

UNIVERSIDADE FEDERAL DE MINAS GERAIS

DOUGLAS MOURA MIRANDA

**O PROBLEMA MULTIOBJETIVO DE ROTEAMENTO
DE VEÍCULOS COM JANELA DE TEMPO E COM
TEMPO DE VIAGEM/SERVIÇO ESTOCÁSTICO.**

Tese apresentada à Escola de Engenharia da
Universidade Federal de Minas Gerais para
obtenção do título de Doutor em Engenharia de
Produção.

Agosto, 2016

UNIVERSIDADE FEDERAL DE MINAS GERAIS

**UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHEARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

DOUGLAS MOURA MIRANDA

**O PROBLEMA MULTIOBJETIVO DE ROTEAMENTO
DE VEÍCULOS COM JANELA DE TEMPO E COM
TEMPO DE VIAGEM/SERVIÇO ESTOCÁSTICO.**

•

Tese apresentada à Escola de Engenharia da Universidade Federal de Minas Gerais para obtenção do título de Doutor em Engenharia de Produção.

Orientador: Prof. Samuel Vieira Conceição.

**DOUTORADO EM ENGENHARIA DE PRODUÇÃO
LINHA DE PESQUISA:
OTIMIZAÇÃO DE SISTEMAS LOGÍSTICOS E DE GRANDE PORTE**

Belo Horizonte
Agosto, 2016

DOUGLAS MOURA MIRANDA

**O PROBLEMA MULTIOBJETIVO DE ROTEAMENTO
DE VEÍCULOS COM JANELA DE TEMPO E COM
TEMPO DE VIAGEM/SERVIÇO ESTOCÁSTICO.**

Esta tese foi julgada e aprovada para a obtenção do grau de Doutor em Engenharia de Produção no Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais

Belo Horizonte, 24 de Agosto de 2016

BANCA EXAMINADORA

Prof. Dr. Samuel Vieira Conceição
Universidade Federal de Minas Gerais
Orientador

Prof. Dr. Marcelo Franco Porto
Universidade Federal de Minas Gerais

Prof. Dr. João Antônio de Vasconcelos
Universidade Federal de Minas Gerais

Prof. Dr. Alexandre Xavier Martins
Universidade Federal de Ouro Preto

Prof. Dr. Gilberto de Miranda Junior
Universidade Federal do Espírito Santo

AGRADECIMENTOS

Meus agradecimentos à:

Deus (mesmo sem entendê-lo) ☺

Minha amada esposa Mariana, pela paciência e apoio incondicional.

Meus pais (Celso e Joana) e minha irmã (Débora) por me apoiarem e desejarem o melhor para mim.

Meu orientador (Samuel) por me motivar, instruir e confiar em meu trabalho.

Meus professores, pelos conhecimentos adquiridos.

RESUMO

O roteamento de veículos com janela de tempo é particularmente importante num sistema logístico. Em aplicações reais é comum encontrar problemas que envolvam uma quantidade muito grande de clientes e que conseqüentemente fogem do alcance de métodos exatos, mas que podem ser resolvidos por metaheurísticas. O ambiente real também possui dados probabilísticos por natureza, como o tempo de viagem entre dois clientes e o tempo de atendimento de cada cliente. Isto faz com que um modelo de roteamento que considere as incertezas envolvidas nestes dados seja mais adequado.

Modelos determinísticos podem encontrar soluções que embora viáveis do ponto de vista do seu modelo matemático, na prática, apresentam casos com baixa probabilidade do veículo visitar o cliente antes do limite máximo de tempo estabelecido. Modelos estocásticos que incluem tais incertezas em sua formulação matemática podem assegurar uma probabilidade mínima de que o veículo visite o cliente dentro do limite de tempo.

Neste contexto, o presente trabalho aborda uma variante do Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT) no qual o tempo de viagem entre os clientes e o tempo de serviço em cada cliente são conhecidos apenas probabilisticamente. Um novo método é desenvolvido não só para estimar o tempo de chegada aos clientes, mas também para calcular a probabilidade dos veículos atenderem os clientes dentro de suas respectivas janelas de tempo. O modelo estatístico proposto é comparado com um benchmark da literatura.

Este trabalho também explora a relação de custo-benefício existente entre minimizar os custos de transporte e maximizar o nível de serviço (probabilidade do cliente ser visitado dentro da janela de tempo) criando-se uma versão multiobjetiva do problema. Conceitos da teoria de otimização multiobjetivo são utilizados para desenvolver um algoritmo memético, até então inédito na literatura, capaz de encontrar um conjunto de soluções não dominadas que ofereça informações importantes ao tomador de decisão. Um método de otimização multicritério é utilizado para permitir a seleção de uma solução a partir do conjunto de soluções obtido pelo algoritmo.

Palavras-chaves. Roteamento de Veículos, Tempo de Viagem Estocástico, Janela de Tempo, Tempo de Serviço Estocástico, Otimização Multiobjetivo.

ABSTRACT

This thesis tackles the vehicle routing problem with hard time windows and stochastic travel and service times using multiobjective optimization. A new method to compute the service level (probability of the vehicle visiting the customer before the end of the time window) is presented. The problem is modeled with two objective functions using the Pareto ranking technique in which the two objective dimensions are the operational cost and the service level. The trade-off between the minimization of the operational cost and the maximization of the service level is explored and the algorithm returns a set of solutions that fairly consider both of these dimensions. A memetic multiobjective algorithm is developed to solve all the 56 well know Solomon instances with 100 customers. It is provided details of the design and implementation of the algorithm and also a number of computational experiments with comprehensive analysis of the results. To our knowledge, this method is unprecedented in the literature.

Keywords. Vehicle routing with time windows, stochastic travel times, stochastic service times, metaheuristic, memetic algorithms, multiobjective optimization

LISTA DE FIGURAS

Figura 1: Nível de serviço para solução ótima determinística	19
Figura 2: Movimento 2-opt (antes e depois, respectivamente)	28
Figura 3: Realocação (antes e depois, respectivamente).....	28
Figura 4: Intercâmbio.....	28
Figura 5: 2-Opt* (Potvin & Rousseau, 1995b).....	40
Figura 6: Cross Exchange (Taillard, 1997)	40
Figura 7: Histogramas do tempo de chegada nos clientes	76
Figura 8: Teste de normalidade usando Anderson-Darling	77
Figura 9: Histogramas do tempo de início de atendimento nos clientes.....	78
Figura 10: Histograma do tempo chegada do problema R104	79
Figura 11: Histograma do tempo chegada do problema RC101	79
Figura 12: Soma de variáveis aleatórias.....	80
Figura 13: Generalização da soma de variáveis aleatórias.....	81
Figura 14: Função de distribuição cumulativa	88
Figura 15: Interpolação linear	88
Figura 16: Casos para normais truncadas no ponto zero.....	89
Figura 17: Função de distribuição cumulativa	92
Figura 18: Ilustração da evolução das buscas locais.....	95
Figura 19: Solução com três rotas e sua correspondente representação matricial.....	96
Figura 20: Zona circular.....	105
Figura 21: Tempo de execução para as três versões.....	113
Figura 22: Erro (percentil de 95%) para as três versões	113
Figura 23: Influência no erro.....	114
Figura 24: Influência no tempo	114
Figura 25: Tempo e erro para as 49 configurações de parâmetros	115
Figura 26: Frequência relativa do erro.....	117
Figura 27: Distribuição acumulada do erro.....	117
Figura 28: Influência da zona circular	119
Figura 29: Influência da estratégia de rotas e movimentos tabus.....	120
Figura 30: Convergência do hipervolume.....	121
Figura 31: Influência do nível de serviço.....	123
Figura 32: Qualidade das soluções para as classes de instâncias	126
Figura 33: Gap médio por classe	127
Figura 35: Intervalos dos objetivos para as classes de problemas.....	128
Figura 36: Intervalos do número de veículos para as classes de problemas	128
Figura 37: Tempo por classe	129
Figura 38: Relação entre tempo e convoluções	129
Figura 39: Fronteira Pareto: instâncias 1 a 9.....	129
Figura 40: Fronteira Pareto: instâncias 10 a 18	130
Figura 41: Fronteira Pareto: instâncias 19 a 27	130
Figura 42: Fronteira Pareto: instâncias 28 a 36	131
Figura 43: Fronteira Pareto: instâncias 37 a 45	131
Figura 44: Fronteira Pareto: instâncias 46 a 54	132
Figura 45: Fronteira Pareto: instâncias 55 a 56	132
Figura 46: Função utilidade para ambos o objetivos	134
Figura 47: Fronteira Pareto – instância C101.....	134
Figura 48: Fronteira Pareto – instância C101.....	135
Figura 49: Sensibilidade.....	136

Figura 50: Fronteira Pareto e solução escolhida – instância C205	137
Figura 51: Fronteira Pareto e solução escolhida – instância R211	138
Figura 52: Fronteira Pareto e solução escolhida – instância RC207	139

LISTA DE TABELAS

Tabela 1: Conjunto de duas soluções – Exemplo 1	18
Tabela 2: Conjunto de duas soluções – Exemplo 2	18
Tabela 3: Rota exemplo	76
Tabela 4: Notação dos parâmetros de entrada do algoritmo	83
Tabela 5: Resumo do desempenho das versões	114
Tabela 6: Resultados para o modelo estatístico	116
Tabela 7: Estatística descritiva	117
Tabela 8: Efeito da estratégia tabu no hipervolume	120
Tabela 9: Efeito dos critérios de saída	122
Tabela 10: Resultados para as 56 instâncias em termos do hipervolume	125
Tabela 11: Escolha para diferentes pesos dos objetivos	135
Tabela 12: Análise de sensibilidade: intervalos	136

LISTA DE ALGORITMOS

Algoritmo 1: Algoritmo para resolver a convolução	83
Algoritmo 2: Função “F” para o cálculo da função de distribuição cumulativa	86
Algoritmo 3: loop principal do algoritmo	93
Algoritmo 4: Gerar solução inicial	97
Algoritmo 5: Busca local - Pré-Intensificação	99
Algoritmo 6: Busca local - Intensificação	100
Algoritmo 7: Estrutura básica da busca local	101
Algoritmo 8: Busca local do tipo “Intercâmbio”	103
Algoritmo 9: Busca local do tipo “Realocação em cadeia”	106

SUMÁRIO

CAPÍTULO 1	14
INTRODUÇÃO	14
1.1- CONTEXTUALIZAÇÃO E MOTIVAÇÃO	14
1.2- OBJETIVOS.....	15
1.2.1- Objetivo geral	15
1.2.2- Objetivos específicos	16
1.3- CONTRIBUIÇÃO CIENTÍFICA DO ESTUDO	16
1.4- METODOLOGIA.....	16
1.5 – MOTIVAÇÃO PARA ABORDAGEM MULTIOBJETIVO.....	17
1.6 – MOTIVAÇÃO PARA ABORDAGEM ESTOCÁSTICA	19
1.7- ORGANIZAÇÃO DO ESTUDO	19
1.8- DELIMITAÇÃO DA PESQUISA.....	20
CAPÍTULO 2	21
PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)	21
2.1 – PRV CAPACITADO (PRVC).....	22
2.2 – EXTENSÕES DO PRVC	24
2.3 – TÉCNICAS DE SOLUÇÃO PARA O PRV	26
2.4 – HEURÍSTICAS PARA O PRV	26
2.4.1 – Algoritmos de Melhoria Intra-Rotas.....	27
2.4.2 – Algoritmos de Melhoria Inter-Rotas.....	28
2.4.3 – Metaheurísticas para o PRV.....	29
CAPÍTULO 3	33
PRV COM JANELAS DE TEMPO (PRVJT)	33
3.1 – TÉCNICAS EXATAS DE SOLUÇÃO PARA O PRVJT	35
3.2 – TÉCNICAS APROXIMADAS DE SOLUÇÃO PARA O PRVJT	36
3.2.1 – Heurísticas Construtivas para o PRVJT.....	36
3.2.1.1 – Algoritmos de Melhoria.....	39
3.2.2 – Metaheurísticas para o PRVJT.....	41
3.3 – PROBLEMAS TESTE DE SOLOMON	43
CAPÍTULO 4	45
PRV E PRVJT ESTOCÁSTICO	45
4.1 – APROFUNDANDO-SE NO PRV ESTOCÁSTICO (PRVE).....	46
4.1.1 – PRV com demanda e clientes estocásticos	48
4.1.2 – PRV com tempo de viagem estocástico.....	49
4.1.3 – PRVJTVE.....	50

4.1.4 – Estudos realizados por Zhang <i>et al.</i> (2013).....	57
4.2 – DISCUSSÃO DO CAPÍTULO 4.....	61
CAPÍTULO 5	64
PRV & PRVJT MULTIOBJETIVO	64
5.1 – PRV MULTIOBJETIVO.....	64
5.2 – PRVJT MULTIOBJETIVO.....	67
5.3 – DISCUSSÃO DO CAPÍTULO 5.....	70
CAPÍTULO 6.....	72
PROBLEMA PROPOSTO: PRVJT COM TEMPO DE VIAGEM/SERVIÇO ESTOCÁSTICO MULTIOBJETIVO (PRVJTTVSEM)	72
6.1 – O PROBLEMA: PRVJTTVSEM.....	73
6.2 – SUPOSIÇÃO DE NORMALIDADE.....	75
6.3 – MODELO ESTATÍSTICO.....	80
6.3.1 – Definição do problema estatístico.....	80
6.3.2 – Solução do problema estatístico.....	81
6.3.4 – A convolução.....	82
6.3.5 – A função de distribuição cumulativa.....	85
6.3.6 – A soma de normais truncadas.....	88
6.4 – METAHEURÍSTICA PARA O PROBLEMA PROPOSTO.....	93
6.4.1 – Solução Inicial.....	97
6.4.2 – Algoritmos VNDs.....	98
6.4.3 – Buscas Locais.....	101
6.4.4 – Operador de Seleção.....	108
6.4.5 – Operador de Cruzamento.....	109
6.5 – DISCUSSÃO DO CAPÍTULO 6.....	110
CAPÍTULO 7	111
RESULTADOS	111
7.1 – RESULTADOS PARA O MÉTODO ESTATÍSTICO.....	111
7.1.1 – Estudo de parâmetros e versões.....	112
7.1.2 – Comparação com benchmark da literatura.....	115
7.2 – RESULTADOS PARA O PRVJT ESTOCÁSTICO.....	118
7.2.1 – Zona circular.....	118
7.2.2 – Rotas e movimentos tabus.....	119
7.2.3 – Critérios de saída.....	121
7.2.4 – Nível de serviço requerido.....	123
7.2.5 – Resultados para as 56 instâncias de Solomon.....	123
7.2.6 – Decisão multi-critério.....	133

7.3 – DISCUSSÃO DO CAPÍTULO 7	139
CAPÍTULO 8	140
CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS	140
REFERÊNCIAS BIBLIOGRÁFICAS.....	143
ANEXOS.....	153
Valores de referência para hipervolume	153

CAPÍTULO 1

INTRODUÇÃO

1.1- CONTEXTUALIZAÇÃO E MOTIVAÇÃO

Em um mundo empresarial globalizado, as empresas que trabalham com distribuição própria de seus produtos ou prestam serviços de entrega ou coleta de mercadorias de qualquer tipo enfrentam diversos desafios como: elevada concorrência, necessidade de lidar com um elevado número de clientes com baixo consumo e pequenos volumes de entrega e, ainda, clientes mais exigentes tanto com a qualidade como também com o prazo de entrega dos produtos.

Para serem competitivas neste mercado, convém que as empresas ofereçam aos seus clientes características diferenciadoras, adotando processos mais rápidos e adaptados ao perfil do cliente. Groër *et al.* (2009) mencionam que para obter tais características, algumas empresas oferecem serviços como: rastreamento on-line de produtos e garantia de entrega no prazo acordado. Ao longo do tempo, diversas iniciativas neste setor tentam otimizar a utilização dos veículos e rotas para a consequente minimização dos custos de distribuição. Este tipo de problema é estudado como o Problema de Roteamento de Veículos, ou simplesmente, PRV.

Cheng & Wang (2009) consideram que o PRV tem um impacto econômico vital em um centro de distribuição. Na prática, a resolução deste problema contribui diretamente para redução de custos no sistema logístico. Como o problema é complexo, a solução geralmente fica nas mãos de uma pessoa experiente com conhecimento geográfico do local e rota. Segundo Toth & Vigo (2002a), por mais experiente que a pessoa responsável pelo planejamento dos processos de distribuição seja, procedimentos computadorizados têm permitido uma importante redução dos custos totais de transporte.

Situações em que o cliente especifica uma janela de tempo para ser atendido dão origem a uma variante chamada Problema de Roteamento de Veículos com Janela de Tempo (PRVJT). Bräysy e Gendreau (2005) dizem que o PRVJT aparece em muitos sistemas de distribuição. Os autores ainda dizem que o PRVJT tem sido objeto de grandes esforços de pesquisa tanto em abordagens de otimização aproximativas quanto exatas.

Para Zhang (2013) existem elementos como o tempo de viagem entre dois clientes que, no mundo real, podem ser altamente estocásticos devido a variações na capacidade das vias e flutuações de tráfego. Ignorar a natureza estocástica do problema pode conduzir a soluções sub-ótimas e até mesmo inviáveis. Sendo assim, uma motivação do presente trabalho é propor um novo método para resolver uma variante do PRVJT ainda pouco estudada e de forte relevância prática que diz respeito à natureza estocástica dos tempos de viagem e dos tempos de serviço. O modelo estatístico desenvolvido é capaz de calcular a probabilidade do veículo visitar o cliente antes do final da janela de tempo (nível de serviço). Isto permite incluir na formulação matemática do problema de otimização uma restrição que assegure o atendimento de um nível de serviço mínimo desejado.

Outra motivação é a aplicação de técnicas de Otimização Multiobjetivo para analisar a relação custo benefício entre a minimização dos custos de transporte e a maximização do nível de serviço relacionado ao atendimento das janelas de tempo dos clientes. Desenvolve-se um algoritmo memético, até então inédito na literatura, capaz de encontrar um conjunto de soluções não dominadas que ofereça informações importantes ao tomador de decisão. Um método de otimização multicritério é utilizado para permitir a seleção de uma solução específica a partir do conjunto de soluções obtido pelo algoritmo.

1.2- OBJETIVOS

Dentro do contexto já descrito no tópico 1.1, este trabalho possui os seguintes objetivos:

1.2.1- Objetivo geral

Desenvolver uma metaheurística capaz de resolver eficientemente o Problema de Roteamento de Veículos com Janela de Tempo e Tempo de Viagem/Serviço Estocástico (PRVJTTVSE).

1.2.2- Objetivos específicos

Para permitir o cumprimento do objetivo geral, são propostos os seguintes objetivos específicos:

- 1) Estudar trabalhos que mostrem uma forma de calcular ou pelo menos estimar o tempo de chegada do veículo nos clientes e a probabilidade dos veículos respeitarem as janelas de tempo para o PRVJTTVSE.
- 2) Propor um novo e eficiente método para estimar o tempo de chegada do veículo nos clientes, e também para estimar a probabilidade dos veículos atenderem os clientes dentro de suas respectivas janelas de tempo para resolver PRVJTTVSE.
- 3) Desenvolver um modelo e algoritmo para resolver o PRVJTTVSE Multiobjetivo “a posteriori”, ou seja, sem dar pesos para os objetivos, mas sim encontrando um conjunto de soluções não dominadas.

1.3- CONTRIBUIÇÃO CIENTÍFICA DO ESTUDO

Este trabalho faz-se relevante pelos seguintes motivos:

- 1) Investiga com profundidade as dificuldades de resolver adequadamente o PRVJTTVSE.
- 2) Desenvolve um método inédito para calcular de forma mais eficiente o nível de serviço nos clientes, fator chave para resolver o modelo estocástico.
- 3) Modela e implementa um algoritmo para o PRVJTTVSE Multiobjetivo na versão “a posteriori”, em nosso conhecimento, inédito na literatura.
- 4) Artigo publicado na revista ESWA com o título: “The vehicle routing problem with hard time windows and stochastic travel and service time”, por Miranda D. e Conceição, S. 2016.

1.4- METODOLOGIA

Esta sessão apresenta a forma como foi executada a pesquisa e a metodologia adotada. A presente pesquisa é classificada, sob o ponto de vista do delineamento adotado, como pesquisa experimental, uma vez que experimentos foram realizados para testar os algoritmos manipulando-se variáveis para criar situações de interesse com o propósito de

analisar e avaliar os métodos desenvolvidos. A pesquisa é de natureza quantitativa, uma vez que as variáveis envolvidas foram quantificadas segundo uma escala de intervalo ou de razão. Do ponto de vista do seu objetivo, o presente estudo pode ser classificado como uma pesquisa explicativa, o qual buscou aprofundar-se no conhecimento dos fenômenos, identificar relações de causa e efeito, descobrir o motivo de dificuldades e propor soluções.

Os procedimentos técnicos envolveram o estudo da arte, baseado em pesquisa bibliográfica sobre o tema e outras etapas descritas a seguir:

- 1) Revisão da literatura sobre o PRV, PRVJT, PRV Estocástico e Multiobjetivo.
- 2) Criar algoritmo de simulação estocástica que sirva de referência para obter os erros dos cálculos de probabilidade de violação das rotas.
- 3) Criar um banco de rotas que force o aparecimento do fenômeno estudado, caracterizado pelo aparecimento de uma função distribuição de probabilidade para o tempo de chegada do veículo que não é gaussiana e cuja forma varia ao longo dos clientes de uma dada rota.
- 4) Desenvolver um método para calcular o nível de serviço em cada cliente e compará-lo com *benchmarks* da literatura.
- 5) Implementar e testar a metaheurística utilizada para resolver o PRVJTTVSE Multiobjetivo.

A ferramenta de programação escolhida para implementar os algoritmos deste trabalho foi o Matlab, que embora não seja uma linguagem considerada rápida, possui benefícios importantes como maior agilidade na programação, uso de gráficos e de ferramentas estatísticas. Isto mostrou-se importante para entender e implementar os conceitos com que esta pesquisa deseja trabalhar. Além disto, o foco deste trabalho é o desenvolvimento de um modelo e método capaz de resolver o PRVJTTVSE Multiobjetivo.

1.5 – MOTIVAÇÃO PARA ABORDAGEM MULTIOBJETIVO

Para ilustrar a motivação de se modelar o problema como multiobjetivo, sejam as duas soluções a seguir (A e B), ambas com apenas um veículo, cada veículo com apenas

três clientes, e $\alpha = 90\%$ (nível de serviço desejado para os clientes). CTT é o custo total de transporte, NST é o nível de serviço total (média dos clientes) e NS_i o nível de serviço no cliente i .

Tabela 1: Conjunto de duas soluções – Exemplo 1

<u>Solução A</u>					
Cliente	1	2	3	CTT	NST
NS_i	90%	90%	90%	5800	90%
<u>Solução B</u>					
Cliente	1	3	2	CTT	NST
NS_i	90%	96%	99%	6000	95%

Para o exemplo 1, com o conjunto de soluções não dominadas formado apenas pelas duas soluções, percebe-se que a solução B possui um nível de serviço de 95%, bastante superior ao da solução A (90%), e um custo de transporte total de 6000, apenas um pouco pior que A (5800). Neste cenário, é provável que o tomador de decisão escolha a solução B.

Agora vamos supor as mesmas duas soluções porém alterando o valor do CTT da solução B, conforme Tabela 2.

Tabela 2: Conjunto de duas soluções – Exemplo 2

<u>Solução A</u>					
Cliente	1	2	3	CTT	NST
NS_i	90%	90%	90%	5800	90%
<u>Solução B</u>					
Cliente	1	3	2	CTT	NST
NS_i	90%	95%	98%	9200	95%

Para o conjunto formado pelas duas soluções acima, a solução B possui um nível de serviço total melhor que a solução A, porém um custo total de transporte bastante pior (quase 60% mais caro). Neste cenário, provavelmente o tomador de decisão iria privilegiar o custo e aceitar correr um risco maior de visitar algum cliente com atraso, até porque a restrição de nível de serviço continua sendo satisfeita e garantindo os 90% para todos os clientes.

Neste contexto, vê-se que pode ser útil para o tomador de decisão ter um conjunto de soluções que lhe permita, dentro do seu conhecimento, experiência e contexto, selecionar a solução que julgar mais conveniente.

1.6 – MOTIVAÇÃO PARA ABORDAGEM ESTOCÁSTICA

Seja a instância RC106 de Solomon (1987), com 100 clientes. A abordagem clássica assume tempo de viagem e de serviço determinísticos, e um único objetivo hierarquizado que visa minimizar primeiro o número de veículos e depois a distância total percorrida. A solução ótima desta instância é conhecida e pode ser vista em “Transportation Optimization Portal”, (2016).

Usando-se simulação, o nível do serviço para cada cliente na solução ótima é dado na figura a seguir:

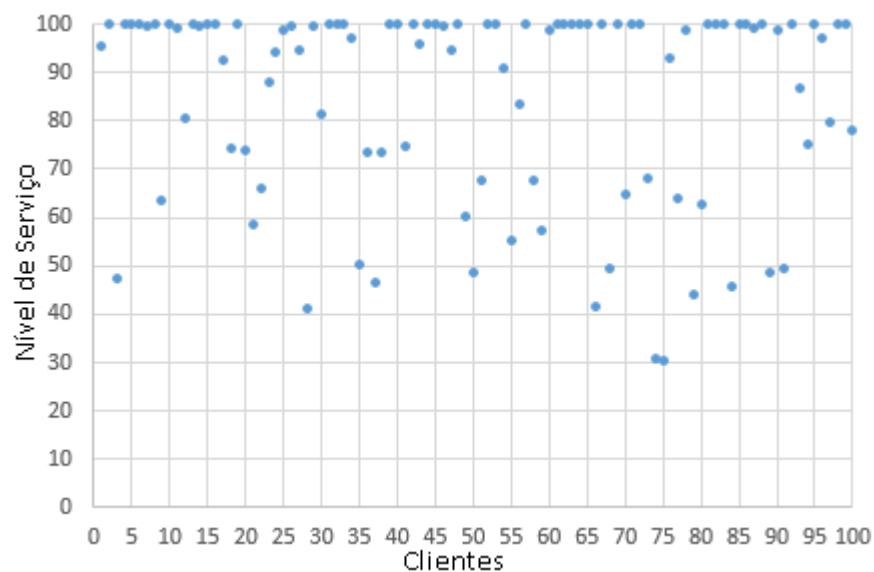


Figura 1: Nível de serviço para solução ótima determinística

A Figura 1 mostra que vários clientes possuem baixo nível de serviço. Dois clientes possuem nível próximos de 30% e 10 clientes entre 40 e 50%. No modelo clássico esta solução seria oferecida ao tomador de decisão sem que ele soubesse que vários clientes possuem baixa probabilidade de serem atendidos dentro da janela de tempo. Entende-se que esta informação possui impacto significativo no negócio e convém ser incluída no modelo do problema.

1.7- ORGANIZAÇÃO DO ESTUDO

O trabalho divide-se em 8 capítulos. No capítulo 1 são abordados aspectos relativos à contextualização e apresentação do problema de pesquisa, relevância, contribuição científica, justificativa da pesquisa e metodologia utilizada.

Os capítulos 2 a 5 são destinados à revisão da literatura de modo a trazer um estudo da arte sobre os assuntos envolvidos na pesquisa, tratando respectivamente os seguintes problemas: PRV, PRVJT, PRV Estocástico e o PRV Multiobjetivo.

O capítulo 6 aborda os algoritmos desenvolvidos no trabalho enquanto o capítulo 7 mostra os experimentos realizados e resultados obtidos. Finalmente, no capítulo 8, apresentam-se as conclusões finais da pesquisa, a síntese dos resultados, a adequação aos objetivos propostos e recomendações para trabalhos futuros.

Esclarecemos que o foco do trabalho é o PRVJTTVSE/Multiobjetivo. Porém, antes de abordar este problema foi preciso também estudar o PRV e o PRVJT, motivo pelo qual este assunto é abordado nos capítulos 2 e 3, separadamente.

1.8- DELIMITAÇÃO DA PESQUISA

Todos os algoritmos implementados neste trabalho assumem: frota homogênea, único depósito, se o veículo chegar antes da abertura da janela de tempo ele deve esperar, o tempo de viagem e o tempo de serviço seguem uma função densidade de probabilidade normal (truncada em zero), os veículos sempre visitam e entregam o serviço aos clientes.

CAPÍTULO 2

PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)

Considere que uma frota de veículos esteja disponível para o transporte de mercadorias demandadas ou ofertadas por um conjunto de clientes. Considere que cada veículo esteja inicialmente situado em um depósito. O Problema de Roteamento de Veículos (PRV) consiste em determinar a rota a ser percorrida por cada veículo, de modo que a demanda de todos os clientes seja satisfeita, e que cada veículo regresse ao depósito de origem ao final do período considerado. O objetivo é minimizar o custo total, definido pela soma dos custos das rotas.

O Problema de Roteamento de Veículos Capacitado (PRVC) é a versão mais conhecida do problema. Nela, cada cliente possui demanda determinística, ou seja, conhecida previamente, e que deve ser atendida integralmente por apenas um veículo. Todos os veículos são semelhantes e partem de um único depósito. Uma restrição de capacidade é imposta ao problema. Essa restrição estabelece que a soma das demandas de todos os clientes pertencentes a uma rota não deve superar a capacidade do veículo a ela designado.

Lenstra (1981) faz um estudo da complexidade computacional de resolver-se o PRV. O autor mostra que como uma generalização do Problema do Caixeiro Viajante (PCV), o PRV pertence à classe de problemas NP - difícil e um algoritmo em tempo polinomial para encontrar a solução ótima não é conhecido. Para uma análise aprofundada sobre complexidade computacional, fazemos referência a Garey & Johnson (1979) e Ziviani (2010).

Segundo Laporte (2009), o PRV foi introduzido há mais de 50 anos por Dantzig & Ramser (1959) com o título “*The truck dispatching problem*”. Alguns anos depois Clarke & Wright (1964) propuseram uma heurística gulosa que melhorou a abordagem de Dantzig & Ramser. Seguindo estes dois trabalhos, considerados pioneiros na modelagem do problema, muitos modelos exatos e heurísticos foram propostos para encontrar soluções ótimas e aproximadas para o PRV.

Um critério de classificação do problema é visto em Desrochers *et al.* (1990). Laporte & Nobert (1987) apresentam uma extensa revisão da literatura totalmente

dedicada a métodos exatos. Outras revisões da literatura, agora abordando modelos aproximados, foram apresentadas por Christofides *et al.* (1979), Fisher (1995), Toth & Vigo (1998a) e Golden *et al.* (1998).

Bergeglia *et al.* (2010) conclui em seu trabalho que o volume de artigos publicados sobre o PRV tem aumentado significativamente na última década. Apesar disto, ainda permanecem questões em aberto sobre o assunto, em virtude principalmente das inúmeras variações que o problema possui para se aproximar das situações do mundo real. O PRVC deu origem a diversos outros problemas, motivo pelo qual iremos estudá-lo em primeiro lugar, apresentando em seguida algumas de suas variações.

2.1 – PRV CAPACITADO (PRVC)

Baseando-se em Vieira (2009), o PRVC pode ser definido da seguinte forma: seja um grafo $G = (V, A)$ completo, em que A é um conjunto de arcos, que representam os caminhos que ligam os clientes entre si e estes ao depósito, e $V = 0, \dots, n$ denota um conjunto de $n+1$ vértices. Convencionamos que o vértice 0 representa o depósito e os outros simbolizam os clientes. A cada arco (i, j) é associado um custo não negativo, c_{ij} , que representa o custo de viagem do vértice i ao vértice j . Neste trabalho, c_{ij} é dado pela distância euclidiana entre os clientes.

Na maioria dos casos, os custos dos arcos satisfazem a desigualdade triangular, $c_{ik} + c_{kj} \geq c_{ij}$ para $i, k, j \in V$. Quando o custo do arco (i, j) é igual ao custo do arco (j, i) , dizemos que o problema é simétrico. Caso contrário, ele é dito assimétrico.

Seja um conjunto K de veículos idênticos, com capacidade C , onde $k \in K$. A cada cliente i é associado uma demanda não negativa, m_i . Para o depósito, definimos $m_0 = 0$.

O PRVC consiste em encontrar um conjunto de rotas, cada uma percorrida por um veículo, de modo a minimizar o custo total de transporte e a satisfazer as seguintes restrições:

- 1) Cada rota deve ter início e fim no depósito;
- 2) Cada cliente deve ser visitado apenas uma vez e somente por um veículo;
- 3) A soma das demandas dos clientes incluídos em uma rota não deve exceder a capacidade do veículo.

Para definir o problema, utilizamos as variáveis binárias de decisão, dadas por:

$x_{ijk} = 1$, se o veículo k perfaz o arco (i, j) ou $x_{ijk} = 0$, caso contrário; para $i, j \in \{0, 1, \dots, n\}, i \neq j$ e $k \in \{1, \dots, K\}$. A formulação matemática do PRVC é apresentada abaixo.

$$\text{Min } M \sum_{k \in K} \sum_{j=1}^n x_{0jk} + \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n \sum_{k \in K} c_{ij} x_{ijk} \quad (1)$$

s. a:

$$\sum_{j=1}^n x_{0jk} = \sum_{j=1}^n x_{j0k} = 1; \quad k = 1, \dots, K \quad (2)$$

$$\sum_{k \in K} \sum_{j=1}^n x_{ijk} = 1; \quad i = 1, \dots, n \quad (3)$$

$$\sum_{j=0}^n x_{ijk} - \sum_{j=0}^n x_{jik} = 0; \quad k = 1, \dots, K; \quad i = 1, \dots, n \quad (4)$$

$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in S} x_{jik} \leq |S| - v(S); \quad \forall S \subseteq V \setminus \{0\}; \quad |S| \geq 2 \quad (5)$$

$$\sum_{i=1}^n m_i \sum_{\substack{j=0 \\ j \neq i}}^n x_{ijk} \leq C; \quad k = 1, \dots, K \quad (6)$$

$$x_{ijk} \in \{0, 1\}; \quad i = 1, \dots, n; \quad j = 1, \dots, n; \quad k \in K \quad (7)$$

Na equação (1), M é um número muito grande para forçar a função objetivo a priorizar a redução do número de veículos. A equação (2) assegura que cada rota tenha início e fim no depósito. As restrições (3) e (4) garantem que cada cliente seja visitado exatamente uma vez, e que o veículo que chega a um cliente saia dele. A restrição (5) evita que seja formado um subciclo que não inclua o depósito. Nesta restrição, $v(S)$ representa o número mínimo de veículos necessário para atender o conjunto de clientes S .

Por exigir que o número de veículos usados para atender os clientes do conjunto S não seja inferior a $\nu(S)$, o mínimo necessário, a família de restrições (5) assegura, indiretamente, que a capacidade dos veículos não seja violada. Entretanto, para dar maior clareza à formulação do problema e facilitar referências futuras, preferiu-se manter explícita a restrição de capacidade dos veículos, motivo pelo qual introduziu-se (6).

O problema capacitado de roteamento possui várias outras formulações, muitas das quais incluindo apenas n^2 variáveis. De fato, pode-se notar que, como todos os veículos possuem a mesma capacidade e como as rotas só se encontram no depósito, não é necessário indicar qual veículo percorrerá cada rota. Desta forma, poderia ser utilizado x_{ij} simplesmente, em lugar de x_{ijk} .

2.2 – EXTENSÕES DO PRVC

Percebe-se que o PRVC não é capaz de representar todas as situações cotidianas enfrentadas pelos setores de logística das empresas de distribuição de mercadorias e serviços. Assim, muitas vezes é necessário introduzir neste problema algumas restrições associadas aos clientes, aos veículos ou aos depósitos, para que ele reflita as características dos problemas reais.

Como exemplo, na prática, os clientes podem requerer que:

- diferentes tipos de produtos sejam entregues;
- apenas um determinado subconjunto dos veículos seja usado para servi-los, em função, por exemplo, de limitações de acesso, ou do tipo de mercadoria transportada;
- o serviço de carregamento ou descarregamento das mercadorias seja executado em um tempo predefinido;
- a entrega dos produtos seja feita em um período determinado do dia ou mês (janela de tempo), de maneira a respeitar os horários de funcionamento de seus estabelecimentos, as limitações de tráfego e os prazos de entrega exigidos pelos consumidores por eles atendidos;
- produtos em excesso sejam devolvidos, quer por terem perdido a validade, como ocorre com jornais e revistas, quer por não terem sido bem recebidos pelos consumidores (exemplo: problemas de qualidade).

Os veículos, por sua vez:

- podem ser de diferentes tipos, ou possuírem capacidades diferentes;
- podem ser subdivididos em compartimentos, permitindo o transporte de diferentes produtos em diversas quantidades;
- podem suportar um tempo máximo de operação, antes de serem submetidos a revisão.

Além disso, é possível que exista mais de um depósito, de modo que:

- a rota de cada veículo possa ter início e fim em um depósito específico, diferente do utilizado por outros veículos;
- cada veículo possa partir de um depósito e terminar sua rota em outro depósito.

Muitos objetivos, frequentemente conflituosos, podem ser considerados para o problema. Segundo Toth & Vigo (2002b), alguns objetivos mais comuns são:

- Minimização do custo total de transporte.
- Minimização do número de veículos requeridos para servir todos os clientes.
- Balanceamento de rotas, seja em relação ao tempo de viagem ou ocupação dos veículos.
- Minimização das penalidades associadas ao atendimento parcial dos clientes.

Em algumas aplicações é necessário considerar versões estocásticas ou dinâmicas do problema, ou seja, problemas em que a priori existe apenas um conhecimento parcial da demanda dos clientes ou do tempo de viagem entre clientes da rota.

Outras variantes do PRVC podem ser obtidas se considerarmos, por exemplo, que:

- o número de veículos utilizados não precisa ser previamente determinado (ou seja, há um grande número de veículos a disposição).
- a demanda é estocástica.

Versões do PRV aplicadas a problemas mais complexos do mundo real demandam a utilização de muitas restrições, dando origem ao chamado “*Rich Vehicle*

Routing Problem” (Caceres, 2015). Um exemplo é o “*Green VRP*”, uma das mais recentes variantes do PRV, em que as frotas podem ser compostas de veículos movidos a algum tipo de energia alternativa para reduzir as emissões de dióxido de carbono (CO₂) adicionando restrições relacionadas ao impacto social e ambiental destas emissões (Koc & Karaoglan, 2016).

Os principais métodos encontrados na literatura para resolver os problemas de roteamento de veículos são divididos em duas grandes categorias: métodos exatos e métodos heurísticos. Os métodos exatos garantem a solução ótima do problema, mas, em geral, apresentam alto custo computacional. Os métodos heurísticos não garantem a solução ótima, mas geralmente resultam em soluções sub-ótimas de qualidade e um esforço computacional menor que os métodos exatos.

2.3 – TÉCNICAS DE SOLUÇÃO PARA O PRV

Desde a década de 60, muitos métodos exatos têm sido apresentados. Graças a estes esforços, o tamanho de problemas que possam ser resolvidos por métodos exatos tem aumentado significativamente. Um estudo particularmente importante que aborda métodos exatos é feito por Fukasawa *et al.* (2006) que aplicaram o método *branch-and-cut-and-price* para resolver o PRVC e conseguiram encontrar a solução ótima de instâncias com até 135 clientes, quantidade considerada alta na literatura.

O foco do presente trabalho está em técnicas aproximadas (heurísticas/metaheurísticas). Para maior aprofundamento em métodos exatos, uma extensa revisão da literatura é vista em Baldacci *et al.* (2012) e Toth & Vigo (2014).

A seguir serão abordadas técnicas aproximadas para o PRV.

2.4 – HEURÍSTICAS PARA O PRV

Nicholson (1971) define heurística como um procedimento para resolver problemas através de um enfoque “intuitivo”, em geral racional, no qual a estrutura do problema possa ser interpretada e explorada inteligentemente para obter uma solução razoável. Reeves (1993) define heurística como uma técnica que busca boas soluções (perto da ótima) com um custo operacional razoável, sem garantir soluções ótimas e, em muitos casos, não é capaz de afirmar quão próximo uma solução factível está da solução

ótima. Para Ribeiro (2004), uma heurística é qualquer método aproximado projetado com base nas propriedades estruturais ou nas características das soluções dos problemas, com complexidade reduzida em relação à dos algoritmos exatos e fornecendo, em geral, soluções viáveis de boa qualidade. Sendo assim, os métodos heurísticos também podem ser chamados de métodos aproximados em oposição aos métodos exatos.

Embora importante progresso venha sendo feito com o desenvolvimento de novos métodos exatos, os maiores problemas que podem ser consistentemente resolvidos por estes métodos possuem cerca de 50 clientes, e instâncias maiores podem ser resolvidas apenas em casos particulares (Renaud & Boctor, 2002). Devido ao limitado sucesso de métodos exatos, pesquisas têm sido desenvolvidas no sentido de desenvolver heurísticas capazes de encontrar boas soluções para instâncias de maior escala.

Além da teoria da complexidade computacional representar uma forte justificativa para a utilização de métodos heurísticos na solução do PRV, outro forte argumento apresentado por Reeves (1993) corresponde à possibilidade de modelar o problema real com maior precisão, uma vez que as heurísticas são mais flexíveis e aptas a operar com funções objetivo e/ou restrições mais complicadas e mais realistas do que os algoritmos exatos.

Ribeiro (2004) cita três categorias para os métodos heurísticos: métodos construtivos, busca local e metaheurísticas. Laporte (2009) coloca as heurísticas construtivas, heurísticas de duas fases e os métodos de melhorias (buscas locais) dentro de uma categoria chamada de “Heurísticas Clássicas”.

2.4.1 – Algoritmos de Melhoria Intra-Rotas

Estas heurísticas buscam fazer melhorias em uma única rota, sem realizar qualquer mudança nas demais rotas. Assim, a maioria dos métodos de melhoria atribuídos ao PCV podem ser utilizados (ver Lin, 1965). Nestes métodos, λ arcos são removidos da rota e reconectados de todas as formas possíveis. Se alguma reconexão apresentar melhoria no custo (a primeira ou a melhor), a mudança é efetivada. O procedimento pára em um ótimo local, quando nenhuma melhoria pode ainda ser obtida. Testar a λ -otimalidade da solução tem um custo computacional da ordem $O(n^\lambda)$. A Figura 2 ilustra um movimento λ -Opt com $\lambda=2$, ou seja, um 2-Opt, onde seleciona-se uma subrota (amarelo) e inverte-se a sua direção.



Figura 2: Movimento 2-opt (antes e depois, respectivamente)

Várias modificações desta ideia principal têm sido desenvolvidas. Lin & Kernighan (1973) modificam λ dinamicamente durante a busca. Or (1976) propôs o método *Or-Opt* que consiste em mover sequencias de 1, 2 ou 3 vértices consecutivos para outro local da rota. Isto cai num caso particular de mudanças λ -*Opt* para $\lambda=3$. Checar a Or-otimalidade requer tempo $O(n^2)$.

2.4.2 – Algoritmos de Melhoria Inter-Rotas

Estes algoritmos também chamados de algoritmos de melhoria multirotas, modificam mais de uma rota simultaneamente buscando melhorar a solução. Por exemplo, o método λ -*Opt* citado em 2.4.1 pode ser analogamente aplicado trocando-se λ arcos entre diferentes rotas.

Um método muito utilizado é chamado “Realocação” ou “*Reallocation*”. Nele, um cliente é removido de uma rota r_1 e inserido numa rota r_2 . A Figura 3 ilustra este tipo de movimento, onde o cliente “13” é realocado para outra rota, entre os clientes “23” e “24”.

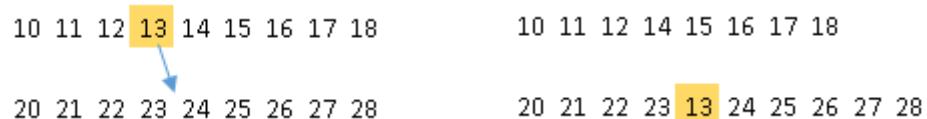


Figura 3: Realocação (antes e depois, respectivamente)

Outro método também muito utilizado é chamado “Intercâmbio” ou “*Interchange*”. Nele x clientes de uma rota r_1 são movidos para r_2 e y clientes de r_2 são movidos para r_1 . A Figura 4 ilustra um movimento para $x=1$ e $y=1$, onde os clientes “13” e “24” são intercambiados.



Figura 4: Intercâmbio.

Os métodos “Realocação” e “Intercâmbio” são derivados do método λ -*Interchange*, desenvolvido por Osman (1993). Dadas duas rotas distintas, em cada uma delas seleciona-se no máximo λ nós (a quantidade de nós pode diferir entre uma rota e outra). Em seguida, troca-se os nós entre as rotas.

Thompson & Psaraftis (1993) e Kindervater & Savelsbergh (1997) fornecem detalhes de trocas de arcos multirrotas para o PRV. O artigo de Thompson & Psaraftis (1993) descreve um *b-cycle* genérico, uma estratégia na qual uma permutação circular de b rotas é considerada e k clientes de cada rota são deslocados para a próxima rota da permutação cíclica. Os autores mostram que a aplicação de sequências específicas com $b=2$ ou variável e com $k=1$ ou 2 implica em resultados interessantes.

2.4.3 – Metaheurísticas para o PRV

Assim como as heurísticas clássicas, as metaheurísticas têm como objetivo explorar apenas parte do espaço solução. Entretanto, geralmente, isto é feito de um modo mais abrangente, fazendo com que as soluções encontradas sejam de melhor qualidade. Contrariamente às heurísticas clássicas, as metaheurísticas são mais gerais e têm capacidade de sair de ótimos locais, como será visto adiante.

São muitas as metaheurísticas aplicadas ao PRV. Algumas das principais são: *Simulated Annealing (SA)*, *Deterministic Annealing (DA)*, *Tabu Search (TS)*, *Genetic Algorithms (GA)*, *Ant Systems (AS)*, *Neural Networks (NN)*, *Greedy Randomized Adaptive Search Procedures (GRASP)*, *Adaptive Memory Procedure (AMP)*, *Guided Local Search (GLS)*, *Iterated Local Search (ILS)*, *VND (Variable Neighborhood Descent)* e *Memetic Algorithms (MA)*.

Todos estes tipos de metaheurísticas podem ser combinados entre si ou ainda com outros tipos não citados. Também é comum que metaheurísticas se utilizem de heurísticas clássicas, usando métodos construtivos para gerar soluções iniciais e de métodos de melhoria em pontos específicos do algoritmo. A seguir será dado mais detalhes sobre algumas metaheurísticas mais comumente aplicadas ao PRV, não necessariamente utilizadas no presente trabalho.

O GRASP (Feo & Resende, 1995; Assis, 2007 e Pereira, 2010) é um algoritmo *multi-start* que consiste em criar uma solução inicial e depois efetuar uma busca local para melhorar a qualidade da solução. Seu diferencial para outros métodos está na geração

desta solução inicial feita através de um método construtivo que se utiliza de uma função gulosa para avaliar os elementos a serem inseridos na solução. Uma vez realizada essa valoração, é construída uma lista de elementos candidatos, de onde escolhe-se um elemento aleatoriamente dentre os melhores avaliados da lista, para ser inserido na solução. Devido a este misto entre a função gulosa na construção de uma lista e a decisão aleatória sobre qual elemento dessa lista utilizar, diz-se que o GRASP é semi-guloso.

Segundo Gendreau *et al.* (1994), a Busca Tabu foi proposta por Glover em 1977, e dois dos primeiros trabalhos associadas ao PRV são de autoria de Willard (1989) e de Pureza (1990). Depois disso, uma série de outros autores publicaram os esforços de suas pesquisas, conseguindo resultados expressivos diante de outros métodos para o PRV. A Busca Tabu é um procedimento adaptativo que utiliza uma lista para guiar um método de descida a continuar a exploração do espaço de soluções mesmo na ausência de movimentos de melhora, evitando assim, a formação de ciclos (retorno numa solução previamente visitada).

Adaptive Memory Procedure (AMP) foi desenvolvido por Rochat & Taillard (1995). Uma memória adaptativa é um conjunto de boas soluções que são dinamicamente atualizadas através de processos de busca. Periodicamente alguns elementos destas soluções são extraídos do conjunto e combinados para produzir uma nova boa solução. No PRV, rotas selecionadas de várias soluções são usadas como ponto de partida. O processo de extração dá um maior peso para as rotas pertencentes às melhores soluções. No momento de selecionar as rotas, deve-se ter o cuidado de evitar a inclusão repetida de um cliente. Isto significa que o processo de seleção frequentemente termina com uma solução parcial que deverá ser completada usando-se alguma heurística de construção. Este procedimento obteve 2 novas melhores soluções nas 14 instâncias clássicas de Christofides *et al.* (1979).

O *Guided Local Search* (Voudouris, 2003) é uma metaheurística que conduz algoritmos de busca local pelo espaço de busca. Toda vez que o algoritmo encontra um mínimo local, uma característica considerada indesejável desta solução (digamos o valor do arco de maior comprimento) é selecionada e soluções que possuem esta característica têm seu custo penalizado. A função objetivo $f(s)$ é substituída por $h(s) = f(s) + \lambda \sum (p_i * I_i(s))$ em que p_i é a penalidade associada à característica i , $I_i(s)$ é uma variável binária que assume valor 1 caso a solução possua a característica i e valor 0 caso contrário, e λ é um parâmetro do algoritmo. Esse mecanismo penaliza características com alto custo na

solução corrente e que não tenham sido muito penalizadas durante a execução do algoritmo.

Iterated Local Search (ILS) possui um tipo de perturbação para fugir de ótimos locais. Dado o espaço de soluções viáveis S e uma função $BL : S \rightarrow S$, tal que dado $s \in S$, e $s^* = BL(s)$ tem-se $f(s) \geq f(s^*)$. A ideia básica do *ILS* (Lourenço, 2003) é fazer uma busca estocástica no espaço $S^* = BL(S)$. Isto é, se a função BL é um procedimento de busca local, o método *ILS* realiza uma busca pelo espaço de mínimos locais dado pela função. O *ILS* mantém uma solução corrente $s^* \in S^*$ e, a cada passo, encontra uma nova solução $s_p \in S^*$ pela geração de uma solução perturbada $s^{*'} \in S$ a partir de s^* seguido pela aplicação de BL em $s^{*'}$ para gerar $s_p^* \in S^*$. Nesse ponto, s_p^* pode ou não ser aceita como solução corrente. Perturbações muito fortes da solução corrente farão com que o espaço de soluções S^* seja explorado de maneira aleatória, levando a um algoritmo *Random Restart*, enquanto perturbações fracas podem fazer com que a busca volte à solução corrente e poucas novas soluções sejam exploradas.

O *Variable Neighborhood Descent* (VND) foi desenvolvido por Hansen & Mladenovic (2003). É uma variante da estratégia *Variable Neighborhood Search* (VNS), cuja ideia básica é a troca sistemática de vizinhanças. Trata-se de uma estratégia que consiste na combinação de busca local (*Descents*) em diversas vizinhanças, baseada no fato de que um mínimo local em uma vizinhança não é necessariamente um mínimo local em outra vizinhança. Assim, o *VND* realiza a troca sistemática de vizinhanças durante a busca local. Ao fim do procedimento, o *VND* alcança uma solução que é um mínimo local em todas as vizinhanças usadas e, conseqüentemente, com boas chances de ser uma boa solução.

Memetic Algorithms (MA) ou Algoritmos Meméticos têm sido amplamente estudados e aplicados a vários problemas de otimização encontrados na literatura. Uma ampla lista de aplicações pode ser encontrada em Moscato e Cotta (2003). O termo “Algoritmo Memético” foi introduzido por Moscato (1989) para descrever um processo evolutivo que tenha a busca local como parte decisiva na evolução. Essa busca pode ser caracterizada como sendo um refinamento local dentro de um espaço de busca, de modo que um agente (indivíduo autônomo) pode ter seu nível de adaptação aumentado após passar por uma etapa de refinamento.

Há ainda outras metaheurísticas citadas na literatura, como:

- *Simulated Annealing (Recozimento Simulado)*: Osman (1993).
- *Deterministic Annealing (Recozimento Determinístico)*: Dueck and Scheurer (1990).
- *Algoritmos Genéticos*: Holland (1975), Baker & Aycchew (2003)
- *Ant Systems (AS)*: Bullnheimer et al. (1999)
- *Neural Networks (NN)*: Hopfield & Tank (1985), um artigo pioneiro na aplicação da técnica ao VRP, e Potvin & Robillard (1995a).

Este capítulo abordou o PRV capacitado. Com o objetivo de aprofundar-se em outra variante diretamente relacionada ao problema abordado pela presente tese, o próximo capítulo retrata o PRV com janela de tempo.

CAPÍTULO 3

PRV COM JANELAS DE TEMPO (PRVJT)

O Problema de Roteamento de Veículos com Janela de Tempo (PRVJT) é a generalização do problema capacitado na qual se associa, a cada cliente, um período de tempo no qual algum veículo deve começar a atendê-lo. A esse intervalo dá-se o nome de janela de tempo.

Segundo Solomon & Desrosiers (1988), o problema de roteamento de veículos com janelas de tempo também é *NP-hard* (NP-difícil) por ser uma extensão do PRV.

Como no PRVC visto no capítulo anterior, o problema é representado por um grafo $G = (V, A)$ em que A é o conjunto de arcos e $V = 0, \dots, n$ é o conjunto de vértices. O vértice 0 indica o depósito, enquanto os demais nós representam os clientes.

O conjunto de clientes tem sua demanda satisfeita por uma frota de veículos que partem do depósito. Todos os veículos possuem uma capacidade constante C , de forma que a frota é homogênea.

A cada cliente i associa-se:

- 1) Uma demanda m_i ;
- 2) Um tempo de serviço s_i ;
- 3) Um instante de início da janela de tempo e_i ;
- 4) Um instante para o final da janela de tempo l_i ;

O depósito também pode possuir uma janela de tempo $[e_0, l_0]$, indicando o momento a partir do qual os veículos podem começar a trafegar e o momento em que o último veículo deve retornar.

Em virtude da existência de janelas de tempo, é necessário associar a cada arco (i, j) , além do custo c_{ij} , o escalar t_{ij} , que representa o tempo necessário para ir do vértice i ao vértice j . Por simplicidade, sem perda de generalidade, é comum adotar-se $c_{ij} = t_{ij}$, o que corresponde a atribuir um valor unitário à velocidade dos veículos. Além disso, a desigualdade triangular é respeitada.

Embora existam problemas em que a janela de tempo pode ser violada mediante pagamento de penalidades (classe conhecida como *Soft Time Windows*), trabalharemos aqui com o caso em que isso não é permitido (classe conhecida como *Hard Time Windows*). Por outro lado, um veículo pode chegar ao endereço de um cliente j antes do início de sua janela de tempo. Neste caso, o veículo deve permanecer parado em espera até o início do serviço.

O objetivo primário do problema é minimizar o número de veículos e o objetivo secundário é minimizar a distância total percorrida, neste caso, o tempo total de viagem. Assim como no PRVC, a variável binária x_{ijk} indica se o veículo k percorreu ou não o arco que liga o nó i ao nó j . Por sua vez, a variável real b_i com $i = 1, \dots, n$, indica o instante de início do atendimento ao cliente i .

A formulação matemática do problema é dada a seguir, sendo baseada no trabalho de Vieira (2009) em que a função objetivo foi modificada para hierarquizar os objetivos:

$$\text{Min } M \sum_{k \in K} \sum_{j=1}^n x_{0jk} + \sum_{i=0}^n \sum_{\substack{j=0 \\ j \neq i}}^n \sum_{k \in K} c_{ij} x_{ijk} \quad (8)$$

s. a:

$$\sum_{j=1}^n x_{0jk} = \sum_{j=1}^n x_{j0k} = 1; \quad k = 1, \dots, K \quad (9)$$

$$\sum_{k \in K} \sum_{j=1}^n x_{ijk} = 1; \quad i = 1, \dots, n \quad (10)$$

$$\sum_{j=0}^n x_{ijk} - \sum_{j=0}^n x_{jik} = 0; \quad k = 1, \dots, K; \quad i = 1, \dots, n \quad (11)$$

$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in S} x_{jik} \leq |S| - v(S); \quad \forall S \subseteq V \setminus \{0\}; \quad |S| \geq 2 \quad (12)$$

$$\sum_{i=1}^n m_i \sum_{\substack{j=0 \\ j \neq i}}^n x_{ijk} \leq C; \quad k = 1, \dots, K \quad (13)$$

$$\sum_{k \in K} \sum_{\substack{i=0 \\ i \neq j}}^n x_{ijk} (b_i + s_i + t_{ij}) \leq b_j; j = 1, \dots, n \quad (14)$$

$$e_i \leq b_i \leq l_i; i = 0, \dots, n \quad (15)$$

$$x_{ijk} \in \{0,1\}; i = 1, \dots, n; j = 1, \dots, n; k \in K \quad (16)$$

A função objetivo (8) e as restrições (10)-(13) e (16) foram herdadas do PRVC. A restrição (14) relaciona o instante de início do atendimento de dois clientes visitados consecutivamente por um mesmo veículo. Ela evita que o intervalo entre os instantes de início do atendimento desses clientes seja inferior à soma do tempo gasto no atendimento do primeiro cliente com o tempo consumido na viagem entre os dois clientes. Por sua vez, (15) impede que o início do atendimento do cliente i ocorra fora de sua janela de tempo.

A folga da restrição (14) corresponde ao tempo de espera do veículo que atende o cliente j , ou seja, o tempo consumido entre o instante de chegada do veículo ao endereço do cliente e o início do atendimento deste.

3.1 – TÉCNICAS EXATAS DE SOLUÇÃO PARA O PRVJT

Os principais métodos exatos utilizados na resolução do PRVJT são o *Branch-and-Bound*, o *Branch-and-Cut*, *Branch-and-Price* e ainda *Branch-and-Cut-and-Price*. Todos eles utilizam alguma técnica de relaxação do problema de programação inteira, combinada, dependendo do método, com o uso de planos de corte e de geração de colunas. Como exemplo da aplicação do *Branch-and-Bound* ao PRVJT citamos: Baker (1982) e Kolen *et al.* (1987).

O método de *Branch-and-Cut* consiste na combinação do procedimento de planos de corte com a técnica de *Branch-and-Bound*. Bard *et al.* (2002) apresentam detalhes da aplicação deste método ao PRVJT. Neste trabalho, limites inferiores são obtidos resolvendo-se uma série de problemas relaxados. Uma grande variedade de cortes são introduzidos para apertar o problema relaxado obtido do programa inteiro misto original. Soluções viáveis ou limites superiores são obtidos com a ajuda de uma heurística *GRASP*, resolvendo-se instâncias de Solomon (1987) de 50 a 100 clientes.

O método *Branch-and-Price* é basicamente um método *Branch-and-Bound* com geração de colunas. Em cada nó da árvore de busca encontrada pelo método *Branch-and-*

Bound, utiliza-se o método de geração de colunas com o propósito de encontrar novas soluções viáveis. Como exemplo da aplicação do *Branch-and-Price* ao PRVJT citamos: Desrochers *et al.* (1992) e Hernandez (2016).

Branch-and-cut-and-price é utilizado por Salani & Batarra (2014) para resolver PRVJT na versão “Soft Time Windows”. Trata-se de uma combinação do *branch-and-price* e do *branch-and-cut*. As variáveis são dadas implicitamente e então adicionadas ao problema apenas quando necessárias. Planos cortantes são adicionados ao problema durante o processo de solução com o objetivo de fortalecer a relaxação de programação linear. Variáveis e planos cortantes são criados de forma alternada. Uma revisão mais completa pode ser vista em Desaulniers *et al.* (2014).

3.2 – TÉCNICAS APROXIMADAS DE SOLUÇÃO PARA O PRVJT

Como já citado, o PRVJT enquadra-se na classe dos problemas NP – difíceis. Sua alta complexidade faz dos métodos heurísticos uma interessante alternativa quando é preciso encontrar boas soluções em um tempo limitado.

A grande maioria dos métodos aproximados vistos no capítulo 2 podem ser adaptados aos problemas com janela de tempo. A seguir, alguns dos métodos mais utilizados na resolução do PRVJT e suas adaptações serão apresentados.

3.2.1 – Heurísticas Construtivas para o PRVJT

Solomon (1986) descreve um método do tipo “rotear primeiro e agrupar depois”. Inicialmente, uma única rota unindo todos os clientes é gerada, e então dividida em rotas menores. A rota inicial pode, por exemplo, ser gerada como em um PCV sem considerar-se as restrições de capacidade e tempo. Nenhum resultado computacional é fornecido no artigo quanto a esta heurística.

Várias heurísticas são descritas em Solomon (1987). Aqui será descrito brevemente duas delas, e uma terceira (de melhores resultados) será descrita com mais detalhes.

Um método consiste na criação de uma lista de economias constituída por pares de nós e então as rotas começam a ser formadas de forma sequencial ou paralela. Para

lidar com as restrições tanto espaciais quanto temporais, Solomon determina um limite para o tempo de espera da rota.

Um segundo método, trata-se de uma estratégia do vizinho mais próximo orientado no tempo. Uma rota é iniciada encontrando-se o cliente mais próximo do depósito que ainda não faça parte da solução. A cada iteração, a heurística procura pelo cliente mais próximo do último adicionado a uma rota e adiciona-se este cliente no final da rota corrente. Uma nova rota é iniciada quando o processo não consegue encontrar um cliente a ser inserido na rota corrente de forma viável.

A terceira heurística é a que apresentou o melhor desempenho, sendo chamada no artigo de *II*. Uma rota é inicializada com um cliente “semente” e os demais clientes que ainda não fazem parte da solução são adicionados até que a rota viole uma das restrições de tempo ou capacidade. Se ainda existem clientes não roteados, o processo de inicialização de uma nova rota e inserção de clientes continua até que todos os clientes sejam servidos. Os clientes sementes são selecionados encontrando-se aqueles mais distantes do depósito ou aqueles com o tempo de atendimento mais cedo. Após inicializar uma rota com um cliente semente, o método usa 2 critérios para selecionar o cliente u a ser inserido entre os clientes i e j .

Seja $(i_0, i_1, i_2, \dots, i_m)$ a rota corrente, onde i_0 e i_m representam o depósito. Para cada cliente não roteado u , primeiro calcula-se o melhor custo de inserção da rota conforme equação (17).

$$c_1(i(u), u, j(u)) = \min\{c_1(i_{p-1}, u, i_p)\} \text{ onde } p = 1, \dots, m \quad (17)$$

A seguir, o melhor cliente u^* a ser inserido na rota é aquele para o qual o problema (18) é resolvido.

$$c_2(i(u^*), u^*, j(u^*)) = \max \left\{ \begin{array}{l} c_2(i(u), u, j(u)) | u \text{ não foi} \\ \text{selecionado e a rota permanece viável} \end{array} \right\} \quad (18)$$

O cliente u^* é então inserido na rota entre $i(u^*)$ e $j(u^*)$. Quando não é mais possível encontrar um cliente a ser inserido de forma viável, inicia-se uma nova rota, a não ser que todos os clientes já façam parte da solução. Mais precisamente, $c_1(i, u, j)$ é calculado conforme equação 19.

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j) \quad (20)$$

Onde $\alpha_1 + \alpha_2 = 1$; $\alpha_1 \geq 0$; $\alpha_2 \geq 0$;

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij}; \mu \geq 0; \quad (21)$$

$$c_{12}(i, u, j) = b_j^{new} - b_j; \quad (22)$$

Os parâmetros d_{iu} , d_{uj} e d_{ij} são as distâncias entre os clientes i e u , u e j e i e j respectivamente. O parâmetro μ controla o cálculo da economia em termos de distância, e b_j^{new} denota o novo tempo de início de atendimento do cliente j , dado que u foi inserido na rota, e b_j é o início do atendimento do cliente j antes da inserção.

O critério $c_2(i, u, j)$ é calculado como a seguir:

$$c_2(i, u, j) = \lambda d_{0u} - c_1(i, u, j); \lambda \geq 0; \quad (23)$$

O parâmetro λ é usado para definir o quanto o local de melhor inserção para o cliente não roteado depende de sua distância em relação ao depósito, por outro lado, dita o quanto sua melhor posição depende da distância extra e do tempo extra requeridos para visitar o cliente na rota corrente.

As três heurísticas citadas por Solomon (1987) possuem complexidade $O(n^2 \log n^2)$. A heurística *II*, que apresentou resultados melhores que as demais, na maioria dos casos obteve soluções a mais de 10% da ótima. De qualquer forma, por ser uma heurística rápida, a qualidade da solução pode ser considerada adequada como uma solução inicial de uma metaheurística.

Dullaert (2000) e Dullaert & Bräysy (2003) argumentam que o critério $c_{12}(i, u, j)$ subestima o tempo adicional necessário para inserir um novo cliente u entre o depósito e o primeiro cliente na rota corrente em construção. Isto pode fazer com que o critério de inserção selecione uma posição de inserção que não é a melhor. Então, uma rota com um número relativamente pequeno de clientes pode ter um tempo de planejamento maior que o necessário. O autor modifica este critério de inserção e tem uma importante melhoria nos resultados.

Ainda em Solomon (1987) é proposto uma heurística chamada de “Heurística de Varredura Tempo-Orientada”. Na primeira etapa (fase de grupamento) atribui-se clientes aos veículos conforme Gillett & Miller (1974), descrita no capítulo 2. Na segunda fase, é feito a construção das rotas aplicando-se uma heurística de construção, por exemplo, *II*, a cada um dos grupamentos. Entretanto, nesta fase, alguns clientes dos grupamentos podem não estar inclusos na solução (em função da janela de tempo). Se isso ocorrer, o processo de *Gillett & Miller* é reiniciado usando apenas os clientes não roteados.

3.2.1.1 – Algoritmos de Melhoria

A vizinhança de uma rota r consiste em um conjunto de rotas que podem ser obtidas a partir de r através da substituição de k de seus arcos por outro conjunto de k arcos. Tais substituições são chamadas de k -trocas e uma rota que não pode ser melhorada por uma k -troca é dita k -ótima. Verificar a k -otimalidade requer um tempo $O(n^k)$.

Russel (1977) é um dos primeiros trabalhos em um PRVJT com o uso de uma heurística de melhoria k -ótima. A chamada abordagem “M-Rotas” foi capaz de resolver problemas com poucos clientes possuindo restrições de janelas de tempo. Para um problema com 163 clientes, onde 15 % deles possuíam janelas de tempo, uma solução foi encontrada em menos de 90 segundos, o que para a época é satisfatório.

Potvin & Rousseau (1995b) comparam diferentes heurísticas de troca para o PRVJT (2 -Opt, 3 -Opt e Or -Opt) e introduz um novo 2 -Opt* cuja principal ideia é combinar 2 rotas de modo que os últimos clientes de uma dada rota são introduzidos após os primeiros clientes da outra rota, de modo a preservar a orientação das rotas. Este operador é ilustrado na Figura 5, onde os quadrados em branco são os depósitos (duplicados) e os círculos em preto são os clientes. Os arcos $(i, i+1)$ e $(j, j+1)$ são substituídos por $(i, j+1)$ e $(j, i+1)$, ou seja, a parte final de ambos são trocados. Como um caso especial, é possível combinar duas rotas em uma única rota se o arco $(i, i+1)$ é o primeiro em sua rota e o arco $(j, j+1)$ é o último sem sua rota ou vice-versa. Uma abordagem híbrida baseada no 2 -Opt* e Or -Opt mostra-se particularmente poderosa. Esta abordagem oscila entre as duas vizinhanças trocando o operador cada vez que um ótimo local é encontrado. Os autores também testam uma implementação onde os dois operadores são unidos. As soluções iniciais são criadas com o algoritmo II de Solomon (1987).

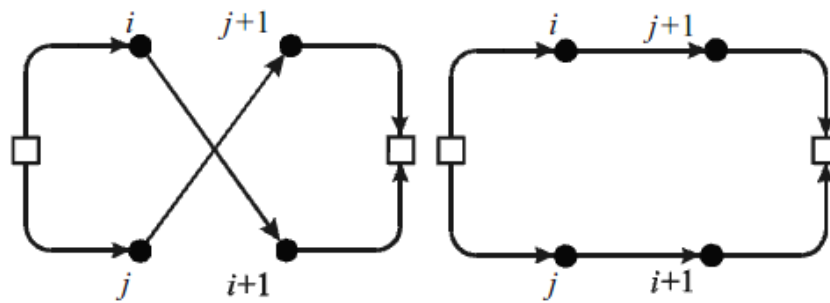


Figura 5: 2-Opt* (Potvin & Rousseau, 1995b)

Outro método muito útil em problemas com janela de tempo é o *Cross Exchange* (Taillard, 1997). O método é ilustrado pela Figura 6.

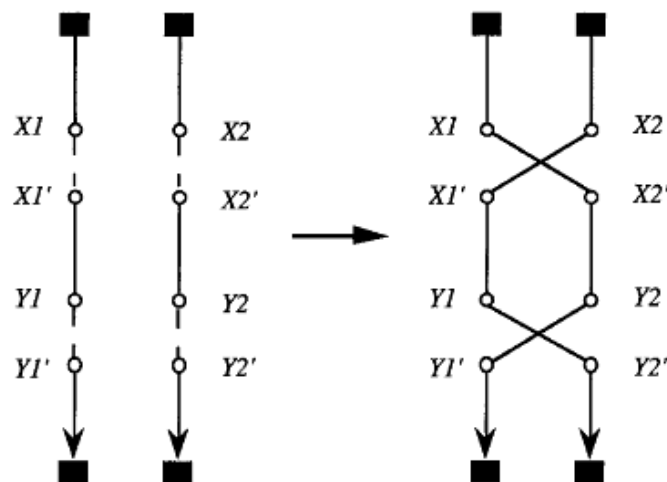


Figura 6: Cross Exchange (Taillard, 1997)

Trata-se de um método que realiza movimentos inter-rotas. Na Figura 6, temos duas rotas, onde os quadrados em preto representam o depósito (duplicado para simplificar o desenho) e os círculos brancos são os clientes. Inicialmente os 2 arcos $(X1, X1')$ e $(Y1, Y1')$ são removidos da primeira rota enquanto os arcos $(X2, X2')$ e $(Y2, Y2')$ são removidos da segunda rota. Então os segmentos $X1'-Y1$ e $X2'-Y2$, os quais possuem uma quantidade arbitrária de clientes entre eles, são trocados pela introdução dos novos arcos $(X1, X2')$, $(Y2, Y1')$, $(X2, X1')$ e $(Y1, Y2')$.

Note que as restrições de janela de tempo definem uma orientação implícita em cada rota. No exemplo da Figura 6, $Y1$ é visitado após $X1'$. O segmento $X1'-Y1$ tem a mesma orientação após o movimento. O mesmo ocorre para $Y2$ e $X2'$.

O caso *2-Opt** ilustrado na Figura 5 é um caso especial do *Cross Exchange*. O *2-Opt** troca 2 arcos tomados de 2 diferentes rotas, estando os arcos sempre diretamente conectados ao depósito. O *Or-Opt* também é um caso especial do *Cross Exchange*, onde há 3 ou menos clientes entre os 2 pontos de corte de cada uma das rotas envolvidas.

3.2.2 – Metaheurísticas para o PRVJT

Todas as metaheurísticas citadas no capítulo 2 podem e são adaptadas por muitos pesquisadores para resolver o PRVJT. É preciso obviamente incluir nos algoritmos testes que garantam a viabilidade da solução no que diz respeito às restrições de janela de tempo dos clientes.

Com a finalidade de facilitar a verificação de viabilidade de uma rota, após a inserção de um cliente, Solomon (1987) criou um procedimento denominado *Push Forward*, que tem como objetivo transferir aos clientes posteriores a u , o acréscimo de tempo que a inclusão deste acarreta nos próximos atendimentos. Assim, se $b_{i_p}^{new}$ é o novo instante de início de atendimento do cliente i_p , definimos o *Push Forward* deste cliente como:

$$PF_{i_p} = b_{i_p}^{new} - b_{i_p} \quad (24)$$

Para os próximos clientes da rota, esse aumento no tempo é dado por:

$$PF_{i_k} = \max\{0, PF_{i_{k-1}} - w_{i_k}\}, k = p + 1, \dots, m; \quad (25)$$

Onde $w_{i_j} = b_{i_j} - b_{i_{j-1}} - s_{i_{j-1}} - w_{i_{j-1}i_j}$ é o tempo de espera para o início do atendimento do cliente i_k . Note que podemos ter $PF = 0$. Isto ocorre quando o tempo de espera é maior ou igual ao aumento ocorrido no PF anterior. Neste caso, não haverá alterações no início de atendimento dos próximos clientes. A inclusão do depósito ao final da rota tem a finalidade de garantir que o veículo retorne dentro do intervalo de tempo especificado. Esta é uma forma eficiente encontrada por Solomon (1987) para testar a viabilidade ao se adicionar um novo cliente em uma rota parcial, sendo bastante útil ao realizar movimentos tanto nas metaheurísticas quanto nas heurísticas.

Uma outra particularidade do PRVJT que muda sua abordagem em relação ao PRVC num método aproximado é o fato de que num PRVC a direção em que os clientes de uma rota são percorridos não importa, ou seja, a rota pode ser percorrida tanto no sentido horário quanto no anti-horário e as restrições do problema continuarão a ser

satisfeitas. Já num PRVJT, a direção em que a rota é percorrida é importante, existindo uma correlação entre a janela de tempo dos clientes e a ordem em que os clientes aparecem nas rotas. Esta característica pode ser útil em estratégias de exploração do espaço de busca (Nagata, 2007).

Metaheurísticas têm sido o foco de muitos trabalhos envolvendo o PRVJT. Taillard *et al.* (1997) descreveu uma metaheurística baseada na Busca Tabu para o *VRP with Soft Windows*. Trata-se de um tipo de PRVJT em que é permitido atender um cliente mesmo após o fim da janela de atendimento, porém em contrapartida, há uma penalidade atribuída à solução, proporcional ao tamanho da violação da restrição. Porém, o mesmo modelo do autor pode ser utilizado para situações em que nenhuma violação é permitida, aumentando-se fortemente as penalidades pelo atraso no atendimento ao cliente. A metaheurística é baseada no conceito de memória adaptativa introduzida por Rochat & Taillard (1995) e no procedimento de decomposição e reconstrução proposto por Taillard (1993). A memória adaptativa utiliza-se de um conjunto de rotas provenientes de boas soluções encontradas durante a busca. Ao selecionar-se rotas deste conjunto de soluções para formar uma nova solução, obtém-se uma solução parcial que é completada utilizando a heurística de inserção *II* proposta por Solomon (descrita no tópico 3.2.1). A cada iteração, uma solução é construída através de um processo de seleção aleatório onde rotas pertencentes a soluções com melhor avaliação da função objetivo têm maior probabilidade de serem selecionadas na lista de rotas que compõem a memória adaptativa. Esta solução é então melhorada através de repetidos chamados a uma heurística de Busca Tabu. As rotas desta solução melhorada são armazenadas na memória adaptativa se a mesma não estiver completa ou se a solução for melhor que as soluções que já estavam na lista da memória adaptativa. Então o processo continua até que um critério de parada seja satisfeito.

Vidal *et al.* (2013) apresentou um algoritmo genético híbrido, introduzindo novas estruturas para lidar com as janelas de tempo. Novos tipos de movimentos foram desenvolvidos penalizando-se soluções inviáveis quanto às janelas de tempo e duração das rotas, permitindo que a avaliação de movimentos numa dada vizinhança seja mais rápida. Também são estudadas formas de decompor o problema a partir de suas características estruturais e geométricas para aplicação em problemas de larga escala.

Mais recentemente, Peiqing *et al.* (2015) apresenta um trabalho prático no setor de transporte de alimentos, em que considera não apenas as janelas de tempo, mas também

o efeito das irregularidades das vias em danificar frutas e vegetais frescos durante o transporte. Um algoritmo genético foi usado para resolver o problema. Os resultados mostraram que o VRPTW considerando as irregularidades das vias pode influenciar significativamente o custo total de transporte quando comparado com os modelos tradicionais.

Armas et al. (2015) resolvem um “*Rich VRPTW*”, tratando restrições adicionais para lidar com as peculiaridades de aplicações práticas. Uma metaheurística do tipo “*Variable Neighborhood Search*” (VNS) é desenvolvida para resolver o problema considerando uma frota heterogênea de veículos, janelas de tempo “soft”, e priorização de clientes. Vários tipos de funções objetivos são consideradas, incluindo a minimização da distância percorrida e da diferença entre a distância percorrida pelos veículos. O algoritmo desenvolvido foi incorporado ao sistema de gestão de frotas da empresa em que o estudo de caso ocorreu e os resultados mostram uma melhoria significativa na qualidade do roteamento, quando comparado com a soluções encontradas pela empresa antes.

3.3 – PROBLEMAS TESTE DE SOLOMON

A heurística para o PRVJT proposta nesta tese foi validada utilizando-se os problemas propostos por Solomon (1987). Estes problemas são baseados em alguns dados de problemas utilizados por Christofides *et al.* (1979) para o PRV capacitado. Trata-se de diferentes classes de instâncias, cada qual com características geográficas e de restrições específicas. Os problemas foram propostos nas dimensões 25, 50 e 100 consumidores. São também divididos em seis grupos: R1, R2, C1, C2, RC1 e RC2. Em todos os casos, os consumidores estão geograficamente distribuídos em um quadrado no plano XY com dimensões 100 por 100.

Algumas características que afetam o desempenho dos algoritmos aplicados ao PRVJT são: distribuição geográfica dos clientes, número de clientes atendidos por um veículo, proporção de clientes com janelas de tempo e o tamanho das janelas de tempo.

Nas classes R1 e R2 os clientes estão uniformemente distribuídos. Nas classes C1 e C2, os clientes estão agrupados. E nas classes RC1 e RC2 há mistura entre as duas classes anteriores. As classes R1, C1 e RC1 possuem um horizonte de tempo curto (janela do veículo ou depósito) e permitem poucos clientes por rota (aproximadamente de 5 a

10). Diferentemente, as classes R2, C2 e RC2 possuem um horizonte de tempo mais longo permitindo que muitos clientes (mais de 30) sejam atendidos por um mesmo veículo.

Uma variação importante é a diminuição ou aumento da janela de tempo dos consumidores. Se por um lado, pequenos intervalos como janelas de tempo diminuem significativamente o espaço de busca, o que pode ser desejável por um lado, requer complexos operadores de obtenção de vizinhança nas heurísticas de busca local (Alvarenga, 2005). As diferentes classes de problemas de Solomon têm como objetivo estabelecer cenários distintos para que diferentes métodos de solução possam ser avaliados de forma mais abrangente possível.

Este capítulo abordou o PRV com janelas de tempo enquanto o capítulo anterior abordou o PRV capacitado, ambos em suas variantes determinísticas, ou seja, a formulação matemática do problema não possui qualquer parâmetro estocástico. O próximo capítulo refere-se a ambos os problemas em suas variantes estocásticas, incluindo a variante estudada pela tese.

CAPÍTULO 4

PRV E PRVJT ESTOCÁSTICO

No PRV clássico, assume-se que parâmetros do problema como custos, distâncias, demanda dos clientes, tempo de viagem dos veículos entre outros, são todos conhecidos (ditos determinísticos) e não mudam (ditos estáticos). No mundo real, esta abordagem falha em capturar alguns aspectos essenciais do problema. Um aspecto dos problemas do mundo real que mostra a necessidade de modificar o modelo clássico é a incerteza. Isto significa que um ou mais parâmetros do modelo não são inicialmente conhecidos ou são conhecidos apenas probabilisticamente. Isto dá origem ao Problema de Roteamento de Veículos Estocástico (PRVE).

Outra situação ocorre quando algumas entradas do modelo se tornam conhecidas apenas quando a execução do modelo já começou, originando o Problema de Roteamento de Veículos Dinâmico (PRVD). Devido a recentes avanços nas tecnologias de informação e telecomunicação, estas classes de problemas têm se tornado importantes, dado a existência de potenciais ganhos financeiros em incluir modelos estocásticos e dinâmicos em aplicações do mundo real. O PRVD não é foco do presente trabalho. Para estudos mais aprofundados citamos: Psaraftis (1980), Psaraftis (1995), Larsen (2000) e Psaraftis et al. (2016).

Até aqui temos então condições de compreender uma classificação genérica para os problemas de roteamento. Eles podem ser do tipo determinístico ou estocástico. E também podem ser do tipo estático ou dinâmico.

Resumidamente:

Problemas de Roteamento Estáticos: Assume-se que todas as informações relevantes são conhecidas antes do início do processo de roteamento. Não há alteração das informações relevantes depois de iniciado o processo de roteamento.

Problemas de Roteamento Dinâmicos: Em oposição aos estáticos, assume-se que não são conhecidas todas as informações relevantes antes do início do processo de roteamento. As informações relevantes depois de iniciado o processo de roteamento podem sofrer alterações.

Problemas de Roteamento Determinísticos: todos os parâmetros do modelo, como tempo de viagem, demanda dos clientes e outros são assumidos conhecidos e constantes.

Problemas de Roteamento Estocásticos: Diferentemente do determinístico, o modelo assume que pelo menos 1 dos parâmetros são conhecidos ou estimados apenas probabilisticamente.

Nos capítulos 2 e 3, os modelos descritos eram essencialmente estáticos e determinísticos. E agora no capítulo 4, expandimos nossa visão para os problemas estocásticos.

4.1 – APROFUNDANDO-SE NO PRV ESTOCÁSTICO (PRVE)

O PRVE introduz elementos de incerteza nos parâmetros que são dados de entrada do modelo matemático. Isto faz com que algumas propriedades do PRV não sejam as mesmas. Uma revisão da literatura sobre o PRVE é dada em Gendreau et al. (1996) e mais recentemente por Ritzinger et al. (2016) incluindo algumas das mais importantes publicações no campo.

O PRVE pode ser brevemente classificado conforme os seguintes critérios:

- 1) Natureza da incerteza no problema: a incerteza pode estar presente em diferentes partes do problema. Podemos ter o PRV com clientes estocásticos, o PRV com demanda estocástica, o PRV com tempo de viagem estocástico e o PRV com tempo de serviço estocástico.
- 2) Método de modelagem: o método de modelagem pode ser um critério para se classificar o PRVE. O problema pode ser abordado usando-se programação estocástica, por exemplo: *Chance Constrained Program (CCP)* e o *Stochastic Program with Recourse (SPR)*. Outra possível abordagem para o modelo é vê-lo como um processo de decisão de Markov.
- 3) Técnica de solução: diferentes técnicas de solução são uma consequência da natureza da incerteza no problema e do método de modelagem. Normalmente implicam em duas categorias: métodos exatos e métodos heurísticos.

Berhan et al. (2014) numa pesquisa bibliográfica consultando diversos mecanismos de busca, encontraram mais de 3000 artigos associados a termos como “*Vehicle Routing*”, “*Vehicle Routing Problem*” ou “*Stochastic Vehicle Routing Problem*” como frases de busca, publicados em inglês. Dentre estes, apenas 49 referem-se ao PRV Estocástico, com uma primeira publicação feita em 1963. Dentre estes, 35 artigos (71%) publicados a partir de 2003, mostrando claramente um maior interesse em anos mais recentes.

Berhan et al. desenvolvem uma classificação estrutural para o PRV Estocástico baseado em oito domínios: cenário estocástico, abordagem da solução, característica dos dados, tipos de grafo, número de veículos, capacidade dos veículos, função objetivo e modelo aplicado. Para estes domínios, as versões mais comuns são respectivamente: demanda estocástica dos clientes, heurísticas, dados sintéticos (instâncias criadas pelos próprios autores ou outros autores), grafo de representação simétrica, múltiplos veículos, veículos capacitados, minimização, e finalmente quanto ao modelo aplicado, a maioria dos trabalhos não o identifica, sendo comum o “*chance constrained programming*” (*CCP*).

Programas estocásticos podem ser modelados em dois estágios. No primeiro, uma solução “*a priori*” é determinada de modo a revelar os valores dos parâmetros associados a variáveis aleatórias. Então no segundo estágio uma ação corretiva (recurso) é aplicada ao primeiro estágio da solução. O recurso geralmente cria uma economia ou um custo que deve ser considerado.

Por exemplo, seja um PRVE com demanda estocástica, onde o parâmetro determinístico d_i (demanda do cliente i) é substituído pela variável aleatória ε_i . O primeiro estágio consiste em encontrar as m rotas de veículos onde todos os clientes sejam visitados uma vez. Após o primeiro estágio ser executado, a verdadeira demanda de cada cliente torna-se conhecida. Então pode ser que realizar as rotas como planejado inicialmente tenha se tornado impossível, pois alguma restrição pode ter sido violada, como a capacidade do veículo de uma determinada rota, por exemplo. Uma possibilidade para o segundo estágio seria percorrer uma certa rota até que sua capacidade fosse violada. Então retornar ao depósito e depois continuar a visitar os clientes da rota inicial.

Como dito anteriormente, um programa estocástico é comumente modelado como um *CCP* ou *SPR*. Um *CCP* força que a probabilidade de satisfazer as restrições com

parâmetros estocásticos seja superior a um valor previamente estabelecido, contudo, os custos de ações corretivas não são considerados em sua formulação. Ao contrário, um *SPR* incorpora o custo esperado das ações corretivas do segundo estágio na função objetivo do modelo. Isto traz consideráveis dificuldades, porém é mais plausível.

Para um dado problema, ações corretivas (ou políticas recursivas) podem ser feitas de diversas formas. Para mais informações sobre as diferentes políticas que podem ser adotadas, ver Dror *et al.* (1989).

A seguir são descritas duas importantes classes de PRV estocásticos: uma com demanda e clientes estocásticos e outra com tempo de viagem estocástico.

4.1.1 – PRV com demanda e clientes estocásticos

Um PRV é dito com demanda estocástica quando pelo menos 1 dos clientes possui uma demanda associada a uma variável aleatória. A demanda do cliente é revelada apenas quando o veículo chega ao local. Dado uma capacidade fixa para o veículo, existe a possibilidade do veículo identificar não possuir capacidade suficiente para atender a demanda, sendo necessária uma viagem adicional. Uma revisão bibliográfica é feita por Dror (2016), explorando diferentes formulações do problema e descrevendo vários algoritmos usados para a construção de rotas.

Tillman (1969) propôs o primeiro algoritmo para este tipo de problema. Penalidades eram aplicadas sempre que um veículo estava vazio ou com ocupação acima de sua capacidade máxima. O modelo foi baseado numa modificação no cálculo das economias proposto por Clarke & Wright (1964). Outra importante contribuição para este tipo de problema foi dada por Bertsimas (1992). O autor considera o caso em que a demanda é 1 com probabilidade p_i e em que a demanda é 0 com probabilidade $1-p_i$.

Um PRV com cliente estocástico é aquele em que pelo menos 1 cliente tem sua presença associada a uma variável aleatória, mas sua demanda é determinística. Em alguns casos, nem os clientes são conhecidos no momento da decisão sobre os veículos e rotas, mas sabe-se que alguns clientes podem requerer serviço ao longo do dia. Antecipar-se a demandas futuras pode permitir que mais clientes sejam servidos pela frota. Ulmer et al. (2015) resolve este problema com o objetivo de maximizar o número de clientes serviços num dado período de tempo.

Numa abordagem *SPR*, os clientes não presentes no estágio são “pulados”. Muitos artigos tratam casos em que a demanda é unitária. Waters (1989) considera um caso em que a demanda não é binária e faz uma comparação entre 3 políticas propostas por ele. Uma em que o veículo segue a rota inicialmente proposta sem “saltar” clientes que se revelaram ausentes. Outra em que tais clientes são “saltados” pelo veículo. E uma terceira política em que a rota restante é re-otimizada sempre que um cliente é confirmado como “ausente”.

Duas propriedades interessantes podem ser aplicadas a ambos os problemas: demanda estocástica e clientes estocásticos. Na primeira propriedade, mesmo que os custos de viagem sejam simétricos, o custo total da solução é dependente da direção em que a rota é percorrida (Dror & Trudeau, 1986). Uma segunda propriedade diz que quanto maior a capacidade dos veículos maior será o custo da solução (Jaillet & Odoni, 1988).

4.1.2 – PRV com tempo de viagem estocástico

O PRV com tempo de viagem estocástico descreve um ambiente de incerteza intrínseco das condições de tráfego das vias. Segundo Kok et al. (2012), congestionamentos são uns dos maiores problemas para as empresas de transporte e os custos gerados podem ser reduzidos se o modelo leva em consideração tais incertezas. Kao (1978) foi o primeiro a propor uma solução a partir de uma heurística baseada em programação dinâmica e com enumeração implícita para o Problema do Caixeiro Viajante (PCV) com tempo de viagem estocástico.

Laporte *et al.* (1992) propuseram 3 modelos para abordar o problema de forma exata: um primeiro modelo *CCP*, um segundo modelo *SPR* de 3 índices e um terceiro *SPR* com 2 índices. No problema abordado, há uma duração máxima para as rotas de forma que uma penalidade proporcional é aplicada ao veículo cujo tempo ultrapassar o limite estabelecido. Tais problemas ocorrem frequentemente no contexto onde os motoristas devem ser pagos por horas extras de trabalho após o período normal. Também é o caso quando veículos são usados para coletar dinheiro das filiais para a matriz, onde todas as transações registradas após um certo horário do dia são creditadas no dia seguinte, implicando em perda financeira pelos juros envolvidos. No modelo *CCP* a função objetivo consiste em minimizar uma combinação linear dos custos das rotas e dos veículos e ao mesmo tempo assegurar que a probabilidade de que o tempo de duração de uma rota

exceda um valor B seja no máximo igual a um dado valor α . Foi apresentado um algoritmo *Branch-and-Cut* para os 3 modelos. Resultados mostraram que o segundo modelo pode ser usado para resolver de forma exata problemas de tamanho moderado.

Miranda & Conceição (2010) fazem um estudo de um problema de roteamento com tempos de viagem estocásticos e janelas de tempo, incluindo penalidades por violação na janela de tempo. Isto permitiu avaliar as rotas baseando-se na incerteza envolvida. O modelo com tempos de viagem estocásticos é comparado com um modelo determinístico simulando-se o funcionamento de uma empresa prestadora de serviços em manutenção. Também são realizados experimentos comparando instâncias capacitadas e não capacitadas, com janela de tempo e sem janela de tempo.

Comparado com os modelos com clientes e com demanda estocásticos, o modelo com tempo de viagem estocástico tem recebido menos atenção, embora não seja menos importante e sim, possivelmente, o mais interessante entre suas variantes (Gendreau, 2010). Segundo Li (2010), dentre os poucos trabalhos que tratam o tempo de viagem estocástico, a grande maioria não considera a presença de janelas de tempo para os clientes. Esta classe especial de problemas pode ser classificada como PRVJTTVE (Problema de Roteamento de Veículos com Janelas de Tempo com Tempo de Viagem Estocástico) que será melhor estudada a seguir.

4.1.3 – PRVJTTVE

Nos problemas determinísticos com janelas de tempo pode-se facilmente classificar as rotas em viáveis e não viáveis, pois assume-se conhecer o momento em que o veículo chegará em cada cliente, sendo simples checar se isto ocorre antes da abertura da janela (portanto a rota é viável, há espera e o tempo de início de atendimento fica sendo a abertura da janela), dentro da janela (portanto a rota é viável, não há espera e o tempo de início de atendimento é o mesmo do tempo de chegada) e após o fechamento da janela (rota não viável).

O PRVTVE (PRV com tempo de viagem estocástico) pode ter três variantes principais. O primeiro é o PRVTVETM (PRV com tempo de viagem estocástico e tempo máximo) em que não há janelas de tempo para os clientes, apenas para o veículo. O segundo é o PRVTVE *with time deadlines*, um tipo de problema em que só existe o tempo de fechamento das janelas, ou seja, o tempo de abertura de todas as janelas é zero. O

terceiro é o PRVJTTVE (PRV com janela de tempo e tempo de viagem estocástico), onde o tempo de abertura da janela de tempo de um ou mais clientes é diferente de zero.

No caso do PRVTVE *with time deadlines* e PRVJTTVSE, uma rota é considerada viável se a probabilidade do tempo de chegada do veículo antes do final da janela deste cliente for maior ou igual ao nível de serviço desejado, ou seja, a probabilidade do veículo não violar o final da janela de tempo deve ser maior ou igual ao nível de serviço desejado. A probabilidade de violar a janela de tempo é calculada para cada um dos clientes da rota, ou seja, cada cliente tem seu próprio nível de serviço. O nível de serviço da rota, em geral, é considerado como sendo o menor nível de serviço dos clientes. No PRVTVETM uma rota é considerada viável se a probabilidade do veículo retornar ao depósito antes do tempo máximo desejado for maior ou igual ao nível de serviço desejado.

Normalmente, para estes três tipos de problemas, assume-se conhecida, através de dados históricos coletados em campo, a função distribuição de probabilidade do tempo de viagem entre cada um dos clientes e os parâmetros desta distribuição. Assume-se ainda que os tempos de viagem entre os clientes são independentes e identicamente distribuídos (Laporte *et. al.*, 1992).

No PRVTVETM e no PRVTVE *with time deadlines*, assumindo-se as condições descritas no parágrafo anterior, a função distribuição de probabilidade do tempo de chegada em cada um dos clientes de uma dada rota pode ser facilmente determinada para vários tipos de variáveis aleatórias cuja soma resulta numa distribuição conhecida. Por exemplo, se o tempo de viagem entre os clientes segue uma distribuição Normal, então o tempo de chegada nos clientes também seguirá uma distribuição Normal. Devido a suposição de independência entre os arcos é possível somar as médias e as variâncias ao longo da rota. Conhecendo-se a distribuição do tempo de chegada num dado cliente e os parâmetros desta distribuição é possível calcular a probabilidade do veículo violar o fechamento da janela de tempo (seja de um cliente ou do depósito) e então verificar o nível de serviço.

No PRVJTTVE, a existência de janelas de tempo com tempo de abertura diferente de zero torna o cálculo do nível de serviço mais difícil. Nesta situação, diferentemente das outras duas, existe tempo de espera dos veículos. O tempo de espera depende das janelas de tempo e do tempo de chegada nos clientes. Uma vez que o tempo de chegada é uma variável aleatória, o tempo de espera também é uma variável aleatória.

Porém ela não terá necessariamente a mesma função distribuição de probabilidade dos arcos, mesmo no caso de tempos de viagem normalmente distribuídos. Isto ocorre devido a existência de não linearidade causada pelas janelas de tempo. Mesmo supondo conhecida a distribuição de probabilidade dos arcos e mesmo assumindo-se que os arcos são independentes, há 2 dificuldades neste tipo de problema: conhecer o tipo de distribuição de probabilidade do tempo de chegada nos clientes e determinar os parâmetros desta distribuição.

Como mencionado anteriormente, problemas com janela de tempo podem ser classificados em “*hard time windows*” e “*soft time windows*”. Especialmente no contexto com tempo de viagem estocástico, esta classificação demanda maior atenção. De acordo com Desaulniers et al. (2014), no caso “*hard time windows*”, um veículo que chega antes da abertura da janela de tempo, deve necessariamente esperar. Ainda conforme o autor, em geral, este tempo de espera não implica em custos. No caso “*soft time windows*”, qualquer janela de tempo pode ser violada, existindo-se a aplicação de uma penalidade. Ainda de acordo com Cordeau et al. (2002), na versão “*soft time windows*”, a janela pode ser violada dado um certo custo, enquanto “*hard time windows*” não permite que o veículo chegue depois do final da janela de tempo, e caso chegue antes da abertura, o veículo deve esperar.

Considerando as definições acima, se o problema é modelado como “*hard time windows*”, com certeza existe tempo de espera. Caso contrário, dizer que a modelagem é “*soft time windows*”, não significa necessariamente que existe tempo de espera. Por exemplo, Zhang et al. (2013) aborda o problema com “*soft time windows*” com tempo de espera e Tas et al. (2014) resolve o problema com “*soft time windows*”, porém sem tempo de espera. Uma outra observação é que para a versão determinística do PRVJT um modelo “*soft time windows*” pode também resolver problemas “*hard time windows*” elevando-se as penalidades para a violação da janela, mas isto não é necessariamente verdade para a variante com tempo de viagem estocástico. O objetivo aqui é frisar que, o importante, é saber se o modelo empregado trata ou não o tempo de espera, e não apenas dizer se é “*soft time windows*” ou “*hard time windows*”. É relevante ter isto em mente porque o foco do presente trabalho consiste em investigar o fenômeno causado pela existência de tempos de espera.

Modelos sem tempos de espera se beneficiam da propriedade da convolução na soma de variáveis aleatórias. A distribuição de probabilidade da soma de duas ou mais

variáveis aleatórias é a convolução de suas distribuições individuais. Muitas distribuições de probabilidade possuem convoluções simples, como a distribuição normal e gamma, por exemplo.

Modelos com tempos de espera não podem usufruir das propriedades da convolução. Mesmo assumindo-se que a distribuição de probabilidade para os arcos seja conhecida e independente, a distribuição do tempo de chegada possui uma forma que varia ao longo dos clientes da rota, tendo, portanto, uma forma de cálculo mais complexa.

No contexto do PRVJTVE “sem tempo de espera”, alguns trabalhos relevantes foram conduzidos por Russell & Urban (2008), Tas et al. (2013) e Tas et al. (2014). Russell & Urban resolvem o VRP com “*soft time windows*” e distribuição Erlang para o tempo de viagem, utilizando uma distribuição gamma para estimar o tempo de chegada do veículo. Penalidades são aplicadas tanto para chegadas antes da abertura quanto para depois do fechamento da janela de tempo.

Tas et al. (2013) estudam o VRP com “*soft time windows*” e distribuição Gamma para o tempo de viagem. O tempo de serviço é constante. A função objetivo considera os custos de transporte (distância total percorrida, número de veículos e a hora extra do motorista) e custos de serviço (chegadas antes da abertura e depois do fechamento da janela). Utiliza-se uma metaheurística baseada em Busca Tabu. De acordo com o autor, as distribuições de probabilidade mais comuns utilizadas para o tempo de viagem são: normal, lognormal, shifted gamma e gamma.

Tas et al. (2014) estudam o PRVJT na versão “*time-dependent*”, com tempo de viagem estocástico (distribuição Gamma) e “*soft time windows*”. Duas metaheurísticas são construídas: Busca Tabu e “*Adaptive Large Neighborhood Search*”. Sendo “*time-dependent*”, a velocidade dos veículos pode variar ao longo do dia. O horizonte de programação é dividido em intervalos, cada intervalo associado a um multiplicador. Estes multiplicadores são usados para especificar que há uma velocidade de deslocamento diferente para diferentes intervalos de tempo, em que um multiplicador maior indica que é necessário mais tempo para viajar.

No contexto do PRVJTVE “com tempo de espera”, estudos foram realizados por Jula et al. (2006), Chang et al. (2009), Li et al. (2010), Miranda (2011), Errico et al. (2013 e 2016), e Zhang et al. (2013), Ehmke et al. (2015) e Binart et al. (2016).

Em Jula et al. (2006), investiga-se não um PRV mas sim um PCV com tempo de viagem estocástico, problemas com características muito similares quanto ao efeito do tempo de viagem estocástico numa rota. No estudo, existem três situações. Na primeira, o tempo de serviço e as janelas de tempo são desconsiderados. Na segunda, o tempo de serviço é uma variável aleatória, sendo independente e identicamente distribuída entre os clientes. Na terceira situação, considera-se a existência de janelas de tempo e seu efeito é estudado. O autor sugere meios de se estimar a média e a variância do tempo de chegada do veículo usando estimativas obtidas a partir de aproximações de primeira ordem de séries de Taylor. Também estima-se limites inferiores para o nível de serviço nos clientes através das desigualdades de Chebyshev e de Chernoff. O autor realizou experimentos com algumas rotas geradas a partir de um VRP com cinco nós e comparou as estimativas de média e variância com os resultados obtidos via simulação. Segundo o autor os resultados apresentaram um erro pequeno, sendo considerados satisfatórios. Os testes também mostraram que um aumento no nível de serviço desejado implicou num aumento do custo total da solução. Também se observou que a utilização da Desigualdade de Chernoff forneceu limites inferiores para o nível de serviço mais apertados que a Desigualdade de Chebyshev.

Chang et al. (2009) também estudam um PCV com janela de tempo e com tempo de viagem estocástico. Para calcular o nível de serviço de uma rota, o autor se depara com as dificuldades citadas anteriormente. A primeira delas é estimar a média e a variância do tempo de partida do veículo. Considerando um tempo de viagem normalmente distribuído, o modelo desenvolvido no trabalho assume que o tempo de chegada no cliente também segue uma distribuição normal. Embora seu modelo ignore o efeito do tempo de espera na distribuição do tempo de chegada, em seus experimentos, 1 rota com 12 clientes foi gerada e simulada. Realizou-se um teste de adesão nos dados coletados aplicando-se o teste de Kolmogorov-Smirnov, e o autor considerou que a adesão à distribuição normal foi satisfatória. Realizou-se também, para a mesma instância de 12 clientes, um experimento com o objetivo de comparar o modelo estocástico com o determinístico. Os resultados mostram que a distância total percorrida no modelo estocástico foi maior que no modelo determinístico. Além disto, 2% das soluções geradas no modelo determinísticos eram inviáveis enquanto que no modelo estocástico isto não ocorreu.

O trabalho de Li et al. (2010) usa simulação estocástica para calcular as probabilidades envolvidas. Utiliza-se uma modelagem do tipo *CCP* e outra para o tipo

SPR. O modelo *CCP* (*Chance Constrained Programming Model*) no contexto do PRVJTTVSE consiste em planejar um conjunto de rotas que está sujeito a restrições de probabilidade com o objetivo de assegurar que a probabilidade de falha na rota seja menor que um nível desejado. Similar a outros PRVs determinísticos, o modelo *CCP* do PRVJTTVE tem funções objetivos hierarquizadas, onde o objetivo primário é a redução do número de veículos e o objetivo secundário é a minimização da distância (ou tempo) total de viagem. Isto significa que uma solução com menor número de veículos e maior distância total é considerada melhor que uma solução com maior número de veículos e menor distância total. O autor prova que para o PRVTVE *with time deadlines*, se o tempo de viagem entre os arcos são independentes e identicamente distribuídos, é possível converter as restrições probabilísticas em suas correspondentes determinísticas. Isto não é possível para o PRVJTTVE devido à existência do tempo de espera como uma variável aleatória.

Ainda em Li et al. (2010), elabora-se um modelo *SPR* (*Stochastic Programming model with Recourse*). No modelo *CCP* é necessário que o nível de serviço de cada rota seja superior a um valor α . Isto significa que quando uma rota tem nível de serviço igual a α , sendo viável, existe uma probabilidade $(1-\alpha)$ de violação da janela de tempo. Então o modelo *SPR* considera um custo para esta probabilidade $(1-\alpha)$ e insere este termo na função objetivo como um termo de penalidade. O autor desenvolveu uma heurística de roteamento baseada na Busca Tabu para resolver o problema. A verificação da viabilidade da rota quanto às restrições probabilísticas é feita através de simulação estocástica. Cada rota era simulada pelo menos 1000 vezes para se coletar os dados do tempo de chegada em cada cliente e então contabilizar a quantidade de violações em cada cliente, estabelecendo o nível de serviço da rota. Embora este método apresente uma boa qualidade para as soluções encontradas, o custo computacional da simulação das rotas é muito alto, de modo que mesmo utilizando heurísticas, apenas problemas de pequena instância podem ser resolvidos em tempo praticável.

Miranda (2011) em sua dissertação de mestrado, apresenta um estudo sobre o VRPTW com tempo de viagem normal e “hard time windows”. Analisa-se com clareza o efeito da abertura da janela de tempo como um ponto de truncamento para a distribuição do tempo de chegada que consequentemente influencia a distribuição do tempo de início do serviço. Diferentemente dos experimentos delineados por Chang et al. (2009) em que tal efeito não foi significativo para alterar a distribuição do tempo de chegada, os

experimentos foram construídos de modo a forçar o aparecimento do fenômeno em diversas situações, com diferentes graus de intensidade. O modelo assume normalidade para o tempo de chegada mas desenvolve um método para estimar o erro causado por esta suposição, e então corrigir o cálculo. O erro é estimado a partir de regressões lineares multivariadas, e diferentes variáveis independentes são estudadas para identificar aquelas com significativa correlação com a variável dependente (erro no cálculo do nível de serviço ao assumir-se normalidade).

Errico (2013) propõem uma formulação para o PRVJT com “hard time windows”, em que o tempo de viagem não é estocástico, mas sim o tempo de serviço nos clientes, o que também lida com a mesma dificuldade de tratar a mudança na distribuição do tempo de chegada ao longo de uma rota. A distância total de viagem é minimizada enquanto o nível de serviço dos clientes é atendido. Considera-se um tempo de serviço com distribuição triangular. Uma solução exata utilizando-se geração de colunas é aplicada, num caso em que as restrições de capacidade são desconsideradas. Como uma continuação de seus trabalhos, Errico et al. (2016) elabora um modelo CCP com dois estágios e duas políticas recursivas para recuperar a viabilidade das soluções obtidas no primeiro estágio. Um algoritmo *branch-cut-and-price* resolve de forma exata instâncias com até 50 clientes. Experimentos utilizaram tempo de serviço com distribuição de probabilidade discreta tanto triangular quanto uniforme. Quanto ao cálculo do nível de serviço nos clientes, o mesmo método de 2013 foi utilizado.

Ehmke et al. (2015) apresenta um método para calcular as distribuições de probabilidade do tempo de início de serviço e do tempo de chegada em cada cliente. Dada a existência de uma relação entre a janela de tempo e os tempos de viagem, aplica-se o “*extreme value theory*” para computar a distribuição do valor máximo da soma de duas variáveis aleatórias. Finalmente, Binart et al. (2016) resolve uma variante do PRV em que ambos os tempos de serviço e de viagem seguem uma distribuição de probabilidade triangular.

Zhang et al. (2013) estudam o PRVJTVE também para o caso em que o veículo precisa esperar caso chegue antes da abertura da janela de tempo. Como este é o benchmark utilizado no presente trabalho (devido aos bons resultados apresentados), ele será melhor detalhado no próximo tópico.

4.1.4 – Estudos realizados por Zhang *et al.* (2013)

Os autores estudam o PRVJTVE. Para resolver o modelo proposto, um algoritmo baseado na Busca Tabu e *Iterated Local Search* foi desenvolvido mediante a incorporação de um mecanismo de redução de rotas. Um método de aproximação discreta é proposto para gerar distribuições de tempo de chegada dos veículos na presença de janelas de tempo. Um exemplo numérico é utilizado para demonstrar a aplicabilidade do algoritmo do modelo e solução proposta.

A formulação matemática é dada a seguir:

Seja $G = (V_0, A)$ um dígrafo completo, onde $V_0 = \{0, \dots, n\}$ é o conjunto de vértices e $A = \{(i, j): i, j \in V_0, i \neq j\}$ é o conjunto de arcos. Vértice 0 representa o depósito onde m_0 veículos idênticos com capacidade Q estão disponíveis. O conjunto cliente é denotado como $V = V_0/\{0\} = \{1, \dots, n\}$. Cada cliente $i \in V$ tem uma demanda não negativa q_i , um tempo de serviço S_i , e uma janela de tempo $[e_i, l_i]$. Espera-se que o serviço em cada cliente i comece dentro de $[e_i, l_i]$. Se o veículo chega ao cliente i do local antes de e_i , tem que esperar até e_i ; se chega ao cliente i depois de l_i , uma penalidade proporcional ao atraso deve ser paga. Uma janela de tempo $[e_0, l_0]$ é também associada ao depósito, onde e_0 representa o menor tempo de partida possível a partir do depósito e l_0 representa o maior tempo de partida possível. Um tempo de viagem T_{ij} está associado com cada arco $(i, j) \in A$. Ambos T_{ij} e S_i são variáveis aleatórias com distribuições conhecidas e independentes. Suposições adicionais são: $Q \geq q_i, i \in V$ (ou seja, cada veículo pode servir pelo menos um cliente) e m_0 é grande o suficiente (ou seja, o tamanho da frota é suficiente para atender a demanda). Notação adicional é listada como segue:

M um número suficientemente grande

f custo fixo de se utilizar um veículo

m número de veículos requeridos na solução viável, $m \leq m_0$

m^* número de veículos requeridos na solução ótima $m^* \leq m_0$

K conjunto de veículos requeridos na solução viável $K = \{1, \dots, m\}$.

x_{ijk} variável binária de valor unitário quando o veículo k percorre arco (i, j)

R_k rota k definida como $R_k = \{r_0 = 0, \dots, r_{n_{k+1}} = 0\}$ onde n_k é o número de

clientes designados ao veículo (ou rota) k.

A_{r_jk} *tempo de chegada do veículo k no vértice r_j*

$T_{r_jk}^{SS}$ *tempo de início de atendimento do veículo k no cliente r_j*

d_{0k} *tempo de partida do veículo k do depósito*

D_{r_jk} *tempo de partida do veículo k do cliente r_j*

W_{r_jk} *tempo de espera do veículo k do cliente r_j*

P_{r_jk} *tempo de atraso do veículo k do cliente r_j*

B_k *duração da rota k*

h *limite superior para duração de cada rota*

Z_k *excesso de duração da rota k*

λ_{1i} *coeficiente de penalidade por chegar mais cedo ao cliente i*

λ_{2i} *coeficiente de penalidade por chegar mais tarde ao cliente i*

$\lambda_{2,0}$ *coeficiente de penalidade pelo veículo retornar mais tarde ao depósito*

λ_3 *penalidade por excesso de duração da rota*

α_i *nível de serviço requerido (probabilidade de chegar em i antes de l_i)*

α_0 *nível de serviço requerido para o depósito*

β *probabilidade da duração da rota ser menor que h*

$$\min \text{Mf. m} + \sum_{(i,j) \in A} \sum_{k \in K} E(T_{ij}) x_{ijk} + \sum_{k \in K} \left(\sum_{j=1}^{n_k} \lambda_{1r_j} E(W_{r_jk}) + \sum_{j=1}^{n_{k+1}} \lambda_{2r_j} E(P_{r_jk}) + \lambda_3 E(Z_{r_jk}) \right) \quad (26)$$

Sujeito às seguintes restrições:

$$\sum_{j \in V_0} \sum_{k \in K} x_{ijk} = 1, \forall i \in V \quad (27)$$

$$\sum_{j \in V} x_{0jk} = 1, \forall k \in K \quad (28)$$

$$\sum_{i \in V} x_{i0k} = 1, \forall k \in K \quad (29)$$

$$\sum_{i \in V_0} x_{ijk} - \sum_{i \in V_0} x_{jik} = 0, \forall j \in V, k \in K \quad (30)$$

$$\sum_{i \in V} q_i \sum_{j \in V_0} x_{ijk} \leq Q, \forall k \in K \quad (31)$$

$$P(A_{r_j k} \leq l_{r_j}) \geq \alpha_{r_j} \quad (32)$$

$$P(B_k \leq h) \geq \beta, \forall k \in K \quad (33)$$

A equação da função objetivo (26) consiste em três partes: 1. Custo fixo do veículo, 2. Tempo médio total de viagem, 3. Penalidade por chegar mais cedo e mais tarde nos clientes, e mais tarde no depósito. O modelo possui função objetivo hierárquica: o principal objetivo é minimizar o número de veículos necessários para satisfazer as restrições (27) à (33); O objetivo secundário é o de minimizar os custos operacionais e de penalidades. Isto implica que uma solução com menos rotas, mas superior custo operacional é melhor que outra com mais rotas, mas menor custo operacional.

A equação (27) indica que cada cliente deve ser visitado exatamente uma vez por um veículo. Equações (28) e (29) asseguram que cada veículo começa e termina sua rota no depósito. A equação (30) garante que cada veículo saia do cliente depois de tê-lo visitado. A equação (31) é a restrição de capacidade. A equação (32) é o nível de serviço do cliente, restrição que assegura que a probabilidade de entrega a cada cliente é de pelo menos um α_i com valor pré-definido. A equação (33) garante que cada percurso é concluído dentro de um tempo h com probabilidade maior ou igual a β .

Uma heurística baseada no método *Iterated Tabu Search* é implementada. Para o PRVJT, $A_{r_{j+1},k}$ pode ser calculado de forma determinística, porém para o PRVTVE determinar $A_{r_{j+1},k}$ é difícil. O autor também mostra que simplesmente assumir normalidade pode implicar em erros não aceitáveis.

O método " α - discrete" baseado no trabalho de Miller-Hooks & Mahmassani (1998) é adaptado para estimar a distribuição do tempo de chegada do veículo na presença de janelas de tempo. O tempo de viagem é assumido ser conhecido probabilisticamente, seja por uma distribuição normal, seja por uma lognormal. Um pseudocódigo do método é dado abaixo. Para detalhes, consultar Zhang (2013).

- 1) Criar um vetor $vF(x)$ com L valores da função de distribuição acumulada $F(x)$ uniformemente distribuídos entre 0 e 1, onde $p = 1, \dots, L$ e $\xi \leftarrow 1/L$ e $vF(x)_p = vF(x)_{p-1} + \xi$. Exemplo: Para $L = 10$, $vF(x) = [0.05, 0.15, 0.25, \dots, 0.95]$.
- 2) Analogamente, criar vetor $vF2(x)$ com L^2 valores da função de distribuição acumulada $F(x)$ uniformemente distribuídos entre 0 e 1, onde $p = 1, \dots, L^2$ e $\xi \leftarrow 1/L^2$.
- 3) Para o cliente i de uma dada rota, sendo o tempo de viagem $TT_{ij} \sim N(\mu_i, \sigma_i^2; 0)$, gerar o vetor vTT com L valores de tempo de viagem de i para j . Exemplo: $vTT = [F_{(0,05)}^{-1}, F_{(0,15)}^{-1}, \dots, F_{(0,95)}^{-1}]$.
- 4) Criar matriz $mTCH$ com dimensões $L \times L$ para o tempo de chegada em j : $mTCH_{p,q} = vTP_p + vTT_q$ onde $p = 1, \dots, L$; $q = 1, \dots, L$ e vTP uma matriz com o tempo de partida no cliente i obtido na iteração anterior. Na primeira iteração, o cliente i é o depósito, ou seja, $vTP_p = 0$.
- 5) Transformar $mTCH_{p,q}$ (matriz $L \times L$) num vetor ordenado $vTCH_p$ com L^2 valores de tempo de chegada.
- 6) Calcular a média e o desvio padrão do tempo de chegada no cliente j ($E[TCH_j]$ e $DEV[TCH_j]$), a partir dos valores da matriz $vTCH$.
- 7) Percorrer o vetor ordenado $vTCH$ calculado no passo 5 a partir da posição $p = 1$, enquanto $vTCH_p \leq l_j$, onde l_j é o fechamento da janela de tempo do cliente j . Após sair do loop, computar o nível de serviço no cliente j , como $NS_j = P(TCH_j \leq l_j) = vF2_p$. Exemplo, sendo $L = 10$, $l_j = 156$, $vTCH = [tch_1, tch_2, \dots, tch_{91}, tch_{92}, \dots, tch_{100}] = [98.5, 99.8, \dots, 155.5, 156.8, \dots, 163.0]$ e $vF2 = [0.005, 0.015, \dots, 0.905, 0.9150, \dots, 0.995]$, então $NS_j = 0.905 = 90.5\%$.
- 8) Percorrer o vetor $vTCH$ de $p = 1$ a $p = L^2$ e obter o vetor $vSST$ com os valores do tempo de início de atendimento no cliente j , onde $vSST_p = e_j$ se $vTCH_p < e_j$, e $vSST_p = vTCH_p$, caso contrário. Onde e_j é a abertura da janela de tempo no cliente j .

- 9) Transformar o vetor $vSST$ com L^2 elementos no vetor $vSST2$ com L elementos, calculando a média dos valores a cada L posições.
- 10) Para o cliente j , sendo o tempo de serviço $ST_j \sim N(\mu_i, \sigma_i^2; 0)$, gerar o vetor vST com L valores de tempo de serviço em j .
- 11) Criar matriz mTP com dimensões $L \times L$ para o tempo de partida do cliente j :

$$mTP_{p,q} = vSST2_p + vST_q \text{ onde } p = 1, \dots, L ; q = 1, \dots, L .$$
- 12) Transformar mTP num vetor ordenado vTP com L valores de tempo de partida, calculando a média de cada L valores em mTP . Fazer $i = j$, onde $j = i + 1$, e retornar ao passo 3.

O autor compara os resultados obtidos com Chang *et. al* (2009) e Li *et. al* (2010). A qualidade das soluções do método " $\alpha - discrete$ " mostram-se melhores para os experimentos realizados, e não há um estudo referente aos tempos computacionais.

4.2 – DISCUSSÃO DO CAPÍTULO 4

De fato, poucos trabalhos na literatura investigam de forma profunda o PRVJT com Tempo de Viagem Estocástico, particularmente para o caso mais desafiante: com tempo de espera do veículo. A presença do tempo de espera como uma variável aleatória faz com que os tempos de chegada nos clientes sigam uma função densidade de probabilidade variável ao longo dos clientes de uma rota.

Li *et al.* (2010) resolve este problema via simulação estocástica. Os resultados quanto à qualidade da solução são bons, mas é fácil perceber que o custo computacional é muito alto. No modelo determinístico, quando deseja-se checar a viabilidade de uma rota quanto ao respeito às janelas de tempo, percorre-se o vetor com os clientes da rota uma única vez. No modelo estocástico, isto é feito pelo menos 1000 vezes para que a amostragem dos dados seja suficientemente grande. Num algoritmo aplicado a resolver um VRP, esta verificação é feita muitas vezes, então simular todas estas rotas faz com que seja possível resolver apenas problemas pequenos mesmo com o uso de heurísticas.

Para contornar este problema (alto custo computacional da simulação), Jula *et al.* (2006) e Chang *et al.* (2009) propuseram métodos para estimar a média e a variância do tempo de chegada nos clientes. Também foram propostos meios de calcular a probabilidade de violação das janelas sem a necessidade da simulação.

No caso de Jula *et al.* (2006), embora o autor tenha considerado o resultado dos experimentos satisfatórios, nota-se que os experimentos foram realizados a partir de uma única instância de 5 nós, com janelas de tempo largas (cerca de 40% da janela do depósito). Em Chang *et al.* (2009), limitação similar ocorre. Os experimentos foram realizados a partir de 1 instância para o PCV com 12 clientes. As janelas dos clientes eram largas (50% da janela do depósito). Além disto, realizou-se um teste de adesão para a distribuição normal que pode ser considerado frágil pois a instância utilizada é muito simples e o nível de confiabilidade utilizado no teste de hipótese é baixo (85%, quando o mais utilizado é 90% ou 95%). Em ambos os casos, o fato das janelas de tempo serem muito largas, fez com que o efeito do tempo de espera na distribuição dos tempos de chegada não fosse significativo.

Errico *et al.* (2013 e 2016), estudam o problema assumindo apenas o tempo de serviço como variável estocástica, e utilizando-se apenas distribuições triangular e uniforme, o que consideramos limitado para termos práticos. Ehmke (2016) também utiliza apenas distribuições triangulares.

Desta forma, percebe-se que os trabalhos anteriores a 2013 (e anteriores ao início do presente trabalho) não conseguiram dar um tratamento adequado ao efeito do tempo de espera no tempo de chegada do veículo. Como calcular a probabilidade de violação da janela de tempo dos clientes de modo eficiente?

O trabalho de Zhang *et al.* (2013) faz avanços na resposta a esta pergunta ao adaptar com sucesso o método " $\alpha - discrete$ " para estimar o tempo de chegada do veículo. Claramente o método depende fortemente do parâmetro L utilizado. Quanto maior o incremento na discretização da função distribuição de probabilidade, melhor a qualidade e pior o tempo de execução. Porém mesmo que este método seja mais rápido que simulação, dado que o PRVJTTVSE é *NP-hard*, faz-se muito importante encontrar um método que permita fazer as estimativas rapidamente. Esta é uma das lacunas que o presente trabalho deseja explorar, investigando alternativas para estimar o tempo de chegada do veículo de modo eficiente e eficaz.

Este capítulo trouxe detalhes de variantes do PRV com parâmetros estocásticos. Buscou-se deixar claro que dentre os modelos com tempo de viagem/serviços estocásticos, faz-se importante separá-los em dois grupos: com e sem espera do veículo. Modelos sem tempo de espera podem usufruir de convoluções já conhecidas e por esta

facilidade tendem a usar um leque maior de distribuições de probabilidade como normal, gamma e lognormal. Já os modelos com tempo de espera possuem a dificuldade de não terem uma forma conhecida para a convolução da soma de variáveis, e tendem a usar funções de distribuição mais simples como uniforme e triangular, ou ao usar distribuições mais complexas, utilizam métodos baseados em simulação que tendem a apresentar maior custo computacional.

CAPÍTULO 5

PRV & PRVJT MULTIOBJETIVO

5.1 – PRV MULTIOBJETIVO

Muitos problemas do mundo real apresentam uma coleção de objetivos a serem otimizados que são, na maioria das vezes, conflitantes entre si, ou seja, a melhoria de um objetivo causa a deterioração de outro. Seja um exemplo genérico, a compra de um produto. A melhor compra pode ser dada pela melhor relação custo-benefício entre a qualidade e o preço do produto. Deseja-se ter um produto de alta qualidade e baixo preço, mas normalmente um produto de alta qualidade tem preço alto e analogamente, um produto de preço baixo tem qualidade ruim. Isto também acontece nos problemas de roteamento do mundo real. Podemos encontrar um solução com a menor distância total percorrida, no entanto esta mesma solução pode apresentar características práticas indesejadas como maior tempo necessário para percorrer as rotas ou um grande desbalanceamento nas rotas, onde a diferença entre a distância da maior rota e da menor rota são grandes.

A teoria de Otimização Multiobjetivo também conhecida como Otimização Multi-critério ou Vetorial é vasta e não cabe ser detalhada aqui. Maiores informações podem ser obtidas em Deb (2001) e Deb (2014). O NSGA-II mostrado em Deb et al., (2002) é um famoso algoritmo genético utilizado para resolver problemas multiobjetivos.

A tomada de decisão em problemas multiobjetivos pode se dar em três formas:

Decisão “a priori”: o decisor é consultado antes da execução do algoritmo. Em geral transforma-se o problema multiobjetivo em mono-objetivo, usando-se alguma técnica de tomada de decisão para agregar os objetivos em um único valor escalar que represente as preferências do decisor. Uma forma de se fazer isto é dar pesos a cada um dos objetivos. Uma vantagem é a redução do custo computacional e uma desvantagem está no fato do decisor articular suas preferências sem conhecer suas alternativas. Qualquer mudança nas preferências do decisor exigirá uma nova execução de todo o algoritmo.

Decisão “a posteriori”: o decisor é consultado após a execução de um algoritmo de busca multiobjetivo. Uma vantagem é o fato do decisor articular suas preferências tendo todo o conjunto de alternativas em mãos. Uma mudança de interesses do decisor não exige a execução do algoritmo de busca novamente. A desvantagem é um maior custo computacional.

Decisão iterativa: o decisor é consultado durante o processo de busca e as informações sobre sua preferência são usadas para guiar o algoritmo de otimização. Há um baixo custo computacional se o decisor não mudar suas preferências muitas vezes. O decisor pode definir suas preferências já tendo algum conhecimento de suas alternativas.

No exemplo da melhor compra citado anteriormente, um produto que tenha o menor preço, ou seja tenha a melhor avaliação neste objetivo, dificilmente terá a melhor qualidade, ou seja, dificilmente vencerá neste segundo objetivo. Portanto, não existe uma solução ótima única e sim um conjunto de soluções. Tais soluções são ótimas porque não existem outras soluções no espaço de busca melhores do que elas, quando todos os objetivos são simultaneamente considerados, sendo conhecidas como soluções ótimas de Pareto.

Assim, nenhuma solução que tenha menor custo e menor qualidade pode ser considerada como superior a outra com maior custo e maior qualidade. Contudo, dentre todos os tipos de produtos, existem alguns que são superiores a outros, isto é, apresentam qualidade maior ou equivalente por um custo menor ou igual. Estas configurações (soluções) que superam outras são chamadas soluções não-dominadas, enquanto que as configurações que são superadas por pelo menos uma outra são as soluções dominadas.

O conceito de dominância pode então ser descrito da seguinte forma: sejam x e y duas soluções. Dizemos que x domina estritamente y se x é melhor que y em todos os objetivos. A solução x domina y se x não é pior que y em todos os objetivos e é melhor em pelo menos um objetivo. A solução x domina fracamente y se x não é pior que y em todos os objetivos. Dizemos que x é indiferente a y se nem x domina fracamente y e nem y domina fracamente x .

Nicolas *et. al.* (2008) apresenta uma *survey* que descreve os métodos multiobjetivos usado para tratar vários tipos de objetivos. Segundo os autores, os problemas multiobjetivos existentes são classificados de acordo com a sua aplicação (extensão, generalização, ou estudo de caso real), e os objetivos são classificados de

acordo com os componentes dos problemas a que eles estão relacionados (turismo, nó / arco ou recursos).

Com base nesse levantamento, duas estratégias principais parecem ser o mais utilizado para a resolução de problemas multiobjetivos. A primeira baseia-se em métodos escalares, um tipo de agregação ponderada das funções objetivos, e a segunda baseia-se em algoritmos evolucionários multiobjetivos. O generalizado uso de métodos escalares é, sem dúvida, ligado ao fato de permitir as agregações dos diferentes objetivos num único objetivo, assim os métodos a serem utilizados não precisam ser adaptados à natureza multiobjetivo dos problemas.

Jie (2010) resolve um problema de roteamento com janelas de tempos e com tempo de viagem estocástico. São utilizadas três funções objetivos. Uma para minimizar o tempo de atraso dos veículos em termos das janelas de tempo. Outra para minimizar o tempo total da viagem, e uma terceira função objetivo para minimizar a distância total percorrida pelos veículos. Os três objetivos são ponderados em um único objetivo e então um algoritmo genético modificado é aplicado para resolver o problema.

Garcia (2011) resolve um problema similar ao de Jie (2010), porém o tempo de viagem é determinístico. Utiliza-se as mesmas três funções objetivos, porém a decisão de otimização é do tipo “a posteriori” de modo que os três objetivos não são transformados num único objetivo. Um algoritmo evolucionário é desenvolvido e seu desempenho é comparado com instâncias da literatura, obtendo ótimos resultados. Os operadores genéticos utilizados são bem detalhados e uma extensa revisão da literatura sobre métricas utilizadas para comparar o desempenho de problemas multiobjetivos é realizada.

Demira et al. (2014) estuda uma variante do PRV Multiobjetivo chamada “*bi-objective Pollution-Routing Problem*” que além de rotear os veículos para servirem os clientes, também determina as velocidades dos veículos em cada segmento da rota. Um objetivo minimiza o consumo de combustível e outro minimiza o tempo total das rotas. Utiliza-se uma metaheurística “*adaptive large neighborhood search algorithm*” (ALNS) para quatro estratégias multiobjetivos do tipo “a posteriore”: “*weighting method*”, “*weighting method*” com normalização, “*epsilon-constraint*” e o método híbrido proposto pelos autores. Realiza-se experimentos com 100 clientes e “hipervolume” como indicador de desempenho.

O próximo tópico aborda o PRV Multiobjetivo na sua versão com Janela de Tempo, mais próximo do foco do presente trabalho.

5.2 – PRVJT MULTIOBJETIVO

Segundo Baños *et al.*(2013), a formulação multi-objetiva mais estudada é de minimizar simultaneamente o número de veículos utilizados e a distância de viagem. Em geral, estes dois objetivos são transformados num único numa abordagem em duas fases, onde o número de rotas é minimizado na primeira etapa e a distância total da viagem é então minimizada na segunda, isto é, minimizar o número de veículos é considerado o principal objetivo, enquanto que para o mesmo número de veículos, a distância total percorrida é usada frequentemente como o objetivo secundário.

O trabalho de Baños *et al.*(2013) tem como objetivo resolver o PRVJT com balanceamento de carga (chamado de WB-PRVJT), onde não só a distância total percorrida pelos veículos da frota é minimizada (DV), mas também o desbalanceamento da carga de trabalho (DCT). O desequilíbrio de carga de trabalho é analisado a partir de dois pontos de vista: o desequilíbrio das distâncias percorridas por parte dos veículos usados, e desequilíbrio nas cargas destes. Apresenta-se uma nova abordagem multi-objetiva baseada em fronteira pareto que se usa *simulated annealing* para resolver uma formulação multi-objetiva “a posteriore” que visa minimizar a distância total de veículos utilizados para atender os clientes, ao mesmo tempo minimizando o desbalanceamento das cargas de trabalho (distâncias percorrida / quantidade veículos). Esta nova abordagem é avaliada em comparação com dois conhecidos algoritmos multiobjetivos evolucionários: NSGA-II (Deb, Agrawal, Pra-tap, e Meyariva n, 2001) e SPEA2 (Zitzler, Laumanns, & Thiele, 2001).

O problema é modelado matematicamente a seguir.

O PRVJT pode ser modelado como uma grafo completo não direcionado $G(V, E)$ onde os vértices $V=\{1, \dots, N\}$ correspondente ao deposito e clientes, e os arcos $e \in E\{(i, j): i, j \in V\}$ como sendo as conexões entre eles.

Variáveis de Decisão:

$$X_{ij}^k = \begin{cases} 1, & \text{if veículo } k \text{ viaja de } i \text{ até } j \\ 0, & \text{caso contrário} \end{cases}$$

Parâmetros:

a_j é a abertura da janela de tempo no cliente j

b_j é o final da janela de tempo no cliente j

C_{ij} é o custo de viagem do nó i até j (seja tempo ou custo)

d_j é a demanda do cliente j

K é o número máximo de veículos disponíveis na frota

N é o número de clientes incluindo o depósito.

Q é a capacidade de cada veículo (frota homogênea)

Funções Objetivas:

$$\text{minimize: } DV = \max \left(\sum_{i=1}^N \sum_{j=2}^N X_{ij}^k C_{ij}^k \right) - \min \left(\sum_{i=1}^N \sum_{j=2}^N X_{ij}^k C_{ij}^k \right) \quad (\forall k \in K) \quad (34)$$

$$\text{minimize: } DCT = \max \left(\sum_{i=1}^N \sum_{j=2}^N X_{ij}^k d_j \right) - \min \left(\sum_{i=1}^N \sum_{j=2}^N X_{ij}^k d_j \right) \quad (\forall k \in K) \quad (35)$$

Restrições:

$$X_{ii}^k = 0 \quad (\forall i \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\}) \quad (36)$$

$$X_{ij}^k \in \{0,1\} \quad (\forall i, j \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\}) \quad (37)$$

$$\sum_{i=1}^N \sum_{j=2}^N X_{ij}^k = 1 \quad (\forall j \in \{2, \dots, N\}) \quad (38)$$

$$\sum_{i=1}^N \sum_{j=2}^N X_{ij}^k d_j \leq Q \quad (\forall k \in \{1, \dots, K\}) \quad (39)$$

$$\sum_{i=1}^N \sum_{j=2}^N X_{1j}^k \leq K \quad (40)$$

$$\sum_{j=2}^N X_{1j}^k - \sum_{j=2}^N X_{j1}^k = 0 \quad (\forall k \in \{1, \dots, K\}) \quad (41)$$

$$a_j \leq S_{kj} \leq b_j (\forall i, j \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\}) \quad (42)$$

$$S_{ki} + C_{ij} - L(1 - X_{ij}^k) \leq S_{kj} (\forall i, j \in \{1, \dots, N\}, \forall k \in \{1, \dots, K\}) \quad (43)$$

Eq. (34) e (35) são os objetivos do problema. Eq. (36) denota que um veículo deve viajar de um nó para outro nó diferente. Eq. (37) indica que X_{ij}^k é igual a 1 se o veículo k vai do nó i para nó j , e é igual a 0 caso contrário. Eq. (38) afirma que um cliente é visitado uma única vez e por exatamente um veículo. A restrição (39) assegura o respeito à capacidade do veículo. Equação (40) assegura que até K veículos podem partir do depósito. A equação (41) tem o propósito de garantir que os veículos retornem ao depósito. Seja S_{kj} a distância percorrida pelo veículo k antes de chegar no cliente j , a restrição (42) assegura o respeito à janela de tempo. Seja L um valor grande, a restrição (43) assegura que o veículo k ao partir de i e ir para j , não pode chegar em j antes de $S_{ki} + C_{ij}$. A variável S_{kj} é o tempo de início de atendimento do veículo k no cliente j . Se o veículo k não atende o cliente j então não precisa ser calculado.

Uma vez que tanto a convergência quanto à diversidade são importantes na solução do problema, a qualidade dos conjuntos de soluções não dominadas obtidas é avaliada por meio de três indicadores: a cobertura de conjunto de soluções, o hipervolume, que são baseados em ideias tiradas das métricas propostas por Zitzler e Thiele (1999) e o espalhamento das soluções na fronteira (proposta por Deb *et al.* 2001). Uma vez que os objetivos a serem otimizados têm diferentes escalas (por exemplo, os valores de distância de viagem são maiores do que os de desbalanceamento de carga), os valores devolvidos pelas três métricas foram normalizados no intervalo $[0, 1]$.

O trabalho de Baños *et al.* (2013) alcança resultados altamente competitivos para o conjunto de soluções não dominadas em termos de hipervolume, cobertura e espalhamento, superando os obtidos pelo NSGA-II e SPEA2. Além disso, o autor afirma que a pesquisa pode ser útil aos pesquisadores interessados em resolver formulações multiobjetivas do problema de roteamento de veículos e suas variantes, aumentando assim a aplicabilidade deste problema para o mundo real.

Qi *et al.* (2015) aperfeiçoam uma metaheurística conhecida como MOEA/D (*multi-objective evolutionary algorithms based on decomposition*) para resolver o PRVJT biobjetivo em que um objetivo é minimizar o número de veículos e o outro é minimizar a distância total percorrida. O MOEA/D decompõem o problema principal num conjunto

de subproblemas escalares com vetores de peso uniformemente distribuídos. Então cada subproblema é minimizado simultaneamente, evoluindo-se suas respectivas populações de soluções. Um novo operador de seleção é proposto para atualizar a melhor solução corrente de cada subproblema. Além disto, três novos operadores de busca local são propostos, explorando diferentes estruturas de vizinhança e formando um algoritmo memético.

Embora os objetivos mais utilizados envolvam o número de veículos e distância/tempo total das rotas, estudos recentes abordam outros tipos de objetivos. Sivaramkumar et al. (2015) resolve o PRVJT biobjetivo, sendo o primeiro objetivo, o mais comum: primeiro minimizar o número de veículos e então a distância total percorrida. O segundo objetivo é minimizar o gap total entre o tempo de chegada do veículo e a abertura da janela de tempo. Quanto mais perto da abertura da janela de tempo o veículo chegar, melhor. O autor testa o desempenho do algoritmo genético desenvolvido na versão mono-objetivo em instâncias de Solomon (1987), e após atestar o bom desempenho, adiciona o segundo objetivo para realizar os experimentos multiobjetivos. Os resultados mostram que é possível melhorar o serviço aos clientes, chegando mais próximo da abertura da janela, sem penalizar tanto o número de veículos e a distância total percorrida.

Outro estudo envolvendo PRVJT biobjetivo é feito por Urquhart et al. (2015) envolvendo outro tipo de objetivo. Além de minimizar o tempo total de viagem, deseja-se minimizar as emissões de CO₂ na atmosfera. Ou seja, trata-se de uma variante do “Green VRP”, que tem emergido com força nos últimos anos em virtude da crescente preocupação com o aumento do efeito estufa. NSGA-II é usado para encontrar um conjunto de soluções não dominadas capazes de oferecer múltiplas opções ao tomador de decisão, confrontando tempo e emissão de gases.

5.3 – DISCUSSÃO DO CAPÍTULO 5

Este capítulo trouxe uma revisão de problemas de roteamento multiobjetivos com e sem janelas de tempo. Percebe-se que a maior diferença entre os trabalhos está nos objetivos utilizados na formulação matemática ou nos diferentes métodos utilizados, sejam métodos aproximativos ou exatos. Não se encontrou na literatura trabalhos que

utilizem otimização multiobjetivo com o tempo de serviço e/ou tempo de viagem estocásticos, o que evidencia a contribuição científica do presente trabalho.

O próximo capítulo irá avançar especificamente para o problema proposto pela tese: PRVJT Multiobjetivo com Tempo de Viagem/Serviço Estocástico. Será apresentado o método estatístico para calcular o nível de serviço e a seguir o algoritmo de otimização usado para resolver o problema.

CAPÍTULO 6

PROBLEMA PROPOSTO: PRVJT COM TEMPO DE VIAGEM/SERVIÇO ESTOCÁSTICO MULTI OBJETIVO (PRVJT TVSEM)

O presente trabalho tem como objetivo resolver o PRVJT TVSEM utilizando a teoria de otimização multiobjetivo numa abordagem “a posteriore”. Deseja-se explorar a relação de custo-benefício existente entre minimizar os custos de transporte e maximizar o nível de serviço criando-se uma versão multiobjetivo do problema. É natural perceber que quanto mais se reduz o tempo de viagem do veículo, menores são as folgas para que o veículo respeite a janela de tempo dos clientes, conseqüentemente menor tende a ser o nível de serviço, ou seja, menor é a probabilidade do veículo chegar antes do fechamento da janela de tempo.

Este capítulo mostra o modelo matemático do problema e explica os estudos realizados para entender a influência da janela de tempo no tempo de chegada do veículo. Serão vistas as técnicas desenvolvidas para atingir os dois objetivos principais deste trabalho: aperfeiçoar o modo de se calcular o nível de serviço dos clientes e resolver o problema multiobjetivo proposto, até então inédito na literatura.

Inicialmente apresentar-se-á o modelo matemático do problema e a seguir iremos mostrar que a existência do tempo de espera como variável aleatória faz com que, em muitas situações, a função densidade de probabilidade da variável aleatória “tempo de chegada” não tenha adesão à distribuição normal. Então, estudam-se formas de se calcular a média e o desvio-padrão desta variável e, ainda, meios de melhorar a estimativa da probabilidade de violação da janela de tempo. Finalmente, aplica-se o conhecimento adquirido nos passos anteriores para a solução do PRVJT Multiobjetivo com Tempo de Viagem/Serviço Estocástico.

6.1 – O PROBLEMA: PRVJTTVSEM

Seja um grafo $G = (V_0, A)$ um dígrafo completo, onde $V_0 = \{0, \dots, n\}$ é um conjunto de vértices e $A = \{(i, j): i, j \in V_0, i \neq j\}$ é um conjunto de arcos. O vértice 0 representa o depósito onde m_0 veículos com capacidade Q estão disponíveis. O conjunto de clientes é $V = V_0/\{0\} = \{1, \dots, n\}$. Cada cliente $i \in V$ possui uma demanda não negativa q_i , tempo de serviço ST_i , e uma janela de tempo $[e_i, l_i]$, onde e_i é a abertura da janela de tempo e l_i é o fechamento da janela de tempo. O início do tempo de atendimento ao cliente deve ocorrer dentro da janela $[e_i, l_i]$. Se o veículo chegar ao cliente i antes de e_i , faz-se necessário esperar até e_i . O tempo de viagem TT_{ij} é atribuído a cada arco $(i, j) \in A$. Ambos TT_{ij} assim como ST_i são variáveis aleatórias independentes e com função densidade de probabilidade normal. $SL_i = P(AT_i \leq l_i)$ é o nível de serviço no cliente i . O conjunto $SL = \{SL_1, \dots, SL_n\}$ é formado pelo nível de serviço de cada cliente. Outras delimitações são: $Q \geq q_i, i \in V$ (cada veículo tem capacidade suficiente para atender qualquer cliente isoladamente) e a frota tem capacidade para atender a todos os clientes, ou seja, $m_0 * Q \geq \sum_{i=1}^n q_i$. Não há janela de tempo para o depósito, ou seja, $[e_0, l_0] = [0, \infty]$. Demais notações são descritas a seguir:

- f custo fixo do veículo
- m número de veículos numa dada solução, $m \leq m_0$
- K conjunto de veículos numa dada solução $K = \{1, \dots, m\}$.
- x_{ijk} variável binária de valor 1 quando o veículo k serve o arco (i, j)
- c custo fixo de cada unidade do tempo de viagem TT_{ij}
- AT_i momento da chegada ao cliente i
- SS_i tempo de início do atendimento ao cliente i
- α_i nível de serviço requerido pelo cliente i

O modelo do problema é dado a seguir:

$$\text{Min } f \cdot m + \sum_{(i,j) \in A} \sum_{k \in K} E(TT_{ij}) \cdot c \cdot x_{ijk} \quad (44)$$

$$\text{Min } - E[SL] \quad (45)$$

Sujeito a:

$$\sum_{j \in V_0} \sum_{k \in K} x_{ijk} = 1, \forall i \in V \quad (46)$$

$$\sum_{j \in V} x_{0jk} = 1, \forall k \in K \quad (47)$$

$$\sum_{i \in V} x_{i0k} = 1, \forall k \in K \quad (48)$$

$$\sum_{i \in V_0} x_{ijk} - \sum_{i \in V_0} x_{jik} = 0, \forall j \in V, k \in K \quad (49)$$

$$\sum_{i \in V} q_i \sum_{j \in V_0} x_{ijk} \leq Q, \forall k \in K \quad (50)$$

$$P(AT_i \leq l_i) \geq \alpha_i, \forall i \in V \quad (51)$$

$$e_i \leq SS_i, \forall i \in V \quad (52)$$

A função objetivo (44) é o custo operacional de transporte, formada por duas partes: custo fixo do veículo (número de veículos m multiplicado pelo custo fixo unitário f) e o custo variável (representado pela soma da média do tempo de viagem de cada arco ativado na solução multiplicado pelo custo unitário c). Desta forma, ambos os termos possuem a mesma unidade de custo. A função objetivo (45) maximiza o nível de serviço da solução, dado pela média do nível de serviço de cada cliente.

A equação (46) assegura que cada cliente é visitado apenas uma vez e por apenas um veículo. Equações (47) e (48) asseguram que cada veículo inicia a rota no depósito e retorna a este. A equação (49) assegura que cada veículo deixe o cliente o qual acabou de visitar. A equação (50) é a restrição de capacidade. A equação (51) trata o tempo de serviço requerido por cada cliente. Assegura-se que a probabilidade do veículo chegar ao cliente antes do fechamento de sua respectiva janela de tempo seja maior ou igual a um valor especificado (parâmetro do problema). A equação (52) assegura que o atendimento ao cliente irá começar apenas depois da abertura da janela de tempo, então se o veículo chegar antes de e_i , ele deve esperar até e_i . O tempo de início de atendimento é dado pela equação (53).

$$SS_i = \max\{AT_i, e_i\}. \quad (53)$$

É válido notar que o tempo de partida do depósito para qualquer veículo é zero. Um tempo de partida maior que zero não melhora quaisquer dos objetivos, em particular, não melhora a função objetivo (44) pois não há penalidade para o tempo de espera do veículo (um caso comum para os problemas do tipo “hard time windows”, segundo Desaulniers et al., 2014).

6.2 – SUPOSIÇÃO DE NORMALIDADE

Para muitos problemas práticos, o tempo de viagem dos arcos segue uma distribuição de probabilidade normal (Kenyon & Morton, 2003; Jie, 2010). Foi visto no capítulo 4 que alguns trabalhos consideram que o tempo de chegada do cliente pode ser aproximado por uma distribuição normal, como Chang *et al.* (2009). Vamos investigar com profundidade se esta suposição é plausível através da rota descrita na Tabela 3. Nesta tabela, têm-se os dados de uma rota com 8 clientes. Na primeira coluna temos a descrição dos dados: abertura da janela, fechamento da janela, tempo de serviço (atendimento), tempo do arco (dado determinístico original da instância assumido aqui como a média teórica do tempo de viagem do cliente $i-1$ ao cliente i), desvio-padrão do arco (assumido como 20% do tempo do arco), tempo de chegada determinístico, tempo de chegada real, desvio-padrão do tempo de chegada real, tempo de início de atendimento real e desvio-padrão real do tempo de início de atendimento. Os valores ditos “reais” foram obtidos via simulação.

A rota foi obtida através da aplicação da heurística II de Solomon (1987) descrita no tópico 3.2.1 aplicada à instância RC101 de Solomon (1987). Os valores da janela de tempo de alguns clientes foram modificados em relação à instância original para melhor ilustração do fenômeno investigado. Considera-se a velocidade do veículo unitária, de modo que o tempo e distância passam a ter os mesmos valores. Assume-se que o tempo seja dado em minutos e que o tempo de serviço nos clientes seja determinístico.

Tabela 3: Rota exemplo

	índice	1	2	3	4	5	6	7	8
Abert. Jan.	e	35	45	72	58	105	119	142	149
Fech. Jan.	l	65	55	102	88	110	124	172	173
T. Atend.	s	10	10	10	10	10	10	10	10
T. Arco	t_{ij}	35,36	3,00	5,39	2,00	9,43	5,00	7,07	11,18
Desv. Arco	σ_{ij}	7,07	0,60	1,08	0,40	1,89	1,00	1,41	2,24
T. Cheg. Det.	TCHi	35,36	48,36	63,74	84,00	103,43	120,00	137,07	163,18
T. Cheg. Real	TCHi	35,36	50,98	66,36	84,44	103,89	120,52	137,66	163,26
Desv. T. Cheg. Real	DEVCHi	7,07	4,29	4,43	1,65	2,50	1,75	2,19	2,32
T. Atend. Real	TATi	37,98	50,98	72,44	84,44	105,52	120,59	142,08	163,26
Desv. T. Atend. Real	DEVATi	4,25	4,29	1,60	1,65	1,43	1,67	0,64	2,32

Pela Tabela 3, linhas 6 e 7, é possível ver a diferença entre os valores do tempo de chegada determinístico e a média do tempo de chegada via simulação. Na linha 9 (penúltima da tabela), pode-se observar a diferença entre o tempo médio de início de atendimento e o tempo médio de chegada. Por exemplo, para o cliente 1, enquanto a média de chegada é 35,36, o tempo médio de início de atendimento é 37,98, sendo significativamente deslocado para a direita em função da abertura da janela em 35 que faz com que todos os valores de chegada anteriores a 35 passem a ser 35.

A partir da simulação estocástica da rota exemplo (uma rodada de 10000 réplicas), foram coletados os dados do tempo de chegada e tempo de início de atendimento em cada cliente. A Figura 7 mostra os histogramas do tempo de chegada do veículo em cada um dos 8 clientes da rota.

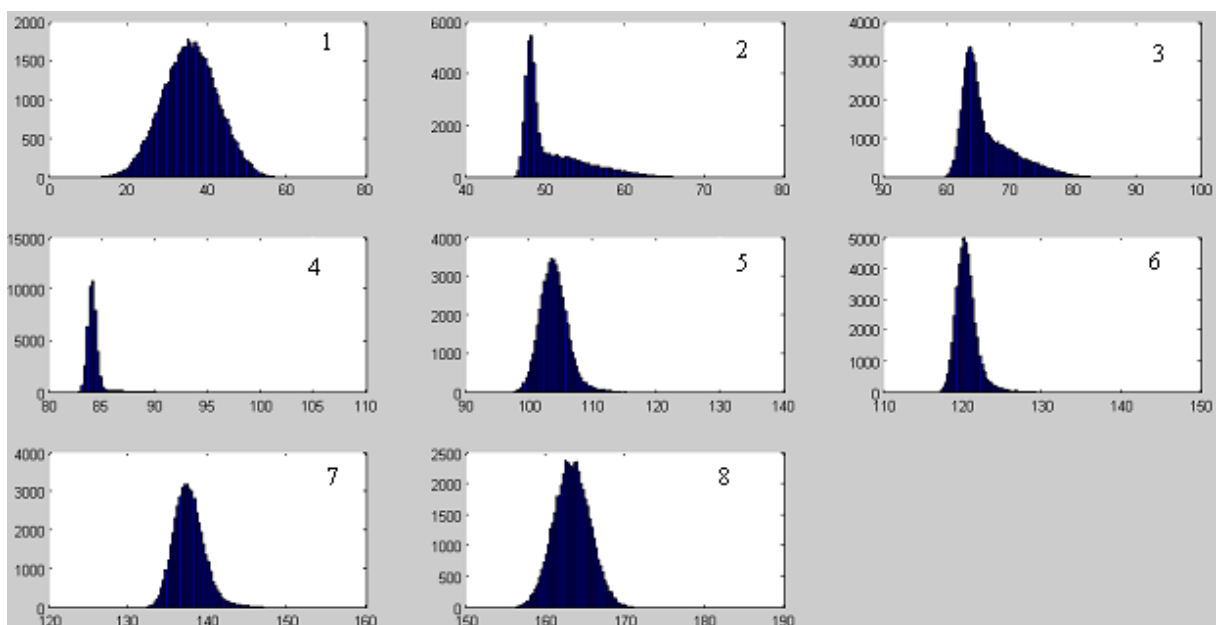


Figura 7: Histogramas do tempo de chegada nos clientes

Observa-se claramente que a distribuição dos clientes 2 e 3 não são normais. Um teste de normalidade foi aplicado aos tempos de chegada de cada um dos clientes, sendo mostrado na Figura 8.

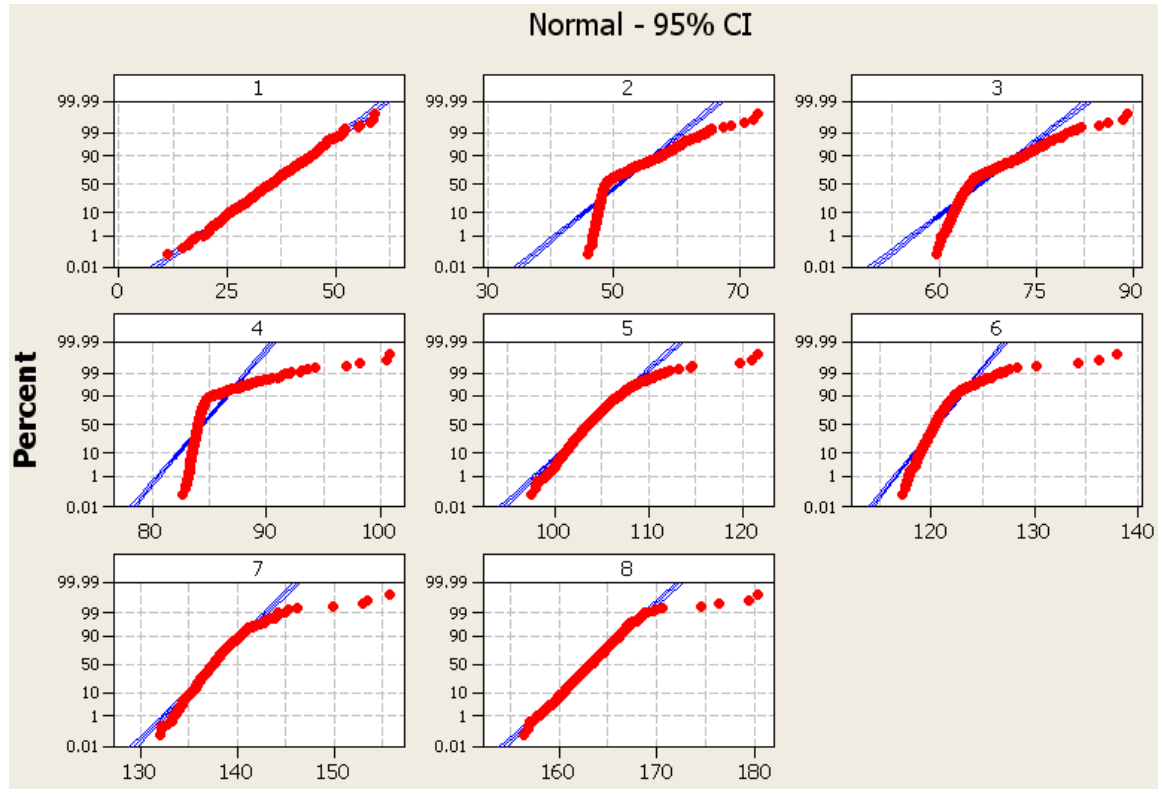


Figura 8: Teste de normalidade usando Anderson-Darling

Para um nível de confiança de 95%, o cliente 1 apresentou P -value de 0.579. Para todos os demais clientes, o P -value foi inferior a 0.005. Uma vez que o tempo de viagem dos arcos possui uma distribuição normal, naturalmente o tempo de chegada no primeiro cliente sempre seguirá uma distribuição normal, pois se assume que o tempo de partida do depósito é determinístico. Já para os demais clientes, neste exemplo, ficou claro que o tempo de chegada de todos eles não tem distribuição normal.

A Figura 9 mostra o histograma do tempo de início de atendimento para cada um dos clientes do exemplo.

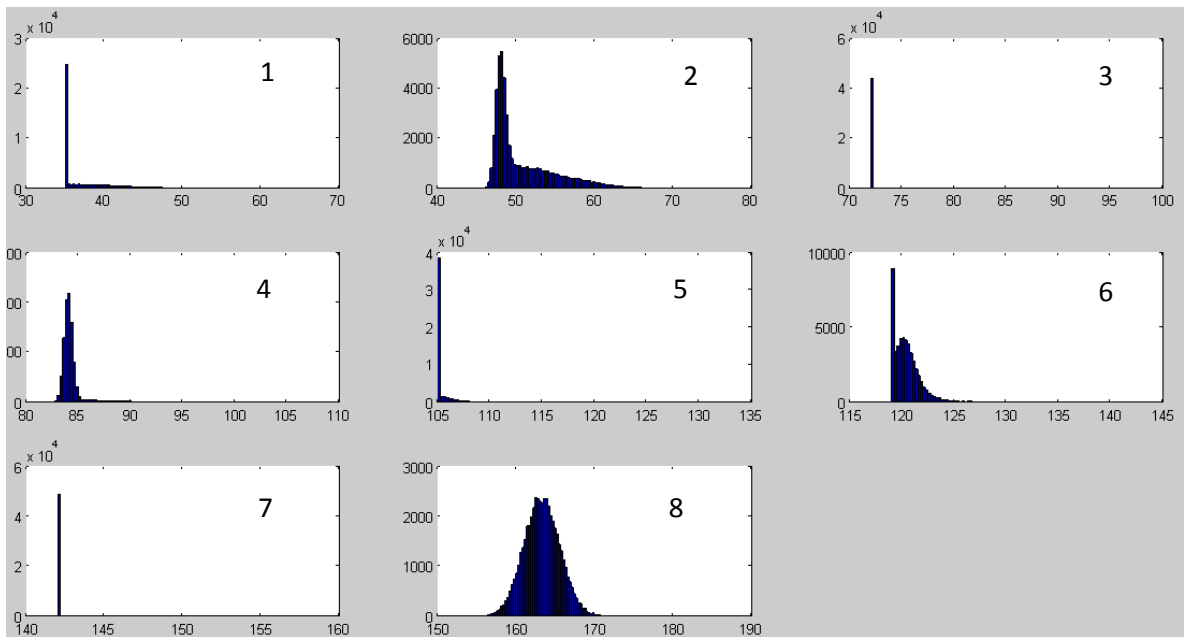


Figura 9: Histogramas do tempo de início de atendimento nos clientes

Na Figura 9 é possível perceber o efeito do tempo de espera na distribuição do tempo de início de atendimento. A média do tempo de chegada no cliente 1 é 35,36 minutos e o desvio-padrão de 7,07 minutos. A abertura da janela de tempo ocorre no minuto 35. Sendo a distribuição de probabilidade neste cliente normal, a probabilidade de que o veículo chegue antes do minuto 35 é de 47,8%. Ou seja, em 47,8% das oportunidades, o veículo chegará antes da abertura da janela, existindo tempo de espera, de modo que, para estes valores, o tempo de início de atendimento passa a ser a abertura da janela. Isto é representado no histograma do cliente 1 pela barra vertical grande que representa os 47,8% dos tempos de chegada cujo tempo de início de atendimento é 35. Para os demais 52,2% dos valores, o tempo de início de atendimento é o mesmo do tempo de chegada, resultando nas demais barras vistas no histograma do cliente 1.

Para os demais clientes o raciocínio é análogo, exceto o fato de que a distribuição de probabilidade do tempo de chegada não seja normal. Os clientes 3 e 7 possuem probabilidade de espera de 87,5% e de 96,7% respectivamente. Ou seja, quase sempre há espera, isto é visto no histograma pela barra vertical, onde nem é possível visualizar as barras correspondentes aos valores em que não há espera.

As figuras 10 e 11 mostram os histogramas do tempo de chegada para duas rotas geradas a partir de instâncias de Solomon (1987) também com algumas mudanças nos

valores das janelas de tempo para ilustrar qualitativamente o comportamento das distribuições de probabilidade ao longo de uma dada rota.

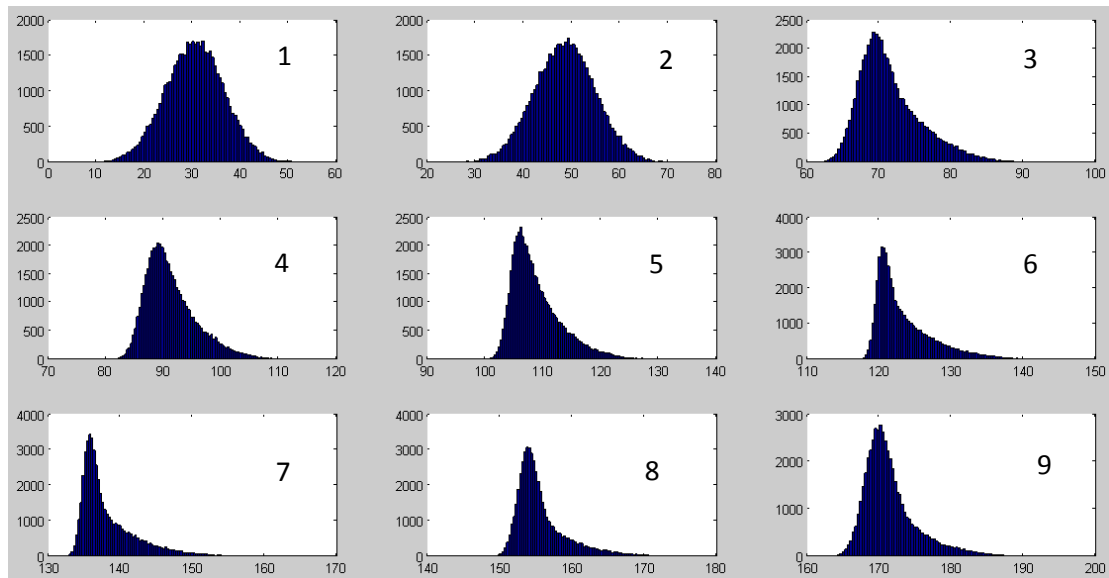


Figura 10: Histograma do tempo chegada do problema R104

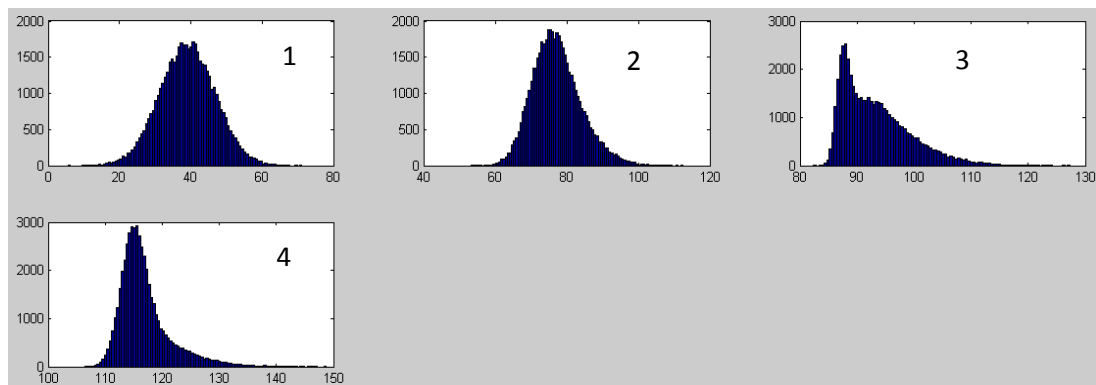


Figura 11: Histograma do tempo chegada do problema RC101

Observa-se claramente que em diversas situações a distribuição não se assemelha a uma normal. Além disto, a forma da distribuição muda ao longo dos clientes de uma rota. Qualquer rota, em que pode haver tempo de espera do veículo, está sujeita ao fenômeno investigado. O próximo tópico desenvolve um método estatístico para resolver este problema.

6.3 – MODELO ESTATÍSTICO

6.3.1 – Definição do problema estatístico

Este tópico introduz o método estatístico proposto para calcular a função cumulativa de probabilidade para o tempo de chegada.

O modelo estatístico é um problema a parte, aplicável ao PRV Estocástico. Sendo assim, por uma questão de clareza, iremos utilizar uma notação específica para o problema estatístico. Seja a variável aleatória X_i o tempo de chegada (AT) ao cliente i , e t o valor de truncamento dado pela abertura da janela de tempo e_i , X_i se torna a variável truncada X_i^t com função f_x truncada à esquerda no ponto t , onde $X_i^t = t$ se $x \leq t$ e $X_i^t = x$ se $x > t$. A variável Y_i é a soma do tempo de serviço (ST) com o tempo de viagem (TT), considerando-se ambos normalmente distribuídos: $ST_i + TT_{i,i+1} = N(\mu_{ST_i}, \sigma_{ST_i}^2) + N(\mu_{TT_{i,i+1}}, \sigma_{TT_{i,i+1}}^2) = N(\mu_{ST_i} + \mu_{TT_{i,i+1}}, \sigma_{ST_i}^2 + \sigma_{TT_{i,i+1}}^2) = Y_i$. Desta forma, o problema se torna resolver $X_{i+1} = X_i^t + Y_i$ recursivamente de $i = 1$ até $i = n$, onde n é o número de clientes de uma dada rota.

Na Figura 12, X_1 é uma variável aleatória que após truncada a esquerda, torna-se X_1^t (distribuição normal truncada à esquerda), e Y_1 é uma distribuição normal não truncada. A soma $X_2 = X_1^t + Y_1$ pode resultar numa distribuição não normal. Deseja-se calcular $X_3 = X_2^t + Y_2$, onde Y_2 é uma variável normal conhecida e X_2^t é uma distribuição não normal truncada à esquerda. Como calcular a função cumulativa de probabilidade de X_3 ?

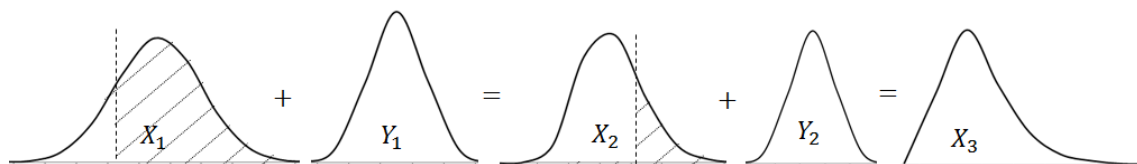


Figura 12: Soma de variáveis aleatórias

Agora, considerando Y_i para $i = 1, \dots, n$ uma variável normal de parâmetros conhecidos, e também considerando $X_i = X_{i-1}^t + Y_{i-1}$ onde X_{i-1}^t é truncado a esquerda, vamos assumir uma sequência de soma de variáveis, como $X_4 = X_3^t + Y_3, \dots, X_n = X_{n-1}^t + Y_{n-1}$. Como calcular a função cumulativa de probabilidade de X_i para $1, \dots, n$? A Figura 13 ilustra o caso e enfatiza o fato que $X_i^t = t$ se $x \leq t$.

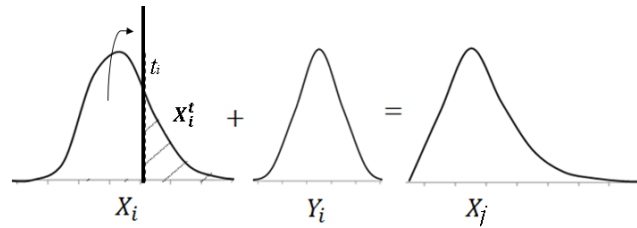


Figura 13: Generalização da soma de variáveis aleatórias

6.3.2 – Solução do problema estatístico

A função cumulativa de probabilidade $X_j = X_i^t + Y_i$ (no contexto da Figura 13) é obtida através da equação a seguir:

$$P\{X_j \leq c\} = P\{X_i^t + Y_i \leq c\} = F_{X_i^t + Y_i}(c) = \int_{-\infty}^{\infty} F_{X_i^t}(c - y_i) f_y(y_i) dy \quad (54)$$

A equação (54) é conhecida como a convolução das distribuições marginais e resulta na distribuição da soma das variáveis aleatórias. Esta equação pode ser aproximada pela equação (55), usando-se uma discretização com I intervalos, onde y_f e y_0 são os limites inferiores e superiores da integral, respectivamente.

$$\int_{-\infty}^{\infty} F_{X_i^t}(c - y_i) f_y(y_i) dy \cong \frac{y_f - y_0}{I} * \sum_{k=1}^I \left(\frac{dy_{k-1} + dy_k}{2} \right) \quad (55)$$

Onde $dy_k = F_{X_i^t}(c_q - y_k) f_y(y_k)$, $\forall q \in \{1, \dots, nint\}$.

A função de distribuição cumulativa $F_{X_i^t}(c_q - y_k)$ precisa ser calculada na iteração anterior i para cada valor de $c_q - y_k$ onde $q = \{1, \dots, nint\}$. Esta função é armazenada de forma discretizada com $nint$ pontos, numa matriz $mprob_i = P\{X_i^t \leq c_q - y_k\}$ com duas linhas e $nint$ colunas, como a seguir:

$$mprob_i = \begin{bmatrix} c_1 - y_k & \dots & c_q - y_k & \dots & c_{nint} - y_k \\ F_{X_i^t}(c_1 - y_k) & \dots & F_{X_i^t}(c_q - y_k) & \dots & F_{X_i^t}(c_{nint} - y_k) \end{bmatrix}$$

Onde $F_{X_i^t}(c_1 - y_k) \cong 0$ e $F_{X_i^t}(c_{nint} - y_k) \cong 1$.

Para cada valor $c_q - y_k$ da matriz $mprob_i$ faz-se necessário resolver a convolução da equação 54 para obter $F_{X_i^t}(c_q - y_k)$. Sendo $mprob_i = P\{X_i^t \leq c_q - y_k\}$ uma matriz com os valores discretos e ordenados da função cumulativa, qualquer valor da variável contínua $F_{X_i^t}$ é calculada através da interpolação linear dos pontos adjacentes. Para calcular-se a próxima iteração ($j + 1$), faz-se necessário primeiro calcular-se $mprob_j$, resolvendo-se a equação (55) para cada $c_q - y_k$ in $q = \{1, \dots, nint\}$. Note que para a primeira iteração, a variável aleatória X_1 é normalmente distribuída, então $mprob_1$ pode ser calculado diretamente usando-se a equação da função de distribuição cumulativa normal.

Observe que a equação (54) não pode ser resolvida de forma direta, não permitindo a obtenção de uma fórmula explícita. Neste contexto, a função distribuição de probabilidade para a variável aleatória X não possui uma forma conhecida; mesmo quando as equações para F_X e f_y são conhecidas, o produto $F_X(c - y) f_y$ resulta em equações complexas que não permitem a utilização de métodos analíticos para resolver a integral. Por este motivo, um algoritmo eficiente para resolver a equação (55) numericamente é desenvolvido e apresentado na próxima seção.

6.3.4 – A convolução

Deseja-se calcular a probabilidade $P\{X^{tr} + Y \leq c\}$ através da equação (55). No contexto do PRV, isto é utilizado para calcular o nível de serviço num dado cliente. O objetivo aqui é criar um método para resolver numericamente a convolução de forma rápida, já que tal método será aplicado a um problema de otimização do tipo *NP-hard*.

A função “Conv” (algoritmo 1) mostra como resolver a convolução da equação (55). Os parâmetros de entrada são listados na Tabela 4. O algoritmo apresenta algumas declarações de variáveis ou linhas de código que podem ser consideradas desnecessárias, mas que foram mantidas neste texto objetivando maior clareza.

Tabela 4: Notação dos parâmetros de entrada do algoritmo

<i>MeanX</i>	Média de X_i
<i>DevX</i>	Desvio Padrão de X_i
t_i	Ponto de truncamento de X_i
<i>MeanY</i>	Média de Y_i
<i>DevY</i>	Desvio Padrão de Y_i
<i>c</i>	Valor para o qual deseja-se calcular a probabilidade $P\{X^{tr} + Y \leq c\}$
<i>mprob_i</i>	Matriz com a função cumulativa F_X , com os valores discretos de X_i na coluna 1 e a coluna 2 com os valores de $P\{X_i \leq c\}$ no espaço de probabilidade $[0,1]$, com <i>nint</i> pontos (número de intervalos), onde $q = \{1, \dots, nint\}$.
<i>vidx</i>	Vetor auxiliar onde $vidx(1) = mprob_i(1,1)$; $vidx(2) =$ incremento δ dos valores de X em <i>mprob_i</i> ; $vidx(3) = mprob_i(1,n)$;
<i>I</i>	Número de intervalos para calcular a equação (55).

A função “Conv” recebe os parâmetros da Tabela 4 e retorna o valor da probabilidade $P\{X^{tr} + Y \leq c\}$.

Algoritmo 1: Algoritmo para resolver a convolução

```

1  function [prob]=Conv(t,MeanY,DevY,c,mprob,vidx,I)
2  y0= MeanY-3.5*DevY; yf= MeanY+3.5*DevY;
3  minX=mprob(1,1);maxX=mprob(vidx(4),1);
4  x0=max(minX,t) +max(y0,0);
5  xf=max(maxX,t) +yf;
6  if c > xf
7      prob=1; return
8  elseif c < x0
9      prob=0; return
10 end
11 if yf > c-t
12     yf= c-t;
13 end
14 inc=(yf-y0)/I;
15 yf= yf+ inc/1000;
16 den=(sqrt(2*pi)* DevY);
17 y=y0; tot=0; dyconv2=0;
18 while y < yf
19     cy=c-y;
20     if cy <=vidx(1)
21         Fx=0;
22     elseif cy >= vidx(3)
23         Fx=1;
24     else
25         q=ceil(((cy)-vidx(1))/vidx(2));
26         den2=(mprob(q+1,1) - mprob(q,1));
27         Fx =((cy)* (mprob(q+1,2) - mprob(q,2))
28             + mprob(q,2) * mprob(q+1,1)
29             - mprob(q+1,2) * mprob(q,1))/den2;
30     end
31     dyconv=Fx * (exp(-0.5 * ((y - MeanY)/DevY)^2)/den);
32     tot= tot+ (dyconv+dyconv2)/2; dyconv2=dyconv;
33     y=y+inc;
34 end
35 prob=tot*inc;
36 end

```

A linha 3 obtém os valores de x para o qual a probabilidade é $\cong 0$ e $\cong 1$, respectivamente, onde $\min X = F_X^{-1}(\cong 0)$ e $\max X = F_X^{-1}(\cong 1)$. As linhas 4 e 5 estimam os valores $x_0 = F_{X+Y}^{-1}(\cong 0) = \text{Max}\{\min X, t\} + \text{Max}\{\text{Mean}Y - 3.5\text{Dev}Y, 0\}$ e $x_f = F_{X+Y}^{-1}(\cong 1) = \text{Max}\{\max X, t\} + (\text{Mean}Y + 3.5\text{Dev}Y)$. Estes valores são importantes para criar uma condição de saída nas linhas 6 a 10 que neste caso economizam esforço computacional, evitando os loops das linhas 18 a 34.

As linhas 2 e 12 definem os limites inferior e superior da integral. Estes limites são dados pelas equações (56) e (57) a seguir.

$$y_0 = (\mu - 3.5 \sigma) \text{ para } Y \sim N(\mu, \sigma^2) \quad (56)$$

$$y_f = \text{Min}\{(\mu + 3.5 \sigma), (c - t)\} \text{ para } Y \sim N(\mu, \sigma^2) \quad (57)$$

Para $Y \sim N(\mu, \sigma^2)$, $P\{Y \leq (\mu - 3.5\sigma)\} \cong 0$ e $P\{Y \geq (\mu + 3.5\sigma)\} \cong 0$. Na equação (57), se $(\mu + 3.5 \sigma) > (c - t)$ então $y_f = (c - t)$. Isto ocorre pois na equação (55), para $f_y(y)$, quando y alcança seu valor máximo $(\mu + 3.5 \sigma)$, para o termo $F_X(c - y)$, o valor $(c - y)$ alcança seu valor mínimo que deve ser maior ou igual a t .

A linha 14 calcula o incremento para o valor de y no intervalo $y_0 \leq y \leq y_f$. Linha 15 apenas adiciona um pequeno valor para o limite superior UB para assegurar que a integração ocorra até $y = y_f$ prevenindo erros numéricos. Linha 16 calcula o denominador da função distribuição de probabilidade normal $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-0.5\frac{(x-\mu)^2}{\sigma^2}}$.

Linha 17 inicializa os valores para calcular a integral no loop das linhas 18 a 34. Linhas 19 a 30 calculam o valor de $F_X(c - y)$. A matriz $mprob$ possui os valores de F_X . Linhas 21 e 23 calculam $F_X(c - y)$ quando $cy = (c - y)$ está for a dos limites de $mprob$ e linha 27 calcula $F_X(c - y)$ para os valores de cy dentro do intervalo. A primeira linha da matriz $mprob$ é um vetor com os valores de $X = (x_1, x_2, \dots, x_{nint})$. Para um dado valor de $x = cy = (c - y)$ é necessário calcular os valores x_q e x_{q+1} onde $x_q < x < x_{q+1}$. O índice q é calculado na linha 25, e a linha 27 calcula $F_X(c - y)$ através de uma interpolação linear de $F_X(x_q)$ e $F_X(x_{q+1})$. Quanto maior $nint$, menor o espaço entre dois pontos consecutivos, então menor é o erro da interpolação linear. Isto será discutido adiante noutro tópico. A equação (58) abaixo, implementada na linha 25 é particularmente importante no algoritmo pois permite encontrar o índice q com custo computacional $O(1)$.

$$q = \left\lceil \frac{x - F_X(x_1)}{\delta} \right\rceil = \left\lceil \frac{cy - vidx(1)}{vidx(2)} \right\rceil \quad (58)$$

$$F_X(x_q) = \frac{x((F_X(x_{q+1}) - F_X(x_q)) + F_X(x_q)x_{q+1} - F_X(x_{q+1})x_q)}{x_{q+1} - x_q} \quad (59)$$

Na equação (58), δ é o valor utilizado para incrementar x_1 a x_{nint} onde $q = 1, \dots, nint$. Na equação (59), $F_X(x_q) = mprob(2, q)$, $x_q = mprob(1, q)$ e $x = cy$ no algoritmo.

O valor de $F_X(c - y) f_y(y)$ é calculado na linha 31. As linhas 32 e 33 armazenam os valores da função cumulativa e incrementa a variável, respectivamente. Finalmente, a linha 35 calcula o valor da integral.

Observe que o presente método funciona para qualquer situação em que a variável Y_i possui uma função distribuição de probabilidade conhecida. Tal função deve ser usada como $f_y(y_i)$ na equação (54), e implementada na linha 31 do algoritmo 1. Estudos usando outros tipos de distribuição para o tempo de viagem são oportunidades para trabalhos futuros.

Até este ponto, mostrou-se como calcular $P\{X^{tr} + Y \leq c\}$. O próximo passo mostra como calcular a função de distribuição cumulativa de X^{tr} .

6.3.5 – A função de distribuição cumulativa

A estratégia é usar o algoritmo da convolução descrito no tópico anterior para calcular $nint$ pontos discretos no espaço de probabilidade $[0,1]$ para a variável aleatória X .

A função “F” (algoritmo 2) descreve como obter a matriz de função cumulativa ($mprob$) com os valores de X_i na linha 1 e com os valores de $P\{X_i \leq c\}$ na linha 2, no espaço de probabilidade $[0,1]$ com $nint$ pontos, onde $q = \{1, \dots, nint\}$. As entradas são as mesmas já descritas na Tabela 4.

Algoritmo 2: Função “F” para o cálculo da função de distribuição cumulativa

```

1  function[mprobN,vidxN]=F(t,MeanY,DevY,mprob,vidx)
2  y0= MeanY-3.0*DevY;
3  yf= MeanY+3.0*DevY;
4  minX=mprob(1,1);
5  maxX=mprob(vidx(4),1);
6  x0=max(minX,t) +max(y0,0);
7  xf=max(maxX,t) +yf;
8  idx=0;dy=1;
9  inc=(xf-x0)/(nint);
10 vidxN(1)=x0; vidxN(2)=inc; ping1=0; gapF=0.03;
11 while idx==0 || (mprobN(idx,2)<0.999 && abs(dy)> 0.0001 )
12     || mprobN(idx,2)< 0.98
13     idx=idx+1;
14     mprobN(idx,1)=x0+(idx-1)*inc;
15     if idx==1 || mprobN(idx-1,2) < 0.05
16         I=I1;
17     else
18         I=I2;
19         if mprobN(idx-1,2) > 0.7
20             if (mprobN(idx-1,2)-mprobN(idx-2,2))<gapF && ping1==0
21                 idx=idx+1;
22                 mprobN(idx,1)=x0+(idx-1)*inc;
23                 ping1=1;
24             end
25         end
26     end
27     c=mprobN(idx,1);
28     [mprobN(idx,2)]=Conv(t,MeanY,DevY,c,mprob,vidx,I);
29     if idx==1
30         dy=mprobN(idx,2);
31     else
32         if ping1==1
33             mprobN(idx-1,2)=0.5*mprobN(idx-2,2)+0.5*mprobN(idx,2);
34             ping1=0;
35             dy=(mprobN(idx-1,2) - mprobN(idx-2,2));
36         end
37         dy=(mprobN(idx,2) - mprobN(idx-1,2));
38     end
39 end
40 vidxN(3)=mprobN(idx,1); vidxN(4)=idx;
41 end

```

O algoritmo primeiramente calcula o valor inicial (x_1) e o valor final x_{nint} ao longo do qual a função cumulativa F_X será calculada. O segundo passo é o loop para calcular o valor $F_X(x_q)$ para cada um dos $nint$ valores x_q entre x_1 e x_{nint} . O algoritmo calcula a média e o desvio padrão para a soma de variáveis $X + Y$.

Linhas 2 a 7 são as mesmas já descritas na função “Conv” (algoritmo 1), calculando os limites inferior e superior no espaço de probabilidade $[0,1]$. As linhas 4 e 5 obtêm os valores de x para o qual a probabilidade é $\cong 0$ and $\cong 1$, respectivamente, onde

$\min X = F_X^{-1}(\cong 0)$ e $\max X = F_X^{-1}(\cong 1)$. De maneira alternativa, a partir de resultados obtidos em experimentos preliminares, o uso de $\min X = F_X^{-1}(p1)$ e $\max X = F_X^{-1}(p2)$ onde $p1 \in [0.01, \dots 0.05]$ e $p2 \in [0.95, \dots 0.99]$ ajudaram a obter valores de *mprob* mais *centralizados* no espaço de probabilidade $[0,1]$. Então, ao invés de capturar o primeiro e o último valor de *mprob*, pode-se obter *minX* e *maxX* a partir dos valores de probabilidade mais próximos de $p1$ e $p2$, respectivamente. Estes limites são usados na linha 9 para obter o incremento a ser utilizado para calcular a probabilidade $P\{X + Y \leq t\}$. Os valores $X + Y$ para os quais a probabilidade $P\{X + Y \leq t\}$ é calculada e os respectivos valores desta probabilidade são armazenados na matriz *mprobN*.

Linha 10 inicializa o loop da linha 11 a 44. Linhas 11 e 12 possuem três condições de permanência no loop: a primeira condição apenas assegura a primeira iteração do loop, a segunda assegura que a iteração continuará enquanto a última probabilidade calculada seja menor que 99.9% e a diferença entre dois cálculos consecutivos de probabilidade (*dy*) seja maior que 0.0001. A terceira condição garante que a iteração continuará pelo menos até $F_x = 0.98$. As condições 2 e 3 são justificadas pelo comportamento assintótico da função de distribuição cumulativa.

Linhas 16 e 18 obtém o número de intervalos que serão utilizados para calcular $P\{X^t + Y \leq t\}$ usando a função “Conv” na linha 28 (algoritmo 1). Enquanto $P\{X^t + Y \leq t\} < 5\%$, utiliza-se $I = 5$, pois experimentalmente concluiu-se que poucos passos são suficiente para resolver a convolução nesta faixa de valores, para os demais casos, $I = I2$ (valores a serem estudados em tópico posterior).

Entre as linhas 19 e 25 implementou-se um mecanismo para acelerar o algoritmo. Experimentos preliminares mostraram que a partir de $F_x = 0.70$ aproximadamente, a taxa de variação da curva de distribuição cumulativa começa a decrescer rapidamente por ser assintótica. Aproximadamente após este ponto, se a diferença entre dois valores consecutivos de probabilidade é menor que 3%, o algoritmo salta o ponto corrente e vai para o próximo ponto. O valor x (aqui um dado valor de $X + Y$) que foi saltado terá sua probabilidade calculada na linha 33 como a interpolação linear de seus pontos vizinhos. Como cada ponto discreto implica em resolver uma convolução, cada “salto”, economiza o cálculo de uma convolução (loop principal do algoritmo 1).

Isto é ilustrado pelas figuras 14 e 15, onde a linha sólida é a função de distribuição cumulativa real (obtida via simulação) e os círculos são os pontos calculados pelo método.

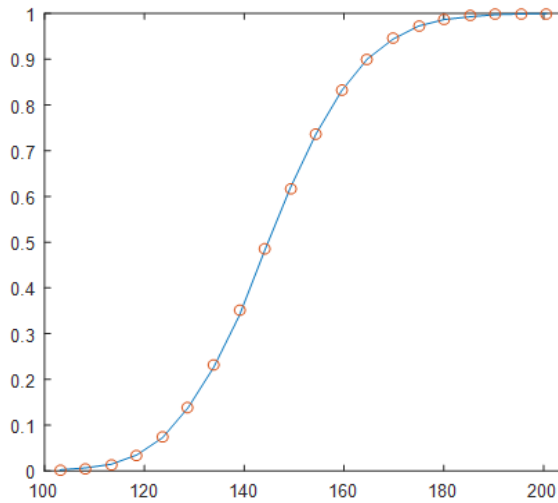


Figura 14: Função de distribuição cumulativa

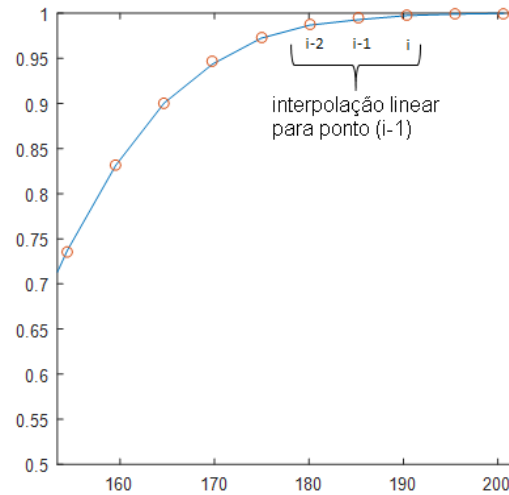


Figura 15: Interpolação linear

A Figura 14 exemplifica uma função cumulativa $F_x(c) = P\{X \leq c\}$ aproximada por 20 pontos discretos (círculos), ou seja $nint = 20$. Observa-se que a derivada diminui severamente no lado direito da curva. A Figura 15 trata-se de um “zoom in” no lado direito para ilustrar um caso em que o ponto $i - 1$ é interpolado pelos pontos $i - 2$ e i .

O método possui dois parâmetros principais: $nint$ (número de intervalos ou tamanho do vetor que armazena os valores da função cumulativa) e I (número de intervalos a serem usados para resolver a convolução). Valores para estes parâmetros serão discutidos adiante.

6.3.6 –A soma de normais truncadas

O modelo explicado nos tópicos anteriores resolve a soma de uma variável aleatória truncada não normal com uma variável aleatória normal. Isto pode ser usado para calcular o nível de serviço dos clientes assumindo-se que a soma do tempo de serviço (ST) com o tempo de viagem (TT), considerando-se ambos normalmente distribuídos, resulta numa outra variável também normal, aqui identificada como $Y_i : ST_i + TT_{i,i+1} = N(\mu_{ST_i}, \sigma_{ST_i}^2) + N(\mu_{TT_{i,i+1}}, \sigma_{TT_{i,i+1}}^2) = N(\mu_{ST_i} + \mu_{TT_{i,i+1}}, \sigma_{ST_i}^2 + \sigma_{TT_{i,i+1}}^2) = Y_i$. Desta forma, o tempo de viagem e de serviço são representados por uma única variável e o problema se torna resolver $X_{i+1} = X_i^{tr} + Y_i$ recursivamente de $i = 1$ até $i = n$, onde n

é o número de clientes de uma dada rota. O detalhe é que, para problemas práticos, o tempo de viagem e o tempo de serviço são uma distribuição normal não negativa, ou seja, truncada no ponto zero, de modo que a soma de duas variáveis normais truncadas não necessariamente é normal. Desta forma, assumir que a soma do tempo de viagem com o tempo de serviço é uma variável gaussiana trata-se de uma aproximação.

A Figura 16 ajuda a esclarecer esta consideração sobre o truncamento. Na parte superior da figura, o ponto zero encontra-se numa região com probabilidade zero ou muito próximo de zero, de forma que o efeito do truncamento não é relevante. A segunda parte da figura mostra um caso em que o efeito do truncamento convém ser considerado.

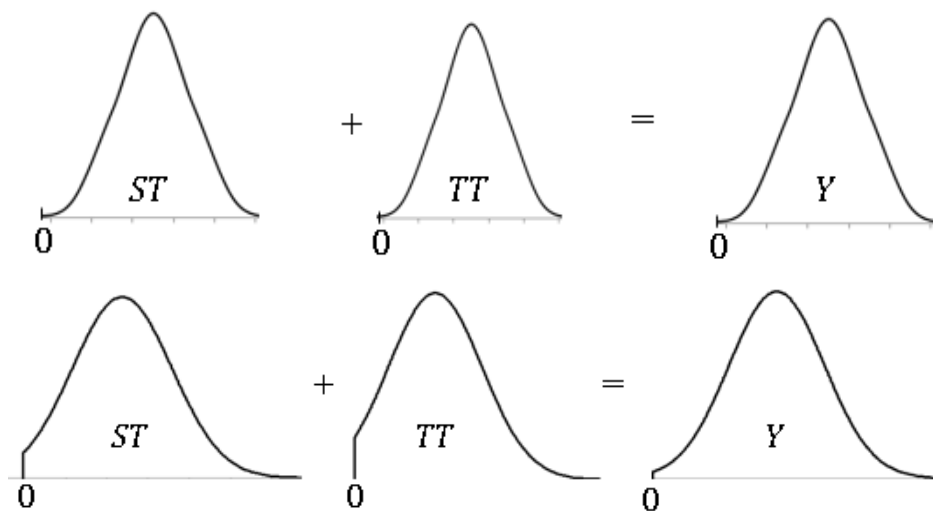


Figura 16: Casos para normais truncadas no ponto zero

Para a soma de duas variáveis normais truncadas (à esquerda, à direita, ou ambos) é possível encontrar uma solução de forma fechada usando-se o Método de Transformação com Jacobiano. Seja $X_1 \sim N(\mu_1, \sigma_1^2)$ truncada em ambos os lados (caso mais completo) nos pontos a_1 pela esquerda e b_1 pela direita e $X_2 \sim N(\mu_2, \sigma_2^2)$ truncada nos pontos a_2 e b_2 . Sendo X_1 e X_2 independentes, a função distribuição de probabilidade conjunta $f(x_1, x_2)$ é o produto das funções de distribuições individuais. A função de distribuição $f(x_1)$ é dada na equação (60) e $f(x_2)$ na equação (61).

$$f(x_1) = \frac{1}{\sigma_1 \sqrt{\pi}} \frac{\exp\left(-\frac{(x_1 - \mu_1)^2}{2\sigma_1^2}\right)}{\operatorname{erf}\left(\frac{b_1 - \mu_1}{\sqrt{2}\sigma_1}\right) - \operatorname{erf}\left(\frac{a_1 - \mu_1}{\sqrt{2}\sigma_1}\right)} \quad (60)$$

$$f(x_2) = \frac{1}{\sigma_2} \sqrt{\frac{2}{\pi}} \frac{\exp\left(-\frac{(x_2 - \mu_2)^2}{2\sigma_2^2}\right)}{\operatorname{erf}\left(\frac{b_2 - \mu_2}{\sqrt{2}\sigma_2}\right) - \operatorname{erf}\left(\frac{a_2 - \mu_2}{\sqrt{2}\sigma_2}\right)} \quad (61)$$

Onde $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt$.

Seja $Z = X_1 + X_2$ e $W = X_2$, a função conjunta $g(z, w)$ é dada pela equação a seguir:

$$g(z, w) = \frac{2 \exp\left(-\frac{(w - z + \mu_1)^2}{2\sigma_1^2} - \frac{(w - \mu_2)^2}{2\sigma_2^2}\right)}{\pi\sigma_1\sigma_2 \left(\operatorname{erf}\left(\frac{a_1 - \mu_1}{\sqrt{2}\sigma_1}\right) - \operatorname{erf}\left(\frac{b_1 - \mu_1}{\sqrt{2}\sigma_1}\right)\right) \left(\operatorname{erf}\left(\frac{a_2 - \mu_2}{\sqrt{2}\sigma_2}\right) - \operatorname{erf}\left(\frac{b_2 - \mu_2}{\sqrt{2}\sigma_2}\right)\right)} \quad (62)$$

Dado $g(z, w)$, a função marginal de Z é dada em quatro intervalos, pelas equações (63) a (66).

$$f_1 = T_6 \sqrt{\frac{2}{\pi}} (T_1 - T_2) \frac{1}{T_5} \quad z < a_2 + b_1 \text{ e } z \leq a_1 + b_2 \quad (63)$$

$$f_2 = T_6 \sqrt{\frac{2}{\pi}} (T_1 - T_3) \frac{1}{T_5} \quad z \geq a_2 + b_1 \text{ e } z \leq a_1 + b_2 \quad (64)$$

$$f_3 = T_6 \sqrt{\frac{2}{\pi}} (-T_2 + T_4) \frac{1}{T_5} \quad z < a_2 + b_1 \text{ e } z > a_1 + b_2 \quad (65)$$

$$f_4 = T_6 \sqrt{\frac{2}{\pi}} (-T_3 + T_4) \frac{1}{T_5} \quad z \geq a_2 + b_1 \text{ e } z \geq a_1 + b_2 \quad (66)$$

Onde:

$$T_1 = \operatorname{erf}\left(\frac{(z - a_1 - \mu_2)\sigma_1^2 + (-a_1 + \mu_1)\sigma_2^2}{\sqrt{2}\sigma_1\sigma_2\sqrt{\sigma_1^2 + \sigma_2^2}}\right) \quad (67)$$

$$T_2 = \operatorname{erf}\left(\frac{(a_2 - \mu_2)\sigma_1^2 + (-z + a_2 + \mu_1)\sigma_2^2}{\sqrt{2}\sigma_1\sigma_2\sqrt{\sigma_1^2 + \sigma_2^2}}\right) \quad (68)$$

$$T_3 = \operatorname{erf}\left(\frac{(z - b_1 + \mu_2)\sigma_1^2 + (-b_1 + \mu_1)\sigma_2^2}{\sqrt{2}\sigma_1\sigma_2\sqrt{\sigma_1^2 + \sigma_2^2}}\right) \quad (69)$$

$$T_4 = \operatorname{erf}\left(\frac{(b_2 - \mu_2)\sigma_1^2 + (-z + b_2 + \mu_1)\sigma_2^2}{\sqrt{2}\sigma_1\sigma_2\sqrt{\sigma_1^2 + \sigma_2^2}}\right) \quad (70)$$

$$T_5 = \left(\operatorname{erf}\left(\frac{a_1 - \mu_1}{\sqrt{2}\sigma_1}\right) - \operatorname{erf}\left(\frac{b_1 - \mu_1}{\sqrt{2}\sigma_1}\right)\right)\left(\operatorname{erf}\left(\frac{a_2 - \mu_2}{\sqrt{2}\sigma_2}\right) - \operatorname{erf}\left(\frac{b_2 - \mu_2}{\sqrt{2}\sigma_2}\right)\right)\sqrt{\sigma_1^2 + \sigma_2^2} \quad (71)$$

$$T_6 = \exp\left(-\frac{(-z + \mu_1 + \mu_2)^2}{2\sqrt{\sigma_1^2 + \sigma_2^2}}\right) \quad (72)$$

Para o problema estudado, o caso de interesse tem a distribuição truncada à esquerda apenas, utilizando-se, portanto, a equação (63) com $b_1 = b_2 \sim \infty$, sendo $\operatorname{erf}(\infty) = 1$. Para reduzir o custo computacional, a função de erro $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt$ pode ser aproximada pela equação (73) descrita em Abramowitz & Stegun (1964) com erro máximo de 0.0005.

$$\operatorname{erf}(x) \cong 1 - \frac{1}{(1 + c_1x + c_2x^2 + c_3x^3 + c_4x^4)^4} \quad (73)$$

Onde $c_1 = 0.278393$, $c_2 = 0.230389$, $c_3 = 0.000972$, $c_4 = 0.078108$

Para ilustrar uma aplicação da equação (63) para somar duas variáveis aleatórias normais truncadas, neste exemplo, $X_1 \sim N(100, 20^2)$ e $X_2 \sim N(80, 80^2)$, ambos truncados à esquerda no ponto $a = 0$ e utilizando-se a equação (73) para aproximar a função $\operatorname{erf}(x)$, realizou-se um experimento comparando o resultado da aproximação com o resultado obtido via simulação.

A simulação foi feita utilizando-se Matlab, conforme a seguir:

- 1) $v1 = \operatorname{truncnormrnd}([1000000,1], mi1, dev1, a1);$
- 2) $v2 = \operatorname{truncnormrnd}([1000000,1], mi2, dev2, a2);$
- 3) $v = v1 + v2;$
- 4) $[f, xi] = \operatorname{ksdensity}(v);$

A Figura 17 com a comparação da simulação (linha contínua) com os resultados calculados (círculos), evidencia a qualidade da aproximação.

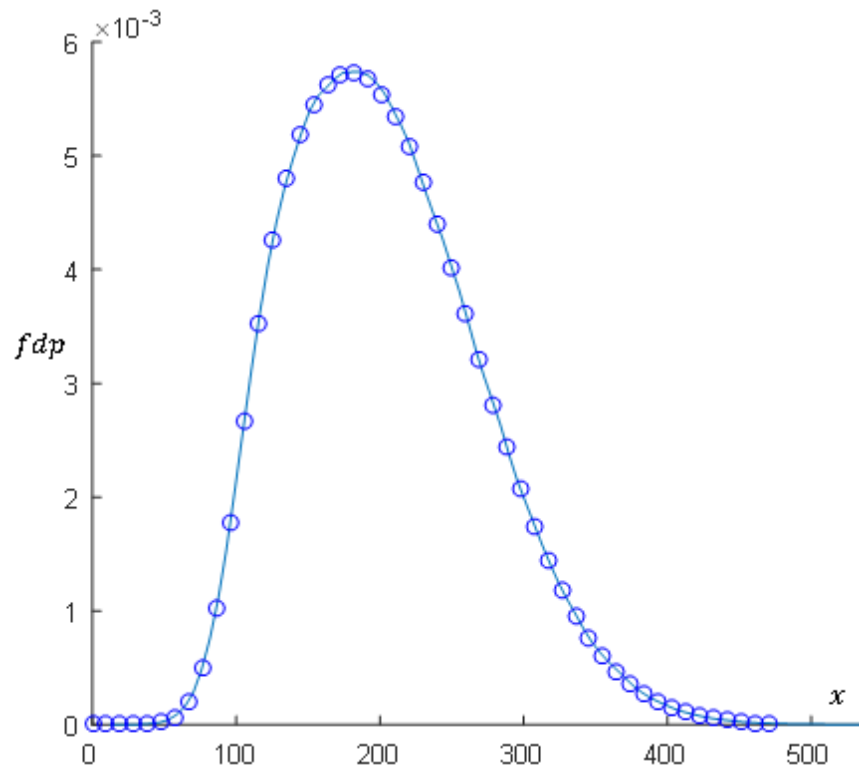


Figura 17: Função de distribuição cumulativa

A função “Conv” (algoritmo 1) usada para resolver a convolução da soma de variáveis aleatórias pode então ser adaptada para considerar o truncamento à esquerda no ponto zero para o tempo de viagem e tempo de serviço. Para isto faz-se suficiente substituir a equação da função de distribuição normal da linha 31, pela equação (63).

A equação (63) precisa ser computada em conjunto com a equação (73), ambas no loop interno da convolução (algoritmo 1), o que significa um maior esforço computacional. Para evitar tais cálculos, uma outra variante do método descrito para resolver o problema evitando o uso destas equações consiste em calcular a média e o desvio padrão da soma do tempo de serviço e de viagem, levando-se em consideração o truncamento. A média e a variância de uma distribuição normal truncada à esquerda no ponto a é dada pelas equações (74) e (75).

$$E[X|X > a] = \mu + \sigma \frac{\phi(\alpha)}{1 - \Phi(\alpha)} \quad (74)$$

$$\text{Var}[X|X > a] = \sigma^2 \left[\left(1 + \frac{\alpha\phi(\alpha)}{1 - \Phi(\alpha)} \right) - \left(\frac{\phi(\alpha)}{1 - \Phi(\alpha)} \right)^2 \right] \quad (75)$$

Assim, a média e a variância de $Y = ST + TT$ é calculada de forma precisa e então utilizadas nos parâmetros de entrada da função “F” (algoritmo 2). Isto significa que na linha 31 do algoritmo 1 (função “Conv”), os parâmetros de média e desvio estarão corretos (ou seja, considerando o truncamento no ponto zero para ST e TT), e apenas a forma da distribuição é aproximada por uma normal. O desempenho desta versão do método proposto é discutido em experimentos do capítulo 7.

6.4 – METAHEURÍSTICA PARA O PROBLEMA PROPOSTO

Este tópico apresenta o algoritmo memético desenvolvido para resolver o PRVJT Multiobjetivo com Tempo de Viagem e Serviço Estocásticos. Qualquer metaheurística pode ser adaptada para usar os algoritmos 1 e 2 para calcular o nível de serviço dos clientes.

O algoritmo 3 mostra o loop principal do algoritmo memético.

Algoritmo 3: loop principal do algoritmo

```

1  Inicializar os parâmetros do problema
2  enquanto MelhoraVolume > minVolume
3      se gen=1 então
4          Gerar nIniSol soluções iniciais e retornar Offspring
5      senão
6          Seleção por Stochastic Universal Sampling em Pop
7          para i=1:nFilhos
8              Aplicar Crossover e adicionar à Offspring
9          fim
10     fim
11     Offspring := Busca Local de Pré-Intensificação (Offspring)
12     Computar o rank dos indivíduos e retornar Offspring Rank1
13     Offspring' := Busca Local de Intensificação (Offspring Rank1)
14     Computar o rank e contribuição do hipervolume na população Pop
15     CuttOff da população Pop
16     Adicionar Offspring' à Pop
17     Computar contribuição em Pop, atualizar BestPop e MelhoraVolume
18 fim
19 Busca Local de Pós-Intensificação(BestPop)

```

Linha 1 inicializa os parâmetros da instância a ser resolvida. Loop das linhas 2 a 18 se aplica enquanto houver uma melhora mínima (de valor *minVolume*) no hipervolume da fronteira pareto *BestPop*. Caso seja a primeira iteração gera-se uma certa quantidade de soluções iniciais (*nIniSol*), caso contrário, já existe uma população (chamada *Pop*), e seleciona-se *nFilhos* indivíduos desta população para aplicar um tipo de

cruzamento que será descrito adiante. Cada cruzamento envolvendo duas soluções dá origem a uma solução filha. Ao conjunto de soluções filhas refere-se como *Offspring*. Linha 11 aplica um conjunto mais simples de buscas locais (usando estratégia baseado no VND), chamado “Pré-intensificação” a todas as soluções de *Offspring*. A seguir computa-se o rank destas soluções, seleciona-se as não dominadas (rank 1), chamado de *Offspring Rank 1*, e aplica-se na linha 13 um conjunto mais completo de buscas locais, chamado “Intensificação” que retorna *Offspring'* (resultado das melhorias feitas em *Offspring Rank 1*). Linha 14 calcula a contribuição ao hipervolume de cada indivíduo da população principal (chamada *Pop*), e caso o número de indivíduos da população seja maior que um dado limite, o excedente (composto pelos indivíduos de menor ranking e contribuição) é removido da população. Linha 16 adiciona todos os filhos de *Offspring'* à *Pop*. Linha 17 atualiza o rank e contribuição ao hipervolume para cada indivíduo de *Pop* (usado na linha 6 para seleção), atualiza a população *BestPop* (fronteira Pareto com o melhor conjunto de soluções não dominadas obtido pelo algoritmo até a presente iteração), e calcula o hipervolume de *BestPop* para atualizar *MelhoraVolume* (aumento percentual acumulado do hipervolume nas *I* iterações anteriores). Após a saída do loop principal, aplica-se na linha 19 um conjunto de buscas locais a cada indivíduo da fronteira Pareto *BestPop* para tentar alguma melhora adicional no hipervolume.

O cálculo do hipervolume utilizado na linha 17 foi implementado conforme descrito em Fleischer (2003), e a contribuição de cada solução no volume da fronteira Pareto da população (linha 14) é implementado como descrito em Emmerich et al. (2005).

A Figura 18 a seguir ajuda a entender o funcionamento do algoritmo ao longo das fases de pré-intensificação e intensificação. A linha vertical é o nível de serviço, aqui multiplicado por (-1) para ser minimizado, e a linha horizontal é o custo de transporte.

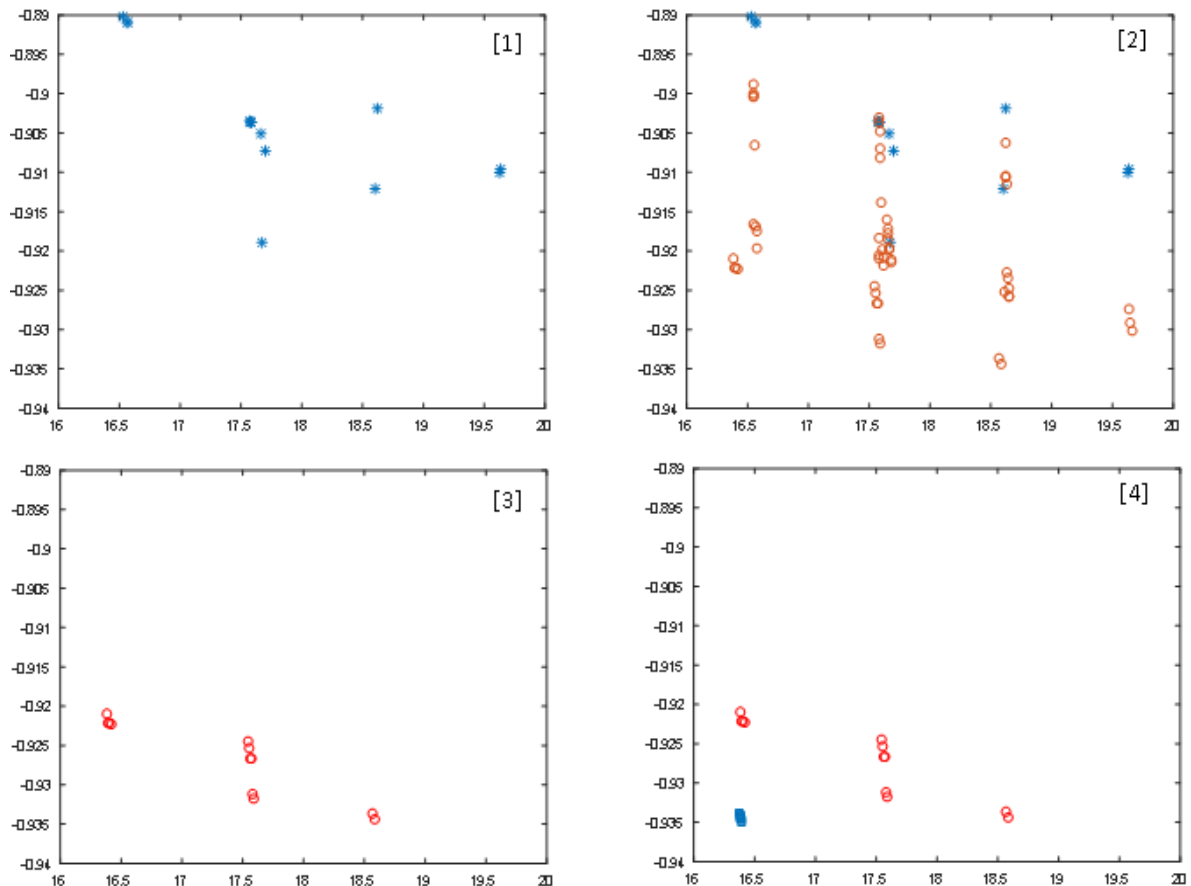


Figura 18: Ilustração da evolução das buscas locais

A Figura 18 (gráfico 1) mostra as soluções iniciais do algoritmo numa dada instância. A Figura 18 (gráfico 2) mostra as mesmas soluções em azul, e as novas soluções em vermelho geradas pela aplicação das buscas locais tipo “pré-intensificação” a cada uma delas. A Figura 18 (gráfico 3) apresenta as soluções não dominadas do gráfico 2. Finalmente, a Figura 18 (gráfico 4) mostra as mesmas soluções do gráfico 3 (em vermelho), e em azul, as novas soluções geradas pela aplicação da busca local tipo “intensificação” a cada solução em vermelho. Vê-se que novas soluções não dominadas foram geradas, contribuindo para o aumento do hipervolume.

Em todos os algoritmos implementados neste trabalho, a solução do PRV tem uma representação matricial e sequencial, ou seja, cada solução é uma matriz em que cada linha é uma rota, e nesta rota os clientes são alocados na sequência em que a rota é percorrida. Pela Figura 19, fica subentendido que o elemento antes do primeiro cliente e após o último cliente de cada linha é o depósito. Esta representação é conhecida como

Multi-part Representation. Esta e outras representações podem ser vistas em Potvin (2007).

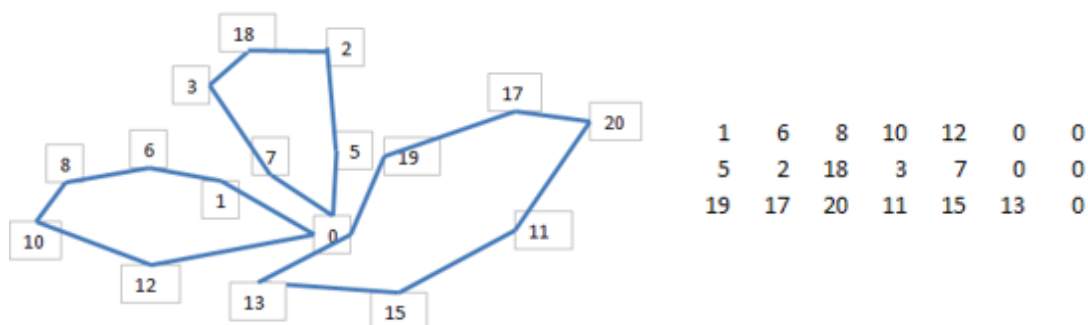


Figura 19: Solução com três rotas e sua correspondente representação matricial

Outras estruturas auxiliares são utilizadas pelo algoritmo. Há uma matriz de duas dimensões para as funções objetivos de cada solução de uma população. Cada linha refere-se a uma solução, uma coluna para o objetivo 1, outra para o objetivo 2, e outras colunas para o rank da solução, contribuição ao hipervolume, número de veículos e tempo de viagem. Há também uma matriz de duas dimensões, cada linha para uma solução, a primeira coluna para o número de veículos de cada solução, e demais colunas com o número de clientes de cada rota.

Referindo-se a parte estocástica, há duas matrizes tridimensionais: $CumFx$, $CumFy$. Ambas se referem à função de distribuição cumulativa discretizada, sendo que a primeira armazena os valores de x e a segunda os valores de $P(x \leq X)$. Uma dimensão refere-se à solução na população, outra dimensão refere-se a cada cliente da respectiva solução, e a terceira dimensão refere-se aos $nint$ valores de x (se $CumFx$) ou de $P(x \leq X)$ (se $CumFy$). Aqui, $nint$ é o número de intervalos no qual a função cumulativa é discretizada, seguindo notação vista no tópico 6.3.2.

Também utiliza-se uma matriz para armazenar o nível de serviço de cada cliente e uma outra matriz para armazenar a soma das violações do nível de serviço dos clientes para cada rota, de cada solução.

6.4.1 – Solução Inicial

O conjunto de soluções iniciais é baseado no algoritmo *I1* de Solomon (1987), descrito no capítulo 3, que é estendido para o problema estocástico tratado neste trabalho. O pseudocódigo é dado a seguir, onde “NS” é o nível de serviço dos clientes.

Algoritmo 4: Gerar solução inicial

```

1  Inicializar Lista de Clientes Sementes
2  Inicializar primeira rota com o primeiro cliente da lista semente
3  q=1; achouInserção=0
4  enquanto q<N
5      para todos os clientes não roteados
6          para cada posição do cliente candidato na rota corrente
7              se janela de tempo determinística é viável então
8                  Calcular Custo de Inserção
9                  se custo é melhor
10                     se NS é viável
11                         Atualizar melhor custo
12                         achouInserção=1
13                     fim
14                 fim
15             fim
16         fim
17     fim
18     se achouInserção=1 então
19         Efetivar inserção e atualizar estruturas auxiliares
20         achouInserção=0
21     ou então
22         Inicializar nova rota e inserir o primeiro cliente segundo lista semente
23     fim
24     q=q+1
25 fim

```

A linha 1 do algoritmo 4 calcula a lista de sementes, conforme equação a seguir:

$$\text{CustoInicialização} = -\lambda d_{0i}^N + \theta l_i^N \quad (76)$$

Na equação (76), λ e θ são pesos a serem escolhidos experimentalmente. Já d_{0i}^N e l_i^N são respectivamente: o tempo médio de viagem entre o cliente i e o depósito, e o fechamento da janela de tempo, todos normalizados entre 1 e 100. A lista de sementes é uma lista ordenada em ordem crescente do custo de inicialização. Desta forma, clientes distantes do depósito e/ou com fechamento da janela de tempo mais largo estarão nas primeiras posições desta lista.

Linha 2 inicializa a primeira rota e a linha 3 inicializa um contador de clientes roteados e uma variável binária que indica se foi encontrado uma posição de inserção para o cliente não roteado candidato à inserção. O loop entre linha 4 e 25 continua até que todos os clientes sejam inseridos na solução. O loop entre linha 5 e 17 seleciona-se cada um dos clientes não roteados que são os candidatos para a inserção. Cada um deles tem sua inserção testada em cada uma das posições da rota corrente. Quando uma posição de inserção viável é encontrada no teste determinístico da janela de tempo, calcula-se o custo da inserção na linha 8, conforme equação (17), explicada no capítulo 3. Se encontrado um melhor custo de inserção, calcula-se o nível de serviço de cada cliente na nova rota (conforme algoritmo 2), se viável, atualiza-se o melhor custo e a variável binária que indica que uma posição de inserção foi encontrada. Após explorar todos os clientes não roteados no loop da linha 5 a 17, caso tenha sido encontrada uma posição de inserção, a inserção é efetivada, caso contrário, uma nova rota é inicializada com o cliente da lista semente de menor custo de inicialização ainda não inserido na solução.

O custo de inserção (equação 17) possui quatro parâmetros: α_1 , α_2 , μ e λ . A equação da lista semente possui outros dois parâmetros: λ e θ . O parâmetro λ aparece em ambos os casos, sendo o peso dado à distância do cliente ao depósito. Ou seja, há cinco parâmetros cujos valores são definidos experimentalmente. Diferentes valores fazem com que diferentes soluções sejam geradas. No algoritmo principal (algoritmo 3), $nIniSol$ soluções iniciais são geradas, cada uma, com uma combinação diferente de valores para os cinco parâmetros do algoritmo de solução inicial.

6.4.2 – Algoritmos VNDs

Conforme visto no loop principal do algoritmo memético (algoritmo 3), existem três conjuntos de buscas locais que utilizam uma estratégia baseada no VND (*Variable Neighborhood Descent*) para realizar a troca sistemática de vizinhanças: pré-intensificação, intensificação e pós-intensificação. O algoritmo 5 detalha a pré-intensificação.

Algoritmo 5: Busca local - Pré-Intensificação

```

1  c=0; popFilha=vazio;
2  para i=1:nFilhos
3      popLocal := população(i)
4      c=c+1
5      enquanto intGen < intGenMax
6          intGen=intGen+1
7          2opt(popLocal, EstruturaAux, idxImp, c)
8          ReallocationChain(popLocal, EstruturaAux, idxImp, c)
9          Interchange(popLocal, EstruturaAux, idxImp, c)
10         Reallocation(popLocal, EstruturaAux, idxImp, c)
11     fim
12     popFilha := popFilha + popLocal
13     c=0
14 fim
15 Retornar popFilha

```

O algoritmo recebe como entrada Offspring, ou seja, uma população de soluções que pode ser uma população oriunda do procedimento que gera a população inicial do algoritmo na primeira geração, ou vinda de um número especificado de cruzamentos, conforme algoritmo 3. A linha 1 inicializa uma matriz com a população filha (“popFilha”) vazia e um contador com o número de soluções nesta população (“c”). Estes, assim como “EstruturaAux” e “idxImp” são variáveis globais no escopo do algoritmo de intensificação, e tratam-se respectivamente de uma estrutura auxiliar com informações da população e um índice.

O loop das linhas 2 a 14 aplica um conjunto de buscas locais a cada indivíduo da população de entrada do algoritmo, aqui identificada simplesmente como “população”. A linha 3 inicializa a população local “popLocal” com um único indivíduo da população de entrada. Loop das linhas 5 a 11 aplica uma sequência de buscas locais por um número especificado de vezes. Há quatro buscas locais que serão detalhadas mais adiante.

Inicialmente a primeira busca local recebe a população filha com apenas uma solução, realiza os movimentos correspondentes a estrutura de vizinhança desta busca, sendo que cada movimento dá origem a uma nova solução, e cada solução viável não dominada pelas soluções que compõem a população local é adicionada a esta população. Desta forma, diferentemente de buscas locais tradicionais, a busca local aqui não retorna uma única solução, mas sim um conjunto de soluções não dominadas (“popLocal”), mais alinhado com os princípios da otimização multiobjetivo, e evita desperdiçar outras boas

soluções, caso fosse escolhido retornar apenas uma solução. Cada busca local também retorna o índice “idxImp” que corresponde a “melhor” (a ser discutido adiante) solução desta população local.

Esta população é então entrada para a próxima busca local que começa realizando os movimentos de sua vizinhança explorando uma solução de sua população de entrada, identificada pelo índice “idxImp”. A partir daí cada busca local recebe um conjunto de soluções não dominadas, realiza movimentos com o objetivo de encontrar outras soluções não dominadas e dominantes para retornar uma população local melhor que a população de entrada.

A linha 12 adiciona esta população local a uma população filha chamada “popFilha”. Então a população local “popLocal” é reinicializada com outra solução de “população”, as buscas locais são realizadas, e um novo conjunto de soluções “popLocal” é adicionado a “popFilha”. Após “nFilhos” iterações retorna-se “popFilha”, equivalente ao observado na Figura 18 (gráfico 2).

O algoritmo do conjunto de buscas locais do tipo “intensificação” é descrito no algoritmo 6.

Algoritmo 6: Busca local - Intensificação

```

1  c=0;
2  popLocal := Offspring Rank1
3  para i=1:nSol
4      c=c+1
5      VolImprov := VolMin + 1
6      enquanto VolImprov > VolMin
7          intGen=intGen+1
8          2opt(popLocal, EstruturaAux, idxImp, c)
9          ExchangeIntra(popLocal, EstruturaAux, idxImp, c)
10         Interchange(popLocal, EstruturaAux, idxImp, c)
11         Reallocation(popLocal, EstruturaAux, idxImp, c)
12         2opt*(popLocal, EstruturaAux, idxImp, c)
13         Atualizar VolImprov
14     fim
15     c=0
16 end
17 Retornar popLocal

```

O algoritmo de “intensificação” é semelhante ao de “pré-intensificação”, com algumas diferenças. Uma diferença é que a população de entrada trata-se do conjunto de

soluções não dominadas retornado por “pré-intensificação”(Offspring Rank1), sendo a população local inicializada por este conjunto. Outra diferença é que há duas buscas locais adicionais (2opt* e “ExchangeIntra”) e não se utiliza a busca local “ReallocationChain”. O loop das linhas 6 a 14 utiliza um critério de saída baseado na melhoria do hipervolume da população local. O cálculo do hipervolume é feito para valores normalizados dados no capítulo de anexos. Última diferença é que a população local inicializada na linha 2 pela população Offspring Rank1 não é reinicializada para cada novo indivíduo do loop das linhas 13 a 16. Assim, todas as buscas locais sempre trabalham com a população local *popLocal*, sendo esta população retornada pelo algoritmo na linha 17.

Já o algoritmo de “pós-intensificação” é semelhante ao de “intensificação”, com duas diferenças. A entrada é a população composta pela melhor fronteira pareto do algoritmo principal (“BestPop” no algoritmo 3) e as buscas locais são apenas três: 2opt, Interchange e Reallocation.

Convém observar que em todas as fases de descida (pré-intensificação, intensificação e pós-intensificação), após a execução de uma busca local, executa-se a próxima busca local, tenha a busca local encontrado/adicionado ou não uma nova solução não dominada à população local. Num algoritmo VND, quando uma busca local encontra uma melhor solução, retorna-se para a primeira busca local da sequência definida.

6.4.3 – Buscas Locais

As buscas locais implementadas seguem a estrutura descrita no algoritmo 7.

Algoritmo 7: Estrutura básica da busca local

```

1 bestLocalSol := popLocal(idxImp)
2 para cada movimento elegível e viável em bestLocalSol
3     AvaliarObjetivos(newSol)
4     Better(bestLocalSol,newSol)
5     se Better = true então
6         bestLocalSol=newSol
7     fim
8     Adicionar à "popLocal" se "newSol" não dominada
9 fim
10 Retornar "popLocal" e "idxImp"

```

A busca local tem como entrada uma população “popLocal”. Linha um seleciona uma solução, usando o índice “idxImp” que também é uma entrada da busca. Loop entre

as linhas 2 e 9 realiza os movimentos para a respectiva estrutura de vizinhança. Nem todas os movimentos são “elegíveis” (descrito mais adiante), e dentre estes, as soluções viáveis possuem os objetivos avaliados na linha 3. A função “Better” avaliar se a nova solução (“newSol”) é “melhor” que a solução corrente “bestLocalSol”. Se sim, ela substitui a melhor solução corrente na linha 6. Linha 8 testa se a nova solução é não dominada por todas as soluções da população local, e as soluções dominadas pela nova solução são removidas da população. Linha 10 retorna a população local e o índice que identifica com a solução “bestLocalSol”.

A função “Better” compara a solução corrente com a nova solução. Considera-se que a nova solução é “melhor” que a solução corrente caso:

- 1) A nova solução domina a solução corrente.
- 2) Se a melhora percentual em um objetivo é maior que a deterioração percentual no outro objetivo. Nas equações (77) e (78) a seguir, isto ocorre quando $melhoraObj1 + melhoraObj2 > 0$.

$$melhoraObj1 = \left(\frac{(Obj1SoluçãoCorrente - Obj1SoluçãoNova)}{abs(Obj1SoluçãoCorrente)} \right) * 100 \quad (77)$$

$$melhoraObj2 = \left(\frac{(Obj2SoluçãoCorrente - Obj2SoluçãoNova)}{abs(Obj2SoluçãoCorrente)} \right) * 100 \quad (78)$$

Onde *Obj1SoluçãoCorrente* é o valor do objetivo 1 da solução corrente. Demais, análogo. Vale observar que no contexto da otimização multiobjetivo, apenas a condição 1 permite dizer que uma solução é “melhor” que a outra. Então, no presente contexto, “melhor” trata-se de um abuso de linguagem, mas com uma definição muito clara expressa pelas condições 1 e 2.

As buscas locais implementadas no algoritmo memético foram citados nos algoritmos 5 e **Error! Reference source not found.:**

- 1) “2opt”: Figura 2.
- 2) “ExchangeIntra”: Figura 3, mas envolvendo trocas na mesma rota.
- 3) “Interchange”: Figura 4 (“Intercâmbio”) e detalhado no algoritmo 8 a seguir.
- 4) “Reallocation”: Figura 3 (“Realocação”)
- 5) “2opt*”: Figura 5.
- 6) “ReallocationChain” ou “Realocação em Cadeia”, descrito no algoritmo 9, adiante.

O algoritmo 8 fornece o pseudocódigo da busca local “Intercâmbio” (descrita na Figura 4) em que duas rotas são selecionadas, a seguir um cliente de cada rota é selecionada, então a troca entre estes dois clientes é avaliada. “NS” é o nível de serviço dos clientes.

Algoritmo 8: Busca local do tipo “Intercâmbio”

```

1  para cada rota r1
2    para cada rota r2 (diferente de r1)
3      se rota não é tabu
4        para cada cliente em r1
5          para cada cliente em r2
6            se dentro da zona circular
7              se restrição de capacidade é atendida
8                se movimento não é tabu
9                  se janela de tempo determinística ok
10                 Computar NS nova rota 1
11                 se a penalidade para o NS não aumentou
12                 Computar NS nova rota 2
13                 se a penalidade para o NS não aumentou
14                 Avaliar FO
15                 Better(bestLocalSol,newSol)
16                 se Better = true
17                   bestLocalSol=newSol
18                 fim
19                 Adicionar à "popLocal" se não dominada
20               fim
21             ... fim(s)
22 fim

```

Loops que se iniciam nas linhas 1 e 2 percorrem as rotas da solução para selecionar duas rotas diferentes. Linha 3 testa se as rotas envolvidas neste movimento são ou não tabu (explicado adiante). Caso possam ser exploradas (não-tabu), loops iniciados nas linhas 4 e 5 selecionam os dois clientes. Linha 6, se os clientes envolvidos no movimento estão próximos o suficiente (dentro da zona circular), então testa-se na linha 7 o atendimento às restrições de capacidade das duas rotas envolvidas ($r1$ e $r2$). Linha 8, se o movimento já foi explorado antes e resultou numa solução dominada, ele é considerado tabu. Linha 9 testa a janela de tempo segundo o modelo determinístico, por se tratar de um teste mais rápido. A seguir calcula-se o nível de serviço dos clientes da nova rota $r1$ e a soma da violação da restrição de nível de serviço para cada cliente da rota. Caso esta violação não tenha aumentado, calcula-se o nível de serviço dos clientes

da nova rota $r2$ e repete-se o teste de violação do nível de serviço. Então a função objetivo é avaliada, a seguir testa-se se a nova solução é considerada melhor que a corrente (equações (77) e (78)), e caso a nova solução não seja dominada por qualquer solução da população local, ela é inserida à população.

As linhas 10 e 12 além de calcular o nível de serviço nos clientes, usa esta informação para calcular a violação total do nível de serviço de uma dada rota r , dado por $VTNS^r = \sum_{i=1}^I \max(0, \alpha - NS_i)$ onde I é a quantidade de clientes da rota r , α é o nível de serviço mínimo especificado e NS_i é o nível de serviço calculado no cliente i . Assim, a violação total do nível de serviço para uma solução s é $VTNS^s = \sum_{k=1}^K VTNS^r_k$ onde k é a quantidade de veículos da solução s . A linha 14 ao calcular o novo valor das funções objetivos, adiciona a cada objetivo um valor de penalidade igual a $VTNS^s * PenF$ onde $PenF$ é um parâmetro do algoritmo. Quanto maior $PenF$ mais rápidas soluções que violam o nível de serviço serão eliminadas da população principal e analogamente será mais difícil que uma solução violada seja adicionada à população.

No teste da linha 3, o par de rotas envolvido no movimento é considerado tabu quando este mesmo par já foi explorado anteriormente no algoritmo e resultou em rotas que não foram inseridas à população local, por serem dominadas, inviáveis ou não elegíveis (exemplo: não atender ao requisito da zona circular ou janela de tempo determinístico). Analogamente, no teste da linha 8, considera-se que um movimento é tabu quando o mesmo par de rotas, com os mesmos dois clientes selecionados, resultou em rotas que não foram inseridas à população local.

Para identificar um par de rotas como tabu, criou-se uma indexação que identifica com um único valor um determinado par de rotas. Isto é feito conforme equação (79). Seja o vetor b_i o tempo de início de atendimento determinístico do cliente i para $i = 1, \dots, N$ clientes da instância. Seja $r1$ um vetor com a sequência dos clientes de uma das rotas do par selecionado no movimento e que possua $nc1$ clientes com $k = 1, \dots, nc1$. Seja a outra rota do par, um vetor $r2$ com $nc2$ clientes com $k = 1, \dots, nc2$. NS é um vetor com o nível de serviço de todos os clientes da solução.

$$idxR = \left\| \left(\sum_{k=1}^{nc1} b_{r1_k} + \sum_{k=1}^{nc2} b_{r2_k} \right) * 1000 \right\| + \left\| \sum_{k=1}^{nc2} NS_{r2_k} \right\| + r1_1 \quad (79)$$

Este índice é utilizado num vetor binário TR_{idxR} . Se $TR_{idxR}=1$, o par de rotas é considerado tabu, caso contrário, não-tabu.

Analogamente, para indexar os movimentos, utilizou-se a equação (80), onde Δ_1 é a variação no tempo de viagem da rota $r1$ quando o cliente $u2$ oriundo da rota $r1$ é inserido na rota $r2$ no lugar do cliente $u1$. Δ_2 é o análogo para a rota $r2$.

$$idxMov = idxR + \|(\Delta_1 + \Delta_2) * 100\| + \|(b_{u1} + b_{u2}) * 100\| \quad (80)$$

Este índice é utilizado em um vetor binário TM_{idxMov} . Se $TM_{idxMov}=1$, o movimento é considerado tabu, caso contrário, não-tabu. Estes índices foram determinados experimentalmente de forma a reduzir a ocorrência de pares de rotas ou movimentos diferentes sejam indexados com o mesmo valor.

A linha 6 do algoritmo 8 testa se os clientes envolvidos num certo movimento estão dentro da mesma zona circular. Conforme Figura 20, uma zona circular de raio R_G com o cliente u_i no centro, de modo que os clientes adjacentes u_{i-1} e u_{i+1} devem estar dentro deste círculo. Esta é uma estratégia similar à utilizada pelo algoritmo *Granular Tabu Search* introduzindo por Toth & Vigo (1998b) e visa evitar arcos muito longos nas soluções, a partir da observação de que tais arcos não fazem parte de boas soluções.

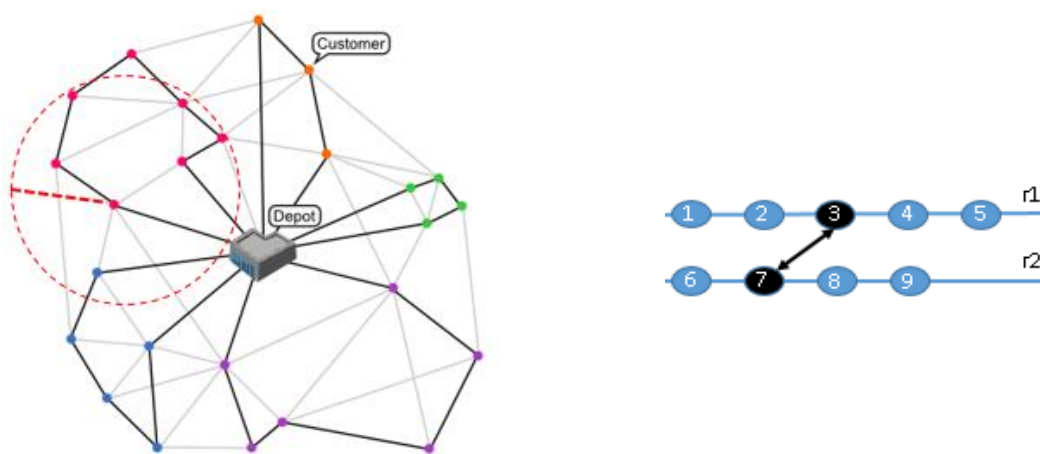


Figura 20: Zona circular

Na Figura 20 (lado direito), quando o movimento de intercâmbio entre os clientes 3 e 7 está sendo avaliado, testa-se se: $TT_{3,6} \leq R_G$ e $TT_{3,8} \leq R_G$ para a nova posição do cliente 3 na rota $r2$, e $TT_{2,7} \leq R_G$ e $TT_{7,4} \leq R_G$ para a nova posição do cliente 7 na rota $r1$. Existe uma exceção: caso $\Delta_1 + \Delta_2 \leq 0$, ou seja, o novo tempo de viagem é menor ou igual ao tempo de viagem do par original de rotas, então o critério de zona circular não é considerado. Entende-se que não faz sentido restringir um movimento cuja distância está sendo melhorada.

A utilização de rotas e movimentos tabus, zonas circulares e teste determinístico das janelas de tempo nas buscas locais tem como objetivo a redução do tempo computacional e a influência destas estratégias no tempo é mostrado adiante nos experimentos realizados.

A busca “Realocação em Cadeia” (“ReallocationChain”) foi implementada com o objetivo principal de reduzir o número de rotas, sendo utilizada na fase de “pré-intensificação” (algoritmo 5). Dentre todas as buscas locais implementadas neste trabalho, apenas “Realocação” também pode reduzir o número de rotas quando uma das rotas envolvidas no movimento possui apenas um cliente. “Realocação em Cadeia” trata-se de uma combinação dos movimentos de “Realocação” e de “Intercâmbio”, conforme algoritmo 9. Para uma dada solução, seleciona-se uma rota $r1$, começando-se com aquela com menor número de clientes e então aplica-se a busca “Realocação” para tentar realocar todos os clientes de $r1$ para outras rotas da solução. Caso não seja possível, aplica-se a busca “Intercâmbio” para trocar um cliente remanescente de $r1$ por algum outro cliente. Então aplica-se “Realocação” novamente para tentar realocar o novo cliente inserido por “Intercâmbio” em alguma outra rota.

Algoritmo 9: Busca local do tipo “Realocação em cadeia”

```

1  rotaEliminada= 1
2  fezIntercambio=0
3  qt=0
4  enquanto rotaEliminada=1 e qt < LimiteTentativas
5      se rotaEliminada= 1 então
6          Atualizar matriz  $m$  com o número de veículos de cada rota
7          Ordenar  $m$  em ordem crescente de clientes
8      fim
9      se fezIntercambio= 0 então
10         qt = qt + 1
11     fim
12     rotaEliminada=0
13     r1=  $m$  (qt,1)
14     Realocação(solução, r1,rotaEliminada)
15     se rotaEliminada= 0 então
16         fezIntercambio=0;
17         Intercâmbio(solução, r1,fezIntercambio)
18     fim
19 fim

```

As linhas 1 e 2 inicializam duas variáveis binárias: “rotaEliminada” e “fezIntercambio”. A primeira tem valor 1 quando a busca “Realocação” consegue eliminar uma rota, e a segunda tem valor 1 quando a busca “Intercâmbio” conseguiu intercambiar um cliente de $r1$. Linha 3 inicializa um contador de tentativas. Linha 4 inicializa o loop principal que continua as iterações enquanto conseguir eliminar rotas ou o número máximo de tentativas não tenha sido excedido ($LimiteTentativas \leq nr$), onde nr é o número de rotas da solução. Entre linhas 5 e 8, caso rota tenha sido eliminada, reinicializa-se na linha 6 a matriz m que possui na coluna 1 o índice da rota e na coluna 2 a quantidade de clientes na respectiva rota. A linha 7 ordena m em ordem crescente de clientes. Linhas 9 a 11, caso intercâmbio não tenha tido sucesso (ou seja, não foi possível trocar qualquer cliente de $r1$ com alguma outra rota), incrementa-se o contador qt , isto significa que o algoritmo desiste de tentar eliminar a rota $r1$ corrente, e vai tentar a eliminar outra rota. Linha 12 reinicializa “rotaEliminada”, linha 13 seleciona a próxima rota $r1$ a tentar eliminação e a linha 14 aplica a busca “Realocação” à rota $r1$ selecionada. Esta busca possui “solução” e “rotaEliminada” como parâmetros de entrada e de saída, enquanto $r1$ é um parâmetro de entrada para indicar qual rota deve ter seus clientes realocados. Caso a busca consiga realocar todos os clientes de $r1$, retorna-se “rotaEliminada”=1 e “solução” com a nova solução. Caso a rota não tenha sido eliminada, linhas 15 a 18 aplicam a busca “Intercâmbio” para substituir algum dos clientes de $r1$. Esta busca possui “solução” e “fezIntercambio” como parâmetros de entrada e de saída, enquanto $r1$ é um parâmetro de entrada para indicar qual rota deve ter pelo menos um cliente intercambiado. Caso a busca consiga fazer algum movimento de intercâmbio, retorna-se “fezIntercambio=1” e “solução” com a nova solução.

Sendo assim, quando “Realocação” consegue eliminar a rota $r1$ corrente, a matriz m é recriada para esta nova solução, e tenta-se novamente eliminar rotas partindo-se da rota com menor número de clientes. Caso “Realocação” não tenha conseguido eliminar a rota $r1$, aplica-se “Intercâmbio” nesta rota. Caso “Intercâmbio” tenha conseguido fazer pelo menos um movimento, qt não é incrementado para que se faça uma nova tentativa na rota $r1$ modificada por “Intercâmbio”. Caso “Intercâmbio” não consiga realizar qualquer movimento, qt é incrementado para tentar-se eliminar uma outra rota $r1$.

6.4.4 – Operador de Seleção

Ao final de uma geração, seleciona-se uma certa quantidade de soluções da população para participar das operações de cruzamento que darão origem a novas soluções. Isto é feito no algoritmo principal (algoritmo 3), linha 6.

A seleção é feita utilizando-se o método *Stochastic Universal Sampling* (SUS) desenvolvido por Baker (1987) e foi escolhido para este trabalho por ser um método que não apresenta desvios entre a taxa de cruzamento esperada e a amostragem do algoritmo. Além disto, o método possui espalhamento mínimo (menor intervalo de valores possíveis). Um estudo detalhado comparando-se diferentes métodos de seleção (dentre eles “método da roleta” e “seleção por torneio”) é visto em Blickle & Thiele (1996).

O método de seleção tem como entrada a matriz com os valores das funções objetivos, rank e contribuição ao hipervolume de cada solução da população. As outras entradas são o tamanho da população (*nbpop*) e a quantidade de indivíduos a serem selecionados (*nrel*). Os passos são:

- 1) Calcular o percentual de soluções de cada *rank*
- 2) Calcular a quantidade a ser selecionada de cada *rank*
- 3) Tratar hipervolume infinito dos extremos de cada rank
- 4) Para cada *rank*, aplicar o método SUS usando como critério a contribuição ao hipervolume.

O primeiro passo, apenas calcula o percentual de cada *rank*. Exemplo: para uma população com 43 indivíduos, sendo 33 rank 1 e 10 rank 2, tem-se 77% rank 1 e 23% rank 2. No passo 2, considerando-se *nrel* = 10, tem-se $0.77 * 10 \cong 8$ indivíduos a serem selecionados de rank 1, e $0.23 * 10 \cong 2$ de rank 2, totalizando as 10 seleções desejadas. No passo 3, considerando que os indivíduos nos extremos de cada rank possuem contribuição de hipervolume igual a infinito, atribui-se um valor de volume igual ao dobro do maior volume diferente de infinito. Assim, haverá uma alta probabilidade de seleção destes extremos. Finalmente, o passo quatro aplica o método SUS para cada rank. No exemplo, será aplicada SUS duas vezes. Primeiro para rank 1, selecionando-se 8 indivíduos dentre os 33 de rank 1, e depois, selecionando-se 2 indivíduos dentre os 10 de rank 2.

A saída da função é um vetor *vs* com o índice de cada indivíduo selecionado.

6.4.5 – Operador de Cruzamento

Após selecionado os indivíduos que irão participar das operações de cruzamento, este é feito utilizando-se um método baseado em Alvarenga & Mateus (2004) no algoritmo principal (algoritmo 3), loop entre linhas 7 e 9. Dado um vetor com o índice de cada indivíduo selecionado para o cruzamento (vetor vs do tópico 6.4.4), para cada cruzamento, escolhe-se dois indivíduos ($pai1$ e $pai2$) que darão origem a outro indivíduo ($filho$). No algoritmo 3, ao desejar-se $nFilhos$, seleciona-se $nFilhos$ soluções formando um vetor vs de tamanho $nFilhos$. Cada par de pais ($pai1$ e $pai2$) é formado a partir de $nFilhos$ escolhas em posições adjacentes de vs , sendo que a primeira escolha seleciona os pais do índice 1 e 2, a segunda escolha usa índices 2 e 3 e assim sucessivamente, de modo que na última escolha serão usados a última e a primeira posição de vs . Caso os pais sejam os mesmos, incrementa-se o índice de $pai2$ até encontrar-se um diferente.

As etapas são descritas a seguir:

- 1) Para cada rota de $pai1$, calcular quantidade de rotas de $pai2$ com algum cliente em comum. Fazer o mesmo para as rotas de $pai2$ em relação a $pai1$. Então, criar um vetor ordenado em ordem crescente para cada uma das situações.
- 2) Inicializar $filho$, escolhendo 1 rota de $pai2$ (ainda não selecionada) com a menor quantidade de rotas com clientes em comum em $pai1$, e copiar esta rota.
- 3) Escolher uma rota de $pai1$ (ainda não selecionada) com a menor quantidade de rotas com clientes em comum em $pai2$. Caso não haja clientes já inseridos em $filho$, copiar toda a rota. Caso haja, identificar qual é a posição i com o primeiro cliente já inserido em $filho$ e copiar esta rota do primeiro cliente até o cliente $i - 1$.
- 4) Escolher uma rota de $pai2$ (ainda não selecionada) com a menor quantidade de rotas com clientes em comum em $pai1$, e copiar esta rota. Caso não haja clientes já inseridos em $filho$, copiar toda a rota. Caso haja, identificar qual é a posição i com o primeiro cliente já inserido em $filho$ e copiar esta rota do primeiro cliente até o cliente $i - 1$.
- 5) Repetir passos 2 a 4 até que um deles não seja mais possível.
- 6) Se existirem clientes não incluídos em $filho$ (clientes não roteados) completar solução aplicando uma adaptação da heurística usada para gerar a solução inicial (algoritmo 4, tópico 6.4.1). Isto pode ser feito mais de uma vez, de modo a escolher a solução

completa com melhor avaliação da função objetivo para ser retornada pela função de cruzamento.

Duas modificações do algoritmo de Alvarenga & Mateus (2004) foram realizadas para permitir que o filho possa herdar mais características dos pais. O passo 1 tem como objetivo, escolher primeiro as rotas de um pai com menos clientes em comum no outro pai, com o objetivo de maximizar a quantidade de rotas e clientes herdados dos respectivos pais para o filho.

Outra mudança é que, nos passos 3 e 4, quando a rota selecionada de um *pai* possui um cliente já inserido em *filho*, ela não é descartada, mas inserida da primeira posição até posição $i - 1$ com os clientes ainda não roteados. Isto significa que o algoritmo copia de forma parcial a rota de *pai*, o que permite herdar informações importantes do *pai*, como os primeiros clientes da rota e orientação da rota. Naturalmente, estas rotas parciais precisam ser completadas no passo 6. O algoritmo para completar as soluções usa a mesma estratégia da solução inicial, e para cada cliente não roteado, primeiro tenta inseri-lo nas rotas de *filho* já existentes, e caso não seja possível, adiciona uma rota.

Este capítulo detalhou a metaheurística desenvolvida para resolver o problema e a mesma é testada em experimentos descritos no próximo capítulo.

6.5 – DISCUSSÃO DO CAPÍTULO 6

O tópico 6.1 deste capítulo ilustrou com exemplos como a distribuição de probabilidade do tempo de chegada pode ser bem diferente de uma distribuição normal. A seguir, este capítulo apresentou os dois pontos mais importantes do presente trabalho:

1. O método para somar uma variável aleatória truncada que não segue uma distribuição normal de probabilidade com outra variável aleatória normal (tópicos 6.2 e 6.3). Isto é usado para computar o nível de serviço nos clientes.
2. O algoritmo memético de otimização para resolver o problema multiobjetivo, com detalhes das estratégias utilizadas para encontrar um conjunto com soluções não dominadas relacionado ao custo de transporte e nível de serviço dos clientes (tópico 6.4).

CAPÍTULO 7

RESULTADOS

Neste capítulo são conduzidos dois tipos de experimentos: um para validar o método estatístico proposto para calcular o nível de serviço nos clientes e outro em que a metaheurística desenvolvida é utilizada para resolver todas as 56 instâncias de Solomon (1987) numa versão estocástica e multiobjetiva. Todos os experimentos foram realizados utilizando-se o software Matlab R2015a e um laptop com 16GB RAM, processador Intel i7, 2.60 GHz.

7.1 – RESULTADOS PARA O MÉTODO ESTATÍSTICO

O método desenvolvido neste trabalho, explicado no tópico 6.3, é comparado com o método proposto por Zhang *et al.* (2013). O tamanho do erro de cada método é dado por sua diferença em relação aos resultados obtidos por simulação estocástica utilizando 20000 rodadas. Um pseudocódigo de como realizar esta simulação é dado em Li *et al.* (2010).

O método de Zhang foi implementado conforme passos descritos no tópico 4.1.4 e o método proposto nesta tese conforme algoritmo 2. Um conjunto de instâncias de teste com 9037 rotas foi especialmente criado para gerar situações em que o efeito do tempo de espera no tempo de chegada do veículo é significativo. Para isto, as 56 instâncias de Solomon foram resolvidas utilizando o algoritmo *I1* de Solomon (1987), selecionando-se aproximadamente 160 rotas diferentes, com pelo menos 3 clientes cada, para cada uma das 56 instâncias, totalizando 9037 rotas com número de clientes variando de 3 a 55 por rota. Todas as instâncias usadas neste experimento possuem os mesmos parâmetros do original, como a capacidade dos veículos, coordenadas geográficas dos clientes, matrizes de distância e tempo de serviço. Apenas um desvio padrão para o tempo de viagem e tempo de serviço foi criado.

Seja $dis(i, j)$ a distância entre dois clientes, o tempo de viagem é dado por uma distribuição normal com média $E[TT_{i,j}] = dis(i, j)$ e desvio padrão $dev[TT_{i,j}] = U[0.1; 0.6] * dis(i, j)$ onde U é uma distribuição uniforme. O tempo de serviço também

é dado por uma distribuição normal com média $E[ST_i] = s_i$ onde s_i é o tempo de serviço determinístico da instância original de Solomon. O desvio padrão do tempo de serviço é $dev[ST_i] = U[0.1; 0.6] * E[ST_i]$. Utilizou-se um intervalo $U[0.1; 0.6]$ um pouco maior que o utilizado em Zhang *et al.* (2013) que é $U[0.2; 0.6]$. Como uma segunda referência, Li *et al.* (2010) utiliza uma proporção do desvio padrão que varia de 0.07 a 0.2 o valor da média. Além destas referências da literatura, em aplicações práticas, dificilmente será encontrado um desvio padrão muito grande quando comparado com a média, pois nestes é mais provável que a distribuição não seja normal, mas sim gamma ou lognormal. Desta forma acreditamos que o intervalo utilizado neste trabalho (0.1; 0.6) seja adequado.

O número total de clientes nesta base de testes é 101701 (não considerando o primeiro cliente de cada rota onde o tempo de chegada é perfeitamente normal). Considerando que dois cálculos de probabilidade são realizados em cada cliente (probabilidade de chegar depois do fechamento e antes da abertura da janela de tempo), o número total de cálculos de probabilidade para validar a precisão dos métodos é 203402.

7.1.1 – Estudo de parâmetros e versões

O algoritmo proposto para computar o nível de serviço possui dois parâmetros principais: $nint$ (número de intervalos usados na discretização da função de distribuição cumulativa) e $I2$ (número de intervalos usados para resolver a convolução. O valor de $nint$ é usado na linha 9 do algoritmo 2, e o de $I2$ é usado na linha 14 do algoritmo 1. Para o experimento, selecionou-se um conjunto de valores (10,15,20,25,30,40,50) para os dois parâmetros, portanto $7*7 = 49$ configurações, e assim analisar sua influência na acurácia e tempo de execução do algoritmo.

Implementou-se três versões do algoritmo proposto:

- V1: Assume normal não truncada, ignorando o truncamento no ponto zero para o tempo de serviço e tempo de viagem.
- V2: Considera o truncamento para ambas as variáveis, aplicando equação (63) sempre que $P(X < 0) \geq 1\%$.
- V3: Considera parcialmente o truncamento, calculando a média e desvio truncados conforme equações (74) e (75) sempre que $P(X < 0) \geq 1\%$, porém assumindo normalidade para a forma da distribuição.

As versões *V2* e *V3* utilizam $P(X < 0) \geq 1\%$ pois estudos preliminares mostraram que para $(X < 0) < 1\%$, o efeito do truncamento pode ser desprezado. As figuras 21 e 22 mostram respectivamente, o tempo computacional e a acurácia (percentil de 95% do erro) para as 49 combinações dos parâmetros *nint* e *I2*.

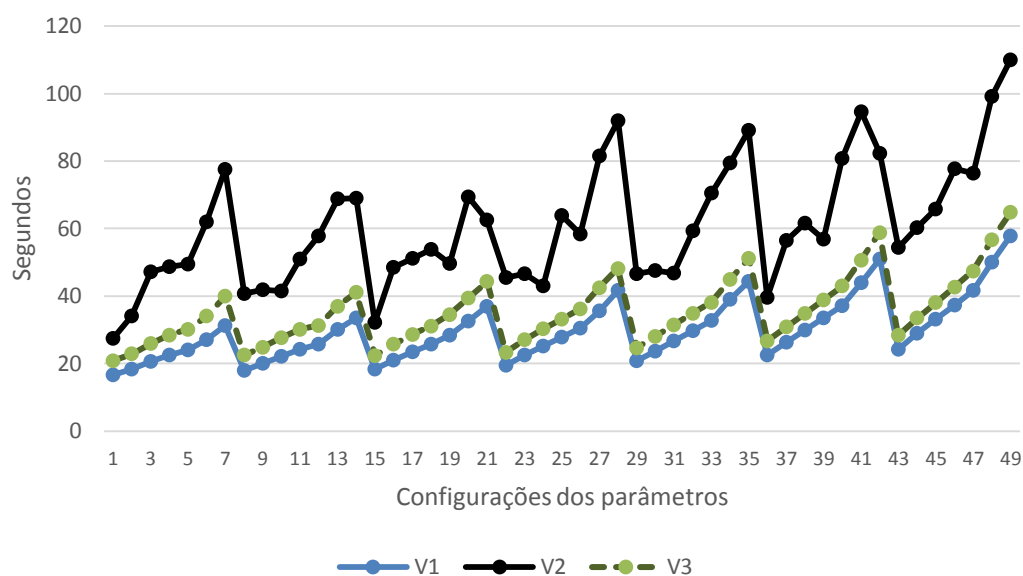


Figura 21: Tempo de execução para as três versões

A Figura 21 mostra que o tempo computacional da versão *V2* é significativamente superior às demais, enquanto as versões *V1* e *V3* possuem tempos mais próximos. Este resultado pode ser associado à utilização da equação (63) para tratar adequadamente o truncamento do tempo de serviço e de viagem.

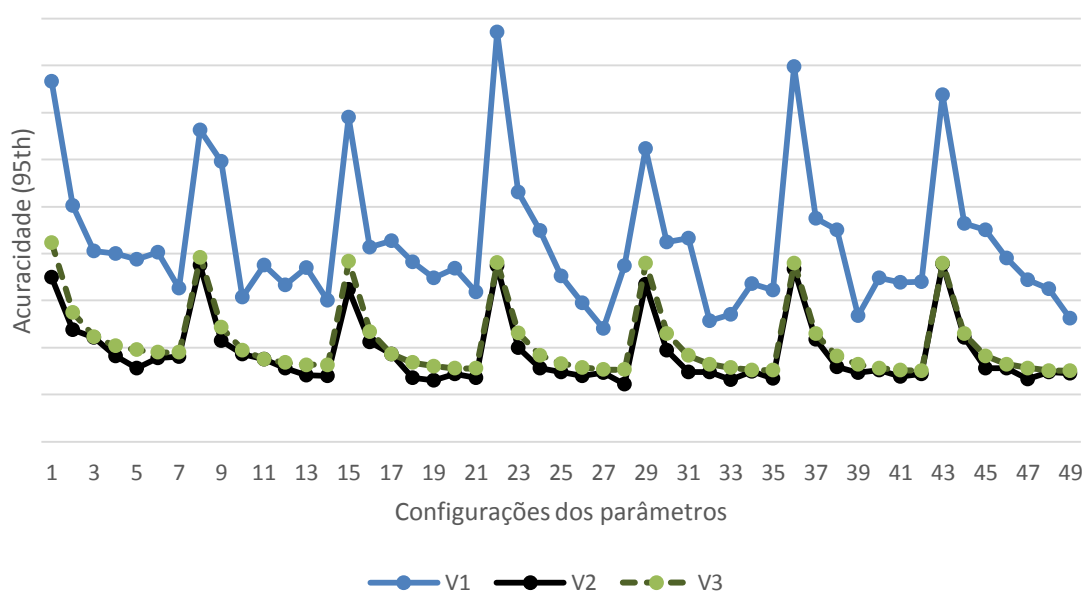


Figura 22: Erro (percentil de 95%) para as três versões

Figura 22 mostra que a versão *V1* possui erros significativamente maiores que as demais, mostrando o impacto de se ignorar o truncamento no ponto zero. Já as versões *V2* e *V3* apresentam erros próximos.

A Tabela 5 mostra a média (dentre os 49 cenários) para acurácia e tempo computacional, reforçando o observado nas figuras anteriores. A versão *V3* parece ser aquela com melhor relação custo-benefício, pois apresenta uma acurácia um pouco pior que a versão *V2*, porém tempo computacional bem melhor. Enquanto a versão *V1* pode ser considerada inadequada pois embora possua bom tempo computacional, a qualidade da solução é significativamente pior.

Tabela 5: Resumo do desempenho das versões

Versão	Segundos	Acurácia (Percentil 95%)
V1	29.75	2.13
V2	60.62	0.95
V3	35.35	1.05

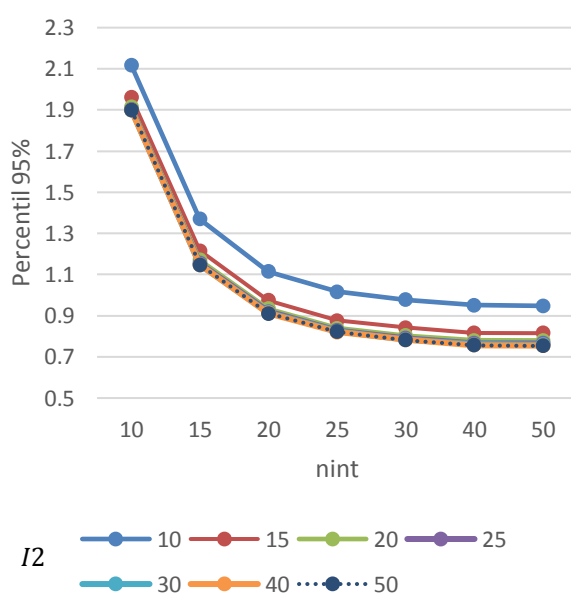


Figura 23: Influência no erro

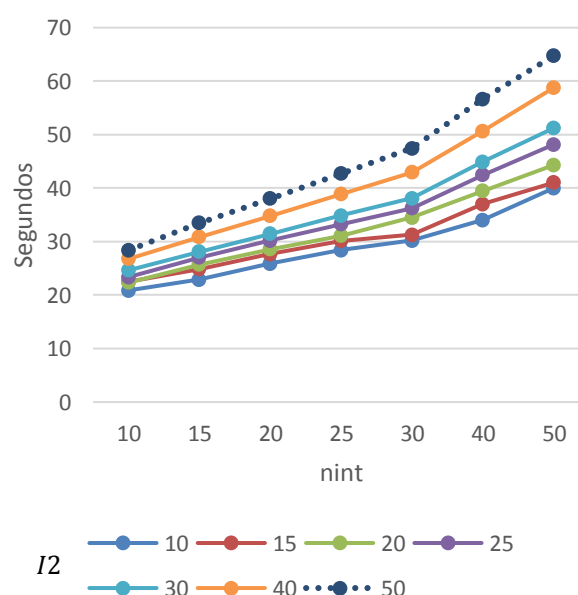


Figura 24: Influência no tempo

As figuras 23 e 24 mostram como $nint$ e $I2$ influenciam o desempenho do algoritmo quanto ao erro e tempo de execução (dados obtidos para a versão *V3*). Pela Figura 23, $I2 = 10$ tem acurácia significativamente pior que os demais, que entre si, possuem acurácia bem similar. Já pela Figura 24, $I2 = 10$, possui melhor tempo, mas

similar ao valor obtido para $I2 = 15$ e $I2 = 20$. Quanto ao parâmetro $nint$, em termos de acurácia, percebe-se que o resultado começa a estabilizar a partir de $nint = 30$ e que ele possui grande influência no tempo computacional.

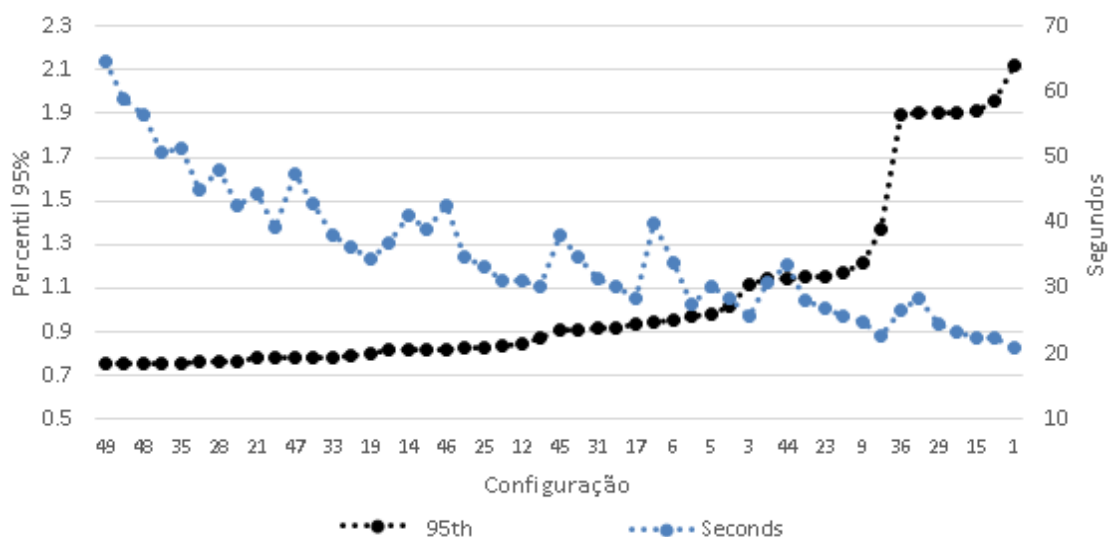


Figura 25: Tempo e erro para as 49 configurações de parâmetros

A Figura 25 mostra de forma mais clara a relação entre o tempo e o erro para as configurações dos parâmetros e ajudam a escolher uma combinação de parâmetros mais adequada. A escolha depende do contexto e preferências do tomador de decisão, aqui julga-se que a configuração 17 com erro em torno de 0.93% e tempo em torno de 28 segundos, equivalente aos parâmetros $nint = 20$ e $I2 = 20$, possui uma boa relação de custo benefício entre tempo e qualidade (neste contexto, medida pelo tamanho do erro).

7.1.2 – Comparação com benchmark da literatura

Este tópico tem como objetivo comparar o método proposto para calcular o nível de serviço com a nossa implementação do método descrito em Zhang *et al.* (2013), usado como benchmark. A Tabela 6 mostra o resumo dos resultados. A segunda coluna descreve as quatro métricas usadas para comparar os métodos. As primeiras três métricas fornecem a precisão do cálculo em pontos percentuais (p.p): a média do erro absoluto, o desvio padrão do erro absoluto e o percentil de 95% do erro absoluto. A métrica 4 é o tempo de execução em segundos (média de cinco rodadas). O tempo de execução é dado pelo tempo necessário para calcular as probabilidades do banco de instâncias. A terceira coluna da tabela fornece os resultados para o método proposto (com $nint = I2 = 20$, versão V3), e

as últimas duas colunas fornecem o resultado do benchmark para dois valores do parâmetro discreto ($L = 10$ e $L = 20$).

Tabela 6: Resultados para o modelo estatístico

N	Métricas	Proposto	L=10	L=20
1	Média do erro (p.p.)	0.198	0.571	0.245
2	Desvio P. do erro (p.p.)	0.337	0.998	0.427
3	Percentil 95% (p.p.)	0.930	2.804	1.168
4	Tempo (segundos)	27.14	109.43	231.75

A Tabela 6 mostra que o método proposto obteve melhores resultados em todas as métricas quando comparado com os resultados de $L = 10$. Quanto ao tempo computacional, o método proposto foi 4 vezes mais rápido, e quanto ao percentil de 95% do erro, o método proposto foi 3 vezes melhor. Quando comparado com $L = 20$, o método proposto apresenta uma precisão um pouco melhor, sendo quase 9 vezes mais rápido que o *benchmark*.

Em nossa implementação do benchmark, uma observação é que, ao aplicar o método para resolver um PRV, os passos 3 e 10 do tópico 4.1.4 (gerar valores para as variáveis tempo de viagem e tempo de serviço) podem ser realizados uma única vez no início do algoritmo, de forma que isto foi feito nestes experimentos para não penalizar o tempo computacional do benchmark.

Com o objetivo de ilustrar o comportamento do erro do método proposto, a frequência relativa assim como a frequência cumulativa do erro é mostrada nas figuras 26 e 27.

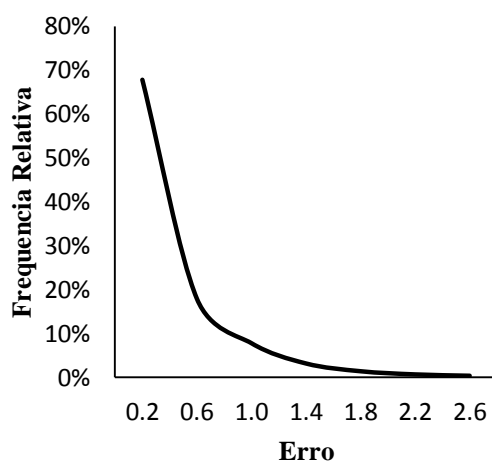


Figura 26: Frequência relativa do erro

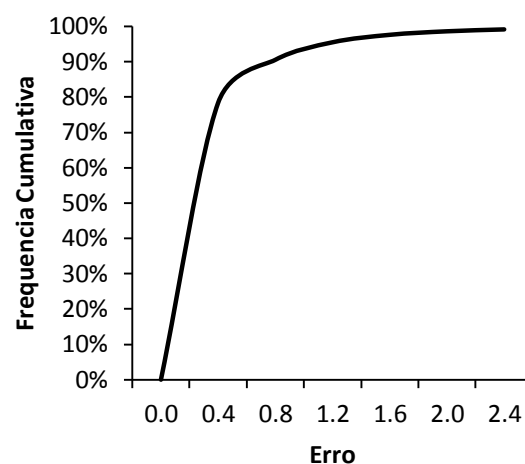


Figura 27: Distribuição acumulada do erro

A Figura 26 mostra que a distribuição de frequência possui uma cauda longa no lado esquerdo, relativamente próxima de uma distribuição exponencial. O fato da distribuição estar deslocada para esquerda indica que a maior parte dos valores do erro estão mais próximos de zero, coerente com o percentil mostrado na Tabela 6, indicando que aproximadamente 95% dos erros são inferiores a 0.93. Isto também é visto na Figura 27, que fornece uma visualização mais fácil de outros percentis. Por exemplo, existe aproximadamente 90% de probabilidade de que o valor do erro seja inferior a 1%.

Tabela 7: Estatística descritiva

Média	0.198
Erro Padrão	0.001
Mediana	0.003
Moda	0.000
Desvio Padrão	0.337
Variância da amostra	0.114
Kurtosis	5.734
Skewness	2.243
Mínimo	0.000
Máximo	3.146
Contagem	203402

A Tabela 7 fornece estatísticas descritivas para o erro. Observa-se que o “skewness” é 2.2 e “kurtosis” 5.7, respectivamente pelo fato da distribuição de frequência ser deslocada para esquerda, e ter um pico acentuado. Para o tamanho da amostra (203402), o erro padrão é bastante baixo (0.001). Também se observa que o maior erro observado teve valor 3.1.

7.2 – RESULTADOS PARA O PRVJT ESTOCÁSTICO

Diferentes experimentos foram feitos para estudar o comportamento da metaheurística desenvolvida neste trabalho. Eles são listados a seguir e melhor explicados adiante.

- 1) Efeito da estratégia de “zona circular”.
- 2) Efeito da estratégia de “rotas e movimentos tabu”.
- 3) Efeito de diferentes critérios de saída.
- 4) Efeito de diferentes níveis de serviço especificados para a restrição do nível de serviço.
- 5) Resultados para as 56 instâncias de Solomon (1987).
- 6) Implementa-se um algoritmo de decisão multicritério para discutir a influência da fronteira pareto na tomada de decisão.

7.2.1 – Zona circular

Como explicado no capítulo 6, a estratégia da zona circular tem como objetivo melhorar o tempo de execução das buscas locais reduzindo-se o número de movimentos possíveis na estrutura de vizinhança. Estudou-se um critério para ser utilizado como a distância máxima (no contexto do problema, o tempo médio de viagem máximo) entre dois clientes consecutivos envolvidos no movimento de uma dada busca local, aqui chamada de “distância granular”.

A forma para se calcular a “distância granular” é baseada no percentil da matriz de tempo médio de viagem entre os clientes. Considerando todos os tempos de viagem entre todos os arcos da matriz $N \times N$ (onde N é o número de clientes), estudou-se a influência de se utilizar como “distância granular” os valores de percentis de 20, 30, 40, 50 e 60%.

O gráfico 28 mostra como estes diferentes percentis influenciam o tempo computacional e o hipervolume das soluções obtidas pelo algoritmo de roteamento. O resultado obtido para percentil de 60% é utilizado como referência para os demais valores, ou seja: $\text{hipervolume}\% = \text{hipervol. (percentil } x\%) / \text{hipervol. (percentil 60\%)}$ e $\text{tempo}\% = \text{tempo(percentil } x\%) / \text{tempo(percentil 60\%)}$. Os resultados foram calculados a partir da média do hipervolume e tempo de execução obtido a partir de cinco

rodadas de seis instâncias de Solomon, a primeira de cada uma das seis classes de problemas: C101, C201, R101, R201, RC101 e RC201.

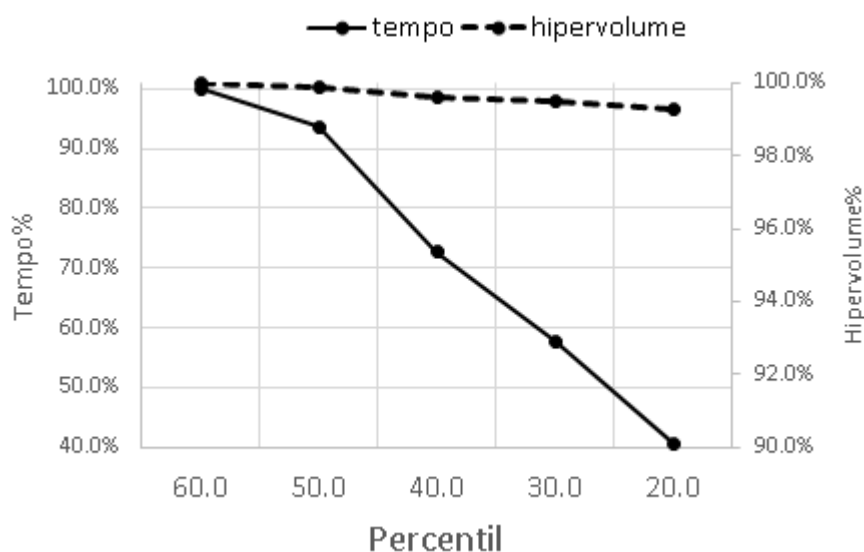


Figura 28: Influência da zona circular

Pelo gráfico da Figura 28, fica claro que quanto menor o percentil utilizado (ou seja, menor distância granular), menor é o tempo computacional do algoritmo, e menor é o hipervolume da fronteira pareto obtido. Por exemplo, o tempo computacional para percentil de 20% é apenas 0.4 vezes o tempo para percentil de 60%, enquanto que o volume é 0.993 vezes o volume obtido usando-se percentil de 60%. É relevante observar que a redução no tempo computacional é muito maior que a redução do hipervolume. Ou seja, conforme o contexto do problema que se deseja resolver, se for admissível uma perda pequena na qualidade da solução final (aqui dada pelo hipervolume), é possível reduzir muito o tempo de execução do algoritmo com a adoção da estratégia de zona circular.

7.2.2 – Rotas e movimentos tabus

Como explicado no capítulo 6, a estratégia que utiliza a classificação de certas rotas e movimentos como tabu também tem como objetivo melhorar o tempo de execução. O maior desafio aqui é encontrar uma forma de indexar as rotas e movimentos de forma a não permitir que um significativo número de rotas seja indevidamente classificado como tabu, proibindo-se movimentos desnecessariamente, e conseqüentemente, afetando o hipervolume da solução obtida.

A Tabela 8 mostra os resultados obtidos para três casos: sem tabu, apenas com utilização de rotas tabus, e com a utilização de ambos (rotas e movimentos) tabus, para as mesmas instâncias do tópico anterior.

Caso	Média	Desvio Padrão
ambos	10898.3	1368.0
sem	10901.6	1365.8
rotas	10905.0	1365.0

A Tabela 8 mostra que a adoção da estratégia não teve influência significativa na média do hipervolume. A comparação das três médias por Análise de Variância (ANOVA) teve $p - value \approx 1$, evidenciando não haver diferença estatística entre as médias, para um nível de confiança de 95%.

A Figura 29 mostra como a estratégia influencia o tempo computacional das soluções obtidas pelo algoritmo de roteamento, para cada uma das seis instâncias testadas. O resultado sem a utilização da estratégia tabu é utilizado como referência para os dois casos, ou seja: $tempo\% = tempo(\text{caso tabu})/tempo(\text{sem tabu})$. Também é mostrado o número médio de veículos obtido nas soluções.

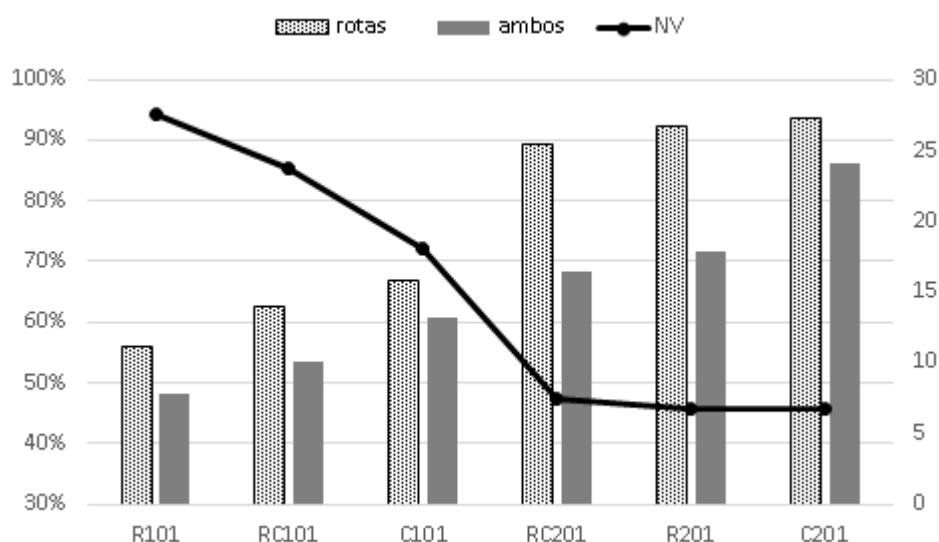


Figura 29: Influência da estratégia de rotas e movimentos tabus

Na Figura 29 observa-se que houve uma significativa redução do tempo computacional para todas as instâncias, nos dois casos: apenas com rotas tabu e

rotas/movimentos tabu. Também pode ser visto a eficácia de se adotar rotas/movimentos tabu, pois isto trouxe uma redução adicional no tempo de execução do algoritmo. Por exemplo, para a instância C101, houve uma redução de pouco mais de 30% no tempo adotando-se apenas rotas tabus, e com a adoção também dos movimentos tabus, esta redução aumentou para 40%, ou seja, o tempo com a estratégia foi cerca de 60% do tempo sem a estratégia tabu.

Observa-se também na Figura 29 que a redução do tempo de execução de diferentes instâncias parece ter uma relação com o número médio de veículos da solução. Quanto maior o número de veículos, maior é a economia de tempo obtida com o uso da estratégia. Entende-se que para uma instância com poucos veículos, as rotas possam mais clientes, sendo assim maiores as chances de que movimentos envolvendo estas rotas resultem numa solução não dominada, reduzindo a quantidade de rotas declaradas tabu, e consequentemente reduzindo o efeito da estratégia.

7.2.3 – Critérios de saída

A Figura 30 mostra a convergência do hipervolume ao longo das gerações, considerando a média do hipervolume para o mesmo conjunto de seis instâncias usados no tópico anterior.

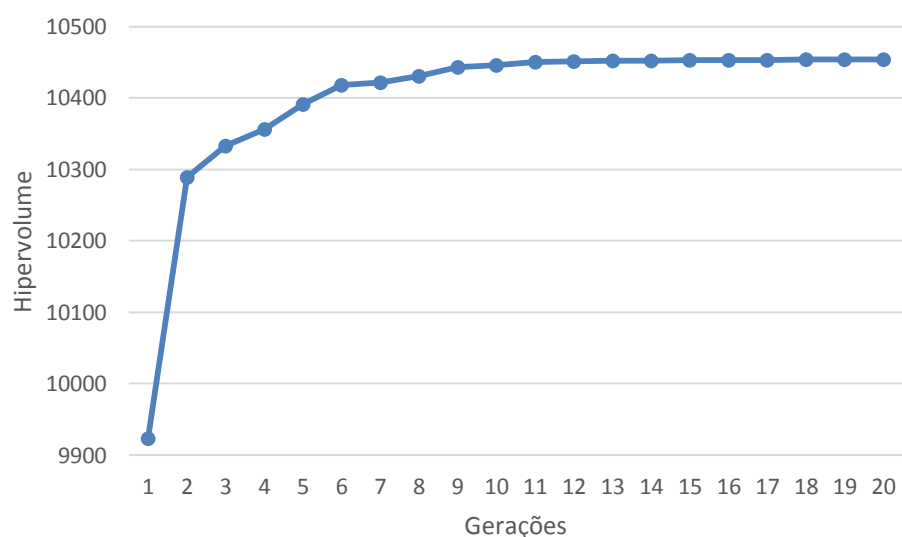


Figura 30: Convergência do hipervolume

Pela Figura 30 observa-se que o hipervolume começa a estabilizar na geração 10, indicando que este pode ser um critério de parada do algoritmo. O critério de parada

não precisa ser um número fixo de gerações mas pode também considerar a taxa de aumento do hipervolume.

Na Tabela 9 considerou-se dois cálculos desta taxa de convergência: uma dada pela soma percentual do incremento do hipervolume nas últimas três gerações (“soma3”) e outra considerando as últimas cinco gerações (“soma5”). Se esta soma for menor que 0.5%, então o algoritmo é terminado. Ou seja, entende-se que se a melhora percentual do hipervolume nas últimas gerações não for significativa (aqui 0.5%), não vale a pena continuar com as iterações. Uma motivação para considerar-se mais de uma geração para calcular a convergência vem da observação de que em não raras situações, o algoritmo tem uma melhora muito pequena ou zero numa geração e então apresenta uma melhora significativa na geração seguinte.

A melhora percentual (pm_g) numa dada geração g é calculada em relação a geração anterior $g - 1$, e os dois critérios alternativos de saída são dados pelas equações a seguir:

$$pm_g = [(hipervolume_g - hipervolume_{g-1})/hipervolume_{g-1}] * 100 \quad (81)$$

$$\sum_{i=g-2}^g pm_i < 0.5 \quad (82)$$

$$\sum_{i=g-4}^g pm_i < 0.5 \quad (83)$$

A Tabela 9 mostra os resultados para estes três critérios de saída: 10 gerações fixas (“fixo10”), critério “sum3” da equação (82) e critério “sum5” da equação (83). Os valores percentuais das colunas três e quatro consideram os resultados de “fixo10” como referência.

Tabela 9: Efeito dos critérios de saída

Critério	Hipervolume	Segundos	%Volume	%Tempo
fixo10	10467	201	100.0%	100.0%
soma5	10455	169	99.9%	83.9%
soma3	10436	124	99.7%	61.8%

Observa-se na Tabela 9 que o critério “soma5” possui tempo de execução aproximadamente 16% inferior que “fixo10” com uma perda de volume muito baixa (0.1%). Já o critério “soma3”, é mais rápido cerca de 38% que “fixo10” e apresenta uma redução mais significativa do volume (0.3%). Ou seja, a decisão sobre qual critério de para utilizar vem do contexto em que o algoritmo é utilizado. Se uma pequena perda de volume é admissível, o critério “soma3” pode ser o mais adequado.

7.2.4 – Nível de serviço requerido

A Figura 31 mostra a influência de diferentes valores para o nível de serviço requerido no resultado do algoritmo quanto ao número médio de veículos e média do tempo total de viagem para as mesmas seis instâncias do tópico anterior.

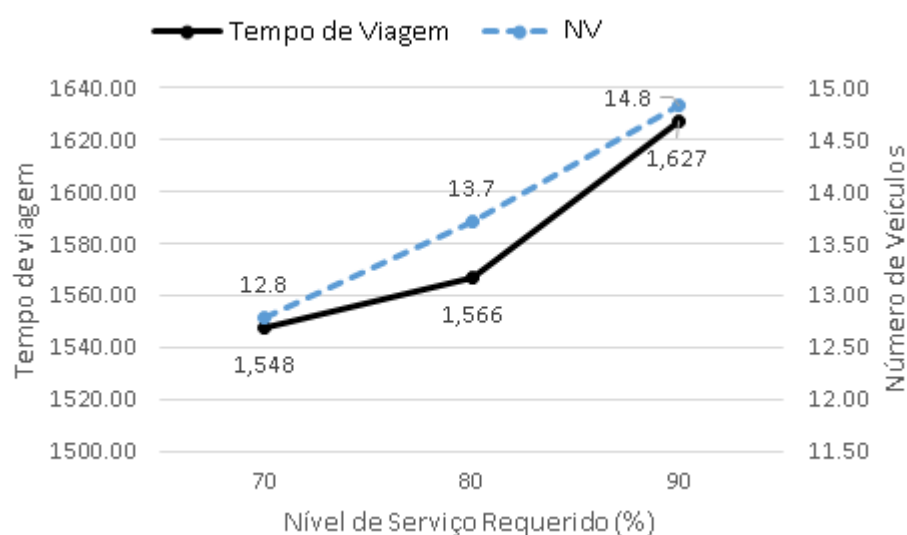


Figura 31: Influência do nível de serviço

A Figura 31 mostra o quanto o número de veículos e o tempo de viagem aumenta com o aumento do nível de serviço requerido. De 70% para 90%, o número de veículos aumentou em cerca de 16%, enquanto o tempo de viagem aumentou aproximadamente 5%.

7.2.5 – Resultados para as 56 instâncias de Solomon

Realizou-se 10 rodadas para cada uma das 56 instâncias de Solomon (1987), utilizando-se os parâmetros a seguir. Critério de parada: tipo “soma3”, ou seja, se a soma da melhora do hipervolume no total das últimas 3 gerações for inferior a 0.5%, terminar algoritmo. “Distância granular”: percentil de 30%. Soluções iniciais: $nIniSol = 15$,

sendo todas viáveis. Filhos gerados no cruzamento: $nFilhos = 10$. Rotas/movimentos tabus ativados. Critério de cut-off da população (linha 15 do algoritmo 3): manter na população apenas soluções com rank 1 e rank 2. Critério de saída na fase de pré-intensificação (linha 5 do algoritmo 5): $intGenMax = 1$. Critério de saída na fase de intensificação (linha 6 do algoritmo 6): $VolMin = 0.5\%$ para cada iteração do loop das linhas 6 a 14. Coeficiente de penalidade de violação do nível de serviço: $PenF = 1000$ (citado no algoritmo 8). Nível de serviço requerido para os clientes: $\alpha_i = 70\%$ para todos os clientes $i = 1, \dots, 100$. Para a função objetivo de custos de transporte, o custo fixo do veículo teve valor $f = 1000$, e o custo de cada unidade de $E[TT_{ij}]$ tem valor $c = 1$.

Considerando que não se encontrou qualquer outro trabalho na literatura para resolver o problema abordado nesta tese, também não foi possível encontrar um benchmark. Para se ter uma ideia da qualidade das soluções, decidiu-se realizar um experimento estendido, de longa duração, para comparar com o algoritmo proposto, em condições normais. Para o experimento estendido (sendo neste contexto o benchmark) utilizou-se uma configuração mais agressiva para os parâmetros que demonstraram ter influência no hipervolume das soluções: 40 gerações como critério de término e uso de percentil de 60% para a “distância granular”. Como benchmark, considerou-se o melhor resultado encontrado dentre 10 rodadas realizadas (coluna “Benchmark” da Tabela 10).

Tabela 10: Resultados para as 56 instâncias em termos do hipervolume

Instância	Benchmark	Pior	Média	Melhor	Pior Gap%	Média Gap%	Melhor Gap%
C101	10492.0	10196.0	10279.6	10379.0	2.82%	2.02%	1.08%
C102	10336.0	10269.0	10317.0	10319.0	0.65%	0.18%	0.16%
C103	10481.0	10392.0	10453.6	10457.0	0.85%	0.26%	0.23%
C104	10505.0	10213.0	10275.2	10296.0	2.78%	2.19%	1.99%
C105	10259.0	10178.0	10232.8	10243.0	0.79%	0.26%	0.16%
C106	10410.0	10389.0	10397.6	10399.0	0.20%	0.12%	0.11%
C107	10533.0	10415.0	10463.0	10485.0	1.12%	0.66%	0.46%
C108	10100.0	10054.0	10079.6	10084.0	0.46%	0.20%	0.16%
C109	10131.0	10117.0	10124.2	10128.0	0.14%	0.07%	0.03%
C201	12286.0	12121.0	12224.0	12248.0	1.34%	0.50%	0.31%
C202	10306.0	10296.0	10301.6	10302.0	0.10%	0.04%	0.04%
C203	10769.0	10730.0	10744.2	10752.0	0.36%	0.23%	0.16%
C204	9739.0	9727.0	9731.6	9739.0	0.12%	0.08%	0.00%
C205	11259.0	11211.0	11244.0	11255.0	0.43%	0.13%	0.04%
C206	9870.0	9854.0	9863.4	9870.0	0.16%	0.07%	0.00%
C207	9816.0	9800.0	9810.6	9812.0	0.16%	0.06%	0.04%
C208	8274.0	8274.0	8274.0	8274.0	0.00%	0.00%	0.00%
R101	10072.0	9864.0	9937.2	10068.0	2.07%	1.34%	0.04%
R102	10121.0	10078.0	10102.6	10118.0	0.42%	0.18%	0.03%
R103	10374.0	10348.0	10364.4	10365.0	0.25%	0.09%	0.09%
R104	10111.0	10084.0	10095.4	10111.0	0.27%	0.15%	0.00%
R105	10144.0	10124.0	10133.0	10133.0	0.20%	0.11%	0.11%
R106	10053.0	10041.0	10049.2	10052.0	0.12%	0.04%	0.01%
R107	10133.0	10115.0	10125.0	10129.0	0.18%	0.08%	0.04%
R108	10115.0	10108.0	10111.0	10113.0	0.07%	0.04%	0.02%
R109	10110.0	10100.0	10105.2	10106.0	0.10%	0.05%	0.04%
R110	10156.0	10149.0	10151.8	10155.0	0.07%	0.04%	0.01%
R111	10280.0	10064.0	10150.6	10171.0	2.10%	1.26%	1.06%
R112	10117.0	10113.0	10115.2	10117.0	0.04%	0.02%	0.00%
R201	11225.0	11161.0	11195.0	11204.0	0.57%	0.27%	0.19%
R202	10161.0	10132.0	10149.8	10155.0	0.29%	0.11%	0.06%
R203	10360.0	10305.0	10329.2	10353.0	0.53%	0.30%	0.07%
R204	10093.0	10064.0	10079.6	10092.0	0.29%	0.13%	0.01%
R205	10105.0	10085.0	10097.6	10101.0	0.20%	0.07%	0.04%
R206	10121.0	10109.0	10113.6	10119.0	0.12%	0.07%	0.02%
R207	10087.0	10064.0	10074.0	10082.0	0.23%	0.13%	0.05%
R208	10509.0	10383.0	10466.0	10503.0	1.20%	0.41%	0.06%
R209	12338.0	12166.0	12327.4	12333.0	1.39%	0.09%	0.04%
R210	10494.0	10106.0	10260.8	10280.0	3.70%	2.22%	2.04%
R211	10114.0	10102.0	10105.8	10113.0	0.12%	0.08%	0.01%
RC101	10220.0	10099.0	10175.6	10182.0	1.18%	0.43%	0.37%
RC102	10172.0	9925.0	9982.4	10006.0	2.43%	1.86%	1.63%
RC103	10154.0	10092.0	10136.6	10147.0	0.61%	0.17%	0.07%
RC104	10201.0	10193.0	10198.6	10201.0	0.08%	0.02%	0.00%
RC105	10357.0	10322.0	10339.4	10340.0	0.34%	0.17%	0.16%
RC106	10159.0	10149.0	10156.4	10158.0	0.10%	0.03%	0.01%
RC107	10035.0	10030.0	10032.6	10035.0	0.05%	0.02%	0.00%
RC108	10154.0	10085.0	10135.2	10146.0	0.68%	0.19%	0.08%
RC201	10202.0	10146.0	10175.4	10198.0	0.55%	0.26%	0.04%
RC202	10145.0	10127.0	10135.0	10142.0	0.18%	0.10%	0.03%
RC203	10598.0	10171.0	10266.8	10358.0	4.03%	3.13%	2.26%
RC204	11589.0	11500.0	11527.2	11550.0	0.77%	0.53%	0.34%
RC205	10185.0	10164.0	10173.8	10182.0	0.21%	0.11%	0.03%
RC206	10159.0	10122.0	10146.4	10149.0	0.36%	0.12%	0.10%
RC207	10159.0	10145.0	10149.4	10150.0	0.14%	0.09%	0.09%
RC208	10068.0	10057.0	10059.2	10060.0	0.11%	0.09%	0.08%
				Média	0.69%	0.39%	0.25%

Na Tabela 10, para calcular-se os hipervolumes utilizou-se os valores de referência descritos no capítulo de anexos. A coluna “Pior” possui o pior resultado do algoritmo dentre as 10 rodadas nas condições normais descritas anteriormente, a coluna “Média”, possui o hipervolume médio dentre as 10 rodadas, e a coluna “Melhor”, o melhor hipervolume. Os gaps foram calculados como sendo:

$$(Benchmark - Proposto) / Benchmark. \quad (84)$$

Dentre os gaps médios para o pior, médio e melhor caso, os valores foram respectivamente 0.69%, 0.39% e 0.25%. Para o caso médio, dentre as 56 instâncias, 46 obtiveram um gap inferior a 0.5% e apenas 4 casos com gap superior a 2%. A Figura 32 ajuda a entender o significado destes gaps, mostrando duas fronteiras pareto para a instância C104, comparando a fronteira benchmark (volume 10505) com a fronteira do pior caso (volume 10213), um gap de 2.78% no hipervolume

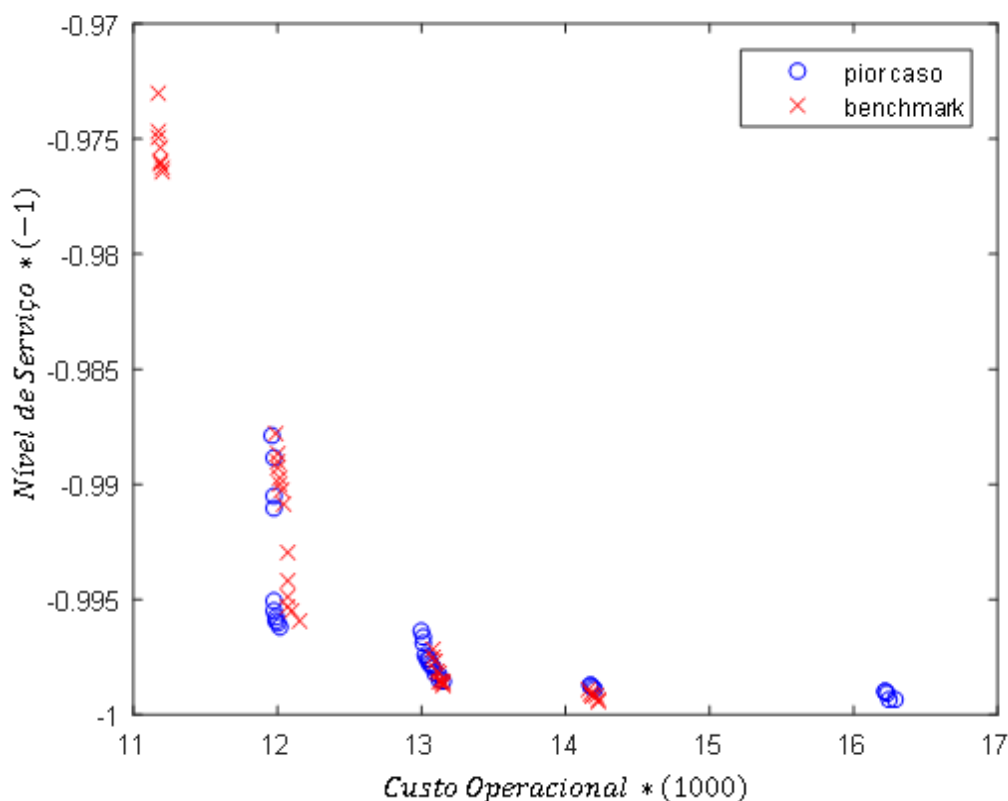


Figura 32: Qualidade das soluções para as classes de instâncias

A Figura 32 mostra que o benchmark forneceu mais soluções com menor custo operacional (com valor em torno de 11200, 10 veículos) e com menor nível de serviço

(em torno de 97.5%). Mesmo com um volume significativamente menor, a fronteira do pior caso ainda encontrou algumas soluções no outro extremo, como um nível de serviço mais próximo de 100% e maior custo operacional (em torno de 16300 e 15 veículos). Pode-se atribuir o volume maior do benchmark principalmente pelo fato de ter oferecido várias soluções não dominadas com custo operacional significativamente menor, enquanto que na outra fronteira, a maior parte das soluções no extremo com maior custo operacional e maior nível de serviço são dominadas em relação ao benchmark.

A Figura 33 resume o gap de hipervolume considerando-se a média para cada uma das classes de problemas. As classes C1 e RC2 apresentaram os maiores gaps. Pela Figura 34, observa-se que estas classes tiveram média menor pela presença de extremos. A classe C1 apresentou duas instâncias com maior gap (C101 e C104), dentre as 9 instâncias da classe. A classe RC2 apresentou apenas uma instância com maior gap (RC203), dentre as 8 instâncias da classe.

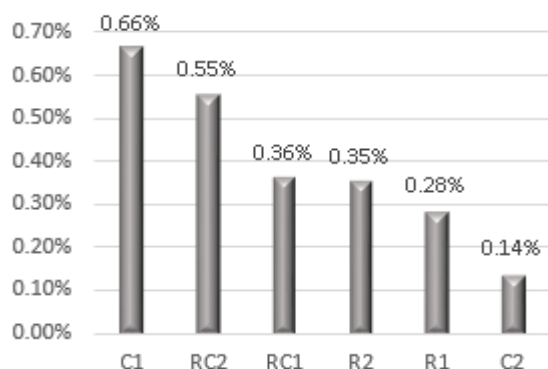


Figura 33: Gap médio por classe

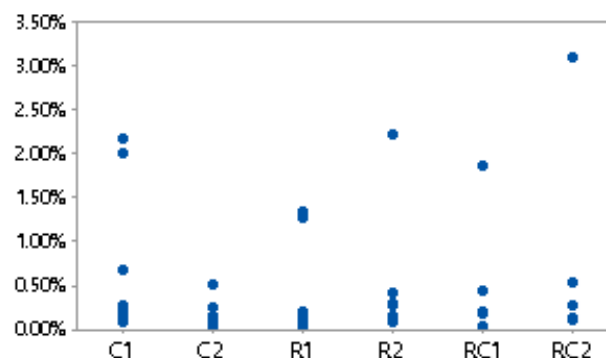


Figura 34: Gap médio por instância

Um dos propósitos de se gerar uma fronteira pareto de soluções é proporcionar ao tomador de decisão um maior número de cenários com o maior intervalo possível para cada um dos objetivos. A Figura 35 mostra o intervalo médio para cada uma das classes de problemas, para cada um dos objetivos (Nível de Serviço e Custo Operacional). A classe C1 foi a que apresentou maiores intervalos para ambos os objetivos, oferecendo soluções com uma diferença de até 5 pontos percentuais de diferença para o nível de serviço e até 4000 unidades monetárias para o custo. As classes R1 e RC1 apresentaram um menor intervalo para o nível de serviço e um intervalo maior para o custo. As classes RC2 e R2 apresentaram um menor intervalo para ambos os objetivos.

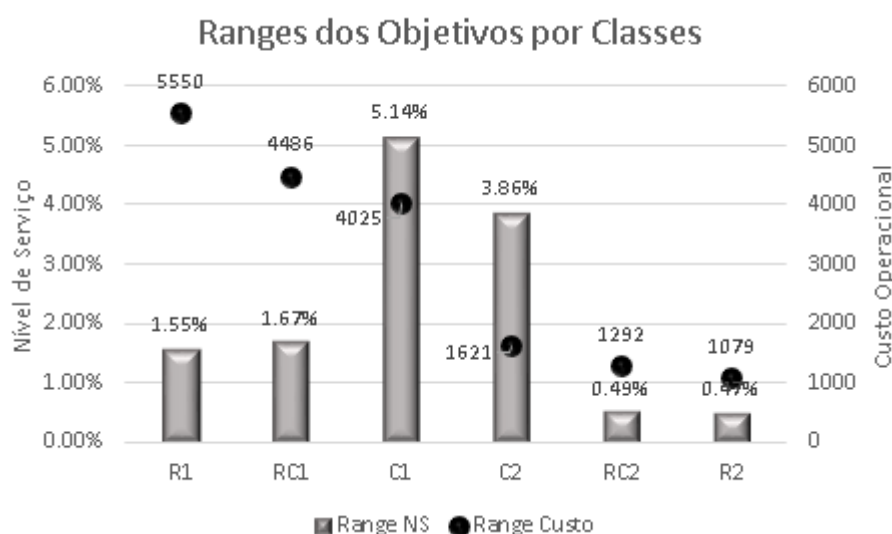


Figura 35: Intervalos dos objetivos para as classes de problemas

A Figura 36 mostra o número médio mínimo e máximo de veículos para cada classe. Observa-se que as instâncias do tipo “2” (C2, RC2 e R2) apresentaram um intervalo menor e poucos veículos. As classes deste tipo possuem janelas de tempo mais largas e maior capacidade dos veículos, ou seja, são instâncias com menor efeito das restrições de janela de tempo.

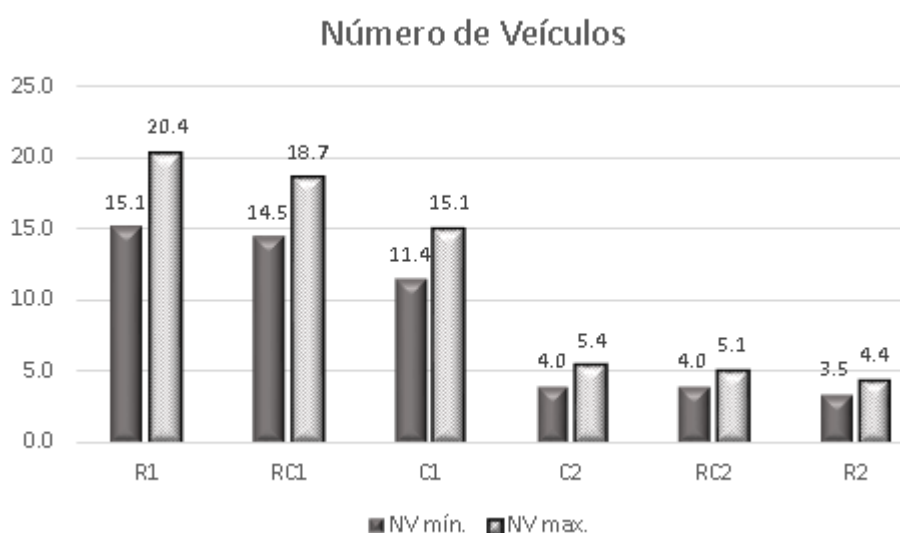


Figura 36: Intervalos do número de veículos para as classes de problemas

A Figura 37 mostra o tempo computacional médio para as diferentes classes. Percebe-se que as classes do tipo “2” possuem um tempo computacional bem maior. Como estas classes possuem janelas de tempo e capacidade dos veículos maiores, a quantidade de movimentos viáveis nas buscas locais é maior, consequentemente realiza-

se um maior número de convoluções para calcular o nível de serviço dos clientes, sendo este um dos motivos para este comportamento. A Figura 38 quantifica a relação entre a quantidade de convoluções (média de convoluções para cada instância) e o tempo de execução para as seis classes.

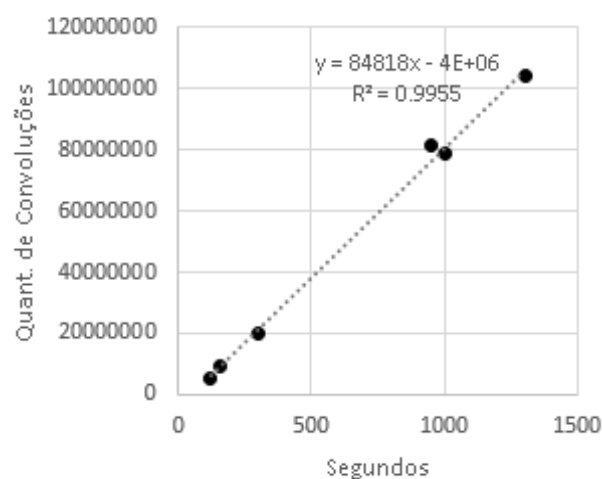
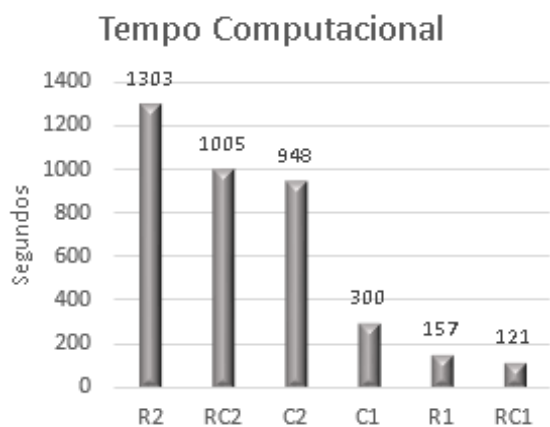


Figura 37: Tempo por classe

Figura 38: Relação entre tempo e convoluções

As figuras 39 a 45 mostram a fronteira pareto das 56 instâncias, obtidas para o melhor caso (em “condições normais” dos parâmetros) citado na Tabela 10, quinta coluna. Os gráficos estão na mesma sequência da Tabela 10. O eixo horizontal é o custo operacional e o eixo vertical refere-se ao nível de serviço que aqui é um valor negativo para que ambos os objetivos sejam do tipo “minimização”.

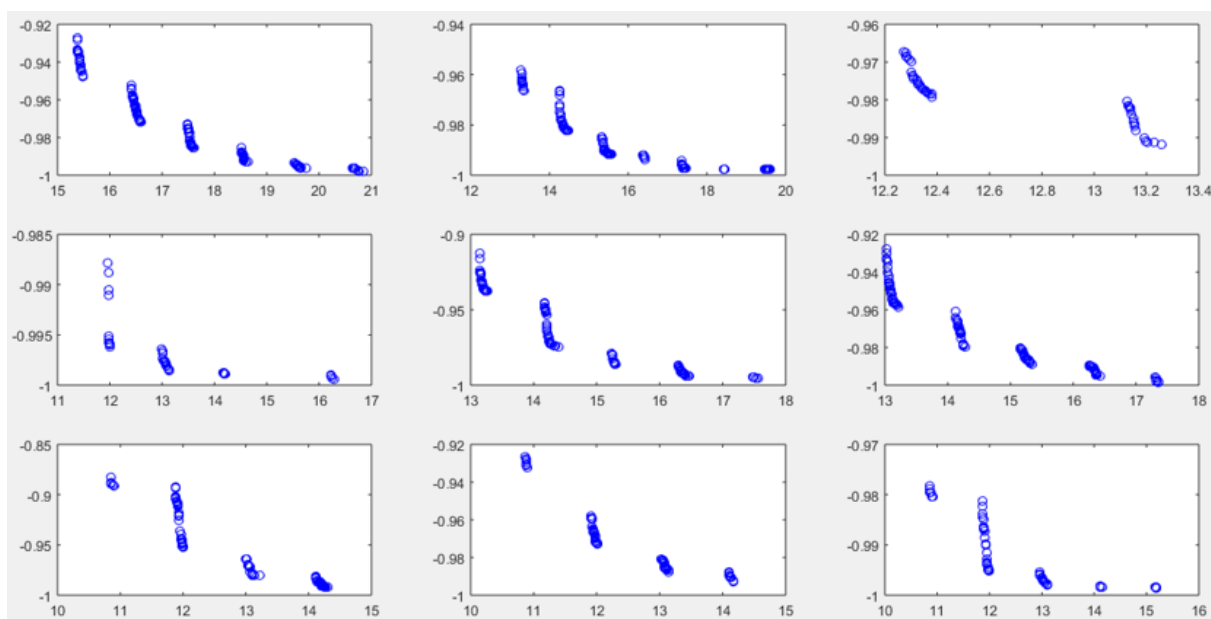


Figura 39: Fronteira Pareto: instâncias 1 a 9

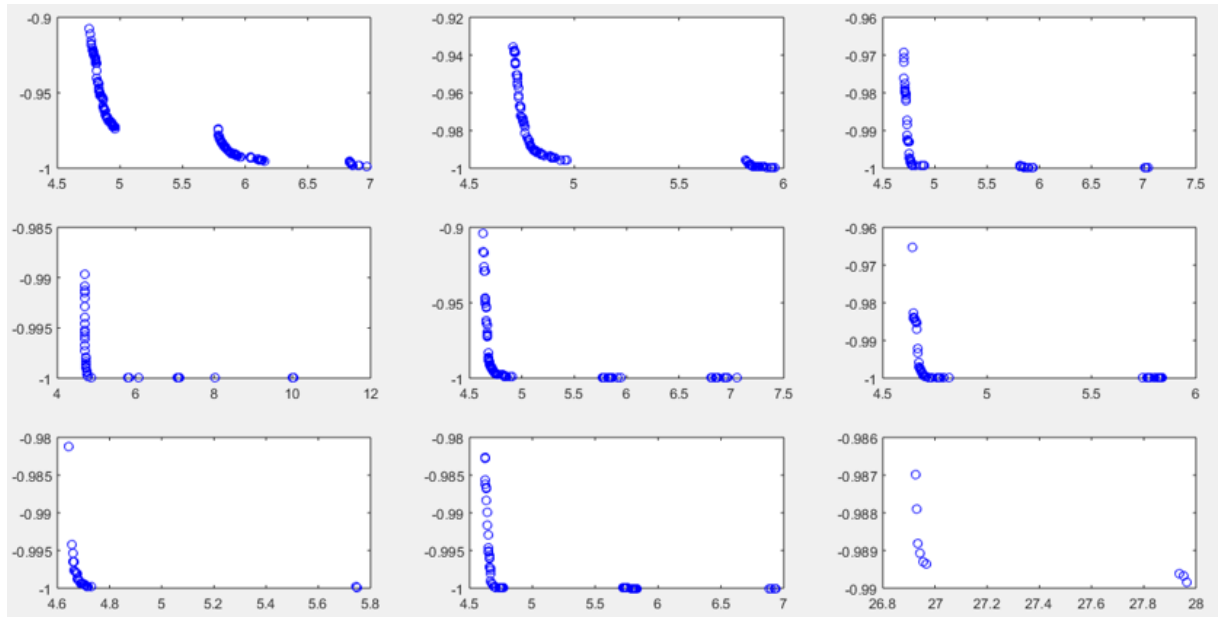


Figura 40: Fronteira Pareto: instâncias 10 a 18

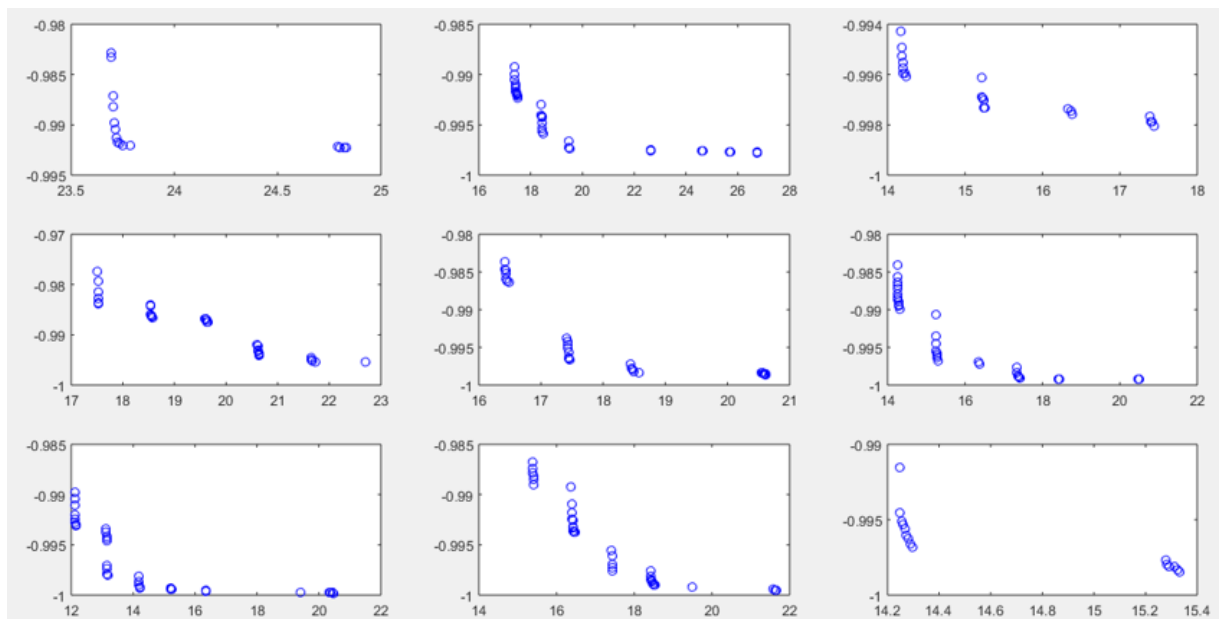


Figura 41: Fronteira Pareto: instâncias 19 a 27

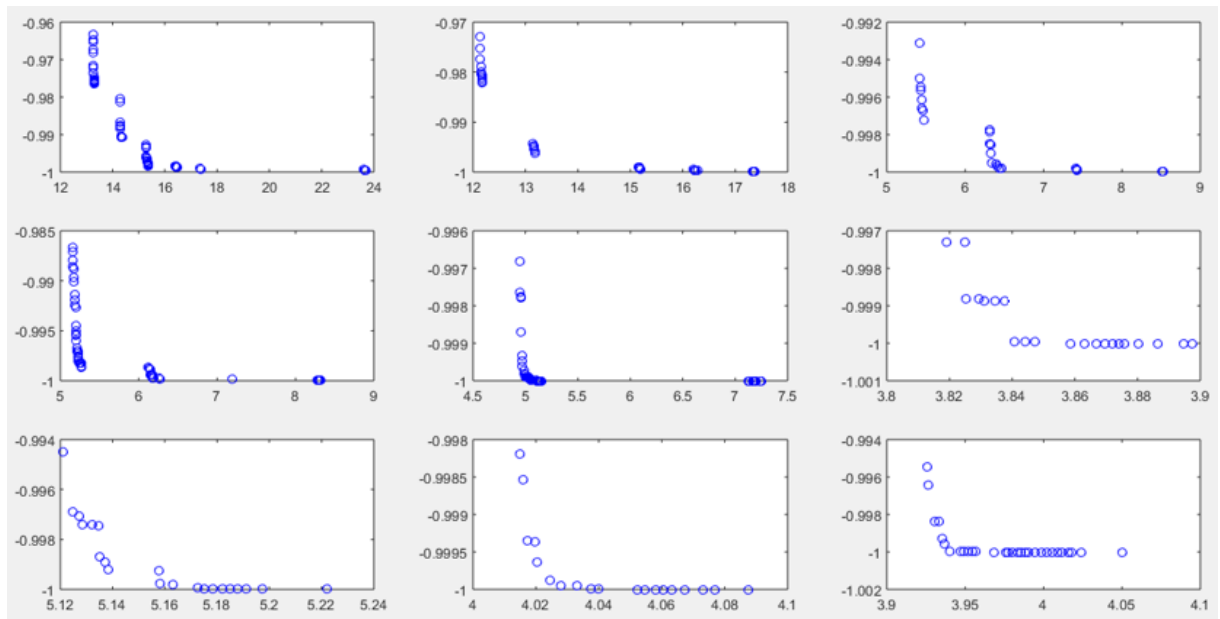


Figura 42: Fronteira Pareto: instâncias 28 a 36

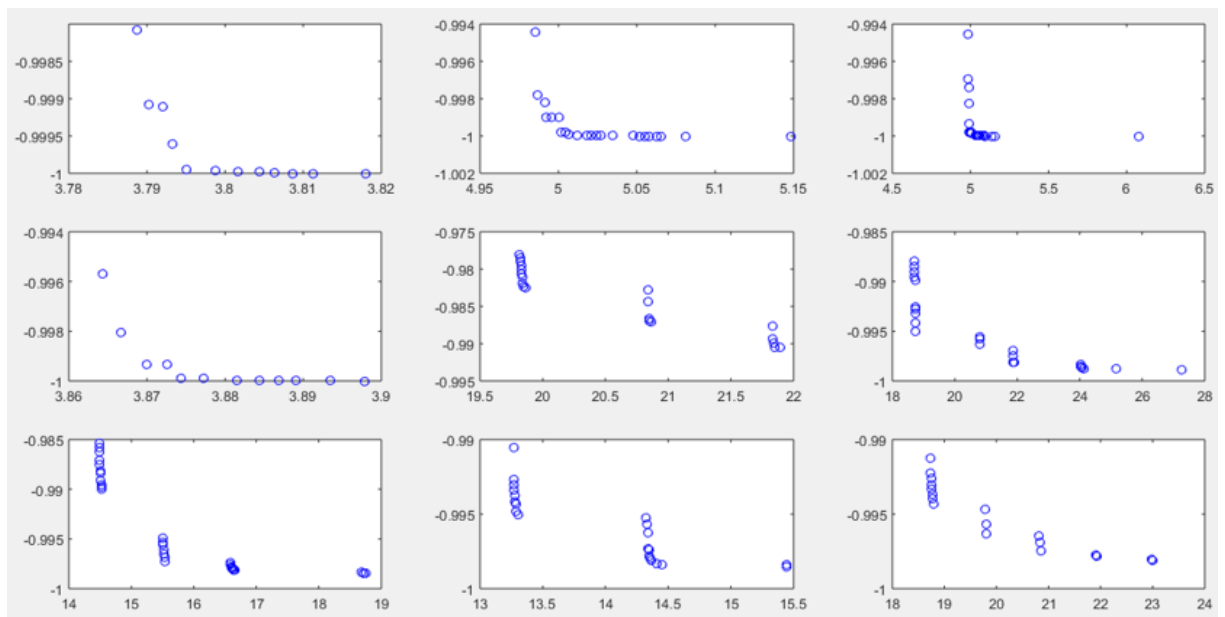


Figura 43: Fronteira Pareto: instâncias 37 a 45

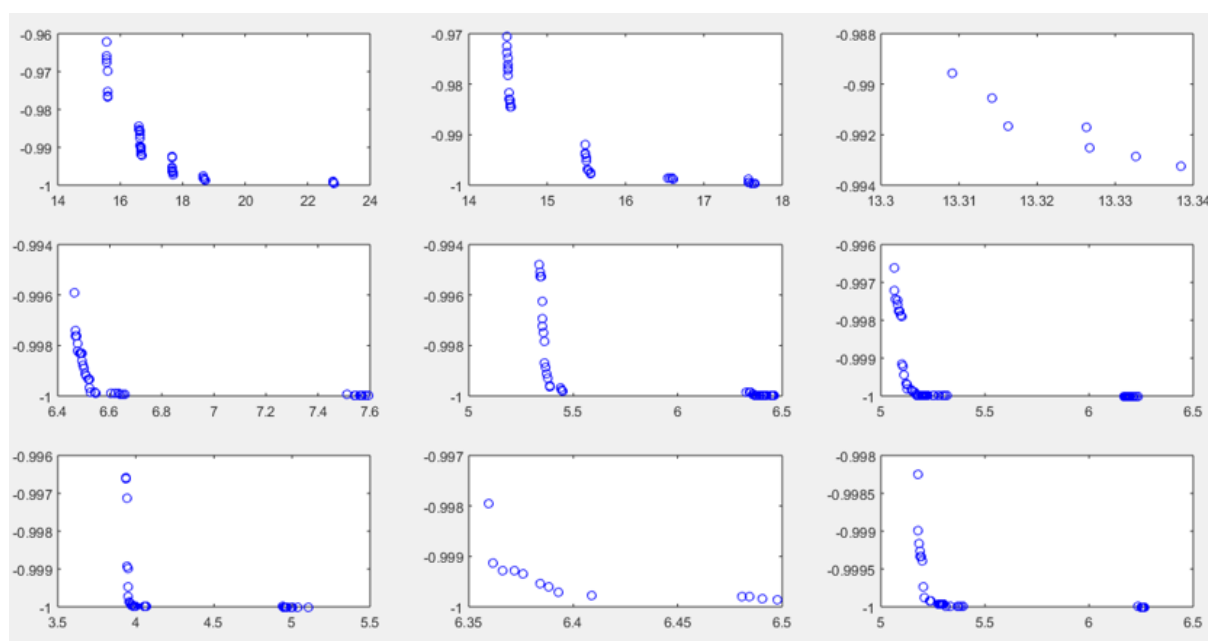


Figura 44: Fronteira Pareto: instâncias 46 a 54

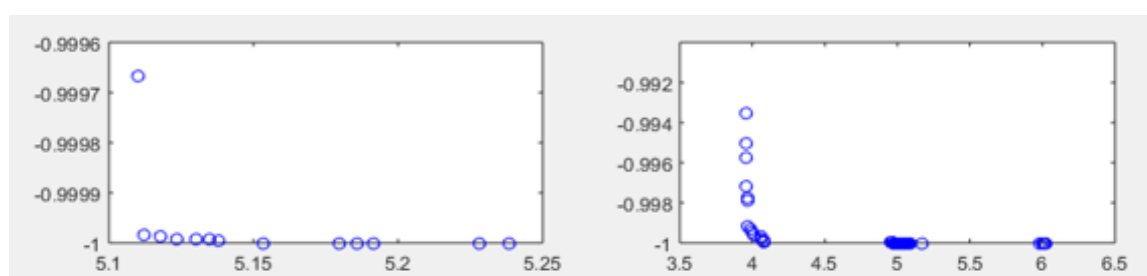


Figura 45: Fronteira Pareto: instâncias 55 a 56

Observa-se uma variedade de formas para a fronteira pareto das soluções. Em geral as fronteiras apresentam uma descontinuidade no objetivo relacionado ao custo operacional, principalmente pelo fato deste custo ser composto pelo custo fixo do veículo. Para um mesmo número de veículos as soluções apresentam uma variação mais suave dada pelo tempo de viagem. No geral, percebe-se que as fronteiras apresentam diversidade, oferecendo soluções com um intervalo significativo de valores para ambos os objetivos.

Algumas fronteiras como a da instância C101 (Figura 39, gráfico da primeira linha e coluna) possui claramente um intervalo maior para ambos os objetivos, oferecendo soluções com diferentes custos/número de veículos, assim como maior diversidade para o nível de serviço. Outras soluções como a da instância R208 (Figura 43, gráfico da primeira linha e coluna) apresenta menor intervalo, oferecendo soluções com custo e nível de serviço muito parecidos. Este é o caso de várias soluções das classes tipo “2”.

7.2.6 – Decisão multi-critério

A otimização multiobjetivo visa oferecer uma diversidade de soluções para o tomador de decisão. Em problemas reais, faz-se necessário escolher apenas uma solução para ser implementada, dentre as várias oferecidas. Uma maneira de ajudar o tomador de decisão nesta escolha é a utilização de métodos de decisão multicritérios conhecidos como MCDM (*Multiple-criteria decision-making*). Estes métodos estruturam o processo de decisão que envolve mais de um critério, auxiliando o tomador de decisão a encontrar uma solução de melhor relação custo-benefício. Não é objetivo deste trabalho aprofundar-se nesta teoria, detalhes podem ser consultados em Zavadskasa, et al. (2014). No entanto, como parte dos experimentos, decidiu-se implementar um método de MCDM para entender melhor o processo de escolha de uma única solução da fronteira pareto obtida pelo algoritmo multiobjetivo proposto pelo presente trabalho.

O método escolhido é o MAVT (*Multi-Attribute Value Theory*) em sua forma mais simples chamada SMART (*Simple Multi-Attribute Rating Technique*), uma das técnicas mais conhecidas de MCDM e cujos detalhes podem ser vistos em Velasquez & Hester (2013).

Para aplicação do método faz-se necessário calcular uma função de utilidade para cada objetivo a fim de se criar uma nota que varia de 0 a 1 para cada objetivo, de cada solução. Quanto maior a nota, maior a atratividade da solução. Quanto ao objetivo relativo ao custo operacional, assume-se neste experimento uma preferência do tomador de decisão de que ele não quer escolher qualquer solução cujo custo operacional seja 50% maior que o da solução de menor custo operacional da fronteira pareto. Esta decisão só foi tomada “a posteriore” quando o tomador de decisão visualizou a fronteira pareto e percebeu que haviam soluções de custo muito superior à solução de menor custo, pelas quais ele não estava disposto a pagar. Assim, a função de utilidade terá valor 1 para o menor valor de custo operacional, e valor 0 para 1.5 vezes este valor.

Quanto ao objetivo associado ao nível de serviço, a função utilidade terá valor 0 para o pior nível de serviço da fronteira pareto e valor 1 para o melhor nível de serviço. Assume-se função utilidade linear e pesos iguais para ambos os objetivos. Dada estas condições, para a instância C101, a Figura 46 mostra a função utilidade para os dois objetivos e a Figura 47 mostra a fronteira pareto obtida pelo algoritmo multiobjetivo e indica a solução escolhida pelo algoritmo MCDM.

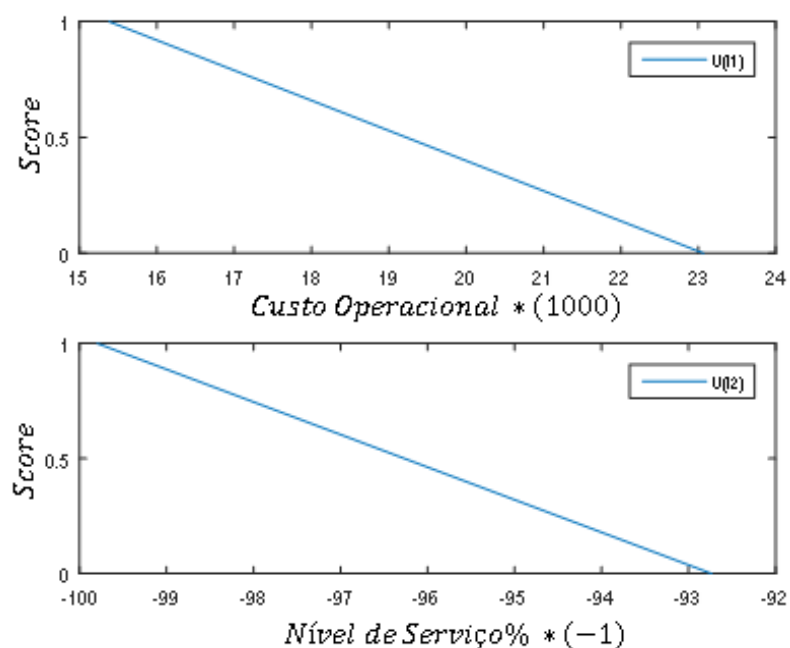


Figura 46: Função utilidade para ambos o objetivos

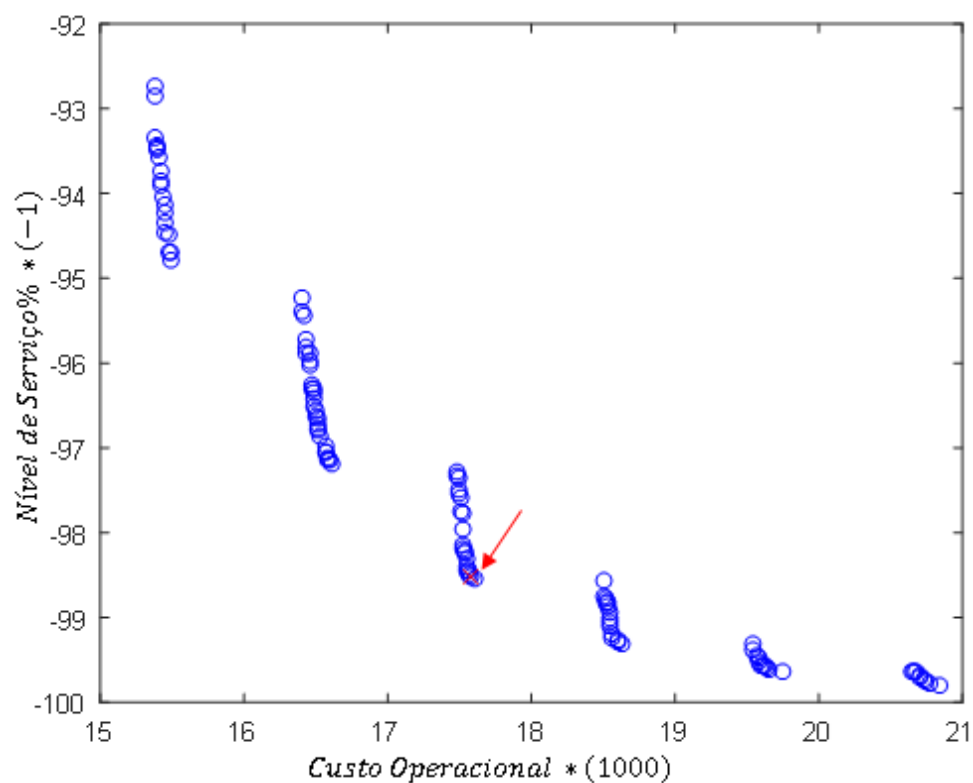


Figura 47: Fronteira Pareto – instância C101

A solução escolhida tem custo operacional \$17582 e nível de serviço de 98.5%. Isto significa que o tomador de decisão está abrindo mão, por exemplo, de uma solução

de custo mais baixo (\$15380) mas com nível de serviço pior (92.7%), um dos extremos da fronteira.

A Figura 48 mostra o mesmo experimento, desta vez indicando cinco diferentes escolhas para diferentes pesos dos objetivos, listados na Tabela 11. Observa-se como a solução escolhida moveu-se de um extremo para o outro extremo da fronteira.

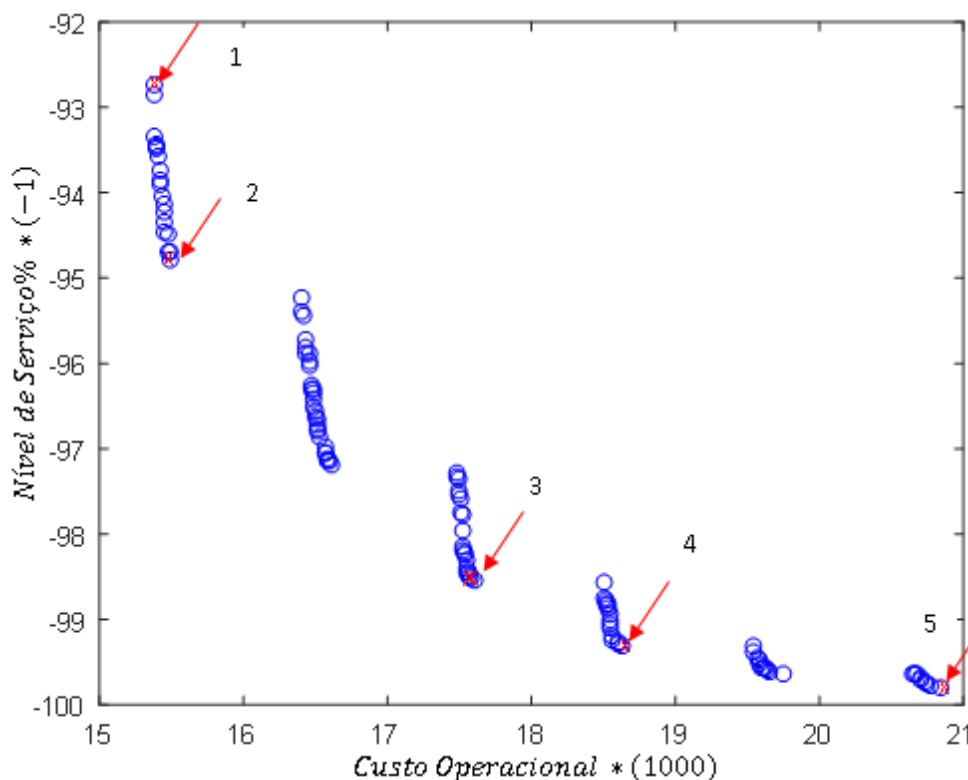


Figura 48: Fronteira Pareto – instância C101

Tabela 11: Escolha para diferentes pesos dos objetivos

Cenário	Peso Custo	Peso NS	Custo x (1000)	NS x (-1)
1	100	0	15.380	-92.7
2	75	25	15.491	-94.8
3	50	50	17.582	-98.5
4	25	75	18.630	-99.3
5	0	100	20.839	-99.8

Outra perspectiva interessante no estudo do processo de decisão é a análise de sensibilidade. A Figura 49 mostra como a solução escolhida muda ao variar-se o peso do custo operacional de 0 a 100. Note que como existem apenas dois objetivos, o peso do nível de serviço é consequência do peso do custo já que a soma de ambos é 100.

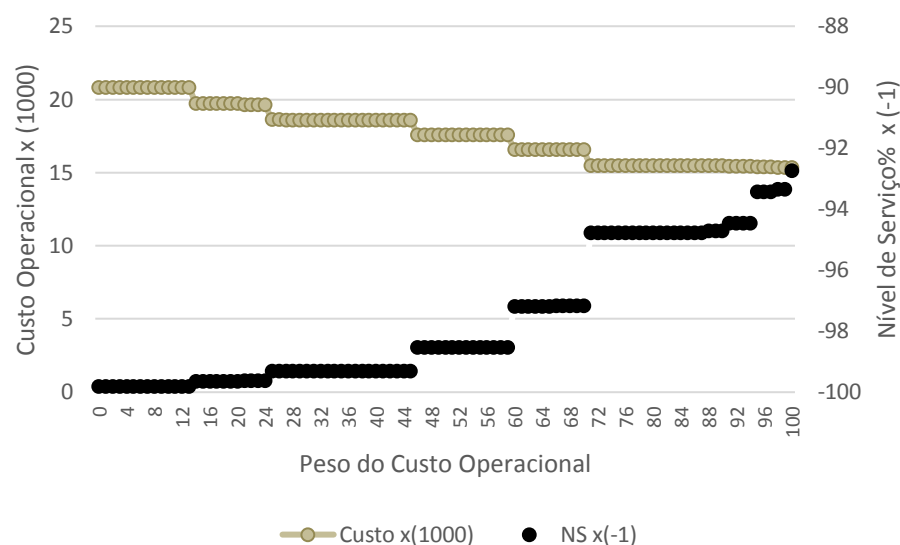


Figura 49: Sensibilidade

Pela Figura 49 é possível observar quais faixas do peso do custo não afetam a escolha da solução na fronteira pareto. Também é possível ver para quais valores de peso em que há uma mudança da solução escolhida, a intensidade desta mudança é maior. Por exemplo, quando o peso muda de 70 para 71 há uma mudança mais abrupta do nível de serviço. A Tabela 12 mostra todos os intervalos para os quais o peso não afeta a solução escolhida. As primeiras duas colunas referem-se ao limite inferior e superior do peso do custo operacional. Pode-se ver que o maior intervalo de peso sem alteração da solução ocorre para peso entre 27 e 45, em que a solução escolhida é a mesma, com custo de \$18604 e nível de serviço de 99.29%.

Tabela 12: Análise de sensibilidade: intervalos

Lim. Inf.	Lim. Sup.	Intervalo	Custo x (1000)	NS x (-1)
0	13	13	20.839	-99.79
14	20	6	19.744	-99.63
21	24	3	19.649	-99.61
25	26	1	18.630	-99.30
27	45	18	18.604	-99.29
46	59	13	17.582	-98.51
60	65	5	16.606	-97.1
66	70	4	16.587	-97.15
71	87	16	15.491	-94.77
88	90	2	15.478	-94.69
91	94	3	15.451	-94.45
95	97	2	15.387	-93.42
98	99	1	15.384	-93.33
100	100	0	15.380	-92.73

As figuras 50 a 51 mostram a solução escolhida para as instâncias C205, R211 e RC207, usando pesos iguais e o mesmo critério usado anteriormente para as funções de utilidade. Estas instâncias foram escolhidas por apresentarem fronteiras pareto com formas e intervalos bem diferentes umas das outras.

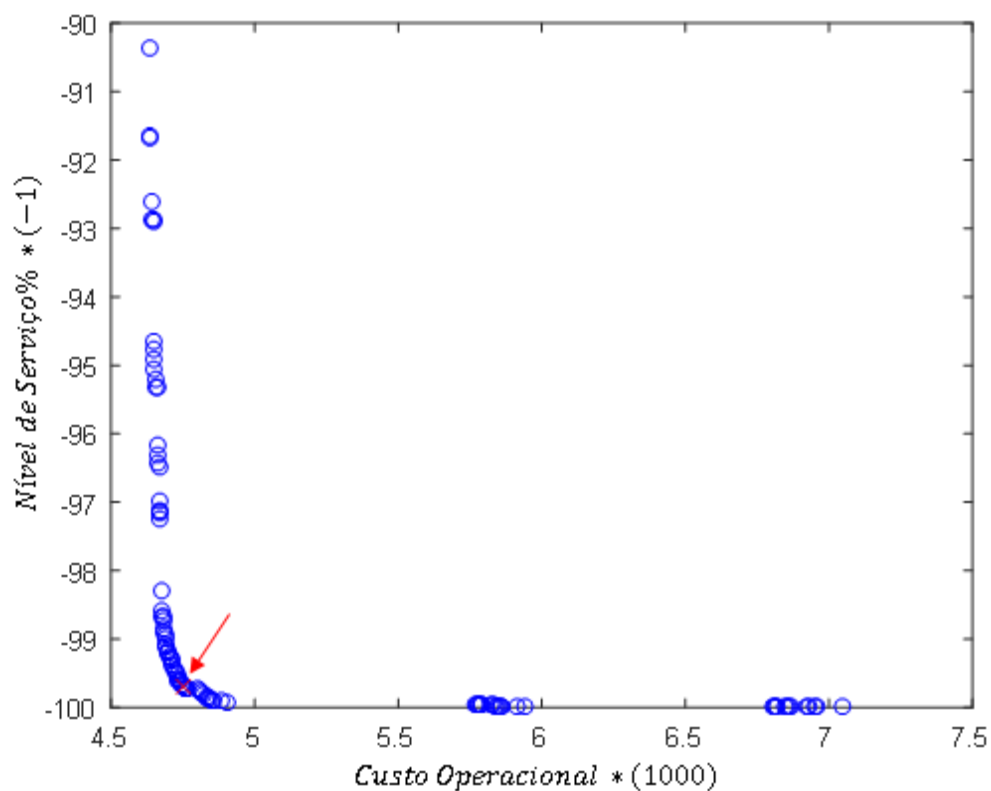


Figura 50: Fronteira Pareto e solução escolhida – instância C205

Para a fronteira da instância C205 (Figura 50), observa-se uma fronteira com um intervalo grande para o nível de serviço e nem tanto para o custo operacional. Observa-se muito mais opções de nível de serviço em particular na faixa de custo entre \$4500 e \$5000. A solução escolhida está nesta faixa, tendo custo de \$4751 e nível de serviço de 99.70%.

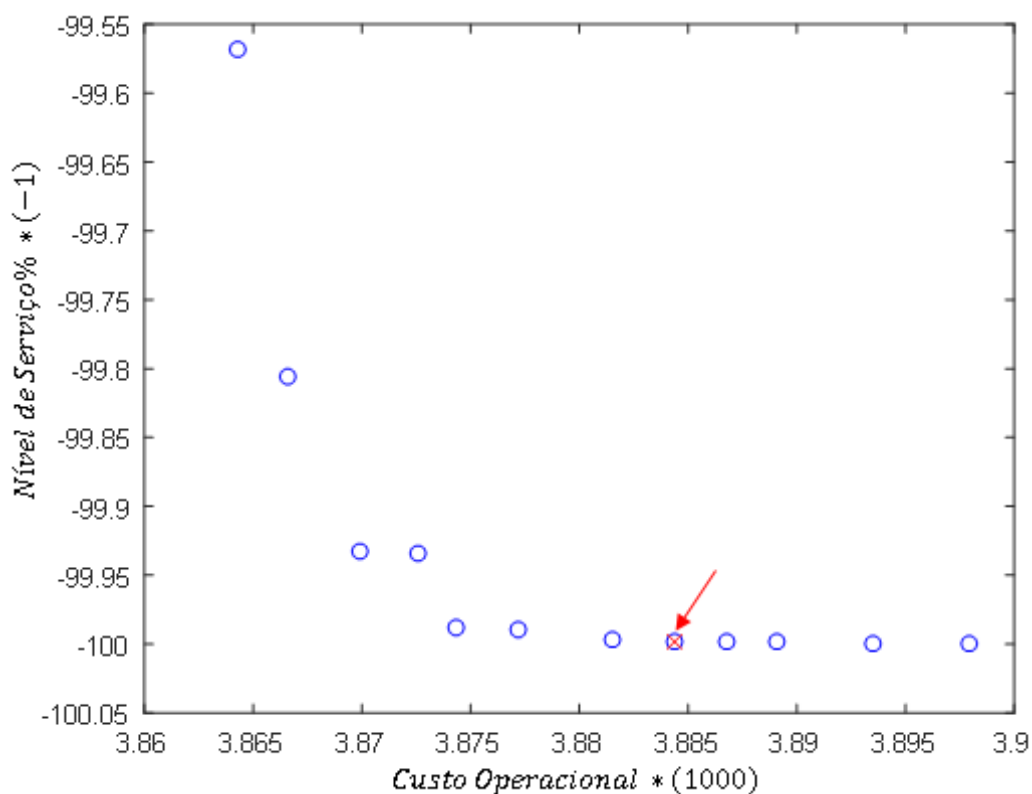


Figura 51: Fronteira Pareto e solução escolhida – instância R211

Para a fronteira da instância R211 (Figura 51), observa-se uma menor quantidade de soluções, assim como um intervalo estreito de valores para ambos os objetivos. Trata-se de uma instância com maior intervalo para as janelas de tempo e maior capacidade da frota, resultando numa solução com menor número de veículos. Todas as soluções apresentam nível de serviço próximo de 100%. A solução escolhida está nesta faixa, tendo custo de \$3884 e nível de serviço de 99.99%.

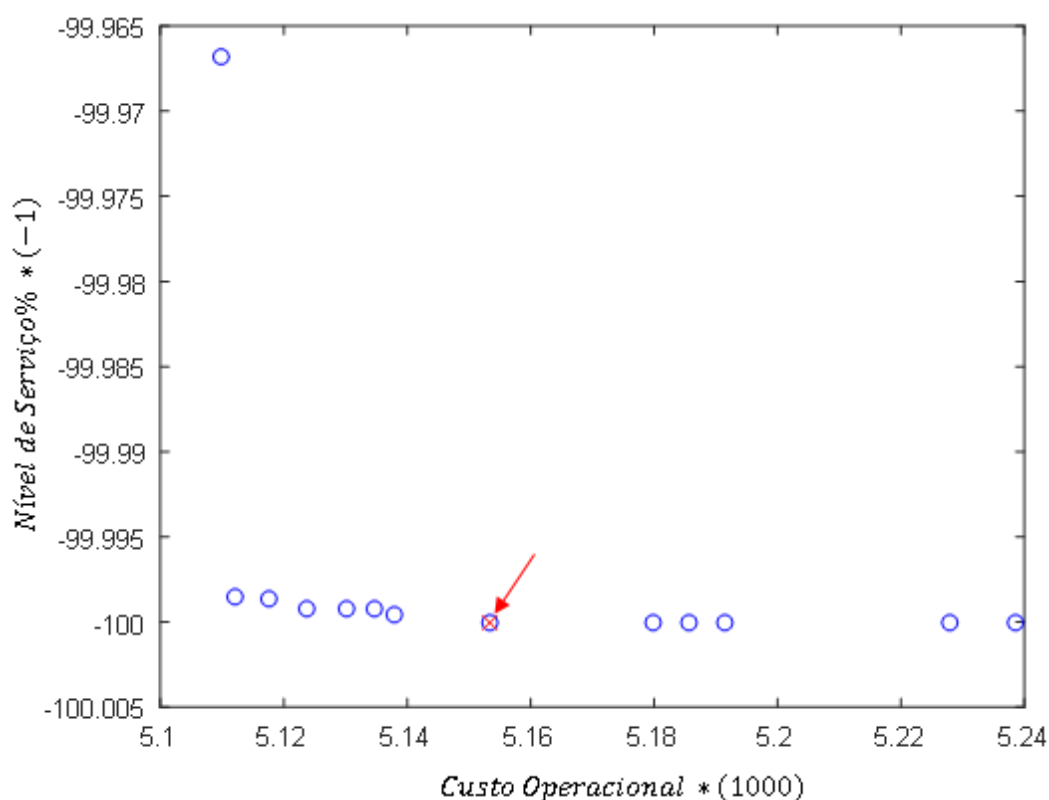


Figura 52: Fronteira Pareto e solução escolhida – instância RC207

Para a fronteira da instância RC207 (Figura 52), também há uma menor diversidade de soluções e um intervalo ainda menor para o nível de serviço, também consequência das características das instâncias de classe tipo “2”. A solução escolhida está nesta faixa, tendo custo de \$5153 e nível de serviço de 100%.

7.3 – DISCUSSÃO DO CAPÍTULO 7

Este capítulo trouxe vários experimentos realizados para testar os dois pontos principais do trabalho: o modelo estatístico para calcular o nível de serviço e o algoritmo memético de otimização usado para resolver o problema multiobjetivo.

O modelo estatístico pode ser comparado com um benchmark da literatura, mostrando resultados satisfatórios. Para o algoritmo multiobjetivo não foi possível encontrar um benchmark (próprio motivo da contribuição deste trabalho), de modo a se realizar um experimento estendido para estimar a qualidade das soluções obtidas, além dos experimentos terem testado de forma específica a eficácia das estratégias utilizadas.

CAPÍTULO 8

CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS

FUTUROS

Este capítulo apresenta as conclusões finais, resume as principais contribuições e identifica oportunidades para futuros trabalhos.

O trabalho faz uma vasta revisão da literatura, partindo do PRV mais básico e aprofundando-se passo a passo nas variantes de interesse do estudo, passando pelo PRVJT, o PRV Estocástico, o PRV Multiobjetivo, até abordar a versão de PRV com janela de tempo, tempo de viagem/serviço estocástico e multiobjetivo. Considerando-se que esta última versão do problema não apresenta estudos publicados na literatura e por ser uma variante relevante por sua capacidade de resolver problemas práticos, um objetivo deste trabalho foi desenvolver uma metaheurística para resolver este problema.

Os estudos realizados no capítulo 6 tratam inicialmente do problema estatístico para calcular o nível de serviço dos clientes. Deixa-se claro que a existência de tempos de espera em problemas em que o tempo de viagem/serviço é assumido normal faz com que a distribuição dos tempos de chegada não siga uma distribuição normal. Este fenômeno é então estudado, e se propôs um método inédito para computar o nível de serviço dos clientes. O modelo desenvolvido explora a convolução da soma de variáveis aleatórias e consiste num método numérico eficiente para resolver a convolução no contexto estudado. Algoritmos detalhados foram fornecidos assim como estudos de seus parâmetros. Para seu estudo foi necessário desenvolver um algoritmo capaz de gerar instâncias que tornassem possível observar o fenômeno em diversos graus de intensidade. Também foi necessário desenvolver um algoritmo de simulação para coletar os dados estatísticos que foram usados para validar as estimativas obtidas pelo método proposto.

A segunda parte do capítulo 6 mostrou de forma concisa a metaheurística desenvolvida para resolver o problema. Vários algoritmos são explicitados para mostrar o mecanismo de funcionamento da metaheurística, desde a geração de soluções iniciais, diferentes estratégias de buscas locais, operadores de seleção, cruzamento e critérios de saída.

O capítulo 7 fornece uma série de resultados para diversos experimentos realizados. Inicialmente realizou-se um experimento com 9037 rotas para testar o método proposto de cálculo de nível de serviço nas situações mais diversas possíveis e compará-lo com um benchmark. Os resultados mostraram que o método proposto foi de quatro a nove vezes mais rápido que o benchmark, fornecendo ainda soluções de melhor qualidade.

Ainda no capítulo 7, experimentos foram conduzidos para explorar o funcionamento de características chaves da metaheurística e evidenciam o funcionamento de estratégias como a zona circular para distância granular, rotas e movimentos tabus, critérios de saída e diferentes níveis especificados para o nível de serviço dos clientes. A seguir, a metaheurística resolve todas as 56 instâncias de Solomon e discute a qualidade das soluções obtidas. Mostra-se a fronteira pareto obtida para todas as instâncias e discute-se a interpretação das mesmas, para finalmente, avaliar o valor adicionado pelas fronteiras pareto no processo de tomada de decisão, implementando-se um método de MCDM e discutindo-se a escolha de uma única solução dentre as soluções disponíveis na fronteira pareto. Os resultados obtidos mostram-se coerentes e evidencia que a abordagem multiobjetiva agrega valor ao ajudar o tomador de decisão a avaliar as relações de custo benefício entre os dois objetivos (custo operacional e nível de serviço).

A contribuição científica desta tese é evidenciada pelo fato de, até onde conhecemos, não existirem trabalhos na literatura para resolver o PRVJT com tempo de viagem/serviço estocástico e abordagem multiobjetiva. Além de propor uma heurística inédita, também se desenvolveu um novo método para calcular o nível de serviço dos clientes, um ponto chave do problema proposto.

As soluções e métodos desenvolvidos nesta tese abrem portas para estender suas aplicações para casos como:

- Instâncias com maior número de clientes.
- Tempos de viagem com diferentes tipos de função distribuição de probabilidade, como por exemplo gamma e lognormal.
- Estudo de diferentes objetivos envolvendo nível de serviço, ou mesmo com mais de dois objetivos.
- Variantes do PRV com frota heterogênea, multi-depósitos e *pick-up and delivery*, por exemplo.

- Caso em que não necessariamente todos os clientes precisam ser visitados. Por exemplo, ao computar-se uma baixa probabilidade de visitar um cliente dentro da janela de tempo, pode-se admitir uma penalidade por não o visitar, para por outro lado, aumentar a probabilidade de visitar os clientes seguintes dentro da janela de tempo.

REFERÊNCIAS BIBLIOGRÁFICAS

ABRAMOWITZ M.; STEGUN I. A. (1964). **Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tools**. National Bureau of Standards Applied Mathematics Series.

ALVARENGA G. B.; MATEUS G. R. (2004). **A Two-Phase Genetic and Set Partitioning Approach for the Vehicle Routing Problem with Time Windows**. Fourth International Conference on Hybrid Intelligent Systems (HIS04), IEEE Computer Society Press.

ARMAS, J.; MELIÁN-BATISTA, B.; MORENO-PÉREZ, J. E BRITO, J. (2015). **GVNS for a real-world Rich Vehicle Routing Problem with Time Windows**. Engineering Applications of Artificial Intelligence. Volume 42.

ASSIS L. P. (2007). **Algoritmos para o Problema de Roteamento de Veículos com Coleta e Entrega Simultâneas**. Dissertação (Mestrado). UFMG. Instituto de Ciências Exatas.

BAKER, E.K. (1982). **Vehicle routing with time windows constraints**. *Logistic and Transportation Review*, 18(4), p.385-401.

BAKER B. M.; AYECHEW M.A. (2003). **A genetic algorithm for the vehicle routing problem**. *Computers & Operations Research* 30 787–800

BALDACCI, R.; MINGOZZIB, A.; ROBERTIC, R. (2012). **Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints**. *European Journal of Operational Research*. Volume 218, Issue 1.

BAÑOS, R.; ORTEGA, J.; GIL, C.; FERNÁNDEZ, A.; TORO, F. (2013). **A Simulated Annealing-based parallel multi-objective approach to vehicle routing problems with time windows**. *Expert Systems with Applications*. Volume 40, Issue 5, 1696–1707

BARD, J.F.; KONTORAVDIS, G.; YU, G. (2002). **A Branch and Cut Procedure for the Vehicle Routing Problems with Time Windows**. *Transportation Science*, 36(2), p.250-269.

BERGEGLIA, G.; CORDEUA, J. & LAPORTE, G. (2010). **Dynamic pickup and delivery problems**. *European Journal of Operations Research*, Volume 202, 8-15

BERTSIMAS D. (1992). **A vehicle routing problem with stochastic demand**. *Operations Research*, 40(3).

BINART, S.; DEJAX, P.; GENDREAU, M.; SEMETA F. (2016). **A 2-stage method for a field service routing problem with stochastic travel and service times**. *Computers & Operations Research*. Volume 65, January 2016, Pages 64–75

BLICKLE, T.; THIELE, L. (1996). **A comparison of selection schemes used in evolutionary algorithms**. *Journal Evolutionary Computation*. Volume 4, Issue 4.

BRÄYSY, O.; GENDREAU, M. (2005). **Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms**. *Transportation Science*. Volume 39, Issue 1.

BULLNHEIMER, B.; HARTL R.F.; STRAUSS C. (1999). **An improved ant system for the vehicle routing problem**. *Annals of Operations Research*, vol. 89.

CACERES-CRUZ J.; ARIAS, P.; GUIMARANS, D.; RIERA, D.; JUAN, A. 2015. **Rich vehicle routing problem: Survey**. *ACM Computing Surveys (CSUR)*. Volume 47.

CHANG, T., WAN Y.; TSANG W. (2009). **A stochastic dynamic traveling salesman problem with hard time Windows**. *European Journal of Operational Research*, vol. 198.

CHENG, C.; WANG, K. (2009). **Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm**. *Expert Systems with Applications*, vol. 36.

CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. (1979). **The vehicle routing problem**. *Combinatorial Optimization*. Wiley, Chichester.

CLARKE G; WRIGHT, J. V. (1964). **Scheduling of vehicles from a central depot to a number of delivery points**. *Operations Research*, vol. 12.

CORDEAU, J.-F.; LAPORTE G.; MERCIER A. (2000). **A unified tabu search heuristic for vehicle routing problems with time windows**. Technical Report CRT-00-03, Centre for Research on Transportation, Montreal, Canada.

DANTZIG G.B.; RAMSER, J.H. (1959). **The truck dispatching problem**. *Management Science*, vol. 6.

DEB, K. (2001). **Multi-Objective Optimization using Evolutionary Algorithms**. Editor: Wiley.

DEB, K. (2014). **Multi-objective Optimization**. In: *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Second Edition. Springer US.

DEMIRA, E.; BEKTAŞB, T.; LAPORTE G. (2014). **The bi-objective Pollution-Routing Problem**. *European Journal of Operational Research*. Volume 232.

DESAULNIERS G. MADSEN O., ROPKE S. (2014). **The vehicle routing problem with time windows**. In: Toth, P., Vigo, D. (Eds.), *Vehicle Routing: Problems, Methods, and Applications*, Second Edition. SIAM.

DESROCHERS, M.; LENSTRA, J.K.; SAVELSBERGH M.W.P. (1990). **A classification scheme for vehicle routing and scheduling problems**. *Journal of Operational Research Society*, vol. 46.

DESROCHERS, M.; DESROSIERS, J.; SOLOMON, M. (1992). **A new optimization algorithms for Vehicle Routing Problem with Time Windows**. Operations Research, vol. 40.

DUECK G.; SCHEURER T. (1990). **A general purpose optimization algorithm**. Journal of Computational Physics, vol. 90.

DROR M.; TRUDEAU P. (1986). **Stochastic vehicle routing with modified savings algorithm**. European Journal of Operational Research, vol. 23.

DROR, M., LAPORTE, G. AND TRUDEAU, P. (1989), **Vehicle Routing with Stochastic Demands: properties and solution frameworks**, Transportation Science, vol. 23.

DROR, M. (2016). **Vehicle routing with stochastic demands: Models & computational methods**. In International Series in Operations Research and Management Science. (Vol. 46, pp. 425-449). Springer New York LLC.

DULLAERT, W. (2000). **Impact of relative route length on the choice of time insertion criteria for insertion heuristics for the vehicle routing problem with time windows**. Knowledge Tools Transportation Logist. Faculty of Engineering, University of Rome, Italy, 153–156.

DULLAERT, W.; O. BRÄYSY. (2003). **New insertion criteria for Solomon's (1987) insertion heuristic for the vehicle routing problem with time windows**. Internal Report STF42 A03002, SINTEF Applied Mathematics, Department of Optimisation, Oslo, Norway (2003).

EHMKE, J., ANN M.; URBAN T. (2015). **Ensuring service levels in routing problems with time windows and stochastic travel times**. European Journal of Operational Research 240.2: 539-550.

EMMERICH, M.; BEUME, N.; NAUJOKS, B. (2005). **An EMO Algorithm Using the Hypervolume Measure as Selection Criterion**. Chapter in: Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science. Volume 3410. pp 62-76.

ERRICO F., DESAULNIERS G., GENDREAU M., REI W., RUSSEAU L. (2013). **The vehicle routing problem with hard time windows and stochastic service times**. Cahier Du GERAD, G-2013-45.

ERRICO, F.; DESAULNIERS, G.; GENDREAU, M.; REI, W.; ROUSSEAU L. (2016). **A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times**. European Journal of Operational Research. 249 (1), 55-66

FEO T. A.; RESENDE M. G. C. (1995). **Greedy randomized adaptive search procedures**. Journal of Global Optimization, vol. 6.

FISHER. M.L. (1995). **Vehicle routing**. Handbooks in Operations Research and Management Science, vol. 8, North-Holland, Amsterdam.

FLEISCHER, M. (2003). **The Measure of Pareto Optima Applications to Multi-objective Metaheuristics**. Chapter in: Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science. Volume 2632. pp 519-533.

FUKASAWA R.; LONGO H.; LYSGAARD J.; ARAGÃO M. P.; REIS M.; UCHOA E.; WERNECK R. F. (2006). **Robust branch-and-cut-and-price for the capacitated vehicle routing problem**. Mathematical Programming, vol. 106.

GARCIA-NAJERA, A.; BULLINARIA, JA. (2011). **An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows**. Computers & Operations Research. Vol. 38 (1), 287-30.

GAREY M. R.; JONHSON D.S. (1979). **Computers and Intractability: A Guide to the Theory of NP-Completeness**. W. H. Freeman.

GENDREAU M.; HERTZ A.; LAPORTE G. (1994). **A Tabu Search Heuristic for the Vehicle Routing Problem**. Management Science, vol. 40.

GENDREAU M.; LAPORTE G.& SEGUIN R. (1996). **Stochastic vehicle routing**. European Journal of Operational Research, vol. 88.

GENDREAU M. (2010). Conferencia no 42º Simpósio Brasileiro de Pesquisa Operacional (SBPO). <http://w3.ufsm.br/42sbpo/index.php?canal=material>. Acessado em maio de 2016.

GILLET B.E.; MILLER L.R. (1974). **A heuristic algorithm for the vehicle dispatch problem**. Operations Research, vol. 22.

GLOVER, F.; LAGUNA, M. (1997). **Tabu Search**. Kluwer, Boston.

GOLDEN, B.L.; WASIL, E.A.; KELLY, J.P.; CHAO. I.M. (1998). **Metaheuristics in vehicle routing**. Fleet Management and Logistics, Kluwer, Boston, pp. 33-56.

GROËR, C.; GOLDEN, B.; WASIL, E. (2009). **The balanced billing cycle vehicle routing problem**. Networks. Volume 54, Issue 4

HANSEN P.; MLADENOVIC N. (2003). **A tutorial on variable neighborhood search**. Technical Report, Les Cahiers du GERAD, HEC Montreal and GERAD.

HERNANDEZ F.; FEILLETB, D.; GIROUDEAUC, R.; NAUD, O. (2016). **Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows**. European Journal of Operational Research. Volume 249.

HOLLAND J.H. (1975). **Adaptation in Natural and Artificial Systems**. The University of Michigan Press, Ann Arbor, MI.

HOPFIELD J.J.; TANK D.W. (1985). **Neural computation of decisions in optimization problems**. Biological Cybernetics, vol. 52.

JAILLET P.; ODONI A. (1988). **Vehicle Routing: Methods and Studies**, In: The Probabilistic Vehicle Routing Problem. North-Holland, Amsterdam.

JIE G. (2010). **Model and Algorithm of Vehicle Routing Problem with Time Windows in Stochastic Traffic Network**. Logistics Systems and Intelligent Management, vol. 3.

JULA H., DESSOUKY M.; IANNOU P. (2006). **Truck route planning in nonstationary stochastic networks with time windows at customer locations**. IEEE Transactions on Intelligent Transportation Systems, vol. 7.

KAO E. (1978). **A preference order dynamic program for a stochastic travelling salesman problem**. Operations Research, vol. 26.

KENYON A. S.; MORTON D. P. (2003). **Stochastic Vehicle Routing with Random Travel Times**. Transportation Science, vol. 37.

KINDERVATER G.A.P.; SAVELSBERGH M.W.P. (1997). **Vehicle routing: Handling edge exchanges**. In E.H.L. Aarts; J.K. Lenstra, editors, Local Search in Combinatorial Optimization, Wiley, Chichester, UK, pp. 337-360.

KOK, A.; HANSB, E.; SCHUTTENB, J. (2012). **Vehicle routing under time-dependent travel times: The impact of congestion avoidance**. Computers & Operations Research. Volume 39, Issue 5.

KOC, C.; KARAOGLAN, I. (2016). **The green vehicle routing problem: A heuristic based exact solution approach**. Applied Soft Computing. Vol. 39.

KOLEN, A.W.J.; RINNOOY, A.H.G.; TRIENEKENS, H. (1987). **Vehicle Routing with Time Windows**. Operations Research, vol. 35.

LAPORTE G.; NOBERT. Y. (1987). **Exact algorithms for the vehicle routing problem**. Annals of Discrete Mathematics, vol. 31.

LAPORTE, G.; LOUVEAUX, F.; MERCURE, H. (1992). **The vehicle routing problem with stochastic travel times**. Operations Research Society of America, vol. 26.

LAPORTE, G. (2009). **Fifty Years of Vehicle Routing**. Transportation Science, vol. 43.

LARSEN A. (2000). **The Dynamic Vehicle Routing**, PhD. Thesis. Department of Mathematical Modelling (IMM) at the Technical University of Denmark (DTU).

LENSTRA, J.; RINNOOY KAN, A. (1981). **Complexity of vehicle routing and scheduling problems**, Networks, vol.11.

LI X.; TIAN P.; LEUNG S. C. H. (2010). **Vehicle routing problems with time windows and stochastic travel and service times: models and algorithms**. Int. J. Production Economics, vol. 125.

LIN S. (1965). **Computer solutions of the traveling salesman problem**. Bell System Technical Journal, vol. 44.

LIN S.; KERNIGHAN B.W. (1973). **An effective heuristic algorithm for the traveling salesman problem**. Operations Research, vol. 21.

LOURENÇO, H. R.; MARTIN, O. C.; STÜZLE T. (2003). **Handbook of Metaheuristics**, chapter Iterated Local Search, pages 321–353. Kluwer Academic Publishers.

MILLER-HOOKS E, MAHMASSANI HS (1998). **Optimal routing of hazardous materials in stochastic, time-varying transportation networks**. Transp. Res. Rec. 1645:143–151

MIRANDA, D. M.; CONCEIÇÃO S. V. (2010). **Um problema de roteamento dinâmico, com tempos de viagem estocásticos, múltiplos veículos e janelas de tempo**. Anais do 42º Simpósio Brasileiro de Pesquisa Operacional (SBPO). http://www.sobrapo.org.br/sbpo2010/xliisbpo_pdf/71954.pdf
Acessado em maio de 2016.

MIRANDA, D. M. (2011). **Metaheurísticas para as variantes do problema de roteamento de veículos: capacitado, com janela de tempo e com tempo de viagem estocástico**. Dissertação de Mestrado. Escola de Engenharia - UFMG.

MIRANDA, D. M.; CONCEIÇÃO S. V. (2016). **The vehicle routing problem with hard time windows and stochastic travel and service time**. Expert Systems with applications. 64: 104-116.

MOSCATO, P. (1989). **On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms**, Concurrent Computation Program, C3P Report 826, Caltech, Pasadena, U.S.A.

MOSCATO, P. E COTTA, C. (2003). **Una introduccion a los algoritmos memeticos**. Revista Ibero-americana de Inteligência Artificial, v. 19, p. 131–148.

NAGATA, W. (2007). **Edge assembly crossover for the capacitated vehicle routing problem**. Lecture Notes in Computer Science, vol. 4446.

NICOLAS, J.; FREDERIC S.; EL-GHAZALI T. (2008) **Multi-objective vehicle routing problems**. European Journal of Operational Research 189 (2008) 293–309

NICHOLSON T. (1971). **Optimization in industry, optimization techniques**. Longman Group Limited, v.1.

OSMAN I.H. (1993). **Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem**. Annals of Operations Research, vol. 41.

OR I. (1976). **Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking**. PhD. dissertation, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.

PEIQING LI, JIE H., DUNYONG Z., YONGSHENG H.; CHENHAO F. (2015). **Vehicle Routing Problem with Soft Time Windows Based on Improved Genetic Algorithm for Fruits and Vegetables Distribution**. *Discrete Dynamics in Nature and Society*, Volume 2015.

PEREIRA D. L. (2010). **Heurísticas e algoritmo exato para o problema de roteamento de veículos com coleta e entrega simultâneas**. Dissertação (Mestrado), Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas (UFMG).

POTVIN J. Y.; ROBILLARD C. (1995a). **Clustering for vehicle routing with a competitive neural network**. *Neurocomputing*, vol. 8.

POTVIN, J.-Y., J.-M. ROUSSEAU. (1995b). **An exchange heuristic for routing problems with time windows**. *J. Oper. Res. Soc.* vol. 46.

POTVIN, J.-Y (2007). **Evolutionary Algorithms for Vehicle Routing**. CIRRELT-2007-48.

PSARAFTIS, H. (1980) **A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem**. *Transportation Science*, vol. 14.

PSARAFTIS, H. (1995). **Dynamic Vehicle Routing: Status and Prospects**. *Annals of Operations Research*, vol. 61.

PSARAFTIS, H.; WEN, M.; KONTOVAS, C. (2016). **Dynamic vehicle routing problems: Three decades and counting**. *Networks*. Volume 67, issue 1.

PUREZA V. M. (1990). **Problemas de Roteamento de Veículos Via Metaheurística Tabu**. Dissertação (Mestrado). Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas.

QI Y.; HOU, Z.; LI, H.; HUANG, J. (2015). **A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows**. *Computers & Operations Research* 62, 61-77.

REEVES C.R. (1993). **Modern Heuristic Techniques for Combinatorial Problems**. John Wiley; Sons. Inc. New York, NY.

RENAUD, J.; BOCTOR, F. F. (2002). **A sweep-based algorithm for the fleet size and mix vehicle routing problem**, *European Journal of Operational Research*, vol. 140.

RIBEIRO C. C. (2004). **Curso “Metaheurísticas”, XI Escola Brasileira de Computação**. Disponível em: < http://www.ic.uff.br/~celso/grupo_de_pesquisa.htm> Acesso em julho/2011.

RITZINGERA, U.; PUCHINGERA, J. AND HARTLB, R. (2016). **A survey on dynamic and stochastic vehicle routing problems**. *International Journal of Production Research*. Volume 54, Issue 1.

ROCHAT, Y.; TAILLARD, E. D. (1995). **Probabilistic diversification and intensification in local search for vehicle routing**. Journal of Heuristics, vol. 1.

RUSSELL, R. (1977). **An effective heuristic for the M-tour traveling salesman problem with some side conditions**. Oper. Research. vol. 25.

RUSSELL, R. A.; T. L. URBAN (2008). **Vehicle routing with soft time windows and Erlang travel times**. Journal of Operational Research Society, 59, 1220-1228.

SALANI, M.; BATTARRA, M.; GAMBARDELLA L. (2014). **Exact Algorithms for the Vehicle Routing Problem with Soft Time Windows**. Operations Research Proceedings, 481-486

SIVARAMKUMAR, V.; THANSEKHAR, MR.; SARAVANAN R. (2015). **Multi-objective vehicle routing problem with time windows: Improving customer satisfaction by considering gap time**. Journal of Engineering Manufacturing.

SOLOMON, M. M. (1986). **On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints**. Networks, vol. 16.

SOLOMON, M. M. (1987). **Algorithms for the vehicle routing and scheduling problems with time window constraints**. Oper. Research, vol. 35.

SOLOMON, M. M.; DESROSIERS, J. (1988). **Time window constrained routing and scheduling problems**. Transportation Science, vol. 22.

TAS DUYGUN. NICO DELLAERT. TOMVAN WOENSEL. TONDE KOK (2013). **Vehicle routing problem with stochastic travel times including soft time windows and service costs**. Computers & Operations Research, vol. 40, 214–224

TAS DUYGUN. NICO DELLAERT. TOMVAN WOENSEL. TONDE KOK. (2014). **The time-dependent vehicle routing problem with soft time windows and stochastic travel times**. Transportation Research, Part C 66–83

TAILLARD E. D. (1993). **Parallel iterative search methods for vehicle routing problems**. Networks, vol. 23.

TAILLARD E. D.; BADEAU P.; GENDREAU M.; GUERTIN F.; POTVIN J.Y. (1997). **A tabu search heuristic for the vehicle routing problem with soft time windows**. Transportation Science, vol. 31.

TARANTILIS C.D. (2005). **Solving the vehicle routing problem with adaptive memory programming methodology**. Computers & Operations Research, vol. 32.

THOMPSON P.M.; PSARAFTIS H.N. (1993). **Cyclic transfer algorithms for multi-vehicle routing and scheduling problems**. Operations Research, vol. 41.

TILLMAN F. (1969). **The multiple terminal delivery problem with probabilistic demands**. Transportation Science, vol. 3.

TOTH P.; VIGO D. (1998a). **Exact algorithms for vehicle routing**. In T. Crainic; G. Laporte, editors, *Fleet Management and Logistics*, Kluwer, Boston, MA, pp. 1-31.

TOTH, P.; VIGO, D., (1998b). **Granular tabu search**. Working paper, DEIS, University of Bologna.

TOTH, P.; VIGO, D., (2002a). **Models, relaxations and exact approaches for the capacitated vehicle routing problem**. *Discrete Applied Mathematics* vol. 123.

TOTH, P.; VIGO, D., (2002b). **The Vehicle Routing Problem**. Copyright © 2002 by Society for Industrial and Applied Mathematics.

TRANSPORTATION OPTIMIZATION PORTAL (2016).

<https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/>

Acessado em 19.05.2016

ULMER, M.; MATTFELD, D.; HENNIG, M.; GOODSON, J. (2015). **A Rollout Algorithm for Vehicle Routing with Stochastic Customer Requests**. In book: *Logistics Management*, Publisher: Springer, pp.217-227

URQUHART, NEIL B, HART, EMMA AND JUDSON, ALISTAIR (2015). **Multi-Modal employee routing with time windows in an urban environment**. In: *Proceedings of the 2015 Genetic and Evolutionary Algorithms Conference*. ACM, Madrid, pp. 1503-1504. ISBN 978-1-4503-3488-4

VIDAL, T.; CRAINIC, T. G.; GENDREAU, M.; PRINS, C. (2013). **A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows**. *Computers & Operations Research*. Volume 40, Issue 1

VIEIRA, H. P. (2009). **Metaheurística para a Solução de Problemas de Roteamento de Veículos com Janela de Tempo**. Dissertação (Mestrado). Unicamp, Instituto de Matemática, Estatística e Computação Científica.

VOUDOURIS C.; TSANG E. P. K. (2003). **Handbook of Metaheuristics**, chapter *Guided Local Search*, pages 186–218. Kluwer Academic Publishers.

WATERS, C.D.J. (1989), **Vehicle-scheduling problems with uncertainty and omitted customers**. *Journal of the Operational Research Society* vol. 40.

WILLARD, J. A. G., (1989). **Vehicle routing using r-Optimal Tabu Search**. M.Sc. Dissertation. The Management School, Imperial College, London.

ZHANG J.; WILLIAM H. K. LAM; BI YU CHEN (2013). **A Stochastic Vehicle Routing Problem with Travel Time Uncertainty: Trade-Off between Cost and Customer Service**. *Networks and Spatial Economics*, vol. 13:471–496

ZITZLER, E.; THIELE, L. (1999). **Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach**. *IEEE Transactions on Evolutionary Computation*, 3(4), 257–271.

ZITZLER E.; LAUMANNNS, M. ; THIELE T. (2001). **SPEA2: Improving the Strength Pareto Evolutionary Algorithm**. Technical report available on:
<http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.112.5073>
Acessado em maio de 2016.

ZIVIANI, N. (2010). **Projeto de Algoritmos com Implementações em Pascal e C**. Terceira edição. Cengage Learning.

ANEXOS

Valores de referência para hipervolume

Definiu-se um método para obter as coordenadas do ponto de referência para calcular o hipervolume da fronteira pareto de cada instância. Um experimento foi conduzido para executar uma geração do algoritmo para as 56 instâncias, armazenando-se as coordenadas correspondentes ao maior e menor valor do objetivo 1 e do objetivo 2, respectivamente: $Ref_{max}^1, Ref_{min}^1, Ref_{max}^2$ e Ref_{min}^2 . Assim, ao longo da execução do algoritmo, quando é necessário calcular o volume da fronteira pareto, os valores dos objetivos são normalizados como: $x_n^i = (x^i - Ref_{min}^i) * 100 / (Ref_{max}^i - Ref_{min}^i)$, onde x_n^i é o valor normalizado quanto ao objetivo $i = 1$ ou $i = 2$, e x^i é o valor original da função objetivo i . Após esta normalização, o hipervolume é calculado usando-se a distância de lebesgue (Deb, 2014) em que o ponto de referência para ambos os objetivos tem valor 2000. Após calculado o hipervolume, ele é dividido por 400 apenas para melhorar a legibilidade do número. Divide-se por 400 pois o valor 2000 ($20 * 20$) usado como referência é 20 vezes maior que o valor 100. Como os valores de referência máximo e mínimo (Ref_{max}^i e Ref_{min}^i) foram feitos para apenas 1 geração, nos experimentos completos é possível encontrar valores menores e maiores que os pontos de referência, permitindo que a normalização retorne valores menores que zero e/ou maiores que 100. Então o valor 2000 foi uma forma de garantir que todas as soluções após normalizadas nunca tivessem um valor superior a 2000, o que poderia afetar o cálculo de hipervolume. A tabela a seguir possui os valores de referência utilizados para a normalização. O valor x refere-se ao custo operacional dividido por 1000, e y refere-se ao nível de serviço.

#	Instância	Ref_{max}^1	Ref_{max}^2	Ref_{min}^1	Ref_{min}^2
1	C101	19.6322	-0.8902	16.5047	-0.9560
2	C102	17.4114	-0.9098	14.4275	-0.9774
3	C103	15.3425	-0.9348	13.381	-0.9815
4	C104	13.3557	-0.9663	12.0341	-0.9902
5	C105	16.2944	-0.9046	13.2029	-0.9644
6	C106	16.5744	-0.9216	14.1241	-0.9773
7	C107	12.0161	-0.8898	10.882	-0.9366
8	C108	15.6113	-0.9172	10.9549	-0.9856
9	C109	15.7363	-0.9372	10.9056	-0.9868
10	C201	6.8564	-0.941	6.4806	-0.996
11	C202	6.7369	-0.9544	5.4193	-0.9965
12	C203	5.6422	-0.9579	5.2553	-0.9963
13	C204	5.1504	-0.9674	3.994	-0.998
14	C205	5.5675	-0.9576	5.2944	-0.9962
15	C206	5.4918	-0.9621	4.1852	-0.9971
16	C207	5.4084	-0.9677	4.071	-0.9985
17	C208	4.111	-0.9756	3.9069	-0.9986
18	R101	29.1895	-0.9433	25.9985	-0.989
19	R102	25.9663	-0.9464	23.8536	-0.9889
20	R103	20.8072	-0.9447	18.5668	-0.989
21	R104	16.5405	-0.9397	14.2697	-0.9909
22	R105	22.9511	-0.9264	17.6047	-0.982
23	R106	18.6599	-0.9257	15.4512	-0.9774
24	R107	15.5073	-0.9374	14.306	-0.9877
25	R108	14.4359	-0.9494	12.1575	-0.9909
26	R109	17.8477	-0.9267	15.4055	-0.9874
27	R110	15.5467	-0.9289	14.3419	-0.986
28	R111	15.5384	-0.9311	13.2615	-0.9749
29	R112	15.6142	-0.9407	12.2221	-0.9895
30	R201	6.9014	-0.9445	6.4396	-0.9927
31	R202	6.769	-0.9539	5.3993	-0.9942
32	R203	5.6083	-0.9588	5.1913	-0.9952
33	R204	5.098	-0.9694	3.9657	-0.9983
34	R205	6.3308	-0.9564	5.266	-0.9986
35	R206	6.3957	-0.9641	4.1976	-0.9955
36	R207	5.2828	-0.9665	4.0691	-0.9991
37	R208	4.1131	-0.9768	3.916	-0.9976
38	R209	5.4367	-0.9664	5.1743	-0.9956
39	R210	6.4126	-0.9589	5.1891	-0.9975
40	R211	5.0909	-0.9704	3.9902	-0.9973
41	RC101	26.4485	-0.9301	20.9141	-0.9808
42	RC102	19.9248	-0.9338	17.7485	-0.9826
43	RC103	16.9166	-0.9345	14.5919	-0.9854
44	RC104	14.6747	-0.9431	13.3205	-0.9842
45	RC105	22.2253	-0.9307	19.8587	-0.986
46	RC106	18.961	-0.9328	15.6245	-0.9837
47	RC107	17.7209	-0.9404	14.4653	-0.9954
48	RC108	16.6845	-0.9438	13.3684	-0.9867
49	RC201	8.0505	-0.9403	6.7425	-0.9905
50	RC202	6.7251	-0.9539	5.5726	-0.9969
51	RC203	6.3309	-0.9675	5.3055	-0.9957
52	RC204	5.487	-0.977	5.0965	-0.9985
53	RC205	8.8259	-0.9551	6.6517	-0.9919
54	RC206	6.8459	-0.9506	5.4213	-0.994
55	RC207	7.0531	-0.9625	5.3346	-0.9946
56	RC208	5.5255	-0.9679	4.0493	-0.9984