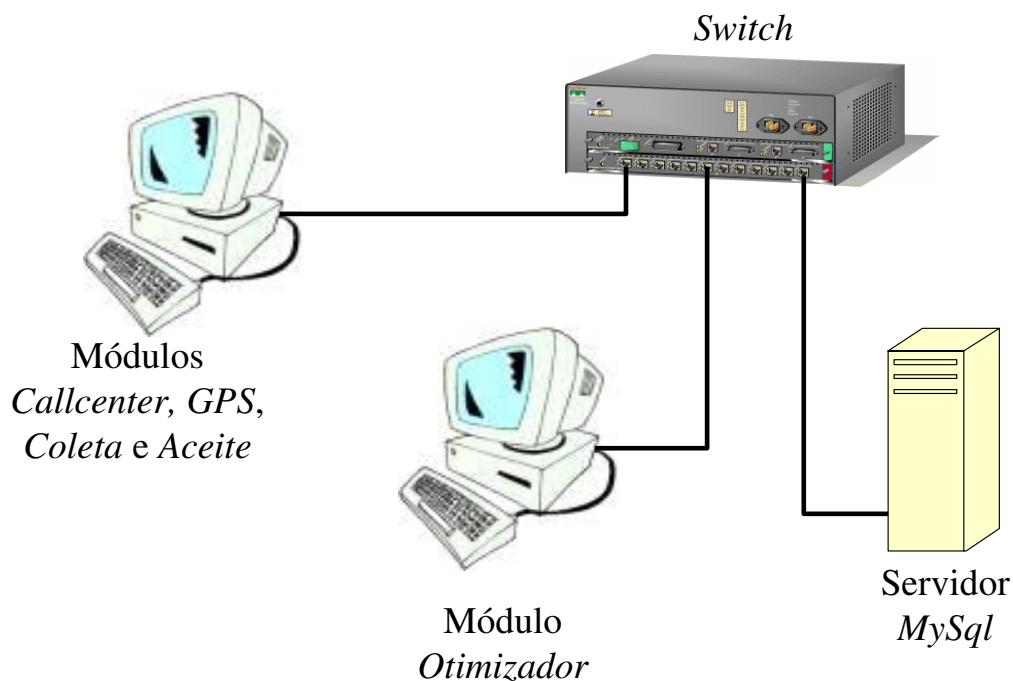


Figura 3.28. Infra-estrutura utilizada na execução dos testes para o PRDVJT

A implementação permite que cada módulo seja executado num computador distinto. Todavia, em virtude do grande número de testes a serem executados e do elevado tempo de execução de cada teste, optou-se por executar os módulos *Callcenter, GPS, Coleta e Aceite* numa máquina, e o módulo *Otimizador* (utilizado nos testes *I+* e *BT*) noutra. Esta arquitetura assume que a empresa disponibilizaria um computador dedicado à execução de processos de otimização.

3.6.1. *LS_BL* e *LS_BT*

As Tabelas 3.13, 3.14 e 3.15 apresentam os resultados para os testes *LS_BL* e *LS_BT*, utilizados para calcular o limitante superior do lucro.

Tabela 3.13. Resultados dos métodos *LS_BL* e *LS_BT* – Problema teste A

Medida	<i>LS_BL</i>	<i>LS_BT</i>
Lucro (R\$)	94.878	94.785
Número de clientes prioritários atendidos	884	884
%	100,0%	100,0%
Número de clientes não prioritários atendidos	187	187
%	86,2%	86,2%
Número de clientes recusados	30	30
%	2,7%	2,7%
Número de atualizações da incumbente	864	681
Tempo até a incumbente (horas)	41,6	4,6
Tempo total de execução (horas)	41,8	4,7

Tabela 3.14. Resultados dos métodos *LS_BL* e *LS_BT* – Problema teste B

Medida	<i>LS_BL</i>	<i>LS_BT</i>
Lucro (R\$)	84.407	84.644
Número de clientes prioritários atendidos	713	713
%	100,0%	100,0%
Número de clientes não prioritários atendidos	253	253
%	72,9%	72,9%
Número de clientes recusados	94	94
%	8,9%	8,9%
Número de atualizações da incumbente	893	733
Tempo até a incumbente (horas)	26,6	4,5
Tempo total de execução (horas)	26,6	4,7

Tabela 3.15. Resultados dos métodos *LS_BL* e *LS_BT* – Problema teste C

Medida	<i>LS_BL</i>	<i>LS_BT</i>
Lucro (R\$)	86.175	85.908
Número de clientes prioritários atendidos	714	714
%	100,0%	100,0%
Número de clientes não prioritários atendidos	265	265
%	72,4%	72,4%
Número de clientes recusados	101	101
%	9,4%	9,4%
Número de atualizações da incumbente	896	731
Tempo até a incumbente (horas)	26,6	4,4
Tempo total de execução (horas)	26,7	4,7

O método *LS_BL* obteve resultados ligeiramente superiores ao método *LS_BT*, atualizando a incumbente um maior número de vezes. Isto ocorreu porque a heurística *LS_BL* foi executada até atingir o máximo local da vizinhança com grafo crescente até $\eta = 25\%$. O tempo computacional para a busca atingir o máximo local foi de aproximadamente 42 horas para o problema teste A e 27 horas para os problemas B e C.

O método *LS_BT* foi executado com $\eta = 1,5\%$ (fixo) e teve como limite de tempo computacional aproximadamente 5 horas. Seu desvio relativo ao lucro encontrado por *LS_BL* foi de -0,1%, 0,3% e -0,3% para os problemas testes A, B e C, respectivamente. Se o tempo de execução da heurística *LS_BL* fosse limitado ao mesmo tempo de *LS_BT*, o desempenho de *LS_BT* é um pouco melhor que o de *LS_BL*, com desvio relativo de 0,0%, 0,4% e -0,1% para cada problema teste.

Os números de clientes atendidos e recusados são idênticos para ambos os métodos porque a decisão sobre quais clientes são atendidos no dia é feita pela heurística construtiva RR na sua versão original. O percentual de clientes recusados é aproximadamente 3% para o problema teste A e 9% para os problemas B e C.

Para comparar o desempenho das heurísticas seguintes, o maior lucro entre os obtidos por *LS_BL* e *LS_BT*, para cada problema teste, é utilizado como limitante superior.

3.6.2. Vizinho mais próximo

A heurística do vizinho mais próximo foi testada sem estratégias de espera e posicionamento (*VMP*), com estratégia de espera (*VMP_e*) e com ambas as estratégias (*VMP_e_p*). Os resultados são mostrados nas Tabelas 3.16, 3.17 e 3.18.

Tabela 3.16. Resultados dos métodos *VMP*, *VMP_e* e *VMP_e_p* – Problema teste A

Medida	<i>VMP</i>	<i>VMP_e</i>	<i>VMP_e_p</i>
Lucro (R\$)	81.721	87.070	85.056
Desvio % do limitante superior	-13,9%	-8,2%	-10,4%
Número de clientes prioritários atendidos	878	877	878
%	99,3%	99,2%	99,3%
Número de clientes não prioritários atendidos	115	180	176
%	53,0%	82,9%	81,1%
Número de clientes recusados	108	44	47
%	9,8%	4,0%	4,3%

Tabela 3.17. Resultados dos métodos *VMP*, *VMP_e* e *VMP_e_p* – Problema teste B

Medida	<i>VMP</i>	<i>VMP_e</i>	<i>VMP_e_p</i>
Lucro (R\$)	58.295	77.602	77.460
Desvio % do limitante superior	-31,1%	-8,3%	-8,5%
Número de clientes prioritários atendidos	709	710	709
%	99,4%	99,6%	99,4%
Número de clientes não prioritários atendidos	60	264	267
%	17,3%	76,1%	76,9%
Número de clientes recusados	291	86	84
%	27,5%	8,1%	7,9%

Tabela 3.18. Resultados dos métodos *VMP*, *VMP_e* e *VMP_e_p* – Problema teste C

Medida	<i>VMP</i>	<i>VMP_e</i>	<i>VMP_e_p</i>
Lucro (R\$)	58.020	78.232	78.310
Desvio % do limitante superior	-32,7%	-9,2%	-9,1%
Número de clientes prioritários atendidos	713	713	712
%	99,9%	99,9%	99,7%
Número de clientes não prioritários atendidos	51	262	265
%	13,9%	71,6%	72,4%
Número de clientes recusados	316	105	103
%	29,3%	9,7%	9,5%

Para os problemas A e B, o melhor método foi o *VMP_e*. No problema teste C, o *VMP_e* praticamente empata com o *VMP_e_p*. O desvio % do *VMP_e* sobre o limitante superior do lucro ficou na faixa de [-9,2%, -8,2%]. O percentual de clientes recusados ficou entre 4% e 9,7%.

Apesar do bom desempenho, em nenhum problema teste a heurística do vizinho mais próximo foi capaz de atender 100% dos clientes prioritários. Isto não chega a ser um problema grave, pois o atendimento foi superior a 99,2%, mas poderia afetar a confiabilidade da empresa ao longo do tempo.

A estratégia de espera é de alta importância para a heurística do vizinho mais próximo, pois evita que veículos de pouca ocupação retornem prematuramente ao depósito. Este comportamento reflete os péssimos resultados do método *VMP*.

Contudo, a estratégia de posicionamento não traz benefícios significativos. Observando os roteiros das soluções finais, percebe-se que é inerente à heurística do vizinho mais próximo espalhar veículos de forma satisfatória, com ou sem estratégia de posicionamento – inclusive para os problemas teste B e C que possuem regiões com diferentes densidades de clientes. A Figura

3.29 mostra, também considerando todo o histórico do dia, o local em que os veículos ficaram em espera para o método *VMP_e*, problema teste B.

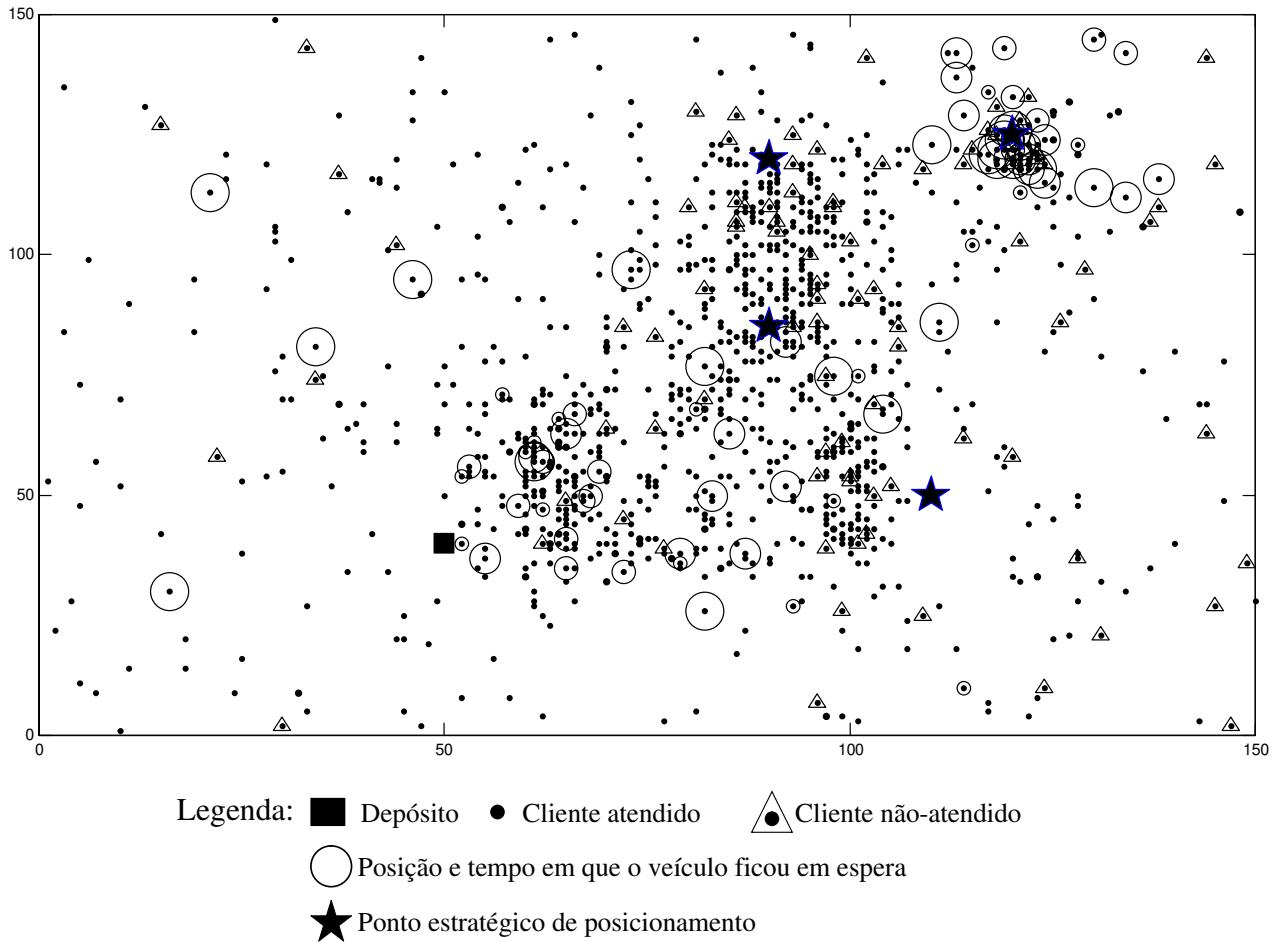


Figura 3.29. Posição e espera dos veículos, *VMP_e*, problema teste B

Diferentemente da Figura 3.26, em que praticamente todos os veículos ficam em espera próximo ao depósito, vê-se na Figura 3.29 que os veículos do método *VMP_e* ficaram em espera em diversas posições do espaço, com uma concentração sobre o ponto estratégico superior direito. Logo, o acréscimo da estratégia de posicionamento ao método *VMP_e* produz menos benefícios do que ao método *Inserção* e *Inserção+*, como visto a seguir.

3.6.3. Inserção

Os resultados do método *Inserção* são mostrados nas Tabelas 3.19, 3.20 e 3.21.

Tabela 3.19. Resultados do método *Inserção* – Problema teste A

Medida	<i>I</i>	<i>I_d</i>	<i>I_e</i>	<i>I_e_d</i>	<i>I_e_p</i>	<i>I_e_d_p</i>
Lucro (R\$)	60.229	61.801	88.847	88.405	90.626	91.018
Desvio % do limitante superior	-36,5%	-34,9%	-6,4%	-6,8%	-4,5%	-4,1%
Núm. de clientes prioritários atendidos	811	824	884	884	884	884
%	91,7%	93,2%	100,0%	100,0%	100,0%	100,0%
Núm. de clientes não prioritários atendidos	7	7	151	147	174	177
%	3,2%	3,2%	69,6%	67,7%	80,2%	81,6%
Número de clientes recusados	283	270	66	70	43	40
%	25,7%	24,5%	6,0%	6,4%	3,9%	3,6%

Tabela 3.20. Resultados do método *Inserção* – Problema teste B

Medida	<i>I</i>	<i>I_d</i>	<i>I_e</i>	<i>I_e_d</i>	<i>I_e_p</i>	<i>I_e_d_p</i>
Lucro (R\$)	63.357	62.448	71.898	71.786	80.646	80.256
Desvio % do limitante superior	-25,1%	-26,2%	-15,1%	-15,2%	-4,7%	-5,2%
Núm. de clientes prioritários atendidos	713	713	713	713	713	713
%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
Núm. de clientes não prioritários atendidos	64	54	155	155	244	241
%	18,4%	15,6%	44,7%	44,7%	70,3%	69,5%
Número de clientes recusados	283	293	192	192	103	106
%	26,7%	27,6%	18,1%	18,1%	9,7%	10,0%

Tabela 3.21. Resultados do método *Inserção* – Problema teste C

Medida	<i>I</i>	<i>I_d</i>	<i>I_e</i>	<i>I_e_d</i>	<i>I_e_p</i>	<i>I_e_d_p</i>
Lucro (R\$)	63.843	63.881	74.457	74.524	80.557	81.183
Desvio % do limitante superior	-25,9%	-25,9%	-13,6%	-13,5%	-6,5%	-5,8%
Núm. de clientes prioritários atendidos	714	714	714	714	714	714
%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
Núm. de clientes não prioritários atendidos	59	59	169	170	236	241
%	16,1%	16,1%	46,2%	46,4%	64,5%	65,8%
Número de clientes recusados	307	307	197	196	130	125
%	28,4%	28,4%	18,2%	18,1%	12,0%	11,6%

O método *I* apresentou os piores resultados, porém, mesmo assim, supera o desempenho do *VMP* nos problemas B e C.

Acrescentar a possibilidade de desvio (métodos *I_d*, *I_e_d* e *I_e_d_p*) gera pequena melhora nos problemas teste A e C – média de melhora de R\$ 376,00. Para o problema teste B, o

desvio sempre degrada o lucro. De uma forma geral, pode-se concluir que desvio para o método *Inserção* não produz vantagem significativa.

Da mesma forma do vizinho mais próximo, a aplicação da estratégia de espera ao método *Inserção* foi muito positiva. Ao comparar I_e com I , observa-se melhora média de R\$ 15.925,00.

Comparando I_e com o método VMP_e , percebe-se que I_e supera VMP_e no problema teste A, mas perde nos problemas B e C. Da análise desse resultado surgiu a idéia da estratégia de posicionamento. Quando o problema possui clientes com distribuição uniforme no espaço (A), os veículos ficam em espera de forma relativamente espalhada. No caso de concentração de clientes (B e C), isto já não ocorre na heurística *Inserção* e deve ser induzido pelos pontos estratégicos.

As versões com estratégia de espera e posicionamento de *Inserção* superaram os resultados do vizinho mais próximo – média de R\$ 3.288,00 sobre o lucro do VMP_e . Destas, a $I_{e_d_p}$ foi a de melhor desempenho.

Uma vantagem do método *Inserção* é que qualquer versão que possua estratégia de espera é capaz de atender 100% dos clientes prioritários.

3.6.4. *Inserção+* e *Busca Tabu*

Com a execução simultânea do módulo *Otimizador* com os outros quatro módulos, foram testadas as buscas local granular (*Inserção+*) e tabu (*BT*). Os resultados estão nas Tabelas 3.22, 3.23 e 3.24

Tabela 3.22. Resultados dos métodos *Inserção+* e *Busca Tabu* – Problema teste A

Medida	$I+_e_p$	$I+_e_d_p$	BT_e_p	$BT_e_d_p$
Lucro (R\$)	91.118	91.359	91111,4	91.333
Desvio % do limitante superior	-4,0%	-3,7%	-4,0%	-3,7%
Número de clientes prioritários atendidos	884	884	884	884
%	100,0%	100,0%	100,0%	100,0%
Número de clientes não prioritários atendidos	174	177	175	178
%	80,2%	81,6%	80,6%	82,0%
Número de clientes recusados	43	40	42	39
%	3,9%	3,6%	3,8%	3,5%

Tabela 3.23. Resultados dos métodos *Inserção+* e *Busca Tabu* – Problema teste B

Medida	<i>I+_e_p</i>	<i>I+_e_d_p</i>	<i>BT_e_p</i>	<i>BT_e_d_p</i>
Lucro (R\$)	82.201	83.122	82.472	82.466
Desvio % do limitante superior	-2,9%	-1,8%	-2,6%	-2,6%
Número de clientes prioritários atendidos	713	713	713	713
%	100,0%	100,0%	100,0%	100,0%
Número de clientes não prioritários atendidos	252	263	255	256
%	72,6%	75,8%	73,5%	73,8%
Número de clientes recusados	95	84	92	91
%	9,0%	7,9%	8,7%	8,6%

Tabela 3.24. Resultados dos métodos *Inserção+* e *Busca Tabu* – Problema teste C

Medida	<i>I+_e_p</i>	<i>I+_e_d_p</i>	<i>BT_e_p</i>	<i>BT_e_d_p</i>
Lucro (R\$)	83.309	83.550	83.601	83.484
Desvio % do limitante superior	-3,3%	-3,0%	-3,0%	-3,1%
Número de clientes prioritários atendidos	714	714	714	714
%	100,0%	100,0%	100,0%	100,0%
Número de clientes não prioritários atendidos	255	257	258	257
%	69,7%	70,2%	70,5%	70,2%
Número de clientes recusados	111	109	108	109
%	10,3%	10,1%	10,0%	10,1%

Os desempenhos dos métodos são praticamente idênticos. Na média, o método *I+_e_d_p* foi o melhor, com faixa [-3,7%, -1,8%] do limitante superior do lucro.

É interessante notar que, para o método *Inserção+*, a aplicação do desvio foi sempre vantajosa, proporcionando um acréscimo médio de R\$ 468,00. A provável justificativa para uma contribuição mais relevante do desvio ao método *Inserção+* do que ao método *Inserção* é que a busca local granular avalia um maior número de movimentos com desvio.

A busca tabu teve desempenho semelhante ao da busca local, sugerindo que o método não teve tempo computacional suficiente para tirar proveito dos componentes tabu que evitam ciclagem e buscam por outros máximos locais de melhor qualidade. Para que isso ocorra, provavelmente é preciso uma implementação mais rápida e eficiente, talvez através de computação paralela.

Por último, comparamos os métodos *LS_BL*, *VMP_e*, *I_e_d_p* e *I+_e_d_p* nas Figuras 3.20, 3.31 e 3.32.

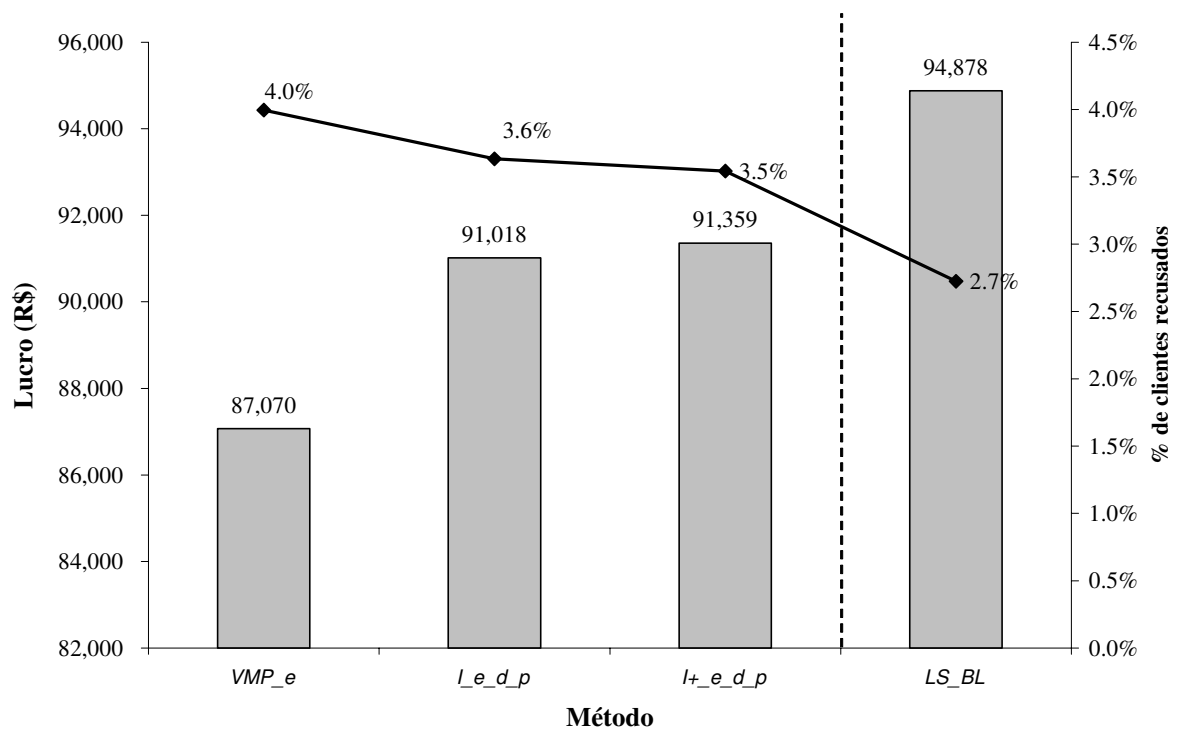


Figura 3.30. Comparação dos métodos – Problema Teste A

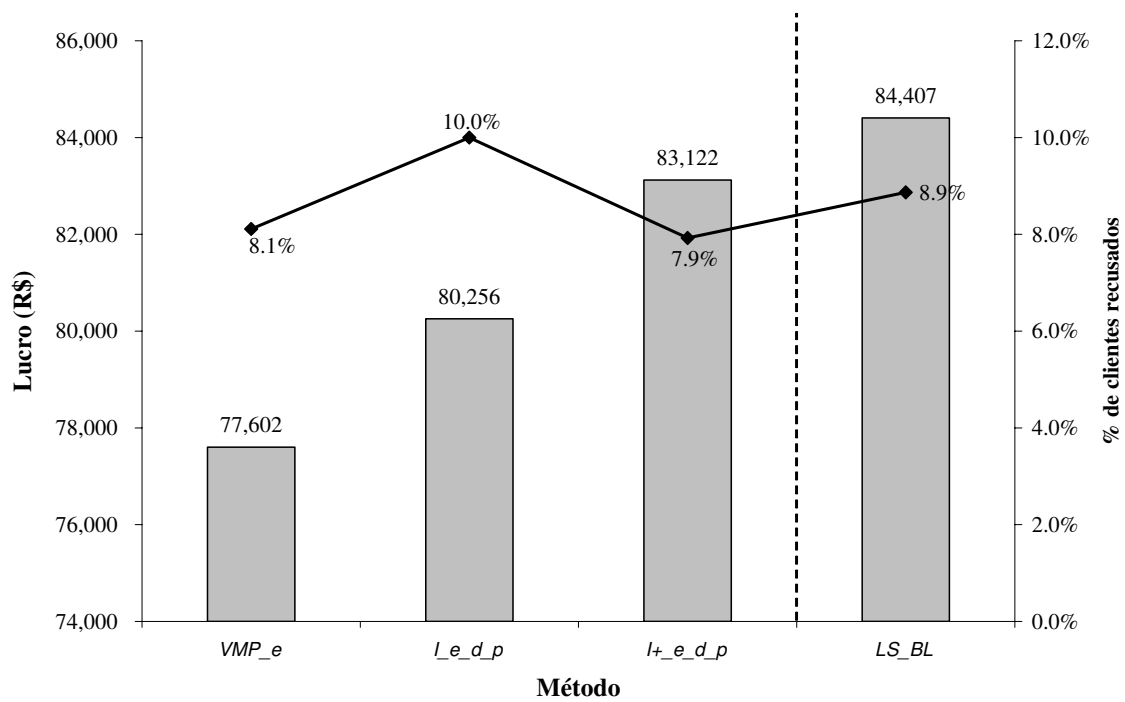


Figura 3.31. Comparação dos métodos – Problema Teste B

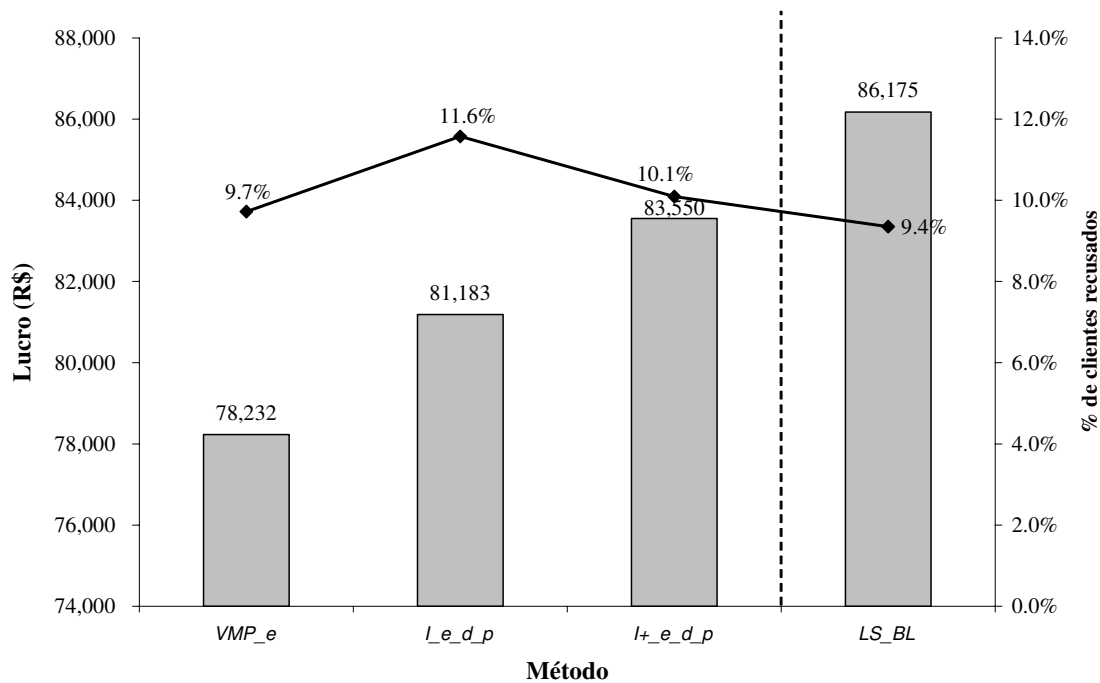


Figura 3.32. Comparação dos métodos – Problema Teste C

A Figura 3.30 mostra que, no problema teste A, o método com busca local granular ($I+_e_d_p$) não é substancialmente melhor que o método sem a busca ($I_e_d_p$), enquanto as Figuras 3.31 e 3.32 mostram que a busca local teve excelentes desempenhos nos problemas B e C. Portanto, quando a distribuição dos clientes é uniforme, apenas o método melhor inserção com estratégias de espera e posicionamento é suficiente para que bons resultados sejam atingidos.

No problema C, o percentual de clientes recusados do método $I+_e_d_p$ é um pouco superior ao do método VMP_e , o que parece um contra-senso porque o lucro produzido por $I+_e_d_p$ é maior. A explicação é que o VMP_e atende um número um pouco maior de clientes (0,4% de clientes adicionais) a um custo muito superior ao do método $I+_e_d_p$ – principalmente em virtude do custo de atraso – como mostra a Figura 3.33.

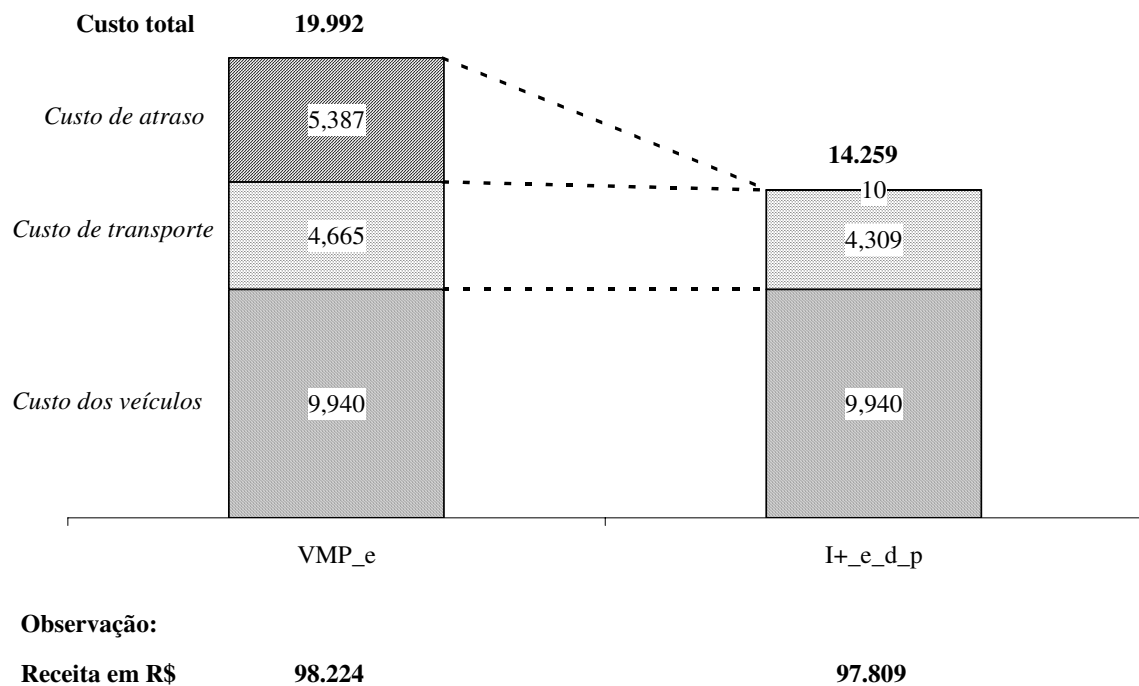


Figura 3.33. Comparação dos custos (em R\$) dos métodos VMP_e e $I+_e_d_p$, Prob. Teste C

No problema teste B, o percentual de clientes recusados de VMP_e e $I+_e_d_p$ é menor do que o de LS_{BL} , indicando que, para o cálculo do limitante superior do lucro, a decisão sobre quais clientes atender no dia não deve ser confiada somente à heurística construtiva RR , mas a um método um pouco mais sofisticado e focado a este objetivo.

3.7. Conclusões para o PRDVJT

A partir dos experimentos computacionais, as seguintes conclusões podem ser listadas para o PRDVJT:

- A implementação de métodos de otimização em tempo real é complexa: a replicação e gerenciamento de estruturas de dados e o sincronismo da atualização de soluções são atividades críticas;
- É muito importante dividir as decisões de número de veículos e otimização de distância para o problema dinâmico. Numa etapa de planejamento, estima-se quantos veículos são necessários em média por dia. Na operação, todos os veículos são ejetados no início do turno;
- A heurística construtiva deve ser personalizada ao problema dinâmico, pois, no início do turno, há poucos clientes para muitos veículos;
- A vizinhança granular é uma boa alternativa para o problema dinâmico porque permite fácil adaptação do tamanho da vizinhança ao tempo computacional disponível;
- Ao tratar a função objetivo como dinheiro, os movimentos de busca local e tabu são caracterizados por proporcionarem pequenos ganhos por movimento (abaixo de R\$ 2,00). Conseqüentemente, é interessante adotar grafos crescentes para que não sejam realizadas iterações demasiadamente demoradas para pequenos ganhos;
- A adaptação direta de heurísticas do problema estático ao problema dinâmico não resulta em bons resultados. Para obter melhor desempenho, é fundamental combinar as heurísticas com estratégias de espera e de posicionamento;
- Para o cálculo do limitante superior, ao invés de se aplicar diretamente a heurística construtiva RR, deve-se implementar um método focado na maximização do número de clientes atendidos;
- A heurística do vizinho mais próximo é uma alternativa simples e adequada ao problema dinâmico. Suas desvantagens são a falta de garantia que todos os pedidos prioritários serão atendidos e que, para cada novo cliente, é impossível informar se o mesmo será ou não atendido no mesmo dia;
- Há grandes vantagens em se acrescentar a estratégia de espera à heurística *VMP*. O mesmo não ocorre com a estratégia de posicionamento;

- A política de desvio não é importante para o método *Inserção*, mas é benéfica para o método *Inserção+*;
- Se o problema tratado tem disposição de clientes semelhante à do problema teste A, o método *Inserção* é suficiente. Caso a disposição seja como a dos problemas B e C (exemplo da Transportadora Americana), é importante adicionar o módulo *Otimizador* com a heurística de busca local granular;
- A busca tabu, ou qualquer outro método centrado em memória, deve possuir uma implementação bastante eficiente para ser competitivo nos problemas dinâmicos, pois a pouca disponibilidade de tempo computacional dificulta a criação de uma memória de qualidade a ser utilizada na procura de outros máximos locais;
- O método *I+_e_d_p* foi o que gerou os melhores resultados para os três problemas teste.

Capítulo 4. Conclusões gerais e sugestões para pesquisa futura

Uma importante lição que aprendemos ao lidar com os problemas PRVJT e PRDVTJ é o grande valor do desenvolvimento de análises (gráficos, tabelas, visualização da estrutura das soluções) para se obter um entendimento razoável do comportamento dos métodos aqui testados. Não é fácil entender porque um método teve melhor desempenho do que outro em determinado problema teste.

Muitas vezes, os diferentes comportamentos são explicados pela disposição dos clientes e veículos no espaço. No problema dinâmico, esta situação é complexa uma vez que a disposição tanto dos clientes quanto dos veículos varia ao longo do tempo.

A disposição dos clientes é ainda de extrema importância para se definir as regiões do espaço dedicadas a cada grupo de veículos. Esta é uma questão que existe na prática e surge como um problema bastante instigante a ser analisado: Como determinar o número e área das regiões? Quantos veículos alocar a cada região? Além disso, é possível imaginar que existe uma relação entre a definição das regiões e a estratégia de posicionamento, estudo que pode resultar em outras alternativas aos pontos estratégicos aqui apresentados.

Outro tema que nos surpreendeu foi a dificuldade de se inserir características práticas aos problemas teste e ambiente de simulação. Coletar dados reais, gerar problemas baseados nesses dados, adaptar os métodos heurísticos para considerar restrições práticas e desenvolver o ambiente de simulação são tarefas que demandam muito tempo do pesquisador. A recompensa é saber que existe uma maior proximidade entre as conclusões encontradas e sua aplicação prática, além do maior sentido dos resultados dos testes computacionais. Por exemplo, há a estimativa de que o método *I+_e_d_p* possa gerar uma contribuição média à empresa de transporte, por dia de operação, de aproximadamente R\$ 5.000,00 superior à contribuição de uma política baseada no vizinho mais próximo.

De uma maneira geral, o conhecimento aqui coletado e gerado nos coloca um pouco mais perto dos resultados e trabalhos realizados por outros autores, num formato mais próximo da

realidade brasileira. Mas há ainda muito a ser feito, principalmente a respeito dos métodos heurísticos e técnicas de implementação, incluindo implementação paralela.

A seguir são colocadas sugestões de pesquisa futura.

Sugestões de temas para pesquisa futura sobre o PRVJT

- Considerar os conflitos entre os objetivos de minimização da distância, número de veículos e atraso através de soluções aproximadas de Pareto;
- Projeto e implementação de métodos específicos para a redução do número de veículos e minimização da distância;
- Desenvolvimento de movimentos mais influentes para a busca tabu, por exemplo, movimentos compostos por uma parcela inter-rota, seguida por outra intra-rota;
- Expansão do espaço da busca tabu incluindo a análise de soluções infactíveis, como propõe a oscilação estratégica de Glover e Laguna (1997);
- Análise do comportamento das heurísticas em problemas teste nos quais o espaço é segmentado em regiões com veículos dedicados.

Sugestões de temas para pesquisa futura sobre o PRDVJT

- Inclusão da operação de entrega no escopo do simulador e otimizador, permitindo que sejam encaixadas coletas também nos veículos que realizam entrega;
- Acréscimo de componentes estocásticos – através de clientes fictícios – nas informações utilizadas para a definição dos roteiros;
- Teste do desempenho das heurísticas em condições dinâmicas de tráfego;
- Aplicação simultânea de heurísticas com diferentes tempos computacionais, sendo as mais rápidas focadas no período da manhã (alta taxa de novos pedidos) e as mais demoradas no período da tarde (baixa taxa de novos pedidos), como alternativa à variação do tamanho do grafo granular ao longo do turno;
- Análise do comportamento das heurísticas em problemas teste nos quais o espaço é segmentado em regiões com veículos dedicados, combinando outras estratégias de espera e de posicionamento de veículos.

Bibliografia

Balakrishnan N. (1993), Simple heuristics for the vehicle routing problem with soft time windows, *Journal of the Operational Research Society*, 44: 3, 279-287.

Ballou R. H. (2001), *Gerenciamento da cadeia de suprimentos: planejamento, organização e logística empresarial*; trad. Elias Pereira, 4^a edição, Porto Alegre: Bookman.

Bausch D.O., Brown G.G., Ronen D. (1995), Consolidating and dispatching truck shipments of mobil heavy petroleum products, *Interfaces*, 25, 1-17.

Bell W., Dalberto L.M., Fisher M.L., *et al.* (1983), Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer, *Interfaces*, 13, 4-23.

Bent R., Hentenryck P. (2004^a), A two-stage hybrid local search for the vehicle routing problem with time windows, *Transportation Science*, 38: 4, 515-530.

Bent R., Hentenryck P. (2004^b), Scenario-based planning for partially dynamic vehicle routing with stochastic customers, *Operations Research*, 52: 6, 977-987.

Bianchi L. (2000), Notes on Dynamic Vehicle Routing – The State of the Art, Technical Report IDSIA-05-01.

Bräysy O. (2003), A reactive variable neighborhood search for the vehicle-routing problem with time windows, *INFORMS Journal on Computing*, 15: 4, 347-368.

Brown G.G., Ellis C.J., Graves G.W., Ronen D. (1987), Real-time wide area dispatching of Mobil tank trucks, *Interfaces*, 17, 107-120.

Chiang W.C., Russel R.A. (1997), A reactive tabu search metaheuristic for the vehicle routing problem with time windows, *INFORMS Journal on Computing*, 9, 417-430.

Clarke G., Wright J.V. (1964), Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, 12, 568-581.

Cordeau J-F, Gendreau M., Laporte G., (1997). A tabu search heuristic for the periodic and multi-depot vehicle routing problems. *Networks* 30: 105–119.

Cordeau J-F, Gendreau M., Laporte G., Potvin J-Y e Semet F. (2002), A guide to vehicle routing heuristics, *Journal of the Operational Research Society*, 53, 512-522.

Cordeau J.F., Laporte G., Mercier A. (2001), A unified tabu search for vehicle routing problems with time windows, *Journal of the Operational Research Society*, 52, 928-936.

Divisão de Transportes da Universidade Federal de Viçosa (2004)
<http://www.cpd.ufv.br/dtr/>

Fisher M.L., Jaikumar R. (1981), A generalized assignment heuristic for vehicle routing. *Networks* 11, 109-124.

Fleischmann B., Gnutzmann S., Sandvoß E. (2004), Dynamic vehicle routing based on online traffic information, *Transportation Science*, 38: 4, 420-433.

Gendreau M., Guertin F., Potvin J.Y., Taillard E. (1999), Parallel tabu search for real-time vehicle routing and dispatching, *Transportation Science*, 33, 381-390.

Gendreau M., Hertz A., Laporte G. (1994), A tabu search heuristic for the vehicle routing problem, *Management Science* 40, 1276-1290.

Gendreau M., Laporte G., Séguin R. (1996), Stochastic vehicle routing, *European Journal of Operational Research*, 88, 3-12.

Gendreau M., Laporte G., Semet F. (2001), A dynamic and parallel tabu search heuristic for real-time ambulance relocation, *Parallel Computing*, 27, 1641-1653.

Gendreau M., Potvin J.Y. (1997), Dynamic vehicle routing and dispatching, Technical. Report. CRT 97-38, Centre de Recherche sur les Transport, Université de Montréal.

Ghiani G., Guerriero F., Laporte G., Musmanno R. (2003), Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 151, 1-11.

Gillet B.E., Miller L.R. (1974), A heuristic algorithm for the vehicle dispatch problem. *Operations Research* 22, 240-349.

Glover F. (1991), Multilevel tabu search and embedded search neighborhoods for the traveling salesman problem, Technical Report, University of Colorado at Boulder.

Glover F. (1996), Ejection chains, reference structures and alternating path methods for traveling salesman problems, *Discrete Applied Mathematics*, 65, 223-253.

Glover F., Laguna M. (1997), *Tabu Search*, Kluwer.

Gutenschwager K., Niklaus C., Voß S. (2004), Dispatching of an electric monorail system: applying metaheuristics to an online pickup and delivery problem, *Transportation science*, 38: 4, 434-446.

Homberger J. (2000)

<http://www.fernuni-hagen.de/WINF/touren/inhalte/probinst.htm#cust200>

Homberger J., Gehring H. (1999), Two evolutionary metaheuristics for the vehicle routing problem with time windows, *INFOR*, 37, 297-318.

Hvattum L., Løkketangen A., Laporte G., (2004) Solving a Dynamic and Stochastic Vehicle Routing Problem with a Sample Scenario Hedging Heuristic, Report CRT-2004-08, Centre for research on Transportation, Montreal, Canada.

Ichoua S., Gendreau M., Potvin J.Y. (2000), Diversion issues in real-time vehicle dispatching, *Transportation Science*, 34, 426-438.

Ichoua S., Gendreau M., Potvin J.Y. (2005), Exploiting Knowledge about Future Demands for Real-Time Vehicle Dispatching, accepted for publication in *Transportation Science*.

Ioannou G., Kritikos M. e Pratascos G. (2001), A greedy look-ahead heuristic for the vehicle routing problem with time windows, *Journal of the Operational Research Society*, 52, 523-537.

Jonker R., Volgenant A. (1987), A shortest augmenting path algorithm for dense and sparse linear assignment problems, *Computing*, 38, 325-340.

Kohl N., Desrosiers J., Madsen O.B.G., Solomon M.M., Soumis F. (1999), 2-Paths cuts for the vehicle routing problem with time windows, *Transportations Science*, 33, 101-116.

Kontoravdis G., Bard J.F., (1995) A GRASP for the vehicle routing problem with time windows, *ORSA Journal on Computing*, 7:1, 10-23.

Laporte G., Gendreau M., Potvin J-Y, Semet F. (2000), Classical and modern heuristics for the vehicle routing problem, *International Transactions in Operational Research*, 7, 285-300.

Laporte G., Louveaux, F.V., Mercure, H. (1994), A priori optimization of the probabilistic traveling salesman problem, *Operations Research*, 42, 543-549.

Larsen A. (2000), The dynamic vehicle routing problem, PhD thesis, Technical University of Denmark.

Larsen A., Madsen O., Solomon M. (2002), Partially dynamic vehicle routing – models and algorithms, *Journal of the Operational Research Society*, 53, 637-646.

Larsen A., Madsen O., Solomon M. (2004), The a priori dynamic traveling salesman problem with time windows, *Transportation Science*, 38: 4, 459-472.

Mazzini, R. (1998), Estudo de Meta-Heurísticas Populacionais para a Programação de Máquinas Paralelas com Tempos de Preparação Dependentes da Sequência e Datas de Entrega, tese apresentada à Faculdade de Engenharia Elétrica e de Computação da UNICAMP.

Michalewicz Z. (1996), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 3rd edition.

Mitrović-Minić S., Krishnamurti R., Laporte G. (2004^a), Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows, *Transportation Research Part B*, 38, 669-685.

Mitrović-Minić S., Laporte G. (2004^b), Waiting strategies for the dynamic pickup and delivery problem with time windows, *Transportation Research Part B*, 38, 635-655.

Or I. (1976), Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking, Ph.D. thesis, Department of Industrial Engineering and Management Sciences, Northwestern University.

Osman I.H. e Christofides N. (1989), Simulated annealing and descent algorithms for capacitated clustering problem, Research Report, Imperial College, University of London.

PMELink.pt (2004) - Centro de negócios On-line do mercado português

http://www.pmelink.pt/pmelink_public/EC/0,1655,1005_5053-3_41104--View_429,00.html

Potvin J.Y., Bengio S. (1996), The vehicle routing problem with time windows – Part II: Genetic search, *INFORMS Journal on Computing*, 8, 165-172.

Potvin J.Y., Kervahut T., Garcia B., Rousseau J.M. (1996), The vehicle routing problem with time windows – Part I: Tabu search, *INFORMS Journal on Computing*, 8, 158-164.

Potvin J.Y. e Rousseau J.M. (1995), An exchange heuristic for routeing problems with time windows, *Journal of the Operational Research Society*, 46, 1433-1446.

Psaraftis, H.N. (1988), Dynamic vehicle routing problems, em *Vehicle routing: methods and studies*, eds. Golden B.L. e Assad A.A., Elsevier, North-Holland, 223-248.

Regan A.C., Mahmassani, Jaillet P. (1994), Improving efficiency of commercial vehicle operations using real-time information: Potential uses and assignment strategies, *Transportation Research Record*, 1493, 188-198.

Regan A.C., Mahmassani, Jaillet P. (1995), Dynamic decision making for commercial fleet operations using real-time information, *Transportation Research Record*, 1537, 91-97.

Resende M.G.C., Ribeiro C.C. (2002), “Greedy randomized adaptive search procedures”, em *State-of-the-Art Handbook of Metaheuristics* (Glover F., Kochenberger G., editores), Kluwer.

Rochat Y., Taillard E.D. (1995), Probabilistic diversification in local search for vehicle routing, *Journal of Heuristics*, 1, 147-167.

Russel R., Chiang W. (2004), Scatter search for the vehicle routing problem with time windows, a ser publicado no *European Journal of Operational Research*.

Savelsbergh M. W. P. (1992), The vehicle routing problem with time windows: minimizing route duration, *ORSA Journal on Computing*, 4, 146-154.

Solomon M. (1987), Algorithms for the vehicle routing and scheduling problems with time windows, *Operations Research*, 35, 254-265.

Solomon M. (2004) <http://w.cba.neu.edu/~msolomon/problems.htm>, Northeastern University, Boston, EUA

Taillard E.D., (1993) Parallel iterative search methods for vehicle routing problems. *Networks* 23, 661-673.

Taillard E.D., Badeau P., Gendreau M., Guertin F., Potvin J.Y. (1997) A tabu search for the vehicle routing problem with soft time windows, *Transportation Science*, 31, 170-186.

Tan K.C., Lee L.H., Zhu Q.L., Ou K. (2001), Heuristic methods for vehicle routing problem with time windows, *Artificial Intelligence in Engineering*, 15, 281-295.

Taniguchi E., Shimamoto H. (2004), Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times, *Transportation Research Part C*, 12, 235-250.

Toth P., Vigo D. (editores, 2002), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications.

Toth P., Vigo D. (2003), The granular tabu search and its application to the vehicle-routing problem, *Inform Journal on Computing*, 15:4, 333-346.

Zhu K., Ong K. (2000), A Reactive Method for Real Time Dynamic Vehicle Routing Problem. in *IEEE International Conference on Tools with Artificial Intelligence*, 12th ICTAI 2000, Vancouver, Canada.

Anexos

A. Guia de entrevista para a pesquisa a campo

Empresa: _____ Data ____ / ____ / ____

Cidade: _____

Nome: _____ Depto.: _____

Nome: _____ Depto.: _____

Informações Gerais

Número de veículos

Tamanho do veículo	Próprios	Terceiros	Total
a)			
b)			
c)			
d)			

Número de funcionários

	Próprios	Terceiros	Total
Todos			
Motoristas			
Operação			

Transporte

	Tráfego urbano	Rodovias
Velocidade média (km/h)		
Km média: diária () anual ()		
Distância máxima percorrida por um veículo num dia (km)		
Número médio de clientes visitados por dia		
Tempo médio de parada por visita (minutos) (inclui tempo de serviço)		
Número de horas no turno do motorista (inclui uma hora de almoço)		
Custo variável de transporte (R\$/km) (combustível, óleo do motor, graxas, pneus, manutenção, outros)		

Custo do motorista (R\$/mês) (salário, benefícios, encargos)		
Outros custos fixos para ter o veículo (R\$/mês) (compra do veículo, licenciamento, seguros)		
Outros custos		
Outros custos		
Outros custos		

Cientes

	Na cidade do depósito	Estado SP	Total
Número de clientes na carteira			
% de novos clientes por mês			

Características dos clientes

Distância entre o depósito e o centróide dos clientes: _____ km

Os clientes estão: () concentrados em ____ regiões () completamente espalhados

% de clientes com janelas de tempo: _____ Tamanho médio da janela (horas): _____

Existe multa se atrasar o transporte? _____

% médio da capacidade do veículo que um cliente ocupa: _____

Receita média por cliente visitado (R\$ / cliente visitado): _____

Número mínimo de clientes visitados para que a rota seja lucrativa: _____

% de clientes atendidos no mesmo dia em que o pedido é colocado (dinâmico): _____

Uso de tecnologia

Existe alguma ferramenta para posicionar um cliente num mapa digitalizado? ()sim ()não

Qual? _____ Custo: _____

Existe alguma ferramenta para criar os roteiros dos motoristas? ()sim ()não

Qual? _____ Custo: _____

Como é tomada a decisão de aceitar um novo pedido? _____

Quantos veículos são equipados com GPS? _____

Como é feita a comunicação entre a empresa e os motoristas?

() telefone fixo () celular () rádio / *trunking* () Outros: _____

O motorista tem condições de cadastrar um novo pedido? _____

O motorista tem condições de emitir notas fiscais? _____

Como e quando a empresa toma conhecimento do término de uma entrega? _____

Caso existam clientes dinâmicos

Distribuição dos pedidos atendidos no mesmo dia em que chegam (%)

Antes das 8hs	8hs – 10hs	10hs – 12hs	12hs – 14hs	14hs – 16hs	Após 16hs	Total
						100%

O motorista pode ser desviado do seu roteiro para incluir um novo cliente? () sim () não

Caso positivo, qual é o tempo entre a chegada do novo pedido e o momento em que motorista começa a seguir para o novo destino (alteração do roteiro, comunicação com o motorista, desvios): _____ minutos.

Após a decisão de colocar um novo veículo na rua, quanto tempo leva para o veículo iniciar sua rota? _____ minutos.

B. Heurística de Rochat e Taillard (1995) para o PRVJT

Inicialização:

1. Gere I soluções distintas a partir de uma busca local;
2. Associe a cada rota um índice contendo o valor da função objetivo da solução a que a rota pertence;
3. Remova as rotas que contém apenas um cliente;
4. Insira as rotas restantes num conjunto T (é permitida a inserção de rotas idênticas, independente do valor de seu índice);
5. Ordene as rotas do conjunto T por ordem crescente dos índices;
6. Limite a cardinalidade do conjunto T a um tamanho L , removendo as últimas rotas da lista;

Método de diversificação e intensificação:

1. Defina $T'=T$ e $S = \emptyset$
2. Enquanto $T' \neq \emptyset$, faça:
3. Selecione uma rota t de T' probabilisticamente (ponderação feita pelos índices);
4. Faça $S = S \cup t$
5. Remova de T' todas as rotas que contém pelo menos um cliente de t
6. Se existirem clientes de S não roteados, crie e modifique rotas para tornar a solução factível. Denomine esta solução S' ;
7. Aplique a busca local em S' ;
8. Atualize o conjunto T a partir das rotas de S' e repita o processo.

C. Estratégias para problemas dinâmicos – Bianchi (2000)

Bianchi (2000) menciona que a estratégia para resolver problemas de roteamento dinâmico de veículos contém dois pontos principais:

- 1) Constituição de um SP_i : Determina quando um sub-problema é formado para ser apresentado a um método heurístico, como, por exemplo, à busca tabu. A cada nova informação? A cada grupo de n novas informações? Este grupo depende da urgência das informações reveladas?
- 2) Posicionamento dos veículos: Existem estratégias, por exemplo, que buscam evitar a concentração de veículos no espaço, numa suposição que existem pontos de demanda do SP_f distribuídos por todas as regiões. Este tipo de estratégia deveria guiar questões como: para onde direcionar o veículo quando não existe um novo destino programado? Quando um pedido deve ser rejeitado em virtude de um posicionamento de veículos indesejável? Como evitar a concentração de veículos na mesma região do espaço?

A autora lista também as principais estratégias propostas pela literatura, sendo as estratégias abaixo aplicadas a problemas com múltiplos veículos:

- 1) Estratégia da mediana da fila estocástica: Posicione o veículo na mediana, sirva os clientes pela ordem *FCFS*¹⁸ (os clientes são atendidos na mesma ordem em que surgem), e retorne à mediana ao término de cada atendimento;
- 2) Estratégia do caixeiro viajante: Forme conjuntos de n demandas consecutivas e coloque os conjuntos numa fila. Atenda os conjuntos da fila pela ordem *FCFS*: o primeiro veículo disponível sai do depósito, atende parcialmente ou totalmente o primeiro conjunto da fila, e retorna ao depósito, o segundo veículo disponível atende a parcela restante do primeiro conjunto (se existir), ou outra parcela do segundo conjunto, e assim por diante. Para cada conjunto, a rota ótima é obtida através da resolução de um problema do caixeiro viajante;
- 3) Estratégia modificada do caixeiro viajante: Modificação da estratégia do caixeiro viajante. Primeiramente o espaço geográfico de atendimento é dividido em k setores. Em seguida, cada conjunto das n demandas consecutivas é separado em outros k conjuntos de acordo com os setores. Os novos conjuntos são atendidos pela ordem *FCFS* por rotas ótimas que começam e terminam no depósito;
- 4) Estratégia do vizinho mais próximo: Seja um depósito com m veículos. As primeiras m demandas são atendidas pelos m veículos. No momento em que um veículo fica ocioso, este veículo atende a demanda mais próxima que ainda não foi visitada;

Por último, Bianchi (2000) menciona a carência de artigos que compare a importância da estratégia adotada com a escolha da heurística. O mais comum é encontrar artigos focados em apenas um dos dois temas. Cita também possíveis extensões como a possibilidade de se permitir que um veículo possa ser desviado de seu destino planejado e o uso de informações probabilísticas na definição da estratégia.

D. Detalhamento sobre desvio

A Figura 6.1 a seguir ilustra uma situação em que um veículo está percorrendo sua rota planejada de acordo com demanda conhecida do cliente A ao cliente F. No instante t ocorre uma nova demanda proveniente do cliente G. A viabilidade de atendimento deste novo cliente depende da restrição de capacidade do veículo (caso exista) e da restrição da janela de tempo.

¹⁸ *First come, First serve*

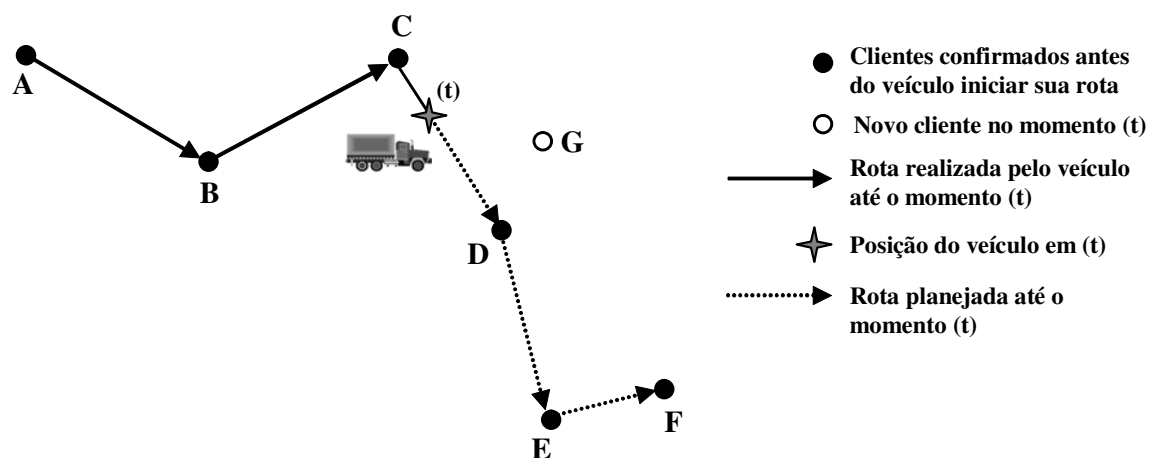
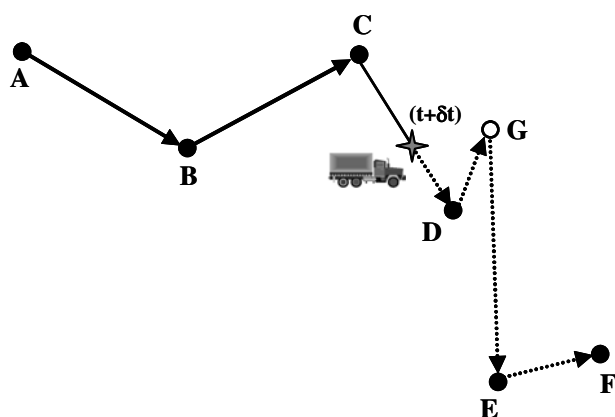


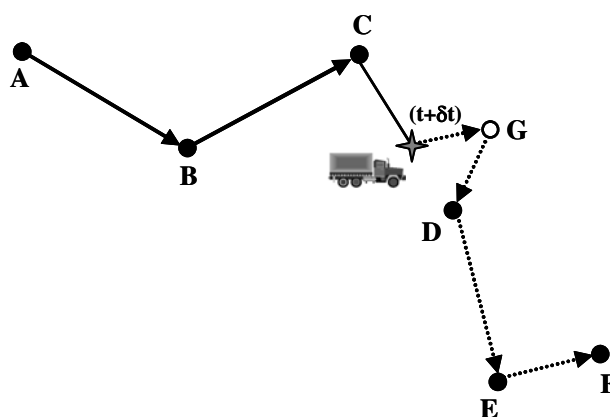
Figura 6.1. Exemplo de problema dinâmico de roteamento de veículos com janelas de tempo

Uma vez decidido pelo atendimento ao cliente G, a modificação da rota planejada pode ser feita de duas maneiras: sem possibilidade de desvio do destino atual ou com possibilidade de desvio do destino atual, como ilustrado na Figura 6.2.

Roteamento dinâmico sem possibilidade de desvio do destino atual



Roteamento dinâmico com possibilidade de desvio do destino atual



● Clientes confirmados antes do veículo iniciar sua rota

○ Novo cliente no momento (t)

→ Rota realizada pelo veículo até o momento (t + δt)

★ Posição do veículo em (t + δt)

⋯ Rota planejada até o momento (t + δt)

(*) δt é a soma dos tempos necessários para o algoritmo apresentar uma nova solução e para comunicar a nova rota ao veículo

Figura 6.2. Roteamento dinâmico sem e com possibilidade de desvio do destino atual

No primeiro caso, o veículo prossegue até o cliente programado D, e uma nova rota é determinada somente após o término do serviço em D. No segundo caso, o veículo é desviado da rota programada para atender o cliente G.

Para a situação com desvio, um valor alto de δt (tempo necessário para o algoritmo apresentar uma nova solução) reduz a economia proveniente do desvio, uma vez que os veículos a serem desviados percorrem uma maior distância desnecessária. Todavia, em qualquer dos casos considerados acima, é fundamental que se tenha um algoritmo que gere uma boa solução em tempo rápido. Ichoua *et al.* (2000) mostram resultados em que a utilização da estratégia de desvio é melhor que a não utilização. Estratégias de desvio foram também propostas em (Regan *et al.* 1994, Regan *et al.* 1995) no contexto de transportadoras de cargas.

E. Busca tabu de Taillard *et al.* (1997)

Esta busca tabu foi utilizada para a resolução do problema estático de roteamento de veículos:

1. Construa I soluções iniciais com uma heurística de inserção estocástica;
2. Aplique a busca tabu a cada solução e armazene as rotas resultantes na memória adaptativa;
3. Por W iterações faça:
 - 3.1. Construa uma solução inicial com as rotas armazenadas na memória adaptativa e defina esta solução como solução corrente;
 - 3.2. Por C ciclos faça:
 - a. Decomponha a solução corrente em D subconjuntos de rotas com intersecção nula entre si;
 - b. Aplique a busca tabu a cada subconjunto de rotas;
 - c. Combine as rotas resultantes para criar uma nova solução corrente;
 - 3.3. Adicione as rotas resultantes na memória adaptativa;

Ao término do procedimento, uma rotina de pós-otimização é aplicada a todas as rotas da melhor solução encontrada.

F. Busca tabu de Gendreau *et al.* (1999)

Esta busca tabu é uma adaptação da busca tabu de Taillard *et al.* (1997):

1. Enquanto não ocorre nenhum evento (novo pedido ou término de serviço de algum veículo), otimize as rotas planejadas utilizando a busca tabu;
2. Se um evento ocorrer, pare a busca tabu, adicione as rotas de suas melhores soluções à memória adaptativa;
 - 2.1. Se o evento for um novo pedido:
 - a. Atualize a memória adaptativa através da inserção do novo pedido em cada solução;
 - b. Caso não haja nenhuma inserção factível, rejeite o pedido;
 - 2.2. Se o evento for o término do serviço de algum veículo:
 - a. Identifique o próximo destino do motorista consultando a melhor solução da memória adaptativa;
 - b. Atualize as outras soluções considerando a decisão anterior;
 - 2.3. Reinicie a busca tabu a partir das novas soluções da memória adaptativa;

A cada evento, a busca local troca-cross (troca de dois subconjuntos de até k clientes em seqüência de duas rotas distintas) é aplicada à melhor solução da memória.

G. Movimentos da busca local granular

1) Movimento intra-rota troca-arcos

Para cada chamada da vizinhança troca-arcos, duas opções de movimentos são avaliadas. A vizinhança retorna a opção de menor custo

Tabela 6.1. Intra-rota troca-arcos – 1ª opção

Notação:	– Aa = cidade anterior a ‘a’ – Ab = cidade anterior a ‘b’ – Pa = cidade posterior a ‘a’ – Pb = cidade posterior a ‘b’
Arestas removidas	– (Aa, a) – (Ab, b)
Arestas inseridas	– (a, b) – (Aa, Ab)
Observações	– A cidade ‘a’ deve preceder a cidade ‘b’ – não imediatamente. Caso ‘b’ esteja antes de ‘a’, o movimento é avaliado com as cidades trocadas; – O trecho entre as cidades ‘Ab’ e ‘a’ tem seu sentido invertido.

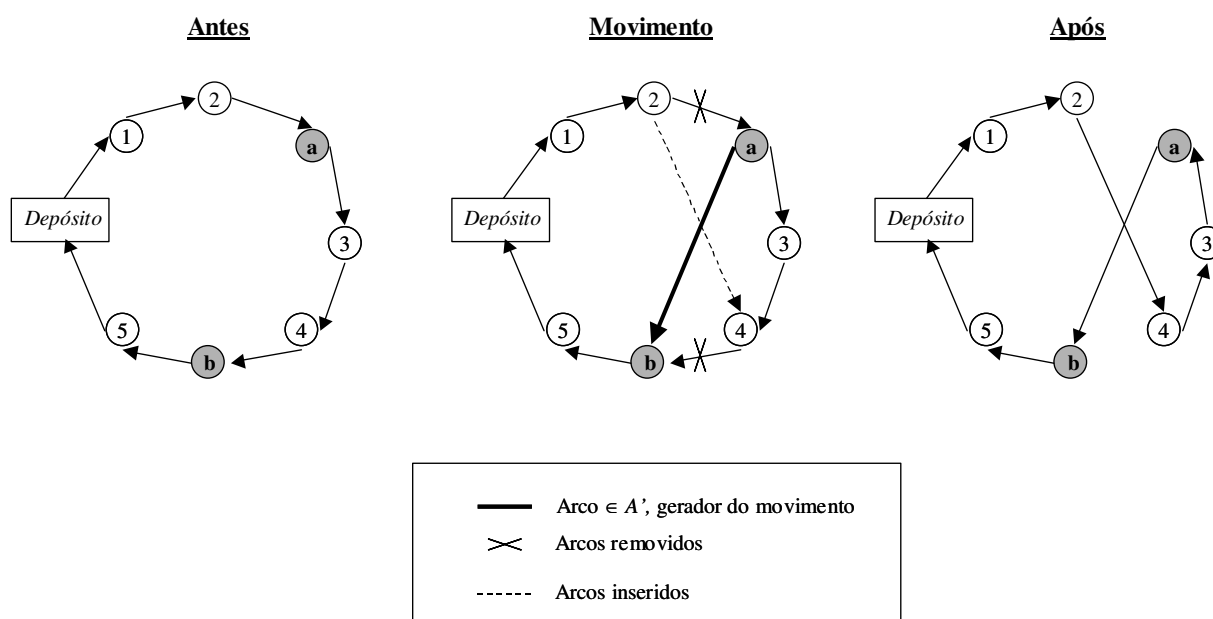
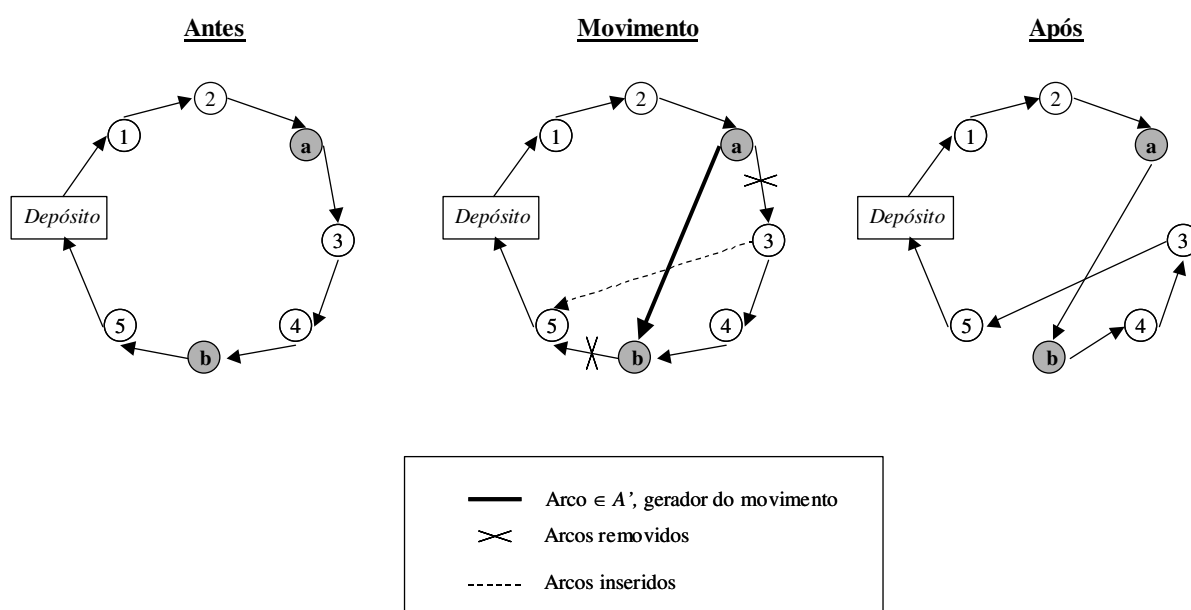
**Figura 6.3. Movimento intra-rota troca-arcos – 1ª opção**

Tabela 6.2. Intra-rota troca-arcos – 2ª opção

Notação:	– Aa = cidade anterior a ‘a’ – Ab = cidade anterior a ‘b’ – Pa = cidade posterior a ‘a’ – Pb = cidade posterior a ‘b’
Arestas removidas	– (a, Pa) – (b, Pb)
Arestas inseridas	– (a, b) – (Pa, Pb)
Observações	– A cidade ‘a’ deve preceder a cidade ‘b’ – não imediatamente. Caso ‘b’ esteja antes de ‘a’, o movimento é avaliado com as cidades trocadas; – O trecho entre as cidades ‘b’ e ‘Pa’ tem seu sentido invertido.

**Figura 6.4. Movimento intra-rota troca-arcos – 2ª opção**

Se a cidade ‘a’ for o depósito¹⁹ os movimentos troca-arcos ficam da seguinte forma:

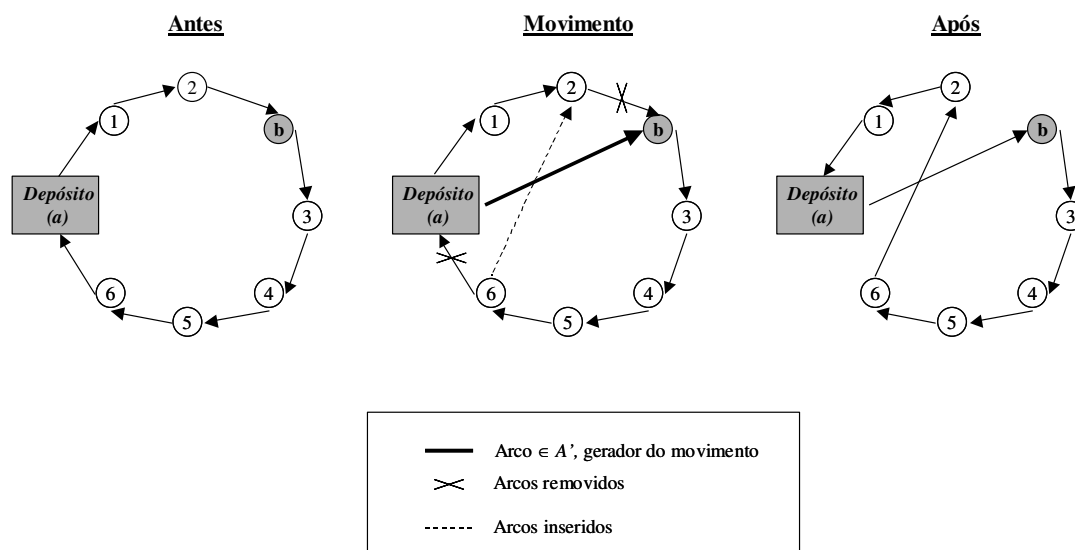


Figura 6.5. Movimento intra-rota troca-arcos com ‘a’ igual ao depósito – 1ª opção

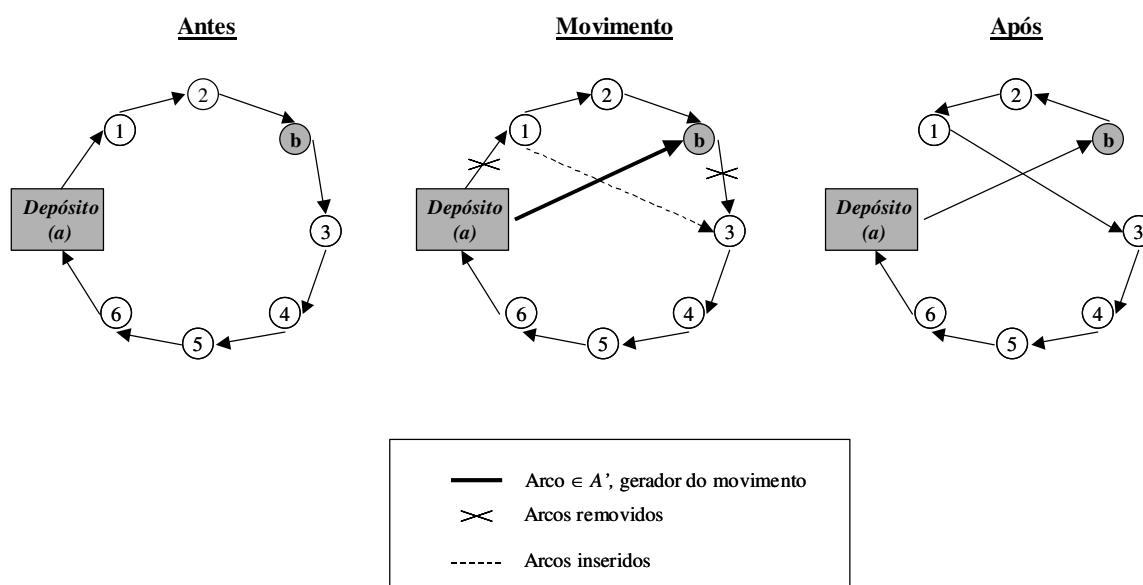


Figura 6.6. Movimento intra-rota troca-arcos com ‘a’ igual ao depósito – 2ª opção

¹⁹ Se b for o depósito, faça b = a e a = 0.

2) Movimento intra-rota inserção

Dado uma aresta (a, b) a ser inserida, o movimento insere as cidades a partir de 'b' (inclusive) entre a cidade 'a' e a cidade imediatamente posterior a 'a'.

Tabela 6.3. Intra-rota inserção

Notação:	– Aa = cidade anterior a 'a' – Ab = cidade anterior a 'b' – Pa = cidade posterior a 'a' – Pb = cidade posterior a 'b' – x = cidade do fim do trecho a ser movido, 'x' está entre 'b' (inclusive) e o depósito (exclusive) – Px = cidade posterior a 'x'
Arestas removidas	– (Ab, b) – (a, Pa) – (x, Px)
Arestas inseridas	– (a, b) – (Ab, Px) – (x, Pa)
Observações	– A cidade 'a' deve preceder a cidade 'b' – não imediatamente. Caso 'b' esteja antes de 'a', o movimento é avaliado com as cidades trocadas; – 'x' começa igual a 'b' e é percorrido até a cidade imediatamente anterior ao depósito.

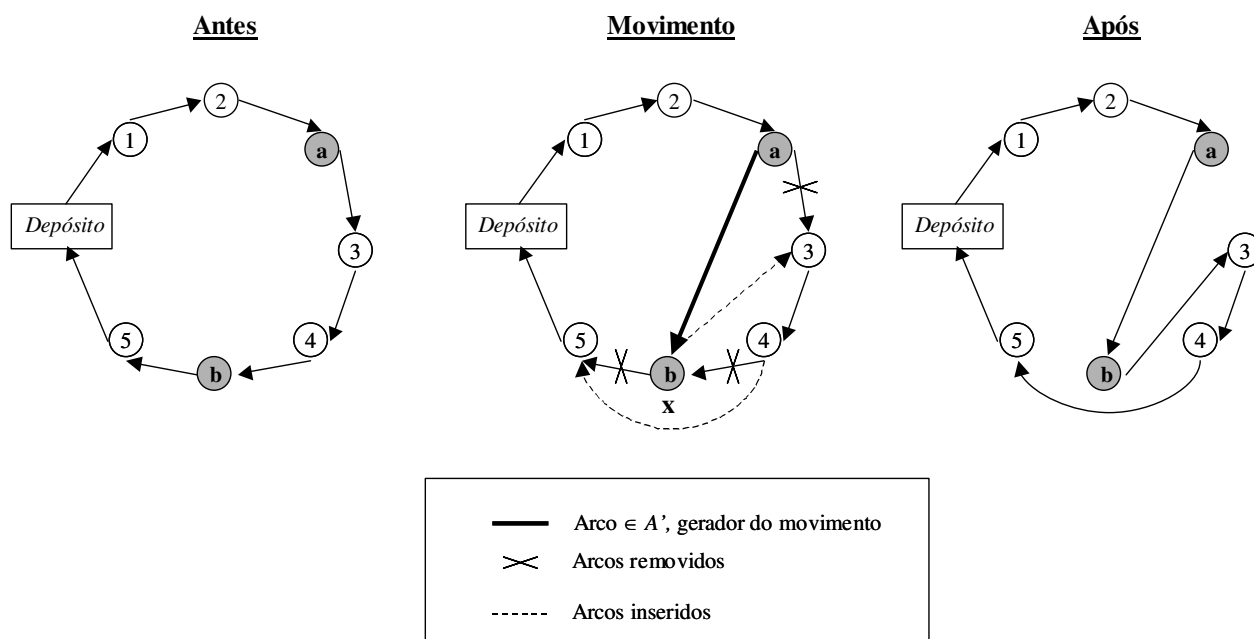


Figura 6.7. Movimento intra-rotas inserção com $x = b$

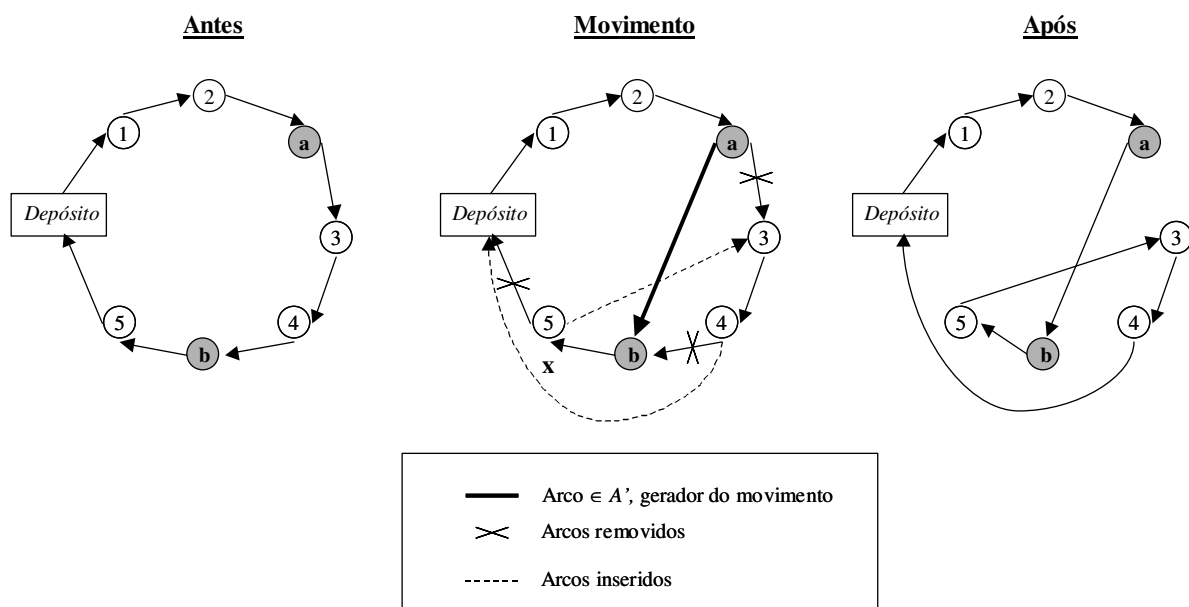


Figura 6.8. Movimento intra-rotas inserção com $x = 5$

3) Movimento inter-rota *crossover*

Extensão do movimento implementado por Toth e Vigo (2003), sendo que para cada aresta (a,b) , as 4 opções a seguir são avaliadas.

Tabela 6.4. Inter-rota *crossover* – 1ª opção

Notação:	– Aa = cidade anterior a ‘a’ – Ab = cidade anterior a ‘b’ – Pa = cidade posterior a ‘a’ – Pb = cidade posterior a ‘b’
Arestas removidas	– (a, Pa) – (Ab, b)
Arestas inseridas	– (a, b) – (Ab, Pa)
Observações	– Se ‘a’ for a última cidade da primeira rota e ‘b’ a primeira cidade da segunda rota é feita a união das rotas.

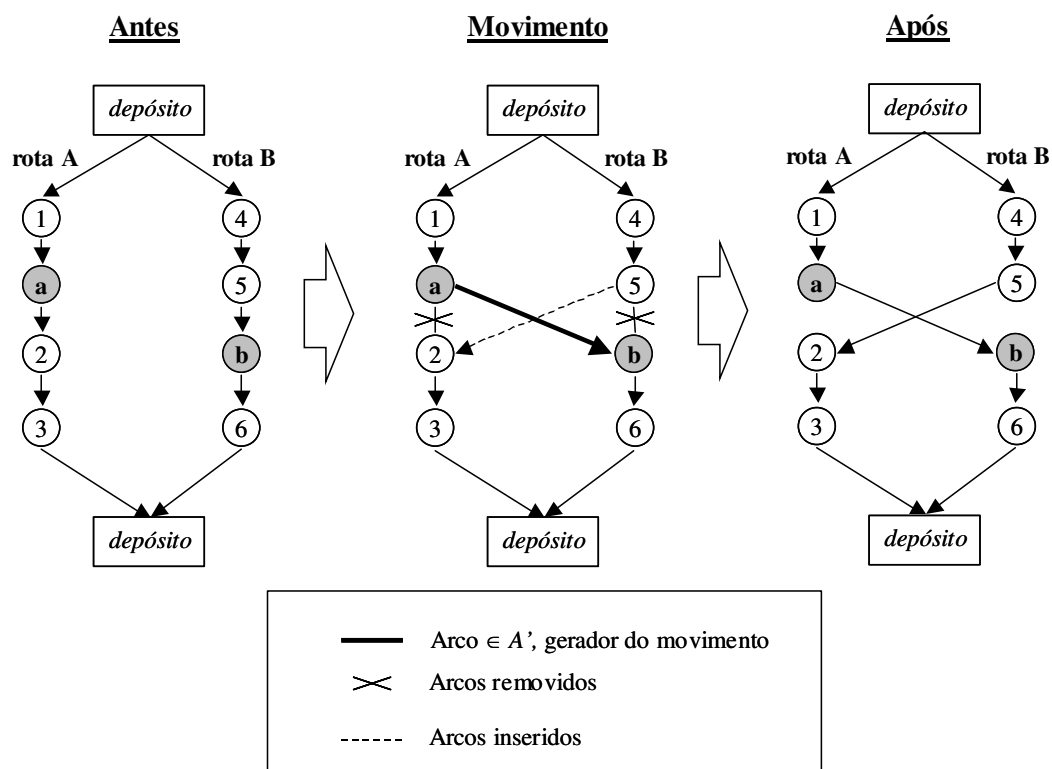


Figura 6.9. Movimento inter-rota crossover – Opção 1

Tabela 6.5. Inter-rota crossover – 2ª opção

Notação:	– Aa = cidade anterior a 'a' – Ab = cidade anterior a 'b' – Pa = cidade posterior a 'a' – Pb = cidade posterior a 'b'
Arestas removidas	– (Aa, a) – (b, Pb)
Arestas inseridas	– (b, a) – (Aa, Pb)
Observações	– Se 'a' for a última cidade da primeira rota e 'b' a primeira cidade da segunda rota é feita a união das rotas.

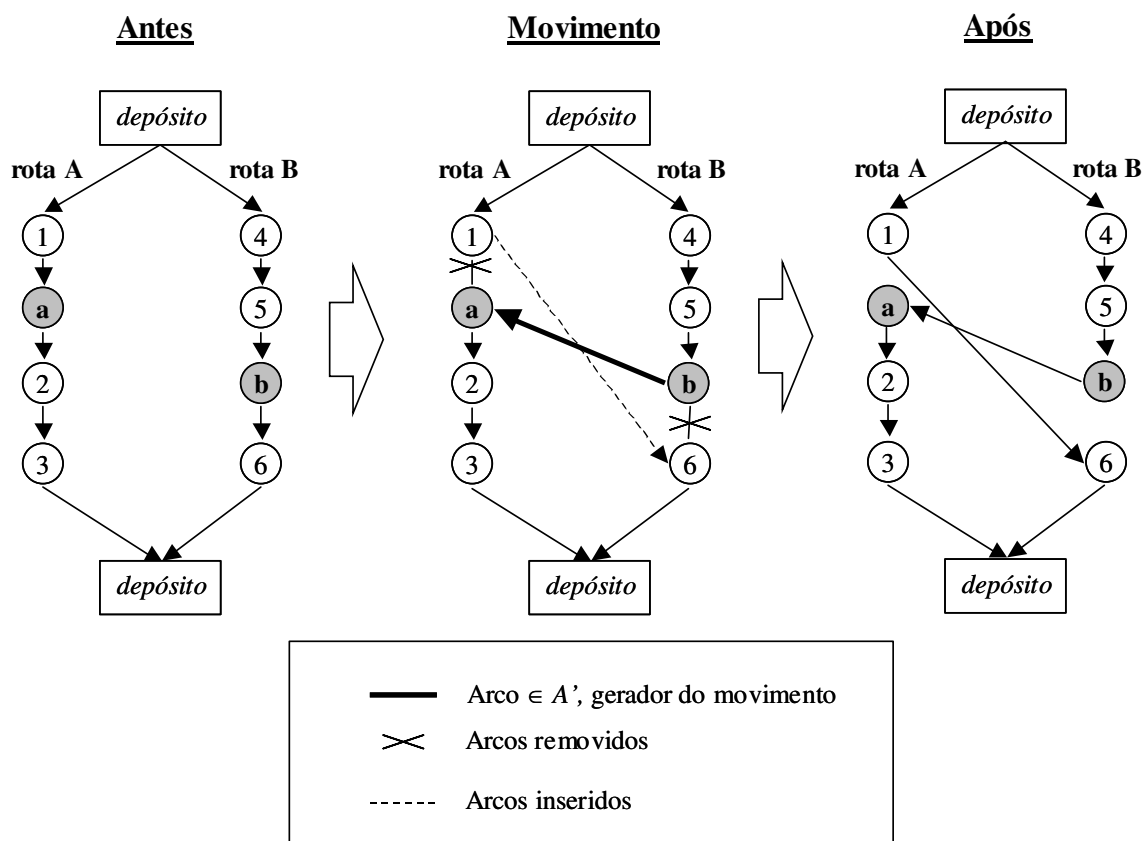


Figura 6.10. Movimento inter-rota *crossover* – Opção 2

Tabela 6.6. Inter-rota *crossover* – 3ª opção

Notação:	– Aa = cidade anterior a 'a' – Ab = cidade anterior a 'b' – Pa = cidade posterior a 'a' – Pb = cidade posterior a 'b'
Arestas removidas	– (a, Pa) – (b, Pb)
Arestas inseridas	– (a, b) ou (b, a) – (Pa, Pb) ou (Pb, Pa)
Observações	– Se 'a' for a última cidade da primeira rota e 'b' a primeira cidade da segunda rota é feita a união das rotas; – Se qualquer das rotas resultantes for infactível em virtude da restrição de janela de tempo, é feita a tentativa de se inverter o sentido da rota.

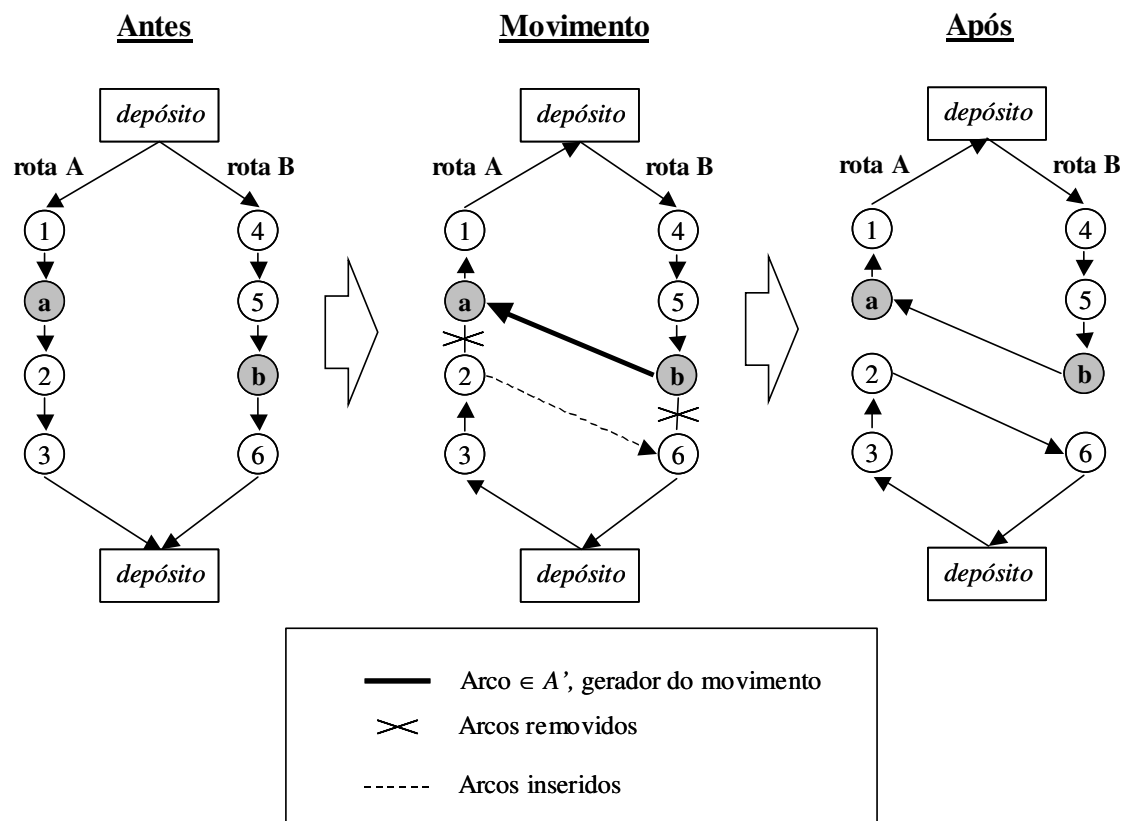
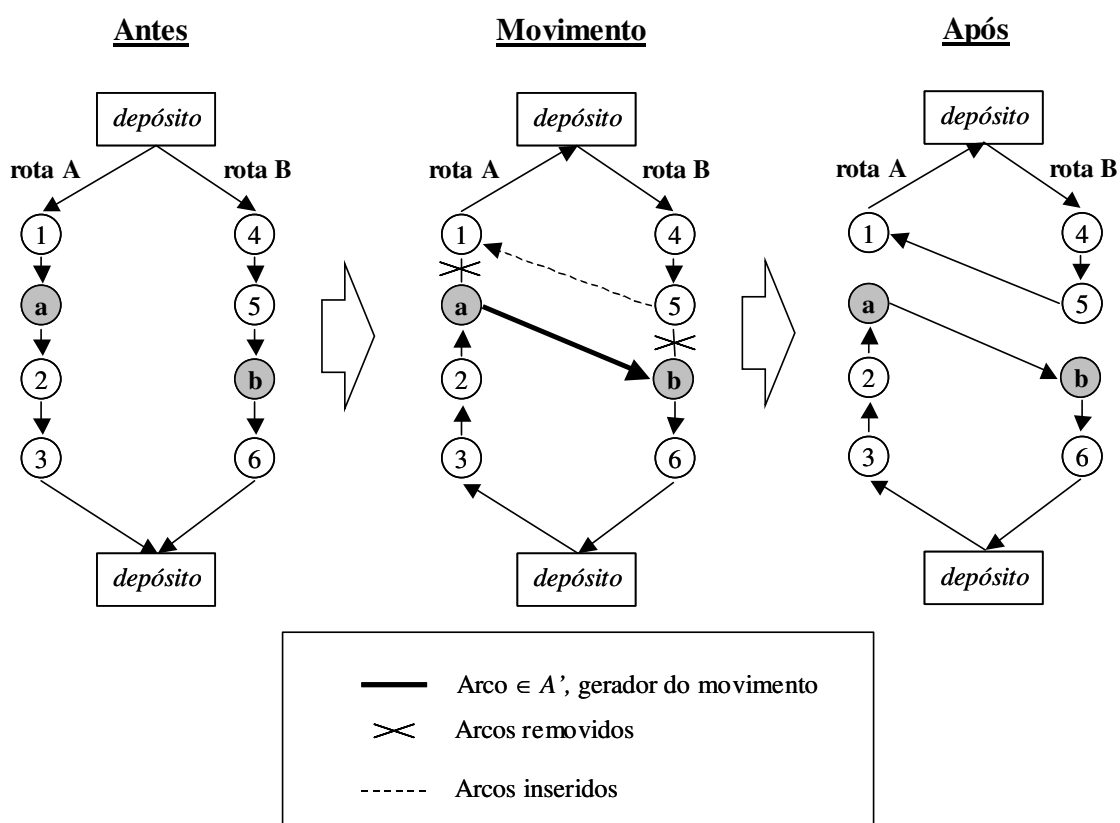


Figura 6.11. Movimento inter-rote *crossover* – Opção 3

Tabela 6.7. Inter-rota crossover – 4ª opção

Notação:	– Aa = cidade anterior a ‘a’ – Ab = cidade anterior a ‘b’ – Pa = cidade posterior a ‘a’ – Pb = cidade posterior a ‘b’
Arestas removidas	– (Aa, a) – (Ab, b)
Arestas inseridas	– (a, b) ou (b, a) – (Aa, Ab) ou (Ab, Aa)
Observações	– Se ‘a’ for a última cidade da primeira rota e ‘b’ a primeira cidade da segunda rota é feita a união das rotas; – Se qualquer das rotas resultantes for infactível em virtude da restrição de janela de tempo, é feita a tentativa de se inverter o sentido da rota.

**Figura 6.12. Movimento inter-rota crossover – Opção 4**

4) Movimento inter-rota inserção

O movimento inter-rota inserção move trechos de uma rota a outra. Para cada aresta (a, b) dois tipos de inserções são avaliadas – demonstradas a seguir. Como os movimentos para o arco (a, b) são diferentes dos movimentos para o arco (b, a) , este movimento considera as duas possibilidades, avaliando, portanto, quatro opções no total.

Tabela 6.8. Inter-rota inserção – Opção 1

Notação:	– Aa = cidade anterior a ‘a’ – Ab = cidade anterior a ‘b’ – Pa = cidade posterior a ‘a’ – Pb = cidade posterior a ‘b’ – x = cidade da rota B do fim do trecho a ser movido, ‘x’ está entre ‘b’ (inclusive) e o depósito (exclusive) – Px = cidade posterior a ‘x’
Arestas removidas	– (Ab, b) – (a, Pa) – (x, Px)
Arestas inseridas	– (a, b) – (Ab, Px) – (x, Pa)
Observações	– ‘x’ começa igual a ‘b’ e é percorrido até a cidade imediatamente anterior ao depósito.

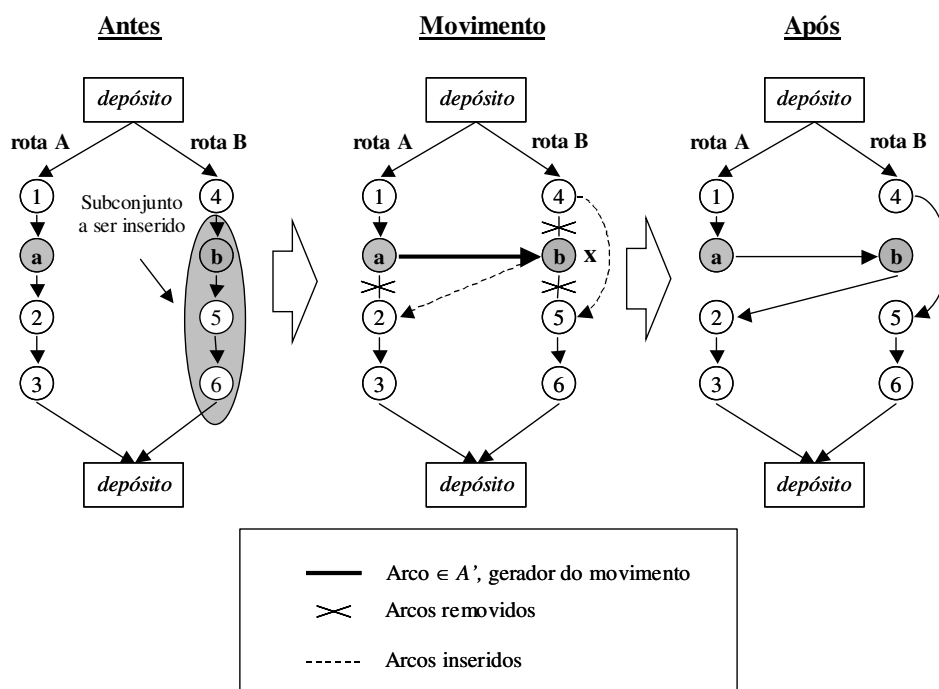


Figura 6.13. Movimento inter-rotas inserção – Opção 1 com $x = b$

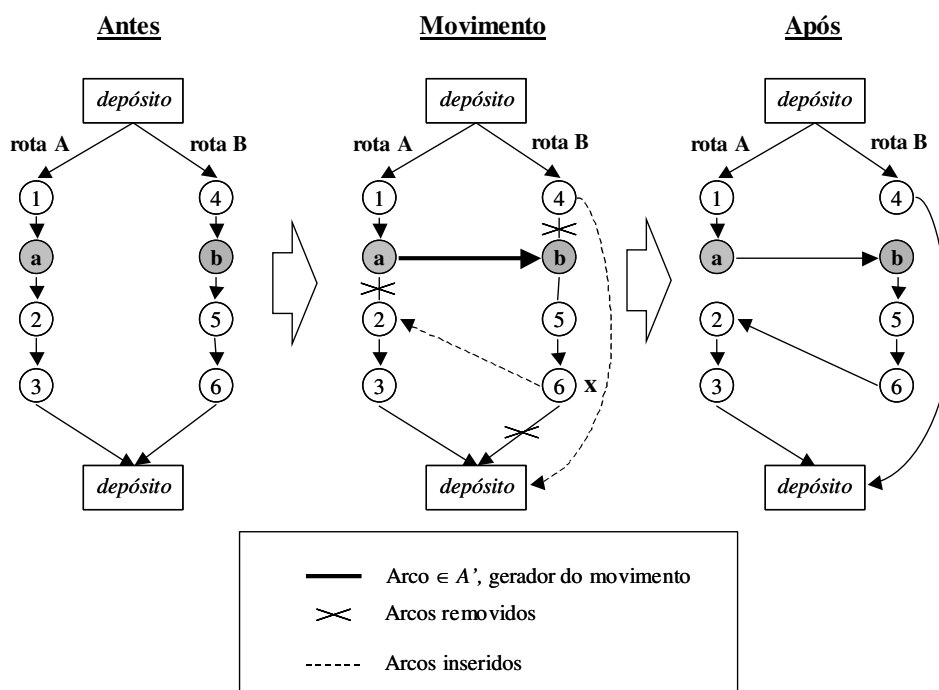


Figura 6.14. Movimento inter-rotas inserção – Opção 1 com $x = 6$

Tabela 6.9. Inter-rota inserção – Opção 2

Notação:	– Aa = cidade anterior a ‘a’ – Ab = cidade anterior a ‘b’ – Pa = cidade posterior a ‘a’ – Pb = cidade posterior a ‘b’ – x = cidade da rota A do início do trecho a ser movido, ‘x’ está entre ‘a’ (inclusive) e o depósito (exclusive) – Ax = cidade anterior a ‘x’
Arestas removidas	– (Ab, b) – (a, Pa) – (Ax, x)
Arestas inseridas	– (a, b) – (Ab, x) – (Ax, Pa)
Observações	– ‘x’ começa igual a ‘a’ e é percorrido até a cidade imediatamente posterior ao depósito (percorre no sentido inverso da rota).

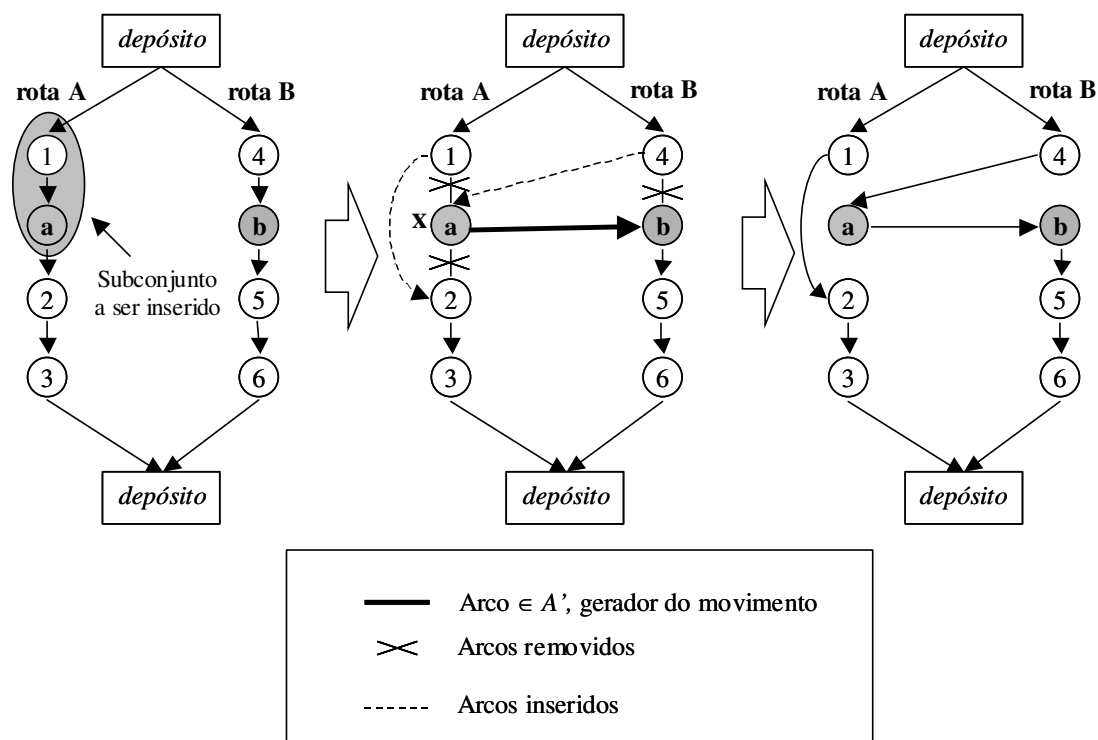


Figura 6.15. Movimento inter-rotas inserção – Opção 2 com $x = a$

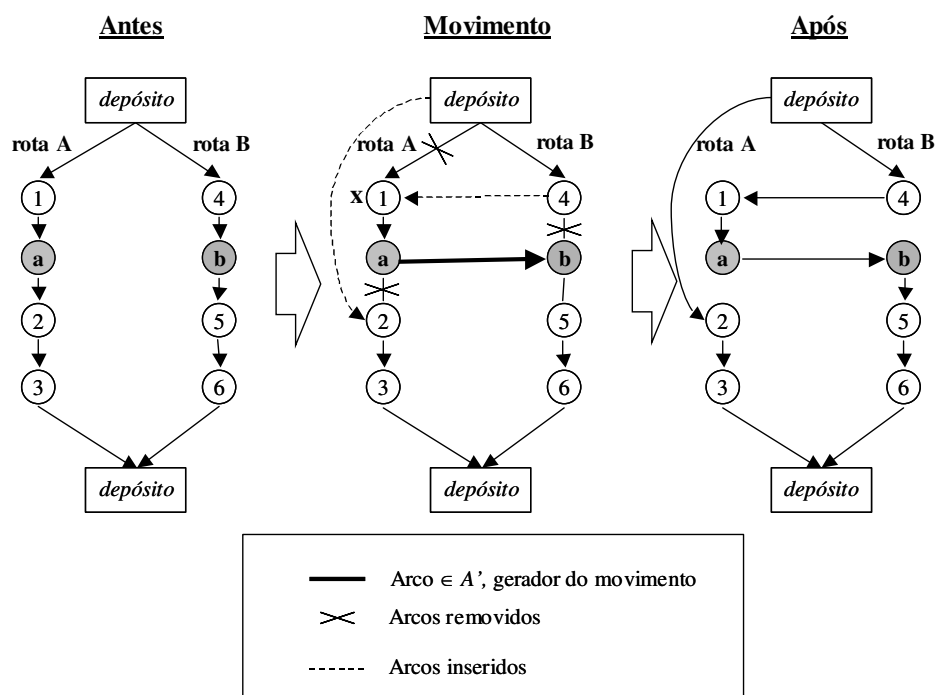


Figura 6.16. Movimento inter-rotas inserção – Opção 2 com $x = 1$

5) Movimento inter-rota troca

Este movimento troca dois trechos de rotas de duas rotas, sempre inserindo a aresta geradora (a, b). Foram implementadas duas opções de trocas.

Tabela 6.10. Inter-rota troca – Opção 1

Notação:	– Aa = cidade anterior a ‘a’ – Pa = cidade posterior a ‘a’ – fima = cidade da rota A do fim do trecho a ser movido, ‘fima’ está entre ‘Pa’ (inclusive) e o depósito (exclusive) – Pfima = cidade posterior a ‘fima’	– Ab = cidade anterior a ‘b’ – Pb = cidade posterior a ‘b’ – fimb = cidade da rota B do fim do trecho a ser movido, ‘fimb’ está entre ‘b’ (inclusive) e o depósito (exclusive) – Pfimb = cidade posterior a ‘fimb’
Arestas removidas	– (Ab, b) – (fima, Pfima)	– (a, Pa) – (fimb, Pfimb)
Arestas inseridas	– (a, b) – (fima, Pfimb)	– (Ab, Pa) – (fimb, Pfima)
Observações	– ‘fima’ começa igual a ‘Pa’ e é percorrido até a cidade imediatamente anterior ao depósito; – ‘fimb’ começa igual a ‘b’ e é percorrido até a cidade imediatamente anterior ao depósito.	

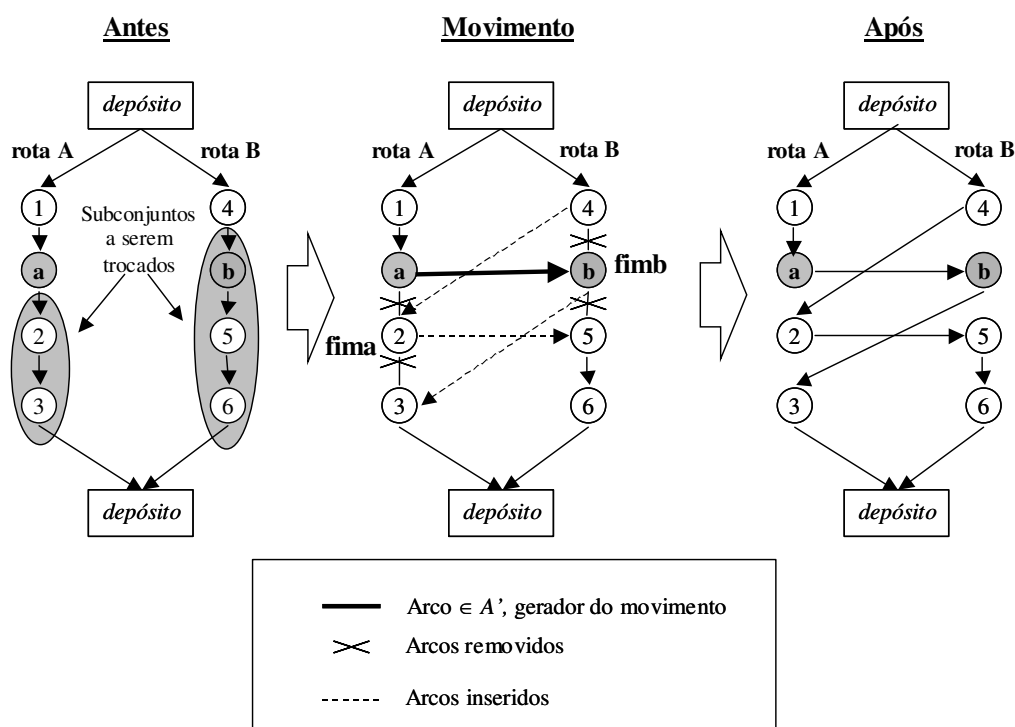


Figura 6.17. Movimento inter-rota troca – Opção 1 com $fima = Pa$ e $fimb = b$

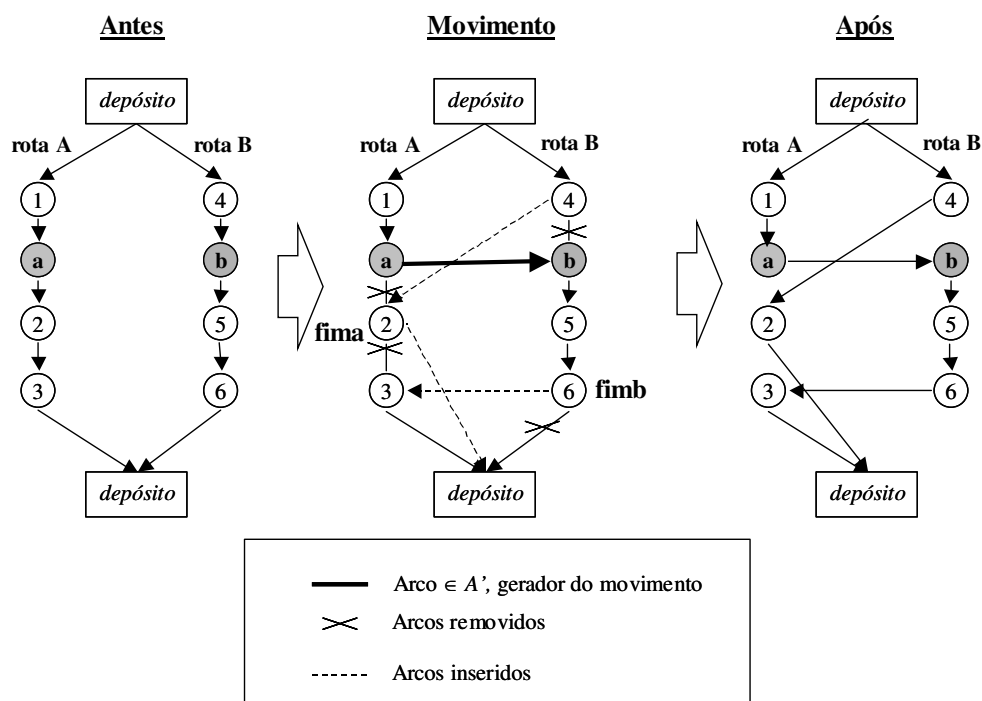


Figura 6.18. Movimento inter-rota troca – Opção 1 com $fima = 2$ e $fimb = 6$

Tabela 6.11. Inter-rota troca – Opção 2

Notação:	– Aa = cidade anterior a ‘a’ – Pa = cidade posterior a ‘a’ – inicioa = cidade da rota A do início do trecho a ser movido, ‘inicioa’ está entre ‘Aa’ (inclusive) e o depósito (exclusive) – Ainicioa = cidade anterior a ‘inicioa’ – Ab = cidade anterior a ‘b’ – Pb = cidade posterior a ‘b’ – fimb = cidade da rota B do fim do trecho a ser movido, ‘fimb’ está entre ‘b’ (inclusive) e o depósito (exclusive) – Pfimb = cidade posterior a ‘fimb’	
Arestas removidas	– (Ab, b) – (fimb, Pfimb) – (Ainicioa, inicioa) – (Aa, a) – (a, Pa)	
Arestas inseridas	– (a, b) – (Aa, Pfimb) – (fimb, Pa) – (Ab, inicioa) – (Ainicioa, a)	
Observações	– ‘inicioa’ começa igual a ‘Aa’ e é percorrido até a cidade imediatamente posterior ao depósito (percorre no sentido inverso da rota); – ‘fimb’ começa igual a ‘b’ e é percorrido até a cidade imediatamente anterior ao depósito.	

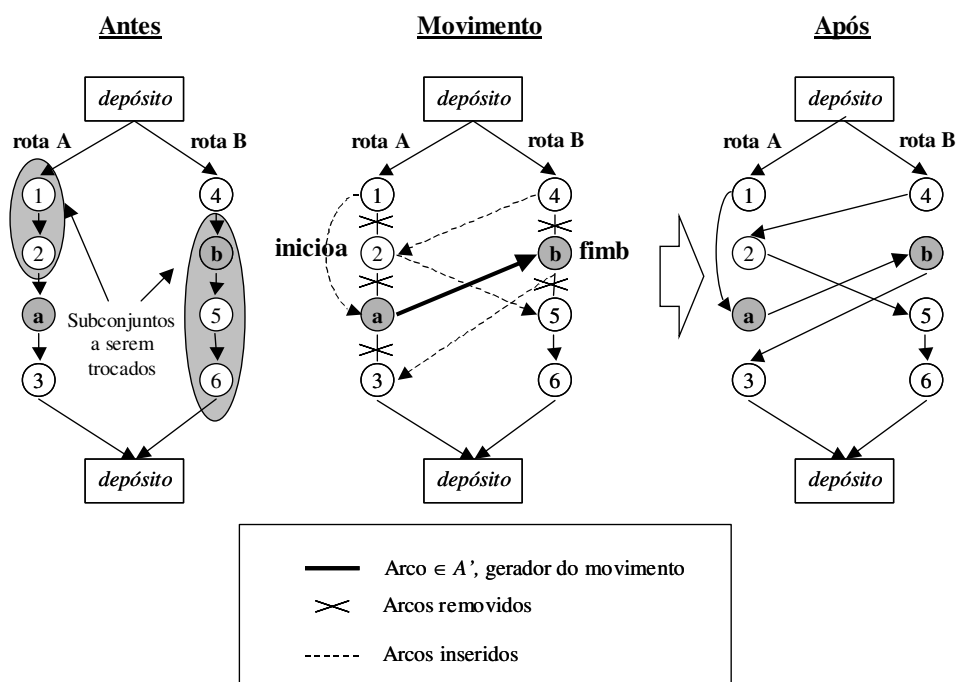


Figura 6.19. Movimento inter-rotas troca – Opção 2 com inícioa = Aa e fimb = b

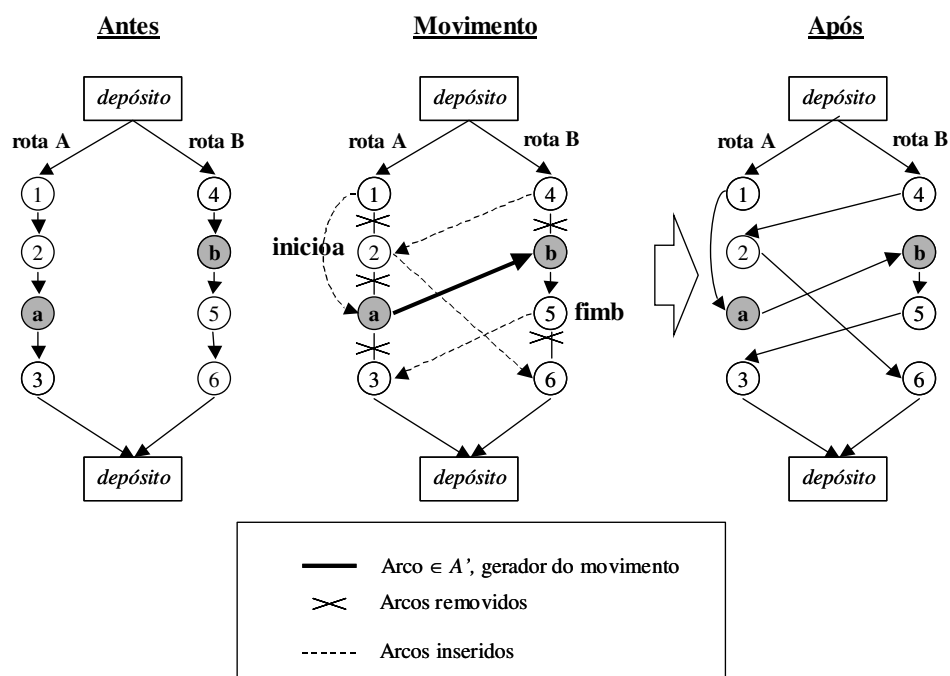


Figura 6.20. Movimento inter-rotas troca – Opção 2 com inícioa = Aa e fimb = 5

Note que os movimentos inter-rota inserção e inter-rota troca possuem em comum a inserção de trechos de uma rota em outra. Estes trechos começam com apenas um cliente e são estendidos a cada iteração com o acréscimo do cliente imediatamente posterior (ou anterior). Visando a redução de tempo computacional, a implementação destes movimentos tenta inserir (ou trocar) trechos até a iteração em que a inserção deixa de ser factível. Por exemplo, seja uma rota com os seguintes clientes (0, 1, 2, 3, 4, 5, 6, 0) e o início do trecho no cliente 2. Neste exemplo, os seguintes trechos poderiam ser movidos (*cliente_inicio_do_trecho*, *cliente_fim_do_trecho*): (2, 2); (2, 3); (2, 4); (2, 5); (2, 6). Se ao tentar inserir o trecho (2, 4) a rota resultante foi infactível, as rotas resultantes com a inserção dos trechos (2, 5) e (2, 6) também são infactíveis, sendo desnecessária a avaliação destas tentativas.

H. Vizinhança cadeias de ejeções de Glover (1991, 1996)

Cadeias de ejeções (*ejection chains*) foram propostas por Glover (1991, 1996) como um tipo de movimento entre soluções. No contexto do PRV, uma ejeção consiste de mover um nó para a posição ocupada por outro nó. Considere uma orientação de rota, seja v_{i-1} o predecessor de v_i e v_{i+1} seu sucessor, e seja (v_{i-1}, v_i, v_{i+1}) a tripla que representa dois arcos consecutivos (v_{i-1}, v_i) e (v_i, v_{i+1}) .

Uma ejeção consiste do movimento de um nó para uma nova posição ocupada por outro nó, que é desconectado de sua posição. A nova posição determina o nível da cadeia.

Seja k um nível da cadeia: em uma cadeia de ejeção cada nó v_i^k ejeta o nó v_i^{k+1} terminando com a ejeção do nó v_b^l . Como resultado, uma cadeia de ejeção de l níveis consiste em substituir as triplas $(v_{i-1}^k, v_i^k, v_{i+1}^k)$ $k = 0, 1, \dots, l$, pelas triplas $(v_{i-1}^k, v_i^{k-1}, v_{i+1}^k)$, $k = 1, 2, \dots, l$.

Seja Γ_l o conjunto de nós centrais de cada tripla, que identificam todos os l níveis da cadeia de ejeções. O nó v_b^l é inserido de duas formas:

- 1) Insira v_b^l entre o último predecessor e sucessor de v_t^0 , criando os arcos (v_{t-l}^0, v_b^l) e (v_b^l, v_{t+1}^0) .
- 2) Crie o arco (v_{t-l}^0, v_{t+1}^0) e escolha dois nós $v_p, v_q = v_{p+1} \notin \Gamma_l$. Insira v_b^l entre v_p e v_q , criando os arcos (v_p^l, v_b^l) , (v_b^l, v_q^l) e excluindo o arco (v_p, v_q) .

A Figura 6.21 ilustra uma cadeia de ejeções.

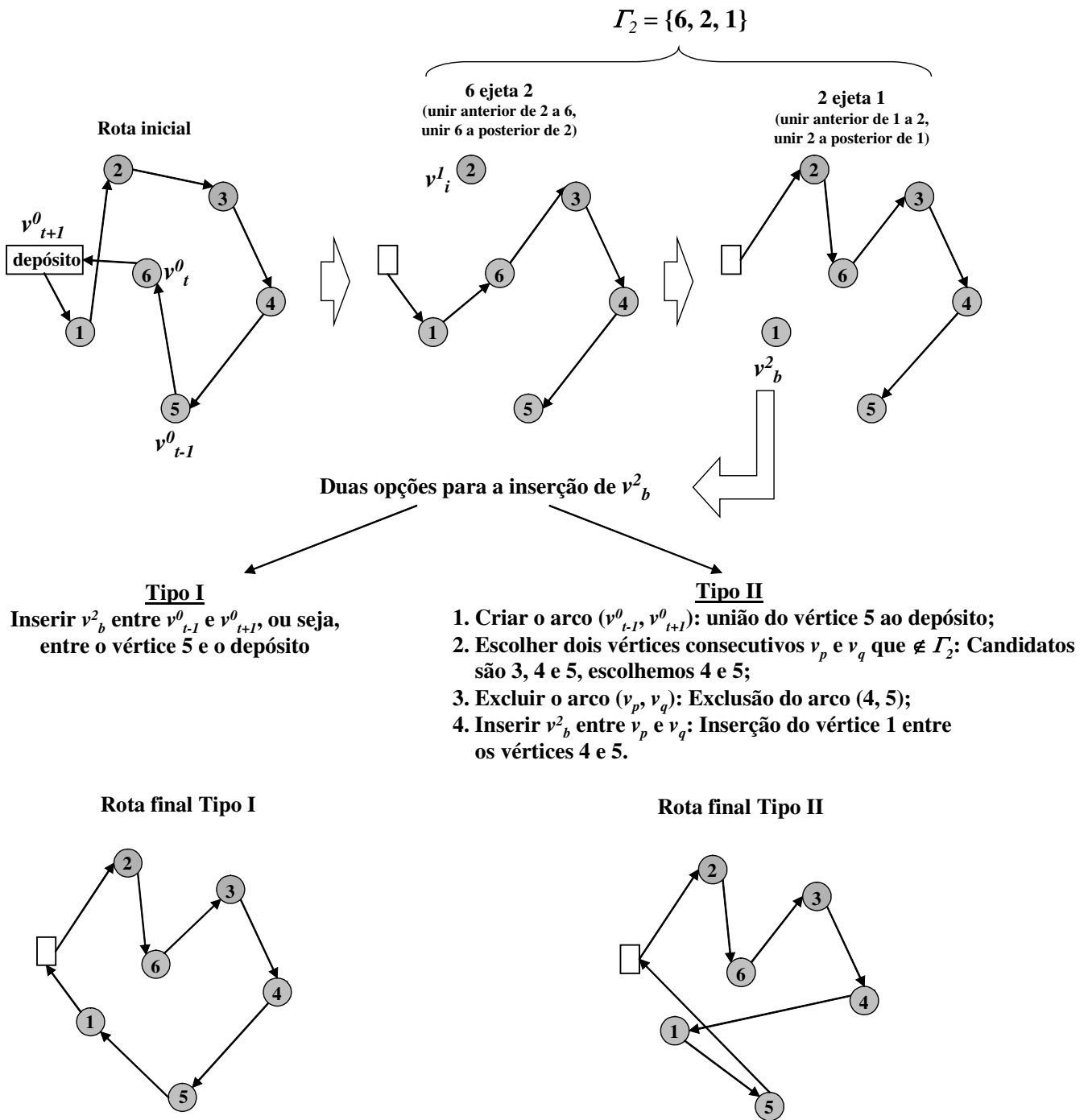


Figura 6.21. Exemplo de cadeia de ejeções para $l = 2$

I. Melhores resultados da literatura para o conjunto de problemas teste do PRVJT

Tabela 6.12. – Melhores resultados por problema teste – objetivo principal é minimizar o número de veículos

Problema teste			Melhor resultado da literatura			Resultado é ótimo?
Grupo	Nome	Número de cidades	Número de veículos	Distancia	Fonte	
C1	C101_25.txt	25	3	191.3	KDMSS	Sim
C1	C102_25.txt	25	3	190.3	KDMSS	sim
C1	C103_25.txt	25	3	190.3	KDMSS	sim
C1	C104_25.txt	25	3	186.9	KDMSS	sim
C1	C105_25.txt	25	3	191.3	KDMSS	sim
C1	C106_25.txt	25	3	191.3	KDMSS	sim
C1	C107_25.txt	25	3	191.3	KDMSS	sim
C1	C108_25.txt	25	3	191.3	KDMSS	sim
C1	C109_25.txt	25	3	191.3	KDMSS	sim
C2	C201_25.txt	25	2	214.7	CR+L	sim
C2	C202_25.txt	25	2	214.7	CR+L	sim
C2	C203_25.txt	25	2	214.7	CR+L	sim
C2	C204_25.txt	25	2	213.1	CR+KLM	sim
C2	C205_25.txt	25	2	214.7	CR+L	sim
C2	C206_25.txt	25	2	214.7	CR+L	sim
C2	C207_25.txt	25	2	214.5	CR+L	sim
C2	C208_25.txt	25	2	214.5	CR+L	sim
R1	R101_25.txt	25	8	617.1	KDMSS	sim
R1	R102_25.txt	25	7	547.1	KDMSS	sim
R1	R103_25.txt	25	5	454.6	KDMSS	sim
R1	R104_25.txt	25	4	416.9	KDMSS	sim
R1	R105_25.txt	25	6	530.5	KDMSS	sim
R1	R106_25.txt	25	3	465.4	KDMSS	sim
R1	R107_25.txt	25	4	424.3	KDMSS	sim
R1	R108_25.txt	25	4	397.3	KDMSS	sim
R1	R109_25.txt	25	5	441.3	KDMSS	sim
R1	R110_25.txt	25	4	444.1	KDMSS	sim
R1	R111_25.txt	25	5	428.8	KDMSS	sim
R1	R112_25.txt	25	4	393	KDMSS	sim
R2	R201_25.txt	25	4	463.3	CR+KLM	sim
R2	R202_25.txt	25	4	410.5	CR+KLM	sim
R2	R203_25.txt	25	3	391.4	CR+KLM	sim
R2	R204_25.txt	25	2	355	C	sim
R2	R205_25.txt	25	3	393	CR+KLM	sim
R2	R206_25.txt	25	3	374.4	CR+KLM	sim
R2	R207_25.txt	25	3	361.6	KLM	sim

Problema teste			Melhor resultado da literatura			Resultado é ótimo?
Grupo	Nome	Número de cidades	Número de veículos	Distancia	Fonte	
R2	R208_25.txt	25	1	328.2	C	sim
R2	R209_25.txt	25	2	370.7	KLM	sim
R2	R210_25.txt	25	3	404.6	CR+KLM	sim
R2	R211_25.txt	25	2	350.9	KLM	sim
RC1	RC101_25.txt	25	4	461.1	KDMSS	sim
RC1	RC102_25.txt	25	3	351.8	KDMSS	sim
RC1	RC103_25.txt	25	3	332.8	KDMSS	sim
RC1	RC104_25.txt	25	3	306.6	KDMSS	sim
RC1	RC105_25.txt	25	4	411.3	KDMSS	sim
RC1	RC106_25.txt	25	3	345.5	KDMSS	sim
RC1	RC107_25.txt	25	3	298.3	KDMSS	sim
RC1	RC108_25.txt	25	3	294.5	KDMSS	sim
RC2	RC201_25.txt	25	3	360.2	CR+L	sim
RC2	RC202_25.txt	25	3	338	CR+KLM	sim
RC2	RC203_25.txt	25	3	326.9	C	sim
RC2	RC204_25.txt	25	3	299.7	C	sim
RC2	RC205_25.txt	25	3	338	L+KLM	sim
RC2	RC206_25.txt	25	3	324	KLM	sim
RC2	RC207_25.txt	25	3	298.3	KLM	sim
RC2	RC208_25.txt	25	2	269.1	C	sim
C1	C101_50.txt	50	5	362.4	KDMSS	sim
C1	C102_50.txt	50	5	361.4	KDMSS	sim
C1	C103_50.txt	50	5	361.4	KDMSS	sim
C1	C104_50.txt	50	5	358	KDMSS	sim
C1	C105_50.txt	50	5	362.4	KDMSS	sim
C1	C106_50.txt	50	5	362.4	KDMSS	sim
C1	C107_50.txt	50	5	362.4	KDMSS	sim
C1	C108_50.txt	50	5	362.4	KDMSS	sim
C1	C109_50.txt	50	5	362.4	KDMSS	sim
C2	C201_50.txt	50	3	360.2	CR+L	sim
C2	C202_50.txt	50	3	360.2	CR+KLM	sim
C2	C203_50.txt	50	3	359.8	CR+KLM	sim
C2	C204_50.txt	50	2	350.1	KLM	sim
C2	C205_50.txt	50	3	359.8	CR+KLM	sim
C2	C206_50.txt	50	3	359.8	CR+KLM	sim
C2	C207_50.txt	50	3	359.6	CR+KLM	sim
C2	C208_50.txt	50	2	350.5	CR+KLM	sim
R1	R101_50.txt	50	12	1044	KDMSS	sim
R1	R102_50.txt	50	11	909	KDMSS	sim
R1	R103_50.txt	50	9	772.9	KDMSS	sim
R1	R104_50.txt	50	6	625.4	KDMSS	sim
R1	R105_50.txt	50	9	899.3	KDMSS	sim
R1	R106_50.txt	50	5	793	KDMSS	sim
R1	R107_50.txt	50	7	711.1	KDMSS	sim
R1	R108_50.txt	50	6	617.7	CR+KLM	sim

Problema teste			Melhor resultado da literatura			Resultado é ótimo?
Grupo	Nome	Número de cidades	Número de veículos	Distancia	Fonte	
R1	R109_50.txt	50	8	786.8	KDMSS	sim
R1	R110_50.txt	50	7	697	KDMSS	sim
R1	R111_50.txt	50	7	707.2	CR+KLM	sim
R1	R112_50.txt	50	6	630.2	CR+KLM	sim
R2	R201_50.txt	50	6	791.9	CR+KLM	sim
R2	R202_50.txt	50	5	698.5	CR+KLM	sim
R2	R203_50.txt	50	5	605.3	C	sim
R2	R205_50.txt	50	4	690.1	C	sim
R2	R206_50.txt	50	4	632.4	C	sim
R2	R209_50.txt	50	4	600.6	C	sim
R2	R210_50.txt	50	4	645.6	C	sim
RC1	RC101_50.txt	50	8	944	KDMSS	sim
RC1	RC102_50.txt	50	7	822.5	KDMSS	sim
RC1	RC103_50.txt	50	6	710.9	KDMSS	sim
RC1	RC104_50.txt	50	5	545.8	KDMSS	sim
RC1	RC105_50.txt	50	8	855.3	KDMSS	sim
RC1	RC106_50.txt	50	6	723.2	KDMSS	sim
RC1	RC107_50.txt	50	6	642.7	KDMSS	sim
RC1	RC108_50.txt	50	6	598.1	KDMSS	sim
RC2	RC201_50.txt	50	5	684.8	L+KLM	sim
RC2	RC202_50.txt	50	5	613.6	C	sim
RC2	RC203_50.txt	50	4	555.3	C	sim
RC2	RC205_50.txt	50	5	630.2	C	sim
RC2	RC206_50.txt	50	5	610	C	sim
RC2	RC207_50.txt	50	4	558.6	C	sim
C1	C101_100.txt	100	10	827.3	KDMSS	sim
C1	C102_100.txt	100	10	827.3	KDMSS	sim
C1	C103_100.txt	100	10	826.3	KDMSS	sim
C1	C104_100.txt	100	10	822.9	KDMSS	sim
C1	C105_100.txt	100	10	827.3	KDMSS	sim
C1	C106_100.txt	100	10	827.3	KDMSS	sim
C1	C107_100.txt	100	10	827.3	KDMSS	sim
C1	C108_100.txt	100	10	827.3	KDMSS	sim
C1	C109_100.txt	100	10	827.3	KDMSS	sim
C2	C201_100.txt	100	3	589.1	CR+KLM	sim
C2	C202_100.txt	100	3	589.1	CR+KLM	sim
C2	C203_100.txt	100	3	588.7	KLM	sim
C2	C204_100.txt	100	3	590.6	RT	não
C2	C205_100.txt	100	3	586.4	CR+KLM	sim
C2	C206_100.txt	100	3	586	CR+KLM	sim
C2	C207_100.txt	100	3	585.8	CR+KLM	sim
C2	C208_100.txt	100	3	585.8	KLM	sim
R1	R101_100.txt	100	19	1645.79	H	não
R1	R102_100.txt	100	17	1486.12	RT	não
R1	R103_100.txt	100	13	1292.68	LLH	não

Problema teste			Melhor resultado da literatura			Resultado é ótimo?
Grupo	Nome	Número de cidades	Número de veículos	Distancia	Fonte	
R1	R104_100.txt	100	9	1007.24	M	não
R1	R105_100.txt	100	14	1377.11	RT	não
R1	R106_100.txt	100	12	1251.98	M	não
R1	R107_100.txt	100	10	1104.66	S97	não
R1	R108_100.txt	100	9	960.88	BBB	não
R1	R109_100.txt	100	11	1194.73	HG	não
R1	R110_100.txt	100	10	1118.59	M	não
R1	R111_100.txt	100	10	1096.72	RGP	não
R1	R112_100.txt	100	9	982.14	GTA	não
R2	R201_100.txt	100	4	1252.37	HG	não
R2	R202_100.txt	100	3	1191.7	RGP	não
R2	R203_100.txt	100	3	939.54	M	não
R2	R204_100.txt	100	2	825.52	BVH	não
R2	R205_100.txt	100	3	994.42	RGP	não
R2	R206_100.txt	100	3	906.14	SSSD	não
R2	R207_100.txt	100	2	893.33	BVH	não
R2	R208_100.txt	100	2	726.75	M	não
R2	R209_100.txt	100	3	909.16	H	não
R2	R210_100.txt	100	3	939.34	M	não
R2	R211_100.txt	100	2	892.71	BVH	não
RC1	RC101_100.txt	100	14	1696.94	TBGGP	não
RC1	RC102_100.txt	100	12	1554.75	TBGGP	não
RC1	RC103_100.txt	100	11	1258	CR+KLM	sim
RC1	RC104_100.txt	100	10	1135.48	CLM	não
RC1	RC105_100.txt	100	13	1629.44	BBB	não
RC1	RC106_100.txt	100	11	1424.73	BBB	não
RC1	RC107_100.txt	100	11	1230.48	S97	não
RC1	RC108_100.txt	100	10	1139.82	TBGGP	não
RC2	RC201_100.txt	100	4	1406.91	M	não
RC2	RC202_100.txt	100	3	1367.09	CC	não
RC2	RC203_100.txt	100	3	1049.62	CC	não
RC2	RC204_100.txt	100	3	798.41	M	não
RC2	RC205_100.txt	100	4	1297.19	M	não
RC2	RC206_100.txt	100	3	1146.32	H	não
RC2	RC207_100.txt	100	3	1061.14	BVH	não
RC2	RC208_100.txt	100	3	828.14	IKMUY	não

Tabela 6.13. – Descrição das fontes da Tabela 6.8.1.

Sigla	Fonte
KLM	B. Kallehauge, J. Larsen, and O.B.G. Madsen. "Lagrangian duality and non-differentiable optimization applied on routing with time windows - experimental results." Internal report IMM-REP-2000-8, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2000.
BBB	J. Berger, M. Barkaoui and O. Bräysy, "A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows," Working paper, Defense Research Establishment Valcartier, Canada, 2001.
BVH	R. Bent and P. Van Hentenryck, "A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows," Technical Report CS-01-06, Department of Computer Science, Brown University, 2001.
C	A. Chabrier, "Vehicle Routing Problem with Elementary Shortest Path based Column Generation," Working Paper, ILOG, Madrid, Spain, 2002.
CC	Z. J. Czech and P. Czarnas, "A Parallel Simulated Annealing for the Vehicle Routing Problem with Time Windows," Proc. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, (January 9--11, 2002), 376--383.
CLM	J.-F. Cordeau, G. Laporte, and A. Mercier, "A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows," Working Paper CRT-00-03, Centre for Research on Transportation, Montreal, Canada, 2000.
CR+KLM	W. Cook and J. L. Rich, "A parallel cutting plane algorithm for the vehicle routing problem with time windows," Working Paper, Computational and Applied Mathematics, Rice University, Houston, TX, 1999. B. Kallehauge, J. Larsen, and O.B.G. Madsen. "Lagrangian duality and non-differentiable optimization applied on routing with time windows - experimental results." Internal report IMM-REP-2000-8, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2000.
CR+L	W. Cook and J. L. Rich, "A parallel cutting plane algorithm for the vehicle routing problem with time windows," Working Paper, Computational and Applied Mathematics, Rice University, Houston, TX, 1999. J. Larsen. "Parellellization of the vehicle routing problem with time windows." Ph.D. Thesis IMM-PHD-1999-62, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1999.
GTA	L. M. Gambardella, E. Taillard, and G. Agazzi, "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows," in New Ideas in Optimization, D. Corne, M. Dorigo and F. Glover (eds), 63-76, McGraw-Hill, London, 1999.
H	J. Homberger, "Verteilt-parallele Metaheuristiken zur Tourenplanung," Gaber, Wiesbaden (2000).
HG	J. Homberger and H. Gehring, "Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows," INFOR, VOL. 37, 297-318, (1999).
IKMUJ	T. Ibaraki, M. Kubo, T. Masuda, T. Uno and M. Yagiura, "Effective Local Search Algorithms for the Vehicle Routing Problem with General Time Windows," Transportation Science 39 (2): 206-232, (2005).
KDMSS	N. Kohl, J. Desrosiers, O. B. G. Madsen, M. M. Solomon, and F. Soumis, "2-Path Cuts for the Vehicle Routing Problem with Time Windows," Transportation Science, Vol. 33 (1), 101-116 (1999).
L+KLM	J. Larsen. "Parellellization of the vehicle routing problem with time windows." Ph.D. Thesis

Sigla	Fonte
	IMM-PHD-1999-62, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1999. B. Kallehauge, J. Larsen, and O.B.G. Madsen. "Lagrangian duality and non-differentiable optimization applied on routing with time windows - experimental results." Internal report IMM-REP-2000-8, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2000.
LLH	H. Li, A. Lim, and J. Huang, "Local Search with Annealing-like Restarts to Solve the VRPTW," Working Paper, Department of Computer Science, National University of Singapore, 2001.
M	D. Mester, "An Evolutionary Strategies Algorithm for Large Scale Vehicle Routing Problem with Capacitate and Time Windows Restrictions," Working Paper, Institute of Evolution, University of Haifa, Israel (2002).
RGP	L.M. Rousseau, M. Gendreau and G. Pesant, "Using Constraint-Based Operators to Solve the Vehicle Routing Problem with Time Windows," Journal of Heuristics, forthcoming.
RT	Y. Rochat and E.D. Taillard, "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing," Journal of Heuristics 1, 147-167, (1995).
S97	P. Shaw, "A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems," Working Paper, University of Strathclyde, Glasgow, Scotland, 1997.
SSSD	G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt and G. Dueck, "Record Breaking Optimization Results Using the Ruin and Recreate Principle," Journal of Computational Physics 159, 139-171, (2000).
TBGGP	E. Taillard, P. Badeau, M. Gendreau, F. Geurtin, and J.Y. Potvin, "A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows," Transportation Science, 31, 170-186, (1997).

J. Estimativa dos custos fixo e variável de um veículo

O objetivo deste estudo é chegar a um custo estimado de reais por quilômetro rodado (custo variável) e outro custo de reais por veículo utilizado (custo fixo). Adotamos no nosso estudo as premissas da Tabela 6.14:

Tabela 6.14. Premissas da análise de custo fixo e variável

Velocidade média (km/h)	40	km/h
Km média anual percorrida pelo veículo <i>(por um único motorista)</i>	44,880	km
Valor do salário mínimo	260	R\$ / mês
Salário motorista (número salários mínimos)	4	salários mínimos
Custo motorista / Salário motorista <i>(inclui encargos, 13o, férias, FGTS, benefícios)</i>	2.1	
Custo mensal de oportunidade	0.50%	
número de dias úteis no mês	22	
Custo do óleo diesel (R\$ / L)	1.46	R\$ / L
Rendimento do veículo	6	km / L
Qtde de litros de óleo no motor	13	L
Periodicidade de troca de óleo	5000	km
Preço por litro de óleo (R\$)	8	R\$
Custo com lavagens e graxas (R\$/mês)	60	R\$/mês
Pneus		
Custo por pneu	331.57	R\$ / pneu
Número de pneus	4	
Custo pneus	1326.28	R\$
Periodicidade de troca dos pneus	30000	km
Custo médio anual de manutenção e reparos	3732.24	R\$ / ano
Outras despesas indiretas	20%	
Valor de mercado do veículo	75000	R\$
Custo seguro obrigatório e licenciamento	1635	R\$

Foi considerado que o veículo é comprado no primeiro ano por R\$ 75.000,00. O veículo é utilizado por 10 anos, com um custo de R\$ 1.635,00 por ano. Após os 10 anos de uso o veículo é vendido por um valor residual de R\$ 7.500,00.

Para cobrir todos estes gastos são necessários R\$ 83.553,00 no instante atual de operação, assumindo a disponibilidade de um investimento que renda no mínimo 0,5% ao mês sobre a inflação.

O valor de R\$ 83.553,00 é equivalente a 120 parcelas de R\$ 922,99, descontadas a 0,5% ao mês. Dividindo este valor por 22 dias úteis, temos como resultado R\$ 41,95 / dia. Acrescentando a este resultado os custos fixos do motorista (salário + encargos), deriva o custo fixo de R\$ 141,23, como demonstra a Tabela 6.15.

Tabela 6.15. Cálculo do custo fixo

Custo		R\$ / mês	R\$ /dia
Salário motorista		1.040,00	47,27
Encargos e benefícios		1.144,00	52,00
Total custo motorista		2.184,00	99,27
Custo fixo veículo		922,99	41,95
Total custo fixo		3.106,99	141,23

Assumindo a velocidade média do veículo como de 40 km / hora, um turno de no máximo nove horas (oito horas de trabalho mais uma hora de almoço, como determina a CLT – Consolidação das Leis Trabalhistas) e um roteiro com 15 paradas de 15 minutos em clientes, temos numa quilometragem média diária de 170 km, como calculado na Tabela 6.16.

Tabela 6.16. Estimativa da quilometragem média diária por veículo

Total de horas no turno:	9,00
(-) tempo para almoço	-1,00
(-) tempo para atender 15 clientes	-3,75
(=) horas do veículo em trânsito	4,25
Assumindo velocidade média de	40 km/h
Quilometragem média diária	170,00

A partir desta quilometragem diária, podemos calcular o custo variável do veículo, mostrado na Tabela 6.17.

Tabela 6.17. Cálculo do custo variável

Custo variável	R\$/km
Combustível	0,24
Lubrificantes	0,02
Lavagens e graxas	0,02
Pneus	0,04
Manutenção	0,08
Subtotal	0,41
Outras despesas	0,08
<i>Total custo variável</i>	<i>0,49</i>

Finalmente, chegamos a um custo fixo de R\$ 141,23 por veículo mais um custo variável de R\$ 0,49 / km rodado. A Tabela 6.18 mostra o resumo da análise de custo de transporte.

Tabela 6.18. Resumo do custo de um veículo

Custo	R\$/dia	R\$/km
Fixo	141,23	0,83
Variável	83,14	0,49

A partir destes resultados, podemos calcular o ponto de equilíbrio de um veículo, ou seja, quantos clientes um veículo deve atender para que sua contribuição para o lucro da empresa seja positiva. Considerando que o veículo percorre 170 km para atender 15 clientes, temos uma quilometragem média por cliente de 11,33 km / cliente. A Tabela 6.19 apresenta a contribuição ao lucro por número de clientes atendidos:

Tabela 6.19. Contribuição ao lucro por número de clientes atendidos (em R\$)

Número de clientes	Receita	Custos			Contribuição ao lucro
		Fixo	Variável	Total	
0	0,00	282,45	-	282,45	(282,45)
1	200,00	282,45	11,09	293,54	(93,54)
2	400,00	282,45	22,17	304,62	95,38
3	600,00	282,45	33,26	315,71	284,29
...					
15	3.000,00	282,45	166,28	448,73	2,551,27
16	3.200,00	282,45	177,36	459,82	2,740,18
17	3.400,00	282,45	188,45	470,90	2,929,10

Note que os custos da Tabela 6.19 estão multiplicados por dois, pois para se atender um cliente é necessário coletar a mercadoria num dia e num endereço de coleta para entregá-la em outro local no dia seguinte. A conclusão é que são precisos dois clientes num roteiro para que haja contribuição positiva. Todavia, vale salientar que não foram incluídos os custos administrativos e de movimentação e armazenagem.

K. Pseudocódigos

K.1. Construção do grafo granular:

1. Leia o parâmetro η = percentual de arcos que serão inseridos no grafo granular;
2. Faça $num_arestas_no_grafo = 0$;
3. Calcule o $num_total_de_arcos_potenciais = n^2 - n$;
4. Gere todos os arcos potenciais (conjunto total de arcos menos os arcos que originam ou terminam no depósito);
5. Ordene os arcos potenciais em ordem crescente de distância;
6. A partir do primeiro da lista, faça:
 - 6.1. Se $(num_arestas_no_grafo + 1) \div (num_total_de_arcos_potenciais / 2) \geq \eta$ vá para o passo 7. Caso contrário, insira a aresta no grafo, faça $num_arestas_no_grafo = num_arestas_no_grafo + 1$, e volte ao passo 6.1;
7. Acrescente ao grafo_granular todos as arestas que originam e terminam no depósito;
8. FIM.

K.2. Busca local no grafo granular:

1. Faça $melhor_ganho = 0$ e $melhor_movimento = \text{NULL}$;
2. Para cada aresta (a, b) do grafo granular faça:
 - 2.1. Calcule o ganho das duas opções do movimento intra-rota troca-arcos e armazene o melhor ganho encontrado na variável $ganho$. Se $ganho > melhor_ganho$, faça $melhor_ganho = ganho$ e $melhor_movimento = \text{intra-rota troca-arcos}$
 - 2.2. Calcule o ganho do movimento intra-rota inserção e armazene o melhor ganho encontrado na variável $ganho$. Se $ganho > melhor_ganho$, faça $melhor_ganho = ganho$ e $melhor_movimento = \text{intra-rota inserção}$;
 - 2.3. Calcule o ganho das 4 opções do movimento inter-rota *crossover* e armazene o melhor ganho encontrado na variável $ganho$. Se $ganho > melhor_ganho$, faça $melhor_ganho = ganho$ e $melhor_movimento = \text{inter-rota crossover}$;
 - 2.4. Calcule o ganho do movimento inter-rota inserção e armazene o melhor ganho encontrado na variável $ganho$. Se $ganho > melhor_ganho$, faça $melhor_ganho = ganho$ e $melhor_movimento = \text{inter-rota inserção}$;
 - 2.5. Calcule o ganho do movimento inter-rota troca e armazene o melhor ganho encontrado na variável $ganho$. Se $ganho > melhor_ganho$, faça $melhor_ganho = ganho$ e $melhor_movimento = \text{inter-rota troca}$;
 - 2.6. Se $melhor_ganho > 0$ e a busca local é *first-improvement*, vá ao passo 3. Caso contrário, selecione a próxima aresta da lista do grafo granular e volte ao passo 2.1. Se todas as arestas da lista já foram selecionadas, vá para o passo 3.
3. Se $melhor_ganho > 0$ efetue o movimento armazenado em $melhor_movimento$ e retorne ao passo 1. Caso contrário pare.
4. FIM.

K.3. Regras de entrada de uma nova solução no conjunto de elite

Para uma solução s_{nova} entrar no conjunto de elite, as seguintes regras são analisadas:

Notação:

$cont_soluções_inseridas$ = número de soluções que já foram inseridas no conjunto de elite;

$tamanho_conjunto$ = número máximo de soluções a serem armazenadas no conjunto;

E = conjuntos das soluções de elite;

$melhor_fo$ = valor da função objetivo da melhor solução do conjunto;

$menor_num_veículos$ = quantidade de veículos da melhor solução do conjunto;

$pior_fo$ = valor da função objetivo da pior solução do conjunto;

$valor_fo(s)$ = valor da função objetivo da solução s ;

$num_veículos(s)$ = número de veículos da solução s ;

$distância(s_1, s_2)$ = distância entre uma solução s_1 e outra solução s_2 ;

$menor_distância_atual_entre_duas_soluções_do_conjunto$ = $\min(distância(s_1, s_2)), \forall s_1, s_2 \in E, s_1 \neq s_2$;

$distância_média_atual_de_todas_soluções_do_conjunto$ = $média(distância(s_1, s_2)), \forall s_1, s_2 \in E, s_1 \neq s_2$;

Regras:

1. Se $cont_soluções_inseridas = 0$, vá para o passo 4;
2. Se $0 < cont_soluções_inseridas < tamanho_conjunto$:
 - 2.1. Se $num_veículos(s_{nova}) < menor_num_veículos$ ou $[num_veículos(s_{nova}) = menor_num_veículos \text{ e } valor_fo(s_{nova}) < melhor_fo]$, vá para o passo 4;
 - 2.2. Calcule $menor_distância = \min(distância(s_{nova}, s)), \forall s \in E$;
 - 2.3. Se $menor_distância > 0.05$, vá para o passo 4, caso contrário $\rightarrow$ FIM.
3. Se $cont_soluções_inseridas \geq tamanho_conjunto$:
 - 3.1. Faça $entrar = \text{falso}$;
 - 3.2. Se $num_veículos(s_{nova}) < menor_num_veículos$ ou $[num_veículos(s_{nova}) = menor_num_veículos \text{ e } valor_fo(s_{nova}) < melhor_fo]$, faça $entrar = \text{verdadeiro}$;
 - 3.3. Se $valor_fo(s_{nova}) > pior_fo$:

-
- 3.3.1. Calcule a *menor_distância_entre_duas_soluções_do_conjunto* e a *distância_média_de_todas_soluções_do_conjunto* supondo que *s_nova* substitui a primeira solução de *E*. Faça o mesmo supondo que *s_nova* substitui cada uma das outras soluções de *E*. Neste processo, armazene qual substituição resultaria na maior *menor_distância_entre_duas_soluções_do_conjunto** e qual substituição resultaria na maior *distância_média_de_todas_soluções_do_conjunto**;
 - 3.3.2. Se *menor_distância_entre_duas_soluções_do_conjunto** for maior que a *menor_distância_atual_entre_duas_soluções_do_conjunto*, faça *entrar* = verdadeiro;
 - 3.3.3. Se *distância_média_de_todas_soluções_do_conjunto** for maior que a *distância_média_atual_de_todas_soluções_do_conjunto*, faça *entrar* = verdadeiro;
 - 3.4. Se *entrar* = verdadeiro, faça $E = E / s_{sai}$, onde *s_sai* é a solução que quando substituída por *s_nova* gera o maior incremento das medidas *menor_distância_atual_entre_duas_soluções_do_conjunto*, ou, *distância_média_atual_de_todas_soluções_do_conjunto*. Vá para o passo 4.
 - 3.5. Se *entrar* = falso → FIM.
 4. $E = E \cup s_{nova}$ e *cont_soluções_inseridas* = *cont_soluções_inseridas* + 1;
 5. Atualize, caso necessário, *melhor_fo*, *menor_num_veículos*, *pior_fo*, *menor_distância_atual_entre_duas_soluções_do_conjunto*, *distância_média_atual_de_todas_soluções_do_conjunto*;
 6. FIM.

L. Relatório gerado pelo sistema de rastreamento – roteiro de um veículo

Control Fleet Sistema de Monitoramento e Rastreamento em Tempo Real				Relatório de Jornada			Página: 1	
Veículo:	301331	Placa:	GOA-3607					
Período Inicial:	20/12/2004	Período Final:	21/12/2004					
Evento	Data Hora	Tempo	Distância percorrida	Velocidade Média				
01-SAIINDO DA BASE (CARREGADO) (1º Destino: rebesat)	20/12/2004 11:35:25	-----	0,00 KM	-----				
CHEGADA NO DESTINO (Cliente: rebesat)	20/12/2004 12:49:59	01:14:34	42,05 KM	28,89				
04-PROXIMO DESTINO (Seguindo Para: mlis)	20/12/2004 13:11:22	00:21:23	1,10 KM	2,13				
CHEGADA NO DESTINO (Cliente: mlis)	20/12/2004 13:24:21	00:12:59	6,54 KM	25,44				
04-PROXIMO DESTINO (Seguindo Para: payo)	20/12/2004 13:38:55	00:14:34	0,49 KM	1,02				
CHEGADA NO DESTINO (Cliente: payo)	20/12/2004 14:02:22	00:23:27	31,29 KM	74,22				
CHEGADA NO DESTINO (Seguindo Para: matrik)	20/12/2004 14:11:32	00:09:10	0,42 KM	1,73				
04-PROXIMO DESTINO (Seguindo Para: matrik)	20/12/2004 14:17:41	00:06:09	4,36 KM	35,74				
CHEGADA NO DESTINO (Cliente: yezzo)	20/12/2004 14:38:35	00:20:54	0,89 KM	1,75				
04-PROXIMO DESTINO (Seguindo Para: yezzo)	20/12/2004 14:44:51	00:06:16	2,87 KM	22,81				
CHEGADA NO DESTINO (Cliente: rehen)	20/12/2004 14:56:17	00:11:26	0,87 KM	2,03				
04-PROXIMO DESTINO (Seguindo Para: rehen)	20/12/2004 14:57:47	00:01:30	0,65 KM	16,43				
CHEGADA NO DESTINO (Cliente: terrol)	20/12/2004 15:02:29	00:04:42	0,26 KM	1,47				
04-PROXIMO DESTINO (Seguindo Para: terrol)	20/12/2004 15:11:38	00:09:09	9,61 KM	53,52				
CHEGADA NO DESTINO (Cliente: terrol)	20/12/2004 15:29:40	00:18:02	0,75 KM	1,46				
04-PROXIMO DESTINO (Seguindo Para: pia)	20/12/2004 16:05:42	00:33:24	1,90 KM	36,55				
CHEGADA NO DESTINO (Cliente: pia)	20/12/2004 16:16:35	00:10:53	10,18 KM	48,00				
04-PROXIMO DESTINO (Seguindo Para: ariston)	20/12/2004 17:20:43	01:04:08	1,21 KM	0,83				
CHEGADA NO DESTINO (Seguindo Para: blauslegel)	20/12/2004 17:35:08	00:14:25	8,30 KM	29,68				
04-PROXIMO DESTINO (Seguindo Para: vecchio)	20/12/2004 17:56:57	00:21:49	1,16 KM	1,99				
CHEGADA NO DESTINO (Cliente: vecchio)	20/12/2004 17:58:20	00:01:23	0,59 KM	23,17				
04-PROXIMO DESTINO (Seguindo Para: polyparts)	20/12/2004 18:08:26	00:10:06	1,05 KM	3,73				
CHEGADA NO DESTINO (Cliente: polyparts)	20/12/2004 18:32:07	00:23:41	25,75 KM	58,47				
04-PROXIMO DESTINO (Seguindo Para: empório In)	20/12/2004 18:49:41	00:17:34	1,51 KM	4,24				
CHEGADA NO DESTINO (Cliente: empório)	20/12/2004 19:09:43	00:20:02	23,66 KM	63,59				
04-PROXIMO DESTINO (Seguindo Para: atlas)	20/12/2004 20:38:02	01:28:19	2,88 KM	0,78				

AutoFax Control

21/12/2004

Control Fleet Sistema de Monitoramento e Rastreamento em Tempo Real					Página : 2
Relatório de Jornada					
Evento	Data Hora	Tempo	Distância Percorrida	Velocidade Média	
09-CHEGADA NA ATLAS (:)	20/12/2004 21:46:27	01:08:25 10:11:02	50,27 KM 232,77 KM	39,54 22,46	

AutoFax Control

21/12/2004