

**ALGORITMO ONLINE PARA O PROBLEMA
DINÂMICO DE ROTEAMENTO DE VEÍCULOS**

HUMBERTO CÉSAR BRANDÃO DE OLIVEIRA

**ALGORITMO ONLINE PARA O PROBLEMA
DINÂMICO DE ROTEAMENTO DE VEÍCULOS**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

ORIENTADOR: GERALDO ROBSON MATEUS

BELO HORIZONTE

2011

© Humberto César Brandão de Oliveira.

Todos os direitos reservados.

Oliveira, Humberto César Brandão de.

O48a Algoritmo online para o problema dinâmico de roteamento de veículos / Humberto César Brandão de Oliveira. — Belo Horizonte, 2011.
 xxiv, 123 f. : il. ; 29cm

 Tese (doutorado) — Universidade Federal de Minas Gerais – Departamento de Ciência da Computação.

 Orientador: Geraldo Robson Mateus.

 1. Computação - Teses. 2. Otimização matemática – Teses. 3. Algoritmos de computador - Teses. I.
Orientador. II.
Título.

519.6*61(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Despacho online para o problema dinâmico de roteamento de veículos

HUMBERTO CÉSAR BRANDÃO DE OLIVEIRA

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. GERALDO ROBSON MATEUS - ORIENTADOR
Departamento de Ciência da Computação - UFMG

PROF. ALEXANDRE SALLES DA CUNHA
Departamento de Ciência da Computação - UFMG

PROF. EDUARDO UCHOA BARBOZA
Departamento de Engenharia de Produção - UFF

DR. GUILHERME BASTOS ALVARENGA
DEVEX

PROFA. LÚCIA MARIA DE ASSUMPCÃO DRUMMOND
Departamento de Ciência da Computação - UFF

PROF. SÉRGIO RICARDO DE SOUZA
CEFET-MG

Belo Horizonte, 20 de dezembro de 2011.

*Dedico este trabalho aos meus pais
Maria Cristina Gontijo Brandão de Oliveira
José Humberto de Oliveira*

Agradecimentos

À Maria Regina Martinez, pelo apoio incondicional.

Ao orientador Geraldo Robson Mateus, por me guiar com sabedoria ao longo destes últimos anos.

Aos colegas da UFMG, que colaboraram para a realização deste trabalho: Cristiano Arbex, Paulo Gomide e Leonardo Conegundes.

Ao discente e amigo Edgar Franco, que durante sua pesquisa alcançou resultados que colaboraram com este trabalho.

Aos colegas de UNIFAL, Rafael Lemos Bastos e Prof. Eric Batista Ferreira, pela consultoria em estatística.

Aos amigos Leonardo Ciscon e Gustavo Rocha, pela colaboração durante as disciplinas do doutorado.

Ao CNPq, pelo apoio financeiro.

E principalmente a Deus, por tudo que me tem fornecido ao longo da minha trajetória.

Resumo

A alocação de veículos para uma determinada demanda de consumidores está sujeita a uma explosão combinatória de possibilidades, devido ao aumento exponencial de alternativas e de acordo com o crescimento do tamanho do problema. Quando são consideradas alterações no ambiente, como o surgimento de novos consumidores, o Problema de Roteamento de Veículos (PRV) se torna dinâmico e ainda mais complexo e imprevisível. É comum encontrar na literatura do Problema Dinâmico de Roteamento de Veículos (PDRV) trabalhos que utilizam uma abordagem periódica para o roteamento dos veículos. Esta forma de tratamento divide o dia em intervalos bem definidos e trata vários problemas estáticos ao longo do tempo. Como principal contribuição, este trabalho propõe a utilização de algoritmos de roteamento como tomada de decisão imediata ou de curto prazo para tratar os PDRVs. Nesta abordagem, os consumidores são informados rapidamente (*online*) quando serão atendidos. Este trabalho mostra que a abordagem de roteamento *online* pode ter maior impacto sobre os custos se comparada a abordagem periódica.

Além da contribuição sobre o roteamento *online*, este trabalho apresenta duas heurísticas híbridas para resolver o Problema estático de Roteamento de Veículos com Janela de Tempo (PRVJT). Ambas exploram a formulação do Problema de Particionamento de Conjuntos para solucionar o PRVJT. O primeiro algoritmo híbrido é especialmente proposto para atender contextos dinâmicos do PRVJT, em que novas soluções devem ser encontradas rapidamente. O segundo algoritmo híbrido é mais robusto, alcança melhores resultados e é indicado para contextos estáticos. Resultados computacionais mostram que as heurísticas propostas são competitivas em comparação com outros algoritmos da literatura que tratam a bem conhecida base de testes de Solomon. Este trabalho superou o melhor resultado conhecido em oito instâncias, considerando a minimização da distância total do PRVJT. Além disso, o segundo algoritmo melhorou ou igualou 82,1% dos melhores resultados conhecidos para a base de testes de Solomon.

Palavras-Chave: Problema Dinâmico de Roteamento de Veículos, Geração de Colunas, Roteamento *Online*, Roteamento Periódico, *Simulated Annealing*.

Abstract

The allocation of vehicles for a specific customers' demand is subject to a combinatorial explosion of possibilities by the exponential increase of alternatives according to growth of the problem size. When environmental changes are considered, such as the advent of new customers, the Vehicle Routing Problem becomes dynamic and even more complex and unpredictable. It is common to find, in the literature of the Dynamic VRP (DVRP), works that use a periodic approach to the vehicles dispatch. This treatment way divides the day at well-defined intervals and handles many static problems over time. As its main contribution, this work proposes the use of online dispatches to treat DVRPs. In this approach, the customers are quickly informed when they will be attended. This work shows that the online dispatch may have a greater impact on costs compared to the periodic dispatch.

In addition of online dispatch contribution, this work presents two hybrid heuristics to solve the static Vehicle Routing Problem with Time Windows (VRPTW). Both heuristics explore the set partitioning formulation for the VRPTW. The first hybrid algorithm is specially proposed to attend dynamic contexts of the VRPTW, where new solutions must be quickly found. The second hybrid algorithm is more robust, it achieves best results and it is indicated for static contexts. Computational results show that the proposed heuristics are competitive in comparison with other algorithms of literature considering the well-known Solomon's database. This work has found eight new best solutions considering the total distance minimization for VRPTW. Moreover, the second algorithm has obtained results better or equal to 82.1% of the best-known results from Solomon's instances.

Keywords: Dynamic Vehicle Routing Problem, Column Generation, Online Dispatch, Periodic Dispatch, Simulated Annealing.

Lista de figuras

FIGURA 1.1 – RESOLUÇÃO DO PDRV ATRAVÉS DE UM ROTEAMENTO PERIÓDICO.....	4
FIGURA 1.2 – DEMANDA INICIAL NO PDRV ($T=A$).....	4
FIGURA 1.3 – PROGRAMAÇÃO INICIAL ANTES DA SAÍDA DOS VEÍCULOS ($T=B$).....	5
FIGURA 1.4 – CONHECIMENTO DE CONSUMIDORES DINÂMICOS PELO ALGORITMO DE ROTEAMENTO PERIÓDICO ($T=C$).....	5
FIGURA 1.5 – REPROGRAMAÇÃO ATRAVÉS DO ALGORITMO DE ROTEAMENTO PERIÓDICO ($T=D$).....	6
FIGURA 1.6 – RESOLUÇÃO DO PDRV ATRAVÉS DE UM ALGORITMO DE ROTEAMENTO <i>ONLINE</i>	7
FIGURA 1.7 – CONHECIMENTO DO CONSUMIDOR DINÂMICO PELO ALGORITMO DE ROTEAMENTO <i>ONLINE</i> ($T=M$).....	8
FIGURA 1.8 – REPROGRAMAÇÃO ATRAVÉS DO ALGORITMO DE ROTEAMENTO <i>ONLINE</i> ($T=N$).....	8
FIGURA 2.1 – POSSÍVEL SOLUÇÃO PARA UM PROBLEMA DE ROTEAMENTO DE VEÍCULOS.....	20
FIGURA 3.1 – ALOCAÇÃO VEICULAR SEM ESPERA VEICULAR PROGRAMADA.....	39
FIGURA 3.2 – ALOCAÇÃO VEICULAR COM ESPERA VEICULAR.....	39
FIGURA 4.1 – EXEMPLO DA DISPOSIÇÃO DOS CONSUMIDORES DAS CLASSES R1 E R2.....	50
FIGURA 4.2 – EXEMPLO DA DISPOSIÇÃO DOS CONSUMIDORES DAS CLASSES C1 E C2.....	51
FIGURA 4.3 – EXEMPLO DA DISPOSIÇÃO DOS CONSUMIDORES DA CLASSE RC1 E RC2.....	51
FIGURA 4.4 – LINHA DO TEMPO PARA ALGORITMOS PERIÓDICOS DE 1 EM 1 HORA.....	53
FIGURA 4.5 – LINHA DO TEMPO PARA ALGORITMOS PERIÓDICOS DE 30 EM 30 MINUTOS.....	54
FIGURA 4.6 – TESTE DA ANAVA SOBRE A DISTRIBUIÇÃO F	57
FIGURA 5.1 – FLUXOGRAMA SIMPLIFICADO DO SIMULADOR EVENTO DISCRETO IMPLEMENTADO PARA O PDRV.....	61
FIGURA 6.1 – FLUXOGRAMA DO ALGORITMO HÍBRIDO 1 PARA O PRVJT.....	68
FIGURA 6.2 – EXEMPLOS DE DECAIMENTO DA TEMPERATURA DO SA.....	71
FIGURA 6.3 – PROBABILIDADE DO SA TROCAR A SOLUÇÃO ATUAL.....	72
FIGURA 6.4 - SOLUÇÃO ATUAL INCOMPLETA ANTES DA INSERÇÃO DE UM NOVO CONSUMIDOR [OLIVEIRA E VASCONCELOS, 2010].....	76
FIGURA 6.5 - POSSÍVEIS SOLUÇÕES ENCONTRADAS PELO ALGORITMO PFIH DE ACORDO COM A FIGURA 6.4 [OLIVEIRA E VASCONCELOS, 2010].....	76
FIGURA 6.6 - SOLUÇÃO SELECIONADA COMO MELHOR OPÇÃO NA INSERÇÃO DO CONSUMIDOR C5 [OLIVEIRA E VASCONCELOS, 2010].....	77
FIGURA 6.7 - EXEMPLO DE APLICAÇÃO DO OPERADOR DE VIZINHANÇA ‘TROCA’.....	78
FIGURA 6.8 - EXEMPLO DE APLICAÇÃO DO OPERADOR DE VIZINHANÇA ‘INSERÇÃO’.....	79
FIGURA 6.9 - EXEMPLO DE APLICAÇÃO DO OPERADOR DE VIZINHANÇA ‘EMBARALHAMENTO’.....	79
FIGURA 6.10 - EXEMPLO DE APLICAÇÃO DO OPERADOR DE VIZINHANÇA ‘INVERSÃO’.....	79
FIGURA 6.11 – FLUXOGRAMA DO ALGORITMO HÍBRIDO 2 PARA O PRVJT.....	82
FIGURA 6.12 – EXEMPLO DO ESTADO SIMPLIFICADO DO AMBIENTE QUANDO O ALGORITMO DINÂMICO É CHAMADO.....	84
FIGURA 6.13 – EXEMPLO DO ESTADO SIMPLIFICADO DO AMBIENTE QUANDO O ALGORITMO DINÂMICO TERMINA SUA EXECUÇÃO.....	85
FIGURA 6.14 – EXEMPLO DE SOLUÇÃO PARA O PRVJT COM MÚLTIPLOS DEPÓSITOS ADAPTADO.....	86
FIGURA 7.1 - DISTÂNCIA TOTAL ACUMULADA PARA TODAS AS INSTÂNCIAS DE SOLOMON [1987].....	93
FIGURA 7.2 – VEÍCULOS ACUMULADOS PARA TODAS AS INSTÂNCIAS DE SOLOMON [1987].....	94
FIGURA 7.3 - TEMPO MÉDIO DE EXECUÇÃO NAS INSTÂNCIAS DE SOLOMON [1987].....	95
FIGURA 7.4 – RESULTADO SUMARIZADO CONSIDERANDO O TIPO DE DISTRIBUIÇÃO DOS CONSUMIDORES.....	100
FIGURA 7.5 – RESULTADO SUMARIZADO CONSIDERANDO A QUANTIDADE DE PERÍODOS.....	101
FIGURA 7.6 – RESULTADO SUMARIZADO CONSIDERANDO O GRAU DE DINAMISMO.....	101
FIGURA 7.7 – RESULTADO SUMARIZADO CONSIDERANDO A INSTÂNCIA BASE DE SOLOMON [1987].....	102
FIGURA 7.8 – RESULTADO SUMARIZADO GERAL.....	103

Lista de tabelas

TABELA 2.1 - CLASSIFICAÇÃO GENÉRICA PARA OS VARIADOS TIPOS DE PROBLEMAS DE ROTEAMENTO	16
TABELA 6.1 - PARÂMETROS DE AH1 DEPENDENTES DO ALGORITMO DE ROTEAMENTO	69
TABELA 6.2 - PARÂMETROS DO SANM (CAIXA 1)	73
TABELA 6.3 - PARÂMETROS DO SANM (CAIXA 3)	74
TABELA 6.4 - PARÂMETROS DO CPLEX DO AH1 (CAIXA 2 DA FIGURA 6.1)	81
TABELA 7.1 - RESULTADOS DOS ALGORITMOS HÍBRIDOS PARA O PRVJT SOBRE O GRUPO R1	90
TABELA 7.2 - RESULTADOS DOS ALGORITMOS HÍBRIDOS PARA O PRVJT SOBRE O GRUPO R2	91
TABELA 7.3 - RESULTADOS DOS ALGORITMOS HÍBRIDOS PARA O PRVJT SOBRE O GRUPO C1	91
TABELA 7.4 - RESULTADOS DOS ALGORITMOS HÍBRIDOS PARA O PRVJT SOBRE O GRUPO C2	91
TABELA 7.5 - RESULTADOS DOS ALGORITMOS HÍBRIDOS PARA O PRVJT SOBRE O GRUPO RC1	91
TABELA 7.6 - RESULTADOS DOS ALGORITMOS HÍBRIDOS PARA O PRVJT SOBRE O GRUPO RC2	92
TABELA 7.7 - PERCENTUAL DE RESULTADOS IGUALADOS OU SUPERADOS POR CADA ALGORITMO	92
TABELA 7.8 - COMPARAÇÃO ENTRE DIFERENTES TRABALHOS QUE MINIMIZAM A DISTÂNCIA VIAJADA NO PRVJT	93
TABELA 7.9 - TEMPO MÉDIO DE PROCESSAMENTO DOS ALGORITMOS QUE RESOLVEM O PRVJT	94
TABELA 7.10 - PROCESSADORES UTILIZADOS PARA RESOLVER AS INSTÂNCIAS DE SOLOMON [1987]	95
TABELA 7.11 - RESULTADOS PARA O GRUPO DE INSTÂNCIAS BASEADAS EM C101	96
TABELA 7.12 - RESULTADOS PARA O GRUPO DE INSTÂNCIAS BASEADAS EM C203	97
TABELA 7.13 - RESULTADOS PARA O GRUPO DE INSTÂNCIAS BASEADAS EM R106	97
TABELA 7.14 - RESULTADOS PARA O GRUPO DE INSTÂNCIAS BASEADAS EM R202	97
TABELA 7.15 - RESULTADOS PARA O GRUPO DE INSTÂNCIAS BASEADAS EM RC104	98
TABELA 7.16 - RESULTADOS PARA O GRUPO DE INSTÂNCIAS BASEADAS EM RC208	98
TABELA 7.17 - AVALIAÇÃO MÉDIA DOS ALGORITMOS DE ROTEAMENTO POR GRUPOS	99
TABELA 7.18 - COMPARAÇÃO COM OS LIMITES INFERIORES	105

Lista de algoritmos

ALGORITMO 1 – DESCRIÇÃO GERAL DO <i>SA</i>	70
ALGORITMO 2 – DESCRIÇÃO GERAL DO <i>SANM</i> (RESPONSÁVEL PELA DIVERSIFICAÇÃO)	74
ALGORITMO 3 – DESCRIÇÃO GERAL DO <i>SANM</i> (RESPONSÁVEL PELA INTENSIFICAÇÃO)	75
ALGORITMO 4 – DESCRIÇÃO GERAL DO <i>GRASP</i>	82
ALGORITMO 5 – GERAÇÃO DE NÚMEROS AUXILIARES PARA CÁLCULO DO CÓDIGO HASH	121
ALGORITMO 6 – CÁLCULO DO CÓDIGO <i>HASH</i> DE UMA ROTA	122

Lista de símbolos

\bar{x}	Média amostral
μ	Média populacional
s	Desvio padrão amostral
Δ	Varição entre valores de diferentes avaliações da função objetivo

Sumário

AGRADECIMENTOS	IX
RESUMO	XI
ABSTRACT	XIII
LISTA DE FIGURAS	XV
LISTA DE TABELAS	XVII
LISTA DE ALGORITMOS	XIX
LISTA DE SÍMBOLOS	XXI
CAPÍTULO 1 - INTRODUÇÃO	1
1.1 MOTIVAÇÃO	1
1.2 PROBLEMATIZAÇÃO	3
1.2.1 Roteamento periódico	3
1.2.2 Roteamento <i>online</i>	7
1.2.3 Vantagens e desvantagens dos roteamentos discutidos	9
1.3 OBJETIVO PRINCIPAL E CONTRIBUIÇÕES DA TESE	10
1.3.1 Objetivo principal	10
1.3.2 Principais contribuições	10
1.4 ORGANIZAÇÃO DA TESE	11
CAPÍTULO 2 - O PROBLEMA DE ROTEAMENTO DE VEÍCULOS E SUAS VARIAÇÕES	15
2.1 CONSIDERAÇÕES INICIAIS	15
2.2 VARIAÇÕES CLÁSSICAS DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS ESTÁTICO	15
2.3 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS CAPACITADO (PRVC)	19
2.4 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE TEMPO (PRVJT)	20
2.4.1 Otimização das viagens no PRVJT	21
2.4.2 Um modelo matemático para o PRVJT	21
2.4.3 A formulação do Problema de Particionamento de Conjuntos adaptada ao PRVJT	24
2.4.4 Complexidade do PRVJT	26
2.5 O PROBLEMA DINÂMICO DE ROTEAMENTO DE VEÍCULOS	28
2.5.1 Complexidade e tratabilidade do PDRV	28
2.6 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS ESTOCASTICO	29
2.7 SUMÁRIO DO CAPÍTULO	31
CAPÍTULO 3 - REVISÃO BIBLIOGRÁFICA	33
3.1 CONSIDERAÇÕES INICIAIS	33
3.2 REQUISIÇÕES DINÂMICAS ANTES DO INÍCIO DO DIA DE SERVIÇO	35
3.3 REQUISIÇÕES DINÂMICAS DURANTE O DIA DE SERVIÇO	36
3.3.1 Subdividindo o tempo dia de serviço em períodos	36
3.3.2 Estratégias de espera para o PDRV	38
3.3.3 Estratégias de desvio para o PDRV	42
3.4 INFORMAÇÕES HISTÓRICAS SOBRE REQUISIÇÕES DE CLIENTES	43
3.5 SUMÁRIO DO CAPÍTULO	45
CAPÍTULO 4 - METODOLOGIA	47
4.1 DEFINIÇÕES INICIAIS	47
4.2 HIPÓTESE DA TESE	49
4.3 BASE DE TESTES	49

4.3.1 Extensão da base de Solomon [1987]	52
4.4 EXPERIMENTO	55
4.4.1 Teste de hipótese	55
4.4.2 Delineamento do experimento	56
4.4.3 ANAVA sobre o DBC	57
4.4.4 Considerações técnicas	57
4.5 CONSIDERAÇÕES FINAIS	58
CAPÍTULO 5 - SIMULAÇÃO EVENTO-DISCRETA PARA O PDRV	59
5.1 CONSIDERAÇÕES INICIAIS	59
5.2 CONCEITOS GERAIS SOBRE SIMULAÇÃO EVENTO-DISCRETA	59
5.3 PROCESSO DE SIMULAÇÃO PARA O PDRV	60
5.3.1 Eventos no simulador do PDRV	62
5.4 CONSIDERAÇÕES FINAIS	65
CAPÍTULO 6 - ALGORITMOS PARA RESOLUÇÃO DO PDRV E DO PRVJT	67
6.1 CONSIDERAÇÕES INICIAIS	67
6.2 ALGORITMO HÍBRIDO 1 PARA RESOLVER O PRVJT ESTÁTICO (APLICÁVEL AO PDRV)	67
6.2.1 Parâmetros gerais do Algoritmo Híbrido	69
6.2.2 <i>Simulated Annealing</i> não-monotônico (SANM)	69
6.2.3 Problema de Particionamento de Conjuntos (PPC) no Algoritmo Híbrido	80
6.3 ALGORITMO HÍBRIDO 2 PARA RESOLVER O PRVJT ESTÁTICO (NÃO APLICÁVEL AO PDRV)	81
6.4 RESOLUÇÃO DO PDRV	83
6.4.1 PRVJT com Múltiplos Depósitos (PRVJTMD) adaptado	84
6.4.2 Resolução do PPC durante a fase dinâmica do PDRV	87
6.5 CONSIDERAÇÕES FINAIS	87
CAPÍTULO 7 - RESULTADOS	89
7.1 CONSIDERAÇÕES INICIAIS	89
7.2 RESOLUÇÃO DA VERSÃO ESTÁTICA DO PRVJT	90
7.2.1 Base de testes para o PRVJT estático	90
7.2.2 Resultados da resolução do PRVJT	90
7.2.3 Comparação com outros trabalhos	92
7.2.4 Tempo de processamento	94
7.3 RESULTADOS DA VERSÃO DINÂMICA DO PRV	96
7.3.1 Base de testes para o PDRV	96
7.3.2 Resultados na resolução do PDRV	96
7.3.3 Teste estatístico para avaliação da hipótese	98
7.3.4 Comparação entre os algoritmos de roteamento no PDRV	99
7.3.5 Limite inferior no PDRV para as instâncias estendidas	104
7.4 CONSIDERAÇÕES FINAIS	106
CAPÍTULO 8 - CONCLUSÕES	109
REFERÊNCIAS BIBLIOGRÁFICAS	113
APÊNDICE A - DETALHES SOBRE A IMPLEMENTAÇÃO E MANUTENÇÃO DO CONJUNTO R' PARA O PCC	119

Capítulo 1

Introdução

1.1 Motivação

Gastos relacionados com o transporte de pessoas e mercadorias recebem a cada dia maior importância nos processos decisórios das empresas, em que a minimização dos custos é um importante foco para aquelas que visam sobreviver e crescer no mercado atual. Pesquisas vêm contribuindo de maneira significativa nas últimas décadas para a solução de problemas relacionados à logística, em que o Problema de Roteamento de Veículos (PRV), proposto por Dantzig e Ramser [1959], possui considerável importância.

De forma geral, o PRV consiste em alocar uma frota veicular para o atendimento de uma determinada demanda de consumidores. Em outras palavras, o objetivo da resolução do PRV é determinar rotas, em que cada uma delas representa uma programação para atendimento. Cada rota apresenta uma ordem bem definida para a visitação de seus consumidores. Todos os consumidores devem ser visitados exatamente uma vez. Além disso, a capacidade de carga de cada veículo não pode ser excedida pelo somatório das demandas de seus consumidores.

Na última década, as empresas de transporte se multiplicaram com o considerável aumento das vendas no comércio eletrônico. Estima-se que, em 2012, as vendas deste setor possam alcançar um montante de 335 bilhões de dólares em todo o mundo. No primeiro semestre de 2010, o comércio eletrônico brasileiro registrou faturamento de 6,7 bilhões de reais, representando 40% de crescimento em comparação com os seis primeiros meses do ano de 2009 [Braun, 2010]. O custo anual de distribuição de mercadorias nos Estados Unidos foi estimado em 400 bilhões de dólares. Deste valor, estima-se que 45 bilhões de dólares representam desperdícios, podendo chegar a até 30% do valor de um produto [King e Mast, 1997]. Em um contexto regional, pesquisas indicam que serviços com transporte representam 2,53% do produto interno bruto (PIB) do Brasil [Fraga, 2007]. Por motivos como estes, o PRV

é um dos problemas mais estudados na área de Pesquisa Operacional [Steiner *et al.*, 2000].

Como o problema de distribuição de mercadorias no mundo real se apresenta de variadas formas, há, naturalmente, uma subdivisão do mesmo em categorias, de acordo com as características e restrições presentes nas situações reais de operação. Uma destas subdivisões mais estudadas considera a capacidade de carga do veículo e o intervalo de tempo em que os consumidores devem ser atendidos (janelas de tempo). Esta caracterização, em particular, passou a ser formalmente conhecida como Problema de Roteamento de Veículos com Janela de Tempo (PRVJT). Esta versão do PRV explicita que os clientes precisam ser atendidos em horários pré-estabelecidos, de acordo com suas necessidades.

Além do avanço nos estudos sobre problemas baseados no PRV, houve também uma rápida evolução tecnológica, como, por exemplo, a popularização dos equipamentos de GPS (do acrônimo do inglês *Global Positioning System*), telefones celulares, sistemas de informação geográficos. Tais avanços oferecem suporte nas decisões de um sistema de roteamento de veículos ao longo do dia de serviço [Ichoua *et al.*, 2000]. Estes recursos possibilitam, por exemplo, a mudança na programação previamente definida para uma frota de veículos, de acordo com as alterações que ocorrem no ambiente e na demanda de serviços. A possibilidade da alteração da alocação previamente definida para os veículos caracteriza o Problema Dinâmico de Roteamento de Veículos (PDRV). Como a popularização do suporte tecnológico para atender eficientemente as necessidades do PDRV é recente, seu estudo na área de Pesquisa Operacional está em plena evidência.

Além da recente disponibilidade de tecnologia, existem os fatores de poluição e ruído que têm se tornado críticos, principalmente nas grandes metrópoles. Uma parcela destes custos e inconvenientes poderia ser reduzida com o tratamento de diversos problemas de roteamento, em que as versões dinâmicas do PRV possuem uma particular relevância.

Outra motivação para o estudo das versões dinâmicas está relacionada ao grande desafio que as soluções destes problemas representam. As versões estáticas do PRV pertencem ao conjunto dos problemas conhecidos como NP-Difíceis [Larsen, 2001]. O Problema Dinâmico de Roteamento de Veículos é agravado quando o algoritmo de roteamento deve responder em um ambiente que se altera com os veículos já em movimento.

1.2 Problematização

Antes da definição dos objetivos da tese, este trabalho apresenta a seguinte pergunta sobre o PDRV:

Durante a resolução do PDRV, seria mais vantajoso reagir ao ambiente rapidamente quando novos pedidos são requisitados, ao invés de esperar instantes pré-programados com tempos mais longos para processamento?

Para contextualizar, antes de apresentar o objetivo principal da tese, as seguintes subseções descrevem duas formas distintas para a realização do roteamento no PDRV.

1.2.1 Roteamento periódico

No caso do PDRV, é comum encontrar na literatura algoritmos de roteamento que tratam o problema de forma periódica, como por exemplo, os trabalhos de Bent e Van Hentenryck [2004], Chen e Xu [2006] e Hvattum *et al.* [2007]. Estes algoritmos subdividem o tempo e tratam diversos problemas de roteamento estáticos ao longo do dia, tendo hora marcada para fornecer possíveis alterações da programação para a frota veicular. A Figura 1.1 apresenta um exemplo de um dia dentro da alocação periódica de veículos.

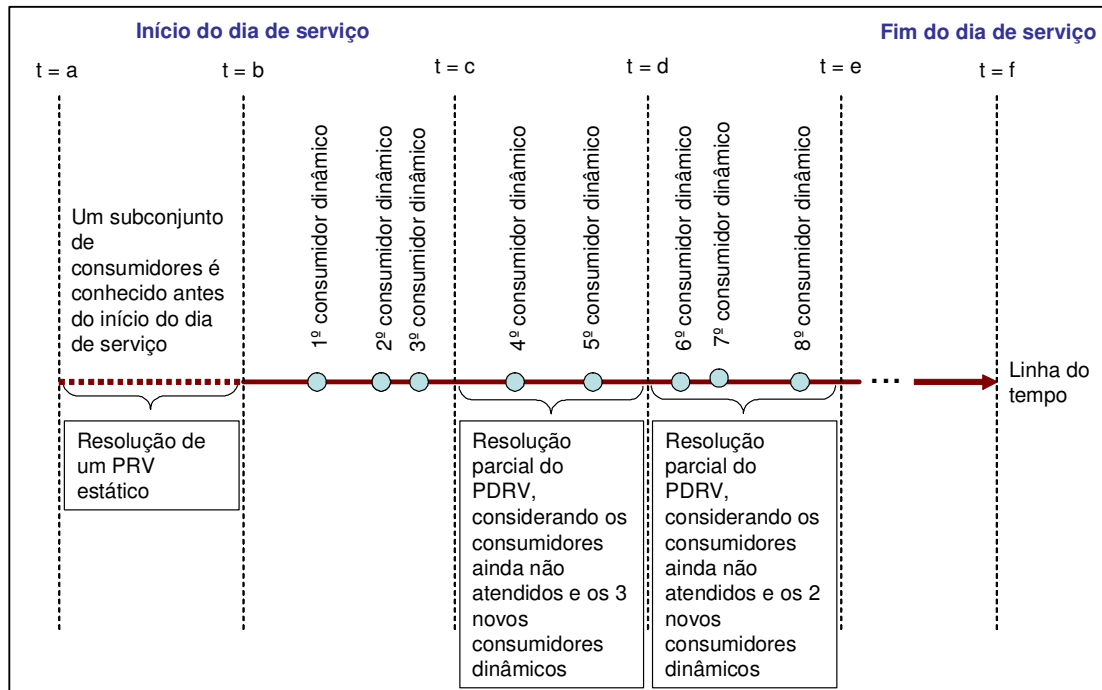


Figura 1.1 - Resolução do PDRV através de um roteamento periódico

No início da linha do tempo apresentada na Figura 1.1 ($t=a$), um conjunto de consumidores (neste trabalho denominados consumidores estáticos), que requisitaram serviço antes da partida dos veículos, é apresentado ao algoritmo de roteamento, que resolve uma versão estática do PRV. No mundo real, esta demanda pode ser, por exemplo, os pedidos conhecidos pelo fornecedor no dia anterior, e que não puderam ser atendidos. Espera-se conhecer a alocação inicial para a frota veicular antes do início da jornada de trabalho ($t=b$). Na prática, geralmente, as empresas possuem toda a madrugada para resolver este problema estático de roteamento de veículos. A Figura 1.2 apresenta um exemplo de demanda inicial com cinco consumidores conhecidos pelo sistema de roteamento no instante $t=a$.

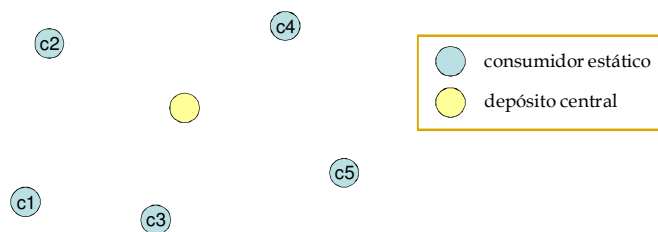


Figura 1.2 - Demanda inicial no PDRV ($t=a$)

Após conhecida a demanda de consumidores antes do início do horário de trabalho, o sistema de roteamento deve encontrar uma solução para o PRV estático. Sendo assim, até o instante $t=b$, a frota veicular deve ser informada sobre sua programação inicial, para que os veículos entrem em movimento. A Figura 1.3 apresenta um exemplo de programação inicial, utilizando dois veículos para atender os cinco consumidores previamente apresentados na Figura 1.2.

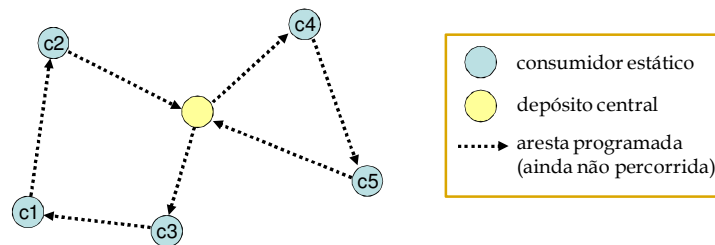


Figura 1.3 – Programação inicial antes da saída dos veículos ($t=b$)

A partir de $t=b$, novos consumidores podem surgir ao longo do tempo (neste trabalho denominados consumidores dinâmicos). Também a partir de $t=b$, os veículos podem iniciar seus trajetos, deixando o depósito. No tratamento periódico do PDRV, os consumidores dinâmicos, por ora, são armazenados para que, no futuro, sirvam de entrada para a resolução parcial do PDRV. No exemplo descrito na Figura 1.1, a partir de $t=c$, os três primeiros consumidores dinâmicos são armazenados pelo algoritmo de roteamento periódico, que irá resolver parcialmente o PDRV (entre $t=c$ e $t=d$). A Figura 1.4 apresenta um exemplo do instante de tempo $t=c$ indicado na Figura 1.1, em que três consumidores dinâmicos ($d1$, $d2$ e $d3$) são conhecidos pelo algoritmo de roteamento, e os veículos já percorreram parte da programação indicada na Figura 1.3.

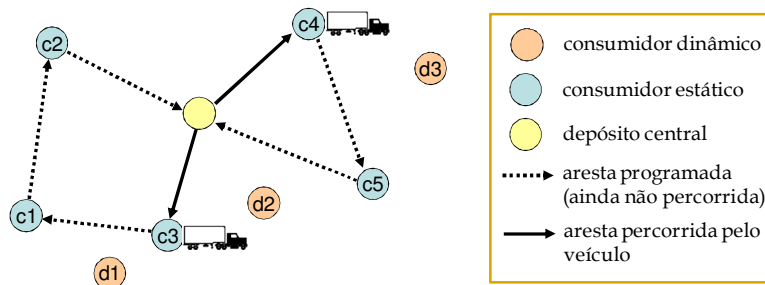


Figura 1.4 – Conhecimento de consumidores dinâmicos pelo algoritmo de roteamento periódico ($t=c$)

No roteamento periódico, a aplicação do algoritmo a partir de $t=c$ não deve considerar os consumidores já atendidos ou em atendimento pela frota veicular ($c3$ e $c4$), considerando somente aqueles ainda não atendidos e os novos consumidores dinâmicos. Vale ressaltar que o roteamento também não deve considerar os consumidores que serão atendidos até o instante $t=d$. A alocação veicular para estes consumidores também não poderá sofrer alterações, pois eles de fato serão atendidos antes que o algoritmo periódico indique a próxima programação para a frota de veículos. Suponha que, além de $c3$ e $c4$, os consumidores $c1$ e $c5$ também serão atendidos até o instante $t=d$. Portanto, algumas arestas programadas inicialmente em $t=b$ não poderão ser alteradas pela nova programação do roteamento periódico: ((depósito, $c3$); ($c3$, $c1$); (depósito, $c4$); ($c4$, $c5$)). No instante de tempo pré-determinado $t=d$, o algoritmo de roteamento informa para cada veículo sua nova programação, encontrada pelo processo de otimização que resolveu o PDRV entre os instantes $t=c$ e $t=d$ (Figura 1.5).

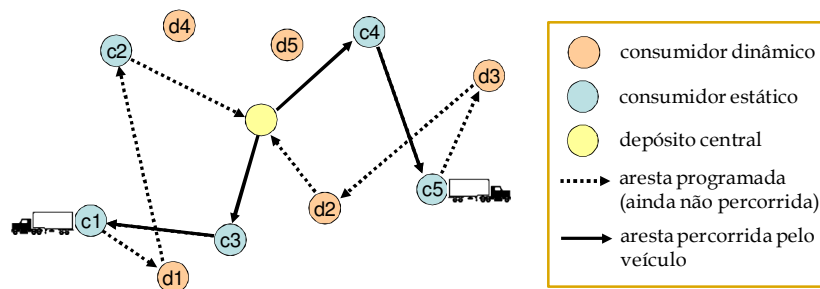


Figura 1.5 - Reprogramação através do algoritmo de roteamento periódico ($t=d$)

Vale também ressaltar que, no instante $t=d$, os consumidores dinâmicos $d4$ e $d5$ já solicitaram atendimento, mas ainda não foram atendidos pelo roteamento periódico, que processava uma reprogramação para atender inicialmente aos consumidores $d1$, $d2$ e $d3$.

Seguindo esta lógica recursiva, o algoritmo de roteamento periódico funciona até o fim do dia, armazenando os últimos pedidos para o próximo dia de trabalho. Estes servirão de entrada para o problema estático resolvido entre $t=a$ e $t=b$ do próximo dia útil.

O roteamento periódico pode também, ao invés de fazer uso de todo o tempo disponível para processamento, por exemplo, de $t=c$ até $t=d$, utilizar um tempo menor em cada algoritmo iniciado periodicamente. Assim, é possível se beneficiar com o

acúmulo de pedidos, e também ter uma maior folga temporal para encaixar os consumidores nas rotas planejadas. Esta alternativa não foi encontrada na literatura e também é avaliada nesta tese.

1.2.2 Roteamento *online*

Uma alternativa ao tratamento periódico no PDRV seria uma abordagem com tomada de decisão imediata ou de curto prazo. Neste contexto, o sistema de roteamento possui um tempo máximo e curto para responder às novas requisições dinâmicas que surgem com os veículos já em movimento. A Figura 1.6 apresenta um exemplo geral de um algoritmo *online* respondendo às demandas dinâmicas ao longo do tempo.

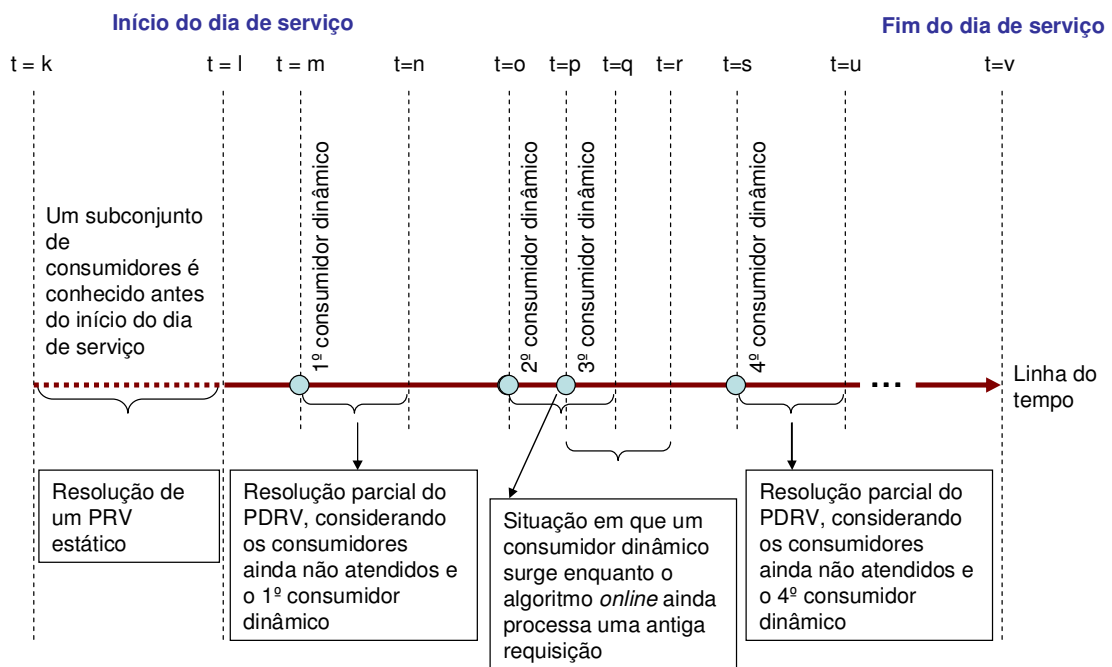


Figura 1.6 - Resolução do PDRV através de um algoritmo de roteamento *online*

Nos instantes $t=k$ e $t=l$, o processo é idêntico ao apresentado na versão periódica nos instantes $t=a$ e $t=b$ (Figura 1.2 e Figura 1.3). As mudanças são visíveis quando a frota já está em andamento (a partir do instante $t=m$ da Figura 1.6). Diferentemente de um algoritmo periódico, quando um novo consumidor dinâmico requisita atendimento ao longo do dia, o algoritmo de roteamento *online* começa a resolver imediatamente o problema, considerando a nova demanda dinâmica, ao invés de armazenar o pedido para processar as novas requisições em um instante futuro pré-determinado. A Figura 1.7 representa o momento $t=m$, quando o novo consumidor $d1$ requisita o atendimento ao algoritmo de roteamento.

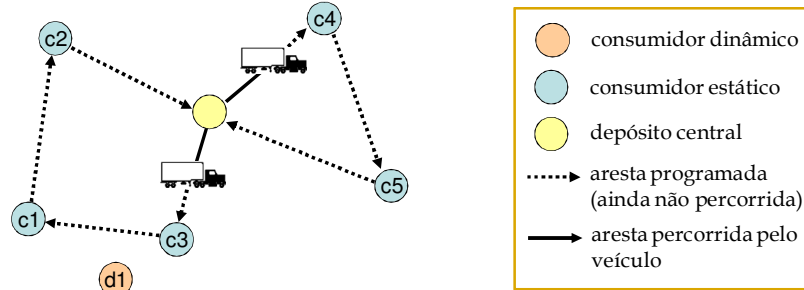


Figura 1.7 - Conhecimento do consumidor dinâmico pelo algoritmo de roteamento *online* ($t=m$)

Assim como na versão periódica, o algoritmo *online* não deve considerar os consumidores já visitados, como também aqueles que serão visitados até existir resposta do algoritmo de roteamento à frota veicular, indicando a reprogramação. O algoritmo de roteamento *online* deve ter um tempo máximo para informar à frota as novas ordens de serviço. Neste exemplo, atender o novo consumidor *d1* antes do consumidor estático *c1* depende de alguns fatores temporais: (i) qual é o tempo máximo de resposta do algoritmo *online*? (ii) quanto tempo falta para o veículo chegar até o consumidor *c3*? (iii) qual é o tempo necessário para o atendimento de *c3*? A Figura 1.8 apresenta o caso do algoritmo de roteamento conseguir responder antes do consumidor estático *c3* ser atendido.

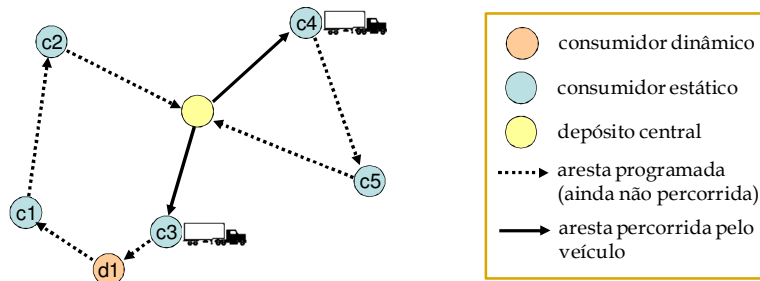


Figura 1.8 - Reprogramação através do algoritmo de roteamento *online* ($t=n$)

Uma situação específica dentro de um algoritmo *online* merece atenção especial. É a situação mostrada no instante $t=p$ indicado na Figura 1.6. Um terceiro consumidor dinâmico surge enquanto o algoritmo de otimização *online* está em execução para atender a demanda conhecida no instante $t=0$, com o aparecimento do segundo consumidor dinâmico. A situação merece atenção, pois o processo de otimização iniciado em $t=0$ não pode ser interrompido por completo para que um

novo processo seja iniciado em $t=p$, pois existe um tempo máximo para o atendimento do segundo consumidor dinâmico, que é definido até $t=q$. Com isto, decisões de projeto são importantes na definição de um algoritmo *online*. Pode-se, por exemplo, interromper o processo de otimização em $t=p$, incluir o terceiro consumidor no processo de otimização e agendar uma parada na otimização até $t=q$, para que o segundo consumidor dinâmico seja atendido. Pode-se também interromper o processo de otimização em execução e considerar o problema definido em $t=0$ já resolvido. Assim, é possível informar a nova alocação para a frota e inicializar um novo processo em $t=p$, comunicando-se novamente com a frota apenas em $t=r$. Isso evitaria a necessidade de resposta em $t=q$. Pode-se também iniciar um processo de otimização em paralelo ao em execução. Neste caso, o problema seria desconhecer como o segundo consumidor dinâmico será alocado, pois a programação do terceiro pode depender diretamente da programação do segundo, de acordo com as soluções encontradas nos dois processos de otimização. Neste trabalho é garantido um tempo máximo para a alocação de consumidores dinâmicos. Todo consumidor é alocado nas rotas programadas para atendimento em no máximo 3 minutos após o instante da requisição. Para isso, o algoritmo dinâmico é configurado para executar em no máximo 1,5 minutos. Isso garante que toda requisição dinâmica é alocada nas rotas planejadas do ambiente em até 3 minutos após sua requisição.

1.2.3 Vantagens e desvantagens dos roteamentos discutidos

Tratar o PDRV com uma abordagem de roteamento periódica fornece a vantagem de reduzir a quantidade de comunicações necessárias entre o algoritmo de roteamento e os veículos, pois esta comunicação sempre acontecerá em horários pré-determinados do dia. Outra vantagem da alocação periódica é uma maior disponibilidade de tempo para a execução do algoritmo de otimização. Se o dia for dividido em três partes, como, por exemplo, manhã, tarde e noite, o algoritmo terá todo o período da tarde para otimizar a demanda conhecida no período da manhã. Considerando que a resolução parcial do PDRV também representa a resolução de um problema NP-Difícil, tal disponibilidade de tempo pode aumentar a qualidade da solução final. Um número maior de requisições a serem tratadas, devido ao acúmulo, permite explorar mais combinações e otimizar as rotas. Por outro lado, possui a desvantagem de consumir muito tempo aguardando o acúmulo de todos os pedidos daquele período, o que pode ocasionar a perda de possibilidades de alocação, já que os veículos estão em atividade, visitando alguns de seus consumidores previamente conhecidos.

Em contrapartida, apesar do aumento na complexidade da comunicação entre o algoritmo de roteamento e a frota de veículos, os algoritmos de roteamento *online* fornecem a possibilidade de responder a nova demanda em um curto espaço de tempo. Esta agilidade pode significar a possibilidade de atender o novo cliente no mesmo dia. Tal eficiência pode também reduzir gastos com a utilização de uma menor quantidade de veículos ou no tamanho do percurso total percorrido pela frota. A principal desvantagem da utilização do roteamento *online* é a dificuldade na resolução de um problema NP-Difícil em um curto espaço de tempo.

1.3 Objetivo principal e contribuições da tese

1.3.1 Objetivo principal

Sob a ótica da redução da distância total percorrida pela frota veicular, o objetivo principal da tese é mostrar que a utilização do roteamento *online* no tratamento do PDRV pode ser mais vantajosa que a utilização do roteamento periódico. Com o alcance deste objetivo, espera-se que as futuras pesquisas, relacionadas com a área de transporte, explorem com mais rigor algoritmos de roteamento mais ágeis.

1.3.2 Principais contribuições

A contribuição principal desta tese, relacionada com a comparação entre os diferentes tipos de roteamento (*online* e duas variações periódicas), foi compilada no artigo "*Online Dispatch for the Dynamic Vehicle Routing Problem*" para ser submetido ao periódico *Transportation Science*. Além da conclusão principal, algumas contribuições foram obtidas ao longo da construção do trabalho.

A primeira delas está relacionada com a descrição de um algoritmo eficaz e eficiente para o PRVJT estático, que pode ser aplicado na resolução do PDRV devido à sua característica de alcançar bons resultados em curtos períodos de tempo. Resultados preliminares deste algoritmo foram publicados no Simpósio Brasileiro de Pesquisa Operacional em 2008, no trabalho "Um Algoritmo Híbrido Baseado na Geração de Colunas para o Problema de Roteamento de Veículos com Janela de Tempo".

Depois de alcançado um bom algoritmo para ser utilizado dentro do contexto dinâmico do problema, o método foi adaptado com o intuito de gerar bons resultados para a versão estática, em que o tempo de resolução do problema não é um fator tão crítico como no PDRV. Geralmente, um problema estático no mundo real possui disponível toda uma noite para processamento do algoritmo, antes da saída dos veículos do depósito. Assim, o tempo de processamento é elevado, e o algoritmo adaptado. Esta adaptação obteve melhores resultados para as bem conhecidas instâncias de Solomon [1987]. Os resultados dos dois algoritmos descritos para o PRVJT estático foram recentemente submetidos ao periódico *Computers & Operations Research*, sob o trabalho intitulado “*Hybrid Heuristics for the Vehicle Routing Problem with Time Windows Based on Set Partitioning Formulation*”.

Para efetuar uma análise confiável em torno dos algoritmos de roteamento em foco, uma alternativa foi a descrição e implementação de um simulador evento-discreto para o PDRV. A partir desta pesquisa, este simulador pode ser estendido para outros problemas derivados do PDRV ou do PRV Estocástico.

Para testes da primeira versão do simulador implementado, foram avaliados os impactos de diferentes meta-heurísticas na resolução do PDRV: Algoritmo Memético, *Simulated Annealing*, *Hill Climbing* e GRASP. Todas as meta-heurísticas foram testadas utilizando e não utilizando a estratégia de espera veicular (detalhes na Seção 3.3.2), considerando a minimização da distância total viajada pelos veículos, diferentemente dos outros trabalhos da literatura que reduzem a quantidade de veículos utilizados. Os resultados foram positivos e publicados no *23rd Annual ACM Symposium on Applied Computing*, sob o título “*A Vehicular Waiting Time Heuristic for Dynamic Vehicle Routing Problem*”.

Outra contribuição deste trabalho foi uma extensão do modelo matemático de Larsen [2001] para o PRVJT. Esta extensão foi proposta com o objetivo de calcular os limites inferiores para as instâncias do problema dinâmico tratado neste trabalho.

1.4 Organização da tese

O Capítulo 2 contextualiza a classe dos Problemas de Roteamento de Veículos, formalizando seus conceitos e variações clássicas. Após a definição genérica da classe de problemas, é apresentado o Problema de Roteamento de Veículos Capacitado, que

é base para o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT). Do PRVJT são mostrados dois modelos matemáticos, descrevendo-os em detalhes. Uma seção é dedicada à explicação da complexidade do Problema de Roteamento de Veículos com Janela de Tempo. Na sequência, o Problema Dinâmico de Roteamento de Veículos (PDRV) é descrito, além de detalhadas algumas de suas variações clássicas encontradas na literatura. Para este, também é discutida sua complexidade e a dificuldade da aplicação de métodos exatos para resolvê-lo. Pelo relacionamento com o PDRV, o PRV Estocástico é apresentado na sequência do Capítulo 2. São descritos alguns exemplos em que informações probabilísticas podem ser utilizadas no PDRV, com o objetivo de prever situações futuras.

O Capítulo 3 apresenta uma revisão bibliográfica sobre os principais trabalhos existentes para solucionar o Problema Dinâmico de Roteamento de Veículos e algumas variações estocásticas.

O Capítulo 4 apresenta detalhadamente a metodologia utilizada neste trabalho. A hipótese central da tese é descrita na Seção 4.2. A base de dados estendida para o PDRV é explicada na Seção 4.3. O experimento realizado para a avaliação estatística da hipótese é detalhado na Seção 4.4.

Para avaliar os tipos de roteamento em análise nesta tese, foi necessária a construção de um simulador para o PDRV. Este é descrito no Capítulo 5. Primeiramente, são descritos na Seção 5.2, conceitos básicos sobre simulação evento-discreta. Eventos relacionados com o processo de simulação para o PDRV são detalhados na Seção 5.3.

Os métodos descritos nesta tese para a resolução do PRVJT estático e do PDRV são descritos no Capítulo 6. As Seções 6.2 e 6.3 apresentam dois algoritmos híbridos para o PRVJT. A resolução do PDRV é detalhada na Seção 6.4.

O Capítulo 7 apresenta os resultados da resolução do PRVJT estático (Seção 7.2) e da solução do PDRV (Seção 7.3). Os resultados relacionados com o experimento que avalia a hipótese da tese são descritos na Seção 7.3.2. A análise estatística é apresentada na Seção 7.3.3. Comparações gráficas entre os algoritmos de roteamento em análise são apresentadas na Seção 7.3.4. Para referência, limites inferiores para as instâncias dinâmicas estendidas são apresentados na Seção 7.3.5.

Todas as conclusões do autor relacionadas com este trabalho são apresentadas no Capítulo 8.

Capítulo 2

O Problema de Roteamento de Veículos e suas variações

2.1 Considerações iniciais

Devido à quantidade de problemas práticos envolvendo o roteamento de veículos, a definição inicial do problema, proposta por Dantzig e Ramser [1959], vem sendo estudada e estendida ao longo das últimas décadas.

Basicamente, estes problemas se resumem ao atendimento de uma demanda que pode se apresentar na forma de coleta e/ou entrega de pessoas ou mercadorias em uma determinada região. A maioria das aplicações do PRV é geográfica e representada por consumidores distribuídos em uma área limitada de atendimento. Desta forma, o objetivo das pesquisas na área de roteamento é desenvolver métodos para atender as demandas do PRV de forma otimizada, visando a redução dos gastos com veículos e com o deslocamento dos mesmos.

Este capítulo apresenta, na Seção 2.2, variações clássicas do PRV. A Seção 2.3 descreve o PRV Capacitado, seguida do PRV com Janela de Tempo (Seção 2.4). A versão dinâmica do problema é descrita na Seção 2.5 e a versão estocástica na Seção 2.6. O sumário do capítulo é apresentado na Seção 2.7.

2.2 Variações clássicas do Problema de Roteamento de Veículos Estático

No trabalho de Gendreau e Potvin [1998] são apresentados quatro tipos genéricos do PRV, conforme a Tabela 2.1:

Tabela 2.1 - Classificação genérica para os variados tipos de problemas de roteamento

	Muitos para muitos	Um para muitos
Capacitado	<i>transporte de passageiros</i>	<i>entrega de comida</i>
Não capacitado	<i>serviços de correio</i>	<i>serviços de reparo</i>

Duas importantes subdivisões para o PRV são detalhadas na pesquisa de Gendreau e Potvin [1998]. A primeira é a clara separação dos problemas com e sem capacidade de carga em seus veículos. Esta divisão existe, pois nem todos os serviços de transporte necessitam carregar mercadorias, ou simplesmente suas cargas possuem pesos e tamanhos irrelevantes. Pode-se citar como exemplo para o PRV sem a necessidade de veículos capacitados a realização de serviços de um eletricista. Estes profissionais não necessitam de um veículo com capacidade de carga, pois as demandas são de serviços, e não de mercadorias. A outra importante subdivisão citada é com relação à carga e descarga de pessoas ou mercadorias. Duas opções são apresentadas: pode-se coletar pessoas ou mercadorias em um único lugar em comum, para deixá-las em diferentes pontos de entrega (relacionamento 1 para N), ou pode-se coletá-las em diferentes pontos, deixando-as em diferentes locais de entrega (relacionamento M para N).

Gendreau e Potvin [1998] exemplificam o relacionamento de um PRV com coleta e entrega do tipo M para N , associado a um veículo capacitado, como um típico problema de coleta e entrega de passageiros. Já o caso no qual as mercadorias saem apenas de um ponto (1 para N) e as mercadorias podem exceder a capacidade do veículo é exemplificado com o típico problema de entrega de comida de um restaurante, no qual o produto deve ser levado a diversas casas e escritórios em pontos alternados da cidade. Os trabalhos nos quais o produto não excede a capacidade do veículo são citados como relacionamento M para N com o exemplo da coleta e entrega de cartas de um serviço de correio, em que as cartas durante um dia de serviço dificilmente irão exceder a capacidade dos veículos de entrega. Para o problema de relacionamento origem/destino de 1 para N adicionada à característica de ausência de veículos capacitados, cita-se o exemplo da prestação de serviços, em que um veículo deixa uma loja especializada levando um prestador de serviços (um eletricista, por exemplo) para atender um conjunto de consumidores durante um período de tempo. Todas estas subdivisões do problema podem ser associadas com características dinâmicas e com janelas de tempo [Gendreau e Potvin, 1998].

Diferentes definições e particularidades são encontradas na literatura na definição e classificação dos problemas de roteamento de veículos. Xu *et al.* [2003] já classificam o PRV de forma diferente ao trabalho de Gendreau e Potvin [1998]. Como todos os autores, consideram que o objetivo do PRV é basicamente o uso eficiente da frota de veículos ao coletar e entregar encomendas, mas separam os problemas em três grandes classes:

- Problema de Roteamento de Veículos Capacitado (PRVC), descrito com detalhes na Seção 2.3. Eles não explicitam o relacionamento entre o número de pontos de entrega e pontos de coleta, mas, segundo seus exemplos, pode-se verificar que se trata do relacionamento 1 para N , como o exemplo da entrega de comida de um restaurante de Gendreau e Potvin [1998];
- Problema de Roteamento de Veículos com Janela de Tempo (PRVJT), que é uma generalização do PRVC. O PRVJT é descrito na Seção 2.4;
- Problema de Roteamento de Veículos com Pedidos de Coleta e Entrega (PRVCE), que, diferentemente do PRVC e do PRVJT, efetua dois tipos de serviços: entrega e coleta de mercadorias. Neste problema, cada cliente faz, ao mesmo tempo, os dois tipos de pedidos, e cada veículo transporta uma mistura de pedidos de entrega e coleta. Mais detalhes sobre o PRVCE podem ser obtidos no trabalho de Assis [2007].

Um exemplo do problema de coleta e entrega é encontrado na indústria de transporte de bebidas e alimentos. Além da entrega de produtos, é necessário recolher outros com prazo de validade vencido [Freitas e Montané, 2008]. Fazendo uma comparação com o trabalho de Gendreau e Potvin [1998], o PRVCE é o relacionamento entre pontos de coleta e entrega M para N . Já o problema apenas de coleta ou de entrega pode ser encontrado nos serviços de entrega de cartas ou encomendas, de entrega de jornais, e de coleta de lixo municipal. Nesses casos, se as encomendas não têm um tempo pré-determinado para as entregas ou para as coletas, pode-se utilizar um algoritmo que resolve o PRVC. Se for estipulado um tempo de atendimento a cada consumidor, é necessária a aplicação de um sistema que resolva o PRVJT. Geralmente, as restrições de janela de tempo estão presentes nos serviços relacionados ao comércio, onde há o estabelecimento de períodos como horário comercial, horário de almoço, turno de trabalhadores, dentre outros detalhes.

O trabalho de Xu *et al.* [2003] não apresenta classificações para situações em que os veículos não necessitam ter capacidade de carga, como é detalhado no trabalho de Gendreau e Potvin [1998]. Em contrapartida, sua classificação é mais específica e contém características como a Janela de Tempo, sendo que este detalhamento não é apresentado no trabalho de Gendreau e Potvin [1998].

Todos os problemas derivados do PRVC possuem uma abordagem associada, considerando-se as restrições de tempo, como a relação entre o PRVC e o PRVJT. Pode-se citar o Problema de Roteamento de Veículos com Janela de Tempo (descrito na Seção 2.4) como uma das extensões mais estudadas do PRVC [Alvarenga, 2005].

Além do PRVJT, outros problemas de roteamento se originam do PRVC, com definições que visam a tratar particularidades que eventualmente ocorrem em problemas no mundo real. Exemplos de problemas derivados do PRVC, comumente encontrados na literatura são:

- Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD);
- Problema de Roteamento de Veículos Periódico (PRVP); e
- Problema de Roteamento de Veículos com Entrega Particionada (PRVEP).

O PRVMD não possui apenas um depósito como o PRVC. Vários depósitos estão disponíveis para atender toda a demanda dos consumidores. Cada depósito possui uma quantidade exclusiva de veículos. No PRVMD é definida, como restrição básica, que todo veículo deve retornar para seu depósito de origem. Após a definição de quais consumidores cada depósito atende, o PRVMD pode ser tratado como múltiplas instâncias do PRVC.

O PRVP define que cada consumidor pode ser atendido em k dias. Para este problema, o veículo pode não retornar ao depósito no mesmo dia da partida. Esta definição se torna útil quando o sistema de roteamento trabalha com grandes áreas de atendimento.

O PRVEP é um problema modelado para o atendimento de grandes entregas de mercadorias. No mundo real isso ocorre quando apenas um veículo não pode atender toda a demanda de um consumidor em uma única viagem. Neste caso, é necessário efetuar uma entrega particionada, na qual vários veículos podem atender a demanda de um consumidor para suprir toda a sua necessidade.

Na maioria das aplicações do PRV no mundo real, ao modelar o problema, as empresas encontram algum tipo de dinamicidade, que são alterações no ambiente durante o tempo de atendimento à demanda. Estes fatores caracterizam a classe conhecida como Problemas Dinâmicos de Roteamento de Veículos. Esta dinamicidade pode ser apresentada na forma de, por exemplo, novas requisições de atendimento ou cancelamento de pedidos. Mais detalhes são encontrados na Seção 2.5. Basicamente, não há uma definição específica para o PDRV como existe para os PRVs estáticos.

Um importante fator a ser considerado para o PDRV, e que não é tratado no PRV estático, é a diferenciação das abordagens correlacionadas a coletas e entregas. Se um problema é exclusivo de coletas ou de entregas, o PRV estático pode modelá-lo de forma equivalente, pois toda a demanda é conhecida *a priori*. Já para o PDRV não é possível adicionar mercadorias em veículos que já estão em circulação e, por este motivo, deve-se tratar os problemas de coleta e entrega de forma diferenciada [Kilby *et al.*, 1998].

2.3 O Problema de Roteamento de Veículos Capacitado (PRVC)

O Problema de Roteamento de Veículos Capacitado é uma extensão do mais conhecido problema de otimização combinatória, o Problema do Caixeiro Viajante (PCV), no qual se visa atender todo um conjunto de cidades (consumidores), minimizando a distância total percorrida pelo veículo do caixeiro viajante [Papadimitriou e Steiglitz, 1982].

O PRVC foi descrito inicialmente em Dantzig e Ramser [1959]. Na literatura, considera-se que existe um depósito central, ou garagem, para fornecimento dos veículos de atendimento. Todos os veículos são idênticos e possuem uma capacidade máxima. Estes veículos devem atender todo um conjunto de consumidores que demandam serviços (coleta ou entrega de mercadorias).

A Figura 2.1 apresenta uma possível solução para uma demanda do PRVC, em que 11 consumidores de recursos estão distribuídos e cada consumidor possui uma demanda associada. Desta forma, a Figura 2.1 apresenta, como possível solução, três rotas veiculares para o atendimento das requisições, sendo que cada veículo tem, como capacidade máxima, 25 unidades de carga.

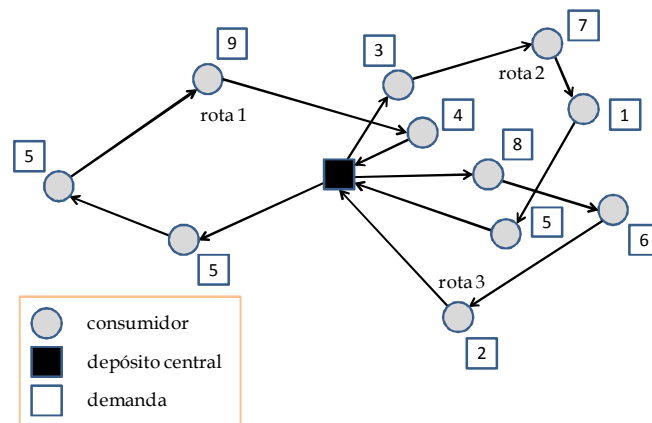


Figura 2.1 – Possível solução para um Problema de Roteamento de Veículos

Se a solução de uma instância do PRVC se resumir apenas a um veículo, este é simplificado ao Problema do Caixeiro Viajante. Esta situação ocorre no PRVC quando a capacidade de carga de um veículo é maior que o somatório de todas as demandas dos consumidores e não existem restrições de outra natureza, como janelas de tempo, fazendo com que o problema se resuma ao encontro do menor percurso do veículo que atenda a todas as demandas dos consumidores, iniciando e terminando a rota no depósito central.

2.4 O Problema de Roteamento de Veículos com Janela de Tempo (PRVJT)

O PRVJT assume todas as características do PRVC descritas na Seção 2.3 com a adição de um novo fator: o tempo.

Assume-se que cada consumidor de recursos possui um intervalo de tempo pré-determinado para ser atendido por um dos veículos da frota do depósito central (garagem). Isso faz com que o atendimento sobre cada consumidor ocorra em uma “janela de tempo”, sendo este o fator que nomeia o problema.

É importante ressaltar que a situação de um veículo chegar a um determinado consumidor antes da abertura de sua janela de tempo é permitida. Neste caso, o mesmo deve aguardar a abertura da janela de tempo do consumidor para iniciar seu

atendimento. A situação não permitida é o veículo chegar ao consumidor após o fechamento de sua janela de tempo (atraso).

Outro fator que tem impacto direto na solução do PRVJT é o tempo de atendimento de cada consumidor, conhecido como tempo de serviço. Após o veículo iniciar o atendimento no consumidor (dentro da janela de tempo permitida), o mesmo ali deve permanecer até cumprir o período de tempo necessário para o completo atendimento. Cada consumidor possui uma janela de tempo e um tempo de atendimento individual.

Como no PRVC, um veículo do PRVJT também não pode atender uma demanda maior que sua capacidade de carga. Estas restrições são detalhadas matematicamente na Seção 2.4.2.

2.4.1 Otimização das viagens no PRVJT

Calcular a melhor solução para o PRVJT depende diretamente dos custos de cada elemento da solução. Como trabalhos acadêmicos resolvem os problemas de roteamento de maneira genérica, ou seja, para quaisquer veículos e custos de transporte, são encontrados, em grande número, trabalhos que reduzem os custos das mais variadas formas ou exploram outras métricas no processo de otimização:

- Minimização do número de veículos;
- Minimização da distância total percorrida;
- Minimização do tempo total de atendimento;
- Maximização do número de consumidores atendidos;

Outros fatores podem estar presentes no mundo real, e cabe ao projetista inserir tais características na função objetivo do modelo aplicado.

2.4.2 Um modelo matemático para o PRVJT

“A importância e influência do modo de formular um problema de otimização, especialmente em áreas complexas como as de roteamento, recobrimento etc. devem

ser bem entendidas. O motivo é evidente: a formulação terá impacto direto no desempenho dos algoritmos de solução.” [Goldbarg e Luna, 2000].

Conforme Larsen [2001], o PRVJT pode ser formulado da seguinte maneira: um conjunto de veículos idênticos, representado pelo conjunto V , necessita realizar coletas ou entregas em uma região, que é representada por um grafo orientado G . O grafo G consiste em $|C|+2$ vértices, em que os consumidores a serem visitados são representados pelo conjunto $C = \{c_1, c_2, \dots, c_n\}$, e o depósito é representado por c_0 (depósito de saída) e c_{n+1} (depósito de chegada). No caso do PRVJT, existe apenas um depósito central. Para fins de simplificação das restrições do modelo, o depósito é representado por c_0 e c_{n+1} . Assim, o conjunto de vértices do grafo G é representado por $N = C \cup \{c_0, c_{n+1}\}$. O conjunto de arestas A do grafo G representa as conexões entre o depósito e os consumidores, e entre consumidores. Não existem arestas que terminam no vértice c_0 , e não existem arestas que se originam do vértice c_{n+1} . A cada aresta (i,j) , em que $i \neq j$, é associado um custo d_{ij} e um período de tempo t_{ij} , proporcionais à distância entre os vértices i e j .

Cada veículo possui uma capacidade q e cada consumidor i possui uma demanda w_i , um tempo de serviço p_i e uma janela de tempo $[a_i, b_i]$. Um veículo deve chegar ao consumidor i antes do fim de sua janela de tempo b_i . É permitida a chegada do veículo no consumidor i antes de a_i , mas este não pode iniciar o serviço antes de a_i . O horizonte de planejamento do PRVJT é descrito pelo intervalo entre a_0 e b_{n+1} . Veículos não podem sair do depósito antes de a_0 e não podem voltar ao depósito após b_{n+1} .

O modelo possui dois conjuntos de variáveis de decisão: x e s . Para cada aresta (i,j) , em que $i \neq j$, $i \neq n+1$, $j \neq 0$, e cada veículo k , a variável x_{ijk} é definida como:

$$x_{ijk} = \begin{cases} 1, & \text{se o veículo } k \text{ percorre a aresta } (i,j) \\ 0, & \text{se o veículo } k \text{ não percorre a aresta } (i,j) \end{cases} \quad (2.1)$$

As variáveis de decisão s_{ik} são definidas para cada vértice $i \in N$ e cada veículo $k \in V$ e representam o instante de tempo em que o veículo k inicia o serviço no vértice i . É assumido que $a_0 = 0$, e $s_{0k} = 0$, para todo veículo $k \in V$.

Com os parâmetros e variáveis de decisão bem definidos, pode-se descrever matematicamente o PRVJT como:

$$\min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijk} \quad \text{sujeito a} \quad (2.2)$$

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1 \quad \forall i \in C \quad (2.3)$$

$$\sum_{i \in C} w_i \sum_{j \in N} x_{ijk} \leq q \quad \forall k \in V \quad (2.4)$$

$$\sum_{j \in N} x_{0,jk} = 1 \quad \forall k \in V \quad (2.5)$$

$$\sum_{i \in N} x_{i(n+1)k} = 1 \quad \forall k \in V \quad (2.6)$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0 \quad \forall h \in C; \quad \forall k \in V \quad (2.7)$$

$$s_{ik} + p_i + t_{ij} - K(1 - x_{ijk}) \leq s_{jk} \quad \forall i, j \in N; \quad \forall k \in V \quad (2.8)$$

$$a_i \leq s_{ik} \leq b_i \quad \forall i \in N; \quad \forall k \in V \quad (2.9)$$

$$x_{ijk} \in \{0,1\} \quad \forall i, j \in N; \quad \forall k \in V \quad (2.10)$$

As restrições indicadas pela Equação 2.3 garantem que cada consumidor i é visitado uma única vez por um único veículo. Cada veículo k atende somente um conjunto de consumidores cuja demanda total não ultrapassa a sua capacidade q (Equação 2.4). As Equações 2.5, 2.6 e 2.7 indicam restrições que garantem que cada veículo k parte do depósito central, visita seus consumidores e ao final retorna ao depósito central. A Equação 2.8 apresenta as restrições que garantem que os veículos visitam seus consumidores, respeitando seus tempos de serviço e seus tempos de deslocamento entre os consumidores, em que o instante de início do atendimento de um consumidor j por um veículo k não poderá ocorrer antes do início do atendimento do consumidor anterior i (s_{ik}), mais o tempo de serviço no consumidor i (p_i), mais o tempo de percurso no trecho (i,j) , que é t_{ij} . É assumida uma velocidade constante tal que o tempo de percurso t_{ij} é igual à distância entre i e j (d_{ij}). Na Equação 2.8, a constante K representa um valor escalar suficientemente grande. O respeito ao intervalo da janela de tempo do consumidor é garantido pelas restrições descritas na Equação 2.9, em que o instante de início de atendimento do consumidor i pelo veículo k está dentro dos limites definidos pela janela de tempo $[a_i, b_i]$. A Equação 2.10 garante que cada variável do conjunto x assume valores binários. Observa-se, no modelo apresentado, um custo associado a cada aresta do grafo (d_{ij}), correspondendo à

distância do consumidor i ao consumidor j . Considera-se neste modelo a minimização da distância total percorrida por todos os veículos da solução do PRVJT (Equação 2.2).

Para fins de simplificação, este trabalho considera t_{ij} igual a d_{ij} para todo par (i,j) , sendo $i, j \in N$. Em casos reais, é interessante atribuir a cada variável t_{ij} o tempo médio que um veículo da frota leva para percorrer o trecho entre os consumidores i e j .

A formulação apresentada nesta seção possui uma quantidade de restrições e variáveis polinomial, em função do número de clientes e veículos.

Uma adaptação deste modelo matemático proposta nesta tese é utilizada para encontrar limites inferiores para as instâncias dinâmicas usadas na avaliação dos algoritmos de roteamento no PDRV. Os resultados são apresentados e discutidos na Seção 7.3.5.

2.4.3 A formulação do Problema de Particionamento de Conjuntos adaptada ao PRVJT

Vários problemas de otimização combinatória podem ser descritos tendo como referência um Problema de Particionamento de Conjunto (PPC) [Santos, 2008]. Para o PRVJT, cada coluna corresponde a uma rota viável candidata a pertencer à solução do problema, e as linhas estão relacionadas com os consumidores que deverão ser atendidos por uma única rota, uma única vez. O modelo matemático do PRVJT na forma de um PPC é descrito pelas Equações 2.11, 2.12 e 2.13.

$$\min \sum_{r \in R} c_r x_r \quad \text{sujeito a} \quad (2.11)$$

$$\sum_{r \in R} \delta_{ir} x_r = 1 \quad \forall i \in C \quad (2.12)$$

$$x_r \in \{0,1\} \quad \forall r \in R \quad (2.13)$$

A Equação 2.11 corresponde à função objetivo. O conjunto R representa todas as rotas viáveis para o PRVJT. Para cada rota r , existe um custo associado, c_r (distância percorrida). O objetivo do problema consiste em encontrar o conjunto de rotas de menor custo, sujeito às restrições do problema (no caso do PPC, cada consumidor deve ser atendido uma única vez). A variável de decisão x_r é binária, conforme a Equação 2.13 (restrição do PPC), sendo igual a 1 se a rota r faz parte da solução e igual

a 0, em caso contrário. O parâmetro δ_r é igual a 1 se o consumidor i pertence a rota r e 0, em caso contrário. Desta forma, a condição imposta pela Equação 2.12 (restrição do PPC) assegura que cada consumidor será atendido por uma única rota, uma única vez.

Em função da quantidade de consumidores que devem ser atendidos e rotas viáveis, esta formulação apresenta um número exponencial de variáveis (colunas) e um número polinomial de restrições (linhas). Por este motivo, na prática, os trabalhos utilizam, ao invés do conjunto R , um conjunto reduzido de rotas R' . Larsen [2001] e Alvarenga [2005] utilizam com sucesso o modelo de geração de colunas através do PPC para encontrar soluções para o PRVJT. Heurísticas são responsáveis pela geração de colunas para o conjunto R' , e depois o PPC é resolvido através de um *software* de programação matemática. Outros problemas combinatórios são também bem resolvidos com a formulação do PPC. Pode-se citar, por exemplo, Mingozzi *et al.* [1999] que resolve o Problema de Programação de Tripulações.

Quando é utilizado o conjunto reduzido R' , pode-se alternativamente relaxar as restrições da equação 2.12 do PPC para o conjunto de desigualdades definidos pela seguinte equação:

$$\sum_{r \in R'} \delta_{ir} x_r \geq 1 \quad \forall i \in C \quad (2.14)$$

A troca das igualdades pelas desigualdades (uma para cada consumidor) permite que um consumidor seja visitado por mais de um veículo ao resolver o PPC. Isso viola a restrição básica do PRVJT, que define que um consumidor deve ser visitado por apenas um veículo. Através de um procedimento guloso, é possível remover a visita (geralmente dupla) que eventualmente pode ser encontrada na resolução do PPC relaxado, minimizando ainda mais os custos da solução do PRVJT. Esta estratégia foi utilizada com sucesso ao longo deste trabalho. Os resultados preliminares apresentaram melhorias de custo da distância total viajada, se comparado com o PPC original, mas o método foi abandonado, devido ao alto tempo computacional exigido pela relaxação. Neste trabalho, é necessário resolver rapidamente o PRVJT estático para ser utilizado dentro do algoritmo de roteamento do PDRV. Mais detalhes sobre a utilização do modelo do PPC neste trabalho são apresentados no Capítulo 6. Em trabalhos em que o tempo para solução do problema não é um fator crucial, esta relaxação pode ser mais explorada.

2.4.4 Complexidade do PRVJT

Na década de 70, Cook [1971] mostrou a existência de um problema com uma importante propriedade, chamado Problema de Satisfatibilidade Booleana (SAT). O SAT pode ser considerado especial, pois, para cada um dos problemas que possuem algoritmos de resolução (pertencentes à classe NP), e para cada uma de suas respectivas instâncias, existe pelo menos uma instância equivalente para o SAT. Assim, um procedimento algorítmico que resolve o SAT pode resolver qualquer problema pertencente à classe NP. Por este motivo, o SAT foi considerado um problema completo, inaugurando a conhecida e especial classe NP-completo.

Desde a prova de Stephen Cook, existem duas iniciativas principais relacionadas aos problemas NP-completos na comunidade científica. A primeira delas é a tentativa de redução de problemas da classe NP-completo para problemas que ainda não são conhecidos algoritmos polinomiais em tempo ou espaço. Isso faz com que tais problemas sejam considerados também da classe NP-completo, pois, se todos podem ser transformados no SAT, e o SAT pode ser transformado neles, direta ou indiretamente, então todos os problemas pertencentes a classe NP-completo podem ser resolvidos através do mesmo algoritmo (considerando as transformações entre os problemas). A segunda iniciativa está relacionada com a busca de um algoritmo polinomial para qualquer um dos problemas pertencentes à classe NP-completo. Isso provaria que todos os problemas da classe NP podem ser resolvidos em tempo polinomial, por um procedimento determinístico, em função do tamanho da entrada. Ainda não se sabe se tal procedimento determinístico existe. Várias tentativas de prova já foram apresentadas, mas nenhuma delas foi aceita pela comunidade científica.

O PRVJT é classificado como NP-completo [Garey e Johnson, 1979]. Devido à dificuldade de tratar o PRVJT, com um número elevado de clientes e veículos, através de algoritmos exatos, frequentemente heurísticas são utilizadas para encontrar soluções em tempo hábil. Contudo, é necessário que a busca de soluções possua uma qualidade mínima. Tal qualidade é dependente do tempo que a heurística tem disponível na busca de soluções. Pode-se, portanto, fazer uso de métodos estatísticos para determinar a capacidade e a maturidade dos modelos heurísticos.

Dentro deste contexto de tempo e qualidade, Eiben e Smith [2003], descrevem três grandes classes de problemas de otimização:

- Problemas urgentes;
- Problemas repetitivos;
- Problemas de projeto.

Os problemas ditos urgentes possuem um tempo curto para obter uma solução viável. Portanto, soluções devem ser encontradas rapidamente, com o apoio de heurísticas ou meta-heurísticas disponíveis para o problema. Este tipo de situação é muito comum dentro do contexto do roteamento *online* para o PDRV, em que rápidas decisões devem ser tomadas, pois a frota de veículos está em constante operação durante a execução do algoritmo e a velocidade da execução pode ser um fator decisivo na qualidade da alocação final.

Os problemas repetitivos possuem um tempo maior para obter uma solução, se comparados aos problemas urgentes. Pode-se considerar este tempo como um período noturno de execução computacional para a implantação da solução no início do próximo dia de trabalho do PRV, por exemplo. Neste caso, é mais indicada a execução de abordagens mais robustas, como as meta-heurísticas, que geralmente demandam um tempo maior de execução, mas as mesmas não deixam de ser executadas em um tempo hábil previamente especificado. A definição do problema repetitivo se enquadra na resolução do PRV estático, que é geralmente útil para resolver a demanda inicial do dia no PDRV. Neste caso, alguns consumidores são conhecidos *a priori*, e um PRV estático é resolvido com os pedidos previamente conhecidos. Na maioria dos trabalhos encontrados na literatura, a solução do PRV estático serve de entrada para o PDRV iniciar a alocação dos veículos no início do dia de trabalho.

Os problemas ditos de projeto, por sua vez, podem ser resolvidos em um tempo maior de processamento e são geralmente problemas de planejamento, não se aplicando à classe dos PRVs. Mais informações sobre problema e projeto são encontradas no trabalho de Eiben e Smith [2003].

2.5 O Problema Dinâmico de Roteamento de Veículos

O Problema Dinâmico de Roteamento de Veículos (PDRV) possui as mesmas características do PRV, além de apresentar mudanças no ambiente ou na demanda ao longo do tempo, como, por exemplo:

- novas requisições de atendimento durante o dia de serviço (coleta ou entrega);
- cancelamento de encomendas (coleta ou entrega);
- bloqueio de caminhos, como, por exemplo, uma estrada é interditada a partir de um momento do dia em função de uma obra;
- velocidade variável dos veículos, podendo ser descritas por funções que computam outras características do ambiente além da distância total viajada como, por exemplo, horários propícios a congestionamentos;
- indisponibilidade de veículos ao longo do dia, por exemplo, um veículo passa a não poder exercer coletas e/ou entregas por apresentar um problema mecânico ao longo do percurso.

Na versão dinâmica do PRV, informações estocásticas podem também estar disponíveis, geralmente obtidas através do histórico de requisições ou de informações sobre o ambiente. Os trabalhos Bent e Van Hentenryck (2004) e Hvattum *et al.* (2006) afirmam que poucos trabalhos relatam o PDRV com informações estocásticas.

O Capítulo 3 apresenta uma revisão bibliográfica sobre o PDRV.

2.5.1 Complexidade e tratabilidade do PDRV

Dado que os problemas da classe dos PRVs são problemas NP-Difíceis e os PDRVs precisam geralmente de rápidas respostas para o atendimento dos consumidores, os algoritmos exatos ainda não são capazes de resolver com eficiência problemas desta natureza em tempo hábil [Ichoua *et al.*, 2000]. Isso faz com que o problema, quando voltado para o mundo real, seja resolvido na maioria das vezes por heurísticas.

Larsen [2001] resume bem a situação dos métodos exatos para o PDRV: “o rápido desenvolvimento no *hardware* disponível, somado ao grande avanço dos *software* de otimização, tem alterado as fronteiras do que pode ser resolvido por métodos de otimização em um tempo computacional razoável. No entanto, mesmo problemas estáticos de roteamento do mundo real têm sido computacionalmente difíceis para tais métodos”.

Outra dificuldade é observada para a resolução dos problemas da classe dos PDRVs. Suponha que exista algoritmo polinomial exato para resolver os problemas da classe dos PRVs estáticos em tempo hábil. Se discretizarmos o tempo, segundo após segundo, pode-se, a cada nova requisição, considerar a situação atual do Problema Dinâmico de Roteamento de Veículos, com os veículos em andamento, como uma variação do PRVC estático. Se for aplicado o suposto algoritmo exato, a cada nova requisição, realocando os veículos com as novas demandas dinâmicas, não necessariamente será alcançado, ao final, a melhor solução possível, seja qual for o objetivo a ser minimizado ou maximizado. Isso porque, além da dificuldade matemática do processo, ele também está sujeito ao acaso. Uma solução ótima no instante n pode levar um veículo, por exemplo, para o lado sul do mapa, e no futuro, um consumidor dinâmico aparecer no lado norte. Isso acaba aumentando os custos no final da otimização. Visando controlar esta situação, alguns pesquisadores criam estratégias de cobertura do território [Ichoua *et al.*, 2006; Branke *et al.*, 2005]. Teoricamente, se o algoritmo de roteamento possui, a cada instante de tempo veículos bem distribuídos geograficamente, é mais fácil atender novas demandas dinâmicas, se comparada com uma configuração da frota veicular aglomerada em uma região do mapa. Mas tal estratégia pode, por exemplo, aumentar a distância total percorrida, devido ao espalhamento da frota, ou aumentar a quantidade de veículos necessária para atender a toda demanda. Se o objetivo é a maximização dos consumidores atendidos, pode ser uma tática interessante. No contexto de uma melhor cobertura, informações históricas podem auxiliar no processo de otimização.

2.6 O Problema de Roteamento de Veículos Estocástico

Alvarenga [2005] considera que o PRV Estocástico (PRVE) é uma forma de tratar alguns tipos de dinamismos de forma antecipada. A maioria dos trabalhos que

trabalha com versões estocásticas ou dinâmicas do PRV, foca em uma das duas versões exclusivamente [Bent e Van Hentenryck, 2004].

Assim como a classe dos PDRVs, o PRVE representa um conjunto de problemas que consideram a ocorrência de novos eventos ao longo do tempo. Diferentemente dos PDRVs, os PRVEs possuem informações sobre eventos futuros (dinâmicos), no formato de probabilidades, que são disponibilizadas para o algoritmo de roteamento. Utilizando estas informações, o algoritmo de roteamento pode tentar otimizar sua alocação de recursos, baseando-se nas probabilidades disponibilizadas sobre os prováveis eventos.

Nesta classe de problemas, os consumidores e suas demandas são variáveis aleatórias discretas ou contínuas. Informações probabilísticas sobre eventos futuros são usadas para construir uma solução que otimiza o valor esperado de uma função objetivo [Ichoua *et al.*, 2006].

O problema base desta classe é o Problema do Caixeiro Viajante Probabilístico, em que cada consumidor possui uma probabilidade de requisitar um serviço. O objetivo é encontrar uma rota que minimiza a distância total percorrida. Este problema foi proposto por Jaillet [1985].

O Problema de Roteamento de Veículos com Demanda Estocástica possui veículos de capacidade fixa e consumidores com uma provável demanda para atendimento. A demanda é representada através de distribuições de probabilidade. Assim, o veículo não deve ultrapassar sua capacidade de carga, utilizando, como previsão, as informações históricas disponíveis. Detalhes sobre este problema podem ser obtidos no trabalho de Secomandi [2000].

Já no Problema de Roteamento de Veículos com Consumidores Estocásticos, cada consumidor possui determinada probabilidade de requisitar o serviço. Esta é uma extensão do Problema do Caixeiro Viajante Probabilístico. Mais informações e detalhes são encontrados em Waters [1989].

A Seção 3.4 apresenta uma revisão bibliográfica de métodos que tratam diferentes PRVEs.

2.7 Sumário do capítulo

Este capítulo apresentou diferentes taxonomias para os problemas da classe dos PRVs. A mais genérica delas é apresentada no trabalho Gendreau e Potvin [1998], na qual os pesquisadores subdividem a classe em quatro principais problemas que se diferenciam basicamente por duas características:

- Com ou sem capacidade de carga nos veículos;
- Um ou mais pontos de coleta.

Ao longo do capítulo foram apresentadas duas formulações para resolver o PRVJT, que está diretamente relacionado com a versão do PDRV tratada neste trabalho. A primeira formulação é específica para o PRVJT, e outra é baseada no Problema de Particionamento de Conjunto (PPC), que também pode servir como base para a resolução de uma série de problemas combinatórios, o que inclui a classe dos PRVs. Foi discutida a complexidade de se resolver as versões estáticas e dinâmicas do PRV.

Foram apresentadas também algumas características do PDRV e do PRVE encontradas na literatura. O Capítulo 3 apresenta uma revisão bibliográfica sobre o PDRV e sobre o PRVE.

Capítulo 3

Revisão bibliográfica

3.1 Considerações iniciais

Devido à variedade de problemas que derivam as classes dinâmicas e estocásticas do PRV, este capítulo é subdividido. Trabalhos que tratam a dinamicidade dos pedidos com os veículos ainda parados no depósito são apresentados na Seção 3.2. A Seção 3.3 apresenta trabalhos que consideram alterações no ambiente ou na demanda ao longo do dia. Na Seção 3.4 são descritos alguns trabalhos que resolvem o PRVE. O sumário do capítulo é apresentado na Seção 3.5.

Do ponto de vista da classificação, não há conformidade entre as pesquisas do PDRV e do PRVE, como existe no PRV estático. Pesquisadores tratam o problema considerando diferentes formas de dinamismo e diferentes informações históricas como descrito ao longo deste capítulo.

O trabalho de Kilby *et al.* [1998] apresenta um estudo sobre a mudança paramétrica, em que os autores avaliam o impacto nos resultados das heurísticas de acordo com o grau de dinamismo imposto ao PDRV, em que, quanto mais alto o grau de dinamismo, maior é o número de eventos dinâmicos que ocorrerem no ambiente. Esta estratégia de variar os graus de dinamismo também é aplicada nos trabalhos de Alvarenga [2005] e Bent e Van Hentenryck [2004]. No caso do trabalho de Kilby *et al.* [1998], o único tipo de evento dinâmico estudado é o conhecimento de novos consumidores pelo algoritmo de roteamento ao longo do dia de serviço. Além disso, eles propõem uma base de testes para o PDRV estendendo uma base de dados do PRV estático, adicionando as seguintes características específicas:

- momentos em que consumidores são conhecidos pelo algoritmo de roteamento;
- tempo de serviço de cada novo consumidor;
- aumento do número de veículos disponíveis: de 25 para 50.

A extensão da base de testes proposta no trabalho Kilby *et al.* [1998] é similar à realizada neste trabalho. Mais detalhes na Seção 4.3.1.

Montemanni *et al.* [2005] aplicam os conceitos do PDRV em um problema do mundo real, propondo um algoritmo baseado na meta-heurística conhecida como Colônias de Formigas (CF). Mais detalhes sobre CF são encontrados no trabalho Dorigo *et al.* [1996]. Como critérios da simulação, Montemanni *et al.* [2005] consideram que os veículos viajam em uma velocidade constante, além do trabalho buscar a redução do tempo total de atendimento aos consumidores. Os autores apresentam um projeto de *software* definindo de maneira clara um gerenciador de eventos que funciona como a interface entre o sistema otimizador e o ambiente, em que as novas demandas são capturadas. Eles identificam também os diferentes tipos de PDRVs, sendo que suas definições possuem uma intersecção clara com os tipos de PRVs descritos no trabalho de Gendreau e Potvin [1998], citados na Seção 2.2. Parte da arquitetura do *software* descrito neste trabalho foi baseada na organização de Montemanni *et al.* [2005].

Um trabalho similar a Montemanni *et al.* [2005] é o descrito por Gendreau *et al.* [1999], em que uma diferença básica pode ser observada entre eles: Em Gendreau *et al.* [1999] a chegada de uma nova demanda durante o período de atendimento a outros consumidores pode alterar o destino do próximo consumidor que um veículo está para visitar (na literatura esta técnica é conhecida como desvio). Em Montemanni *et al.* [2005] isso não acontece, pois a nova requisição pode alterar alocações veiculares somente depois do próximo consumidor a ser visitado pela frota de veículos. O tratamento destas alterações deve ser feito de maneira cuidadosa, sendo que uma das importantes características do PDRV é o tempo de comunicação entre o algoritmo de roteamento e a frota de veículos. Mais detalhes sobre a técnica de desvio são encontrados na Seção 3.3.3.

Para tratar o Problema de Roteamento de Veículos com Tempo de Viagem Dependente do Tempo, Ichoua *et al.* [2003] propõem um algoritmo que utiliza a meta-heurística Busca Tabu e técnicas de paralelismo. Ichoua *et al.* [2003], diferentemente da maioria dos pesquisadores do PRV, descrevem um ambiente em que os veículos não seguem tempos de viagens constantes, já que outros fatores, além da distância viajada, podem influenciar no tempo de viagem dos veículos. O fator utilizado para variar o tempo de viagem é a hora atual do dia que, segundo os autores, possui forte influência sobre a velocidade dos veículos. Por este motivo, o problema é nomeado com a extensão “dependente do tempo”. Um exemplo para demonstrar que o tempo

de viagem é dependente da hora do dia são os congestionamentos urbanos, que são mais frequentes em determinados intervalos do dia, ocorrendo geralmente na chegada e saída de funcionários de seus respectivos locais de trabalho.

Além de tratar o dinamismo apresentando novos pedidos de coleta durante o processo de atendimento, Alvarenga [2005] trata também o cancelamento de pedidos. Alvarenga [2005] trata o PDRV utilizando a bem conhecida base de testes de Solomon [1987] para o Problema de Roteamento de Veículos com Janela de Tempo. Desta forma, a comparação dos resultados obtidos no PDRV torna-se simples, pois o objetivo passa a ser a aproximação dos resultados do seu sistema aos resultados obtidos pelos sistemas que resolvem o PRVJT estático. Em seu trabalho, Alvarenga [2005] demonstra que métodos que aproveitam suas soluções já conhecidas, mesmo depois de receber uma notificação de alteração no ambiente, possuem maior qualidade se comparados aos chamados algoritmos de re-otimização, que se caracterizam por reiniciar o processo de otimização ao receberem notificações sobre alterações no ambiente. Outra contribuição de Alvarenga [2005] é a clara diferenciação dos problemas em tempo real e dos problemas *online*. Ele define que problemas em tempo real são aqueles que estão sujeitos a algum tipo de dinamicidade e, ao mesmo tempo, devem responder às novas requisições em tempo hábil. Já os problemas do tipo *online* são aqueles que precisam de respostas mais rápidas para as alterações no ambiente.

Tanto os problemas em tempo real quanto os problemas *online* quase eliminam a possibilidade de aplicação de métodos de re-otimização, pois fica impraticável reiniciar o processo de busca por soluções quando existe a necessidade de, rapidamente, alterar a alocação dos veículos em atividade. Neste contexto, se tornam muito mais atraentes os métodos que aproveitam suas soluções conhecidas antes da alteração no ambiente para adequá-las às novas demandas, embora esta adequação não seja tão trivial.

3.2 Requisições dinâmicas antes do início do dia de serviço

Alguns pesquisadores caracterizam o PDRV com a frota veicular ainda parada no depósito central. Simulam a situação de, por exemplo, o algoritmo de roteamento

estar calculando o escalonamento dos veículos antes do início do dia e, em um momento anterior ao da partida dos veículos, novos pedidos são requisitados ao sistema, alterando a requisição inicial e transformando o PRV estático inicial em uma versão do PDRV. A partir da(s) nova(s) requisição(ões) existe um novo problema estático que deve ser resolvido, utilizando ou não a solução do problema passado.

O trabalho de Alvarenga [2005] trata exatamente este cenário. Considera que seu horizonte de planejamento possui 60 minutos, e que novos pedidos chegam ao sistema de roteamento entre os minutos 45 e 55. Para avaliar seu algoritmo, Alvarenga [2005] considera dois graus diferenciados de dinamismo (a definição de grau de dinamismo é dada na Seção 4.3.1.1). No primeiro deles, 30% dos consumidores surgem entre os minutos 45 e 55. No segundo cenário, é considerado que 50% dos consumidores fazem o pedido no mesmo intervalo de tempo descrito.

O objetivo de Alvarenga [2005] é mostrar que seu método híbrido é capaz de responder com eficiência, mesmo recebendo as requisições dinâmicas. Comparando com os resultados do PRVJT estático, ele mostrou a eficiência do seu algoritmo de roteamento para este tipo de ambiente dinâmico.

3.3 Requisições dinâmicas durante o dia de serviço

A maioria das pesquisas sobre o PDRV e sobre o PRVE trata o dinamismo com a frota veicular em andamento. Ao longo desta seção são apresentadas diferentes abordagens sobre esta classe de problemas.

3.3.1 Subdividindo o tempo dia de serviço em períodos

Alguns trabalhos dividem o dia de serviço e tratam as partes do PDRV como várias instâncias de um PRV estático. Seguindo esta linha podemos citar, por exemplo, Bent e Van Hentenryck [2004], Chen e Xu [2006] e Hvattum *et al.* [2007].

Chen e Xu [2006] resolvem a demanda estática (antes do dia começar) através da formulação do Problema de Particionamento de Conjuntos (PPC), descrita na Seção 2.4.3 deste trabalho. No trabalho de Chen e Xu [2006], o PPC é resolvido utilizando o *software* de programação matemática CPLEX.

A abordagem de roteamento utilizada no PDRV é periódica. Durante a fase dinâmica do problema, em cada período, uma nova instância do PPC é resolvida.

Chen e Xu [2006] minimizam a distância total percorrida pela frota de veículos. Não consideram informações históricas para prever os consumidores futuros, tratando, assim, de uma versão exclusivamente dinâmica do problema. Eles consideram que toda a demanda deve ser atendida ao longo do dia de serviço do PDRV.

Para mostrar a eficiência do modelo proposto, Chen e Xu [2006] comparam sua proposta com dois métodos diferentes na estratégia de roteamento dos veículos:

- o primeiro deles é baseado no *Push-forward Insertion Heuristic* (PFIH). Neste modelo, sempre que uma requisição chega ao sistema de roteamento, o mesmo insere o consumidor de forma gulosa na solução existente. Mais detalhes a respeito do algoritmo guloso PFIH podem ser vistos na Seção 6.2.2.4;
- o segundo não considera um tempo limite para a execução do algoritmo exato que resolve o modelo de particionamento de conjuntos em cada período do dia. O CPLEX pode demorar o tempo que for necessário resolvendo o modelo de particionamento de conjuntos de forma ótima, não sendo interrompido ao longo de sua execução. Vale ressaltar que a resolução ótima é para um subconjunto R' de rotas, que não contempla todas as rotas possíveis. Mais detalhes sobre a aplicação da formulação do PPC no PRV são encontrados na Seção 2.4.3. Na prática, esta estratégia é irreal, pois o algoritmo de roteamento precisa fornecer aos veículos em trânsito as novas ordens antes que seja tarde demais. Chen e Xu [2006] argumentam que, assim, podem medir o *gap* entre a abordagem proposta por eles e um universo imaginário, que resolve com exatidão o PPC com um conjunto de rotas reduzido R' ao longo dos vários períodos do dia no PDRV.

Chen e Xu [2006] conseguiram mostrar, neste cenário dinâmico, que a abordagem proposta por eles é melhor que a inserção heurística de consumidores, amplamente utilizada na literatura (vide Bent e Van Hentenryck [2004] e Oliveira *et al.* [2007]). Outro ponto positivo do trabalho é a medida de qualidade entre a proposta dos autores para um ambiente simulado que não possui tempo limite de

processamento nos períodos. Ou seja, mesmo interrompendo o processamento de cada período no tempo predeterminado em sua versão periódica, a solução final apresenta resultados estatisticamente equivalentes, se comparada às situações em que a resolução do PPC não possui limite de tempo.

Esta subdivisão do dia de serviço é interessante para simplificar a implementação do sistema de otimização. O ponto negativo desta abordagem é o tempo que o sistema otimizador leva para responder às novas requisições dinâmicas. A melhor solução possível em um PDRV pode depender diretamente do tempo que o algoritmo de roteamento possui para realocar a frota veicular, em função do posicionamento dos consumidores. Se eles se locomovem com o passar do tempo, cada vez menos possibilidades no roteamento estão disponíveis.

3.3.2 Estratégias de espera para o PDRV

Alguns trabalhos que tratam as requisições dinâmicas durante o dia de serviço utilizam uma estratégia de espera veicular programada para maximizar a quantidade de clientes atendidos, além de reduzir ainda mais a distância total viajada no PDRV ou no PRVE.

Quando um veículo termina o processo de atendimento em algum cliente, ele pode:

- imediatamente se dirigir para o próximo consumidor (ou para o depósito, finalizando a rota); ou
- ficar parado nas proximidades do consumidor que acabou de atender, efetuando uma espera programada.

Obviamente, o veículo não pode esperar por muito tempo, pois suas entregas futuras podem ser comprometidas, já que os consumidores ainda não atendidos também possuem janelas de tempo rígidas que não podem ser violadas. Existe, portanto, um intervalo de tempo que o veículo pode esperar.

A Figura 3.1 apresenta um cenário em que o veículo não efetua esperas controladas, percorrendo seu trajeto sem interrupções, gastando tempo somente viajando entre um consumidor e outro e efetuando serviços. Nesta figura, são mostrados dois instantes de tempo distintos: o primeiro é quando o algoritmo de

roteamento é informado sobre a existência do consumidor c_5 ; e, no segundo, é quando o algoritmo de roteamento reage à nova demanda, informando o novo planejamento.

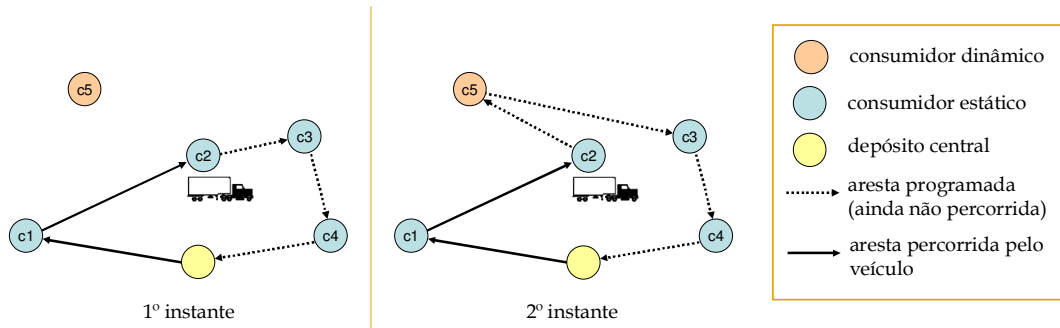


Figura 3.1 – Alocação veicular sem espera veicular programada

Como o veículo já completou duas arestas da viagem $\{(depósito, c_1); (c_1, c_2)\}$, o novo consumidor c_5 apenas pode ser inserido entre as possibilidades deixadas pelas arestas: $\{(c_2, c_3); (c_3, c_4); (c_4, depósito)\}$. Neste exemplo (Figura 3.1, 2º instante) o consumidor c_5 foi inserido entre os consumidores c_2 e c_3 , gerando o novo roteamento $\{(c_2, c_5); (c_5, c_3); (c_3, c_4); (c_4, depósito)\}$. Esta estratégia se chama dirigir primeiro (DP).

A Figura 3.2 apresenta a mesma demanda da Figura 3.1, mas com o veículo efetuando uma espera programada após o atendimento do consumidor c_1 . Desta forma, a inserção do novo consumidor c_5 pode ser feita sobre mais possibilidades, já que existe uma maior quantidade de arestas ainda não percorridas: $\{(c_1, c_2); (c_2, c_3); (c_3, c_4); (c_4, depósito)\}$, se comparada com a estratégia DP. Assim, o algoritmo do roteamento pode efetuar uma melhor alocação para o veículo, como mostrado na Figura 3.2 (2º instante), se comparado à alocação da Figura 3.1 (2º instante). Esta estratégia se chama esperar primeiro (EP).

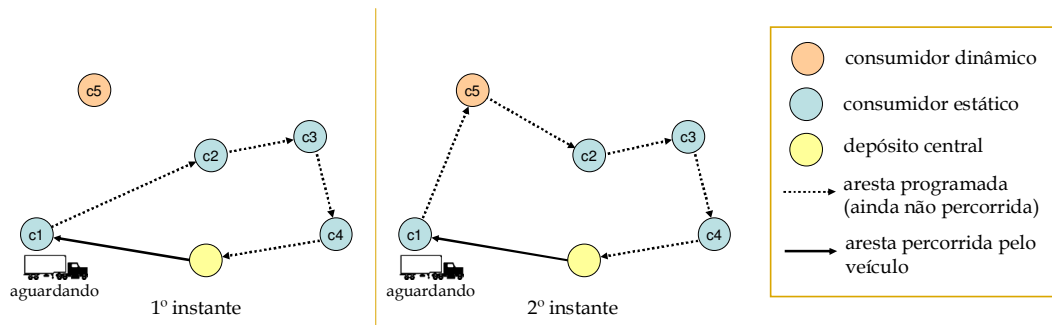


Figura 3.2 – Alocação veicular com espera veicular

A estratégia DP possui fácil gerenciamento, se comparada à estratégia EP, além de ser a condição adequada para o problema estático [Gendreau *et al.*, 2006]. A

estratégia EP mantém o veículo parado em um consumidor até o último instante possível [Gendreau *et al.*, 1999].

A estratégia EP (esperando o máximo possível) gera situações não desejadas em um sistema de roteamento no PDRV. Se um veículo efetua uma espera em um cliente, como no exemplo anterior, mais possibilidades são dadas a ele quando novos clientes surgem ao longo do tempo. Mas se a espera efetuada for máxima, o veículo não pode mais se atrasar, pois qualquer atraso implicará na violação de pelo menos uma janela de tempo dos seus futuros consumidores ainda não atendidos. A falta de possibilidade de atraso também impede que qualquer novo consumidor dinâmico seja inserido nesta rota que não possui folga temporal, perdida com a espera programada.

Apesar da literatura no roteamento dinâmico ser extensa, poucos trabalhos relatam a utilização de estratégias de espera veicular [Pureza e Laporte, 2008].

Segundo Pureza e Laporte [2008], poucos trabalhos analisam o impacto das estratégias de espera na qualidade de soluções. Um deles é o Mitrović-Minić e Laporte [2004], que compara diferentes estratégias de espera para o Problema Dinâmico de Roteamento de Veículos de Coleta e Entrega com Janela de Tempo (PDRVCEJT). Experimentos mostram que a estratégia EP tende a produzir rotas mais curtas quando comparada à estratégia DP. A espera faz com que ocorra um acúmulo de pedidos, oferecendo mais possibilidades para a otimização das rotas. Por outro lado, a EP necessita de uma maior quantidade de veículos do que DP. Como os veículos esperam muito nos estágios iniciais do dia de serviço, ao final eles não possuem folgas para incluir novos consumidores em suas rotas, precisando de mais veículos para o atendimento destas demandas. Mitrović-Minić e Laporte [2004] propõem uma estratégia híbrida de espera, na expectativa de mesclar os ganhos oferecidos por DP e EP. Caso a localidade que o veículo atende no instante atual seja próxima geograficamente da próxima localidade a ser atendida pela rota, então a estratégia DP é aplicada. Caso contrário, é aplicada a estratégia EP.

Outro trabalho que estuda a estratégia de espera é Branke *et al.* [2005]. Os autores têm como objetivo único maximizar a probabilidade de um consumidor dinâmico ser atendido, considerando que o depósito central não possui veículos extras para serem utilizados no caso de necessidade. Com os veículos em andamento, Branke *et al.* [2005] inserem apenas um consumidor dinâmico ao longo do dia para medir a capacidade que as diferentes estratégias de espera têm de atender o novo pedido sem descartá-lo. No experimento, o instante que o novo consumidor surge na

simulação é definido por uma distribuição uniforme. Neste caso, não é possível utilizar informações históricas, o que transformaria o PDRV em um PRVE, pois o aparecimento do consumidor é imprevisível (devido à total aleatoriedade da distribuição utilizada para o aparecimento do novo consumidor).

Para maximizar a probabilidade do novo consumidor ser atendido, Branke *et al.* [2005] avaliam diferentes estratégias:

- sem espera: idêntica a estratégia DP definida anteriormente, em que os veículos viajam imediatamente quando terminam os serviços em cada consumidor;
- espera no depósito: aguarda o máximo possível, mas apenas no depósito. Utiliza a estratégia EP no depósito e, depois, no atendimento de cada consumidor, utiliza a estratégia DP;
- distância máxima: aguarda o máximo possível no cliente mais distante do depósito central. A idéia é manter a frota de veículos espalhada geograficamente, cobrindo grande parte do mapa;
- espera média: esta estratégia é uma mescla da DP com a EP. O algoritmo de roteamento calcula qual é a folga de tempo daquela rota, e divide o tempo de espera igualmente para cada consumidor;
- espera baseada na distância: é uma estratégia EP adaptada, mas o veículo não espera até o máximo possível. Em cada consumidor, ele espera um tempo proporcional à distância para chegar até o próximo consumidor. Esta estratégia é semelhante à aplicada por Mitrović-Minić e Laporte [2004], em que DP é aplicada para consumidores próximos e EP é aplicado quando os consumidores são distantes geograficamente;
- algoritmo evolucionário: neste caso o espaço de buscas é o conjunto de todas as diferentes estratégias de espera. Um ponto interessante do algoritmo é a avaliação de cada cromossomo, já que o sistema não sabe quando o novo consumidor dinâmico vai requisitar o atendimento. Para avaliar os indivíduos, são sorteados 100 instantes distintos através de uma distribuição uniforme. Além disso, também é sorteada a posição geográfica de cada possível consumidor. Através deste método, os autores

tentam maximizar a probabilidade de atendimento do consumidor que ainda não surgiu.

Como previsto pelos autores, o trabalho mostrou que diferentes estratégias de espera veicular possuem diferentes probabilidades de atender um consumidor futuro. O experimento mostrou que a estratégia de esperar no depósito é a pior possível dentre as descritas. Efetuar a espera veicular no cliente mais distante do depósito é uma estratégia ligeiramente pior que simplesmente não aguardar (DP). Considerando o desempenho de maximizar a probabilidade do atendimento do novo consumidor, quatro estratégias se apresentaram melhor que DP e bem parecidas entre si: Espera Média, Espera Baseada na Distância, Espera Variável e a espera definida pelo Algoritmo Evolucionário.

As diferenças do trabalho de Branke *et al.* [2005] para o trabalho de Mitrović-Minić e Laporte [2004] são: (i) Mitrović-Minić e Laporte [2004] consideram o problema de coleta e entrega e Branke *et al.* [2005] consideram o PRV (coleta ou entrega); (ii) Mitrović-Minić e Laporte [2004] minimizam a distância total enquanto Branke *et al.* [2005] maximizam a probabilidade de um novo consumidor ser atendido pelos veículos já em andamento.

Ao longo deste trabalho, experimentos foram realizados para avaliar os impactos da espera veicular sobre o PDRV que objetiva a minimização da distância total viajada pelos veículos, diferentemente dos outros trabalhos citados nesta seção, que reduzem a quantidade de veículos utilizados. Os resultados foram positivos e publicados no trabalho Oliveira *et al.* [2008].

3.3.3 Estratégias de desvio para o PDRV

Como descrito no trabalho Ichoua *et al.* [2000], uma das formas de minimizar os gastos de uma frota de veículos é desviar veículos em movimento (que ainda não chegaram em seus destinos) para um novo consumidor que eventualmente surgiu dinamicamente nas proximidades do veículo. Isso se torna possível, na prática, devido aos equipamentos de GPS e aos *software* de navegação, que podem informar ao motorista do veículo a nova rota recalculada em função de eventos dinâmicos.

Segundo Ichoua *et al.* [2000], a motivação para a aplicação do desvio de rotas já iniciadas e não finalizadas vem de aplicações nos serviços de correios que precisam efetuar coletas/entregas em longas distâncias. Em alguns casos, enquanto o veículo

dos correios se dirige para a próxima localidade, uma nova demanda surge nas proximidades do caminho até o próximo consumidor.

Os objetivos no trabalho Ichoua *et al.* [2000] são: (i) minimizar a distância total viajada pela frota; e (ii) minimizar o atraso para o atendimento dos clientes (é considerado o problema sem janela de tempo rígida). Para tratar o desvio, classificam as arestas do problema em três categorias:

- arestas que o veículo já completou;
- aresta que o veículo está atualmente percorrendo;
- arestas que estão programadas para o veículo percorrer no futuro.

A maioria dos trabalhos que tratam o PDRV considera que apenas as arestas de movimentos programados podem ser alteradas, sem considerar que podem interromper o movimento atual do veículo para atender novas demandas. O trabalho de Lorini *et al.* [2011] foi o único encontrado na literatura que considera a possibilidade da mudança em tempo real das arestas que estão sendo percorridas pelos veículos da frota. Outros trabalhos que consideram estratégias de desvio são Regan [1997] e Gendreau *et al.* [1999].

3.4 Informações históricas sobre requisições de clientes

Se no PDRV estão disponíveis informações históricas sobre algum conjunto de eventos dinâmicos, então o problema passa a ser o PRV Estocástico (PRVE). Como já citado no Capítulo 2, o PRVE é uma forma de tratar alguns tipos de dinâmismos de forma antecipada [Alvarenga, 2005].

A utilização de informações históricas para a resolução do PRVE é investigada em Hvattum *et al.* [2004]. A informação histórica considerada é relacionada à ordem no aparecimento de novos consumidores. Isso pode aumentar consideravelmente a qualidade da solução final do sistema de roteamento que trata o problema, pois situações podem ser tratadas antes mesmo de ocorrerem durante o período de serviço. A previsão destas situações pode consideravelmente amenizar o impacto da dinamicidade do PRVE.

Bent e Van Hentenryck [2004] apresentam uma abordagem nomeada de Múltiplos Cenários para resolver o PRVE. Seus resultados indicam consideráveis melhorias para a abordagem que considera as informações históricas do problema, se comparada à versão puramente dinâmica. A base de testes de Solomon [1987], que apresenta informações para o PRVJT, foi adaptada por Bent e Van Hentenryck [2004] com as informações relacionadas às requisições dinâmicas necessárias em seu trabalho. Eles estudam o problema com dois graus de dinamismo distintos: 30% e 80%. O horizonte de tempo é subdividido em 4 períodos (estratégia idêntica a apresentada na Seção 3.3.1). O período t_0 representa a demanda conhecida antes dos veículos saírem do depósito. Os períodos t_1 , t_2 e t_3 podem ser visualizadas como manhã, início da tarde e tarde, respectivamente. As informações estocásticas em Bent e Van Hentenryck [2004] são em relação ao instante que cada consumidor vai requisitar o atendimento. Os autores consideram a categoria da instância de Solomon [1987] para determinar diferentes probabilidades. Mais detalhes sobre as categorias das instâncias são encontrados na Seção 4.3. Este trabalho também mostra que a manutenção de múltiplos planejamentos é fundamental na obtenção de soluções de alta qualidade no PDRV, mesmo quando não existe informação estocástica disponível.

Ichoua *et al.* [2006] apresentam uma estratégia que utiliza informações históricas para gerenciar o tempo de espera nos consumidores. O mapa da cidade é dividido em quadrantes e apenas informações históricas dos consumidores em quadrantes próximos ao veículo são consideradas.

No trabalho de Ichoua *et al.* [2006] a área é particionada em J zonas geográficas (quadrantes). Os consumidores são agrupados de acordo com suas janelas de tempo. Por exemplo, todos os consumidores que possuem janela de tempo de 13:00 às 13:30 estão em um mesmo grupo w ($W = \{w_1, w_2, w_3, \dots, w_M\}$). São definidos também tipos de consumidores, em que os consumidores do mesmo tipo estão na mesma zona geográfica e também estão no mesmo grupo, ou seja, possuem a mesma janela de tempo. Assim como a maioria dos trabalhos que tratam o PDRV, Ichoua *et al.* [2006] não consideram a possibilidade de desvio, como os exemplos da Seção 3.3.3. Diferentemente da maioria dos trabalhos que tratam alguma variação do PRVJT, os veículos não são permitidos chegar em consumidores antes da abertura de sua janela de tempo. Este fator aumenta o risco de ser afetado por eventuais congestionamentos no mundo real, por exemplo. Por outro lado, este sistema fornece uma alternativa para atender eventuais novos consumidores com mais facilidade. Cada quadrante é associado a um processo de Poisson. Este processo descreve a taxa de aparecimento

de novos consumidores ao longo do tempo. Desta forma, Ichoua *et al.* [2006] verificam se, no quadrante atual, e nos quadrantes vizinhos, a probabilidade de aparecer novos consumidores é alta no horário corrente do dia. Se for, eles optam por efetuar uma espera veicular no consumidor que acabou de ser atendido.

3.5 Sumário do capítulo

Neste capítulo foi apresentada a revisão bibliográfica dos trabalhos relacionados ao tema central deste doutorado.

Inicialmente, foi descrita a estratégia do tratamento de requisições dinâmicas, enquanto os veículos ainda estão parados no depósito central (antes do início do dia de serviço). Os trabalhos que consideram requisições dinâmicas durante o dia de serviço são apresentados, sendo são destacados três grandes grupos, que podem se relacionar entre si. O primeiro deles é a forma de visualizar o PDRV como uma sequência de PRV Estáticos. Até que o ambiente ou a demanda não sofra alterações, um PRV Estático deve ser resolvido ao longo do dia de serviço. Isso facilita a forma de tratar o PDRV, mas obriga que os pesquisadores desenvolvam eficientes algoritmos para as versões estáticas do problema.

Posteriormente, este capítulo apresentou as estratégias de espera veicular programada, que na última década foram aplicadas com sucesso no PDRV.

A estratégia de desvio empregada no PDRV foi descrita, seguida de trabalhos que utilizam informações históricas acerca do PRV. Foram apresentados alguns dos tipos mais usados de informações que podem auxiliar na resolução do PRVE.

Este trabalho não analisa as estratégias de desvio, espera veicular e previsões através de informações históricas. A análise se concentra na variação do tipo de roteamento (*online* e duas variações periódicas). Assim é possível realizar uma análise efetiva, sem a interferência de ruídos oriundos de outras técnicas para melhoria do PDRV. Em aplicações reais, torna-se interessante mesclar as conclusões destes diferentes autores para obtenção de ganhos sobre a resolução do PDRV.

Capítulo 4

Metodologia

4.1 Definições iniciais

A pesquisa desenvolvida neste trabalho foi de caráter experimental. Foram efetuadas simulações de demandas dinâmicas, juntamente com a aplicação dos algoritmos de roteamento para a avaliação da eficácia dos modelos em avaliação (*online* e duas variações periódicas). Tal investigação, empírica, tem como principal finalidade comparar as abordagens *online* e periódica, que diz respeito às relações de causa e efeito dos algoritmos de roteamento testados sobre o PDRV (mais detalhes sobre a hipótese podem ser vistos adiante na Seção 4.2).

Este trabalho considera o PDRV com novas requisições dinâmicas ao longo do dia do serviço. Em outras palavras, com os veículos já em andamento, novos clientes podem solicitar atendimento. Para isolar a comparação no experimento entre os algoritmos de roteamento avaliados, não são utilizadas informações estocásticas sobre os pedidos que podem ocorrer dinamicamente ao longo do dia. Esta utilização é um interessante objeto de estudo futuro. Em tese, o algoritmo de roteamento periódico tende a aumentar sua qualidade com a utilização de informações históricas sobre o comportamento dos clientes dentro de uma determinada região. Estas informações, em conjunto com estratégias de espera, podem melhorar a qualidade dos algoritmos de roteamento.

Para realizar a investigação empírica, este trabalho propõe um simulador de ambiente. Inicialmente, este simulador emula a demanda inicial do dia de serviço, disponibilizando um problema estático para ser resolvido pelo algoritmo. Assim que o algoritmo possui uma solução para a demanda inicial, é informado o planejamento para a frota veicular. Com hora marcada, os veículos saem do depósito, seguindo o planejamento inicial criado pelo algoritmo de otimização. Durante o dia de serviço no PDRV, já com os veículos em andamento, novas demandas dinâmicas informadas pelo simulador podem surgir na área de cobertura, e os algoritmos de roteamento devem otimizar suas rotas planejadas com a nova situação, e, posteriormente,

informar para cada veículo o seu novo planejamento. Mais detalhes sobre o processo de simulação do PDRV podem ser encontrados no Capítulo 5. Dado que o algoritmo de roteamento recebeu a informação da nova demanda, este pode reagir basicamente de duas formas: (i) de forma rápida; ou (ii) em tempos pré-programados, acumulando assim requisições dinâmicas e resolvendo diversos problemas estáticos ao longo do dia de serviço. É neste contexto, como descrito na Seção 1.2, que esta tese investiga a seguinte pergunta:

Durante a resolução do PDRV, seria mais vantajoso reagir ao ambiente rapidamente quando novos pedidos são requisitados, ao invés de esperar instantes pré-programados com tempos mais longos para processamento?

Esta pergunta foi levantada nesta tese porque, com frequência, outros trabalhos na literatura utilizam a abordagem periódica para resolver o PDRV.

Do ponto de vista da otimização para replanejamento das rotas, o acúmulo de pedidos pode ser interessante, pois o algoritmo de roteamento tem a possibilidade de encontrar uma solução que considera uma quantidade maior de variáveis de decisão, que muitas vezes se influenciam. Além do acúmulo de pedidos, o roteamento periódico pode usar ao seu favor um tempo maior para o processamento das requisições.

Apesar das vantagens da abordagem periódica, este trabalho propõe o roteamento *online* para o tratamento do PDRV. Para investigar a pergunta apresentada nesta seção, uma hipótese foi estabelecida e é discutida na Seção 4.2. Além do tratamento *online*, esta tese também investiga uma variação da abordagem periódica, que realiza o acúmulo de pedidos, mas não utiliza todo o tempo de processamento disponível para informar o novo planejamento aos veículos. Esta última abordagem de roteamento utiliza o mesmo tempo de processamento do algoritmo *online*, mas não inicia o algoritmo logo que o pedido é realizado. O início do algoritmo acontece periodicamente em instantes pré-programados.

Assim, os algoritmos de roteamento avaliados nesta tese são:

- *online*, que dispara o algoritmo de otimização assim que um pedido é realizado, e possui curto tempo de processamento;

- periódico 1, comumente encontrado na literatura, que acumula pedidos e executa o procedimento de otimização até o início do próximo algoritmo periódico;
- periódico 2, que acumula pedidos, mas executa o procedimento de otimização rapidamente, com o mesmo tempo de processamento do roteamento *online*.

O restante do capítulo apresenta os componentes metodológicos utilizados na tentativa de responder a pergunta principal da tese. Na Seção 4.3, é apresentada a base de testes utilizada pelo simulador. O experimento é discutido na Seção 4.4. As considerações finais sobre este capítulo são apresentadas na Seção 4.5.

4.2 Hipótese da tese

O objetivo central desta tese é demonstrar que uma abordagem mais rápida para o roteamento no PDRV pode ser mais eficaz, se comparado aos tratamentos que acumulam os pedidos, respondendo-os em intervalos de tempo iguais e bem definidos.

Assim, a hipótese *H* desta tese pode ser definida como:

H: Utilizando um competitivo algoritmo para alocação de veículos, o roteamento online para atendimento de novos pedidos no PDRV pode ser estatisticamente mais eficaz se comparado ao roteamento periódico.

O competitivo algoritmo utilizado para avaliação da hipótese é apresentado na Seção 6.2.

4.3 Base de testes

Para comparar os tipos de roteamento em análise, faz-se necessário:

- ser capaz de simular ambientes;

- considerar diversos tipos de ambiente na simulação.

Para a simulação de ambientes, este trabalho criou um simulador evento-discreto como base de toda a experimentação. O simulador faz o papel do mundo real, onde toda a emulação é executada. Mais detalhes sobre o processo de simulação podem ser vistos no Capítulo 5.

Para tratar diferentes tipos de ambiente, esta seção fornece informações sobre a extensão da base de testes de Solomon [1987], que servem de entrada para o simulador desenvolvido. A base de Solomon [1987] é amplamente reconhecida pela literatura, sendo a principal base de comparação entre trabalhos que pesquisam variações do PRVJT.

Na base de Solomon [1987], existem 56 instâncias que são divididas em seis grupos de instâncias: R1, R2, C1, C2, RC1 e RC2. As instâncias dos grupos R1 e R2 apresentam consumidores com as coordenadas euclidianas aleatórias (consumidores distribuídos aleatoriamente no espaço bidimensional), como mostrado na Figura 4.1.

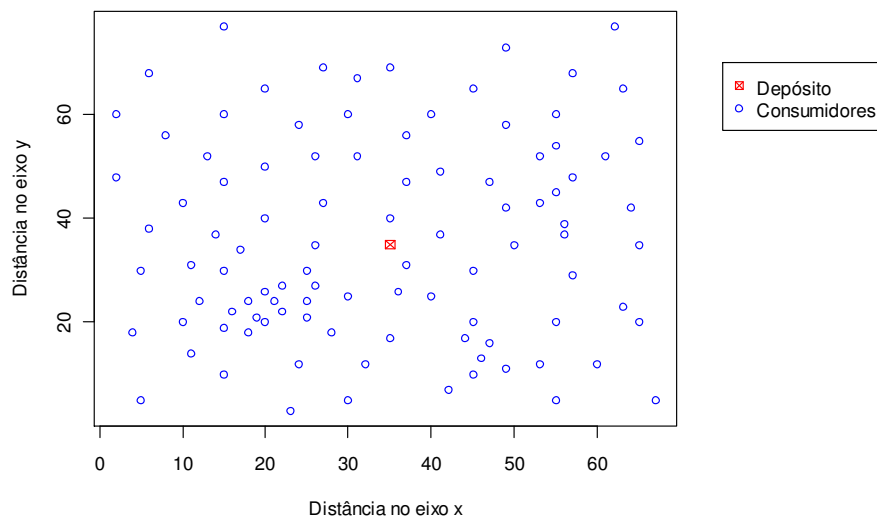


Figura 4.1 - Exemplo da disposição dos consumidores das classes R1 e R2

Já as instâncias dos grupos C1 e C2 apresentam os consumidores em conglomerados (Figura 4.2).

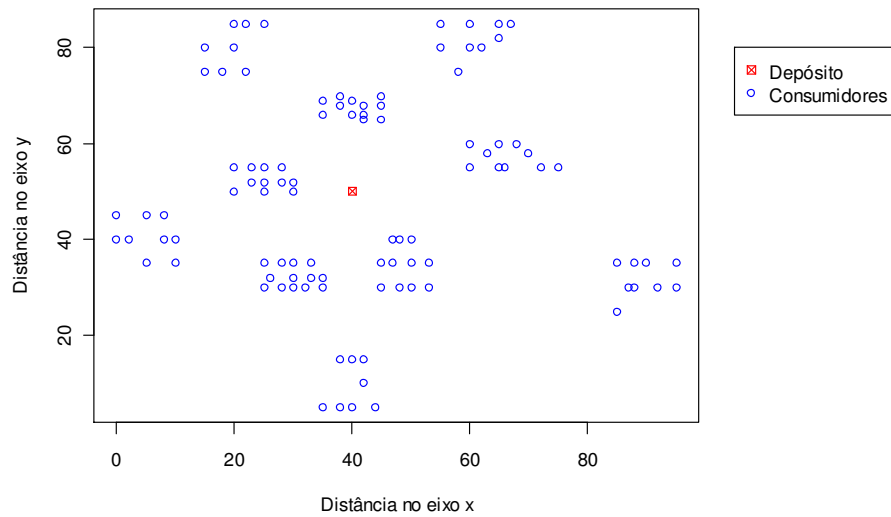


Figura 4.2 - Exemplo da disposição dos consumidores das classes C1 e C2

As instâncias dos grupos RC1 e RC2 apresentam um misto das duas primeiras características (esparso e aglomerado), como mostrado na Figura 4.3.

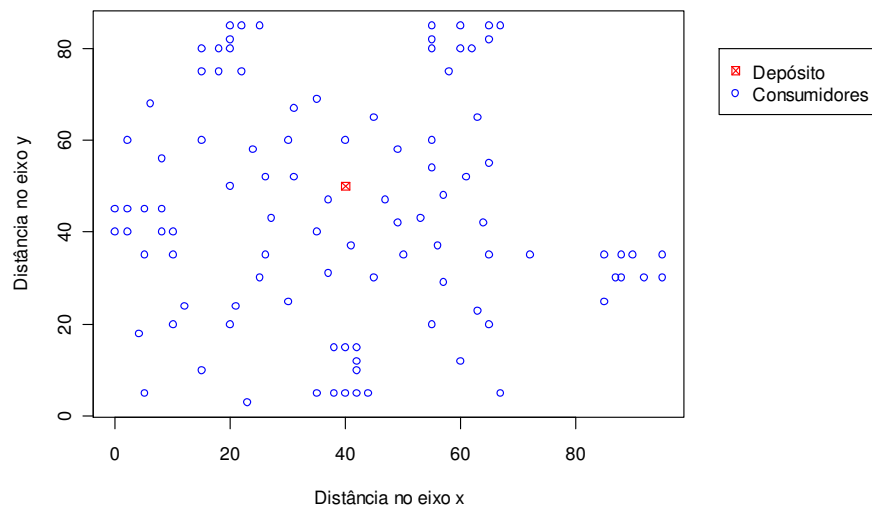


Figura 4.3 - Exemplo da disposição dos consumidores da classe RC1 e RC2

Uma característica entre os grupos R1, C1 e RC1 é que suas instâncias possibilitam que poucos consumidores sejam atendidos por um veículo, necessitando de uma frota maior para atender toda a demanda. Tal fato ocorre, por exemplo, porque cada consumidor possui uma alta demanda de carga e/ou uma curta janela de tempo. Já os tipos R2, C2 e RC2 apresentam a necessidade de poucos veículos disponíveis para o atendimento de toda a demanda, assim apresentando uma grande quantidade de consumidores em cada rota da solução. Tal situação pode ser

explicada, por exemplo, porque os consumidores demandam pouca carga e/ou largas janelas de tempo, facilitando a alocação de um veículo para vários consumidores.

A base de testes original de Solomon [1987] para o PRVJT pode ser encontrada no endereço:

http://neo.lcc.uma.es/radi-aeb/WebVRP/data/instances/solomon/solomon_100.zip

4.3.1 Extensão da base de Solomon [1987]

Para ser capaz de simular um dia de serviço do PDRV, faz-se necessária a inclusão de informações na base de testes de Solomon [1987]. Para estender a base de testes original e para estudar diferentes tipos de cenários, foram selecionadas aleatoriamente seis instâncias de Solomon, uma de cada grupo da base original: C101, C203, R106, R202, RC104, RC208. Para a adequação destas instâncias estáticas ao PDRV, alguns fatores foram considerados na extensão, tendo como base a avaliação da hipótese central da tese em diferentes cenários. Estes fatores, detalhados nas próximas seções, são:

- Grau de dinamicidade de pedidos;
- Horário limite para requisições dinâmicas;
- Distribuição dos pedidos dinâmicos em função do tempo.

4.3.1.1 Graus de dinamismo

O grau de dinamismo descreve o percentual de pedidos dinâmicos, se comparado a quantidade total de pedidos do PDRV. Outros trabalhos também utilizam esta característica de ambiente para avaliar a qualidade de seus algoritmos dinâmicos como, por exemplo, Bent e Van Hentenryck [2004] e Alvarenga [2005].

Dois graus de dinamismo foram utilizados neste trabalho: 25% e 75% dos consumidores. Isso significa que, no primeiro caso, 75% dos consumidores são conhecidos antes da saída dos veículos do depósito e 25% dos veículos são conhecidos entre 8 e 18 horas, ou seja, dinamicamente. No segundo caso, 25% dos consumidores são conhecidos ainda na versão estática do problema, e o restante é conhecido na fase dinâmica.

4.3.1.2 Horário limite para requisições dinâmicas

Como o roteamento periódico foi avaliado neste trabalho, foram utilizados horários limite para as requisições dinâmicas. Se tal componente não fosse considerada nas instâncias estendidas, alguns pedidos dinâmicos poderiam ser atendidos pelo algoritmo de roteamento *online*, e não ser atendidos pelo algoritmo de roteamento periódico, devido à natureza de seu funcionamento. Para comparar os três tipos de roteamento, sem beneficiar o roteamento *online*, este trabalho aplicou o horário limite para as requisições dinâmicas.

Nesta base de testes estendida, não existem pedidos próximos ao término do dia de serviço, de maneira que seja impossível atender todas as requisições, independente do algoritmo de roteamento utilizado. Por exemplo, se é utilizado o roteamento periódico que divide o dia em 10 partes, o horário limite para o aparecimento de novos consumidores é 16 horas. Este caso é exemplificado na Figura 4.4.

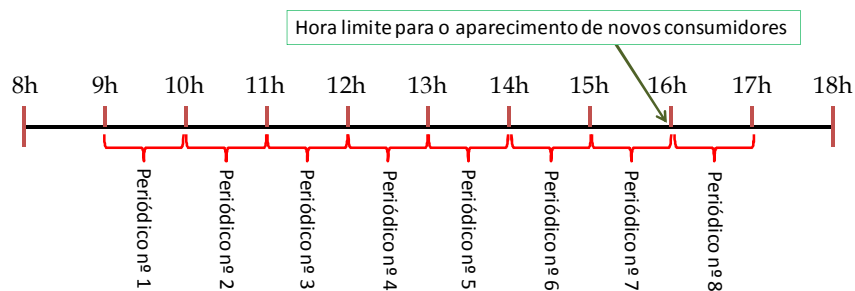


Figura 4.4 - Linha do tempo para algoritmos periódicos de 1 em 1 hora

No caso em que existem 10 períodos (com 8 execuções de algoritmos ao longo do dia), os consumidores dinâmicos podem surgir entre 8 e 16 horas. Qualquer consumidor que surja após este horário limite poderá não ser atendido. Um segundo caso é também considerado nesta base estendida (com hora limite de 17 horas para requisições dinâmicas). Assim, é permitida a execução de algoritmos periódicos que dividem o dia em 20 partes iguais, tendo 30 minutos para a execução cada um dos 18 algoritmos dinâmicos. A Figura 4.5 apresenta este caso.

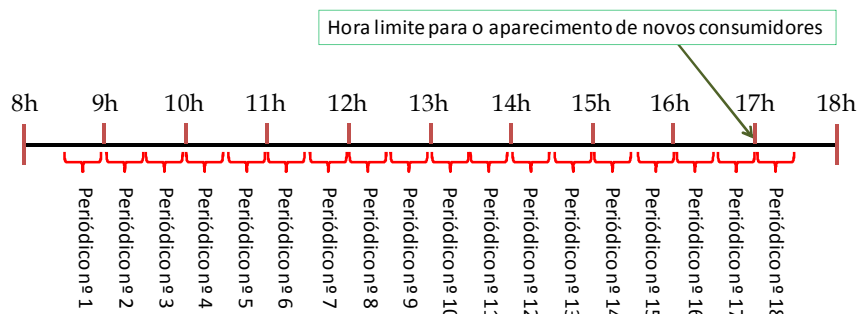


Figura 4.5 – Linha do tempo para algoritmos periódicos de 30 em 30 minutos

4.3.1.3 Distribuição dos pedidos dinâmicos no tempo

Com o objetivo de estudar o comportamento dos algoritmos de roteamento em análise, também é considerada a distribuição dos pedidos no tempo. Para isso, foram utilizadas duas distribuições bem conhecidas:

- Uniforme;
- Normal (gaussiana).

Na distribuição uniforme, os consumidores foram espalhados aleatoriamente entre 8 horas da manhã e o horário limite para aparecimento de consumidores dinâmicos (16 ou 17 horas, dependendo da instância). Na distribuição normal, foi utilizada média de 2 horas após o início do atendimento e desvio padrão de 30 minutos. Se o horário sorteado para o consumidor dinâmico requisitar o serviço for anterior às 8 horas ou posterior ao tempo limite de requisições dinâmicas, um novo horário é sorteado.

O horário do surgimento da requisição dinâmica também obedece às restrições de janela de tempo. Se o horário para requisição dinâmica de um consumidor foi sorteado para um instante posterior ao término de sua janela de tempo na base original, na base estendida a janela de tempo é deslocada, mantendo sua duração original, para instantes posteriores na linha do tempo, de maneira a permitir que aquele consumidor seja visitado durante a simulação do dia de serviço. Esta decisão de projeto foi tomada para evitar a comparação de resultados que apresentam o atendimento de todos os consumidores com resultados que não apresentam. Isso complicaria a comparação da métrica da distância total percorrida pelos veículos das soluções. Em aplicações reais, caso surjam consumidores após determinado horário, uma alternativa seria armazená-los para o próximo dia útil. Também foram

consideradas informações como tempo de viagem e tempo de serviço, para que não exista instância em que não seja possível atender a todas as requisições feitas dinamicamente. Ou seja, em todas as instâncias da base de testes estendida é possível atender todos os consumidores. Em um sistema de roteamento, as requisições feitas ao longo do dia, que não podem ser atendidas por restrição de tempo, são armazenadas para o próximo dia de serviço, se o cliente permitir. Esta possibilidade não é considerada nesta extensão, com o intuito de fornecer uma base sólida para a avaliação da hipótese central da tese. Assim, tanto o algoritmo de roteamento *online*, como o periódico sempre encontram soluções viáveis que atendem a todos os consumidores.

4.3.1.4 Acesso à base estendida

As 48 instâncias estendidas geradas por este trabalho podem ser acessadas no endereço:

<http://bcc.unifal-mg.edu.br/~humberto/doutorado/instanciasEstendidas/>

4.4 Experimento

4.4.1 Teste de hipótese

Como descrito na Seção 4.2, a hipótese central da tese H afirma que o algoritmo de roteamento *online* pode ser estatisticamente mais eficaz que as abordagens periódicas para o PDRV. Neste contexto, primeiramente foi utilizada a Análise de Variância (ANAVA) para tentar refutar a hipótese da tese, verificando se os tratamentos são equivalentes do ponto de vista estatístico. A ANAVA testa a hipótese H_0 da Equação 4.1. Esta afirma que todos os tratamentos (algoritmos de roteamento) produzem médias populacionais estatisticamente equivalentes [Aquino, 1991; Lima, 2000]. A Equação 4.1 apresenta as duas hipóteses da ANAVA:

$$\begin{cases} H_0 : \mu_{online} = \mu_{\text{periódico } 1} = \mu_{\text{periódico } 2} \\ H_1 : \mu_{online} \neq \mu_{\text{periódico } 1} \neq \mu_{\text{periódico } 2} \end{cases} \quad (4.1)$$

Assim, seria possível refutar a hipótese da tese H caso H_0 fosse confirmada. Caso contrário, a hipótese H ainda não estaria estatisticamente corroborada, pois um

dos tratamentos periódicos pode apresentar média inferior, sendo assim mais adequado ao PDRV. Isso também refutaria a hipótese H_0 . Caso a hipótese H_1 seja considerada estatisticamente mais relevante que H_0 na ANAVA, as médias produzidas pelos algoritmos de roteamento devem ser comparadas diretamente, com o objetivo de se concluir sobre a hipótese central da tese H . Os resultados da ANAVA e da comparação direta entre as médias são apresentados na Seção 7.3.3.

4.4.2 Delineamento do experimento

É comum encontrar na literatura de algoritmos de otimização e inteligência artificial a comparação entre médias de diferentes métodos através do teste t . Este poderia ter sido utilizado neste trabalho, mas seriam necessários 48 testes distintos, já que são 48 cenários (instâncias) aplicados na simulação do PDRV. Para contornar esta questão, foi realizada uma ANAVA sobre o Delineamento em Blocos Casualizados (DBC). O DBC é adequado quando há a necessidade de estudar o impacto de uma variável (tipo de roteamento) sobre o ambiente, eliminando o efeito causado pelos blocos (instâncias), sendo possível analisar de forma geral os tratamentos (tipos de roteamento) [Pimentel-Gomes, 2009].

O DBC foi proposto pela primeira vez na experimentação agrícola [Lima, 2000]. O objetivo era criar um projeto experimental capaz de isolar os efeitos observados em diferentes tipos de solo (blocos) para analisar diferentes fertilizantes (tratamentos). Neste trabalho, observa-se algo semelhante, pois é requerida uma conclusão geral sobre o tipo de roteamento, independente do cenário em que ele se apresenta.

No DBC é fundamental existir homogeneidade no mesmo bloco, o que é garantido pela repetição das mesmas situações diárias do PDRV através do simulador implementado. Além da uniformidade dentro do mesmo bloco, o DBC exige heterogeneidade entre diferentes blocos, o que é assegurado pelas 48 instâncias distintas analisadas. Todos os tipos de roteamento foram aplicados para todos os 48 blocos distintos. Assim, foi possível afirmar se existe diferença estatisticamente significativa entre os algoritmos de roteamento *online*, periódico 1 e periódico 2, para os diversos contextos, sendo possível concluir sobre a hipótese desta tese. Para realizar o teste de hipótese sobre os resultados apresentados dos diferentes algoritmos de roteamento, foi utilizado o grau de 99% de confiança estatística.

4.4.3 ANAVA sobre o DBC

O modelo da ANAVA sobre o DBC é dado pela Equação 4.2:

$$y_{ijk} = \mu + \alpha_i + \beta_j + \varepsilon_{ijk} \quad (4.2)$$

Considerado cada tratamento i , cada o bloco j e a repetição k , uma observação y_{ijk} pode ser descrita como a soma da média geral μ , com o efeito do tratamento α_i , com o efeito do bloco β_j , e com um erro experimental ε_{ijk} [Pimentel-Gomes, 2009].

A execução da ANAVA sobre o DBC constrói o modelo, isolando o efeito causado pelos blocos, podendo assim identificar se o impacto de cada tratamento é estatisticamente equivalente aos demais. Para inferir sobre suas hipóteses, a ANAVA utiliza a distribuição F , sendo esta ilustrada na Figura 4.6.

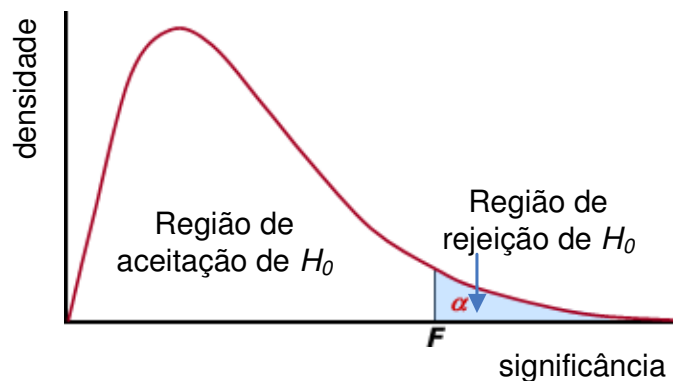


Figura 4.6 - Teste da ANAVA sobre a distribuição F

A hipótese H_0 da ANAVA é aceita ou refutada de acordo com seu valor p calculado após a análise. Neste trabalho foi utilizado 99% de confiança para o teste de hipótese da ANAVA. Assim, a área da região para aceitação de H_0 , que diz que os algoritmos de roteamento são iguais, possui 99% de toda a área da distribuição. Para a ANAVA refutar H_0 e considerar H_1 , o valor p do método deve apresentar valores inferiores a 0,01. Mais detalhes sobre o DBC e sobre a análise de variância podem ser encontrados em Pimentel-Gomes [2009].

4.4.4 Considerações técnicas

Para todas as 48 instâncias estendidas, os três tipos de roteamento em análise foram submetidos a 10 execuções sobre o simulador evento-discreto do PDRV.

O experimento foi realizado no Laboratório de Inteligência Computacional (LInC) da Universidade Federal de Alfenas (UNIFAL-MG). Foram utilizados oito computadores idênticos com processador Core2Quad e 8 GB de memória RAM. Para a realização da análise com o DBC, foi utilizado o pacote *ExpDes* [Cavalcanti ,2010], descrito para o *software* estatístico R.

4.5 Considerações Finais

Este capítulo descreveu o método científico utilizado durante a execução deste trabalho, com o intuito de gerar dados conclusivos para corroborar ou não a hipótese levantada, que avalia três tipos de roteamento sobre o PDRV.

Em complemento a este capítulo, o Capítulo 5 apresenta o simulador por este trabalho descrito, que serve de base para todo o experimento da tese. O Capítulo 6 demonstra o algoritmo utilizado durante os processos de roteamento. O Capítulo 7 apresenta os resultados sobre a resolução do PRVJT estático e também os resultados na otimização do PDRV. As conclusões sobre a hipótese da tese, com base no experimento desenvolvido, são vistas no Capítulo 8.

Capítulo 5

Simulação evento-discreta para o PDRV

5.1 Considerações iniciais

“Um sistema pode ser entendido como uma parte da realidade. A modelagem de sistemas visa à criação de modelos, que procuram definir os componentes de um dado sistema, bem como seus relacionamentos e restrições, com o objetivo de permitir a visualização de sua estrutura, e facilitar o entendimento do seu comportamento, o que não seria simples pela observação do sistema real, infinitamente mais complexo” [Souza, 2007].

Para estudar, compreender e avaliar a qualidade dos algoritmos de roteamento da hipótese da tese sobre o PDRV, o ambiente e seus componentes principais foram emulados através de um processo de Simulação Evento-Discreta (SED).

A Seção 5.2 apresenta considerações gerais sobre a SED, descrevendo seu vocabulário básico. A Seção 5.3 detalha o processo de SED implementado neste trabalho para o PDRV. As considerações finais sobre o simulador implementado neste trabalho são apresentadas na Seção 5.4.

5.2 Conceitos gerais sobre simulação evento-discreta

A SED pode ser utilizada quando eventos estão relacionados ao tempo no qual alguma atividade importante inicia ou termina. Para tais problemas, não é eficiente avançar o tempo da simulação em pequenos passos. Pode-se avançar o tempo para o próximo evento de interesse e relevante para o ambiente. Assim, na SED, o avanço do tempo não necessariamente é uniforme [Silva, 2001]. Em geral, eventos podem ocorrer em qualquer tempo.

Dentro da SED os conceitos de estado e tempo são fundamentais. Nance [1981] identifica os seguintes conceitos primitivos da SED:

- um **instante** é um valor de tempo do sistema, em que pelo menos um atributo de um objeto pode ser alterado na simulação;
- um **intervalo** é a duração entre dois instantes sucessivos;
- um *span* é a sucessão contígua de um ou mais intervalos;
- o **estado de um objeto** é a enumeração de todos os valores de atributos do objeto em um instante particular.

Em outro nível de abstração, são definidos conceitos relacionados diretamente com a simulação:

- uma **atividade** é um estado de um objeto sobre um intervalo;
- um **evento** é a alteração no estado do objeto;
- uma **atividade objeto** é o estado do objeto entre dois eventos, descrevendo sucessivas alterações de estado para aquele objeto;
- um **processo** é a sucessão de estados de um objeto sobre um *span*.

5.3 Processo de simulação para o PDRV

O mecanismo geral do simulador implementado neste trabalho pode ser visto na Figura 5.1. O simulador possui uma variável que ordena seus eventos em função do tempo (uma fila de prioridades). Possui maior prioridade para sair da fila, e ser processado, aquele evento que possui o menor instante.

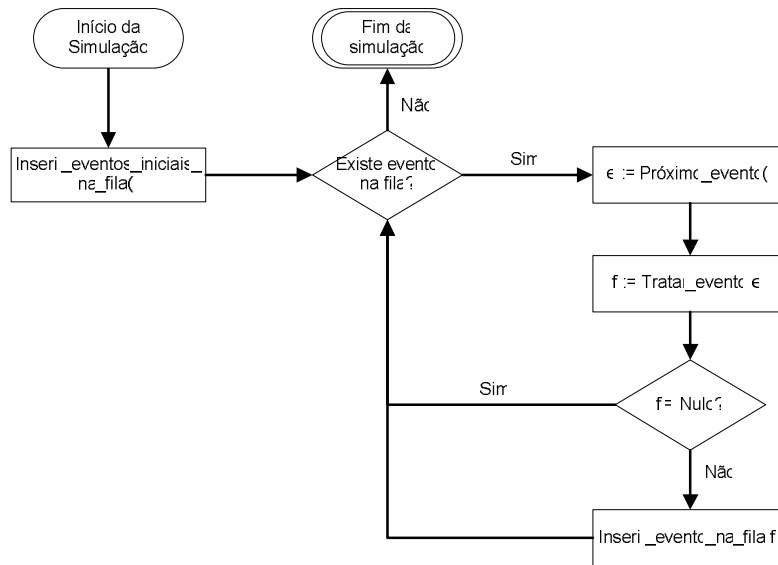


Figura 5.1 – Fluxograma simplificado do Simulador Evento Discreto implementado para o PDRV

Ao longo do processo de simulação, novos eventos podem ser inseridos na fila, que mantém seus elementos devidamente ordenados. Sempre, o próximo evento a ser tratado no simulador, gerando uma alteração no estado do objeto da simulação, é o próximo evento na linha do tempo. Assim, o simulador pode efetuar “saltos” no tempo. Este mecanismo permite considerar o avanço do tempo em uma velocidade não constante, mas nunca permitindo o processamento de um evento antes de um evento antecessor, impossibilitando a geração de inconsistências no estado do objeto da simulação.

A função Tratar Evento da Figura 5.1 representa a atividade objeto, que é capaz de alterar o estado do objeto da simulação. Eventos tratados pelo simulador podem gerar novos eventos para o ambiente, por exemplo:

- o pedido de atendimento de um consumidor dinâmico gera o início da execução de um algoritmo de roteamento, quando utilizado o roteamento *online*;
- a partida de um veículo de sua posição gera a chegada deste mesmo veículo em outro local, em um instante posterior na linha do tempo;

Nem todos os eventos tratados geram novos eventos. Detalhes sobre os tipos de eventos criados para emular o ambiente do PDRV e o processamento dos mesmos podem ser vistos na Seção 5.3.1.

As instâncias estendidas para o PDRV se relacionam diretamente com o simulador implementado. As informações sobre o instante de requisição para atendimento de cada consumidor é utilizada na geração de novos eventos, que indicam os pedidos de atendimento. Inicialmente, são inseridos na fila os pedidos de atendimento dos consumidores estáticos (aqueles que já requisitaram serviço antes da saída dos veículos do depósito central), com instantes de processamento iguais a zero, e os pedidos de atendimento dinâmicos (aqueles que requisitarão serviço ao longo do dia), indicando seus respectivos tempos de aparecimento. Desta forma, é possível replicar o mesmo PDRV diversas vezes através do simulador. Quando a simulação usa um dos algoritmos de roteamento periódicos para alocação dos veículos, também são inseridos na fila os eventos pré-programados que indicam o início de um algoritmo de roteamento dinâmico (Seção 5.3.1.7). No caso do algoritmo de roteamento *online*, o início de um novo algoritmo será inserido com o surgimento de novos pedidos.

Outros tipos de eventos surgem com o processamento destes eventos inicialmente inseridos na fila de prioridades. Todos eles são detalhados na Seção 5.3.1.

5.3.1 Eventos no simulador do PDRV

O simulador do PDRV implementado neste trabalho apresenta oito tipos de eventos distintos. Em alguns deles, seu tratamento possui variação em função do tipo de roteamento utilizado. Todos são detalhados nas subseções seguintes.

5.3.1.1 Pedido de atendimento

Um evento básico dentro da simulação de um dia de serviço no PDRV é o pedido de atendimento de um consumidor específico. Este evento possui dois argumentos: o código identificador do consumidor que solicita o serviço e o instante em que o consumidor solicita o serviço.

Esta tese separa os pedidos de atendimentos em dois grupos: estáticos e dinâmicos. Os estáticos são aqueles conhecidos antes da saída dos veículos do depósito central, e os dinâmicos são sempre identificados após a saída dos veículos do depósito central. O percentual de pedidos realizados dinamicamente determina o grau de dinamismo da instância (detalhes na Seção 4.3.1.1).

Quando o evento de um novo pedido de atendimento é processado, o comportamento do simulador varia de acordo com o tipo de roteamento que está sendo utilizado. Se ele está configurado para executar o algoritmo de roteamento *online*, um evento para o início de um algoritmo é gerado para início imediato, caso não haja algoritmo em execução. Se o roteamento em questão for um dos periódicos, o pedido é armazenado e nenhum novo evento é gerado para o simulador, pois todos os algoritmos dinâmicos programados para serem processos no futuro já estão na fila de eventos. O novo consumidor será tratado quando o próximo algoritmo de roteamento for processado pelo simulador. Mais detalhes sobre o início de um algoritmo podem ser vistos na Seção 5.3.1.7.

5.3.1.2 Partida de um veículo

Alguns eventos fundamentais para o PDRV estão relacionados com a movimentação da frota veicular. Um deles é a ordem de partida de um veículo, que, antes do processamento do evento, deve estar parado em um consumidor ou depósito de origem, para um destino bem definido. Este evento gera automaticamente um novo evento para o simulador, que indica a chegada do mesmo veículo no destino (Seção 5.3.1.3). O instante referente ao processamento do novo evento de chegada é diretamente proporcional a distância entre o ponto de origem e o ponto de destino, seguindo a característica do PRV. Nas versões estocásticas do PDRV, este tempo pode sofrer variações baseadas em distribuições de probabilidade para gerar perturbações, simulando, por exemplo, congestionamentos em horários específicos.

5.3.1.3 Chegada de um veículo

Quando um evento de chegada de um veículo em um consumidor ou no depósito é tratado, o simulador verifica automaticamente se não foi violada a janela de tempo do consumidor em que o veículo chega.

Se a janela de tempo é violada, ou seja, o instante de chegada do veículo é superior ao instante máximo permitido no intervalo da janela de tempo, um evento de partida do veículo é gerado para início imediato, como descrito na Seção 5.3.1.2.

Se o veículo chega ao consumidor antes do início permitido pela janela de tempo, um novo evento de espera é gerado com o instante de processamento no simulador igual ao início da janela de tempo do consumidor. Eventos de espera são detalhados na Seção 5.3.1.4.

Se o veículo chega ao consumidor durante o intervalo permitido para o atendimento, é gerado um evento de início de atendimento imediato (Seção 5.3.1.5).

5.3.1.4 Espera de um veículo

O evento de espera de um veículo é sempre gerado quando um veículo chegou antes do horário permitido para o atendimento de um consumidor específico, e sempre gera um novo evento para o simulador indicando o início do atendimento (Seção 5.3.1.5).

5.3.1.5 Início do atendimento de um consumidor

O processamento deste evento sempre se inicia dentro da janela de tempo do consumidor.

Quando processado, o evento de início de atendimento de um consumidor gera outro referente ao fim do atendimento do mesmo consumidor (Seção 5.3.1.6). O tempo do novo evento é calculado de acordo com o instante do evento de início de atendimento, somado ao tempo de serviço naquele consumidor. O instante gerado que determina o fim do atendimento do consumidor não necessariamente está dentro da janela de tempo do consumidor, de acordo com as restrições básicas do PRVJT.

5.3.1.6 Fim de atendimento de um consumidor

Terminado o atendimento em um consumidor, é gerado um novo evento de partida do veículo (Seção 5.3.1.2), para o próximo consumidor a ser visitado por aquele veículo, ou para o depósito central.

5.3.1.7 Início de um algoritmo de roteamento dinâmico

O evento de início de um algoritmo dinâmico pode ser criado de duas formas dentro do simulador do PDRV, dependendo do tipo de roteamento. Se o roteamento for *online*, o evento de início de um algoritmo dinâmico é criado imediatamente, se nenhum algoritmo estiver em execução, ou é indicado para o instante em que o algoritmo em execução termina. A segunda forma é referente ao roteamento periódico. No início da simulação, todos os algoritmos periódicos pré-programados são inseridos na fila de eventos do simulador.

Este evento, ao ser tratado, dispara, em uma linha de execução paralela, o algoritmo que resolve um PRVJT com Múltiplos Depósitos adaptado, para dar

resposta ao PDRV durante a simulação (mais detalhes sobre esta aplicação pode ser vista na Seção 6.4). O estado do problema deve ser capturado no instante de início do algoritmo, e armazenado. Assim, o algoritmo pode resolver o problema sem interferência na alteração do estado do ambiente pelo simulador. Após iniciado o algoritmo, um evento é gerado para o simulador referente ao fim do algoritmo dinâmico. O instante do fim do algoritmo também depende do roteamento utilizado pelo simulador.

5.3.1.8 Fim de um algoritmo de roteamento dinâmico

O evento do fim do algoritmo dinâmico é sincronizado com a linha de processamento paralela disparada pelo evento de início de algoritmo dinâmico. Terminado o algoritmo, a programação dos veículos é alterada em função da solução encontrada pelo algoritmo para o PRVJT com Múltiplos Depósitos adaptado (detalhes na Seção 6.4). Assim, quando o evento de partida de veículos for executado, existe a possibilidade do veículo se dirigir a um novo consumidor, diferente do planejamento passado. A alteração na programação da frota pode inclusive indicar a necessidade de um ou mais novos veículos, ainda parados no depósito central. Estes novos veículos saem imediatamente do depósito central, após o processamento deste evento.

5.4 Considerações finais

Este capítulo apresentou a descrição do simulador de eventos discretos implementado neste trabalho. Seu objetivo é fornecer a tese um ambiente emulado para avaliação dos três tipos de roteamento em análise.

Além da base fornecida para avaliação da hipótese desta tese, o simulador descrito neste capítulo pode ser estendido em outras pesquisas para problemas da classe dos PDRVs, inclusive inserindo características estocásticas ao ambiente.

Capítulo 6

Algoritmos para resolução do PDRV e do PRVJT

6.1 Considerações iniciais

Com o intuito de resolver com eficiência o PDRV, primeiramente foi desenvolvido um algoritmo para a versão estática do PRVJT, em um formato aplicável dentro do contexto dinâmico do problema (Seção 6.2). Neste caso, a exigência fundamental é o tempo de processamento do algoritmo. Outro algoritmo para a versão estática do PRVJT foi criado e é apresentado na Seção 6.3. Esta segunda versão não é aplicável dentro do contexto dinâmico do problema pelo seu alto tempo de processamento. A Seção 6.4 apresenta a resolução do PDRV.

6.2 Algoritmo Híbrido 1 para resolver o PRVJT Estático (aplicável ao PDRV)

Pode-se visualizar o PDRV como uma sequência de PRVs estáticos, quando o tempo é discretizado em períodos bem definidos. Este método é apresentado no trabalho de Chen e Xu [2006] e é detalhado na Seção 3.3.1. Desta forma, para resolver com qualidade o PDRV, é necessário resolver primeiramente com eficiência o PRV Estático.

Para tratar com eficiência o PRV Estático, este trabalho apresenta um Algoritmo Híbrido 1 (AH1), descrito em alto nível na Figura 6.1, que é uma combinação de um algoritmo exato (sobre a formulação do Problema de Particionamento de Conjuntos - PPC) e uma heurística estocástica (*Simulated Annealing* Não-Monotônico - SANM). O objetivo do AH1 é criar soluções de

qualidade para o PRVJT, para também ser utilizado no algoritmo de roteamento do PDRV, seja ele periódico ou *online*.

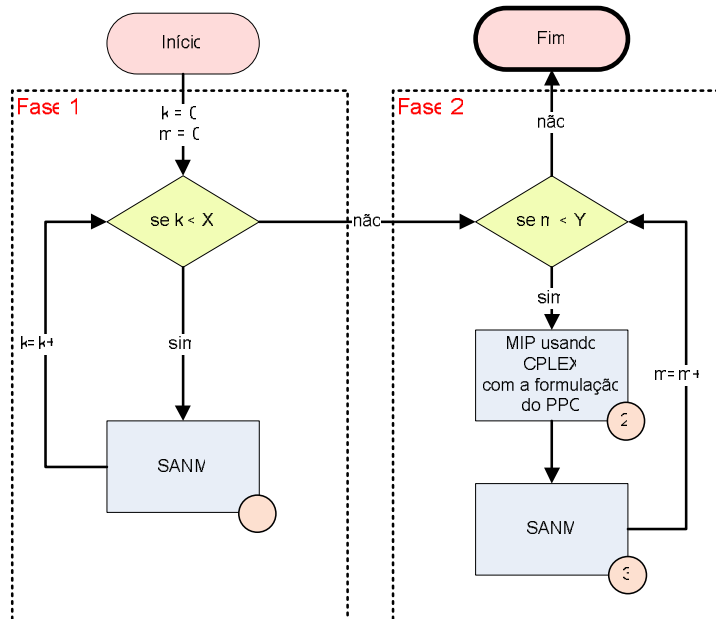


Figura 6.1 – Fluxograma do Algoritmo Híbrido 1 para o PRVJT

O AH1 possui duas fases distintas. Em linhas gerais, a primeira delas é responsável pela geração de colunas, em que servirão de entrada para o modelo do PPC, resolvido pelo *software* de programação matemática CPLEX na fase 2. As colunas são geradas pelo SANM, em X execuções distintas. O parâmetro X foi adaptado para cada tipo de roteamento. Detalhes são vistos na Seção 6.2.1.

A fase 2 é iniciada com o banco de colunas já gerado na fase 1. Utilizando as colunas do banco, o CPLEX resolve o modelo do PPC e determina uma nova solução para o problema. A solução gerada em cada execução do CPLEX é explorada por uma busca local que utiliza o SANM com parâmetros específicos (regulados para a intensificação, ao invés de diversificação, como na fase 1). Durante a execução do SANM da fase 2, novas colunas são geradas, e estas são incluídas no banco de colunas do AH1. Este procedimento é repetido Y vezes, retornando, ao final, a melhor solução encontrada durante todo o processo de otimização.

6.2.1 Parâmetros gerais do Algoritmo Híbrido

Os parâmetros X e Y , presentes no AH1 descrito na Figura 6.1, foram regulados de acordo com o tempo disponível para o processamento dos algoritmos de roteamento *online*, periódico 1 e periódico 2. No caso do algoritmo de roteamento periódico 1, duas foram as regulagens, dependendo da quantidade de períodos considerada dentro do PDRV (dez ou vinte). A Tabela 6.1 apresenta seus valores. Os valores atribuídos aos parâmetros X e Y do algoritmo *online* e do periódico 2 são iguais.

Tabela 6.1 - Parâmetros de AH1 dependentes do algoritmo de roteamento

Parâmetro	<i>Online</i> /Periódico 2	Periódico 1 (10)	Periódico 1 (20)
X	15	300	150
Y	3	60	30

No caso dos algoritmos de roteamento *online* e periódico 2, o tempo máximo de processamento foi de 3 minutos. No caso do algoritmo de roteamento periódico 1 o tempo estipulado foi de 60 minutos para o periódico que dividiu o dia em 10 partes, e de 30 minutos para o algoritmo periódico que dividiu o dia em 20 partes iguais.

6.2.2 *Simulated Annealing* não-monotônico (SANM)

O algoritmo SANM foi utilizado em dois pontos do AH1. O primeiro SANM na caixa 1 da Figura 6.1 tem o objetivo de gerar colunas de qualidade que servirão de entrada para o CPLEX na caixa 2. Já o segundo SANM na caixa 3 tem o objetivo de realizar uma busca local em torno da solução encontrada pelo CPLEX na resolução do PPC e também de gerar novas colunas para futuras execuções do CPLEX sobre a formulação matemática do PPC.

6.2.2.1 Descrição do *Simulated Annealing*

O *Simulated Annealing* é uma meta-heurística probabilística, proposta originalmente por Kirkpatrick *et al.* [1983], sendo um método de busca local que aceita movimentos que podem oferecer piora com relação à avaliação da solução, como estratégia para escapar de mínimos locais.

Esta meta-heurística é baseada em um método natural, fundamentado em uma analogia com a termodinâmica ao simular o resfriamento de um conjunto de átomos aquecidos. Esta operação é conhecida como recozimento (*annealing*) [Kirkpatrick *et al.*,

1983]. Em linhas gerais, o *Simulated Annealing* é descrito no Algoritmo 1, que considera um problema de minimização.

Algoritmo 1 - Descrição geral do SA

```

1: begin
2:   s = CriaSolucaoInicial(); //solução inicial do sistema
3:   t = 100; //temperatura do sistema começa com 100 unidades
4:   while t > 10-5;
5:     t = t * 0,95; //redução percentual da temperatura do sistema
6:     s' = SolucaoNaVizinhancaDe( s );
7:     Δ = f( s' ) - f( s );
8:     if Δ < 0 or NumeroAleatorio( ) < e-Δ/t then
9:       s = s'; //troca solução corrente por solução vizinha
10:    end if
11:  end while
12:  return s;
13: end

```

Em sua descrição padrão, o *Simulated Annealing* começa a busca a partir de uma solução inicial qualquer. Neste trabalho, a solução inicial é encontrada por uma heurística gulosa, detalhada na Seção 6.2.2.4. Logo após, na linha 3 do Algoritmo 1, é descrita a atribuição inicial da temperatura t do *Simulated Annealing*. Neste exemplo, foi dado o valor de 100 unidades. O laço de iterações, que caracteriza o procedimento principal, pode ser composto por diferentes critérios de parada. No exemplo do Algoritmo 1, o critério atribuído foi a proximidade da temperatura do valor zero. Assim como em outras meta-heurísticas probabilísticas, o momento de interromper a otimização no *Simulated Annealing* não é um processo de decisão trivial.

A cada iteração, a temperatura é reduzida, de forma não linear, atribuindo um valor percentual de t no instante anterior. A Figura 6.2 mostra sete decaimentos distintos da temperatura no *Simulated Annealing*. Quanto mais próximo de 1,0 for o fator de redução (FR), mais lento se torna a redução da temperatura ao longo do processo de otimização.

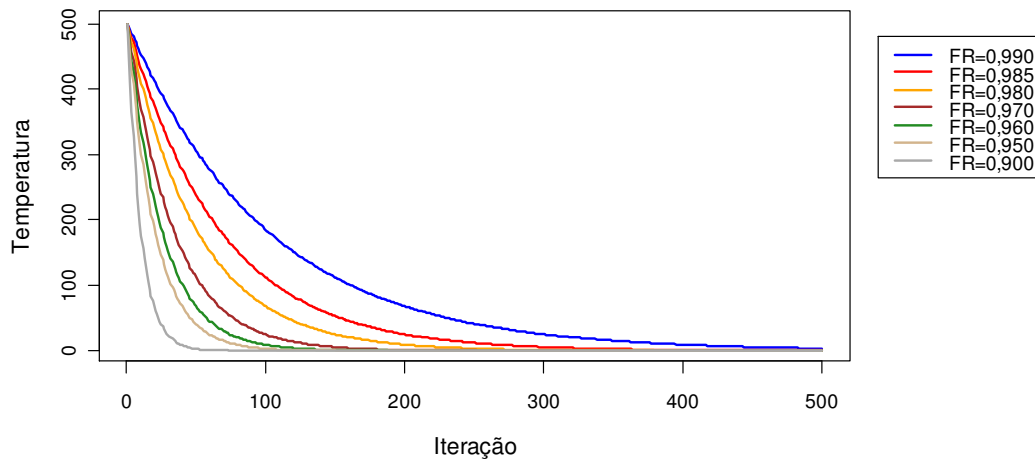


Figura 6.2 - Exemplos de decaimento da temperatura do SA

O trabalho de Oliveira *et al.* [2007] concluiu, através do método de análise de variância, que o fator de redução de temperatura é um dos parâmetros que possui influência estatisticamente significativa na qualidade da solução na resolução do PRVJT, considerando a minimização da distância total percorrida pela frota veicular.

A cada iteração, é gerado aleatoriamente um único vizinho s' da solução corrente s , através do procedimento explicado na Seção 6.2.2.5. A variação Δ (delta) do valor da função objetivo f (de minimização, por exemplo) é testada a cada iteração. Para o teste desta variação, é realizado o cálculo da linha 7 do Algoritmo 1. Se o valor de Δ for menor que zero, a nova solução s' é automaticamente aceita para substituir s . Caso contrário, a aceitação da nova solução s' dependerá da probabilidade estabelecida pelo Critério de Metropolis ($e^{-\Delta/T}$), que utiliza a temperatura t atual do sistema.

O Critério de Metropolis aceita com maior probabilidade soluções que possuem um pequeno valor para Δ . Altos valores de Δ terão chances menores se comparados a baixos valores de Δ . Quanto maior a temperatura, maior é a probabilidade de ser aceita a nova solução s' , justificando sua analogia ao resfriamento de sólidos.

O ponto chave do bom emprego do algoritmo *Simulated Annealing* depende diretamente da probabilidade de aceitação de uma solução vizinha pior, se comparada à solução atual do sistema (característica fundamental para fuga de mínimos locais do método). Para esclarecimento, a Figura 6.3 apresenta a relação do Δ

com a temperatura, exemplificando em alguns pontos a probabilidade de aceite da nova solução vizinha com avaliação pior na função objetivo.

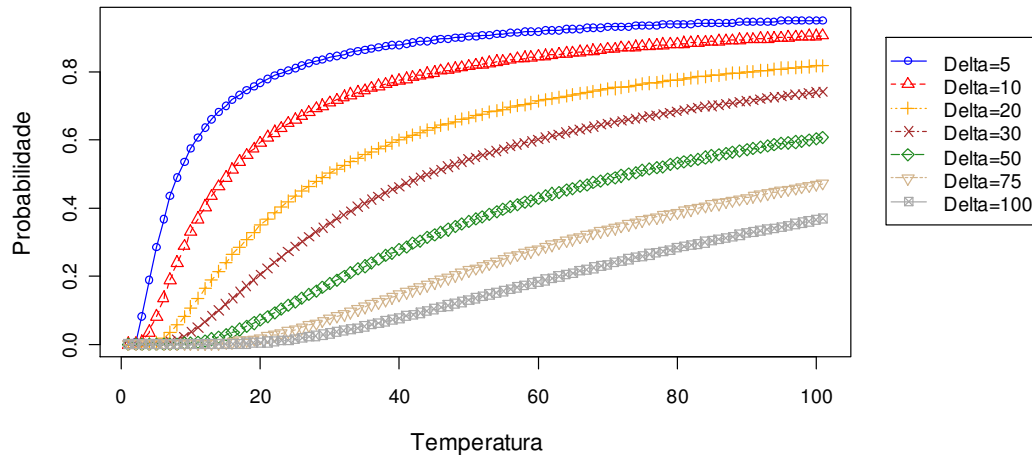


Figura 6.3 – Probabilidade do SA trocar a solução atual

Vale ressaltar que a grandeza escalar da função objetivo impacta diretamente na probabilidade de aceite das novas soluções, devido à alteração do Δ .

Uma descrição detalhada sobre o *Simulated Annealing* pode ser encontrada no trabalho de Kirkpatrick *et al.* [1983].

6.2.2.2 A Variação não-monotônica do *Simulated Annealing*

Além do decaimento da temperatura do SA padrão, o algoritmo híbrido apresentado neste capítulo efetua acréscimos na temperatura, caracterizando a não monotonicidade da variável t .

Tal procedimento pode ser visto como uma tentativa de fuga de mínimos locais, pois quando a temperatura t se aproxima de zero, o SA aceita com menores probabilidades novas soluções s' que possuem avaliação pior que o estado atual do sistema (solução s). Tal comportamento pode ser observado na Figura 6.3.

Desta forma, dois novos parâmetros são necessários:

- Temperatura mínima;
- Temperatura de reaquecimento.

Quando a temperatura do sistema atinge o valor da temperatura mínima, o sistema é reaquecido, possibilitando uma maior exploração do espaço de soluções.

Pode-se citar o trabalho de Thangiah *et al.* [1994], que também utiliza a não monotonicidade no *Simulated Annealing* para escapar de mínimos locais ao resolver o PRV.

6.2.2.3 Parâmetros do *Simulated Annealing*

Os valores dos parâmetros do *Simulated Annealing* Não-Monotônico, capaz de gerar as colunas iniciais para o PPC (Caixa 1 da Figura 6.1), são apresentados na Tabela 6.2.

Tabela 6.2 - Parâmetros do SANM (caixa 1)

Parâmetro	Valor
Solução inicial	algoritmo PFIH
Temperatura inicial	10 unidades
Fator de redução da temperatura	0,99%
Temperatura mínima	0,01 unidades
Temperatura de reaquecimento	10 unidades
Condição de parada	3 segundos

Com estes valores paramétricos, o SANM da caixa 1 possui comportamento de diversificação durante sua execução. Isso ocorre porque rapidamente a temperatura é reaquecida, permitindo ao algoritmo explorar soluções com características diferentes das anteriores. Quando sua temperatura está alta, ele funciona como um algoritmo de exploração, capaz de efetuar grandes saltos no espaço de soluções ao manipular um conjunto de rotas do PRVJT. Quando sua temperatura está baixa, possui característica de uma busca local. Estes passos se alternam quando o sistema é reaquecido ao atingir a temperatura mínima. Este SANM é executado X vezes, dependendo do algoritmo de roteamento utilizado, como explicado na Seção 6.2.1, com o objetivo único de gerar um banco de dados inicial de rotas de qualidade (banco de colunas). A solução inicial (de partida) do SANM da caixa 1 da Figura 6.1 é calculada através de um método heurístico conhecido como PFIH (*Push-Forward Insertion Heuristic*). O PFIH é descrito detalhadamente na Seção 6.2.2.4. Os parâmetros descritos na Tabela 6.2 foram regulados para satisfazer a necessidade de diversificação do meta-modelo.

O Algoritmo 2 descreve resumidamente a implementação do SANM da Caixa 1 da Figura 6.1 deste trabalho.

Algoritmo 2 - Descrição geral do SANM (responsável pela diversificação)

```

1: begin
2:   s = CriaSolucaoInicialComPFIH();
3:   t = 10;
4:   While tempoDeExecucao < 3 segundos
5:     t = t * 0,99; //redução percentual da temperatura do sistema
6:     if t < 0,01 then
7:       t = 10; //reaquecimento da temperatura
8:     end if
9:     s' = SolucaoNaVizinhancaDe( s );
10:    Δ = f( s' ) - f( s );
11:    if Δ < 0 or NumeroAleatorio() < e-Δ/t then
12:      s = s'; //troca solução corrente por solução vizinha
13:    end if
14:  end while
15:  return s;
16: end

```

Outro *Simulated Annealing* Não-Monotônico, também utilizado no Algoritmo Híbrido (caixa 3 da Figura 6.1), tem seus parâmetros descritos a seguir na Tabela 6.3. Com estes valores paramétricos, o sistema se comporta basicamente como uma busca local, pois, durante toda sua execução, não adquire temperaturas altas o suficiente para efetuar grandes saltos no espaço de soluções, a não ser com pequenas probabilidades, garantidas pelo Critério de Metropolis. Ao longo da execução, este SANM também armazena rotas no banco de rotas (somente aquelas ainda não encontradas durante a execução do Algoritmo Híbrido), que servirão de entrada para futuras execuções do CPLEX sobre a formulação matemática do PPC. Detalhes sobre este procedimento são encontrados no Apêndice A deste trabalho.

Tabela 6.3 - Parâmetros do SANM (caixa 3)

Parâmetro	Valor
Temperatura inicial	5 unidades
Fator de redução	0,999%
Temperatura mínima	0,1 unidades
Temperatura de reaquecimento	5 unidades
Condição de parada	20 segundos

O Algoritmo 3 descreve resumidamente a implementação do SANM da Caixa 3 da Figura 6.1 deste trabalho.

Algoritmo 3 - Descrição geral do SANM (responsável pela intensificação)

```
1: begin
2:   s = CriaSolucaoInicialComPFIH();
3:   t = 5;
4:   while tempoDeExecucao < 20 segundos
5:     t = t * 0,999; //redução percentual da temperatura do sistema
6:     if t < 0,1 then
7:       t = 5; //reaquecimento da temperatura
8:     end if
9:     s' = SolucaoNaVizinhancaDe( s );
10:    Δ = f(s') - f(s);
11:    if Δ < 0 or NumeroAleatorio( ) <  $e^{-\Delta/t}$  then
12:      s = s'; //troca solução corrente por solução vizinha
13:    end if
14:  end while
15:  return s;
16: end
```

Os parâmetros apresentados nesta seção foram regulados através da metodologia descrita no trabalho de Oliveira *et al.* [2007].

6.2.2.4 Solução inicial do SANM através do *Push-forward Insertion Heuristic*

O algoritmo *Push-Forward Insertion Heuristic* (PFIH) foi introduzido no trabalho de Solomon [1987]. Como citado em Larsen [2001], o PFIH possui uma estratégia construtiva eficiente para calcular o custo da inserção de um novo consumidor em uma rota. Este custo é calculado de acordo com sua posição geográfica, o fim de sua janela de tempo e o ângulo polar existente entre o consumidor e o depósito central.

Para o entendimento desta heurística, considere a Figura 6.4 como uma solução sendo construída, antes da inserção do consumidor C5.

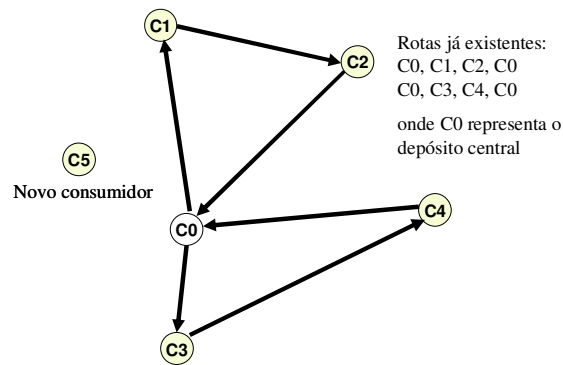


Figura 6.4 - Solução atual incompleta antes da inserção de um novo consumidor [Oliveira e Vasconcelos, 2010]

A qualidade das prováveis soluções é verificada através da inserção do consumidor em todas as possíveis posições do grafo da solução atual do PRVJT. Posteriormente é escolhida a posição onde a inserção do consumidor representou o menor custo (lembrando que este capítulo considera a distância total para computar a qualidade de suas soluções). Quanto menor a distância total de uma solução, maior sua qualidade.

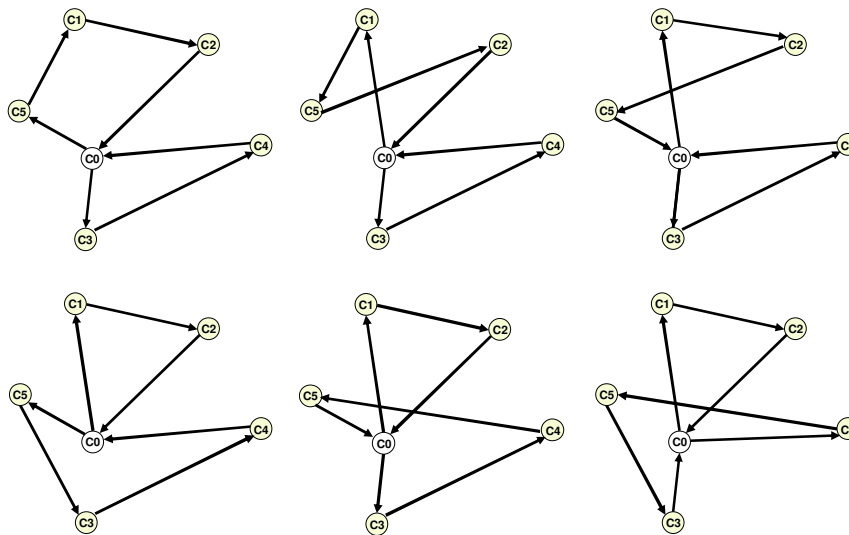


Figura 6.5 - Possíveis soluções encontradas pelo algoritmo PFIH de acordo com a Figura 6.4 [Oliveira e Vasconcelos, 2010]

A Figura 6.5 representa as possibilidades da inserção do consumidor c_5 em todas as possíveis posições da solução base apresentada na Figura 6.4.

Além de verificar os custos para a escolha da melhor posição, deve-se checar se a configuração após a inserção do consumidor não viola nenhuma das restrições do

PRVJT. Supondo que nenhuma das soluções descritas na Figura 6.5 viola restrições de capacidade de carga dos veículos ou tempo de atendimento dos consumidores, a nova solução para o PRVJT é representada pela possibilidade que apresenta a menor distância, sendo, neste caso, o grafo mostrado na Figura 6.6.

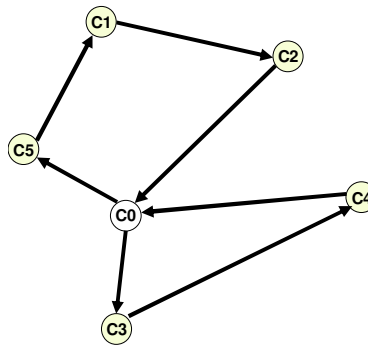


Figura 6.6 - Solução selecionada como melhor opção na inserção do consumidor C5 [Oliveira e Vasconcelos, 2010]

A ordem em que os consumidores são inseridos na solução do PRVJT, através do algoritmo PFIH, determina diretamente a qualidade da solução final produzida pelo método heurístico. Pesquisando sobre o problema, Solomon [1987] desenvolveu uma função heurística para determinar em que ordem os consumidores devem ser inseridos na solução (Equação 6.1). Esta função heurística computa o custo da inserção para cada consumidor i na solução em construção e, através deste custo, é estabelecida uma ordem para inserção de todos os consumidores. Por este motivo, o método PFIH é classificado como uma heurística construtiva e não de melhoria.

$$c_i = -\alpha \cdot d_{0i} + \beta \cdot b_i + \gamma \cdot \left[\left(\frac{p_i}{360} \right) \cdot d_{0i} \right] \quad \forall i \in C \quad (6.1)$$

sendo:

- $\alpha = 0,7$; $\beta = 0,1$; $\gamma = 0,2$; valores definidos empiricamente em Solomon [1987];
- d_{0i} : distância do depósito central ao consumidor i ;
- b_i : limite superior da janela de tempo de chegada ao consumidor i ;
- p_i : ângulo da coordenada polar do consumidor i , referente ao depósito central;

- c_i : custo de inserção do consumidor i .

Após a ordenação dos consumidores, seguindo a Equação 6.1, os mesmos são inseridos, um a um, na solução do PRVJT. A posição que resultar no menor acréscimo da distância total percorrida, sem violação de capacidade e janela de tempo, é aquela escolhida para a inserção do consumidor. Caso não exista posição para inserção do consumidor sem a violação de restrições, uma nova rota é criada abrigoando, a princípio, este consumidor. O método termina quando não mais existem consumidores na fila definida através da Equação 6.1.

6.2.2.5 Operadores para navegação no espaço de buscas

O algoritmo *Simulated Annealing* possui a necessidade de navegar no espaço de buscas do PRVJT, simulando a capacidade que os átomos têm de se mover no espaço 3D. Dentro deste contexto, é necessária a utilização de operadores de movimentação sobre seu espaço combinatório. Para navegar sobre este espaço, este trabalho utiliza cinco operadores distintos capazes de, através de uma solução s , gerar uma solução s' , sem violar as restrições do PRVJT. Quatro destes cinco operadores são métodos bem conhecidos de movimentação, utilizados amplamente na literatura como operadores de mutação em Algoritmos Evolucionários [Goldbarg e Luna, 2000; Eiben e Smith, 2003]. O quinto operador é específico para o PRVJT.

Os quatro primeiros possuem complexidade $O(n)$, e efetuam trocas de consumidores em busca de uma solução vizinha s' à solução atual s . Se s' viola alguma restrição do PRVJT, a mesma é descartada. Se s' não violar nenhuma restrição do problema, a mesma poderá ser aceita pelo *Simulated Annealing* de acordo com a avaliação do Critério de Metropolis.

O primeiro operador de vizinhança é o mais simples e é nomeado como operador de troca no trabalho de Eiben e Smith [2003]. A aplicação do operador de troca em uma solução s representa a permuta entre dois consumidores c_1 e c_2 de quaisquer rotas r_1 e r_2 , gerando uma solução s' . No exemplo da Figura 6.7, r_1 é igual a r_2 , ou seja, a troca de exemplo é efetuada na mesma rota do PRVJT.

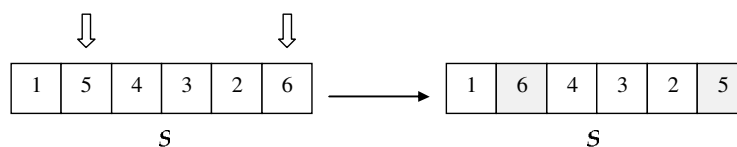


Figura 6.7 - Exemplo de aplicação do operador de vizinhança 'troca'

Outro operador de vizinhança utilizado por este trabalho, também baseado em mudanças aleatórias, é chamado de operador de inserção. Seu procedimento se resume em retirar um consumidor c_1 de uma rota qualquer r_1 , e inserir o mesmo consumidor c_1 em uma rota qualquer r_2 , em uma posição p aleatória. r_1 e r_2 podem ser a mesma rota, como no exemplo da Figura 6.8.

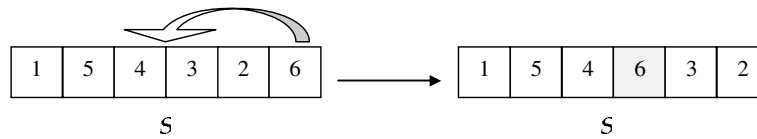


Figura 6.8 - Exemplo de aplicação do operador de vizinhança ‘inserção’

O terceiro operador aleatório utilizado neste trabalho é nomeado no trabalho de Eiben e Smith [2003] como operador de embaralhamento. Ele atua misturando uma sequência q qualquer de consumidores de uma rota qualquer r da solução s , gerando a solução s' com a nova sequência q' . Sua atuação é em somente uma rota. Um exemplo da aplicação do operador de embaralhamento é apresentado na Figura 6.9.

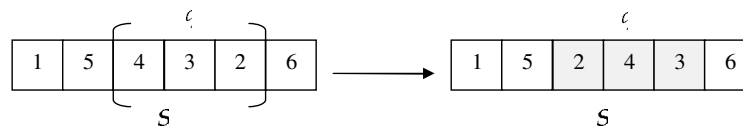


Figura 6.9 - Exemplo de aplicação do operador de vizinhança ‘embaralhamento’

O quarto operador aleatório utilizado é baseado na inversão de consumidores de uma sequência q qualquer de uma rota r . Uma aplicação de seu procedimento é exemplificada na Figura 6.10.

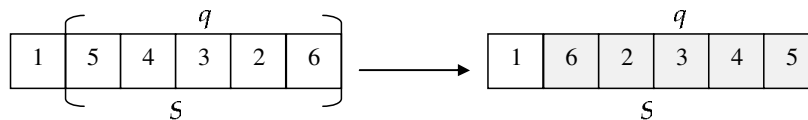


Figura 6.10 - Exemplo de aplicação do operador de vizinhança ‘inversão’

Todas as escolhas destes quatro primeiros operadores descritos, como por exemplo, as posições dos elementos a serem permutados no operador de troca, são feitas de maneira aleatória.

O quinto operador, denominado OP5 neste trabalho, é um operador heurístico e possui complexidade $O(n^3)$. Primeiramente são retirados m_i consumidores de cada rota i da solução s . O número de consumidores retirados m_i varia para cada rota r e é

escolhido retirando um valor de uma distribuição uniforme de zero até o número de consumidores presentes na rota i .

Após a retirada dos consumidores de s (gerando a solução incompleta h) todos os consumidores retirados são inseridos em h através do método PFIH, formando uma solução completa s' para o PRVJT. Este operador de vizinhança nunca gera uma nova solução s' que viola restrições, já que o algoritmo PFIH não fornece tal possibilidade. Portanto, diferentemente dos quatro primeiros operadores de vizinhança, para o OP5 não é necessário checar a viabilidade da nova solução encontrada. OP5 possui uma característica de acelerar a otimização durante a execução do *Simulated Annealing* no PRVJT por ter decisões que utilizam informações sobre a função objetivo, diferentemente dos outros quatro operadores utilizados.

Sempre que o método que busca uma solução na vizinhança do *Simulated Annealing* é invocado, o algoritmo escolhe aleatoriamente um dos cinco operadores para aplicação. No caso dos quatro primeiros, a nova combinação é aceita se não viola restrições do PRVJT. No caso do OP5, sempre a nova solução é aceita, pois o operador sempre retorna uma solução viável.

6.2.3 Problema de Particionamento de Conjuntos (PPC) no Algoritmo Híbrido

Para o PRVJT, existe uma quantidade exponencial de rotas em função da quantidade de consumidores a serem atendidos, como explicado na Seção 2.4.3. O PRVJT é um problema categorizado como NP-Difícil, e por este motivo, não é possível, computacionalmente, resolvê-lo com exatidão através de algoritmos exatos, em instâncias de grande porte. Em função desta restrição, este trabalho resolve o PPC, adaptado ao PRVJT, para um subconjunto R' do conjunto R . Este subconjunto é povoado com todas as rotas encontradas pela heurística SANM ao longo da execução do Algoritmo Híbrido 1. Neste trabalho, todas as novas rotas encontradas ao longo de sua execução são adicionadas ao conjunto R' . Como o SANM é uma busca direcionada em função de regiões promissoras no espaço de soluções, considera-se que o conjunto R' utilizado é um subconjunto de qualidade, podendo eventualmente conter todas as rotas ótimas do próprio conjunto R para a formulação do PPC. O Apêndice A apresenta detalhes algorítmicos para a implementação e manutenção do conjunto R' .

Para resolver o PPC foi utilizado o *software* de programação linear inteira CPLEX versão 12.2.0.0.

6.2.3.1 Parâmetros utilizados na resolução do PPC

Os parâmetros utilizados como entrada no pacote CPLEX, referentes à formulação do PPC, são descritos na Tabela 6.4.

Tabela 6.4 – Parâmetros do CPLEX do AH1 (Caixa 2 da Figura 6.1)

Parâmetro	Valor
Vetor solução x	Melhor solução conhecida até o momento
Tempo de processamento	30 segundos (máximo)
Colunas	Conjunto de todas as rotas viáveis já encontradas

O *software* CPLEX recebe três entradas. A primeira é referente às colunas da melhor solução inteira conhecida pelo sistema. Assim, o CPLEX já parte de um ponto do espaço de buscas conhecido, viável e inteiro. A segunda entrada é a limitação do tempo de processamento do CPLEX, estipulado em 30 segundos. Com esta limitação de tempo, não é garantido que o CPLEX retorne as colunas referentes à solução ótima para aquele subconjunto de colunas R' , mas sempre irá retornar uma solução viável, já que a melhor solução viável previamente conhecida já lhe foi informada. O terceiro parâmetro recebido são todas as colunas (rotas viáveis) conhecidas pelo algoritmo até o início da execução do CPLEX.

6.3 Algoritmo Híbrido 2 para resolver o PRVJT Estático (não aplicável ao PDRV)

Apesar do Algoritmo Híbrido 1 ser desenvolvido para ser aplicado dentro do contexto do PDRV, o mesmo obteve resultados próximos aos melhores descritos na literatura para a minimização da distância total na resolução do PRVJT. Após a análise destes resultados, foi proposto o Algoritmo Híbrido 2 (AH2), com o objetivo de ser aplicado exclusivamente em contextos estáticos, onde é permitido um tempo de processamento mais flexível. Este é descrito em alto nível na Figura 6.11.

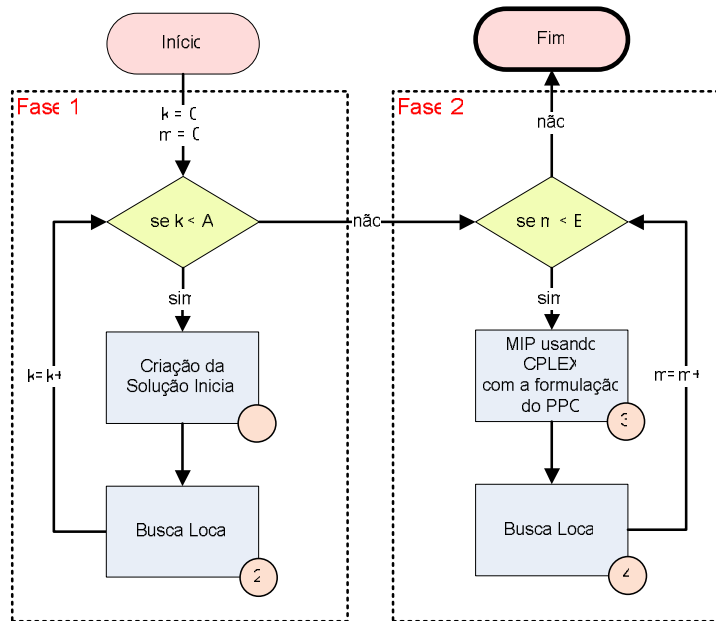


Figura 6.11 - Fluxograma do Algoritmo Híbrido 2 para o PRVJT

O AH2 faz uso de uma estratégia semelhante à utilizada no AH1 para gerar colunas para o Problema de Particionamento de Conjuntos na fase 1. A diferença fundamental é que no AH1 são aplicadas várias execuções do método *Simulated Annealing*. Já no AH2 a fase 1 corresponde ao método *Greedy Randomized Adaptive Search Procedure* (GRASP). O GRASP definido no trabalho de Resende [1995] é descrito no Algoritmo 4.

Algoritmo 4 - Descrição geral do GRASP

```

1: begin
2:   while condicaoDeParadaNaoSatisfeita();
3:     s = CriaSolucaoInicial ();
4:     s' = BuscaLocal( s );
5:     if MelhorSoluçãoEncontrada( s' ) then
6:       z = s'; //armazena a melhor solução encontrada
7:     end if
8:   end while
9:   return z;
10: end
  
```

Para a navegação no espaço de buscas, são utilizados os operadores descritos para o AH1 (Seção 6.2.2.5). No caso, a busca local opta por trocar sua solução quando localiza o primeiro vizinho que possui melhoria com relação à função objetivo.

A condição de parada de cada busca local é seu tempo de processamento, sendo fixado em 1 minuto e 40 segundos. Neste trabalho, o GRASP realiza 50 buscas locais ($A=50$).

A fase 2 do AH2 também é semelhante a segunda fase do AH1, tendo, contudo, diferença quanto ao tempo de processamento. Além disso, ela resolve o Problema de Particionamento de Conjuntos com um conjunto de colunas R' consideravelmente maior que no AH1, pois a quantidade de rotas distintas encontradas no AH2 é superior pelo maior tempo de exploração do espaço de buscas. A resolução do PPC é limitada a 8 minutos de processamento. A busca local, presente na segunda fase, é limitada a 2 minutos de execução. Todos os parâmetros foram regulados manualmente.

6.4 Resolução do PDRV

A estratégia para resolver o PDRV é subdividida em duas fases distintas. A primeira delas é referente à resolução da versão estática do PRVJT antes da saída dos veículos do depósito central, como exemplificado na Figura 1.1, na introdução desta tese. Com os veículos em movimento, e com a apresentação de demandas dinâmicas (novos clientes), o algoritmo que resolve o problema é chamado novamente, em um instante que depende do tipo de algoritmo de roteamento utilizado. Na fase dinâmica, no entanto, não se resolve um problema exatamente igual ao PRVJT. Deve-se considerar uma adaptação do PRVJT com Múltiplos Depósitos. Esta adaptação do problema é detalhada na Seção 6.4.1.

O tempo de execução para resolver estes problemas na fase dinâmica varia de acordo com o tipo de algoritmo de roteamento utilizado. Se o roteamento for o *online* ou o periódico 2, este trabalho utiliza um limite de tempo igual a 3 minutos de execução. Neste caso, Todas as requisições recebem resposta no máximo em 3 minutos após o surgimento da demanda. No caso do roteamento periódico 1, os pedidos são acumulados até o instante de tempo que o início do novo algoritmo foi programado. Em qualquer um dos três processos de roteamento, quando um algoritmo é invocado, o sistema reconhece todas as variáveis observáveis do ambiente, e calcula como estará o ambiente no futuro (no instante de resposta do algoritmo). Este processo é detalhado na Seção 6.4.1.

6.4.1 PRVJT com Múltiplos Depósitos (PRVJTMD) adaptado

Ao receber requisições dinâmicas para o atendimento, o algoritmo de roteamento deve reconhecer quais são os trechos de rota que poderão ser alterados, quando o próprio algoritmo responder ao ambiente em um instante futuro. Dois estados distintos devem ser considerados neste caso. O primeiro deles é o estado do ambiente no instante que o algoritmo será iniciado (exemplo na Figura 6.12).

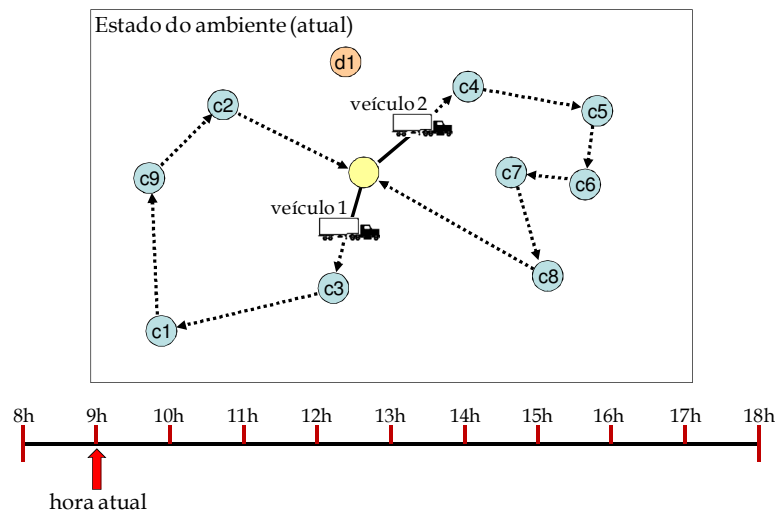


Figura 6.12 - Exemplo do estado simplificado do ambiente quando o algoritmo dinâmico é chamado

O segundo será o estado do ambiente no instante em que o algoritmo responderá com a nova programação para a frota veicular, após a sua execução (exemplo na Figura 6.13).

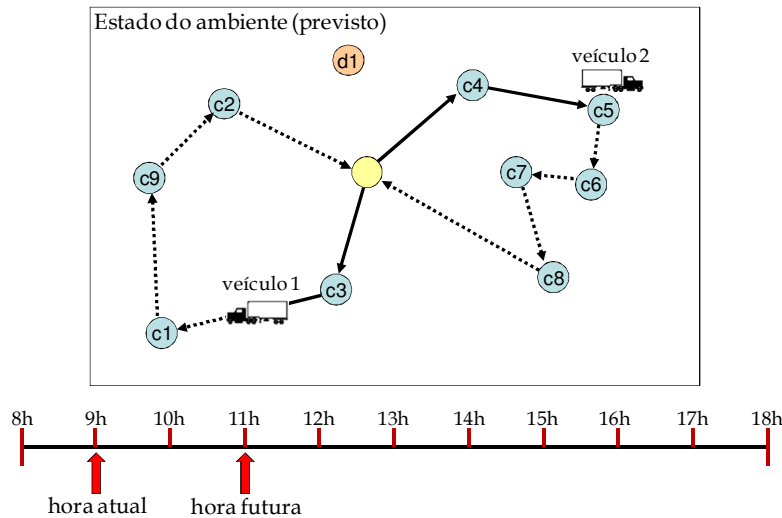


Figura 6.13 – Exemplo do estado simplificado do ambiente quando o algoritmo dinâmico termina sua execução

O algoritmo dinâmico deve considerar a previsão do estado futuro para executar a otimização. Caso contrário, pode reprogramar arestas que, no futuro, já terão sido percorridas pelos veículos, criando uma alocação inviável no PDRV.

No caso do PDRV sem componentes estocásticas para os tempos de viagem e atendimento, a previsão representa o estado exato do ambiente quando o algoritmo dinâmico responder no futuro. Caso existam componentes estocásticas no modelo, a previsão pode levar o algoritmo a criar uma alocação inviável para o instante futuro.

Identificado o estado futuro do ambiente, referente ao instante de resposta do algoritmo, o sistema de roteamento deve transformar este estado em um PRVJT com Múltiplos Depósitos adaptado, como exemplificado na Figura 6.14.

Determinado o estado futuro do ambiente, o PRVJTMD adaptado é composto por:

- Veículos que ainda não se dirigiram para o depósito central;
- Consumidores ainda não visitados. Alguns consumidores estarão necessariamente no início de cada programação, pois serão os pontos de origem dos veículos ativos (múltiplos depósitos);
- Depósito central para a volta dos veículos.
- Novos consumidores dinâmicos.

Diferente do PRVJTMD original, este trabalho realiza três adaptações para resolver o PDRV de forma adequada:

- O problema aqui tratado possui múltiplos depósitos de saída (como no PRVJTMD original) e apenas um depósito de chegada (diferente do PRVJTMD original). O depósito de chegada é o depósito central do PDRV.
- O PRVJTMD adaptado também é composto por parâmetros que indicam o horário de saída de cada consumidor de origem para o algoritmo de otimização tratar de forma adequada as restrições temporais do problema;
- No PRVJTMD adaptado, cada depósito de origem possui apenas um veículo em sua frota.

Considerando o exemplo Figura 6.12, um exemplo de programação que pode ser informada ao ambiente é mostrado na Figura 6.14, em que os consumidores c1 e c5 representam dois depósitos de saída dos veículos no PRVJTMD adaptado.

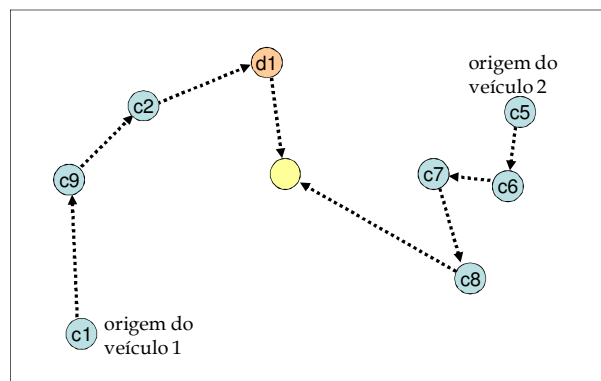


Figura 6.14 - Exemplo de solução para o PRVJT com Múltiplos Depósitos adaptado

Nesta tese, é considerada a programação já realizada por algoritmos executados antes de iniciar o processo de otimização do PRVJTMD adaptado. Assim, os consumidores dinâmicos são inseridos na solução candidata através do método *Push-Forward Insertion Heuristic* e o *Simulated Annealing Não Montônico com Geração de Colunas* (descrito nas Seções 6.2.2.1, 6.2.2.2 e 6.2.3) é iniciado sobre a nova proposta de solução.

6.4.2 Resolução do PPC durante a fase dinâmica do PDRV

Durante a fase dinâmica do PDRV, é necessária uma adaptação da aplicação do modelo do PPC, explicado nas Seções 2.4.3 e 6.2.3.

Para a resolução do PRVJT através do modelo do PPC, o depósito não é levado em consideração. Diferente da resolução do PRVJT estático, ao resolver um PRVJTMD adaptado, é necessário considerar o depósito de saída de cada veículo k em sua rota correspondente r . Assim, além dos consumidores que ainda devem ser atendidos, o parâmetro δ_{ir} é igual a 1 também para cada depósito de origem do PRVJTMD adaptado. Isso evita que a resposta ao PPC contenha o mesmo veículo sendo selecionado mais de uma vez para compor a solução do problema.

6.5 Considerações finais

Este capítulo apresentou dois Algoritmos Híbridos (AH1 e AH2), que combinam o método *Simulated Annealing* não-monotônico (SANM) e o GRASP com a execução do *software* de programação matemática (CPLEX) que resolve a formulação do Problema de Particionamento de Conjuntos (PPC) aplicado ao PRVJT.

O SANM é o algoritmo que gera colunas de qualidade para o PPC ser resolvido sobre a plataforma CPLEX. O Algoritmo Híbrido 1 é dividido basicamente em duas fases: a primeira é composta de rápidas execuções do SANM apenas com o intuito de gerar colunas. A segunda fase é composta de ciclos, que, após resolver o PPC, efetua, sobre a solução retornada pelo CPLEX, uma busca local. Desta forma, ocorre uma mútua colaboração entre um algoritmo exato (implementado no pacote CPLEX) e uma heurística probabilística (*Simulated Annealing*).

O AH1 proposto neste capítulo oferece uma contribuição significativa na pesquisa das melhores técnicas para resolução rápida do PRVJT, aplicável dentro de contextos dinâmicos, apresentando bons resultados na computação da base de testes de Solomon [1987].

O AH2 faz uso do GRASP para localizar soluções robustas do PRVJT, embora não tenha a mesma necessidade de um processamento ágil do AH1. Neste sentido, sua utilização é exclusiva dentro de contextos estáticos.

Durante o detalhamento da resolução do PDRV, este capítulo apresentou o processo para constatar o PRVJT com Múltiplos Depósitos, referente ao estado futuro do ambiente para a resolução do PDRV. Assim, independente do tipo de roteamento utilizado, o AH1 para versões estáticas do problema é aplicado a sucessivos PRVJTMD para resolução do PDRV.

Os resultados obtidos pelos algoritmos apresentados nas seções anteriores são apresentados no Capítulo 7.

Capítulo 7

Resultados

7.1 Considerações iniciais

Como exemplificado no Figura 1.1 (página 4), a resolução do PDRV apresenta duas fases bem definidas:

- A fase responsável pela resolução da demanda estática inicial (referente aos consumidores conhecidos antes da partida dos veículos do depósito central); e
- A fase em que replanejamentos devem ser considerados para o atendimento de demandas dinâmicas (consumidores que surgem ao longo do dia de serviço, com parte da frota veicular já em andamento).

Para resolver a versão estática do problema e para gerar planejamentos considerando demandas dinâmicas, é necessário resolver de forma eficaz a versão estática do PRVJT. Considerando este contexto, esta tese implementou um Algoritmo Híbrido (descrito na Seção 6.2) para o PRVJT com o intuito de resolver de forma eficaz o PDRV. Resultados sobre a versão estática do PRVJT são encontrados na Seção 7.2. Para a avaliação da hipótese da tese, a Seção 7.3 apresenta os resultados e testes estatísticos sobre a versão dinâmica do problema. Para referências comparativas, foram encontrados limites inferiores para as instâncias estendidas para o PDRV. Assim, têm-se base para analisar a qualidade dos algoritmos de roteamento em função dos limites dos 48 cenários estudados. Todos os resultados apresentados neste capítulo foram alcançados com o objetivo de reduzir a distância total percorrida pela frota de veículos.

7.2 Resolução da versão estática do PRVJT

Para a avaliação dos Algoritmos Híbridos deste trabalho (AH1 e AH2), a Seção 7.2.2 apresenta os resultados sobre a base de testes de Solomon [1987]. A Seção 7.2.3 mostra um comparativo com outros trabalhos que resolvem o PRVJT considerando a redução da distância total. Considerações sobre o tempo de processamento são realizadas na Seção 7.2.4.

7.2.1 Base de testes para o PRVJT estático

Existe uma grande quantidade de publicações utilizando heurísticas e meta-heurísticas na resolução do PRVJT. Para analisar a qualidade e robustez dos algoritmos, estes são geralmente aplicados sobre as instâncias de Solomon [1987], justamente pelo fato de haver um maior número de resultados publicados sobre elas, possibilitando assim fácil comparação e análise. Mais detalhes sobre as instâncias de Solomon [1987] podem ser encontrados na Seção 4.3 desta tese.

7.2.2 Resultados da resolução do PRVJT

Para facilitar a visualização dos resultados, as tabelas desta seção indicam se os Algoritmos Híbridos, propostos para resolver o PRVJT estático, obtiveram resultados melhores, piores ou iguais aos melhores resultados já publicados na literatura na minimização da distancia total viajada.

Tabela 7.1 – Resultados dos Algoritmos Híbridos para o PRVJT sobre o grupo R1

Instância	Outros trabalhos			AH1		AH2	
	Veículos	Distância	Autor	Veículos	Distância	Veículos	Distância
R101	20	1642,88	Jung e Moon [2002]	20	1642,88	20	1642,88
R102	18	1472,62	Alvarenga <i>et al.</i> [2007]	18	1472,81	18	1472,81
R103	14	1213,62	Rochat e Taillard [1995]	14	1213,62	14	1213,62
R104	11	976,61	Jung e Moon [2002]	11	976,61	11	976,61
R105	13	1360,78	Jung e Moon [2002]	15	1360,78	15	1360,78
R106	13	1239,37	Ribas <i>et al.</i> [2011]	13	1239,37	13	1239,37
R107	11	1073,34	Jung e Moon [2002]	11	1075,21	11	1072,12
R108	10	948,57	Alvarenga [2005]	11	959,12	10	938,20
R109	13	1101,84	Jung e Moon [2002]	13	1151,84	13	1151,84
R110	12	1072,41	Jung e Moon [2002]	12	1072,41	12	1072,41
R111	12	1053,50	Jung e Moon [2002]	12	1053,50	12	1053,50
R112	10	953,63	Rochat e Taillard [1995]	11	980,10	10	953,68

Tabela 7.2 – Resultados dos Algoritmos Híbridos para o PRVJT sobre o grupo R2

Instância	Outros trabalhos			AH1		AH2	
	Veículos	Distância	Autor	Veículos	Distância	Veículos	Distância
R201	8	1147,80	Oliveira <i>et al.</i> [2007]	6	1157,79	8	1147,80
R202	8	1034,35	Jung e Moon [2002]	7	1037,23	6	1036,27
R203	6	874,87	Jung e Moon [2002]	6	874,87	6	874,87
R204	5	735,80	Oliveira <i>et al.</i> [2007]	5	735,80	5	735,80
R205	5	954,16	Oliveira <i>et al.</i> [2007]	5	957,83	5	954,16
R206	5	879,89	Jung e Moon [2002]	5	884,85	5	884,25
R207	4	797,99	Oliveira <i>et al.</i> [2007]	4	799,86	4	797,99
R208	4	705,45	Jung e Moon [2002]	3	707,01	3	706,74
R209	5	859,39	Jung e Moon [2002]	5	860,11	5	860,11
R210	5	910,70	Jung e Moon [2002]	6	911,34	5	908,88
R211	4	755,82	Oliveira <i>et al.</i> [2007]	4	757,64	4	753,15

Tabela 7.3 – Resultados dos Algoritmos Híbridos para o PRVJT sobre o grupo C1

Instância	Outros trabalhos			AH1		AH2	
	Veículos	Distância	Autor	Veículos	Distância	Veículos	Distância
C101	10	828,94	Rochat e Taillard [1995]	10	828,94	10	828,94
C102	10	828,94	Rochat e Taillard [1995]	10	828,94	10	828,94
C103	10	828,06	Rochat e Taillard [1995]	10	828,06	10	828,06
C104	10	824,78	Rochat e Taillard [1995]	10	824,78	10	824,78
C105	10	828,94	Rochat e Taillard [1995]	10	828,94	10	828,94
C106	10	828,94	Rochat e Taillard [1995]	10	828,94	10	828,94
C107	10	828,94	Rochat e Taillard [1995]	10	828,94	10	828,94
C108	10	828,94	Rochat e Taillard [1995]	10	828,94	10	828,94
C109	10	828,94	Rochat e Taillard [1995]	10	828,94	10	828,94

Tabela 7.4 – Resultados dos Algoritmos Híbridos para o PRVJT sobre o grupo C2

Instância	Outros trabalhos			AH1		AH2	
	Veículos	Distância	Autor	Veículos	Distância	Veículos	Distância
C201	3	591,56	Rochat e Taillard [1995]	3	591,56	3	591,56
C202	3	591,56	Rochat e Taillard [1995]	3	591,56	3	591,56
C203	3	591,17	Rochat e Taillard [1995]	3	591,17	3	591,17
C204	3	590,60	Rochat e Taillard [1995]	3	590,60	3	590,60
C205	3	588,88	Rochat e Taillard [1995]	3	588,88	3	588,88
C206	3	588,49	Rochat e Taillard [1995]	3	588,49	3	588,49
C207	3	588,29	Rochat e Taillard [1995]	3	588,29	3	588,29
C208	3	588,32	Rochat e Taillard [1995]	3	588,32	3	588,32

Tabela 7.5 – Resultados dos Algoritmos Híbridos para o PRVJT sobre o grupo RC1

Instância	Outros trabalhos			AH1		AH2	
	Veículos	Distância	Autor	Veículos	Distância	Veículos	Distância
RC101	15	1623,58	Rochat e Taillard [1995]	15	1623,58	15	1623,58
RC102	14	1461,23	Jung e Moon [2002]	14	1461,23	14	1461,23
RC103	11	1249,86	Jung e Moon [2002]	12	1284,63	11	1261,67
RC104	10	1135,48	Cordeau <i>et al.</i> [2001]	10	1135,83	10	1135,52
RC105	16	1518,58	Jung e Moon [2002]	16	1518,58	16	1518,58
RC106	13	1377,35	Alvarenga <i>et al.</i> [2007]	13	1378,23	12	1376,26
RC107	12	1212,83	Jung e Moon [2002]	12	1215,66	12	1211,11
RC108	11	1117,53	Jung e Moon [2002]	11	1118,12	11	1114,42

Tabela 7.6 – Resultados dos Algoritmos Híbridos para o PRVJT sobre o grupo RC2

Instância	Outros trabalhos			AH1		AH2	
	Veículos	Distância	Autor	Veículos	Distância	Veículos	Distância
RC201	9	1265,56	Jung e Moon [2002]	9	1265,56	9	1265,56
RC202	8	1095,64	Jung e Moon [2002]	8	1095,64	8	1095,64
RC203	5	926,89	Oliveira <i>et al.</i> [2007]	5	926,82	5	926,82
RC204	4	786,38	Jung e Moon [2002]	4	786,38	4	786,38
RC205	7	1157,55	Jung e Moon [2002]	7	1157,66	7	1157,55
RC206	7	1054,61	Jung e Moon [2002]	5	1068,51	6	1055,59
RC207	6	966,08	Jung e Moon [2002]	5	970,78	6	966,08
RC208	4	778,93	Oliveira <i>et al.</i> [2007]	4	778,93	4	778,93

Comparado com os resultados já apresentados na literatura, oito novos limites primais foram encontrados pelo AH2 para a minimização da distância total percorrida na base de testes de Solomon [1987]. Isso representa uma melhoria em 14,28% das instâncias de teste.

Das 56 instâncias de Solomon [1987], a Tabela 7.7 apresenta o percentual de instâncias que foram superadas ou igualadas pelos Algoritmos Híbridos deste trabalho, em comparação com os melhores resultados da literatura.

Tabela 7.7 – Percentual de resultados iguais ou superados por cada algoritmo

Classe	AH1	AH2
R1	58,33%	75,00%
R2	18,18%	63,64%
C1	100,00%	100,00%
C2	100,00%	100,00%
RC1	37,50%	75,00%
RC2	62,50%	87,50%
Geral	60,71%	82,14%

7.2.3 Comparação com outros trabalhos

Os trabalhos Alvarenga [2005], de Backer e Furmon [1997], Jung e Moon [2002], Oliveira e Vasconcelos [2010], Ribas *et al.* [2011], Mao e Deng [2010] e Riise e Stølevik [1999] são comparados com os resultados dos Algoritmos Híbridos 1 e 2 através da base de testes de Solomon [1987]. São mostradas, por algoritmo (colunas), as médias do número de veículos (NV) e da distância total viajada (DT) para cada classe (linhas), além do número de veículos acumulado (NVA) e da distância total acumulada (DTA) para todos os grupos da base de testes de Solomon [1987].

Tabela 7.8 - Comparação entre diferentes trabalhos que minimizam a distância viajada no PRVJT

Classe	Trabalho	Trabalho								
		Jung e Moon [2002]	Oliveira e Vasconcelos [2010]	Alvarenga [2005]	de Backer e Furmon [1997]	Riise e Stølevik [1999]	Mao e Deng [2010]	Ribas <i>et al.</i> [2011]	Algoritmo Híbrido 1	Algoritmo Híbrido 2
R1	NV	13,25	13,33	13,25	14,17	13,92	13,75	13,17	13,42	13,25
	DT	1179,95	1186,94	1183,38	1214,86	1211,22	1192,17	1181,03	1183,19	1178,99
R2	NV	5,36	5,36	5,55	5,27	4,91	5,36	5,36	5,09	5,09
	DT	878,41	878,79	899,90	930,18	917,54	935,14	883,10	880,39	878,18
C1	NV	10,00	10,00	10,00	10,00	10,56	10,56	10,00	10,00	10,00
	DT	828,38	828,38	828,38	829,77	846,88	833,35	828,38	828,38	828,38
C2	NV	3,00	3,00	3,00	3,25	3,88	3,25	3,00	3,00	3,00
	DT	589,39	589,86	589,86	604,84	598,70	591,86	589,86	589,86	589,86
RC1	NV	13,00	13,25	12,88	14,25	13,75	13,50	12,75	12,87	12,63
	DT	1343,64	1362,44	1341,67	1385,12	1399,76	1354,63	1338,54	1341,98	1337,80
RC2	NV	6,25	6,13	6,50	6,25	5,63	7,00	6,13	5,87	6,13
	DT	1004,21	1004,59	1015,90	1099,96	1055,61	976,62	1009,17	1006,28	1004,07
Tempo médio		3480s	660s	3600s	---	1740s	---	600s	180s	9030s
Todas as classes	NVA	486	488	489	508	502	509	482	481	479
	DTA	54779	55020	55134	56998	56682	55781	54842	54843	54717

A Figura 7.1 ilustra a distância acumulada para todas as instâncias de Solomon [1987], comparando seus desempenhos.

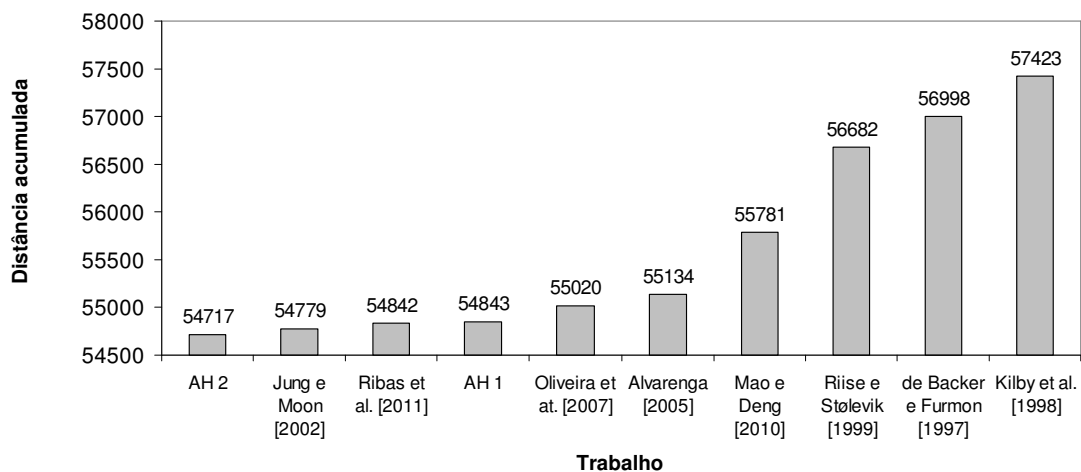


Figura 7.1 - Distância total acumulada para todas as instâncias de Solomon [1987]

A Figura 7.2 apresenta os veículos acumulados na resolução de todas as instâncias de Solomon [1987], para cada um dos trabalhos comparados.

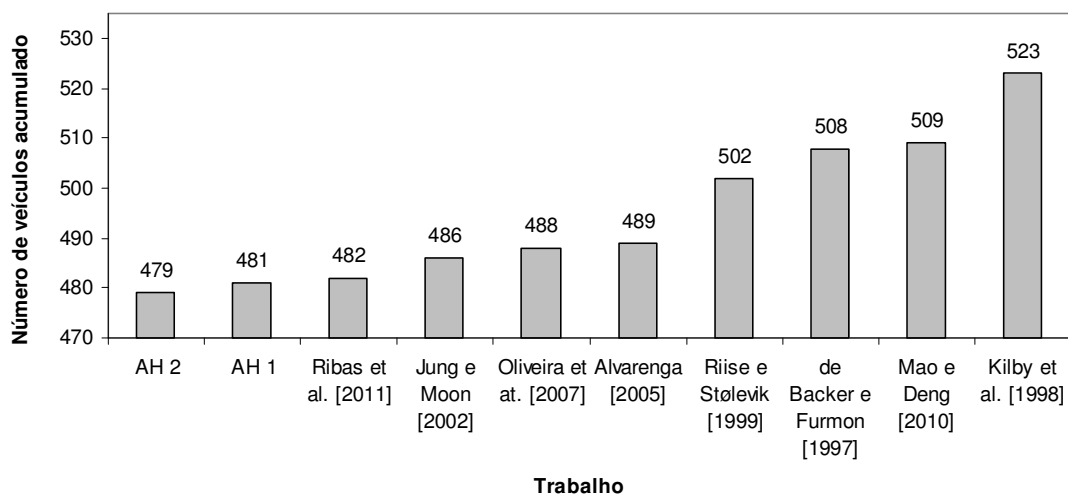


Figura 7.2 – Veículos acumulados para todas as instâncias de Solomon [1987]

7.2.4 Tempo de processamento

O tempo médio de processamento do AH1 e do AH2, para cada grupo de instâncias da base de testes de Solomon [1987], pode ser visto na Tabela 7.9.

Tabela 7.9 – Tempo médio de processamento dos algoritmos que resolvem o PRVJT

Classe	Tempo médio de processamento (em segundos)	
	AH1	AH2
R1	177,15	9005,71
R2	171,61	9045,74
C1	183,38	8998,76
C2	181,48	9084,23
RC1	184,86	9345,31
RC2	191,49	8705,80
Geral	180,83	9030,92

A Figura 7.3 compara o tempo de processamento médio dos algoritmos mais competitivos que se propõem a solucionar o PRVJT com a minimização da distância total.

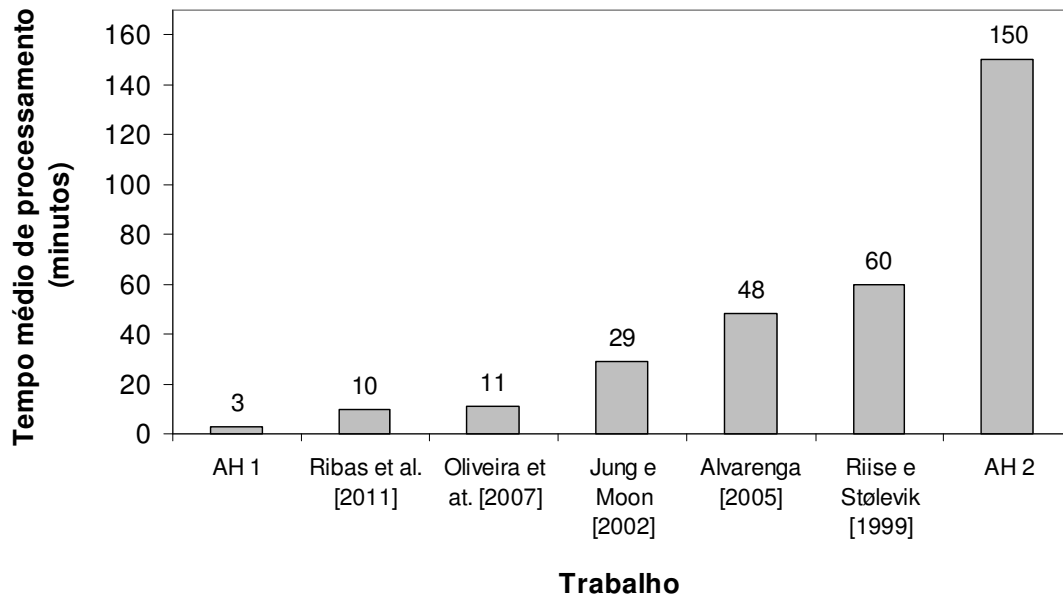


Figura 7.3 - Tempo médio de execução nas instâncias de Solomon [1987]

Para este tipo de comparação, considerando o tempo de processamento, é interessante informar o tipo de computador utilizado para o processamento dos algoritmos propostos. A Tabela 7.10 apresenta os dados dos trabalhos que disponibilizaram esta informação.

Tabela 7.10 - Processadores utilizados para resolver as instâncias de Solomon [1987]

Trabalho	Processador	Linguagem
Algoritmo Híbrido 1	Core2Quad, 2,4 GHz	Java
Algoritmo Híbrido 2	Core2Quad, 2,4 GHz	Java
Oliveira <i>et al.</i> [2007]	Centrino (Pentium M), 1.7 GHz	C#
Jung e Moon [2002]	Pentium III, 1.0 GHz	C++
Alvarenga [2005]	Pentium 4, 2.4 GHz	Delphi
Ribas <i>et al.</i> [2011]	Intel Quad Core, 2.4 GHz	C++
Mao e Deng [2010]	Intel Core Duo, 1.6 GHz	Matlab

Vale ressaltar que os algoritmos AH1 e AH2 utilizam todos os núcleos do processador durante a resolução do PPC executada pelo CPLEX. No restante dos mesmos, apenas um núcleo é utilizado. Detalhes sobre a quantidade de núcleos utilizados pelos algoritmos não são encontrados nos outros trabalhos citados na Tabela 7.10, impossibilitando para uma comparação mais detalhada.

7.3 Resultados da versão dinâmica do PRV

7.3.1 Base de testes para o PDRV

A base de testes utilizada, para comparação das abordagens *online* e periódica e na tentativa de corroboração da hipótese desta tese, foi estendida da base de Solomon [1987]. Detalhes podem ser vistos no capítulo que detalha a metodologia, especificamente na Seção 4.3.1.

7.3.2 Resultados na resolução do PDRV

As tabelas desta seção apresentam o desempenho dos três tipos de roteamento avaliados nesta tese (*online*, periódico 1 e periódico 2). Nas tabelas, são mostradas médias e desvios-padrão para as 48 instâncias estendidas. Cada tabela apresenta um grupo de instâncias derivado de uma instância original de Solomon [1987]. Para as tabelas desta seção, a sigla TD indica o tipo de distribuição dos consumidores dinâmicos ao longo do dia (detalhes na Seção 4.3.1.3), QP apresenta a quantidade máxima de períodos permitida para a versão do roteamento periódico (detalhes são vistos na Seção 4.3.1.2), GD representa o grau de dinamismo da instância (detalhes na Seção 4.3.1.1), \bar{X} é a média amostral e *s* indica o desvio padrão amostral.

Tabela 7.11 - Resultados para o grupo de instâncias baseadas em C101

Tipo de roteamento			Online		Periódico 1				Periódico 2						
			Veículos		Distância		Veículos		Distância		Veículos		Distância		
Instância	TD	QP	GD	\bar{X}	<i>s</i>	\bar{X}	<i>s</i>	\bar{X}	<i>s</i>	\bar{X}	<i>s</i>	\bar{X}	<i>s</i>	\bar{X}	<i>s</i>
N	10	25		19,20	0,40	1399,30	8,00	22,10	0,30	1564,33	6,00	18,50	0,81	1398,07	13,34
N	10	75		26,50	0,50	1879,57	12,02	42,00	0,00	2610,76	1,08	27,10	0,54	1892,28	14,20
N	20	25		15,00	0,00	1187,60	3,16	16,00	0,00	1208,94	0,00	15,00	0,00	1217,99	0,53
N	20	75		24,10	0,30	1757,94	16,34	29,50	0,50	2167,62	27,87	23,70	0,46	1777,62	21,40
U	10	25		14,00	0,00	1278,52	17,58	25,10	0,30	1796,98	12,90	15,20	0,40	1301,38	8,00
U	10	75		17,00	0,00	1667,81	8,85	50,00	0,00	3414,62	0,00	18,80	0,60	1902,21	18,31
U	20	25		15,00	0,00	1308,71	0,00	25,20	1,47	1722,74	34,03	16,00	0,00	1319,43	0,00
U	20	75		18,00	0,00	1826,45	0,00	36,60	0,49	2770,79	11,72	18,80	0,40	1921,80	48,65

Tabela 7.12 – Resultados para o grupo de instâncias baseadas em C203

Tipo de roteamento			Online		Periódico 1				Periódico 2					
													Instância	
TD	QP	GD	Veículos		Distância		Veículos		Distância		Veículos		Distância	
			\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s
N	10	25	5,00	0,00	694,58	0,00	4,40	0,49	736,13	4,80	4,10	0,30	693,97	2,74
N	10	75	7,30	0,46	932,59	14,85	9,00	0,00	1004,14	8,81	7,00	0,00	954,19	12,72
N	20	25	4,00	0,00	692,09	0,00	4,30	0,46	686,27	4,94	3,20	0,40	681,92	3,29
N	20	75	5,00	0,00	742,03	0,00	6,80	0,60	867,24	15,66	5,00	0,00	744,13	3,31
U	10	25	4,00	0,00	910,11	0,00	6,00	0,00	1077,58	0,00	4,40	0,49	930,91	3,76
U	10	75	7,00	0,00	1049,91	0,45	13,00	1,00	1519,85	33,37	6,10	0,30	1131,22	4,93
U	20	25	6,00	0,00	940,29	0,00	8,00	0,00	1032,38	9,38	6,00	0,00	940,29	0,00
U	20	75	7,20	0,40	1198,63	5,44	11,90	0,70	1496,98	25,62	10,00	0,00	1323,22	6,86

Tabela 7.13 – Resultados para o grupo de instâncias baseadas em R106

Tipo de roteamento			Online		Periódico 1				Periódico 2					
													Instância	
TD	QP	GD	Veículos		Distância		Veículos		Distância		Veículos		Distância	
			\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s
N	10	25	15,40	0,49	1305,68	13,30	16,90	0,54	1405,70	36,49	15,30	0,46	1308,04	17,87
N	10	75	17,80	0,60	1379,29	13,04	26,00	0,00	1832,86	1,86	18,10	0,30	1448,51	14,79
N	20	25	16,20	0,60	1333,64	16,38	16,70	0,46	1369,82	21,32	15,80	0,60	1353,61	16,17
N	20	75	19,60	0,66	1460,53	12,70	23,50	0,81	1670,32	15,04	19,40	0,49	1493,27	13,70
U	10	25	14,00	0,63	1313,42	23,30	18,70	1,10	1454,83	25,10	14,50	0,67	1353,78	21,08
U	10	75	17,00	0,77	1386,62	22,69	26,00	0,77	1781,16	32,98	16,80	0,75	1460,16	28,55
U	20	25	15,10	1,14	1301,95	18,89	17,20	1,33	1437,43	45,99	15,20	0,60	1326,38	23,01
U	20	75	16,60	0,66	1431,09	22,24	25,40	0,92	1776,07	30,69	17,40	0,66	1445,42	30,23

Tabela 7.14 – Resultados para o grupo de instâncias baseadas em R202

Tipo de roteamento			Online		Periódico 1				Periódico 2					
													Instância	
TD	QP	GD	Veículos		Distância		Veículos		Distância		Veículos		Distância	
			\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s
N	10	25	7,40	0,49	1056,69	3,02	7,56	0,53	1098,06	5,25	7,10	0,54	1064,65	5,65
N	10	75	8,90	0,30	1055,88	3,64	8,80	0,42	1113,82	2,65	8,50	0,81	1051,27	7,60
N	20	25	7,90	0,30	1062,38	1,67	8,00	0,47	1103,71	8,00	7,10	0,30	1069,64	7,69
N	20	75	8,00	0,89	1073,24	6,80	8,56	0,73	1128,94	7,99	8,50	0,50	1072,81	6,65
U	10	25	6,20	0,40	1160,67	3,78	9,00	0,00	1309,67	2,84	6,00	0,00	1182,98	19,48
U	10	75	6,20	0,40	1125,40	1,77	10,90	0,32	1461,78	9,14	6,00	0,00	1152,58	11,23
U	20	25	6,90	0,54	1176,96	13,32	9,00	1,25	1284,83	15,72	6,50	0,81	1170,78	10,20
U	20	75	7,00	0,00	1222,29	11,98	9,50	0,53	1512,21	8,47	6,70	0,46	1303,93	9,14

Tabela 7.15 – Resultados para o grupo de instâncias baseadas em RC104

Tipo de roteamento			Online		Periódico 1				Periódico 2					
Instância			Veículos		Distância		Veículos		Distância		Veículos		Distância	
TD	QP	GD	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s
N	10	25	12,80	0,87	1298,69	28,44	12,80	0,60	1345,35	28,60	12,70	0,78	1295,07	35,78
N	10	75	14,40	1,20	1392,60	28,56	16,90	0,30	1564,40	3,99	13,50	0,50	1418,03	25,10
N	20	25	12,10	0,54	1280,05	17,70	12,00	0,00	1296,77	11,13	11,80	0,40	1278,92	11,10
N	20	75	13,30	0,46	1349,48	14,70	14,30	0,90	1438,57	14,85	13,60	0,49	1384,72	16,82
U	10	25	12,20	0,40	1302,99	26,04	14,30	0,90	1400,41	34,62	12,10	0,54	1299,87	28,65
U	10	75	15,20	0,75	1464,15	31,42	23,70	1,42	1938,25	88,43	17,40	0,66	1575,34	33,61
U	20	25	12,70	0,64	1319,42	13,56	13,80	1,08	1367,10	36,53	12,80	0,75	1339,21	20,42
U	20	75	18,90	1,22	1673,21	71,08	23,90	1,76	1943,55	99,47	19,40	0,92	1698,59	48,82

Tabela 7.16 – Resultados para o grupo de instâncias baseadas em RC208

Tipo de roteamento			Online		Periódico 1				Periódico 2					
Instância			Veículos		Distância		Veículos		Distância		Veículos		Distância	
TD	QP	GD	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s	\bar{X}	s
N	10	25	4,80	0,40	810,10	5,25	4,80	0,60	844,33	15,64	4,80	0,40	814,53	11,93
N	10	75	4,20	0,40	768,33	1,08	4,00	0,00	811,51	0,00	4,50	0,50	786,45	10,35
N	20	25	4,80	0,40	789,24	9,58	5,10	0,30	816,88	17,01	4,80	0,60	812,29	16,16
N	20	75	5,00	0,00	796,92	0,30	5,00	0,00	813,87	0,00	4,20	0,40	798,03	6,42
U	10	25	5,00	0,00	939,07	1,42	8,00	0,45	1164,23	13,38	4,70	0,64	991,20	24,33
U	10	75	6,50	0,50	1112,89	11,68	10,10	0,30	1555,79	28,53	6,00	0,00	1138,08	48,30
U	20	25	4,80	0,60	934,46	44,29	5,90	1,04	1078,96	27,73	5,10	0,83	1005,30	33,80
U	20	75	5,80	0,40	1262,10	6,94	9,00	0,89	1534,83	76,26	5,60	0,49	1260,78	7,19

A única instância que o algoritmo de roteamento periódico 1 conseguiu uma média na distância percorrida menor que o algoritmo de roteamento *online* foi a C203, com TD normal, QP 20, e GD 25. Para todas as outras 47 instâncias, o algoritmo de roteamento *online* alcançou na média uma menor distância percorrida pelos veículos da frota veicular, se comparado ao periódico 1.

O algoritmo *online* obteve melhores médias em 38 cenários, de um total de 48. Em 10 casos, o periódico 2 obteve melhores médias se comparado aos algoritmos de roteamento *online* e periódico 1. Em nenhum cenário, o algoritmo de roteamento periódico 1 atingiu melhor média que o roteamento *online* e o roteamento periódico 2 simultaneamente.

7.3.3 Teste estatístico para avaliação da hipótese

Como descrito no Capítulo 4, foi aplicado uma Análise de Variância (ANAVA) sobre o Delineamento de Blocos Casualizados (DBC). Ambos são descritos na Seção 4.4 da metodologia da tese.

Eliminando o efeito causado pelos diferentes blocos (instâncias), com 99% de confiança, a hipótese H_0 da ANAVA foi rejeitada através do teste F , concluindo que os tipos de roteamento produzem médias estatisticamente diferentes ao resolver o PDRV nos diversos contextos. Esta conclusão foi extraída porque o valor p da ANAVA apresentou valor menor que a significância de 1% (0,01), indicando a região de rejeição para H_0 sobre o teste de hipótese F . Sendo assim, conclui-se que existe evidência estatística para aceitação da hipótese H_1 da Equação 4.1, onde é indicado que os algoritmos de roteamento *online*, periódico 1 e periódico 2 produzem médias populacionais estatisticamente diferentes.

Este resultado estatístico indica que a diferença entre as médias dos algoritmos de roteamento analisados não é em função do acaso, e sim em função do efeito dos tratamentos sobre o PDRV.

Para concluir sobre a hipótese central da tese, além do indicativo que os algoritmos de roteamento produzem médias diferentes, faz-se necessária a comparação das médias dos três tratamentos, para identificar qual deles produz resultados mais satisfatórios sobre o PDRV. Esta comparação pode ser feita através das distâncias médias percorridas, conforme a Tabela 7.17.

Tabela 7.17 - Avaliação média dos algoritmos de roteamento por grupos

Grupo de instância	<i>Online</i>	Periódico 1	Periódico 2
C101	1538,24	2157,10	1591,35
C203	895,03	1052,57	924,98
R106	1364,03	1591,02	1398,65
R202	1116,69	1251,63	1133,58
RC104	1385,07	1536,80	1411,22
RC208	926,64	1077,55	950,83
Geral	1204,28	1491,97	1235,12

Em todos os grupos de instâncias o roteamento *online* produziu resultados médios mais adequados se comparado aos algoritmos de roteamento periódicos. Esta informação, em conjunto com o resultado do teste de hipótese da ANAVA, produz indícios estatísticos para corroboração da hipótese central da tese.

7.3.4 Comparação entre os algoritmos de roteamento no PDRV

Esta seção apresenta uma comparação gráfica para os tipos de roteamento em análise nesta tese (periódico 1, periódico 2 e *online*), apresentando resultados agrupados de acordo com a característica do cenário simulado. As quatro características são:

- Instância base, referente a base de testes original de Solomon [1987];
- Tipo de distribuição dos consumidores dinâmicos no tempo (TD);
- Quantidade de períodos permitida (QP);
- Grau de dinamismo (GD).

Considerando o tipo de distribuição dos consumidores ao longo do dia de serviço, a primeira análise pode ser feita através da Figura 7.4, que considera dois grupos de instâncias na comparação entre os algoritmos de roteamento (cada um com 24 instâncias). Para cada um dos dois grupos analisados, a Figura 7.4 apresenta o percentual da distância que cada frota percorreu a mais quando usados os algoritmos de roteamento periódicos 1 e 2, se comparado ao algoritmo de roteamento *online*.

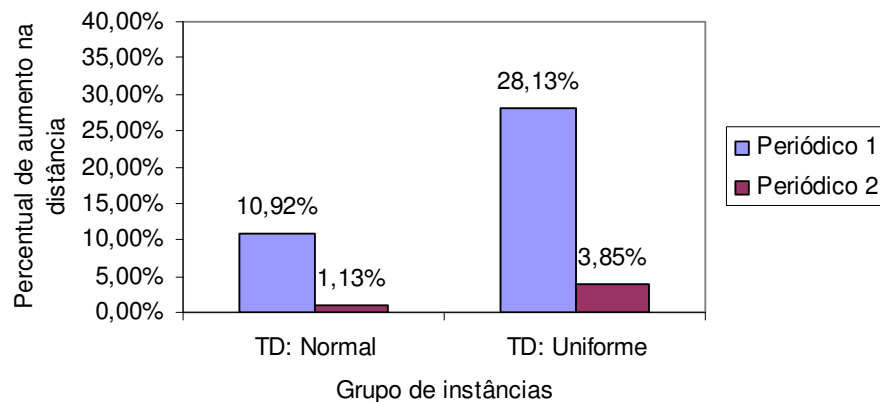


Figura 7.4 - Resultado sumarizado considerando o tipo de distribuição dos consumidores

De acordo com a Figura 7.4, o algoritmo de roteamento *online* é mais vantajoso quando os consumidores estão distribuídos uniformemente ao longo do dia. No caso da distribuição normal, que concentra os pedidos em um intervalo do dia, o algoritmo de roteamento *online* também obteve vantagem se comparado aos algoritmos de roteamento periódicos 1 e 2. O algoritmo periódico 2 se destaca sobre o algoritmo periódico 1 em ambos os cenários.

A Figura 7.5 apresenta análise semelhante a realizada na Figura 7.4, mas considerando a quantidade de períodos no uso dos algoritmos de roteamento periódicos.

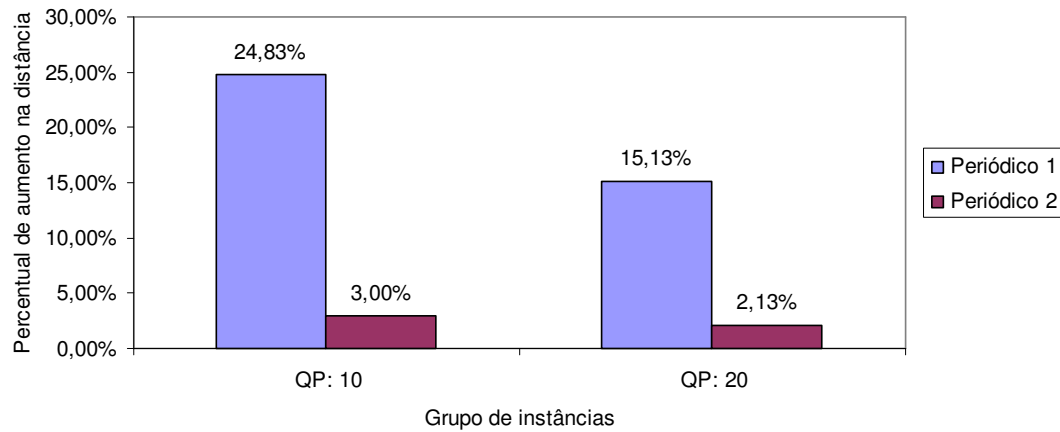


Figura 7.5 – Resultado sumarizado considerando a quantidade de períodos

Embora os algoritmos de roteamento periódicos sejam mais eficientes se realizados com uma frequência maior ao longo do dia, ambos não foram capazes de superar a qualidade do algoritmo de roteamento *online* na redução da distância total do PDRV.

A Figura 7.6 apresenta os resultados considerando o agrupamento de acordo com o grau de dinamismo das instâncias.

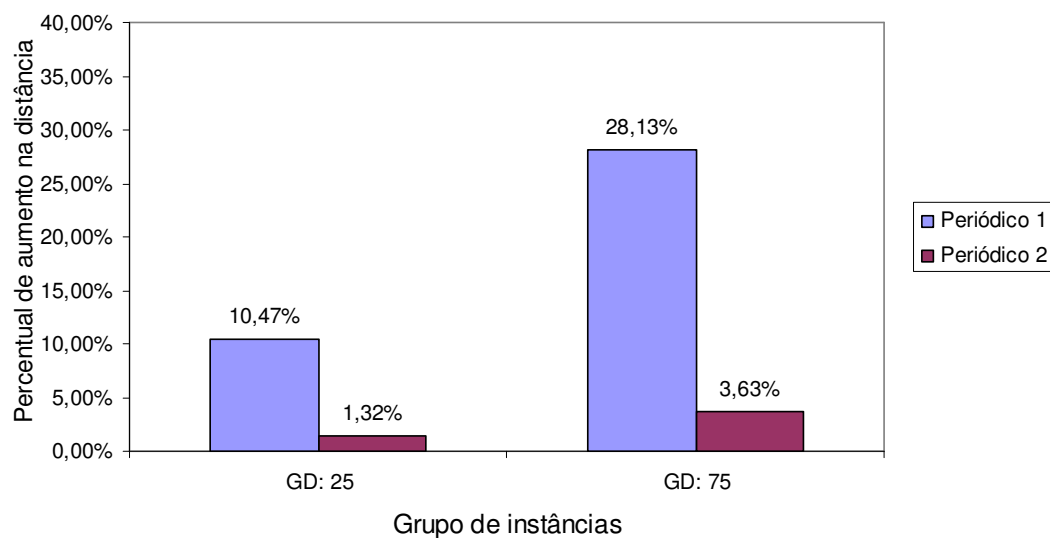


Figura 7.6 – Resultado sumarizado considerando o grau de dinamismo

De acordo com o resultado apresentado pela Figura 7.6, o algoritmo de roteamento *online* se adapta melhor aos ambientes onde existe uma maior quantidade

de consumidores dinâmicos, se comparado aos algoritmos de roteamento periódicos. Em ambos os contextos, o algoritmo de roteamento periódico 2 apresentou melhores resultados que o periódico 1.

A Figura 7.7 apresenta um comparativo para o ganho percentual médio na utilização do algoritmo de roteamento *online*, se comparado com os algoritmos de roteamento periódicos, agrupando os resultados por cada uma das seis instâncias utilizadas da base de testes de Solomon [1987].

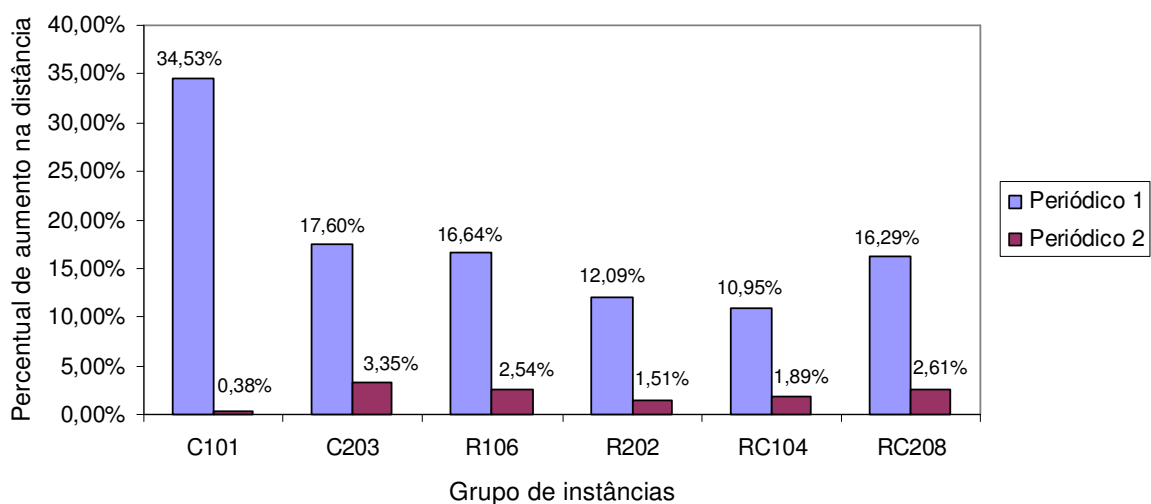


Figura 7.7 – Resultado sumarizado considerando a instância base de Solomon [1987]

Segundo a Figura 7.7, embora o algoritmo de roteamento *online* tenha se destacado em todos os cenários em relação aos algoritmos periódicos, este obteve mais ganhos nas seguintes configurações do ambiente, se comparado ao periódico 1:

- Consumidores em grupos geograficamente bem definidos, com tempos de serviço maiores e curtas janelas de tempo (C101);
- Consumidores espalhados aleatoriamente, com tempos de serviço mais curtos e longas janelas de tempo (R202).

Com relação ao periódico 2, os ganhos foram menores, mas seguindo relações semelhantes aos ganhos obtidos sobre o periódico 1, com exceção ao primeiro cenário, onde o comportamento de ambos apresentou resultados próximos.

Para uma análise mais ampla, foram agrupadas as instâncias que possuem a mesma distribuição de consumidores dinâmicos (TD), mesma quantidade máxima de períodos permitida (QP) e o mesmo grau de dinamismo (GD). Assim são considerados oito grupos, sendo que cada grupo possui 6 instâncias (uma para cada instância original de Solomon [1987]). Para estes grupos em análise, a Figura 7.8 apresenta o percentual médio do aumento da distância percorrida pela frota veicular, quando utilizado os algoritmos de roteamento periódicos, se comparado com a utilização do algoritmo de roteamento *online*.

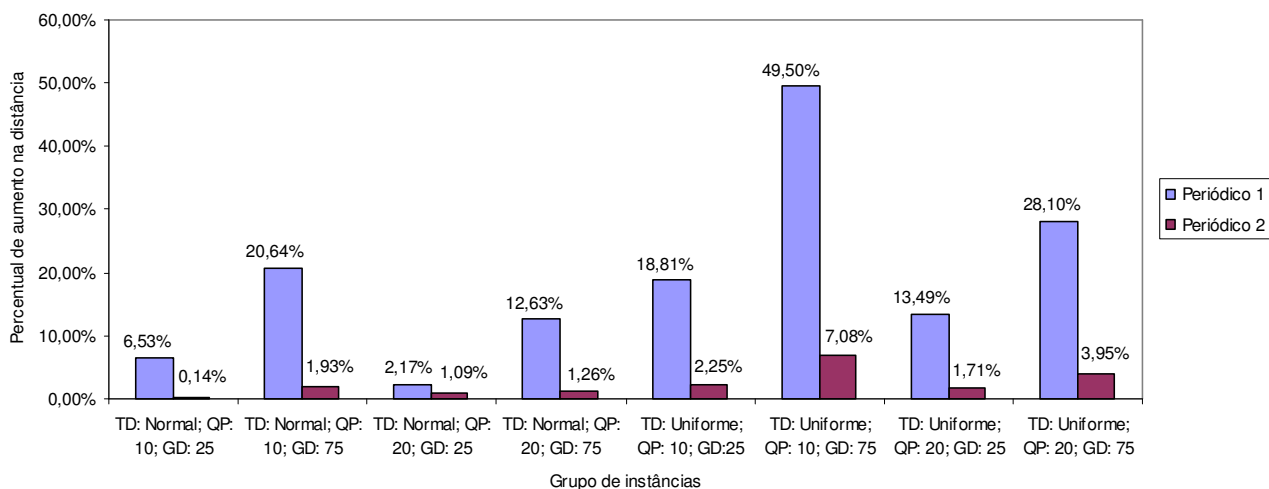


Figura 7.8 - Resultado sumarizado geral

A Figura 7.8 sumariza as conclusões que podem ser tiradas através de algumas análises anteriores desta seção. O cenário em que o algoritmo de roteamento periódico 1 mais se aproxima em qualidade do algoritmo de roteamento *online* é quando os algoritmos periódicos executam em tempos mais curtos (com 20 períodos), quando a quantidade de consumidores dinâmicos não é alta (25% no grau de dinamismo) e quando os pedidos são mais concentrados em uma faixa de tempo (distribuição normal). Já o cenário oposto é onde o algoritmo de roteamento *online* mais se destaca na redução da distância total viajada em relação ao periódico 1.

Em relação ao periódico 2, o algoritmo *online* mais se destaca sobre o cenário de distribuição de pedidos uniformes no tempo, com 10 períodos e grau de dinamismo em 75%. O cenário em que ambos apresentam resultados mais próximos é com distribuição normal de consumidores dinâmicos em relação ao tempo, com 10 períodos e grau de dinamismo em 25%.

7.3.5 Limite inferior no PDRV para as instâncias estendidas

Com o intuito de ter referência com relação a qualidade dos algoritmos de roteamento em análise, em função do melhor roteamento possível, esta seção apresenta limites inferiores e melhores soluções primais encontradas em uma adaptação do modelo matemático de Larsen [2001], apresentado na Seção 2.4.3, para a resolução do PRVJT.

No PDRV, um veículo não pode visitar um consumidor antes que o mesmo requisite serviço. Neste contexto, o modelo de Larsen [2001] foi adaptado neste trabalho. Além de não permitir a visitação antes do tempo de aparecimento do consumidor j (g_j), com o objetivo de melhorar a qualidade do limite inferior, as restrições adicionadas ao modelo de Larsen [2001] (equação 7.1) não permitem que um veículo qualquer k saia de um consumidor qualquer i , com destino a um consumidor qualquer j , antes que o consumidor j requisite o serviço, pois esta seria uma situação impossível na resolução do PDRV.

$$g_j x_{ijk} \leq s_{ik} + p_i \quad \forall i, j \in C; \quad \forall k \in V \quad (7.1)$$

Assim o *software* CPLEX 12.2.0.0 foi utilizado para resolver o PRVJT, considerando toda a demanda das instâncias do PDRV, mas não permitindo que consumidores dinâmicos fossem visitados antes de seus respectivos aparecimentos. Para as 48 instâncias estendidas, a Tabela 7.18 apresenta os limites inferiores e as melhores soluções primais encontradas pelo *software* de programação matemática, além de apresentar as melhores soluções encontradas pelos algoritmos de roteamento *online*, periódico 1 e periódico 2, para fins de comparação. Cada limite inferior representa a menor relaxação linear considerando todos os nós do *branch-and-bound* analisados pelo CPLEX. Foi utilizado um limite de tempo de 48 horas de processamento para a execução de cada uma das 48 instâncias no CPLEX.

Tabela 7.18 - Comparação com os limites inferiores

Tipo de roteamento				Roteamento dinâmico			CPLEX		
				<i>Online</i>	Periódico 1	Periódico 2	Solução Inteira (SI)	Limite Inferior (LI)	GAP
Características da instância				Distância	Distância	Distância	Distância	Valor	(SI-LI)/SI
Base	TD	QP	GD	Distância	Distância	Distância	Distância	Valor	(SI-LI)/SI
C101	Normal	10	25	1395,30	1562,33	1388,03	1276,79	1276,79	0,00%
	Normal	10	75	1868,05	2610,40	1872,02	1784,88	1770,10	0,83%
	Normal	20	25	1183,74	1208,94	1217,72	1150,30	1150,30	0,00%
	Normal	20	75	1742,95	2140,49	1740,65	1592,78	1592,78	0,00%
	Uniforme	10	25	1267,40	1766,67	1291,74	1230,19	1201,27	2,35%
	Uniforme	10	75	1657,01	3414,62	1868,34	1584,53	1548,74	2,26%
	Uniforme	20	25	1308,71	1671,55	1319,42	1220,28	1220,28	0,00%
	Uniforme	20	75	1826,45	2756,44	1813,52	1415,96	1415,96	0,00%
C203	Normal	10	25	694,58	732,21	693,03	688,56	593,38	13,82%
	Normal	10	75	910,71	1001,20	928,05	837,20	682,24	18,51%
	Normal	20	25	692,09	683,04	680,27	661,71	611,88	7,53%
	Normal	20	75	742,03	820,25	742,03	730,70	654,86	10,38%
	Uniforme	10	25	910,11	1077,58	928,02	829,61	647,64	21,93%
	Uniforme	10	75	1049,68	1486,48	1128,26	977,99	705,59	27,85%
	Uniforme	20	25	940,29	1027,69	940,29	813,94	655,28	19,49%
	Uniforme	20	75	1190,61	1480,63	1320,79	1099,84	893,83	18,73%
R106	Normal	10	25	1275,56	1334,36	1283,80	1285,17	988,85	23,06%
	Normal	10	75	1358,45	1828,75	1428,05	1321,61	1017,60	23,00%
	Normal	20	25	1300,23	1342,92	1325,48	1299,79	982,87	24,38%
	Normal	20	75	1441,43	1651,63	1478,85	1373,37	1058,39	22,93%
	Uniforme	10	25	1290,40	1393,01	1303,24	1329,70	974,27	26,73%
	Uniforme	10	75	1357,16	1745,32	1413,24	1344,67	1017,46	24,33%
	Uniforme	20	25	1264,08	1384,06	1292,38	1259,95	963,22	23,55%
	Uniforme	20	75	1387,38	1701,11	1411,16	1303,02	981,58	24,67%
R202	Normal	10	25	1051,85	1091,32	1055,99	1052,53	806,05	23,42%
	Normal	10	75	1052,46	1108,97	1040,38	1009,85	807,68	20,02%
	Normal	20	25	1059,47	1094,37	1063,19	1053,64	803,00	23,79%
	Normal	20	75	1062,33	1118,89	1062,72	1017,67	795,26	21,85%
	Uniforme	10	25	1155,67	1307,67	1154,98	1119,90	867,88	22,50%
	Uniforme	10	75	1122,75	1458,90	1130,93	1066,22	837,43	21,46%
	Uniforme	20	25	1159,67	1256,49	1149,85	1131,16	858,28	24,12%
	Uniforme	20	75	1210,31	1504,17	1294,83	1124,25	933,72	16,95%
RC104	Normal	10	25	1268,07	1300,79	1242,78	1227,02	767,24	37,47%
	Normal	10	75	1350,26	1556,38	1384,12	1354,49	825,18	39,08%
	Normal	20	25	1258,48	1280,06	1259,62	1237,47	749,39	39,44%
	Normal	20	75	1336,59	1419,70	1361,15	1372,16	770,59	43,84%
	Uniforme	10	25	1269,70	1332,08	1261,68	1231,65	749,78	39,12%
	Uniforme	10	75	1411,11	1819,81	1532,58	1348,49	809,48	39,97%
	Uniforme	20	25	1300,45	1316,21	1299,38	1255,08	741,98	40,88%
	Uniforme	20	75	1636,00	1807,52	1642,76	1357,98	802,24	40,92%
RC208	Normal	10	25	805,16	819,20	801,88	784,33	627,61	19,98%
	Normal	10	75	767,79	811,51	775,12	742,90	629,88	15,21%
	Normal	20	25	772,22	790,20	789,51	769,17	627,31	18,44%
	Normal	20	75	796,55	813,87	784,58	734,43	617,30	15,95%
	Uniforme	10	25	937,02	1138,57	945,64	894,55	630,44	29,52%
	Uniforme	10	75	1096,79	1512,31	1068,76	927,12	678,08	26,86%
	Uniforme	20	25	892,78	1050,91	961,73	854,13	656,47	23,14%
	Uniforme	20	75	1246,05	1454,31	1251,97	988,29	689,45	30,24%

Os algoritmos de roteamento utilizados durante as simulações do PDRV (*online*, periódico 1 e periódico 2), não possuíam as mesmas informações como o

CPLEX ao iniciar a alocação da frota veicular. Eles tiveram acesso as informações sobre o problema apenas após a última requisição dinâmica (ao longo do dia de serviço). É fato que se existisse um algoritmo que resolvesse todos os PRVJT com Múltiplos Depósitos adaptados ao longo do dia, de forma ótima, em um tempo infinitamente pequeno, não necessariamente este algoritmo de roteamento idealizado alcançaria a melhor solução possível ao resolver o PDRV. Isso ocorre porque o processo está sujeito ao acaso, pois um veículo mal alocado em um instante preliminar (com uma solução não-ótima), pode no futuro ter a sorte de estar perto de uma nova requisição dinâmica, o que não necessariamente acontecerá com um veículo que foi programado com uma sequência de opções ótimas em instantes passados. Como o *software* de programação matemática acessa todas as informações *a priori*, mas adiciona apenas restrições referentes às visitas de consumidores dinâmicos, é impossível que um sistema de roteamento consiga soluções melhores às apresentadas pela coluna do limite inferior da Tabela 7.18.

Das 48 instâncias estendidas para o PDRV, a resolução do modelo matemático adaptado alcançou a solução ótima em 5 casos, destacados em negrito na Tabela 7.18.

Para melhorar a qualidade dos algoritmos de roteamento no PDRV, métodos de previsão podem ser utilizados, pois, se bem realizados, podem diminuir o ruído gerado pela aleatoriedade na realização dos pedidos dinâmicos. Estas previsões podem ser realizadas através de informações estocásticas, como apresentado nas Seções 2.6 e 3.4 sobre o PRVE.

7.4 Considerações finais

Este capítulo apresentou os resultados do algoritmo proposto nesta tese para a resolução do PRVJT estático. Esta apresentação teve o intuito de mostrar a qualidade do algoritmo utilizado nos algoritmos de roteamento do PDRV.

Posteriormente, foram apresentados os resultados comparativos, entre as abordagens *online* e periódicas, necessários para as conclusões sobre a hipótese da tese. Na sequência, a conclusão estatística também foi apresentada, corroborando a hipótese da tese, que afirma que o algoritmo de roteamento *online* pode ser mais eficaz que versões periódicas de roteamento.

Ao final, foi apresentado um novo modelo matemático para encontrar limites primais para instâncias do PDRV. Tais limites foram confrontados com os resultados dos algoritmos de roteamento em análise para fins de comparação.

Capítulo 8

Conclusões

Relacionada à resolução do Problema Dinâmico de Roteamento de Veículos (PDRV), esta tese levantou a seguinte pergunta:

Durante a resolução do PDRV, seria mais vantajoso reagir ao ambiente rapidamente quando novos pedidos são requisitados, ao invés de esperar instantes pré-programados com tempos mais longos para processamento?

Em uma análise inicial, pode parecer interessante responder ao ambiente em períodos pré-agendados (abordagem periódica), e com tempos maiores para processamento, se comparado à abordagem mais rápida (*online*). Neste contexto, pode-se enumerar:

- Possuir um tempo maior para o processamento das requisições, juntamente com as requisições ainda não atendidas pelos veículos, pode ser interessante do ponto de vista da qualidade do algoritmo de roteamento, pois o PRV é classificado como NP-Difícil;
- Considerar problemas maiores (através do acúmulo de pedidos) cria a possibilidade de encontrar soluções mais robustas nos algoritmos de roteamento.

Considerando estas vantagens, alguns trabalhos na literatura resolvem o PDRV com o algoritmo de roteamento periódico como, por exemplo, Bent e Van Hentenryck [2004], Chen e Xu [2006] e Hvattum *et al.* [2007].

Na tentativa de responder à pergunta central da tese, este trabalho descreveu a hipótese *H* para ser analisada:

H: Utilizando um competitivo algoritmo para alocação de veículos, o roteamento online para atendimento de novos pedidos no PDRV pode ser estatisticamente mais eficaz se comparado ao roteamento periódico.

Para analisar a hipótese, foi construído um simulador para o PDRV capaz de emular um dia de serviço considerando as principais variáveis do problema. Além do simulador, esta tese descreveu um algoritmo híbrido que mescla a utilização de um *software* de programação matemática para resolução de um problema linear inteiro com uma meta-heurística denominada *Simulated Annealing*. Foram também criados 48 cenários distintos (instâncias) para a avaliação da hipótese, aos quais os três diferentes algoritmos de roteamento (*online*, periódico 1 e periódico 2) foram submetidos. O roteamento *online* tem como característica principal sua rápida reação quando novos pedidos são realizados. O algoritmo de roteamento periódico 1, ao invés de reagir rapidamente, acumula os pedidos e executa um procedimento de otimização mais longo, se comparado ao roteamento *online* (semelhante aos algoritmos de roteamento utilizados nos trabalhos de Bent e Van Hentenryck [2004], Chen e Xu [2006] e Hvattum *et al.* [2007]). Já o algoritmo de roteamento periódico 2 também acumula pedidos, mas executa um procedimento de otimização mais rápido, assim como a versão *online*.

Antes de avaliar a qualidade dos algoritmos de roteamento no PDRV, com o intuito de ter confiança na qualidade do algoritmo que resolve o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT) estático, um Algoritmo Híbrido, adaptável ao roteamento dinâmico, foi proposto neste trabalho e aplicado nas 56 instâncias de Solomon [1987]. A robustez do algoritmo foi atestada através de comparações com os melhores trabalhos da literatura na redução da distância total do PRVJT.

Após a apresentação dos resultados do Algoritmo Híbrido sobre uma versão estática do problema, esta tese apresentou os resultados da resolução do PDRV, com o intuito de analisar a hipótese *H*. Através do experimento e de um teste de hipótese na análise estatística, pôde-se afirmar que o roteamento mais rápido (*online*) pode ser mais eficaz se comparado às versões periódicas de roteamento. Assim, existe evidência empírica que verifica a existência de uma diferença estatisticamente significativa entre os três tipos de roteamento analisados, corroborando a hipótese da tese.

Em alguns cenários, a diferença entre os três algoritmos de roteamento apresentou-se mais acentuada. Neste contexto, pode-se destacar um conjunto de situações onde o algoritmo de roteamento *online* obteve ganhos mais significativos:

- Cenários com alto grau de dinamismo;

- Cenários onde as requisições dinâmicas não estão concentradas em determinado período do dia;
- Cenários com a junção das seguintes características:
 - Consumidores em grupos geograficamente bem definidos, tempos de serviço maiores, e curtas janelas de tempo;
 - Consumidores espalhados aleatoriamente, tempos de serviço mais curtos, e longas janelas de tempo;

Embora o algoritmo de roteamento *online* tenha se destacado nos cenários descritos, o mesmo obteve ganhos sobre os dois algoritmos de roteamento periódicos em todos os grupos de instâncias analisados.

Com os resultados desta tese, fica explícita a importância para um algoritmo de roteamento mais rápido, como aqui nomeado roteamento *online*. Aliado aos sistemas de roteamento que se preocupam em responder mais rapidamente às requisições dinâmicas, informações estocásticas podem ser utilizadas na tentativa de prever onde, e quando novas requisições poderão surgir. Esta combinação pode melhorar a qualidade dos algoritmos que resolvem o PDRV. Além de informações estocásticas, futuras pesquisas podem também explorar a relação dos algoritmos de roteamento *online* com a técnica de espera programada.

Referências bibliográficas

- Alvarenga, G. B. *Um algoritmo híbrido para os problemas de Roteamento de Veículos Estático e Dinâmico com Janela de Tempo*. Departamento de Ciência da Computação. Universidade Federal de Minas Gerais, 2005
- Alvarenga, G., Mateus, G. & de Tomi, G. *A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows*. *Computers & Operations Research*, 2007, Vol. 34, pp. 1561-1584
- Aquino, L. H. de. *Estatística Experimental*. Universidade Federal de Lavras, Lavras, 1991.
- Assis, L. P. *Algoritmos para o problema de roteamento de veículos com coleta e entrega simultâneas*. Dissertação de Mestrado. Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, 2007
- Backer, B. & Furmon, V. *Meta-heuristics in Constraint Programming Experiments with Tabu Search on the Vehicle Routing Problem*. Second International Conference on Metaheuristics (MIC'97). 1997
- Bent, R. W. & Hentenryck, P. V. *Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers*. *Operations Research, INFORMS*, 2004, Vol. 52(6), pp. 977-987
- Branke, J., Middendorf, M., Noeth, G. & Dessouky, M. *Waiting Strategies for Dynamic Vehicle Routing*. *Transportation Science, INFORMS*, 2005, Vol. 39(3), pp. 298-312
- Braun, D., *Comércio eletrônico fatura R\$ 6,7 bilhões e cresce 40% no 1º semestre*. *Jornal O Globo*. 10 de agosto de 2010. Endereço: <http://oglobo.globo.com/economia/mat/2010/08/10/comercio-eletronico-fatura-6-7-bilhoes-cresce-40-no-1-semester-917366187.asp>. Acessado em 07 de março de 2011.
- Cavalcanti, P. P. *Experimental Designs: um pacote R para análise de experimentos*. Trabalho de Conclusão de Curso do curso de Matemática. Universidade Federal de Alfenas, 2010, pp. 90.
- Chen, Z.-L. & Xu, H. *Dynamic Column Generation for Dynamic Vehicle Routing with Time Windows*. *Transportation Science, INFORMS*, 2006, Vol. 40(1), pp. 74-88
- Cook, S. A. *The complexity of theorem-proving procedures* *Proceedings of the third annual ACM symposium on Theory of computing*, ACM, 1971, pp. 151-158

- Cordeau, J. F., Laporte, G., Hautes, É., Commerciales, É. & Gerad, L. C. D. *A unified Tabu Search heuristic for vehicle routing problems with time windows*. Journal of the Operational Research Society, 2001, Vol. 52, pp. 928-936
- Cormen, T. H., Leiserson, C.E., Rivest, R.L. & Stein, C. *Introduction to Algorithms*. The MIT Press, 2001
- Dantzig, G. B. *Discrete-Variable Extremum Problems*. Operations Research, 1957, Vol. 5(2), pp. 266-288
- Dantzig, G. B. & Ramser, J.H. *The Truck Dispatching Problem*. Management Science, 1959, Vol. 6, pp. 80-91
- Dorigo, M., Maniezzo, V. & Colorni, A. *The Ant System: Optimization by a Colony of Cooperating Agents*. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 1996, Vol. 26, pp. 29-41
- Eiben, A. E. & Smith, J.E. *Introduction to Evolutionary Computing*. SpringerVerlag, 2003
- Fraga, M. C. P. *Uma Metodologia Híbrida: Colônia de Formigas – Busca Tabu – Reconexão por Caminhos para Resolução do Problema de Roteamento de Veículos com Janela de Tempo*. Dissertação de Mestrado. CEFET-MG, 2007
- Franco, E. F. & Oliveira, H. C. B. de; *Adaptação da Meta-Heurística GRASP na Resolução do Problema de Roteamento de Veículos com Janela de Tempo* Pesquisa Operacional para o Desenvolvimento, 2011
- Freitas, L. M. B.; Montané, F. A. T.; *Metaheurísticas VNS-VND e GRASP-VND para Problema de Roteamento de Veículos com Coleta e Entrega Simultânea*, XV Simpósio de Pesquisa Operacional e Logística da Marinha, Rio de Janeiro, 2008
- Garey, M. & Johnson, S. *Computer Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979
- Gendreau, M., Guertin, F. & Potvin, J.Y. *Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries*. Transportation Research, 2006, Vol. C, pp. 157-174
- Gendreau, M., Guertin, F., Potvin, J.-Y. & Taillard, E. *Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching*. Transportation Science, INFORMS, 1999, Vol. 33(4), pp. 381-390
- Gendreau, M., Potvin, J.Y. & Laport, G. *Dynamic vehicle routing and dispatching*. Fleet Management and Logistic, 1998, pp. 115-226
- Goldberg, M. C. & Luna, H. P. L. *Otimização Combinatória e Programação Linear -- Modelos e algoritmos*. Editora Campus, 2000

- Hvattum, L. M.; Løkketangen, A. & Laporte, G. A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems. *Networks*, 2007, 49, 330-340
- Hvattum, L. M., Løkketangen, A. & Laporte, G. *Solving a Dynamic and Stochastic Vehicle Routing Problem with a Sample Scenario Hedging Heuristic*. *Transportation Science*, INFORMS, 2006, Vol. 40(4), pp. 421-438
- Hvattum, L. M., Løkketangen, A. & LAPORTE, G. *A heuristic solution method to a stochastic vehicle routing problem*. TRISTAN V – The Fifth Triennial Symposium on Transportation Analysis, 2004
- Ichoua, S., Gendreau, M. & Potvin, J.-Y. *Vehicle dispatching with time-dependent travel times*. *European Journal of Operational Research*, 2003, Vol. 144(2), pp. 379-396
- Ichoua, S., Gendreau, M. & Potvin, J.-Y. *Diversion Issues in Real-Time Vehicle Dispatching*. *Transportation Science*, INFORMS, 2000, Vol. 34(4), pp. 426-438
- Ichoua, S., Gendreau, M. & Potvin, J.-Y. *Exploiting Knowledge About Future Demands for Real-Time Vehicle Dispatching*. *Transportation Science*, INFORMS, 2006, Vol. 40(2), pp. 211-225
- Jaillet, P. *Probabilistic traveling salesman problems*. Massachusetts Institute of Technology (MIT), 1985
- Jung, S. & Moon, B.R. *A Hybrid Genetic Algorithm For The Vehicle Routing Problem With Time Windows*. *Genetic and Evolutionary Computation Conference (GECCO)*. Morgan Kaufmann, 2002, pp. 1309-1316
- Kilby, P., Prosser, P. & Shaw, P. *Dynamic VRPs: A study of scenarios*. Technical Report APES-06-1998, School of Computer Science, University of St. Andrews, St. Andrews, Scotland., 1998
- King, G. F. & Mast, C. F. *Excess travel: Causes, extent and consequences*. *Transportation Res.Record*, 1997, pp. 126-134
- Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. *Optimization by simulated annealing*. *Science*, 1983, Vol. 220, pp. 671-680
- Larsen, J. *Parallelization of the Vehicle Routing Problem with Time Windows*. Ph.D. Thesis. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2011
- Lima, P. C.; Abreu, A. R. de. *Estatística Experimental: ensaios balanceados*. FAEPE - Universidade Federal de Lavras, Lavras, 2000

- Lorini, S.; Potvin, J.-Y. & Zufferey, N. *Online vehicle routing and scheduling with dynamic travel times*. Computers & Operations Research, 2011, Vol. 38, pp. 1086-1090
- Mao, Y. & Deng, Y. Solving Vehicle Routing Problem with Time Windows with Hybrid Evolutionary Algorithm. *Second WRI Global Congress on Intelligent Systems*. 2010.
- Mingozzi, A., Boschetti, M.A., Ricciarde, S. & Bianco, L. *A Set Partitioning Approach to the Crew Scheduling Problem*. Oper. Res., INFORMS, 1999, Vol. 47(6), pp. 873-888
- Mitrović-Minić, S. & Laporte, G. *Waiting strategies for the dynamic pickup and delivery problem with time windows*. Transportation Research Part B, 2004, Vol. 38, pp. 635-655
- Montemanni, R., Gambardella, L.M., Rizzoli, A.E. & Donati, A.V. *Ant Colony System for a Dynamic Vehicle Routing Problem*. Journal of Combinatorial Optimization, 2005, Vol. 10(4), pp. 327-343
- Nance, R. E. The time and state relationships in simulation modeling. *Communications of the ACM*, 1981, 24, 173-179
- Oliveira, H. C. B. de, Vasconcelos, G. C., Alvarenga, G. B., Mesquita, R. V. & de Souza, M. M. *A Robust Method for the VRPTW with Multi-Start Simulated Annealing and Statistical Analysis*. IEEE Symposium on Computational Intelligence in Scheduling. IEEE Computer Society Press, 2007, pp. 198-205
- Oliveira, H. C. B. de; Rocha, G. M.; de Souza, M. M.; Cicon, L. A.; Borges, V. R.; Mateus, G. R. *A vehicular waiting time heuristic for dynamic vehicle routing problem* *Proceedings of the 2008 ACM symposium on Applied computing*, ACM, 2008, pp. 13-17
- Oliveira, H. C. B. de & Vasconcelos, G. C. *A hybrid search method for the vehicle routing problem with time windows*. Annals of Operations Research, Springer Netherlands, 2010, 180, 125-144.
- Papadimitriou, C. H. & Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, Paperback, 1998
- Pimentel-Gomes, F. *Curso de Estatística Experimental*. 15ª edição. FEALQ, Piracicaba, 2009
- Pureza, V. M. M. & Laport, G. *Estratégias de Programação de Veículos e Pedidos para Problemas Dinâmicos de Coleta e Entrega com Janelas de Tempo*. XL SBPO - Simpósio Brasileiro de Pesquisa Operacional. Sociedade Brasileira de Pesquisa Operacional, 2008

- Regan, A. C. *Real-time information for improved efficiency of commercial vehicle operations*. The University of Texas at Austin, 1997
- Resende, M. G. C. Greedy *randomized adaptive search procedures*. *Journal of Global Optimization*, 1995, 6, 109-133
- Ribas, S.; Subramanian, A.; Coelho, I. M.; Ochi, L. S.; Souza, M. J. F.; *A hybrid algorithm for the Vehicle Routing Problem with Time Windows*. International Conference on Industrial Engineering and Systems Management. 2011. pp. 1-10
- Riise, A. & Stølevik, M. *Implementation of Guided Local Search for the Vehicle Routing Problem* SINTEF, 1999 (STF42 A99013)
- Rochat, Y. & Taillard, R. D. *Probabilistic diversification and intensification in local search for vehicle routing*. *Journal of Heuristics*, 1995, Vol. 1, pp. 147-167
- Santos, A. G. *Método de geração de colunas e meta-heurísticas para alocação de tripulação*. Departamento de Ciência da Computação. Universidade Federal de Minas Gerais. 2008
- Secomandi, N. *Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands*. *Computers and Operations Research*, Elsevier Science Ltd., 2000, Vol. 27(11-12), pp. 1201-1225
- Silva, F. A. das D. Um Modelo de simulação de processos de software baseado em conhecimento para o ambiente PROSOFT. *Universidade Federal do Rio Grande do Sul. Instituto de Informática. Dissertação de Mestrado*, 2001
- Solomon, M. M. *Algorithms for the vehicle routing and scheduling problems with time window constraints*. *Operations Research, INFORMS*, 1987, Vol. 35(2), pp. 254-265
- Souza, M. M. de. Uma Metodologia de Predição Estatística de Projetos Baseada em Simulação. *Dissertação de Mestrado. Centro de Informática. Universidade Federal de Pernambuco*, 2007
- Steiner, M. T. A., Zamboni, L. V. S., Costa, D. M. B., Carnieri, C. & da Silva, A. L. *O Problema de Roteamento no Transporte Escolar*. *Pesquisa Operacional*, 2000, Vol. 20(1), pp. 83-99
- Thangiah, S. R., Osman, I. H. & Sun, T. *Hybrid Genetic Algorithm Simulated Annealing and Tabu Search Methods for Vehicle Routing Problem with Time Windows*. Technical Report 27. Computer Science Department. Slippery Rock University, 1994
- Waters, C. D. J. *Vehicle routing problems with uncertainty and omitted customers*. *Journal of the Operational Research Society*, 1989, Vol. 40, pp. 1099-1108

Xu, H., Chen, Z.-L., Rajagopal, S. & Arunapuram, S. *Solving a Practical Pickup and Delivery Problem*. *Transportation Science*, INFORMS, 2003, Vol. 37(3), pp. 347-364

Apêndice A

Detalhes sobre a implementação e manutenção do Conjunto R' para o PCC

A Caixa 2 da Figura 6.1 (página 68) sempre utiliza todas as colunas viáveis encontradas, geradas pelo Algoritmo Híbrido, armazenadas em uma tabela *hash* que possui a funcionalidade de evitar que colunas repetidas sejam adicionadas. Uma dificuldade encontrada foi a criação de um método capaz de gerar um código *hash* exclusivo para cada rota distinta possível.

Obviamente, se existe uma quantidade exponencial de rotas possíveis, então é necessária uma variável capaz de comportar longos números inteiros, sendo que cada um destes números representaria um código *hash* para sua coluna correspondente.

Se for considerado que um PRV necessita de apenas um veículo para atender toda sua demanda, então o PRV é equivalente ao PCV. Considerando o conjunto C como o conjunto de todos os consumidores a serem visitados, e n sendo a cardinalidade do conjunto C , já eliminando permutações que produzem mesmo Ciclo Hamiltoniano, a quantidade de rotas distintas no PCV é descrita na Equação A.1, que representa o tamanho do espaço de busca S do PCV ($|S_{PCV}|$).

$$|S_{PCV}| = \left\lfloor \frac{n!}{2n} \right\rfloor = \left\lfloor \frac{(n-1)!}{2} \right\rfloor \quad (A.1)$$

$|S_{PCV}|$ vai além do que um computador moderno pode enumerar em um código *hash* com seus números inteiros de 4 bytes, caso precise atender uma demanda de médio ou grande porte. No caso do PRV, a quantidade de rotas distintas é muito maior que no PCV, pois existem rotas que não atendem todos os consumidores, já que estes outros são atendidos por outros veículos. Assim, a quantidade de rotas distintas no PRV é dada pela Equação A.2, em que n_i representa a quantidade de elementos do i -ésimo subconjunto do conjunto potência $P(C)$.

$$|S_{PRV}| = \sum_{i \in P(C)} \left\lceil \frac{(n_i - 1)!}{2} \right\rceil \quad (A.2)$$

Dada tal limitação, foi proposto, ao longo deste trabalho, um método alternativo que reduz a quantidade de códigos *hash* necessários para representar todas as rotas possíveis.

Uma coluna do PPC não distingue diferentes ciclos hamiltonianos, para determinado subconjunto de consumidores atendidos. Na formulação matemática do PPC, a coluna indica apenas se o consumidor foi ou não atendido por aquela rota, sem considerar a ordem de atendimento. Considere duas rotas distintas (r_1 e r_2), que atendem o mesmo subconjunto de consumidores. Estas podem possuir o mesmo código *hash*. Supondo que r_1 possui uma menor distância percorrida, se comparada com r_2 , então r_2 não fará parte da solução ótima do PPC, se ambas coexistirem no mesmo PPC. Isso devido à sua atribuição no vetor de custos do modelo de programação linear inteira.

Desta forma, sempre que uma coluna é submetida ao armazenamento na tabela *hash*, previamente é verificado se existe alguma coluna de mesmo código *hash* na tabela. Se não existir, é porque até então, nenhum ciclo hamiltoniano com aquele subconjunto de consumidores foi encontrado, e assim a coluna é adicionada automaticamente. Se existir outra coluna de mesmo código *hash*, os custos das colunas que atendem os mesmos consumidores são comparados. Se a nova coluna possui custo inferior ao custo da coluna já adicionada anteriormente, então a antiga é removida, e a nova é adicionada, pois esta representa um ciclo hamiltoniano de maior qualidade, se comparada à antiga coluna. Outra possibilidade é a nova coluna possuir custo maior se comparada à coluna já presente na tabela *hash*. Neste caso, a nova coluna é descartada pois matematicamente não fará parte de uma solução ótima do PPC concorrendo com o outro ciclo hamiltoniano de custo inferior. Se possuírem custos iguais, então a nova coluna também é descartada.

Eliminando tais permutações, a quantidade de códigos *hashes* necessários no PPC ($|S_{PPC}|$) é reduzida para a quantia determinada pela Equação A.3.

$$|S_{PPC}| = |P(C)| = 2^{|C|} \quad (A.3)$$

Esta redução também é interessante para evitar o estouro da memória *heap* destinada ao processo pelo sistema operacional. Assim, uma quantidade reduzida de

rotas é armazenada na tabela *hash*. Vale ressaltar que, em problemas de médio e grande porte, tal redução não possibilita a representação de um código *hash* distinto para cada rota possível, pois o crescimento é exponencial em função da quantidade de consumidores do problema. Suponha que o PRV possui 100 consumidores para atendimento. A quantidade de códigos *hashes* necessários no PPC seria igual a 2^{100} . A linguagem de programação utilizada neste trabalho limita o código *hash* a um inteiro de 4 *bytes*. Assim, o inteiro é capaz de representar 2^{32} códigos *hashes* distintos. Apesar do choque de códigos *hashes*, nenhuma irregularidade foi encontrada durante os experimentos, possivelmente porque o PRVJT é um problema muito restritivo, ou seja, das 2^{100} rotas possíveis, a grande maioria não é computada por violar restrições de capacidade ou tempo.

Dada a redução na quantidade de colunas, podemos observar a relação da Equação A.4 que facilita a aplicação do modelo do PPC na resolução do PRVJT.

$$|S_{PPC}| < |S_{PCV}| < |S_{PRV}| \quad (A.4)$$

Outra preocupação, não menos importante que a complexidade de memória exigida pelo PPC, é também a complexidade de tempo para sua implementação. Ou seja, como gerar códigos *hash* eficientemente para cada subconjunto possível do PRV representado pelas rotas/colunas nos Algoritmo Híbridos.

A solução apresentada neste trabalho de tese possui probabilidades relacionadas com o choque de *hashes* de rotas distintas. Tal risco foi inevitável para gerar códigos *hash* rapidamente, já que tal método é invocado sempre que uma nova coluna é gerada pelos algoritmos híbridos. O método é aqui descrito com dois algoritmos. O Algoritmo 5 é invocado uma única vez, quando a demanda do PRVJT é conhecida. Este tem como objetivo gerar números que auxiliarão a identificar ciclos hamiltonianos que atendem os mesmos consumidores ao longo da otimização. Já o Algoritmo 6 calcula o código *hash* de uma rota específica.

Algoritmo 5 - Geração de números auxiliares para cálculo do código hash

```

1: begin
2:   int valorMaximo = MAX_INT / numeroTotalDeConsumidores();
3:   for i = 0 until i < numeroTotalDeConsumidores()
4:     aux[i] = inteiroAleatorio(0, valorMaximo);
5:   end loop
6: end

```

O primeiro passo do Algoritmo 5 é limitar o número máximo auxiliar possível que representará parte do código *hash* de cada consumidor (linha 2). Esta limitação faz sentido quando é analisada em conjunto com o Algoritmo 6. Ela tem o objetivo de impedir o *overflow* da variável *codigoHash* no Algoritmo 6 (linha 5), onde são efetuadas sucessivas somas utilizando o vetor *aux*, que recebe valores limitados pela variável *valorMaximo* no Algoritmo 5 (linha 4). O valor para cada consumidor no vetor *aux* é dado por um número aleatório proveniente de distribuição uniforme entre 0 e *valorMaximo*. Isso significa que, mesmo que cada consumidor receba no vetor *aux* o valor máximo permitido, não haverá possibilidade de *overflow* no Algoritmo 6, devido à limitação do valor máximo a ser sorteado na distribuição uniforme.

O Algoritmo 6 soma todos os valores auxiliares dos consumidores presentes no ciclo hamiltoniano da rota que se deseja conhecer o código *hash*. Este somatório representa, ao final, o código *hash* da rota em questão.

Algoritmo 6 - Cálculo do código *hash* de uma rota

```

1: parameter: rota; //rota que deseja calcular o código hash
2: begin
3:   int codigoHash = 0; //variável de retorno da função
4:   for i = 0 until i < rota.numeroDeConsumidores( )
5:     codigoHash = codigoHash + aux[rota.consumidor(i)];
6:   end loop
7:   return codigoHash; //retornando código hash da rota
8: end

```

Obviamente, que se no Algoritmo 5 na linha 2, for sorteado o mesmo código auxiliar para dois ou mais consumidores distintos, haverá problemas constantes de existirem rotas com ciclos hamiltonianos distintos com o mesmo código *hash*. Isso prejudicaria o PPC, que perderia colunas antes mesmo da sua execução. Visando contornar este problema, a implementação real do Algoritmo 5 verifica se o número sorteado para o consumidor *i* atual já foi sorteado para algum consumidor anterior. Caso positivo, outro número da distribuição uniforme é selecionado. Isso não evita todas as falhas do algoritmo, mas reduz sua chance de errar. Aqui, foi mostrado um algoritmo simplificado para facilitar a compreensão.

Vale ressaltar que, quanto maior a quantidade de consumidores a ser atendida, maior é a probabilidade do algoritmo ter choques de códigos *hash*. Pois, quanto maior a quantidade de consumidores, menor será o intervalo de números possíveis a serem sorteados na linha 2 do Algoritmo 5.

Utilizando a notação θ , utilizada para medir a complexidade de um algoritmo, o Algoritmo 6 possui complexidade polinomial e linear $\theta(n)$, onde n é o número de consumidores a serem atendidos. Mais detalhes sobre a notação θ em Cormen *et al.*, [2001].

A complexidade computacional de tempo seria alta para gerar códigos *hashes* distintos para cada subconjunto de consumidores. Isso implicaria na comparação da coluna gerada, com todas as colunas geradas anteriormente. Tal procedimento seria invocado sempre que uma nova coluna fosse encontrada pelos Algoritmos Híbridos. Tal fato inviabilizaria a utilização do PPC neste trabalho. Apesar do risco existente no método implementado, nenhuma falha foi observada durante os experimentos apresentados neste capítulo.

Considerando possíveis adaptações, as otimizações implementadas neste trabalho para a criação de rápida estrutura de armazenamento de colunas do PPC podem ser utilizadas em outros trabalhos que utilizam o PPC, mesmo para outros problemas combinatórios.