

UNIVERSIDADE FEDERAL DE OURO PRETO

Análise e Otimização do Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis

Lucas Carvalho Oliveira Matsueda
Universidade Federal de Ouro Preto

Orientador: Alan Robert Resende de Freitas

Coorientador: Frederico Gadelha Guimarães

Dissertação submetida ao Instituto de Ciências Exatas e Biológicas da Universidade Federal de Ouro Preto para obtenção do título de Mestre em Ciência da Computação

Ouro Preto, Maio de 2015

Análise e Otimização do Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis

Lucas Carvalho Oliveira Matsueda
Universidade Federal de Ouro Preto

Orientador: Alan Robert Resende de Freitas

Coorientador: Frederico Gadelha Guimarães

M434a Matsueda, Lucas Carvalho Oliveira.
Análise e otimização do problema de roteamento de veículos com muitos objetivos e janelas de tempo flexíveis [manuscrito] / Lucas Carvalho Oliveira Matsueda. - 2015.
171f.:

Orientador: Prof. Dr. Alan Robert Resende de Freitas.
Coorientador: Prof. Dr. Frederico Gadelha Guimarães.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Departamento de Computação. Programa de Pós-graduação em Ciência da Computação.
Área de Concentração: Otimização e Inteligência Computacional.

1. Veículos. 2. Otimização combinatoria. 3. Programação evolutiva (Computação). I. Freitas, Alan Robert Resende de. II. Guimarães, Frederico Gadelha. III. Universidade Federal de Ouro Preto. IV. Título.

CDU: 004.75:656.07

Catálogo: www.sisbin.ufop.br



Ata da Defesa Pública de Dissertação de Mestrado

Aos 27 dias do mês de maio de 2015, às 15:30 horas na Sala Multimídia do Instituto de Ciências Exatas e Biológicas (ICEB), reuniram-se os membros da banca examinadora composta pelos professores: **Prof. Dr. Alan Robert Resende de Freitas (presidente e orientador)**, **Prof. Dr. Frederico Gadelha Guimarães**, **Prof. Dr. Haroldo Gambini Santos** e **Profa. Dra. Elizabeth Fialho Wanner**, aprovada pelo Colegiado do Programa de Pós-Graduação em Ciência da Computação, a fim de arguirem o mestrando **Lucas Carvalho Oliveira Matsueda**, com o título “**Análise e Otimização do Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis**”. Aberta a sessão pelo presidente, coube ao candidato, na forma regimental, expor o tema de sua dissertação, dentro do tempo regulamentar, sendo em seguida questionado pelos membros da banca examinadora, tendo dado as explicações que foram necessárias.

Recomendações da Banca:

Aprovada sem recomendações

Reprovada

Aprovada com recomendações: _____

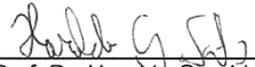
Banca Examinadora:



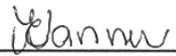
Prof. Dr. Alan Robert Resende de Freitas



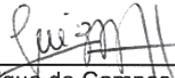
Prof. Dr. Frederico Gadelha Guimarães



Prof. Dr. Haroldo Gambini Santos



Profa. Dra. Elizabeth Fialho Wanner



Prof. Dr. Luiz Henrique de Campos Merschmann
Coordenador do Programa de Pós-Graduação em Ciência da Computação
DECOM/ICEB/UFOP

Ouro Preto, 27 de maio de 2015.

Dedico este trabalho a Deus e aos meus pais, Satoshi e Magda.

Análise e Otimização do Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis

Resumo

Para explorar a interseção entre problemas de roteamento de veículos propostos na literatura, esta dissertação propõe um problema de roteamento de veículos com muitos objetivos e janelas de tempo flexíveis (MOPRV). É proposta uma abordagem baseada em dois algoritmos evolucionários multiobjetivo (NSGA-II e NSGA-III) e um método para a redução e visualização de objetivos (Árvores de Agregação) é proposta. Através de um estudo sobre a harmonia e conflito entre os objetivos do problema, foi observada a possibilidade de agregação entre os mesmos, reduzindo o problema de seis para três objetivos. Os experimentos demonstram que as soluções para o problema reduzido possuem bons valores para todos os objetivos quando comparado com as soluções do problema completo. Mais ainda, os resultados demonstram que é mais vantajoso visualizar a relação entre os objetivos do MOPRV e em seguida otimizar o problema com menos objetivos do que tentar otimizar diretamente o problema considerando todos os objetivos do MOPRV.

Analysis and Optimization of Many-Objective Vehicle Routing Problems with Flexible Time Windows.

Abstract

In order to explore the intersection between vehicle routing problems proposed in the literature, this dissertation proposes a many-objective vehicle routing problem with flexible time windows. We propose an approach based on two multiobjective evolutionary algorithms (NSGA-II and NSGA-III) and a method for reduction and visualization of objectives (Aggregation Trees). We observed the possibility of aggregation between the objectives through a study of the harmony and conflict between them, reducing the problem from six to three objectives. The experiments show the solutions for the reduced problem have good values for all objectives when compared to solutions for the complete problem. Moreover, the results show that it is more advantageous to visualize the relationship between objectives for the many-objective vehicle routing problem and then to optimize the reduced problem than to directly optimize the original formulation of the problem considering all six objectives.

Declaração

Esta dissertação é resultado de meu próprio trabalho, exceto onde referência explícita é feita ao trabalho de outros, e não foi submetida para outra qualificação nesta nem em outra universidade.

Lucas Carvalho Oliveira Matsueda

Agradecimentos

Primeiramente, agradeço a Deus pelas oportunidades que me foram dadas na vida, por me manter em pé quando não conseguia mais caminhar e por me dar forças para conseguir realizar um de nossos sonhos. Para mim, é simplesmente impossível negar que Ele conduz todos os meus passos e que sempre me dá a vitória.

Agradeço aos meus pais Satoshi e Magda, por todo apoio, incentivo e compreensão. Não há palavras que sejam capazes de expressar minha gratidão a vocês. Sem seu leal e incondicional apoio não teria chegado até aqui. Agradeço ao meu irmão Felipe pelo carinho. Agradeço à minha namorada Carol pela força, pelo amor e por sempre estar ao meu lado, mesmo nos momentos mais difíceis.

E por último e não mais importante, agradeço a todos os professores, colegas e amigos que contribuíram de alguma forma para que eu chegasse até aqui, muito obrigado. Agradeço em especial aos meus orientadores Alan Robert Resende de Freitas e Frederico Gadelha Guimarães pela dedicação, apoio, comprometimento destinados a mim e ao meu trabalho, bem como por todo suporte que me deram. A orientação de vocês foi imprescindível para o sucesso deste trabalho. Sou grato também aos professores Haroldo Santos, Gustavo Peixoto e Anderson Ferreira pelo conhecimento transmitido durante todo o curso.

Bom, para não esquecer de ninguém termino agradecendo a todos que me ajudaram direta ou indiretamente neste trabalho. Obrigado.

Sumário

Lista de Figuras	xxi
Lista de Tabelas	xxv
Lista de Algoritmos	xxvii
Nomenclatura	1
1 Introdução	3
1.1 Objetivos	5
1.1.1 Objetivos Específicos	5
1.2 Motivação	5
1.3 Contribuições	7
1.4 Organização do trabalho	8
2 Revisão Bibliográfica	11
2.1 Problema de Roteamento de Veículos	11
2.1.1 Distância Total Percorrida	12
2.1.2 Número de Veículos	13
2.1.3 Violação da Restrição de Tempo	14
2.1.4 Tempo de Espera dos Veículos	15

2.1.5	Maior Rota (Makespan)	16
2.1.6	Balanceamento de Rota	17
2.1.7	Problema de Roteamento de Veículos Multiobjetivo	18
2.2	Otimização Multiobjetivo	20
2.2.1	Conceitos Básicos	21
2.2.2	Algoritmos Evolucionários Multiobjetivo	23
2.2.3	Visualização	25
2.3	Otimização de Problemas com Muitos Objetivos	27
2.3.1	Métodos de Classificação de Soluções	29
2.3.2	Relações de Dominância	30
2.3.3	Medidas de Aptidão Envolvendo Indicadores	31
2.3.4	Redução de Dimensionalidade	32
2.3.5	Visualização	33
2.4	Conclusão	35
3	Problema de Roteamento de Veículos	37
3.1	Problema de Roteamento de Veículos Capacitado	38
3.2	Problema de Roteamento de Veículos com Janelas de Tempo	39
3.3	Problema de Roteamento de Veículos com Janelas de Tempo Flexíveis	42
3.4	Problema de Roteamento de Veículos com Balanceamento de Rota	43
3.5	Problema de Roteamento de Veículos Min-max	44
3.6	Outras Variantes do Problema de Roteamento de Veículos	45
3.7	Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis	48
3.7.1	Variáveis de Decisão	49

3.7.2	Funções Objetivo	50
3.7.3	Restrições	51
3.8	Conclusão	52
4	Fundamentos	55
4.1	Nondominated Sorting Genetic Algorithm II	55
4.2	Nondominated Sorting Genetic Algorithm III	60
4.2.1	Normalização Adaptativa dos Membros da População	62
4.2.2	Associação	64
4.2.3	Preservação do Nicho	65
4.3	Árvores de Agregação	67
4.3.1	Medidas de Conflito	71
4.3.2	Localidade de Conflito	73
4.3.3	Construção do Algoritmo	74
4.3.4	Exemplo de Execução	77
4.4	Conclusão	82
5	NSGA-II e NSGA-III para Resolução do MOPRV	85
5.1	Representação de uma Solução	85
5.2	Geração de uma Solução Inicial	86
5.3	Operadores de Cruzamento	87
5.3.1	Cruzamento de Mapeamento Parcial	88
5.3.2	Cruzamento de Ordem	90
5.4	Operadores de Mutação	91
5.4.1	Mutação por Inserção	91
5.4.2	Mutação por Inversão Simples	91

5.4.3	Mutação por Troca	92
5.5	Operador de Seleção	92
5.6	Pontos de Referência	93
5.7	Conclusão	95
6	Resultados	97
6.1	Instâncias	98
6.2	Avaliação dos Objetivos	99
6.3	Parâmetros	100
6.4	Planejamento Experimental	102
6.5	Resultados e Discussões	103
6.5.1	Agregação dos Objetivos	103
6.5.2	Otimizando o MOPRV Completo	115
6.5.3	Redução do MOPRV	119
6.5.4	Otimizando o MOPRV Reduzido	120
6.5.5	MOPRV Completo x MOPRV Reduzido	123
6.6	Conclusão	128
7	Considerações Finais	131
7.1	Conclusão	131
7.2	Trabalhos Futuros	133
	Referências Bibliográficas	135

Lista de Figuras

2.1	Mapeamento do espaço de parametros X no espaço de objetivo Y feito pela função $f(x)$	22
2.2	Visualização em 2 dimensões (Freitas et al., 2015)	26
2.3	Visualização em 3 dimensões (Freitas et al., 2015)	26
2.4	Visualização do resultado de um problema de otimização multiobjetivo com 7 funções objetivo (Freitas et al., 2015)	34
4.1	Ordenação por dominância	56
4.2	Distância de Multidão	58
4.3	Procedimento do NSGA-II (Deb et al., 2002)	58
4.4	Procedimento de formação do hiperplano para um problema com 3 objetivos (Deb e Jain, 2014)	64
4.5	Associação dos membros da população com os pontos de referência (Deb e Jain, 2014)	65
4.6	Exemplo de um conjunto de soluções com vários tipos de harmonia e conflito (Freitas et al., 2015)	69
4.7	Árvore de Agregação referente ao conjunto de soluções	69
4.8	Coordenadas paralelas com ordem diferente de objetivos adjacentes (Freitas et al., 2015)	70
4.9	Gráfico polar para o conjunto de soluções das Figuras 4.6 e 4.7 (Freitas et al., 2015)	71

4.10	Exemplos de vários tipos de conflito (Freitas et al., 2015)	75
4.11	Estrutura da árvore inicial com um nó raiz como pai de todos os objetivos	78
4.12	Estrutura da árvore com a agregação dos objetivos f_1 e f_4	79
4.13	Estrutura da árvore com a agregação dos objetivos $f_4 + f_1$ e f_2	81
4.14	Estrutura da árvore com a agregação dos objetivos $f_4 + f_1 + f_2$ e f_3	82
5.1	Exemplo da seção de mapeamento do PMX (Malaquias, 2006)	88
5.2	Cópia dos elementos da seção de mapeamento dos pais para os filhos (Malaquias, 2006)	89
5.3	Cópia dos elementos restantes dos pais para os filhos (Malaquias, 2006) .	89
5.4	Resultado da cópia da seção de mapeamento nos filhos (Malaquias, 2006)	90
5.5	Resultado de execução do OX (Malaquias, 2006)	91
5.6	Representação do processo feito pelo ISM (Malaquias, 2006)	91
5.7	Representação do processo feito pelo SIM (Malaquias, 2006)	92
5.8	Representação do processo feito pelo EM (Malaquias, 2006)	92
5.9	Pontos de referência em duas dimensões	94
5.10	Pontos de referência em três dimensões (Deb e Jain, 2014)	95
6.1	Disposição dos consumidores nos três grupos de instâncias de Solomon (1987)	99
6.2	Árvores de Agregação Média	105
6.3	Árvores de Agregação Específicas Semelhantes às Árvores Médias	107
6.4	Gráficos Polares das Instâncias Semelhantes	108
6.5	Gráficos Polares das instâncias do grupo R para o MOPRV Completo . .	116
6.6	Gráficos Polares das instâncias do grupo C para o MOPRV Completo . .	117
6.7	Gráficos Polares das instâncias do grupo RC para o MOPRV Completo .	117

6.8	Gráficos Polares das instâncias do grupo R para o MOPRV Reduzido . .	121
6.9	Gráficos Polares das instâncias do grupo C para o MOPRV Reduzido . .	122
6.10	Gráficos Polares das instâncias do grupo RC para o MOPRV Reduzido .	122
6.11	Árvore de Agregação para o MOPRV Reduzido	123

Lista de Tabelas

6.1	Parâmetros	101
6.2	Cobertura das instâncias do grupo R	126
6.3	Cobertura das instâncias do grupo C	127
6.4	Cobertura das instâncias do grupo RC	128

Lista de Algoritmos

4.1	Pseudocódigo FNDS	57
4.2	Pseudocódigo NSGA-II	59
4.3	Pseudocódigo NSGA-III	61
4.4	Pseudocódigo <i>Normalização</i> (f^n, S_t, Z^r, Z^s, Z^a)	62
4.5	Pseudocódigo <i>Associação</i> (S_t, Z^r)	65
4.6	Pseudocódigo <i>Preservação do Nicho</i> ($K, \rho_j, \pi, d, Z^r, F_t, P_{t+1}$)	66
4.7	Pseudocódigo Construção de uma Árvore de Agregação	76

“I want to put a ding in the universe.”

— Steve Jobs

Nomenclatura

AG	Algoritmos Genéticos
AGM	Algoritmos Genéticos Multiobjetivo
AT	<i>Aggregation Trees</i>
MOEA	<i>Multi-objective Evolutionary Algorithms</i>
NSGA	<i>Nondominated Sorting Genetic Algorithm</i>
NSGA-II	<i>Nondominated Sorting Genetic Algorithm II</i>
NSGA-III	<i>Nondominated Sorting Genetic Algorithm III</i>
PCV	Problema do Caixeiro Viajante
PRV	Problema de Roteamento de Veículos
PRVBR	Problema de Roteamento de Veículos com Balanceamento de Rota
PRVC	Problema de Roteamento de Veículos Capacitados
PRVJT	Problema de Roteamento de Veículos com Janelas de Tempo
PRVMO	Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis
PRVMM	Problema de Roteamento de Veículos Min-max
TI	Tecnologia de Informação

Capítulo 1

Introdução

A busca pela otimização dos processos, auxiliados por sistemas de informação, tem se tornado cada vez mais corrente em diversas áreas de pesquisa. Contudo, a logística é uma das áreas que mais sentiu a necessidade de aprimorar seus métodos. Devido ao constante desenvolvimento do mercado global, diversas exigências, relacionadas à logística e ao transporte, surgem a todo instante. Redução de custos, programação das entregas, redução e cumprimento dos prazos de entrega e disponibilidade constante dos produtos, são alguns exemplos de problemas logísticos.

Um dos principais problemas de ampla aplicação prática na logística é o Problema de Roteamento de Veículos (PRV) (*Vehicle Routing Problem - VRP*). De modo geral, o PRV consiste no atendimento de um conjunto de clientes (consumidores) por intermédio de uma frota de veículos. Tal problema apresenta uma demanda associada a cada cliente, uma capacidade de carga associada a cada veículo e um depósito central. Assim, a demanda de todos os clientes deve ser atendida por veículos que iniciam e terminam seu trajeto no depósito (rota). Quando apenas a restrição de capacidade de carga máxima da frota de veículos é definida, o problema é denominado de Problema de Roteamento de Veículos Capacitados (PRVC) (*Capacitated Vehicle Routing Problem - CVRP*).

Desde que o problema de roteamento de veículos foi proposto diversas variações surgiram com a finalidade de atender a uma série de problemas práticos, estes são formulados adicionando restrições ao PRV. Uma importante variação deste problema é o Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT) (*Vehicle Routing Problem with Time Windows - VRPTW*). Esta variante, além de possuir as restrições de capacidade, possui restrições que determinam o horário de atendimento aos

clientes. Assim, as restrições de tempo aproximam o problema da realidade, ampliando sua utilidade prática e, por consequência, resultando em diversas contribuições para a literatura.

Grande parte dos trabalhos presentes na literatura que abordam os PRVs apresentam soluções em que apenas um objetivo, relacionado à minimização da distância total percorrida pela frota de veículos, é tratado. Entretanto, diversos problemas de logística não limitam-se somente à esta meta visto que as soluções para estes problemas podem envolver outros objetivos relacionados ao custo de transporte, bem como envolver outros fatores como a satisfação do consumidor, satisfação do motorista, segurança e violação de restrições. O estudo de problemas de roteamento de veículos que procuram minimizar mais de um objetivo simultaneamente (otimização multiobjetivo) tem se tornado um atrativo para pesquisadores na área de otimização combinatória.

Diante deste contexto, este trabalho propõe um estudo sobre os principais objetivos tratados na literatura para o PRV, bem como para o PRVJT. Assim, uma formulação do PRV com restrições de tempo e muitos objetivos é proposta. Na metodologia deste trabalho, um *Nondominated Sorting Genetic Algorithm III* (NSGA-III) foi utilizado para otimizar todos os objetivos do modelo proposto em paralelo e as soluções foram comparadas com Árvores de Agregação (*Aggregation Tree* - AT). Sugestões de agregação levaram a uma nova formulação do problema em que os objetivos com maior harmonia foram agregados em novos objetivos. Posteriormente, o novo modelo reduzido foi otimizado por um *Nondominated Sorting Genetic Algorithm II* (NSGA-II). A parte experimental considera um conjunto bem conhecido de instâncias do problema (Solomon, 1987). Assim, a AT sugere muitas agregações possíveis no problema formulado, que tem bons resultados em todos os objetivos do modelo reduzido subsequente.

Este capítulo tem por objetivo contextualizar o problema e se divide da seguinte maneira. O objetivo geral e específicos são apresentados na Seção 1.1. Em seguida, a motivação para o estudo é descrita na Seção 1.2. As contribuições esperadas com essa dissertação encontram-se na Seção 1.3. Por fim, a estrutura geral do texto é apresentada na Seção 1.4.

1.1 Objetivos

O objetivo principal deste trabalho é formular um problema de roteamento de veículos com restrições de tempo que envolvem muitos objetivos. Dada a praticidade do problema diversos objetivos podem ser considerados. O problema proposto então deverá abordar os principais objetivos tratados na literatura para os PRVs. No entanto, os esforços deste trabalho são direcionados a desenvolver uma versão reduzida (com menos objetivos) do PRVJT com muitos objetivos, de modo que a perda na qualidade das soluções da frente de Pareto seja a menor possível.

1.1.1 Objetivos Específicos

Este trabalho se propõe aos seguintes objetivos específicos:

- Estudar as variações dos problemas de roteamento de veículos, mais especificamente do PRVJT, bem como estudar os principais objetivos abordados nestes problemas;
- Propor uma formulação do PRVJT com muitos objetivos;
- Estudar técnicas e abordagens de soluções para problemas de Otimização com Muitos Objetivos;
- Executar a ferramenta AT para verificar o conflito entre os objetivos propostos na formulação do PRVJT com muitos objetivos;
- Propor uma formulação reduzida do PRVJT com muitos objetivos
- Implementar algoritmos evolutivos multiobjetivo para os PRVJTs propostos;
- Verificar, através de testes, se os resultados obtidos pelos algoritmos evolutivos multiobjetivo confirmam o conflito apresentado pela AT e viabilizam a redução do problema com muitos objetivos.

1.2 Motivação

O conforto e a praticidade que o transporte rodoviário proporciona ao seu usuário contribuíram para que houvesse um crescimento significativo no tráfego de veículos, pro-

vocando um aumento no número de engarrafamentos, acidentes e desgaste das vias. Segundo o *Centro de Estudos em Logística* (2007) o transporte de carga no Brasil é tipicamente rodoviário, e em média, as grandes empresas transportam 88,3% de suas cargas por rodovia.

Para empresas transportadoras que atuam no segmento rodoviário, fatores adversos aliados ao custo de manutenção da frota de veículos encarecem o valor dos serviços prestados além de causar insatisfação dos envolvidos em todo processo de transporte. Dados apresentados pela *Associação Brasileira de Logística* (2008) afirmam que os custos de transporte correspondem a 60% dos custos logísticos das empresas, sendo que estes mesmos custos representaram em 2008, 6,9% do Produto Interno Bruto (R\$ 207,0 bilhões) do Brasil.

A redução dos custos com transporte tem se tornado cada vez mais determinante para aplicações logísticas, uma vez que o transporte representa uma parcela considerável no valor do produto final repassado ao consumidor. Trabalhos como o de Toth e Vigo (2001) afirmam que o uso de métodos computacionais em processos de distribuição frequentemente resulta em economia da ordem de 5% a 20% nos custos de transporte. Golden et al. (2001) ainda descrevem vários estudos de caso nos quais a aplicação de algoritmos ao PRV tem levado a reduções de custos substanciais. Assim, 74% das grandes empresas industriais têm priorizado a Tecnologia da Informação (TI) com a intenção de obter um maior controle das atividades relativas à área de transporte (*Centro de Estudos em Logística*, 2007).

Entretanto, recursos computacionais aplicados ao PRVJT podem gerar boas soluções que envolvam outros aspectos, além da redução de custos (minimização da distância total percorrida e minimização do número de veículos). Em tempos de globalização, a satisfação do consumidor é um ponto crucial a ser tratado pelas organizações, por exemplo. Neste sentido, a TI é capaz de apresentar soluções que garantem que o consumidor receba prontamente seu produto ou que a entrega seja feita com o menor atraso possível. De modo geral, a produção dos funcionários de uma empresa está diretamente ligada a sua satisfação. Assim, outra questão que pode ser tratada no PRVJT é a satisfação dos motoristas. Aqui, o importante é promover a equidade da carga de trabalho dos motoristas. Segurança de transporte, acessibilidade e aspectos geográficos são exemplos de outros objetivos que podem ser tratados no PRVJT.

Além disso, as exigências de performance do mundo corporativo, pressionado pela sociedade, questões políticas e legislativas, vão além dos aspectos financeiros e de satisfação

dos clientes e funcionários, englobando também questões de responsabilidade ambiental. A área da logística que se preocupa com os aspectos e impactos da atividade logística sobre a comunidade, bem como sobre o meio ambiente é denominada de “Logística Verde” (*Green Logistics*). A importância da Logística Verde é motivada pelo fato de que as estratégias de produção e logística de distribuição atuais não são sustentáveis a longo prazo. Assim, os efeitos ambientais, ecológicos e sociais devem ser considerados na concepção de políticas de logística. Uma melhor utilização dos veículos alcançaria mais diretamente sistemas de transporte sustentáveis. Neste sentido, é possível explorar a relação entre o efeito ambiental e o transporte através do planejamento de rotas, o que ajuda a diminuir a distância percorrida dos veículos e, conseqüentemente, a emissão de diversos poluentes. Assim, considerações de objetivos mais amplos e restrições mais operacionais que estão preocupados com as questões de logística sustentável também podem ser consideradas nos PRVs (Lin et al., 2014).

Na prática, diversas indústrias e empresas procuram atingir, no mínimo, duas destas metas simultaneamente. Assim, técnicas para resolução de PRVs que procuram otimizar dois ou mais objetivos ao mesmo tempo têm sido propostas constantemente. Porém, a aplicabilidade do PRVJT, bem como os objetivos que o compõem, não é o único motivador para o estudo deste problema. Visto que o PRVJT pertencente à classe de problemas NP-difíceis (Garey e Johnson, 1979), torna-se um grande desafio resolvê-lo de forma eficiente, dada a dificuldade de obter a melhor solução em tempos aceitáveis.

Sob qualquer ponto de vista: econômico, político e militar; o transporte é, inquestionavelmente, a indústria mais importante do mundo (Menchik, 2010).

1.3 Contribuições

As principais contribuições deste trabalho são:

- Disponibilizar um estudo sobre as variações do PRV, bem como dos objetivos tratados nesta classe de problemas.
- Descrever técnicas de redução de objetivos aplicadas ao PRVJT, apresentando o comportamento de algoritmos de redução de dimensionalidade aplicados a esta classe de problemas.
- Produzir conhecimento sobre o conflito/harmonia dos principais objetivos tratados

na literatura para o PRVJT. Este estudo permite que, abordagens que envolvem muitos objetivos para o problema tratado possam ser remodeladas, seja eliminando objetivos ou mudando fatores do mundo real.

- Disponibilizar modelos matemáticos reduzidos que representam uma generalização do PRVJT com muitos objetivos. Estes modelos possibilitam a aplicação da maioria dos Algoritmos Evolucionários Multiobjetivo (*multiobjective evolutionary algorithms* - MOEAs) ao problema, sem que haja perda significativa na qualidade das soluções das frentes de Pareto aproximada.
- Produzir conhecimento sobre MOEAs, bem como adequações de operadores genéticos aplicados ao PRVJT.
- Disponibilizar uma análise sobre ganhos e perdas de se reduzir o PRVJT proposto, descrevendo assim, se a redução de objetivos é viável, quais as melhores possibilidades de agregação, quais os motivos que levam à agregação de objetivos, quais motivos que levam à não agregação de objetivos, entre outros.

1.4 Organização do trabalho

Dado que o trabalho aborda muitos objetivos no problema de roteamento de veículos com janelas de tempo, o Capítulo 2 apresenta uma revisão da literatura sobre o Problema de Roteamento de Veículos, bem como sobre problemas com muitos objetivos.

O Capítulo 3 apresenta as principais definições sobre os problemas de roteamento de veículos capacitados, conceituando também outras variações do problema. O problema proposto neste trabalho é ainda descrito no Capítulo 3. Assim, o problema de roteamento de veículos proposto é definido formalmente e a formulação matemática do problema é apresentada. O problema abordado é então denominado de Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis (MOPRV).

O Capítulo 4 apresenta a metodologia utilizada para resolver o MOPRV. Neste capítulo os algoritmos NSGA-II e NSGA-III são descritos e o processo padrão realizado por ambos algoritmos é apresentado. Este capítulo ainda descreve o processo realizado pelas Árvores de Agregação para que o conflito e a harmonia entre os objetivos do problema sejam verificados.

As adequações e representações que foram necessárias para a execução dos NSGAs

para o problema proposto são descritas no Capítulo 5. Assim, este capítulo apresenta a representação dos indivíduos de uma população, bem como os métodos de cruzamento, mutação e seleção utilizados pelos NSGAs.

Os resultados são apresentados no Capítulo 6. Este capítulo apresenta os resultados retornados pelo NSGA-III para o MOPRV Completo. Ainda assim, uma análise da harmonia apresentada pelas Árvores de Agregação foi realizada e uma discussão relata o comportamento de cada objetivo do problema abordado. Deste modo, uma formulação reduzida do MOPRV (MOPRV Reduzido), que considera a agregação dos objetivos mais harmoniosos, é descrita e os resultados retornados pelo NSGA-II para otimizar o problema reduzido são apresentados. Por fim, uma comparação entre as soluções retornadas pelo NSGA-III e pelo NSGA-II é descrita neste capítulo.

A conclusão em torno do que foi desenvolvido, bem como as propostas para trabalhos futuros são discutidas no Capítulo 7.

Capítulo 2

Revisão Bibliográfica

Esta dissertação recai sobre o campo da otimização de problemas com muitos objetivos (*Many-Objective Problems - MOP*) que propõe técnicas e métodos para solucionar problemas de roteamento de veículos. Assim, este capítulo apresenta diversos trabalhos que fazem referência aos problemas de roteamento de veículos. Os principais objetivos acerca do problema, bem como abordagens que possuem características multiobjetivo do mesmo também são apresentadas. Além disso, uma revisão bibliográfica é feita sobre otimização multiobjetivo. Neste sentido, uma revisão da literatura sobre diversos algoritmos evolucionários multiobjetivo é descrita. Entretanto, quando grande parte destes algoritmos não é capaz de resolver problemas de otimização devido à grande quantidade de objetivos, uma nova classe de problemas é estudada. A esta classe se dá o nome de Problemas de Otimização com Muitos Objetivos.

A Seção 2.1 descreve os principais trabalhos da literatura relacionados com o problema abordado nesta dissertação. A Seção 2.2 apresenta os conceitos básicos sobre problemas multiobjetivo, bem como os principais algoritmos evolucionários multiobjetivo descritos na literatura. A Seção 2.3 apresenta as principais ideias para resolver problemas que envolvem muitos objetivos. Por fim, a Seção 2.4 conclui o capítulo.

2.1 Problema de Roteamento de Veículos

O problema de roteamento de veículos é um dos principais problemas de ampla aplicação prática em logística, pois abrange desde a produção de mercadorias até a entrega aos clientes. Dantzig e Ramser (1959) foram os primeiros a formular o PRV, no qual pro-

curavam resolver um problema real de distribuição de combustível. Desde então, o problema é abordado de diferentes maneiras e diversas metodologias já foram propostas para resolvê-lo. Os trabalhos de Brown e Graves (1981), Fisher et al. (1982), Bell et al. (1983), Evans e Norback (1985) e Golden e Wasil (1987) descrevem aplicações do PRV em indústrias de petróleo, químicas, alimentícias e de bebidas. As subseções a seguir apresentam uma revisão dos principais objetivos tratados na literatura em diversas variações do problema de roteamento de veículos.

2.1.1 Distância Total Percorrida

Grande parte dos trabalhos da literatura, que envolvem o PRV, procuram minimizar os custos com transporte. Estes custos estão diretamente ligados à distância total percorrida pela frota de veículos (custos variáveis). Deste modo, minimizar este objetivo tem sido uma das metas mais abordadas no PRV clássico, bem como em suas variantes.

Achuthan et al. (2003), por exemplo, procuraram resolver o problema de roteamento de veículos capacitados no qual a única meta avaliada pelos autores considera a minimização da distância total percorrida pelos veículos. Os autores propuseram um método exato em que novos planos de corte baseados em um algoritmo *branch-and-cut* eram utilizados para resolver o problema.

Dell'Amico et al. (2006) propõem uma abordagem exata para solução do problema de roteamento de veículos com coleta e entrega simultânea. Neste caso, o consumidor deve ser visitado uma única vez, sendo que as demandas de coleta e entrega devem ser atendidas simultaneamente. Neste trabalho, os autores apresentam um modelo matemático que considera a minimização da distância total percorrida pela frota de veículos, e em seguida, utilizam o método *branch-and-price* para solucionar o problema.

Mirabi et al. (2010) propuseram abordagens híbridas para o problema de roteamento de veículos com múltiplos depósitos. O único objetivo considerado por Mirabi et al. (2010) foi a minimização da distância percorrida pelos veículos. Para resolver o problema os autores combinam heurísticas construtivas e de melhoramento.

Yu et al. (2011) introduzem como objetivo a minimização da distância total percorrida pelos veículos ao PRVJT. Nesta abordagem do problema, além das restrições de janelas de tempo, restrições de tempo máximo da rota são consideradas. Esta restrição garante que os veículos não ultrapassem um tempo máximo de viagem pré-estabelecido. Assim, para resolver o problema, os autores propõem uma abordagem híbrida que con-

siste em um algoritmo de colônia de formigas e busca tabu.

2.1.2 Número de Veículos

No problema de roteamento de veículos cada veículo possui uma capacidade de transporte. Assim, algumas abordagens tratam o problema considerando custos de aquisição e manutenção (custos fixos) destes mesmos veículos. Diante disso, os PRVs podem considerar objetivos relacionados aos custos totais de roteamento (somando-se custos fixos de aquisição e manutenção dos veículos) e custos variáveis (proporcional à distância percorrida por cada veículo), como também pode se encontrar o número mínimo de veículos que execute o roteamento. Quando objetivos relacionados ao tamanho da frota de veículos é considerado, uma variante do PRV é tratada. Na maioria dos casos o número de veículos é igual ao número de rotas.

Ombuki et al. (2002) procuram resolver o problema de roteamento de veículos com janelas de tempo. A função objetivo adotada considera a minimização da soma ponderada dos custos de transporte. Neste caso, o número de veículos utilizados e a distância percorrida para atender a demanda de todos os consumidores são avaliadas. Os autores apresentam uma técnica de pesquisa híbrida baseada em metaheurísticas para resolver o problema. A abordagem é baseada em duas fases: uma fase de agrupamento global de clientes com base em Algoritmos Genéticos (AG) (*Genetic Algorithm* - GA) e uma técnica de pós-otimização com base em Busca Tabu.

Wassan e Osman (2002) implementaram a metaheurística Busca Tabu para o problema de roteamento de veículos capacitados com frota heterogênea. Este problema é uma variante do PRV clássico, no qual os veículos da frota possuem capacidade de carga distintas. Assim, o objetivo é determinar a composição ótima da frota e os roteiros de entrega de forma a minimizar a soma ponderada da quantidade de veículos e da distância total percorrida. O problema foi resolvido por quatro mecanismos de vizinhança baseados na troca entre operadores.

Sousa e Silveira (2011) apresentam uma versão mono-objetivo do problema de roteamento de veículos com janelas de tempo. Neste trabalho os autores introduzem uma função objetivo que considera a minimização da soma ponderada de dois objetivos principais: o primeiro verifica o número de veículos utilizados na distribuição de mercadorias e o segundo calcula o tempo de viagem dos veículos. O modelo proposto foi resolvido utilizando o software GLPK.

2.1.3 Violação da Restrição de Tempo

Em alguns casos da literatura a formulação do problema de roteamento de veículos com restrições de tempo permite o atendimento aos consumidores fora do intervalo de tempo dos mesmos. Estas generalizações do problema geralmente transformam as restrições referentes à janela de tempo em um objetivo que minimiza o número de restrições violadas, o tempo total de espera dos veículos nos clientes devido à precocidade ou atraso na chegada, ou ambos objetivos ao mesmo tempo. A penalidade de chegada após o fechamento da janela de tempo envolve a satisfação do cliente e reputação da empresa, onde o serviço insatisfeito pode trazer diversos prejuízos para a distribuidora.

Taillard et al. (1997) abordam uma versão do PRVJT em que as restrições de janelas de tempo são flexíveis. Nesta abordagem os veículos podem atender os clientes após o tempo de fim da janela de tempo, porém, caso chegue antes do início da janela de tempo, este deve esperar pela abertura da janela para iniciar o serviço. Como objetivo Taillard et al. (1997) consideram a minimização da soma ponderada entre os custos de transporte e a violação da restrição de tempo. A estratégia de punição para os veículos que violam a restrição de tempo utiliza um coeficiente de penalidade de atraso associado a cada consumidor. Os autores utilizam uma Busca Tabu para resolução do problema proposto.

Rahoual et al. (2001) propõem uma modelagem do PRVJT que, além das restrições de janelas de tempo, possui restrições de distância máxima da rota e tempo máximo de tráfego dos veículos. Assim, esta abordagem transforma a maior parte das restrições em objetivos do problema, sendo eles: minimizar a distância trafegada pelos veículos, minimizar o número de veículos, minimizar o número de violações da restrição de distância máxima, minimizar o número de violações da restrição de capacidade máxima do veículo, minimizar o número de violações referente a restrição de tempo máximo de tráfego dos veículos, minimizar o número de violações referente a restrição de janela de tempo. O método proposto para solucionar o problema pondera a função objetivo e aplica um algoritmo genético mono-objetivo. Outro método proposto aplica um algoritmo genético multiobjetivo.

Beham (2007) apresenta uma formulação semelhante a de Taillard et al. (1997). Assim, caso um veículo chegue antes do tempo de início da janela de tempo, este tem que esperar a abertura da janela para iniciar o atendimento. Em contrapartida, é permitido que os veículos atendam os clientes após o final da janela de tempo. Neste caso, uma penalidade proporcional ao atraso é contabilizada na função objetivo para violação do

tempo de atendimento, ou seja, a abordagem proposta por Beham (2007) penaliza a violação da restrição contabilizando a diferença entre o tempo de chegada do veículo no consumidor e o tempo de fim da janela de tempo. Duas funções objetivo são propostas, a primeira considera a minimização dos custos de transporte e a segunda o grau de violação da janela de tempo. O autor utiliza uma versão multiobjetivo do algoritmo de busca tabu para resolver o problema. Nesta versão, além da lista tabu, duas outras memórias são utilizadas. Uma delas armazena as soluções não-dominadas de vizinhanças passadas e a outra armazena as soluções não-dominadas localizadas no decorrer de todo procedimento.

2.1.4 Tempo de Espera dos Veículos

Grande parte dos trabalhos presentes na literatura que abordam o PRVJT consideram minimizar o tempo total de atendimento aos clientes (soma do tempo de viagem entre os consumidores e o tempo de espera dos veículos nestes mesmos consumidores). Ainda assim, muitos destes trabalhos estimam que o tempo de viagem entre dois consumidores é proporcional a distância percorrida entre os mesmos. Diante disso, é possível notar que minimizar o tempo total de atendimento aos clientes é um objetivo composto formado pela minimização da distância total percorrida pela frota de veículos e a minimização do tempo de espera dos veículos nos consumidores.

Deste modo, quando o problema envolve restrições de tempo, o custo total de encaminhamento e agendamento não incluem apenas a distância e o tempo total de viagem, mas também o custo do tempo incorrido na espera quando um veículo chega muito cedo em um determinado cliente. Esta quantidade de tempo de espera é significativa na prática. Primeiramente, o tempo de espera implica tempo sem fins lucrativos do distribuidor de logística e recursos da empresa que são subutilizados. Passar muito tempo na espera não só afeta a perda de oportunidade de gerar mais lucros, mas também incorre em custos adicionais, tais como custo do veículo/trabalho operacional, custo de manutenção, e taxa de estacionamento. Em segundo lugar, pode contribuir para problemas de trânsito e problemas ambientais, tais como o congestionamento do tráfego devido à espera em lugar inadequado, e a poluição do ar, se o veículo aguarda com o motor ligado.

Baràn e Schaerer (2003) apresentam uma formulação do PRVJT, no qual a minimização do número de veículos, tempo total de viagem e tempo total de entrega são considerados. Uma particularidade desta abordagem é que o tempo total de viagem considera apenas o tempo de percurso entre os clientes, desconsiderando o tempo de

ociosidade dos veículos nos consumidores. Já o tempo total de entrega de determinada demanda é dado pelo tempo para percorrer a rota até o ponto de entrega da respectiva demanda (consumidor) acrescido do tempo de espera do veículo no consumidor caso o veículo chegue antes do início da janela de tempo determinada. Nesta abordagem do problema, o último objetivo acaba por considerar, de forma indireta, apenas a minimização do tempo de espera dos veículos nos consumidores. Os autores utilizam um sistema múltiplo de colônia de formigas para resolver o problema.

Li e Lim (2003) propõem uma modelagem do PRVJT que considera a minimização do número de veículos utilizados, o custo de viagem, o tempo de programação das entregas e o tempo total de espera dos motoristas para iniciar o serviço. Uma metaheurística baseada em *Tabu-embedded Simulated Annealing* (TSA) é aplicada para resolver o problema.

Bhusiri et al. (2014) abordam uma versão do PRV com janelas de tempo flexíveis. Nesta abordagem do problema os veículos têm permissão para atender os consumidores após o fim da janela de tempo. Assim, os autores procuram minimizar o tempo total de espera de todos os veículos e o atraso no atendimento aos consumidores. Para o caso do atendimento após o fim da janela de tempo uma penalidade é considerada, podendo ser um valor negociado com cada consumidor ou multa definida pela própria empresa. Uma abordagem *Branch-and-price* é empregada para resolver o problema proposto.

2.1.5 Maior Rota (Makespan)

Diversos autores propõem formulações do PRV em que a minimização da maior rota é considerada como meta. Assim, procuram-se soluções para o problema em que o comprimento de viagem da maior rota seja a menor possível. Este objetivo foi motivado por situações em que devido às grandes distâncias entre os locais de coleta de passageiros, as rotas de ônibus tendiam a ser longas e nunca completadas. Neste sentido, minimizar a maior rota garante alguma equidade em termos de tempo gasto no ônibus pelo primeiro passageiro em comparação com o tempo gasto pelo último passageiro. Minimizar a maior rota também pode envolver custos de operação e satisfação do consumidor. Assim, quanto menor é o percurso da maior rota, menor é o tempo em que o depósito central necessita ficar aberto e, menor é a espera do último cliente a ser atendido.

Um problema de roteamento de veículos aberto com serviços de coleta e entrega de passageiros é proposto por Corberan et al. (2002). Os autores apresentam uma mode-

lagem composta por dois objetivos conflitantes: minimização do número de veículos e minimização do tempo máximo que um passageiro trafega no interior do veículo. O segundo objetivo pode ser calculado pelo tamanho da rota com maior distância percorrida pelo veículo.

Os trabalhos de Lacomme et al. (2003, 2006) consideram uma versão do PRVC multiobjetivo, onde os objetivos considerados são minimizar o custo total de transporte e minimizar o *makespan*, ou seja, o comprimento de rota mais longo. Os autores apresentam um algoritmo genético para resolver o PRVC multiobjetivo. Murata e Itai (2005) apresentam uma abordagem semelhante as de Lacomme et al. (2003, 2006) para o PRVC, nesta abordagem os objetivos referem-se à minimização do número de veículos utilizados e a minimização da maior rota. No entanto, a maior rota é aquela em que o custo do veículo para concluí-la é o maior dentre todas as rotas. Os autores utilizam algoritmos de otimização multiobjetivo para resolver o problema proposto.

Karabulut e Tasgetiren (2014) tratam uma modelagem do PRVJT que considera o relaxamento das restrições de capacidade dos veículos. De forma geral, os autores consideram duas funções objetivo para o problema. A primeira função considera a minimização da soma dos custos de transporte e a segunda é utilizada para minimizar o tempo de conclusão das rotas, ou seja, o *makespan*. Os autores propõem um algoritmo guloso iterativo para resolução do problema.

2.1.6 Balanceamento de Rota

Alguns objetivos são projetados para equilibrar as disparidades entre percursos de rotas. Tais objetivos são frequentemente introduzidos com a finalidade de implementar elementos de equidade nos problemas de roteamento de veículos. Para definir este objetivo é necessário considerar a carga de trabalho nas rotas que podem ser expressadas como o número de cliente visitados, a quantidade de produtos entregues e o tempo necessário ou comprimento das rotas, por exemplo.

Uma das questões que motivam o estudo deste problema refere-se a satisfação dos motoristas, uma vez que a carga de trabalho destes deve ser próxima. Assim, Lee e Ueng (1998) incorporam o equilíbrio em uma versão do PRV para melhorar a equidade de trabalho entre os motoristas. A carga de trabalho das rotas considera o tempo de viagem necessário para completar o percurso da rota e o tempo necessário para carregar e descarregar o caminhão. Deste modo, o objetivo é modelado como a minimização da

soma das diferenças entre a carga de trabalho de cada rota e a menor carga de trabalho entre todas as rotas. No entanto, um modelo de programação inteira, juntamente com uma heurística são propostas para solucionar o problema.

El-Sherbeny (2001) apresenta um caso em que uma empresa de transporte belga considerou minimizar a diferença entre a carga de trabalho da rota mais longa e a carga de trabalho da rota mais curta (balanceamento de rota). O problema abordado envolve restrições de janela de tempo, bem como pontos de coleta e entrega de mercadorias. Neste caso, a carga de trabalho das rotas pode ser considerada como o tempo necessário para viajar entre os diferentes pontos de coleta e entrega, mais o tempo necessário para carregar e descarregar os caminhões, mais o tempo de espera dos veículos nos consumidores. Além do objetivo que considera a otimização do equilíbrio entre as rotas, outros objetivos são considerados no problema abordado. Assim, El-Sherbeny (2001) propõe uma abordagem multiobjetivo do algoritmo *Simulated Annealing* para gerar uma aproximação do conjunto de soluções eficientes.

O equilíbrio entre as rotas também é considerado como meta para Baños et al. (2013). Os autores propõem um modelo multiobjetivo do PRV com restrições de tempo que considera a minimização da distância total percorrida e o desequilíbrio das rotas. Este desequilíbrio é analisado a partir de duas perspectivas: o desequilíbrio nas distâncias percorridas pelos veículos, e o desequilíbrio nas cargas entregues pelos veículos. Esta formulação permite a obtenção de soluções com diferentes distâncias e desequilíbrios de rota, de modo que o tomador de decisão possa decidir, com base em soluções específicas, qual das soluções é a mais adequada de acordo com determinados critérios. Um procedimento multiobjetivo com base em *Simulated Annealing* foi proposto para resolução do problema abordado.

2.1.7 Problema de Roteamento de Veículos Multiobjetivo

Vários trabalhos que analisam o PRV apresentam um único objetivo referente a minimização dos custos de transporte, na maior parte dos casos a minimização da distância total percorrida pela frota de veículos é considerada. Estas abordagens não consideram o fato de que muitos problemas logísticos têm natureza multiobjetivo. Grande parte das empresas têm interesse em minimizar o atraso das entregas, minimizar o tempo total de transporte, minimizar a distância total percorrida, minimizar o tempo de espera dos veículos, minimizar a quantidade de veículos, maximizar o benefício e equilibrar a utilização dos recursos. Logo, o tomador de decisão, em diversas situações reais, neces-

sita avaliar estes outros aspectos. Diante da importância do tratamento de problemas multiobjetivo, bem como do alto contexto prático que envolve os PRVs multiobjetivo, houve um aumento no número de trabalhos que tratam esta classe de problemas. Assim, existe um grande número de possibilidades de formulação que envolvem os PRVs multiobjetivo, seja introduzindo objetivos no problema ou formulando novas restrições para o mesmo.

Segundo Jozefowicz et al. (2008), as principais motivações para tratar problemas de roteamento multiobjetivo são:

- Estender problemas clássicos acadêmicos: quando a definição do problema permanece inalterada e novos objetivos são adicionados.
- Generalizar problemas clássicos: quando uma ou mais restrições e/ou parâmetros são trocados por funções objetivo.
- Estudar casos reais: quando os objetivos foram claramente identificados pelo tomador de decisões e são dedicados a um problema real muito específico.

Bowerman et al. (1995) apresenta uma abordagem multiobjetivo do problema de roteamento de ônibus escolar urbano (município de Wellington, Ontário). Os autores consideraram os seguintes objetivos: minimizar o número de veículos, minimizar a distância total percorrida pelos veículos, minimizar o desequilíbrio das rotas, minimizar a distância percorrida pelos alunos de suas residências aos pontos de ônibus e minimizar o desequilíbrio de carga. O último objetivo se relaciona com a variação do número de alunos transportados ao longo de cada rota. Os autores utilizam heurísticas do tipo Dividir e Rotear para solucionar o problema.

Geiger (2003) considera uma abordagem multiobjetivo do PRVJT. No problema proposto os autores transformam as restrições referentes à janela de tempo em um objetivo. Deste modo, o problema considera a minimização da distância total trafegada, número de veículos, grau de violação da janela de tempo e número de violações da janela de tempo. O autor faz uso de um algoritmo genético para solucionar o problema proposto.

Meng et al. (2005) apresentam um problema de localização e roteamento para estudar casos reais de transporte de produtos perigosos. Os objetivos do problema consistem em minimizar os custos de transporte e propiciar maior segurança no transporte do produto. Alguns objetivos específicos são utilizados para garantir a segurança no transporte, estes

são: minimizar o risco total de acidentes e minimizar o total da população exposta ao produto.

Uma proposta do problema de roteamento de veículos com demanda estocástica cujos objetivos são minimizar a distância total percorrida pelos veículos, minimizar o número de veículos e minimizar a remuneração dos motoristas é proposta por Tan et al. (2007). A remuneração dos motoristas é calculada a partir do tempo gasto para percorrer a rota. Os autores propuseram um algoritmo evolutivo multiobjetivo para solucionar o problema.

Chiang (2008) apresenta um modelo matemático multiobjetivo para uma versão do Problema de Roteamento de Veículos Periódico (PRVP) (*Periodic Vehicle Routing Problems - PVRP*). Neste problema um conjunto de clientes deve ser visitado uma ou mais vezes em um horizonte de planejamento de T dias. Nesta abordagem, três objetivos conflitantes são considerados: minimização da distância total percorrida pelos veículos, minimização da variação das cargas dos veículos e minimização do número de entregas divididas. As entregas divididas são aquelas que não puderam ser atendidas por um único veículo, em uma única visita. Um algoritmo genético multiobjetivo foi proposto para resolução do problema.

F. Doerner e F. Hartl (2008) estuda uma versão do problema de cobertura de trajeto (*Covering Tour Problem*). Os autores apresentam um estudo de caso real do problema de roteamento de unidades de saúde móveis da região de Thiés no Senegal. Os objetivos tratados no problema procuram aumentar a eficiência da implantação da força de trabalho medida pela razão entre o tempo gasto em procedimentos médicos e tempo gasto pela viagem acrescido do tempo de instalação da facilidade móvel, melhorar a acessibilidade média dada pela distância média que os habitantes precisam caminhar até alcançar uma facilidade, aumentar a cobertura medida pela porcentagem de habitantes que residem dentro de um raio de abrangência do ponto de atendimento. Os autores adotam três métodos para resolver o problema, *Vector Evaluated Genetic Algorithm* (VEGA), algoritmo genético multiobjetivo e colônia de formiga.

2.2 Otimização Multiobjetivo

Grande parte dos problemas reais encontrados na área de otimização envolve a obtenção de diversas metas (objetivos) que devem ser atingidas simultaneamente. Estes problemas

são chamados de problemas de otimização vetorial (POV) ou problemas de otimização multiobjetivo (POM). Os objetivos para esta classe de problemas frequentemente são combinados em uma única função objetivo que reflete a utilidade de cada objetivo, enquanto alguns objetivos são transformados em restrições. Porém, em Otimização Multiobjetivo, os objetivos de um problema são tratados separadamente como objetivos não comparáveis (Coello et al., 2007; Deb, 2009; Fonseca e Fleming, 1993).

Visto que a maioria dos objetivos colocados em questão são conflitantes, os problemas de otimização multiobjetivo geralmente não possuem uma única solução que otimize todas suas metas. Portanto, esta classe de problemas tem como resultado um conjunto de soluções eficientes. Estas soluções são ditas não dominadas e compõem o conjunto de Pareto-ótimo. Para compreender o que são estas soluções não dominadas, é necessário apresentar algumas definições.

Diante deste contexto, esta seção apresenta os conceitos básicos sobre problemas multiobjetivo, bem como suas principais características. Os principais Algoritmos Evolucionários Multiobjetivo também são descritos nesta seção.

2.2.1 Conceitos Básicos

Os problemas de otimização multiobjetivo podem ser representados da seguinte forma:

$$(POV) = \begin{cases} X^* = x^* \in \mathbb{R}^n \\ x^* = \arg \min_x f(x) \\ \text{sujeito a: } x \in F_x \end{cases}$$

O vetor de objetivos e a região factível são definidos por $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ e $F_x \subset \mathbb{R}^n$, respectivamente. No entanto a região factível é composta por soluções que satisfazem restrições de desigualdade e restrições de igualdade. Os vetores $x \in \mathbb{R}^n$ são chamados de vetores de parâmetros ou vetores de decisão do problema de otimização vetorial. Estes vetores compõem o espaço de parâmetros X . Já os vetores $f(x) \in \mathbb{R}^m$ encontram-se no espaço de objetivos, ou seja, o espaço no qual se representa a imagem da função $f(x)$ que é denotada por Y . Considerando um POV, deseja-se portanto, determinar um conjunto de soluções, ou Pareto-Ótimo, que otimize o vetor objetivo $f(x)$. Entretanto, a dificuldade é definir o que é essa otimização, ou seja, determinar quais soluções fazem parte do conjunto X^* (Takahashi et al., 2012). O conceito de Dominância de Pareto é

considerado e apresentado na Definição 1. A Figura 2.1 ilustra o mapeamento do espaço de parâmetros X no espaço de objetivo Y feito pela função $f(x)$.

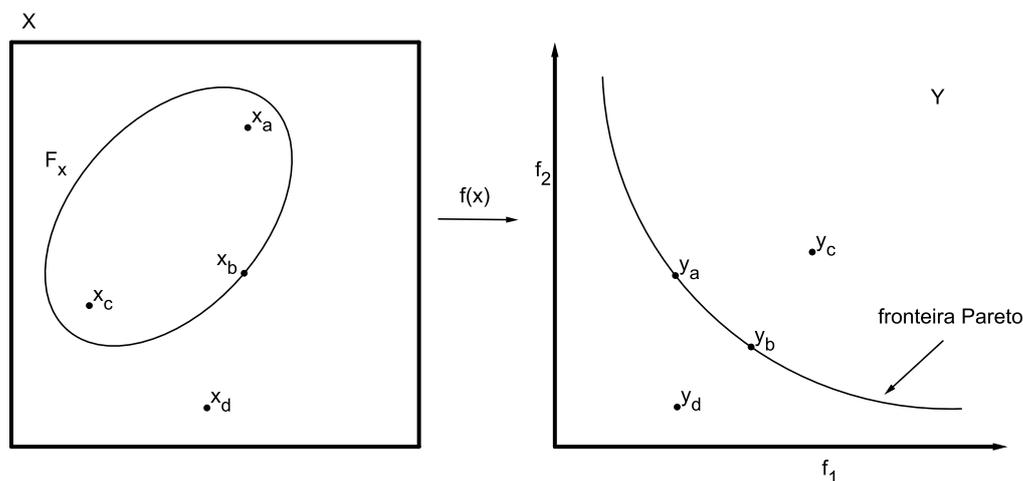


Figura 2.1: Mapeamento do espaço de parâmetros X no espaço de objetivo Y feito pela função $f(x)$

É possível observar na Figura 2.1 que a região factível no espaço de parâmetros X é designada por F_x . Cada ponto \mathbf{x}_i no espaço de parâmetros tem seu respectivo ponto \mathbf{y}_i no espaço de objetivos. Assim, as soluções \mathbf{x}_a e \mathbf{x}_b apresentam os melhores valores na fronteira Pareto aproximada para os objetivos f_2 e f_1 , respectivamente. Apesar da solução \mathbf{x}_d apresentar valores melhores que \mathbf{x}_a e \mathbf{x}_b no espaço de objetivos, esta solução não é considerada por não pertencer ao conjunto de soluções viáveis.

Definição 1 (Dominância). *Uma solução $x' \in X$ domina outra solução $x'' \in X$ se $f(x') \leq f(x'')$ e $f(x') \neq f(x'')$. Da mesma forma, diz-se que $f(x') \in Y$ domina $f(x'') \in Y$, nessas mesmas condições. Esta relação de dominância é representada pela notação $f(x') \preceq f(x'')$. Assim, se uma solução $x' \in X$ é estritamente melhor que a solução x'' para todos os objetivos, pode-se dizer que x' domina fortemente x'' . Esta relação pode ser representada pela notação $f(x') \prec f(x'')$.*

Entretanto, existe a possibilidade de duas soluções serem incomparáveis, não possuindo nenhuma relação de dominância. Este conceito é dado pela Definição 2.

Definição 2 (Soluções incomparáveis). *Supondo $x' \in X$ e $x'' \in X$, diz-se que estes pontos são não-dominados entre si, ou incomparáveis entre si, caso as relações de dominância $f(x') \preceq f(x'')$ e $f(x'') \preceq f(x')$ não sejam verificadas.*

Desta forma, um vetor x^* é Pareto-Ótimo se não existir nenhum outro vetor viável x que melhore um objetivo, sem que haja piora em algum outro objetivo. Em outras palavras, uma solução x^* pertence ao conjunto Pareto-ótimo se não existir nenhuma outra solução x que domine x^* . Portanto, a solução de um POV é um conjunto Pareto-Ótimo $X^* \subset X$. (Takahashi et al., 2012). A Definição 3 e a Definição 4 descrevem o conceito de solução Pareto-ótima e conjunto Pareto-Ótimo, respectivamente.

Definição 3 (Solução Pareto-Ótima). $x^* \in F_x$ é uma solução Pareto-Ótima se não existe qualquer outra solução $x \in F_x$ tal que $f(x) \preceq f(x^*)$, ou seja, se x^* não é dominado por nenhum outro ponto factível.

Definição 4 (Conjunto Pareto-Ótimo). $X^* \subset X$ é um conjunto Pareto-Ótimo se todas as soluções que o compõem são soluções Pareto-Ótimas. O conjunto-imagem $Y^* \subset Y$ associado ao conjunto Pareto-Ótimo é chamado de fronteira Pareto-Ótima.

Encontrar o conjunto Pareto-Ótimo em problemas de otimização multiobjetivo pode ser computacionalmente intratável (Deb, 2001), fazendo-se necessário o uso de métodos heurísticos, que tentam encontrar um conjunto de soluções aproximadas, composto por soluções não dominadas entre si. Por outro lado, é possível estabelecer a definição de um conjunto Pareto-Ótimo num sentido local. Assim, como na otimização escalar (otimização mono-objetivo), a otimização vetorial frequentemente trabalha com soluções apenas locais. Isto porque, encontrar a globalidade das soluções pode ser uma tarefa complexa, além de não ser necessária em diversos casos práticos. As soluções que compõem o conjunto aproximado de Pareto-Ótimo são denominadas de soluções localmente Pareto-Ótima (fronteira Pareto aproximada). Este conceito é apresentado na Definição 5.

Definição 5 (Solução Localmente Pareto-Ótima). Dada uma vizinhança N , diz-se que $x \in F_x$ é uma solução localmente Pareto-Ótima se, e somente se, nenhuma outra solução na vizinhança N domine x .

Definição 6 (Estrutura de Vizinhança). Dada uma solução $x \in F_x$, uma estrutura de vizinhança $N(x)$ gera um conjunto de soluções viáveis, denominadas vizinhos da solução x (Lust, 2010).

2.2.2 Algoritmos Evolucionários Multiobjetivo

A proposta de utilização de Algoritmos Evolucionários Multiobjetivo (*Multi-objective Evolutionary Algorithms* - MOEA) para a solução de problemas multiobjetivo teve início

no final dos anos 60 quando Rosenberg (1967) indicou a possibilidade de usar algoritmos genéticos para essa classe de problemas. Entretanto, o estudo de Rosenberg (1967) não abordou problemas multiobjetivo diretamente, indicando apenas essa possibilidade. Assim, a primeira tentativa real de estender um algoritmo evolucionário para resolver problemas multiobjetivo é o *Vetor Evaluated Genetic Algorithm* (VEGA) desenvolvido por Schaffer (1984). Entretanto, pouca atenção foi dada a este algoritmo devido à convergência prematura de soluções específicas, além de não conseguir manter uma diversidade desejada do conjunto de soluções não dominadas (Deb, 2008).

Segundo Deb (2008) os MOEAs passaram a ter uma maior atenção no final dos anos 80, quando Goldberg (1989) propôs um método revolucionário de classificação por não dominância (*Non-dominated Sorting*) para geração dos descendentes. Na década de 90 diversos MOEAs foram propostos para resolver problemas multiobjetivo, inicialmente baseados em algoritmos genéticos. Assim, a primeira geração de MOEAs (Coello et al., 2007) utilizavam o conceito de classificação baseada em dominância Pareto e comparação para avaliação de aptidão. Desde então, MOEAs têm sido largamente reconhecidos como métodos apropriados e bem sucedidos para resolver problemas multiobjetivo. Isto se deve ao fato de que os MOEAs utilizam uma população de soluções candidatas que garantem a convergência para soluções Pareto-ótimas. Esta evolução baseada em conjuntos oferece a possibilidade de busca por um conjunto de estimativas em apenas uma execução.

Um dos algoritmos evolucionários desta primeira geração que foi muito utilizado é o *Multi-Objective Genetic Algorithm* (MOGA), proposto por Fonseca e Fleming (1993). Este algoritmo classifica todas as soluções não dominadas com *ranking* 1 e as outras soluções são classificadas de acordo com o número de soluções que as dominam, sendo que a seleção é feita a partir das soluções com menor *ranking*. Entretanto, um algoritmo denominado de *Nondominated Sorting Genetic Algorithm* propôs classificar todas as soluções com valores de *ranking*. Segundo Deb (2008), estes algoritmos foram importantes para incentivar as pesquisas para os MOEAs, porém sua eficiência era extremamente baixa.

Deste modo, a segunda geração dos MOEAs produziu algoritmos que são mais eficientes do ponto de vista computacional. Estes algoritmos combinam mecanismos de preservação elitista com operadores de diversidade. Assim, o *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) proposto por Deb et al. (2002) trata alguns problemas do seu antecessor. O NSGA-II divide os indivíduos da população em fronteiras. Assim, as soluções não dominadas fazem parte da primeira fronteira, ou seja, a fronteira 1 possui soluções com os melhores valores de aptidão. A fronteira 2 possui as soluções com o

segundo melhor valor de aptidão, e assim por diante. Deste modo, este algoritmo seleciona as soluções das primeiras fronteiras para fazer parte da próxima população. Caso a última fronteira a ser inserida na população ultrapasse o número máximo de soluções permitidas, o NSGA-II escolhe as soluções mais distantes de seus vizinhos. Este processo garante uma maior diversidade das soluções na fronteira Pareto aproximada.

O *Strength Pareto Evolutionary Algorithm 2* (SPEA2), proposto por Zitzler et al. (2001), classifica as soluções não dominadas de forma diferente do NSGA-II, porém, este algoritmo também utiliza a ideia da divisão das soluções em fronteiras. Ele incorpora o número de indivíduos que uma solução domina e o número de indivíduos que dominam uma solução. Um operador de diversidade é utilizado para evitar que soluções muito próximas entre si sejam selecionadas para a geração seguinte. O *Niched Pareto Genetic Algorithm* (NPGA) (Horn et al., 1994), *Strength Pareto Evolutionary Algorithm* (SPEA) (Zitzler e Thiele, 1999) e *Pareto Envelope-based Selection Algorithm-II* (PESA-II) (Bentley e Corne, 2002) são exemplos de outros MOEAs encontrados na literatura.

Contudo, Deb e Jain (2014) apresentaram recentemente uma versão do NSGA-II que utiliza pontos de referência para guiar a busca quando todos os indivíduos são não dominados. Este algoritmo é denominado de *Non-dominated Sorting Genetic Algorithm III* (NSGA-III). O procedimento realizado pelo NSGA-III é semelhante ao do NSGA-II. Assim como seu antecessor o algoritmo utiliza dominância Pareto para classificar as soluções de uma população. Porém, quando existem soluções que pertencem ao mesmo nível de dominância o algoritmo utiliza pontos de referência no espaço de objetivos, de tal forma que as soluções mais próximas destes pontos são melhores classificadas. O NSGA-III, quando comparado ao NSGA-II, apresenta melhor comportamento para problemas que envolvem mais de 3 objetivos (Deb e Jain, 2014).

2.2.3 Visualização

A visualização de resultados de problemas multiobjetivo em duas dimensões (dois objetivos) pode ser feita através de um plano cartesiano com dois eixos como na Figura 2.2. Nesta representação cada solução é reproduzida como um ponto, no qual o número indica a fronteira deste mesmo ponto. Sendo assim, o número 1 representa as soluções da primeira fronteira, o número 2 as soluções da segunda fronteira, e assim por diante.

Por meio de uma simples extensão no número de eixos, pode-se representar problemas com três objetivos como na Figura 2.3. Neste caso, as soluções não dominadas também

têm uma dimensão adicional.

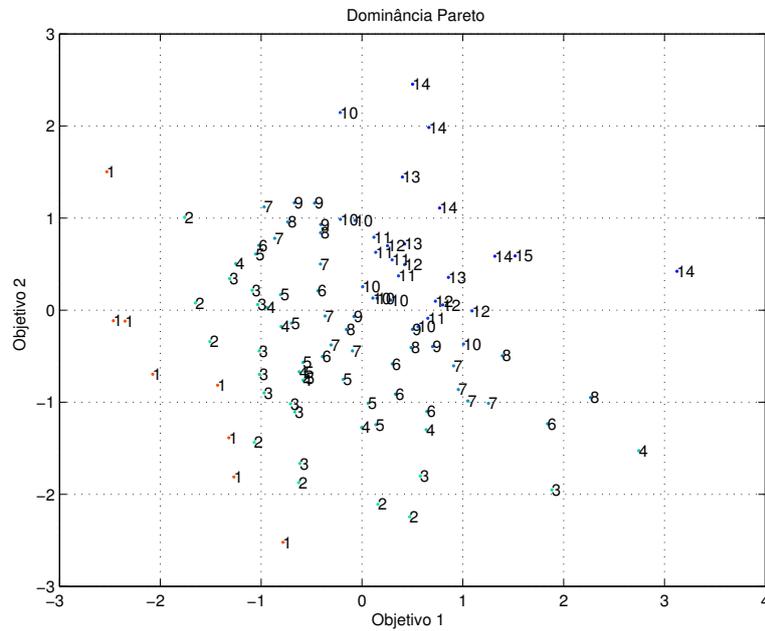


Figura 2.2: Visualização em 2 dimensões (Freitas et al., 2015)

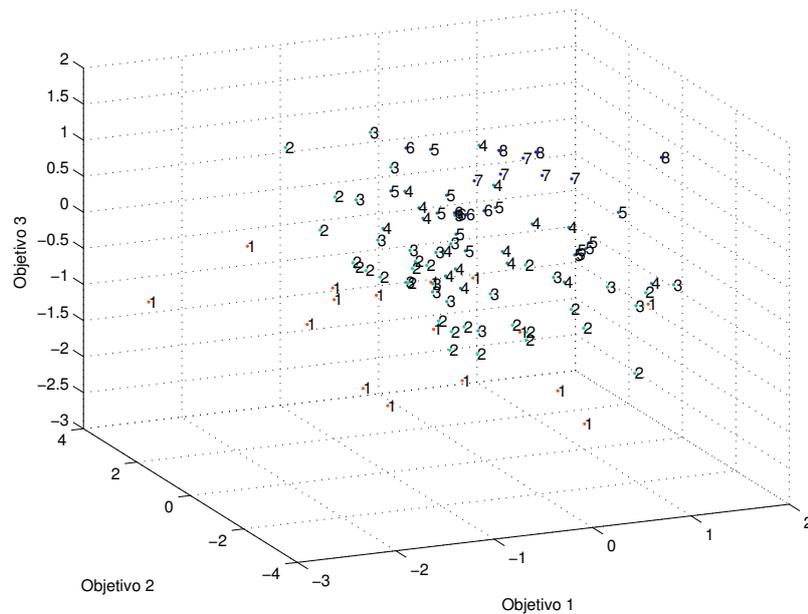


Figura 2.3: Visualização em 3 dimensões (Freitas et al., 2015)

Muitas vezes utilizado por razões estéticas, o gráfico tridimensional não melhora a

identificação de uma informação, ao contrario, ele faz a interpretação das informações ainda mais difícil devido ao efeito de perspectiva associado à tridimensionalidade. Assim, a interpretação da relação entre as soluções de uma população se torna mais difícil em três dimensões.

2.3 Otimização de Problemas com Muitos Objetivos

Os algoritmos evolucionários têm sido amplamente utilizados para resolver problemas multiobjetivo devido sua generalidade e flexibilidade em relação a características do problema. Todavia, pesquisas (Khare et al., 2003; Knowles e Corne, 2007) indicam que quanto maior o número de objetivos no problema, menor é o desempenho dos MOEAs. Isto por que o número de soluções necessárias para uma boa representação da aproximação da frente de Pareto cresce exponencialmente em função do número de objetivos, fazendo com que os algoritmos mais usuais geralmente não resolvam de maneira satisfatória problemas com mais de três objetivos.

Apesar da maioria dos estudos em Problemas de Otimização Multiobjetivo serem focados em problemas com dois ou três objetivos, é fácil perceber que problemas de otimização práticos envolvem um grande número destes critérios. No entanto, esforços de pesquisa foram orientados a investigar a escalabilidade dos MOEAs com respeito ao número de objetivos (Carvalho e Pozo, 2012; Ishibuchi et al., 2008) e foi observado que o desempenho destes algoritmos reduz dramaticamente. Este fato levou a um termo para melhor referenciar Problemas de Otimização Multiobjetivo com mais de três objetivos, definido como Problemas com Muitos Objetivos (*Many-Objective Problems*) (Ishibuchi et al., 2008). A área que estuda soluções para os problemas causados pelo aumento no número de objetivos é chamada de Otimização com Muitos Objetivos (*Many-Objective Optimization*).

Um dos principais problemas encontrados para a classe de problemas com muitos objetivos, segundo Ishibuchi et al. (2008), está relacionado ao aumento exponencial no número de soluções requeridas para aproximar toda a fronteira de Pareto-Ótimo. O objetivo dos MOEAs é encontrar um conjunto de soluções não dominadas que melhor se aproxime da fronteira de Pareto-Ótimo. Uma vez que esta fronteira é uma hipersuperfície no espaço de objetivos, o número de soluções requeridas para esta aproximação aumenta exponencialmente com o número de objetivos. A este problema se dá o nome de problema de escalabilidade.

Uma maneira de analisar o problema de escalabilidade é que se temos apenas um objetivo, precisaríamos de apenas uma solução para representar o melhor valor de função objetivo possível. Para 2 objetivos conflitantes, se queremos uma granularidade que representa apenas as soluções extremas, precisaríamos de 2 soluções da curva formada pela frente de Pareto: um com o melhor valor para cada um dos objetivos. Para 3 objetivos conflitantes, a frente de Pareto é então uma superfície e precisaríamos de 4 soluções para representar as soluções extremas. Da hipersuperfície formado por soluções em 4 objetivos, precisaríamos então de 8 soluções apenas para representar soluções extremas e assim por diante. Deste modo, mesmo no caso em que consideramos apenas as soluções extremas, que usualmente não são as mais apropriadas, precisamos de pelo menos 2^{M-1} soluções para representar este compromisso, onde M é o número de objetivos do problema.

Esta relação exponencial mostra a importância de se tratar otimização com muitos objetivos, com apenas soluções incomparáveis, como uma classe diferente de problemas. Nestas circunstâncias, é mais conveniente entender a relação entre os objetivos e facilitar a tomada de decisão, do que investir na tarefa não polinomial de se encontrar uma frente apropriada com as melhores soluções.

Entretanto, em problemas com muitos objetivos, menos soluções podem ser claramente reconhecidas como melhores que outras no espaço de objetivos, o que compromete a convergência dos MOEAs, já que eles dependem de comparações de dominância de Pareto. Khare et al. (2003) mostraram a baixa capacidade de escalabilidade dos métodos NSGA-II, *Pareto Envelope-based Selection Algorithm-II* (PESA) e *Strength Pareto Evolutionary Algorithm 2* (SPEA2) em funções de teste escaláveis. Hughes (2005) afirmou que métodos de agregação com multipartida desempenham melhor que MOEAs. Knowles e Corne (2007) ilustraram que a habilidade dos operadores de variação de produzir soluções que dominam seus pais diminui com um número crescente de objetivos. Assim, existem outros problemas que envolvem a otimização de problemas com muitos objetivos. Além do problema de escalabilidade, estes são:

- Perda de pressão seletiva pois a maior parte da população se torna não dominada, não existindo tendência seletiva para direcionar a busca em qualquer direção. (Battista et al., 2011; Garza-Fabre et al., 2011; Purshouse e Fleming, 2007)
- Alta dimensionalidade, levando a um custo computacional aumentado pois é necessário ter mais indivíduos na população para a representação da frente de Pareto

- Dificuldade em se visualizar compromissos entre objetivos em várias dimensões
- Dificuldade em visualizar as soluções. Geralmente se assume que a escolha de uma solução final de um conjunto de soluções não dominadas é feita por um tomador de decisões baseada em sua preferência. Com o aumento do número de objetivos, a visualização das soluções obtidas torna-se muito difícil. O que significa que a escolha de uma solução final é um processo difícil.

Diante deste contexto, diversas abordagens foram propostas na literatura para resolver problemas de Otimização com Muitos Objetivos, bem como para resolver a baixa capacidade de escalabilidade dos MOEAs para este tipo de problema. Estas são:

- Diferentes métodos de classificação de soluções (Drechsler et al., 2001; Knowles e Corne, 2007; Sato et al., 2009; Wang et al., 2007)
- Relações de dominância diferentes tais que mais soluções sejam dominadas a cada geração (Batista et al., 2011; Sato et al., 2009; Saxena et al., 2009)
- Outras medidas de aptidão envolvendo indicadores (Auger et al., 2012; Beume et al., 2007; Brockhoff e Zitzler, 2007; Saxena e Deb, 2007; Zitzler e Kunzli, 2004)
- Redução de dimensionalidade em objetivos redundantes (Brockhoff e Zitzler, 2006, 2007; Saxena e Deb, 2007; Singh e Singh, 2011)

Independentemente do número de objetivos, um problema com muitos objetivos é dado quando a maior parte da população se torna não dominada e então há perda de pressão seletiva, seja com 5, 10 ou 30 objetivos. Quando chegamos ao ponto onde as técnicas correntes não podem resolver o problema adequadamente, estamos no domínio de problemas com muitos objetivos. As subseções a seguir apresentam uma revisão das principais técnicas apresentadas na literatura para a resolução de problemas com muitos objetivos. Como descrito no capítulo anterior, a formulação de problemas com muitos objetivos é de fundamental importância para esta dissertação devido ao grande número de critérios que podem ser considerados nos problemas de roteamento de veículos.

2.3.1 Métodos de Classificação de Soluções

Algoritmos evolucionários multiobjetivo são frequentemente utilizados para resolver problemas que envolvem muitos objetivos. Um problema que surge quando se utiliza estes

algoritmos é o de avaliar as soluções de uma população de forma que estas possam ser comparadas. Neste sentido, é necessário determinar um *ranking* dos indivíduos da população para identificar quais soluções são melhores que outras. Entretanto, diferentes métodos de classificação de soluções são frequentemente utilizados para resolver problemas que envolvem muitos objetivos.

Esbensen e Kuh (1996) apresentam uma abordagem no qual o espaço de busca é dividido em satisfatório, aceitável e inválido. Nesta abordagem, o decisor tem que especificar os limites entre soluções satisfatórias, aceitáveis e inválidas. Para obter resultados de alta qualidade os limites devem ser adaptados durante a execução do método. Já Drechsler et al. (2001) propõem um método de classificação em que o espaço de busca é dividido em classes. Assim, o método proposto divide o espaço de busca em mais de três categorias, como por exemplo, superior, muito bom, bom, satisfatório e inválido. Em seguida as soluções são divididas em classes de acordo com sua qualidade. Cada classe é formada por soluções de mesma qualidade.

Kong et al. (2010) propõem um método de classificação média que utiliza técnicas de *Average Ranking* (AR) (Bentley e Wakefield, 1998). O método proposto considera cada objetivo de forma independente. Para um objetivo f_j , $j \in \{1, 2, \dots, m\}$, as soluções na população são classificadas de acordo com os valores de tal objetivo, e uma lista de classificação é construída onde cada solução tem a sua posição no *ranking*. Assim, cada solução x classificada pode ser representada pelo vetor $R(x) = \{r_1(x), r_2(x), \dots, r_m(x)\}$, onde $r_j(x)$ é a classificação da solução x para o objetivo f_j . Uma vez que $R(x)$ é calculado, uma única classificação é obtida para cada solução por combinar as posições de *ranking* dos m objetivos. Geralmente, soma-se os valores de *ranking* de cada objetivo para obter a classificação geral das soluções. Esta classificação é usada para refletir a qualidade global de cada solução comparada.

2.3.2 Relações de Dominância

Tratando-se de problemas com muitos objetivos, algoritmos evolucionários multiobjetivo apresentam problemas diretamente relacionados à baixa capacidade de comparações de dominância Pareto, ou seja, à medida que o número de objetivos de um problema cresce, menos soluções podem ser reconhecidas como melhores que outras no espaço de objetivos. Este fato compromete a convergência dos algoritmos evolucionários multiobjetivo. Para resolver este problema, alguns trabalhos na literatura propõem estratégias de solução que possibilitam que mais soluções possam ser dominadas a cada geração.

Sato et al. (2009) propõem um método para controlar a área de domínio de soluções a fim de induzir uma classificação adequada para melhorar a seleção e, por consequência, melhorar o desempenho dos MOEAs em problemas com muitos objetivos. Utilizando um parâmetro definido pelo usuário, o método proposto pode controlar o grau de expansão ou contração da área de domínio de soluções. Modificando a área de domínio de soluções, a relação de domínio também é alterada, induzindo um *ranking* de soluções que é diferente para o domínio convencional. Para analisar os efeitos do método proposto os autores procuram resolver uma versão multiobjetivo do problema da mochila.

Batista et al. (2011) utilizam o conceito de ϵ -dominância (Laumanns et al., 2002) para resolver problemas de convergência dos MOEAs. O conceito ϵ -dominância permite que o decisor forneça o valor de ϵ para controlar o tamanho do conjunto de soluções dominadas. No entanto, como as características geométricas da frente de Pareto são geralmente desconhecidas pelo decisor, a estratégia de dominância ϵ pode perder um número alto de soluções viáveis quando o valor de ϵ é mal avaliado. Diante disso, a principal dificuldade é calcular um valor apropriado para ϵ que forneça o número desejado de pontos não dominados. A fim de resolver algumas destas limitações, os autores propõem uma extensão do regime de ϵ -dominância, denominado de Cone ϵ -Dominância. Esta estratégia visa manter as boas propriedades de convergência de ϵ -dominância, proporcionando um melhor controle que seja menos sensível às características geométricas da frente de Pareto.

2.3.3 Medidas de Aptidão Envolvendo Indicadores

Alguns trabalhos da literatura utilizam indicadores de qualidade para resolução de problemas com muitos objetivos. Grande parte dos algoritmos baseados em indicadores transformam um problema de otimização multiobjetivo em um problema que envolve um único objetivo, permitindo incorporar implicitamente as preferências do usuário na busca. Estas técnicas utilizam indicadores específicos de qualidade para atribuir a cada indivíduo um único valor de *fitness*. Assim, em vez de otimizar as funções objetivo diretamente, algoritmos baseados em indicadores visam encontrar um conjunto de soluções que maximiza o indicador de qualidade subjacente.

A medida de hipervolume é um dos indicadores mais utilizados e aplicados para comparar resultados de algoritmos evolucionários multiobjetivo. No entanto, a ideia é apontar explicitamente para a maximização do hipervolume dominado dentro do processo de otimização. Beume et al. (2007) apresentam uma abordagem em que a seleção

de soluções candidatas depende da contribuição destas soluções para o hipervolume dominado. Assim, os autores propõem uma técnica de otimização que combina ideias emprestadas de MOEAs, como o bem-sucedido NSGA-II e estratégias de arquivamento. O algoritmo é então estabelecido por dois pilares: (1) classificação por não-dominância é utilizada como um critério de *ranking* e (2) o hipervolume é aplicado como critério de seleção para descartar os indivíduos que contribuíram para uma pior classificação do próprio hipervolume.

Auger et al. (2012) apresentam uma ideia semelhante à de Beume et al. (2007). Os autores utilizam indicadores de hipervolume ponderado para resolver problemas multi-objetivo. O indicador de hipervolume ponderado orienta a busca para regiões do espaço de objetivos que foram definidas pelo usuário e, ao mesmo tempo, tem a propriedade de aperfeiçoar a relação de dominância Pareto com o resultado da maximização dos indicadores para soluções ótimas de Pareto.

2.3.4 Redução de Dimensionalidade

Para resolver problemas com muitos objetivos uma abordagem frequentemente utilizada é a redução de dimensionalidade em objetivos redundantes. Esta abordagem defende a ideia de que alguns objetivos podem ser agregados para uma conveniente redução dos mesmos. Neste cenário é necessário identificar qual combinação e redução de objetivos ou grupos de objetivos, apresenta o menor grau de conflito, provocando a menor distorção na representação das frentes de Pareto.

Deb e Saxena (2005) propuseram um método para reduzir o número de objetivos por meio de uma abordagem baseada em Análise de Componentes Principais (*Principal Component Analysis* - PCA) juntamente com algoritmos evolutivos multiobjetivo. O procedimento proposto de forma iterativa identifica os objetivos redundantes de soluções obtidas por um NSGA-II e os elimina a partir de uma análise mais profunda. Outros trabalhos também apresentam técnicas baseadas em PCA para redução de objetivos (Saxena e Deb, 2007; Saxena et al., 2013).

Brockhoff e Zitzler (2006) propõem um método de redução de objetivos voltado para a integração direta na busca evolutiva. O algoritmo depende de uma mudança na estrutura de dominância. O problema de se encontrar o subconjunto mínimo de objetivos, mantendo a estrutura de dominância dada com um dado erro é apresentado pelos autores. Assim, um algoritmo guloso remove objetivos se as relações de dominância

não mudam com a remoção do objetivo comparado em questão. A remoção é considerada viável pois o objetivo é considerado um objetivo sem conflito. De forma similar Brockhoff e Zitzler (2007), propuseram um método para redução de objetivos que procura pelo subconjunto mínimo de objetivos com erro mínimo.

López et al. (2008) apresentaram dois algoritmos para reduzir o número de objetivos em problemas multiobjetivo identificando os objetivos mais conflitantes. Ambos os algoritmos são baseados na correlação entre cada par de objetivos para detectar os objetivos essenciais, sendo que objetivos distantes são tratados como conflitantes. Já López e Coello (2009) desenvolveram uma técnica de redução de objetivos que pode ser utilizada durante a pesquisa ou no processo de tomada de decisão.

Singh et al. (2011) propõem uma busca de soluções extremas da frente Pareto aproximada e desprezam as demais soluções. Uma vez que as soluções extremas são identificadas, uma técnica heurística verifica o número de soluções não dominadas resultantes de uma redução nas soluções alcançadas, indicando se os objetivos são importantes ou se eles podem ser reduzidos.

Freitas et al. (2013) apresentam uma formulação matemática que envolve o conceito de harmonia anteriormente definido por outros artigos (Giagkiozis et al., 2013; Purshouse e Fleming, 2007). Nesta proposta quanto mais harmônicos dois objetivos são, mais ele indica que os objetivos podem ser agregados em uma função objetivo composta sem muita perda na representação do conjunto de Pareto. Freitas et al. (2015) apresentam medidas de conflito e harmonia para identificar os objetivos que podem ser agregados. Esta abordagem apresenta os objetivos em ramos de árvore, de forma que os objetivos mais harmoniosos são agregados nas primeiras iterações.

2.3.5 Visualização

Para visualizar soluções de problemas de otimização com muitos objetivos (acima de três objetivos), a abordagem mais comum é a de coordenadas paralelas, como apresentado na Figura 2.4 para 7 objetivos.

Neste gráfico, o eixo horizontal representa os objetivos e o eixo vertical os valores normalizados entre o máximo e o mínimo. Esta normalização é realizada frequentemente através dos valores mínimos até o valor máximo conhecido ou em relação a uma meta mínima esperada pelo usuário para cada objetivo. Assim, a qualidade de uma solução é exibida como uma linha poligonal que intercepta cada eixo no ponto correspondente

ao valor associado para aquele objetivo. Quando há várias linhas se cruzando entre dois objetivos adjacentes, isto significa que há conflito entre estes objetivos. Ainda neste gráfico, cores diferentes também são atribuídas às linhas para tornar a visualização mais fácil e soluções dominadas são representadas como uma linha preta fina pontilhada.

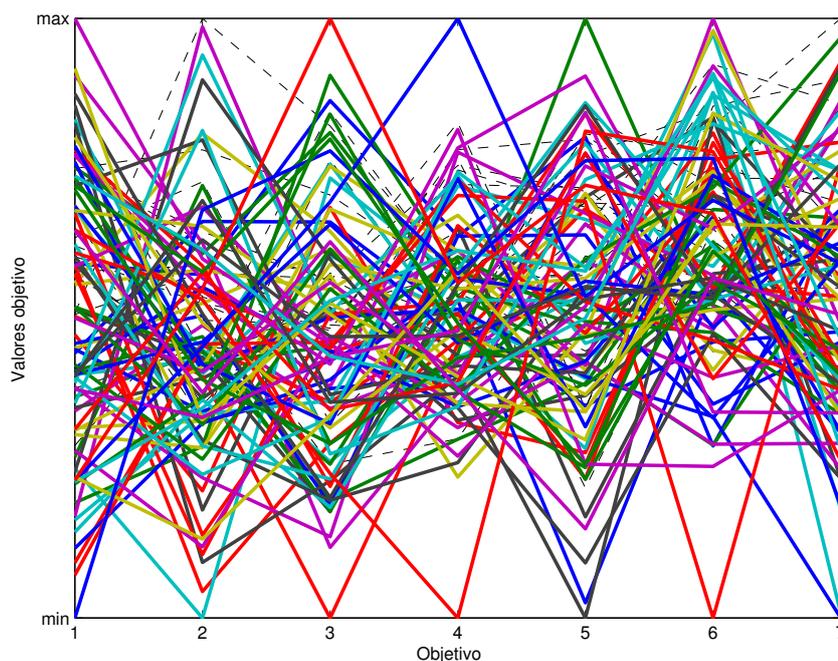


Figura 2.4: Visualização do resultado de um problema de otimização multiobjetivo com 7 funções objetivo (Freitas et al., 2015)

Quando se deseja analisar correlações entre pares de variáveis (objetivos), é necessário que estes estejam em sequências, em alguns casos pode ser feita a ordenação dos eixos conforme os valores de suas correlações ou então ordená-los de forma iterativa conforme intuição e necessidade do usuário.

Eddy e Lewis (2002) propõem visualização em nuvem, em que o projetista pode visualizar informações de projeto, bem como o espaço de objetivos simultaneamente. Mattson e Messac (2003) utilizam uma fronteira *s*-Pareto como o resultado de um filtro aplicado aos valores objetivos não-dominados originais. No entanto, esta abordagem leva a uma perda de dimensão na representação. Lotov (2005) usa uma técnica de decisão de Mapas Interativos para representar soluções no espaço de objetivos como uma alternativa aos gráficos de coordenadas paralelas para problemas que têm entre três e sete objetivos.

Obayashi e Sasaki (2003) usam mapas de auto-organização a explorar e encontrar

soluções de compromisso nas soluções de Pareto de quatro dimensões. Agrawal et al. (2005) usam um método Hiperespaço de Contagem Diagonal como ferramenta de visualização de soluções.

2.4 Conclusão

Os problemas do roteamento de veículos vêm sendo uma das mais importantes abordagens para a otimização de distribuição em redes, desde que foi proposto inicialmente por Dantzig e Ramser (1959). Deste modo, o PRV é amplamente estudado na literatura desde 1959, resultando em diversas classes de problemas com diversas técnicas empregadas na busca de soluções. Diante disso, este capítulo procurou abordar as principais referências que envolvem o problema de roteamento de veículos. As Seções 2.1.1 a 2.1.6 descreveram os principais objetivos abordados na literatura para o problema, bem como as principais variações do PRV em que estes objetivos foram considerados. Diversos trabalhos que abordam o PRV com mais de um objetivo foram descritos na Seção 2.1.7.

Diante deste contexto, diversos métodos foram propostos na literatura para a resolução de problemas multiobjetivo. Dentre estes, os algoritmos evolucionários multiobjetivo têm sido estudados constantemente para resolver, não só PRVs multiobjetivo, mas também diversos outros problemas combinatórios. Assim, a Seção 2.2 descreveu trabalhos que fizeram o uso destes algoritmos para resolver problemas multiobjetivo. O NSGA, NSGA-II e NSGA-III foram alguns métodos descritos nesta seção.

No entanto, a otimização de problemas de roteamento de veículos que envolvem muitos objetivos é algo complexo. Isto porque o número de soluções requeridas para aproximar toda a fronteira de Pareto cresce exponencialmente em função do número de objetivos considerados, sendo necessárias técnicas de otimização próprias para resolver essa classe de problemas. Assim, a Seção 2.3 apresenta os principais trabalhos da literatura que propuseram soluções para problemas com muitos objetivos. Neste contexto, esta seção teve como foco as estratégias que utilizaram redução de objetivos para resolver problemas com muitos objetivos. Além disso, as Seções 2.2.3 e 2.3.5 descrevem as principais formas de visualização de resultados para problemas multiobjetivo, bem como trabalhos que apresentam diversas formas de visualização para os mesmos.

Em virtude do contexto prático e inúmeras possibilidades de aplicações, neste trabalho será investigada a decomposição dos problemas de roteamento de veículos com janelas

de tempo. Ainda assim, a construção de mecanismos que resolvem este problema com muitos objetivos será estudada. Assim, o Capítulo 3 apresentará as principais definições que envolvem os problemas de roteamento de veículos, bem como as diversas variações do problema.

Capítulo 3

Problema de Roteamento de Veículos

Problema de Roteamento de Veículos é o nome genérico dado a uma classe de problemas de distribuição, cuja ideia principal é determinar a melhor rota possível que os veículos possam cumprir entre um depósito e um conjunto de consumidores. Por sua vez, o PRV é um dos problemas mais importantes no âmbito da logística de transporte e distribuição devido a sua aplicação prática. Estudos exaustivos sobre o problema renderam diversas características particulares que podem ser incorporadas ao mesmo, representando assim, uma grande variedade de restrições adicionais em problemas práticos. Deste modo, o PRV pode ser classificado em diversas categorias e tipos de acordo com as características presentes nas situações reais de operação.

O Problema de Roteamento de Veículos Capacitado é a versão mais simples do problema. Nela, todos os clientes possuem demandas conhecidas previamente, que devem ser atendidas integralmente pela frota de veículos. Todos os veículos são semelhantes (frota homogênea) e partem de um único depósito central. Cada cliente só pode ser visitado uma única vez por um único veículo e uma restrição de capacidade é imposta ao problema. Essa restrição estabelece que a soma das demandas de todos os clientes pertencente a uma rota não deve superar a capacidade do veículo a ela designada. O PRVC deu origem a diversos outros problemas de roteamento de veículos, motivo pelo qual iremos estudá-lo em primeiro lugar, apresentando, em seguida, as suas variações. No entanto, o problema proposto nesta dissertação é apresentado e descrito formalmente. Este, por sua vez, é denominado de “Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis”.

3.1 Problema de Roteamento de Veículos Capacitado

O Problema de Roteamento de Veículos Capacitado (PRVC) pode ser definido como um grafo não direcionado $G = (C, A)$, onde $C = \{c_1, c_2, c_3, \dots, c_n\}$ é o conjunto de vértices e $A = \{(c_i, c_j) : c_i, c_j \in C \text{ e } c_i \neq c_j\}$ o conjunto de arestas. Dado que os demais vértices representam os consumidores, o conjunto N é a união entre o conjunto C e os vértices c_0 e c_{n+1} . Entretanto, Cada aresta (c_i, c_j) tem um valor $d_{ij} \geq 0$ associado que representa o custo de se alcançar o vértice c_j a partir do vértice c_i e cada consumidor c_i tem uma demanda w_i de entrega. Para atender os consumidores, tem-se disponível uma frota $V = \{v_1, v_2, v_3, \dots, v_m\}$ de veículos homogêneos com capacidade máxima de carga Q .

Assim, o PRVC trata de encontrar um conjunto de rotas a partir de um depósito central para atender com o menor custo possível um conjunto de pontos de demanda (clientes). No PRVC as seguintes restrições devem ser satisfeitas:

- cada cliente é visitado uma única vez por um único veículo;
- cada rota deve ter início e fim no depósito;
- a soma das demandas dos clientes incluídos em uma rota não deve exceder a capacidade do veículo.

A variável de decisão do problema é dada por:

$$x_{ij}^v = \begin{cases} 1, & \text{se o veículos } v \text{ trafega no trecho } (i, j); \\ 0, & \text{caso contrário} \end{cases}$$

Pode-se agora definir matematicamente o problema básico de roteamento de veículos como:

$$\text{Minimize } \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}^v \quad (3.1)$$

Sujeito a

$$\sum_{v \in V} \sum_{j \in N} x_{ij}^v = 1; \quad \forall i \in C \quad (3.2)$$

$$\sum_{i \in N} w_i \sum_{j \in N} x_{ij}^v \leq Q; \quad \forall v \in V \quad (3.3)$$

$$\sum_{j \in N} x_{0j}^v = 1; \quad \forall v \in V \quad (3.4)$$

$$\sum_{i \in N} x_{i(n+1)}^v = 1; \quad \forall v \in V \quad (3.5)$$

$$\sum_{i \in N} x_{ij}^v - \sum_{i \in N} x_{ji}^v = 0; \quad \forall j \in C, \forall v \in V \quad (3.6)$$

$$x_{ij}^v \in \{0, 1\}; \quad \forall i, j \in N, \forall v \in V \quad (3.7)$$

A função objetivo, dada na equação 3.1, visa a minimização do custo (distância percorrida). A equação 3.2 garante que cada consumidor é visitado somente por um veículo. A equação 3.3 especifica que os veículos não devem ultrapassar a capacidade máxima de carga. As equações 3.4 e 3.5 demonstram que todos os veículos devem partir e retornar ao depósito central. A equação 3.6 garante que os veículos partam de um consumidor para outro (continuidade), enquanto a equação 3.7 indica a bivalência das variáveis de decisão.

3.2 Problema de Roteamento de Veículos com Janelas de Tempo

O problema de roteamento de veículos com janelas de tempo é uma variante do PRVC que consiste em incluir intervalos de tempo que limitam o atendimento aos consumidores. O problema é mais complexo, pois nessas condições existem outras variáveis de decisão que devem ser consideradas. Deste modo, no PRVJT o sentido da viagem do veículo na rota se torna relevante e alguns fatores como o tempo de atendimento no cliente, velocidade dos veículos e questões relacionadas direta ou indiretamente ao tempo se tornam significativas.

Dada a praticidade do problema e motivados pelos princípios da filosofia *Just-in-Time* e da consciência de competitividade, empresas têm forçado o uso da janela de tempo em todos os campos da distribuição de mercadorias (Ioannou et al., 2003). Assim sendo, esta seção descreve uma formulação do PRVJT com restrições de tempo forte.

No PRVJT cada cliente está associada uma janela de tempo $[a_i, b_i]$ e um tempo de serviço p_i (descarregamento), onde a_i é o horário mais cedo e b_i o horário mais tarde que se pode começar o atendimento. Os veículos devem chegar ao consumidor antes de b_i . Caso chegue ao consumidor antes do horário definido por a_i , este deve esperar pela abertura da janela.

O problema de roteamento de veículos com janelas de tempo pode ser definido como: seja um grafo não direcionado $G = (C, A)$, onde $C = \{c_1, c_2, c_3, \dots, c_n\}$ é o conjunto de vértices e $A = \{(c_i, c_j) : c_i, c_j \in C \text{ e } c_i \neq c_j\}$ o conjunto de arestas. Dado que os demais vértices representam os consumidores, o conjunto N é a união entre o conjunto C e os vértices c_0 e c_{n+1} . Entretanto, Cada aresta (c_i, c_j) tem um valor associado $d_{ij} \geq 0$ e $t_{ij} \geq 0$ que representam respectivamente, o custo e o tempo de se alcançar o vértice c_j a partir do vértice c_i . Cada consumidor c_i está associado a uma janela de tempo $[a_i, b_i]$, um tempo de serviço p_i , e uma demanda de entrega w_i . Para atender aos consumidores, tem-se disponível uma frota $V = \{v_1, v_2, v_3, \dots, v_m\}$ de veículos homogêneos com capacidade máxima de carga Q . As rotas que compõem uma solução são representadas pelo conjunto $R = \{r_1, r_2, \dots, r_m\}$, onde r_m é o custo da rota m . Cada veículo deve atender uma única rota, assim $|R| = |V|$.

Assim como no PRVC, a variável de decisão x_{ij}^v determina se o veículo v faz o percurso do consumidor i para o consumidor j , recebendo o valor 1, se verdadeiro, e 0 caso contrário. Já a variável de decisão s_i^v representa o instante de tempo em que o veículo v atende o consumidor i . Dada a definição formal e as variáveis de decisão do problema, pode-se agora definir matematicamente o PRVJT como:

$$\text{Minimize } \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}^v \quad (3.8)$$

Sujeito a

$$\sum_{v \in V} \sum_{j \in N} x_{ij}^v = 1; \quad \forall i \in C \quad (3.9)$$

$$\sum_{i \in N} w_i \sum_{j \in N} x_{ij}^v \leq Q; \quad \forall v \in V \quad (3.10)$$

$$\sum_{j \in N} x_{0j}^v = 1; \quad \forall v \in V \quad (3.11)$$

$$\sum_{i \in N} x_{i(n+1)}^v = 1; \quad \forall v \in V \quad (3.12)$$

$$\sum_{i \in N} x_{ij}^v - \sum_{i \in N} x_{ji}^v = 0; \quad \forall j \in C, \forall v \in V \quad (3.13)$$

$$s_i^v + p_i + t_{ij} - L(1 - x_{ij}^v) \leq s_j^v; \quad \forall i, j \in N, \forall v \in V \quad (3.14)$$

$$a_i \leq s_i^v \leq b_i; \quad \forall i \in N \quad (3.15)$$

$$x_{ij}^v \in \{0, 1\}, \forall i, j \in N; \quad \forall v \in V \quad (3.16)$$

Esta formulação difere da formulação do PRVC, apresentada na Seção 3.1, apenas na adição das equações 3.14 e 3.15. Por sua vez, estas equações representam as restrições de tempo que garantem a integralidade do PRVJT. Sendo que, a equação 3.14 assegura que o tempo de chegada de um veículo v a um consumidor j não ocorra antes do tempo de chegada ao consumidor anterior i , mais o tempo de serviço no primeiro, mais o tempo de percurso no trecho (i, j) , que é t_{ij} . A constante L sendo suficientemente grande garante que a equação seja somente uma restrição efetiva quando x_{ij}^v seja igual a 1, ou seja, quando o veículo v percorre o trecho (i, j) . Já a equação 3.15 garante que o serviço só pode ser iniciado entre a abertura e o fechamento da janela de tempo.

O PRVJT, em termos de complexidade computacional, é considerado NP-difícil uma vez que generaliza o PRVC. Como descrito por Alvarenga (2013), o trabalho de Cook (1971) prova a existência de uma sub-classe de problemas NP-Completo, incluindo o Problema da Satisfabilidade (conhecido como SAT). Cook (1971) provou que se houver um algoritmo determinístico capaz de resolver o SAT em tempo polinomial, então $P = NP$. Através de transformações em tempo polinomial, o SAT já foi reduzido a uma série

de outros problemas, como o Problema do Caixeiro Viajante, o Problema de Coloração de Grafos, entre outros. Contudo, se algum destes problemas forem resolvidos por algoritmos polinomiais o SAT também estará resolvido indiretamente, fazendo $P = NP$.

Deste modo, encontrar a solução para o PRVJT implica em obter simultaneamente a solução de vários problemas NP-difíceis, sendo por isso, também NP-difícil (Alvarenga, 2013). Contudo, com apenas um veículo, o PRVJT é NP-Completo (Tan et al., 2001a), idêntico ao Problema do Caixeiro Viajante. Em casos mais práticos, o número de veículos utilizados é maior do que um, fazendo do problema NP-difícil (Garey e Johnson, 1979).

3.3 Problema de Roteamento de Veículos com Janelas de Tempo Flexíveis

A definição do PRVJT básico implica que as janelas de tempo são tratadas como restrições rígidas. No entanto, uma característica que pode ser considerada no PRVJT é se as janelas de tempo devem ser rigorosamente respeitadas. Algumas abordagens do problema consideram a possibilidade dos veículos atenderem aos consumidores fora do intervalo da janela de tempo. O relaxamento da restrição de tempo conduz à redução do tempo de viagem, distância total percorrida, bem como do número de veículos utilizados no processo de entrega. Por outro lado, o atendimento tardio acarreta na insatisfação dos consumidores e reputação da empresa.

Assim, a redução de diversos custos operacionais só é possível desde que a violação de restrições de tempo seja penalizada, ficando a critério do decisor a escolha da solução mais adequada de acordo com as necessidades da distribuidora.

Diante deste contexto, a restrição temporal no PRVJT pode ser violada ou não. Quando a janela de tempo deve ser estritamente respeitada, ou seja, quando o cliente não pode ser atendido fora do horário determinado pela janela, se diz que o problema possui restrições de tempo forte (*hard*). Porém, caso seja permitido o atendimento aos consumidores fora dos limites da janela de tempo, é dito que o problema possui restrições de tempo fraca (*soft*) ou flexíveis (PRVJTF) (*Vehicle Routing Problem with Flexible Time Windows - VRPFTW*).

$$a_i \leq s_i^v; \quad \forall i \in N \quad (3.17)$$

$$s_i^v \leq b_i; \quad \forall i \in N \quad (3.18)$$

O PRVJTF pode ser modelado, basicamente, de três formas distintas. A primeira permite que o atendimento aos consumidores seja feito a qualquer momento após o início da janela de tempo, ou seja, nesta abordagem do problema é permitido que um veículo atenda a um consumidor após o fechamento da janela. Neste caso, a Equação 3.15 é substituída pela Equação 3.17 na formulação básica do PRVJT. A segunda maneira de modelar o PRVJTF é permitir que o atendimento aos consumidores seja feito a qualquer momento antes do fim da janela de tempo, ou seja, é permitido que um determinado veículo atenda a um consumidor antes da abertura da janela. Neste caso, a Equação 3.15 é substituída pela Equação 3.18. Por fim, a terceira modelagem permite que o atendimento aos consumidores seja feito antes, durante e depois dos limites da janela de tempo. Assim, a Equação 3.15 é removida da formulação do PRVJT, descrita na Seção 3.2.

Deste modo, quando o atendimento de algum consumidor pode ser feito fora dos limites da janela de tempo, uma variante do PRVJT com restrições de tempo fraca é considerada.

3.4 Problema de Roteamento de Veículos com Balanceamento de Rota

Em alguns casos basta adicionar novos objetivos ao problema de roteamento de veículos capacitado para gerar novas variações do problema. Assim, o Problema de Roteamento de Veículos com Balanceamento de Rota (PRVBR) (*Vehicle Routing Problem with Routing Balancing* - VRPRB) considera um objetivo que consiste em melhorar o balanceamento das rotas, ou seja, fazer com que as rotas tenham custo aproximado.

Esse tipo de abordagem é muito útil para as empresas de transporte, pois contri-

buem principalmente para que a jornada de serviço seja balanceada entre os motoristas. Quando apenas o custo é avaliado, é possível que um motorista tenha que trafegar muito mais que outro, o que certamente geraria um descontentamento por parte do motorista que trafega uma distância maior. Além disso, o balanceamento pode encobrir desgastes prematuros em um veículo em relação a outro, fazendo com que a empresa tenha uma maior economia em manutenção.

$$\text{Minimize } \text{Max}_v \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}^v - \text{Min}_v \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}^v \quad (3.19)$$

Na formulação básica do PRVBR as restrições tratadas são as mesmas do PRVC. A diferença é que um objetivo relacionado a equidade das rotas é considerado no problema. Assim, uma das estratégias mais utilizadas para alcançar equilíbrio entre as rotas é dada pela diferença entre a carga de trabalho da maior rota é a carga de trabalho da menor rota. A carga de trabalho das rotas pode ser considerada como o número de clientes visitados, a quantidade de produtos entregues e o tempo necessário ou o comprimento das rotas, por exemplo.

A Equação 3.19 apresenta um exemplo de função objetivo que considera a minimização da diferença entre a maior rota e a menor rota, sendo que a carga de trabalho das rotas é dada pelo comprimento da rota (distância percorrida). No entanto, outras funções objetivo podem ser formuladas a fim de obter um melhor balanceamento entre as rotas de uma solução. Assim, o balanceamento de rota pode ser modelado como a minimização da maior rota, como a minimização do desvio padrão entre as rotas, ou até mesmo como a minimização da soma das diferenças entre a carga de trabalho de cada rota e a menor carga de trabalho.

3.5 Problema de Roteamento de Veículos Min-max

Assim como o PRVBR, o Problema de Roteamento de Veículos Min-max (PRVMM) (*Min-max Vehicle Routing Problem* - MMVRP) introduz um novo objetivo ao PRV. Este objetivo procura minimizar o custo da maior rota, fazendo com que custos operacionais, bem como a satisfação do cliente sejam otimizados. Este objetivo pode ser calculado minimizando a rota que apresenta a maior distância, o maior tempo, ou o maior número

de consumidores atendidos.

A aplicação deste problema representa, em muitos casos, situações de gestão de emergência em que o objetivo é usar todos os veículos disponíveis para minimizar o tempo necessário para atender a todos os clientes que necessitam de recursos de emergência. A otimização de rotas de veículos para os esforços de gestão de emergência e de socorro tem sido um tema de muito interesse recentemente, e versões diferentes de problemas de roteamento de veículos foram formulados na literatura motivado pela gestão de emergências.

$$\text{Minimize } \text{Max}_v \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}^v \quad (3.20)$$

Na formulação básica do PRVMM as restrições tratadas são as mesmas do PRVC. A diferença é que um objetivo relacionado a minimização da maior rota é considerado no problema. Este objetivo é muitas vezes introduzido nos PRVs quando a minimização do tempo necessário para atender todos os clientes é mais importante do que a distância total percorrida.

Assim, a Equação 3.20 apresenta um exemplo de função objetivo que considera a minimização da maior rota, neste caso a maior rota é aquela que apresenta maior distância. No entanto, outras funções objetivo podem ser formuladas a fim de minimizar a rota que obtêm o maior custo.

3.6 Outras Variantes do Problema de Roteamento de Veículos

Pode-se perceber que apenas a restrição de capacidade de carga máxima não é capaz de representar todas as situações cotidianas enfrentadas pelos setores de logística das empresas de distribuição de mercadorias e serviços. Fazendo-se necessário, em muitos casos, introduzir ao problema restrições associadas aos clientes, veículos e depósito, ou até mesmo, abordar novos objetivos. Entretanto, o problema de roteamento de veículos pode ser definido considerando os seguintes componentes (Jozefowicz et al., 2008):

Rede: A rede pode ser representada por um grafo valorado, no qual os vértices representam cidades, clientes e/ou depósitos. As arestas representam as conexões entre

os vértices e cada uma delas possui um custo referente a sua utilização. Assim, a rede pode ser simétrica, assimétrica ou mista. Janelas de tempo associadas aos vértices e/ou as arestas também podem ser definidas em alguns problemas. Deste modo, um intervalo de tempo determina o horário permitido para percorrer uma aresta ou atender a um cliente.

Demandas: As demandas podem ser fixas, no qual todas as solicitações de pedido são conhecidas antecipadamente (antes da definição das rotas), ou dinâmicas, no qual novas solicitações podem surgir ao longo da jornada. Além disso, as demandas podem ser associadas não só aos vértices, mas também às arestas. Os clientes também podem requerer que diferentes tipos de produtos sejam entregues, e em alguns casos, os produtos podem ser devolvidos pelos consumidores.

Frota: A frota de veículos pode ter capacidade de carga igual (frota homogênea) ou diferente (frota heterogênea). Pode também ser composta por um único veículo ou por vários veículos, cuja utilização pode, ou não, ser limitada pela capacidade de carga, altura, distância, tempo máximo de operação, entre outros. Por sua vez, O Problema do Caixeiro Viajante (PCV) (*Traveling Salesman Problem* - TSP), por exemplo, pode ser definido como um problema de roteamento de veículos com um único veículo. Ainda assim, os veículos podem ser divididos em compartimentos, permitindo o transporte de diferentes produtos em diversas quantidades. No entanto, cada veículo pode iniciar e terminar sua rota no mesmo depósito, em depósitos diferentes, ou terminar sua rota no último cliente atendido.

Custo: Os custos são normalmente relacionados aos veículos, podendo ser a distância percorrida, tempo gasto de utilização, ou a quantidade de veículos utilizados. Os custos também podem incluir penalidades quando o cliente recebe uma entrega tardia ou incompleta, por exemplo. Por outro lado, pode haver um ganho associado às arestas ou vértices quando estes são percorridos e visitados.

Objetivos: Diversos objetivos podem ser considerados no problema. No entanto, a função objetivo pode ser calculada em um único período ou em vários períodos, dependendo se a demanda é fixa ou dinâmica, respectivamente. Os objetivos mais comuns incluem minimizar a distância total percorrida, o tempo total necessário, o custo total de execução, o tamanho da frota, maximizar a satisfação dos clientes e/ou dos motoristas, e maximizar a qualidade de serviço e/ou o lucro recolhido. Assim, quando múltiplos objetivos conflitantes são identificados, a utilização de abordagens multiobjetivo é extremamente vantajosa.

Diversas características podem ser introduzidas no problema, tornando cada modelo proposto mais complexo que o problema básico de roteamento de veículos. Isto porque, geralmente, cada modificação acarreta em novas restrições para o mesmo. Deste modo, as principais variações do PRV, surgidas da combinação dos diversos componentes descritos nesta seção, são descritas abaixo.

Problema de Roteamento de Veículos com Frota Heterogênea: O Problema de Roteamento de Veículos com Frota Heterogênea diferencia-se do problema capacitado pelo fato da frota possuir veículos distintos, ou seja, cada veículo apresenta uma capacidade de carga específica que pode ser diferente dos demais veículos da frota. Duas variantes deste problema podem ser consideradas. A primeira considera que o número de veículos disponíveis, bem como a capacidade de carga dos mesmos é conhecido previamente. Neste caso, basta atribuir uma rota a cada veículo para obter uma solução para o problema. A segunda variante considera o número de veículos ilimitado. Nesta abordagem do problema, além de construir uma rota para cada veículo, é necessário determinar o número de veículos utilizados (dimensionamento da frota). A esse último problema dá-se o nome de Problema de Dimensionamento e Roteamento de uma Frota Heterogênea de Veículos (PDRFHV) (*Fleet Size and Mix Vehicle Routing Problem - FSMVRP*).

Problema de Roteamento de Veículos com Múltiplos Depósitos: O Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD) (*Multi-Depot Vehicle Routing Problem - MDVRP*) se diferencia do problema básico por considerar a possibilidade dos veículos em iniciar sua rota em mais de um depósito. Deste modo, o problema considera a existência de vários depósitos, cada qual abrigando uma frota de veículos. Ao final de cada rota, um veículo deve sempre retornar ao seu depósito de origem, sem passar por outros depósitos. Assim, o problema consiste em decidir por qual depósito e por qual veículo o cliente será atendido.

Problema de Roteamento de Veículos com Coleta e Entrega: O Problema de Roteamento de Veículos com Coleta e Entrega (PRVCE) (*Vehicle Routing Problem with Pickup and Delivery - VRPPD*) permite que os clientes possam receber mercadorias dos veículos e/ou entregar mercadorias aos veículos. Entretanto, o problema geral de coleta e entrega pode ser dividido em duas categorias. A primeira é denominada de *one-to-many-to-one*. Nesta abordagem todas as cargas que irão satisfazer os clientes devem partir de um ou vários depósitos e todas as cargas coletadas nos clientes devem ser transportadas para algum depósito. A segunda

categoria é denominada *one-to-one*. Neste problema os veículos partem vazios de um ou de vários depósitos e, durante o percurso efetuam entregas aos clientes com mercadorias provenientes de coletas efetuadas anteriormente.

Problema de Roteamento de Veículos com Entregas Fracionadas: O Problema de Roteamento de Veículos com Entregas Fracionadas (PRVEF) (*Split Deliveries Vehicle Routing Problem - SDVRP*) permite que os clientes sejam atendidos por vários veículos desde que o custo total seja reduzido por esse tipo de atendimento. Para a formulação deste problema a restrição de capacidade é modificada, definindo uma nova restrição que garante que a soma das frações de demanda dos clientes visitados por um veículo não exceda sua capacidade.

Problema de Roteamento de Veículos Periódico: No problema capacitado o planejamento é feito considerando apenas um dia de trabalho. No caso do Problema de Roteamento de Veículos Periódico (PRVP) (*Periodic Vehicle Routing Problem - PVRP*) o período de planejamento é feito para no mínimo dois dias de trabalho. Assim, um veículo pode não retornar ao depósito no mesmo dia de sua partida. No período de planejamento de diversos dias alguns clientes necessitam ser visitados mais uma vez.

Problema de Roteamento de Veículos Aberto: O Problema de Roteamento de Veículos Aberto (PRVA) (*Open Vehicle Routing Problem - OVRP*) é caracterizado pelo fato do veículo partir de um depósito e não necessitar retornar ao mesmo depois de atender o último consumidor. Assim, cada rota do PRVA é um caminho hamiltoniano sobre um conjunto de pontos de demanda a serem atendidos.

3.7 Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis

Conforme apresentado na Seção 2.1.7, existem diversos trabalhos na literatura que apresentam abordagens multiobjetivo para problemas de roteamento de veículos com janelas de tempo. Deste modo, diversas formulações são tratadas procurando otimizar objetivos relacionados ao custo de transporte, satisfação do consumidor, satisfação dos motorista, segurança e violação das restrições do problema.

Assim, o problema proposto neste trabalho é denominado: Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis (MOPRV) (*Many-*

objective Vehicle Routing Problem with Flexible Time Windows). Este problema é uma generalização do problema de roteamento de veículos com janelas de tempo (Larsen e Danmarks, 1999), no qual a restrição referente à janela de tempo é transformada em objetivo. Além disto, outras funções objetivo são introduzidas no problema. Assim, fundamentando-se nas abordagens e na constância dos objetivos apresentados na literatura, foram escolhidos seis objetivos para o problema de roteamento de veículos com muitos objetivos e janelas de tempo flexíveis. São eles:

- f_1 = minimizar a distância total percorrida pela frota de veículos;
- f_2 = minimizar o número de rotas;
- f_3 = minimizar o grau de violação da restrição de tempo;
- f_4 = minimizar a espera dos veículos nos clientes;
- f_5 = minimizar a maior rota (makespan);
- f_6 = minimizar a diferença entre a maior e a menor rota (balanceamento de rota).

No entanto, o objetivo é construir rotas para o MOPRV que minimizem os custos de transporte (distância total percorrida e número de veículos utilizados), atendendo a todas as demandas, e ao mesmo tempo, minimizando a violação da restrição de tempo, o tempo de espera dos veículos, a maior rota e a diferença entre a maior e a menor rota. A modelagem matemática para o problema proposto é uma adaptação da modelagem proposta por Larsen e Danmarks (1999) para o problema de roteamento de veículos com janelas de tempo. Esta adaptação difere da modelagem original dado que diversos outros objetivos foram introduzidos ao problema, e que a restrição em que todos os consumidores devem ser visitados antes do fechamento da janela de tempo foi removida. Assim, as funções objetivo e as restrições do problema são formuladas matematicamente nas Seções 3.7.2 e 3.7.3, respectivamente.

3.7.1 Variáveis de Decisão

São três as variáveis de decisão do MOPRV. A variável de decisão x_{ij}^v determina se o veículo v faz o percurso do consumidor i para o consumidor j , recebendo o valor 1, se verdadeiro, e 0 caso contrário. Já a variável de decisão z_i^v demonstra o instante de tempo em que o veículo v chega ao consumidor i . Por fim, a variável de decisão s_i^v representa o instante de tempo em que o veículo v atende o consumidor i .

3.7.2 Funções Objetivo

O objetivo f_1 , definido na Equação 3.21, visa a minimização dos custos relacionados com a distância total percorrida pelos veículos. Este custo é determinado pela soma das distâncias entre os clientes na ordem em que eles foram visitados. O objetivo f_2 , definido na Equação 3.22, é formulado a fim de que o número de rotas ($|R|$) necessárias para atender os clientes seja o menor. No entanto, uma rota é determinada por uma viagem que começa e termina no depósito. Cada veículo deve atender uma única rota, sendo que um número ilimitado de veículos é considerado.

$$\text{Minimize } f_1 = \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}^v \quad (3.21)$$

$$\text{Minimize } f_2 = |R| \quad (3.22)$$

O objetivo f_3 , definido na Equação 3.23 é o resultado da transformação da restrição de tempo no próprio objetivo. Assim, f_3 visa a minimização do somatório dos atrasos dos veículos. Caso um veículo chegue a um consumidor depois do tempo de fim da janela de tempo, o atraso é computado como a diferença entre o tempo de chegada do veículo e o tempo de fim da janela.

$$\text{Minimize } f_3 = \sum_{i \in N} \sum_{v \in V} \text{Max}(z_i^v - b_i, 0) \quad (3.23)$$

O objetivo f_4 , definido na Equação 3.24, visa a minimização do somatório do tempo de espera dos veículos. Caso um veículo chegue a um consumidor antes do tempo de início da janela de tempo, a espera é computada como a diferença entre o tempo de abertura da janela e o tempo de chegada do veículo.

$$\text{Minimize } f_4 = \sum_{i \in N} \sum_{v \in V} \text{Max}(a_i - z_i^v, 0) \quad (3.24)$$

O objetivo f_5 , definido na Equação 3.25, considera a minimização do maior percurso (rota) existente na solução. O objetivo f_6 , definido na Equação 3.26, visa a minimização da diferença entre a rota mais longa e a rota mais curta. Estes dois objetivos são calculados em termos de distância.

$$\text{Minimize } f_5 = \text{Max}_v \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}^v \quad (3.25)$$

$$\text{Minimize } f_6 = \text{Max}_v \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}^v - \text{Min}_v \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}^v \quad (3.26)$$

3.7.3 Restrições

Apresentadas as funções objetivo, todas as soluções devem satisfazer um conjunto de restrições:

$$\sum_{v \in V} \sum_{j \in N} x_{ij}^v = 1; \quad \forall i \in C \quad (3.27)$$

$$\sum_{i \in N} w_i \sum_{j \in N} x_{ij}^v \leq Q; \quad \forall v \in V \quad (3.28)$$

$$\sum_{j \in N} x_{0j}^v = 1; \quad \forall v \in V \quad (3.29)$$

$$\sum_{i \in N} x_{i(n+1)}^v = 1; \quad \forall v \in V \quad (3.30)$$

$$\sum_{i \in N} x_{ij}^v - \sum_{i \in N} x_{ji}^v = 0; \quad \forall j \in C, \forall v \in V \quad (3.31)$$

$$s_i^v + p_i + t_{ij} - L(1 - x_{ij}^v) \leq z_j^v; \quad \forall i, j \in N, \forall v \in V \quad (3.32)$$

$$a_i \leq s_i^v; \quad \forall i \in N \quad (3.33)$$

$$x_{ij}^v \in \{0, 1\}, \forall i, j \in N; \quad \forall v \in V \quad (3.34)$$

A Equação 3.27 garante que cada cliente é visitado por um único veículo. A Equação 3.28 especifica que os veículos não devem ultrapassar a capacidade máxima de carga. As Equações 3.29 e 3.30 demonstram que todos os veículos devem partir e retornar ao depósito central. A Equação 3.31 garante que os veículos partam de um consumidor para outro (continuidade).

A restrição de tempo é garantida pela Equação 3.32, onde o instante de chegada de um veículo v a um consumidor j não poderá ocorrer antes do tempo de chegada ao consumidor anterior i mais o tempo de serviço no primeiro, mais o tempo de percurso no trecho (i, j) que é t_{ij} . A constante L sendo suficientemente grande garante que a equação seja somente uma restrição efetiva quando x_{ij}^v seja igual a 1, ou seja, quando o veículo v percorra o trecho (i, j) .

A garantia de que o serviço só pode ser iniciado após a abertura da janela é dada pela Equação 3.33. Finalmente, a Equação 3.34 garante a integralidade das variáveis do problema.

3.8 Conclusão

Esta dissertação recai sobre os principais conceitos e fundamentos que envolvem os problemas de roteamento de veículos, bem como as variações deste problema. Entretanto, desde que foi proposto, O problema de roteamento de veículos já foi formulado de diversas maneiras, seja remodelando as restrições do problema, ou incluindo novos objetivos ao mesmo. A forma básica do PRV inclui apenas restrições de capacidade máxima dos veículos, esta restrição garante que a soma das demandas atendidas por um determinado veículo não ultrapasse sua capacidade. Assim, a versão básica do PRV é conhecida com Problema de Roteamento de Veículos Capacitado. Contudo, o PRVC não é capaz de representar todas as situações práticas que empresas distribuidores vivenciam no dia-

a-dia, fazendo-se necessário introduzir novos componentes que conduzem o problema o mais perto possível da realidade.

Diante deste contexto, este capítulo apresentou os principais conceitos acerca do PRV, apresentando de maneira detalhada a formulação do problema básico (problema capacitado). No entanto, diversas outras variações do PRVC foram apresentadas, estas podem ser definidas alterando características da rede (características dos vértices e aresta do grafo), demanda dos consumidores, frota de veículos, custos de operação e objetivos. Em virtude do contexto prático e inúmeras possibilidades de aplicações, este trabalho investiga a decomposição dos problemas de roteamento de veículos com janelas de tempo, bem como os objetivos que podem ser introduzidos nestes problemas. Assim, este capítulo ainda descreve as principais características do PRVJT, apresentando uma formulação matemática para o mesmo.

Para explorar a interseção entre problemas de roteamento de veículos e problemas com muitos objetivos, este capítulo propõe o problema de roteamento de veículos com muitos objetivos e janelas de tempo flexíveis e apresenta a sua definição e formulação matemática, tornando-o uma generalização do problema de roteamento de veículos com janelas de tempo. A formulação proposta apresenta seis funções objetivo referentes à distância percorrida pelos veículos, número de veículos, violação da restrição de tempo, espera dos veículos nos consumidores, cálculo da maior rota e balanceamento de rota. Entretanto, estas funções objetivo envolvem diversos interesses de todos os envolvidos no processo de entrega, seja por minimizar os custos, maximizar a satisfação do cliente, e até mesmo promover a equidade de trabalho entre os motoristas.

Em algumas situações práticas os atendimentos às demandas dos consumidores podem ser adiados, deixando a cargo do tomador de decisões escolher entre custo e satisfação do consumidor. Assim, no problema proposto, todas as demandas de entrega têm igual importância e devem ser obrigatoriamente atendidas, por outro lado, o atendimento pode ser feito com o atraso necessário, desde que este atraso reduza o custo de transporte consideravelmente.

Capítulo 4

Fundamentos

Algoritmos evolucionários baseados em Pareto-dominância, como NSGA-II, têm sido amplamente utilizados para resolver problemas de roteamento de veículos multiobjetivo devido a sua generalidade e flexibilidade em relação à formulação do problema. No entanto, quanto maior o número de objetivos de um problema, pior é o desempenho desses algoritmos.

Neste contexto, um NSGA-III é utilizado para otimizar o MOPRV Completo (problema com seis objetivos) e uma ferramenta denominada Árvores de Agregação é utilizada para analisar a harmonia e o conflito entre os seis objetivos do problema. Assim, uma versão reduzida do MOPRV que considera a agregação dos objetivos mais harmoniosos no conjunto de dados é então proposta e um NSGA-II é usado para resolver a versão reduzida do problema, ou seja, o problema com os objetivos agregados. Deste modo, este capítulo apresenta os processos realizados pelo NSGA-II, NSGA-III e Árvores de Agregação.

4.1 Nondominated Sorting Genetic Algorithm II

O *Non-Dominated Sorting Genetic Algorithm* (NSGA) foi um dos primeiros Algoritmos Genéticos Multiobjetivo (AGM) mencionados na literatura especializada (Srinivas e Deb, 1994). Contudo, este algoritmo apresenta alguns pontos negativos, como a alta complexidade na ordenação de indivíduos não-dominados e a dependência de um parâmetro específico para garantir diversidade, os quais foram tratados em seu sucessor NSGA-II (Deb et al., 2002).

Deste modo, a vantagem do NSGA-II, quando comparado com os demais AGM, se deve à sua baixa complexidade computacional, $O(MN^2)$, sendo M o número de objetivos e N o tamanho da população. Além disso, o método é capaz de gerar soluções de boa qualidade para a maioria dos problemas do qual é aplicado, sendo estas soluções bem distribuídas na fronteira de soluções não-dominadas.

Para isso, o NSGA-II qualifica os indivíduos de sua população utilizando-se da definição de dominância. Este procedimento, denominado de *fast non-dominated sorting* (FNDS), permite a categorização dos indivíduos em distintas fronteiras (*fronts*) conforme o seu grau de dominância. Este, por sua vez, apresenta uma complexidade computacional igual à $O(MN^2)$. Métodos de classificação por não dominância mais recentes já categorizam indivíduos em fronteiras com uma complexidade igual à $O(N \log^{M-2} N)$ (Fang et al., 2008). A Figura 4.1 ilustra um problema de minimização com duas funções objetivo e três fronteiras.

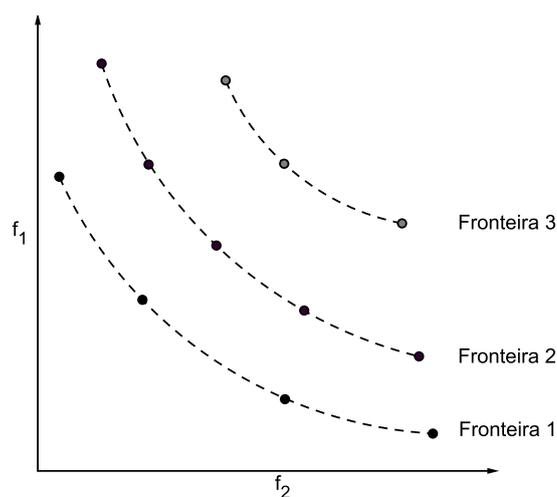


Figura 4.1: Ordenação por dominância

Assim, na fronteira 1 estão contidas as soluções dominantes, ou seja, as que não são dominadas por nenhuma outra solução do conjunto. A fronteira 2 é composta por soluções que são dominadas apenas pelas soluções da fronteira 1. Por sua vez, a fronteira 3 é formada por soluções que são dominadas pelas soluções da fronteira 1 e pelas soluções da fronteira 2, e assim sucessivamente. O Algoritmo 4.1 mostra todo o procedimento feito pelo FNDS.

Para cada solução i contida na população R (linha 4) é calculado o número total de soluções que dominam a solução i (nd_i) (linhas 10-11) e o conjunto de soluções dominadas pela solução i (U_i) (linhas 8-9). Deste modo, a ordenação de não dominância é executada

Algoritmo 4.1: Pseudocódigo FNDS

```

1  $nd_i$ : Número total de soluções que dominam a solução  $i$ 
2  $U_i$ : Conjunto de soluções dominadas pela solução  $i$ 
3  $F_k$ : Fronteira  $k$ 
4 para cada solução  $i \in R$  faça
5    $nd_i = 0$ 
6    $U_i = \emptyset$ 
7   para cada solução  $j \neq i$  e  $j \in R$  faça
8     se  $i < j$  então
9        $U_i = U_i \cup \{j\}$ 
10    se  $j < i$  então
11       $nd_i = nd_i + 1$ 
12    se  $nd_i = 0$  então
13       $F_1 = F_1 \cup \{i\}$ 
14  $k = 1$ 
15 enquanto  $F_k \neq \emptyset$  faça
16    $temp = \emptyset$ 
17   para cada solução  $i \in F_k$  faça
18     para cada solução  $j \in U_i$  faça
19        $nd_j = nd_j - 1$ 
20       se  $nd_j = 0$  então
21          $Temp = Temp \cup \{j\}$ 
22    $k = k + 1$ 
23    $F_k = Temp$ 

```

em duas etapas. Na primeira etapa (linhas 4-13) todos os indivíduos da população R são classificados de acordo com o grau de dominância nd_i . Caso este grau seja igual a zero o indivíduo pertence à fronteira 1 (linhas 12-13). Na etapa 2 (linhas 14-23) o contador nd_j , de cada uma das soluções j , é decrementado (linha 19) até que nd_j seja igual a zero (linha 20). Quando isso ocorre, a solução j é incluída (pertence) na fronteira corrente (linha 21). Desta forma, esta etapa consiste em separar cada indivíduo em diferentes fronteiras, de acordo com seus valores de dominância indicados por nd_j .

Para garantir a diversidade das soluções ao longo de uma fronteira, o NSGA-II utiliza um operador denominado distância de multidão (*Crowding Distance*) (Deb et al., 2002). Este operador calcula a distância média entre um ponto central i e dois pontos localizados nas extremidades deste ponto central, $(i - 1)$ e $(i + 1)$. Esta distância é normalizada pela diferença entre o maior e o menor valor que existe para cada objetivo. Assim, a

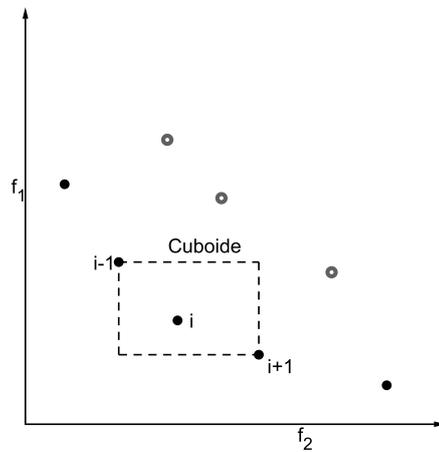


Figura 4.2: Distância de Multidão

distância de multidão de uma solução i (d_i) representa uma estimativa do perímetro formado pelo cuboide cujos vértices são os seus vizinhos mais próximos. Os pontos contidos nas extremidades da fronteira recebem um valor arbitrariamente grande, sendo priorizados durante a seleção. A Figura 4.2 ilustra o cálculo da distância de multidão.

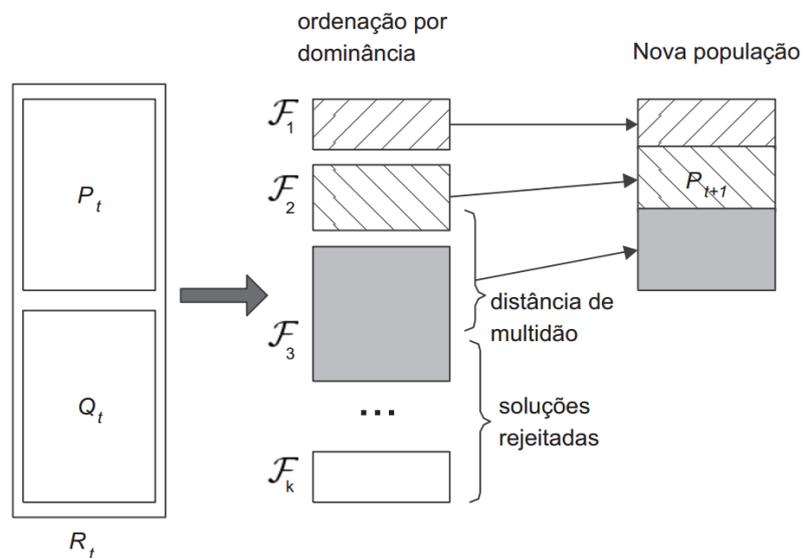


Figura 4.3: Procedimento do NSGA-II (Deb et al., 2002)

O Algoritmo 4.2 apresenta todo o processo feito pelo NSGA-II. Dados os procedimentos básicos do NSGA-II, o algoritmo primeiramente gera uma população inicial (pais) P_t de tamanho N (linha 2). Posteriormente os operadores de cruzamento, mutação e seleção são aplicados nesta população (linha 3), resultando em uma nova população

Q_t (filhos), também de tamanho N . Um vez aplicados os operadores genéticos, o NSGA-II une as populações P_t e Q_t resultando em uma população R_t de tamanho $2N$ (linha 5).

Algoritmo 4.2: Pseudocódigo NSGA-II

```

1  $t = 0$ : Contador de gerações
2  $P_t =$  Gerar população inicial (População Pai)
3  $Q_t =$  Aplicar operadores de Seleção, Cruzamento e Mutação ( $P_n$ ) (População
   Filha)
4 repita
5    $R_t = P_t \cup Q_t$ 
6    $F[] =$  Ordenação por dominância ( $R_t$ )
7    $P_t + 1 = \emptyset$ 
8    $i = 1$ 
9   enquanto  $|P_t + 1 + F_i| \leq N$  faça
10     $P_t + 1 = P_t + 1 \cup F_i$ 
11     $i = i + 1$ 
12   para todo  $s_j \in F_i$  faça
13     $d_j =$  Distância de Multidão ( $s_j$ )
14    $s =$  Ordenar as soluções  $s \in F_i$  crescentemente de acordo com o valor de  $d_j$ 
15   para  $j = 1$  até  $j = N - |P_t + 1|$  faça
16     $P_t + 1 = s_j$ 
17    $Q_t + 1 =$  Aplicar operadores de Seleção, Cruzamento e Mutação ( $P_t + 1$ )
18    $t = t + 1$ 
19 até Critério de parada ;
20  $R_t = P_t \cup Q_t$ 
21 Fronteira = Obter soluções não-dominadas ( $R_t$ )
22 retorna Fronteira

```

O seguinte passo faz uso do FNDS sobre a população R_t (linha 6), selecionando as melhores fronteiras, até que a população $P_t + 1$ volte a ter tamanho igual a N . Caso a última fronteira a ser inserida na população $P_t + 1$ ultrapasse o número de indivíduos máximos da população (N) (linhas 9-11), utiliza-se o algoritmo *Crowding Distance* para julgar quais serão os indivíduos, pertencentes a esta fronteira, que farão parte da nova população (linhas 12-16). Este ciclo se repete até que um critério de parada seja atingido. A Figura 4.3 ilustra o procedimento de inserção das soluções da população R na população P .

4.2 Nondominated Sorting Genetic Algorithm III

Os algoritmos de otimização multiobjetivo evolucionários demonstraram sucesso em vários problemas práticos envolvendo principalmente dois e três objetivos. Todavia, há uma necessidade crescente de desenvolvimento de algoritmos de otimização multiobjetivo evolutivos para lidar com problemas com muitos objetivos (Deb e Jain, 2014). Diante disto, Deb e Jain (2014) propuseram uma versão do NSGA-II que utiliza pontos de referência para resolver problemas que envolvem muitos objetivos. Este algoritmo enfatiza os membros da população não-dominada próximos de um conjunto de pontos de referência fornecidos previamente. O algoritmo proposto por Deb e Jain (2014) foi denominado de *Nondominated Sorting Genetic Algorithm III* (NSGA-III).

A estrutura básica proposta pelo NSGA-III permanece semelhante ao algoritmo NSGA-II original com mudanças significativas em seu mecanismo de seleção. Ao contrário do NSGA-II que utiliza o cálculo do perímetro formado pelo cuboide dos vizinhos mais próximos (*Crowding Distance*), a manutenção da diversidade entre os membros da população no NSGA-III é auxiliado pelo fornecimento de uma série de pontos de referência bem espalhados. Estes pontos de referência podem ser predefinidos de forma estruturada ou fornecido preferencialmente pelo usuário. Na ausência de qualquer informação de preferência, qualquer colocação estruturada de pontos de referência pode ser adotada (Deb e Jain, 2014). O Algoritmo 4.3 apresenta o processo feito pelo NSGA-III.

Assim, o procedimento para identificar as frentes não-dominadas (FNDS), apresentado na Seção 4.1, também é usado no NSGA-III. Sendo que todos os indivíduos da população não-dominada da fronteira 1 até a fronteira l são incluídos inicialmente na população S_t (linhas 5-8). Se S_t contiver exatamente o número máximo N de indivíduos, nenhuma outra operação é necessária e a próxima geração é iniciada com $P_{t+1} = S_t$ (linhas 9-10). Caso $|S_t| > N$ (linha 11), os membros da fronteira $(l - 1)$ são selecionados, ou seja, $P_{t+1} = \cup_{i=1}^{l-1} F_i$ (linha 12). O restante ($K = N - |P_{t+1}|$) dos membros da população são escolhidos a partir da última fronteira F_l (linha 13). O processo de seleção então é formado por outros três procedimentos, a normalização (linha 14), a associação (linha 15) e a preservação do nicho (linha 17).

Entretanto, após a população P_{t+1} ser formada, operadores genéticos comuns são utilizados para criar a Q_{t+1} . Assim, o NSGA-III faz uma seleção elitista tentando manter a diversidade entre as soluções da população, enfatizando soluções mais próximas da linha de referência gerada por cada ponto de referência. Além disso, o número de

Algoritmo 4.3: Pseudocódigo NSGA-III

Entrada: H pontos de referência Z^s estruturados ou ponto de aspiração fornecidos Z^a , população pai P_t

Saída: $P_{t=1}$

- 1 $S_t = \emptyset, i = 1$
- 2 $Q_t = \text{Recombinação} + \text{Muta\c{c}\~{a}o} (P_t)$
- 3 $R_t = P_t \cup Q_t$
- 4 $(F_1, F_2, \dots) = \text{Non-dominated-Sort}(R_t)$
- 5 **repita**
- 6 | $S_t = S_t \cup F_i$ e $i = i + 1$
- 7 **at\~{e}** $|S_t| \geq N$;
- 8 \c{U}ltima frente a incluir: $F_l = F_i$
- 9 **se** $|S_t| = N$ **ent\~{a}o**
- 10 | $P_{t+1} = S_t$, pare
- 11 **sen\~{a}o**
- 12 | $P_{t+1} = \cup_{j=1}^{l-1} F_j$
- 13 | Pontos a serem escolhidos a partir de $F_l : K = N - |P_{t+1}|$
- 14 | $\text{Normalize}(f^n, S_t, Z^r, Z^s, Z^a)$
- 15 | $\text{Associate}(S_t, Z^r)$
- 16 | $\rho_j = \sum_{s \in S_t / F_l ((\pi(s)=j)?1:0)}$
- 17 | $\text{Niching}(K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})$

indivíduos N de uma população do NSGA-III se aproxima do número de pontos de referência H que foram gerados para este mesmo algoritmo, ou seja, $N \approx H$. Isto é, o tamanho da população do NSGA-III é definido como o menor múltiplo de quatro maior que o número de pontos de referência H gerados. Deste modo, a cada membro da população do NSGA-III é dada uma igual importância. Por estas razões o NSGA-III não emprega qualquer tipo de operação de seleção explícita para gerar a população filha Q_{t+1} . Esta população é portanto construída através da aplicação de operadores de cruzamento e de mutação habituais, escolhendo aleatoriamente os pais de P_{t+1} . Uma geração do NSGA-III possui uma complexidade computacional de $O(N^2 \log^{M-2} N)$ ou $O(N^2 M)$, o que for pior, sendo M o número de objetivos do problema e N o número de soluções (Deb e Jain, 2014).

As próximas seções apresentam detalhadamente o processo realizado pela normalização adaptativa dos membros da população, associação e preservação de nicho.

4.2.1 Normalização Adaptativa dos Membros da População

O processo realizado pela normalização é apresentado pelo Algoritmo 4.4.

Algoritmo 4.4: Pseudocódigo *Normalização* (f^n, S_t, Z^r, Z^s, Z^a)

Entrada: S_t, Z^a (pontos estruturados), Z^a (pontos fornecidos)
Saída: f^n, Z^r pontos de referência no hiperplano normalizado

- 1 **para** $j = 1$ até M **faça**
- 2 Calcule o ponto ideal: $z_j^{\min} = \min_{s \in S_t} f_j(s)$
- 3 Traduzir objetivos: $f'_j(s) = f_j(s) - z_j^{\min}, \forall s \in S_t$
- 4 Calcule os pontos extremos: $z^{j, \max}$
- 5 Calcule interceptações a_j para $j = 1, \dots, M$
- 6 Normalizae os objetivos f^n usando a Equação 4.3
- 7 **se** Z^a é dado **então**
- 8 Mapeie cada ponto normalizado no hiperplano usando a Equação 4.3 e salve os pontos no conjunto Z^r
- 9 **senão**
- 10 $Z^r = Z^a$

Inicialmente, o processo de normalização determina o ponto ideal através da identificação do valor mínimo (Z_i^{\min}) na primeira frente de Pareto para cada função objetivo $i = 1, 2, \dots, M$ (linha 2), construindo $Z = (Z_1^{\min}, Z_2^{\min}, \dots, Z_M^{\min})$. Cada valor objetivo de S_t é então traduzido subtraindo o valor da função objetivo f_i por Z_i^{\min} (linha 3), de modo que o ponto ideal da população S_t traduzida seja um vetor de zeros. O objetivo traduzido é denotado como $f'_i(x) = f_i(x) - Z_i^{\min}$. No passo seguinte, o ponto extremo em cada eixo objetivo é identificado (linha 4) por encontrar a solução $x \in S_t$ que retorna o valor mínimo para a Equação 4.1 (função escalar) com vetor de pesos w sendo a direção do eixo. Onde $w^j = (\epsilon, \dots, \epsilon)^T$, $\epsilon = 10^{-6}$ e $w_j^j = 1$.

$$ASF(x, w) = \max_{i=1}^M f'_i / w_i, \quad \text{para } x \in S_t \quad (4.1)$$

Entretanto, os pontos extremos são encontrados minimizando cada objetivo individualmente, de modo que, o ponto extremo para o eixo objetivo i , possui um alto valor para f_i e valores baixos para os demais objetivos do problema. Em termos geométricos, os pontos extremos de um problema de otimização são os pontos mais próximos de cada eixo objetivo, considerando apenas as soluções não-dominadas. Os vetores extremos de cada

objetivo M são então utilizados para constituir um hiperplano linear M -dimensional. A interceptação a_i do i -ésimo objetivo e o eixo hiperplano linear podem então ser calculados. Assim, dada uma matriz $E_{M \times M}$ formada pelos pontos extremos, e um vetor $b = [1, \dots, 1]$ de tamanho M , as interceptações a_i nos eixos objetivos i são encontradas por $Ea = b$. Em outras palavras, um sistema linear com M variáveis deve ser resolvido para encontrar as interceptações nos eixos.

Tome como exemplo um problema com três objetivos e que os pontos $(-1, 1, 2)$, $(2, 0, -3)$ e $(5, 1, -2)$ são os pontos extremos deste problema. Um sistema linear com três variáveis é formulado considerando estes pontos (Equação 4.2).

$$\begin{aligned} (-1)x + y + 2z &= 1 \\ 2x - 3z &= 1 \\ 5x + y - 2z &= 1 \end{aligned} \tag{4.2}$$

Este sistema retorna como resultado: $x = 0.4$, $y = 1.8$ e $z = 0.6$, ou seja, $a = \{-0.4, 1.8, 0.6\}$. Isto significa que as interceptações nos eixos x , y e z são os pontos $(\frac{1}{0.4}, 0, 0)$, $(0, \frac{1}{1.8}, 0)$ e $(0, 0, \frac{1}{1.6})$, respectivamente. Assim, os objetivos podem ser normalizados utilizando estas interceptações (linhas 5-6), como segue na Equação 4.3 (Deb e Jain, 2014).

$$f_i^n(x) = \frac{f_i'(x)}{a_i - z_i^{\min}} = \frac{f_i(x) - z_i^{\min}}{a_i - z_i^{\min}}, \quad \text{para } i = 1, 2, \dots, M \tag{4.3}$$

A Figura 4.4 ilustra a formação de um hiperplano para 3 objetivos.

Dado M como o número de objetivos e N como a quantidade de soluções de uma população. A identificação do ponto ideal tem uma complexidade computacional igual à $O(MN)$. Da mesma forma, traduzir os objetivos também requer um total de $O(MN)$ cálculos. A identificação de pontos extremos requer $O(M^2N)$ cálculos enquanto determinar as interceptações nos eixos objetivo exige uma inversão de matriz de tamanho $M \times M$, exigindo $O(M^3)$ operações. Em seguida, a normalização de um máximo de $2N$ membros da população exige $O(N)$ cálculos. Por fim, mapear cada ponto normalizado no hiperplano requer $O(MH)$ operações, onde H é o número de pontos de referência

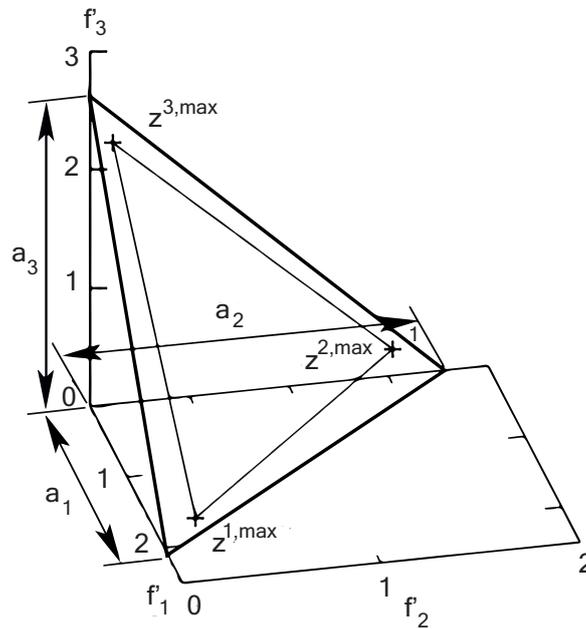


Figura 4.4: Procedimento de formação do hiperplano para um problema com 3 objetivos (Deb e Jain, 2014)

(Deb e Jain, 2014).

4.2.2 Associação

Depois de normalizar cada objetivo de forma adaptativa com base na extensão dos indivíduos (membros) S_t no espaço de objetivos, o passo seguinte associa cada indivíduo da população com um ponto de referência. O Algoritmo 4.5 apresenta todo o processo realizado na etapa de associação.

Para associar cada indivíduo a um ponto de referência, primeiramente uma linha de referência correspondente a cada ponto de referência sobre o hiperplano é definida (linhas 1-2), unindo o ponto de referência com a origem. Em seguida, calcula-se a distância perpendicular de cada um dos membros da população S_t para cada uma das linhas de referência (linhas 3-5). O ponto de referência, cuja linha de referência está mais próxima de um membro da população no espaço de objetivos normalizado é considerado relacionado com o elemento da população (linhas 6-7) (Deb e Jain, 2014). A Figura 4.5 ilustra o processo de associação.

Todas as operações de associação para um máximo de $2N$ membros da população e

Algoritmo 4.5: Pseudocódigo *Associação* (S_t, Z^r)**Entrada:** S_t, Z^r **Saída:** $\pi(s \in S_t), d(s \in S_t)$

- 1 **para** cada ponto de referência $z \in Z^r$ **faça**
- 2 └ Calcule a linha de referência $w = z$
- 3 **para** cada $s \in S_t$ **faça**
- 4 └ **para** cada $w \in Z^r$ **faça**
- 5 └ Calcule $d^\perp(s, w) = s - w^T s / \|w\|$
- 6 Associe $\pi(s) = w : \operatorname{argmin}_{w \in Z^r} d^\perp(s, w)$
- 7 Associe $d(s) = d^\perp(s, \pi(s))$

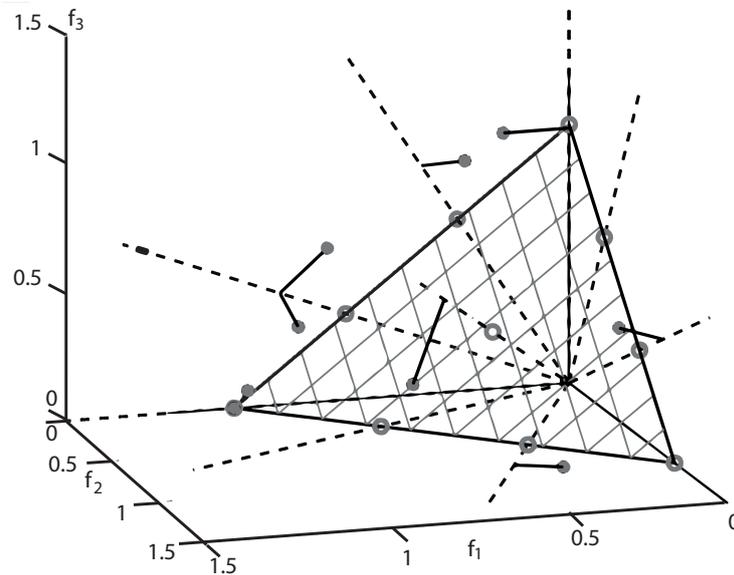


Figura 4.5: Associação dos membros da população com os pontos de referência (Deb e Jain, 2014)

H pontos de referência exigiria $O(MNH)$ operações (Deb e Jain, 2014).

4.2.3 Preservação do Nicho

Os pontos de referência podem ter um ou mais membros da população a ele associados, ou ainda, não ter nenhum membro associado ao mesmo. Assim, na etapa de preservação de nicho é contado o número de membros da população $P_{t+1} = S_t / F_l$ que são associados a cada ponto de referência. Essa contagem é denotada como ρ_j para cada ponto de referência j .

Algoritmo 4.6: Pseudocódigo *Preservação do Nicho* ($K, \rho_j, \pi, d, Z^r, F_l, P_{t+1}$)

Entrada: $K, \rho_j, \pi(s \in S_t), d(s \in S_t), Z^r, F_l$

Saída: P_{t+1}

```

1  $k = 1$ 
2 enquanto  $k \leq K$  faça
3    $J_{\min} = j : \arg \min_{j \in Z^r} \rho_j$ 
4    $\bar{j} = \text{acaso}(J_{\min})$ 
5    $I_{\bar{j}} = s : \pi(s) = j, s \in F_l$ 
6   se  $I_{\bar{j}} \neq \emptyset$  então
7     se  $\rho_{\bar{j}} = 0$  então
8        $P_{t+1} = P_{t+1} \cup (s : \arg \min_{s \in I_{\bar{j}}} d(s))$ 
9     senão
10       $P_{t+1} = P_{t+1} \cup \text{acaso}(I_{\bar{j}})$ 
11       $\rho_{\bar{j}} = \rho_{\bar{j}} + 1, F_l = F_l \setminus s$ 
12       $k = k + 1$ 
13   senão
14      $Z^r = Z^r / \{\bar{j}\}$ 

```

Primeiro, este procedimento identifica o ponto de referência $J_{\min} = j : \arg \min_j \rho_j$ que tem o ρ_j mínimo, ou seja, o ponto de referência que possui o menor número de membros associados a ele é escolhido (linha 3). No caso em que mais de um ponto de referência tenha o número mínimo de soluções associadas, um destes é escolhido de forma aleatória (linha 4).

Identificado o ponto de referência mínimo, o passo seguinte procura os indivíduos na fronteira F_l que estão associados a este ponto (linha 5). Caso a fronteira F_l não tenha qualquer membro associado com o ponto de referência j , o ponto de referência é excluído (linha 14). Caso contrário outra condição é testada, se $\rho_{\bar{j}} = 0$ (linha 7) (o que significa que não há nenhum membro de P_{t+1} associado ao ponto de referência \bar{j}), o membro que tem a menor distância perpendicular a partir da linha de referência é adicionado à P_{t+1} (linha 8). Se $\rho_{\bar{j}} \geq 1$ (linha 9) (o que significa que existe um ou mais membros de S_t/F_l associados ao ponto de referência), um membro que pertence à F_l e está associado ao ponto de referência é escolhido aleatoriamente (linha 10). A contagem $\rho_{\bar{j}}$ é incrementada em um (linha 12). A contagem de nicho é atualizada (linha 11), e o procedimento é repetido para um total de k vezes para preencher todos os espaços vazios da população P_{t+1} . O processo é apresentado no Algoritmo 4.6.

No procedimento de nicho, identificar o ponto de referência que possui o menor

número de membros associados a ele requer $O(H)$ comparações. Assumindo que $L = |F_l|$, procurar os indivíduos na fronteira L que estão associados a este ponto requer $O(L)$ cálculos. No entanto, identificar o membro que tem a menor distância perpendicular a partir da linha de referência também exige, no pior caso, um total de $O(L)$ cálculos. Outras operações têm menor complexidade. Assim, os cálculos acima no algoritmo de preservação de nicho precisam ser realizados no máximo L vezes, necessitando assim de $O(L^2)$ ou $O(LH)$ operações, o que for maior. No pior cenário ($S_t = F_1$, ou seja, a primeira frente não dominada excede o tamanho da população), $L \leq 2N$ (Deb e Jain, 2014).

4.3 Árvores de Agregação

A Árvore de Agregação (Freitas et al., 2015) é uma ferramenta que permite visualizar a redundância e o conflito entre os objetivos de um problema de otimização. Esta abordagem baseia-se na organização de objetivos em ramos de árvores que representam as melhores possibilidades de agregação de objetivos em um problema. A cada iteração do algoritmo, o par de objetivos mais harmoniosos são agregados em um novo objetivo, até que haja apenas um objetivo no problema que represente a soma simples ou a soma normalizada de todos os objetivos para em um problema mono-objetivo. Assim, porcentagens sobre nós pais demonstram o conflito entre seus filhos, sendo que qualquer nó pai representa a agregação de seus nós objetivo filhos. Mais especificamente, a Árvore de Agregação tem as seguintes propriedades:

- Os nós folhas representam os objetivos na forma f_n , onde n é o número do objetivo.
- Pais dos nós folhas representam um objetivo composto na forma $f_a + f_b - c$, onde f_a e f_b são os objetivos agregados e c é o conflito existente entre eles.
- Outros pais representam objetivos compostos de classes mais altas formados de maneira semelhante, os valores de conflito nestes nós consideram somente o conflito entre os objetivos compostos de seus filhos.
- Objetivos são respectivamente agrupados de acordo com sua harmonia e, consequentemente, primos distantes na árvore representam objetivos menos harmoniosos.

- Nós em preto representam conflito global. Valores em vermelho representam conflito local mais concentrado em bons valores para os objetivos. Valores em azul representam conflito de soluções ruins nos objetivos.

Quando dois objetivos compostos são agrupados em um novo objetivo composto, o conflito nos objetivos compostos anteriores não é considerado no conflito do objetivo novo. No entanto, os valores de conflito na árvore podem ser normalizados ou redimensionados se valores elevados de conflito dificultam a percepção do usuário. Uma aplicação útil da árvore considera a posição final dos nós folha para selecionar objetivos e organizá-los em paralelo em uma estrutura de gráficos polares. Isto é, o algoritmo de Árvores de Agregação propõe o uso de uma melhor representação dos valores absolutos em um gráfico polar. De maneira semelhante às coordenadas paralelas, cada linha no gráfico polar representa uma solução, sendo que a principal diferença entre estes gráficos é que o gráfico polar tem uma estrutura toroidal que permite a representação de objetivos extremos em coordenadas paralelas. Além disso, o interior do círculo dos gráficos polares representa valores ruins para cada um dos objetivos enquanto a parte externa representa valores bons para os mesmos objetivos. Para esta representação, os valores de objetivos são normalizados linearmente e a gama de valores absolutos, em cada objetivo está representada fora do círculo. Assim, uma técnica de agrupamento, *Part-and-Select Algorithm* (PSA) é aplicada para facilitar a visualização dos resultados (Salomon et al., 2014).

Para exemplificar a utilidade das Árvores de Agregação, bem como dos gráficos polares, considere um conjunto de soluções (40 soluções) representadas em coordenadas paralelas como na Figura 4.6. A relação entre os objetivos e sua redutibilidade (possibilidade de agregação de objetivos) para estas quarenta soluções é representada em uma estrutura de árvore pela Figura 4.7.

Neste exemplo, a Árvore de Agregação sugere a redução dos objetivos $f_1, f_2, f_4, f_6, f_8, f_{10}, f_{12}$ nos primeiros passos. Todos estes objetivos estão em harmonia completa dado que existe 0% de conflito entre eles. Assim, somando os valores da redução proposta no primeiro passo em um novo objetivo composto f_a , temos agora 6 objetivos restantes. Aplicando mais uma redução, f_a é agrupado com o objetivo f_9 com 5% de conflito (ou 95% de harmonia). Definindo que $f_b = f_9 + f_a$, dos 5 objetivos restantes, o algoritmo de redução agrupa então nos próximos passos $f_c = f_b + f_5$ com 25% de conflito, $f_d = f_c + f_7$ com 25% de conflito e resultando em um problema com 3 objetivo, $f_e = f_{11} + f_3$ com 25% de conflito e, por fim, $f_f = f_d + f_e$ com 100% de conflito. O objetivo f_f representa

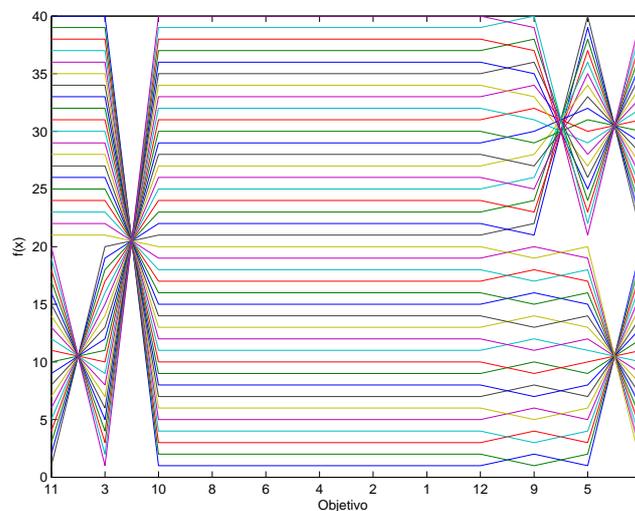


Figura 4.8: Coordenadas paralelas com ordem diferente de objetivos adjacentes (Freitas et al., 2015)

Assim, o gráfico polar deste exemplo demonstra a relação de extremo conflito entre os objetivos 11 e 7. Além disso, é possível visualizar o efeito de cada salto de um ramo da árvore para outro. Por exemplo, é possível visualizar que as soluções em verde, geralmente tem valores elevados para os objetivos f_{10} , f_8 , f_6 , f_4 , f_2 , f_1 , f_{12} e f_9 , sendo que bons valores para estes objetivos resultam em valores baixos para f_3 e valores médios para f_{11} e f_5 . Por outro lado, valores baixos (valores em vermelho) nos objetivos de completa harmonia acarretam em valores elevados para f_{11} e f_3 .

Assim, a Árvore de Agregação possibilita visualizar a relação entre os objetivos, bem como os agrupamentos de objetivos, de forma que a interpretação do conflito é facilitada. Assim, em vez de mostrar apenas o conflito entre cada par de objetivos, a árvore também demonstra o conflito entre grupos de objetivos e como eles se relacionam. Isto é, os nós da árvore disponibilizam informações sobre qual seria o melhor agrupamento de objetivos, além de demonstrar a localidade de conflito. Esta abordagem possibilita que o decisor analise objetivos compostos de acordo com sua harmonia. Assim, as Seções 4.3.1 e 4.3.2 apresentam as medidas de conflito e harmonia e a localidade de conflito, respectivamente. A Seção 4.3.3 descreve os detalhes para a construção da Árvore de Agregação. E por fim, a Seção 4.3.4 ilustra um exemplo de execução da Árvore de Agregação.

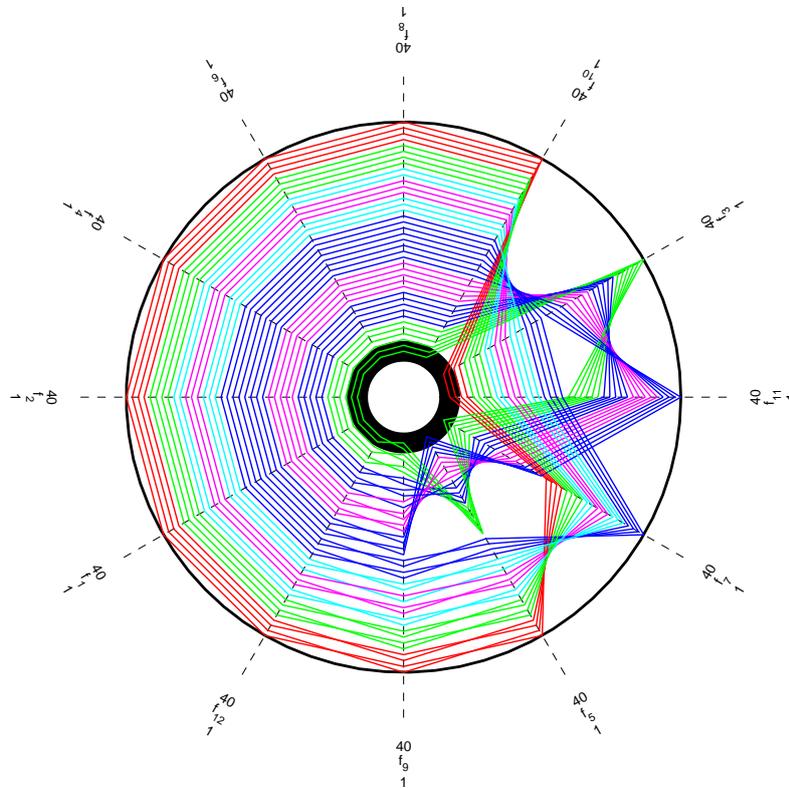


Figura 4.9: Gráfico polar para o conjunto de soluções das Figuras 4.6 e 4.7 (Freitas et al., 2015)

4.3.1 Medidas de Conflito

O algoritmo Árvores de Agregação conceitua conflito entre dois objetivos, quando bons valores para um deles implicam valores ruins para o outro. No entanto, o conceito de harmonia apenas implica que a melhoria de um objetivo conduz a uma melhoria no outro. Neste sentido, a harmonia nem sempre é exatamente o oposto do conflito, sendo mais relacionada à possibilidade de unir os objetivos por meio de uma soma, sem perda de qualidade na frente de Pareto. Assim, se queremos agrupar dois objetivos em um novo objetivo composto, o melhor é agrupar os objetivos com maior harmonia até mesmo se existir um certo nível de conflito entre eles (Freitas et al., 2015).

De acordo com estas definições, a Árvore de Agregação pode considerar três medidas gerais de conflito: direta, maxmin e não-paramétrica. O conflito direto considera a diferença absoluta entre um objetivo e outro. Assim, ele funciona para objetivos que podem ser convertidos nas mesmas unidades (por exemplo, através da redução dos valores

objetivos de custo financeiro em uma aplicação prática). A medida de conflito direta é definida na Equação 4.4, onde C_{ab} é o conflito entre os objetivos a e b , X_{ij} é o valor da solução i no objetivo j e $\min(X_j)$ é o menor valor para o objetivo j .

$$C_{ab} = \sum_i |(X_{ia} - \min(X_a)) - (X_{ib} - \min(X_b))| \quad (4.4)$$

O conflito maxmin, que é definido pela Equação 4.5, normaliza os valores de objetivos em um intervalo de 0 (mínimo) a 1 (máximo) antes de fazer a comparação, consequentemente, esta medida implica que a importância dos objetivos é inversamente proporcional a sua faixa de valores alcançáveis. Entretanto, a Equação 4.5 define a medida de conflito C_{ab} maxmin, sendo que, X_{ij} é o valor da solução i no objetivo j , $\min(X_j)$ é o menor valor para o objetivo j e $\max(X_j)$ o maior valor para o objetivo j . A medida maxmin retorna um valor máximo de conflito igual a n (número de soluções).

$$C_{ab} = \sum_i \left| \left(\frac{X_{ia} - \min(X_a)}{\max(X_a) - \min(X_a)} \right) - \left(\frac{X_{ib} - \min(X_b)}{\max(X_b) - \min(X_b)} \right) \right| \quad (4.5)$$

Por fim, a medida mais usual de conflito, e a que utilizamos neste trabalho, é a medida não-paramétrica definida na Equação 4.6, onde C_{ab} é o conflito entre os objetivos a e b , R_{ij} é a classificação da solução i no objetivo j , e C_{\max} , representada na Equação 4.7, é o valor máximo de conflito não-paramétrico para n soluções. Assim, a abordagem não-paramétrica classifica os valores absolutos de cada um dos objetivos antes de serem comparados. A métrica reflete o grau pelo qual linhas se cruzariam em coordenadas paralelas.

$$C_{ab} = \sum_i |R_{ia} - R_{ib}| \quad (4.6)$$

$$C_{\max} = \sum_{i=1}^n |2i - n - 1| \quad (4.7)$$

A partir dessa perspectiva, a harmonia entre dois objetivos é inversamente proporcional ao conflito não-paramétrico, ou seja, o conceito de harmonia apenas implica que a melhoria em um objetivo conduz a uma melhoria no outro. Assim, objetivos harmoniosos são apresentados em coordenadas paralelas de maneira que as linhas não se cruzam. O cálculo de harmonia é apresentado pela Equação 4.8.

$$H_{ab} = 1 - \frac{C_{ab}}{C_{\max}} \quad (4.8)$$

4.3.2 Localidade de Conflito

No entanto, a Árvore de Agregação não verifica somente a quantidade de conflito entre os objetivos, verificando também, a região de conflito/harmonia no espaço de objetivos. A informação sobre a localidade do conflito é importante porque se o conflito ocorre para soluções com valores objetivo ruins, isto significa que não há conflito inerente entre os objetivos sendo analisados em si. Isto porque as soluções com valores objetivo ruins para dois objetivos estão apenas na frente de Pareto por causa de outros objetivos (Freitas et al., 2015).

Diante disso, ramos das árvores também são coloridos de acordo com sua região de conflito. A medida da região de conflito $L(a, b)$ do objetivo a em relação ao objetivo b é definida na Equação 4.9. Os valores de $L(a, b)$ são normalizados entre -1 e +1 pelo seu valor máximo $L(a, b)_{\max}$, definido na Equação 4.10. Os valores negativos indicam desarmonia mais concentrada em soluções “ruins”, enquanto os valores positivos indicam desarmonia concentrada em valores de “bons”. A Figura 4.10 apresenta vários exemplos de compromissos entre dois objetivos em coordenadas paralelas.

$$L(a, b) = \frac{\sum_{i=1}^n |R_{ia} - R_{ib}| R'_{ia}}{L(a, b)_{\max}} \quad (4.9)$$

$L(a, b)_{\max}$ é o valor máximo de $L(a, b)$, que é descrita pela seguinte equação:

$$L(a, b)_{\max} = \sum_{i=\lfloor n/2+1 \rfloor}^n |2i \lfloor 3n/2 \rfloor - 1| R'_{ia}$$

$$L(a, b)_{\max} = \frac{2 \lfloor n/2 \rfloor^2 - 2 \lfloor n/2 \rfloor}{2} \quad (4.10)$$

Na Figura 4.10(a), uma melhora no objetivo 1 sempre leva a uma melhora no objetivo 2. Neste caso não há conflito entre os objetivos e este comportamento é global. Este é o melhor caso para fazer um novo objetivo composto com a simples simetria dos objetivos. Se há pouco conflito e muita harmonia, a maior parte do comportamento ainda será global pois a harmonia é global. Na Figura 4.10(b), o comportamento também é global pois há conflito em todos os locais.

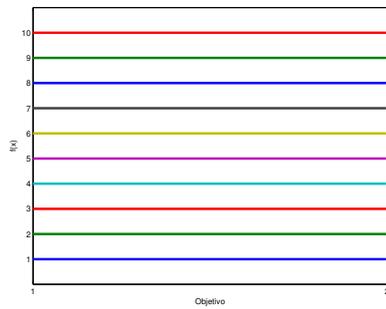
As Figuras 4.10(c) e 4.10(d), por outro lado, são exemplos nos quais o conflito é apenas local (Figura 4.10(c)). No primeiro caso o conflito apenas acontece para soluções que são ruins em relação a estes objetivos, considerando-se um problema de minimização. Nestes casos, a remoção de outros objetivos eventualmente causaria um desaparecimento do compromisso, pois para os objetivos sendo analisados, apenas uma solução estaria na frente de Pareto. No caso seguinte (Figura 4.10(d)), a eliminação de outros objetivos não conduziria a um menor conflito entre os objetivos sendo considerados pois 5 soluções sempre estarão na primeira frente de Pareto, isto porque o conflito entre estas soluções nestes objetivos é inerente. Contudo, percebe-se que representar a localidade do conflito é relevante para a agregação dos objetivos.

O exemplo da Figura 4.10(e) mostra um caso de conflito global médio, enquanto a Figura 4.10(f) apresenta a existência de conflito local mas a posição do conflito é diferente para os objetivos.

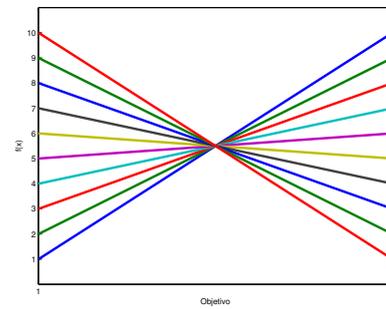
4.3.3 Construção do Algoritmo

O processo de construção de uma árvore de Agregação envolve as medidas de conflito e harmonia entre os objetivos, bem como a região de conflito. Assim, a estrutura do algoritmo para gerar a árvore é descrita no Algoritmo 4.7.

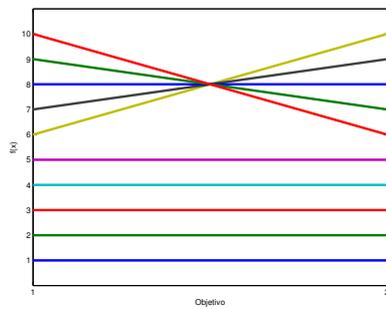
O Algoritmo 4.7 tem como entrada uma matriz X com os valores de função objetivo de possíveis soluções para o problema. Na linha 1, a estrutura da árvore é inicializada com um nó raiz como pai de todos os objetivos. Na linha 2, todos os valores dos



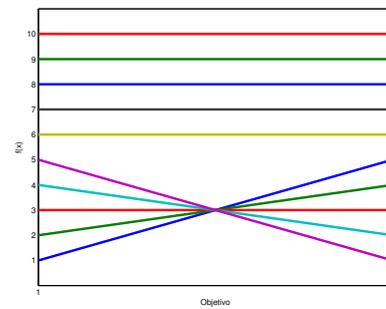
(a) Harmonia total



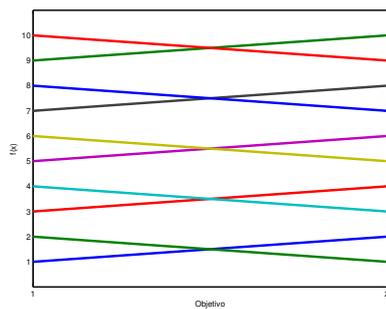
(b) Conflito máximo global



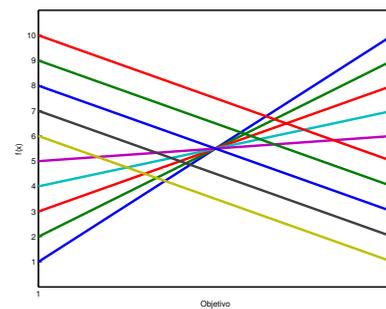
(c) Conflito concentrado em valores altos



(d) Conflito concentrado em valores baixos



(e) Conflito médio global



(f) Conflito local concentrado em locais diferentes em cada objetivo

Figura 4.10: Exemplos de vários tipos de conflito (Freitas et al., 2015)

objetivos são normalizados com seus valores de classificação, como exigido pelas medidas de conflito não-paramétrico. A partir da linha 3, um laço iterativo começa. A cada

Algoritmo 4.7: Pseudocódigo Construção de uma Árvore de Agregação

Entrada: Matriz $X_{n \times m}$ com conjunto de valores de função objetivo para m objetivos e n soluções

Resultado: Árvore t

- 1 Inicializar a árvore t com um nó raiz e todos os objetivos como filhos;
 - 2 $X \leftarrow \text{normalizar}(X)$;
 - 3 **enquanto** *ainda há objetivos a serem agrupados* **faça**
 - 4 $X' \leftarrow \text{reduzir}(X)$;
 - 5 $X' \leftarrow \text{normalizar}(X')$;
 - 6 $H \leftarrow \text{matriz_de_harmonia}(X')$;
 - 7 $a, b \leftarrow$ nós folha ou nós com objetivos compostos de X' com maior harmonia;
 - 8 $c \leftarrow \text{conflito}(X', a, b)$;
 - 9 $L \leftarrow \text{localidade}(X', a, b)$;
 - 10 t recebe um novo nó nn ;
 - 11 nn recebe a e b como filhos;
 - 12 nn guarda os valores (c, L) ;
 - 13 a e b são agrupados. (A próxima geração tem um objetivo a menos);
 - 14 *Imprimir a Árvore de Agregação t ;*
 - 15 *Ordenar os nós folhas de t na ordem em que aparecem em t ;*
 - 16 *Imprimir o gráfico polar considerando a ordem*
-

iteração deste laço os dois objetivos (entre originais ou compostos) mais harmoniosos, que são filhos do nó raiz, são agrupados em um novo nó pai da árvore, que se torna filho da raiz.

Na linha 4, uma nova versão dos valores dos objetivos é criada para a iteração do laço. Esta nova versão X' já inclui o somatório dos objetivos agrupados até o momento. Esta etapa consiste em somar os valores de classificação dos objetivos que foram agrupados. Na linha 5, esta nova versão é normalizada mais uma vez para uma comparação justa dos objetivos. A normalização é feita de acordo com a medida de conflito desejada que, no caso de conflito não-paramétrico, corresponde a trocar o valor por sua posição de rank. Basicamente, o processo de “ranqueamento” atribui qualidade às soluções que foram dadas como entrada (assinala os piores e os melhores valores para cada objetivo). Quando se tem valores iguais para o mesmo objetivo, o “ranqueamento” é feito, de tal forma que o conflito resultante entre dois objetivos seja mínimo.

Na linha 6, o cálculo da harmonia entre todos os objetivos é feito. Por sua vez, a linha 7 computa o par de objetivos com maior harmonia. Assim, a medida da distância (conflito) entre duas dimensões é a diferença de fileiras entre suas soluções, sendo que as dimensões agregadas (objetivos agregados) são tratadas como uma nova dimensão

(novo objetivo), onde valores de classificação são somados e, em seguida, classificados mais uma vez.

Entretanto, a Árvore de Agregação apresenta um processo diferenciado para objetivos que possuem o mesmo valor de conflito para todos os objetivos do problema. Assim, se um dado objetivo f possui um valor h de harmonia igual para todos os outros objetivos do problema, e esta harmonia é a maior entre todas as outras, o objetivo f fica na reserva para que os objetivos i e j , que possuem a segunda maior harmonia, sejam agregados. Na iteração seguinte, o objetivo f é agregado ao objetivo composto $(i + j)$. Este processo, primeiro agrega os objetivos i e j para depois agregar o objetivo f ao objetivo $(i + j)$, resultando no objetivo $(i + j + f)$. Este processo garante que f não seja agregado arbitrariamente com outro objetivo, fazendo do resultado o mais próximo do original. Tal procedimento pode ser chamado de critério de desempate.

Nas linhas 8 e 9, o conflito e a localidade do conflito são calculados para o par de objetivos mais harmoniosos. Na linha 10, um novo nó é incluído na árvore como um filho do nó raiz. Na linha 11, este novo nó recebe como filhos os nós que estavam representando os objetivos mais harmoniosos até o momento. Na linha 12, os valores de conflito e localidade de conflito para os objetivos mais harmônicos são guardados também neste novo nó. Neste ponto, o nó raiz tem um objetivo a menos e uma nova iteração do algoritmo se inicia. Durante o processo de se construir a árvore, é preciso de matrizes simples com informação sobre o conflito entre cada par de objetivos. Após o processo iterativo, as linhas 13, 14 e 15 apresentam a Árvore de Agregação resultante, ordena os nós folhas na ordem em que eles aparecem na árvore, e apresenta o gráfico polar, respectivamente.

4.3.4 Exemplo de Execução

Tome como exemplo um problema com quatro objetivos e que cinco soluções foram dadas como entrada para o algoritmo. Considere ainda, que a medida de conflito utilizada foi a não-paramétrica. Assim, temos como entrada para a Árvore de Agregação uma matriz com quatro colunas, representando os objetivos, e cinco linhas, representando as soluções, como segue abaixo:

$$Pontos = \begin{bmatrix} 300 & 3 & 750 & 72 \\ 200 & 3 & 720 & 35 \\ 250 & 3 & 800 & 20 \\ 270 & 3 & 550 & 70 \\ 305 & 3 & 650 & 80 \end{bmatrix}$$

A árvore é iniciada com um nó raiz e cada nó filho representa um objetivo ainda não agregado. A Figura 4.11 ilustra a árvore inicial.

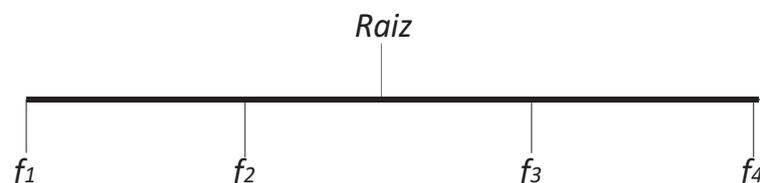


Figura 4.11: Estrutura da árvore inicial com um nó raiz como pai de todos os objetivos

Os valores de objetivos da matriz *Pontos* são então normalizados de acordo com o conflito não-paramétrico. Neste caso, a normalização funciona como um *ranking*, onde o menor valor (para um problema de minimização) recebe classificação 1 e o maior valor recebe classificação n , de um total de n soluções. As colunas 1, 2, 3 e 4 representam respectivamente o *ranking* das soluções para os objetivos f_1 , f_2 , f_3 e f_4 , respectivamente. Assim, a matriz normalizada fica da seguinte forma:

$$Normalizada = \begin{bmatrix} 4 & 1 & 4 & 4 \\ 1 & 2 & 3 & 2 \\ 2 & 3 & 5 & 1 \\ 3 & 4 & 1 & 3 \\ 5 & 5 & 2 & 5 \end{bmatrix}$$

A partir deste passo começa o laço iterativo. Como nenhum objetivo foi agregado até o momento o passo seguinte calcula a harmonia entre todos os objetivos do problema. O conflito é dado pelo somatório da diferença dos valores de *ranking* de cada linha entre

os objetivos comparados, e a harmonia é inversamente proporcional a este conflito. É possível notar que os valores para a segunda coluna (objetivo f_2) da matriz são todos iguais. O *rankeamento* é feito em ordem crescente para esta coluna. Porém, no momento que se calcula o conflito, bem como a harmonia entre o objetivo f_2 (coluna 2) e qualquer outro objetivo, a coluna 2 se ajusta de tal forma que a harmonia máxima é considerada. Neste caso, como todos os valores da coluna 2 são iguais, a coluna 2 se ajusta resultando em uma coluna igual ao do objetivo comparado, e por consequência, obtendo 100% de harmonia com qualquer outro objetivo do problema. Deste modo, a matriz de harmonia é calculada utilizando a Equação 4.8 e o resultado é apresentado na seguinte matriz:

$$Harmonia = \begin{bmatrix} - & 100\% & 16.6667\% & 83.3333\% \\ 100\% & - & 100\% & 100\% \\ 16.6667\% & 100\% & - & 16.6667\% \\ 83.3333\% & 100\% & 16.6667\% & - \end{bmatrix}$$

Pode-se observar que a linha 2, bem como a coluna 2 são preenchidas com valores iguais a 100%. Isso significa que o objetivo f_2 possui 100% de harmonia com os demais objetivos. A relação entre os objetivos f_1 e f_3 retornou 16% de harmonia, os objetivos f_1 e f_4 retornaram 83% de harmonia, e os objetivos f_3 e f_4 também possuem 16% de harmonia. Assim, teríamos que agregar o objetivo f_2 com algum outro objetivo, visto que o maior grau de harmonia na tabela foi apresentado pelo objetivo f_2 . Porém, qualquer escolha neste passo do algoritmo, que tenha que agrupar o objetivo f_2 com outro objetivo, implica em uma escolha totalmente arbitrária.

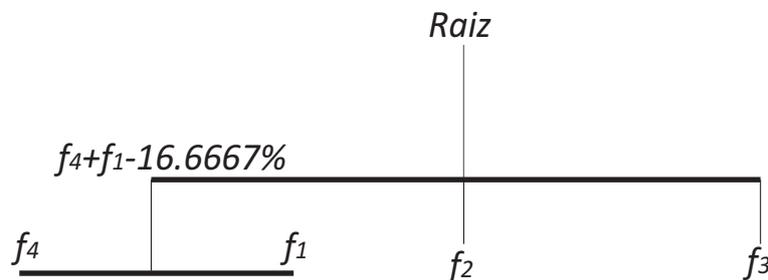


Figura 4.12: Estrutura da árvore com a agregação dos objetivos f_1 e f_4

Deste modo, a Árvore de agregação agrupa os objetivos com o segundo maior grau de harmonia, agregando o objetivo f_2 posteriormente. Neste caso o objetivo f_1 é agregado ao objetivo f_4 com 83.3333% de harmonia. Como o conflito não-paramétrico é

inversamente proporcional à harmonia, os objetivos f_1 e f_4 possuem aproximadamente 16.6667% de conflito. Um novo nó que representa a agregação entre os objetivos é criado e os filhos deste nó são os objetivos f_1 e f_4 , como ilustrado na Figura 4.12.

Os objetivos originais f_1 e f_4 são convertidos para um único objetivo composto ($f_4 + f_1$). Isto é, o problema que continha quatro objetivos agora é composto por três objetivos. Sendo estes os objetivos originais f_2 e f_3 , e o objetivo composto ($f_1 + f_4$). Assim, o somatório das colunas 1 e 4 da matriz normalizada é feito. As colunas 1 e 4 são então removidas da matriz e uma coluna que representa o objetivo agregado ($f_1 + f_4$) é acrescentada. o somatório e o *rankeamento* são então apresentados abaixo:

$$\begin{array}{ccccccc}
 & f_1 & & f_4 & & f_1 + f_4 & & ranking \\
 \begin{bmatrix} 4 \\ 1 \\ 2 \\ 3 \\ 5 \end{bmatrix} & + & \begin{bmatrix} 4 \\ 2 \\ 1 \\ 3 \\ 5 \end{bmatrix} & = & \begin{bmatrix} 8 \\ 3 \\ 3 \\ 6 \\ 10 \end{bmatrix} & \xrightarrow{\text{rankeamento}} & \begin{bmatrix} 4 \\ 1 \\ 2 \\ 3 \\ 5 \end{bmatrix}
 \end{array}$$

A matriz resultante normalizada é apresentada a seguir, na qual as colunas 1, 2 e 3 representam o *ranking* para os objetivos f_2 , f_3 e ($f_4 + f_1$), respectivamente.

$$\text{Normalizada} = \begin{bmatrix} 1 & 4 & 4 \\ 2 & 3 & 1 \\ 3 & 5 & 2 \\ 4 & 1 & 3 \\ 5 & 2 & 5 \end{bmatrix}$$

Como o objetivo f_2 apresentou a maior harmonia (100% de harmonia), e esta harmonia foi igual para todos os objetivos do problema, este objetivo foi deixado na reserva para ser agregado aos objetivos que apresentaram a segunda maior harmonia. A coluna que representa o objetivo f_2 , por possuir todos os valores absolutos iguais, é então ajustada para que a harmonia entre ele e o objetivo agregado ($f_4 + f_1$) seja o maior possível. Isto é, a coluna do objetivo f_2 da matriz normalizada é ajustada para ficar idêntica à coluna do objetivo ($f_4 + f_1$), quando duas colunas da matriz normalizadas são iguais o

somatório da diferença entre os valores é igual a zero, ou seja, obtém-se harmonia igual a 100%. Assim, o passo seguinte agrega o objetivo f_2 ao objetivo composto ($f_4 + f_1$) com 0% de conflito. A árvore resultante é apresentada pela Figura 4.13.

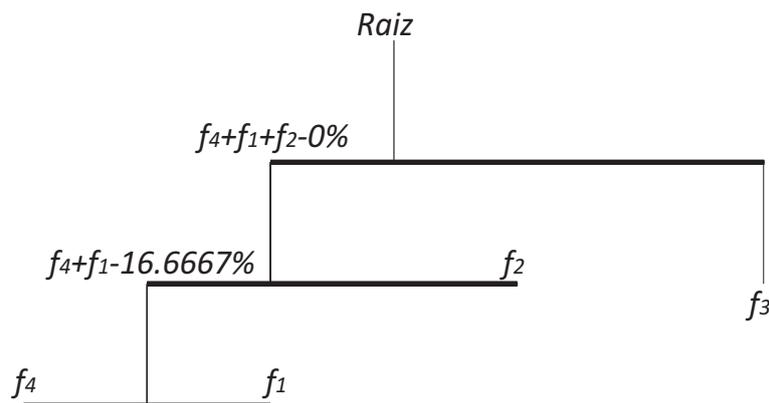


Figura 4.13: Estrutura da árvore com a agregação dos objetivos $f_4 + f_1$ e f_2

Deste modo, o problema foi reduzido em mais um objetivo, sendo composto agora pelo objetivo original f_3 e o objetivo composto ($f_4 + f_1 + f_2$). Assim, os valores de *ranking* dos objetivos f_1 , f_2 e f_4 devem ser somados e retirados da matriz para gerar uma nova coluna que represente o novo objetivo composto. O somatório, bem como o *rankeamento* destes objetivos é apresentado a seguir.

$$\begin{array}{c} f_1 \\ \left[\begin{array}{c} 4 \\ 1 \\ 2 \\ 3 \\ 5 \end{array} \right] \end{array} + \begin{array}{c} f_4 \\ \left[\begin{array}{c} 4 \\ 2 \\ 1 \\ 3 \\ 5 \end{array} \right] \end{array} + \begin{array}{c} f_2 \\ \left[\begin{array}{c} 4 \\ 1 \\ 2 \\ 3 \\ 5 \end{array} \right] \end{array} = \begin{array}{c} f_1 + f_4 + f_2 \\ \left[\begin{array}{c} 12 \\ 4 \\ 5 \\ 9 \\ 15 \end{array} \right] \end{array} \quad \begin{array}{c} \text{rankeamento} \\ \Rightarrow \end{array} \begin{array}{c} \text{ranking} \\ \left[\begin{array}{c} 4 \\ 1 \\ 2 \\ 3 \\ 5 \end{array} \right] \end{array}$$

A matriz resultante normalizada é apresentada a seguir, no qual as colunas 1 e 2 representam o *ranking* para os objetivos f_3 e ($f_4 + f_1 + f_2$), respectivamente.

$$Normalizada = \begin{bmatrix} 4 & 4 \\ 3 & 1 \\ 5 & 2 \\ 1 & 3 \\ 2 & 5 \end{bmatrix}$$

O problema, portanto, é composto agora por apenas dois objetivos. O conflito entre estes objetivos é aproximadamente 83.3333%, ou seja, a harmonia entre eles se aproxima de 16.6667%. Assim, a árvore resultante inclui um novo nó que representa a agregação dos quatro objetivos do problema. Esta é apresentada na Figura 4.14.

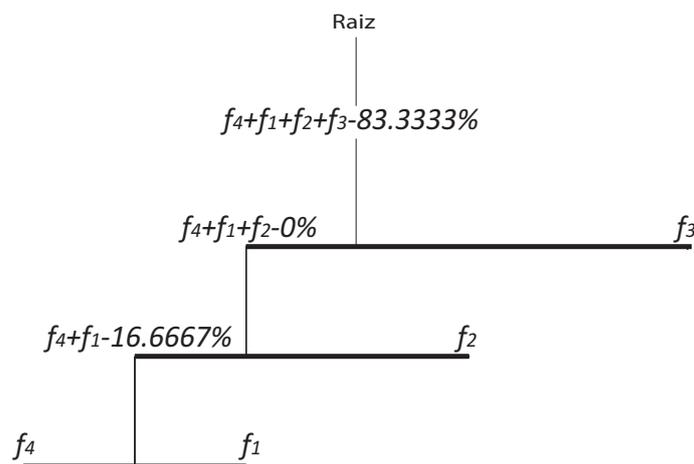


Figura 4.14: Estrutura da árvore com a agregação dos objetivos $f_4 + f_1 + f_2$ e f_3

4.4 Conclusão

Este capítulo apresentou o procedimento realizado por três algoritmos propostos para resolver o MOPRV. Assim, um NSGA-III é implementado para otimizar o MOPRV com todos os objetivos (problema completo). As soluções otimizadas pelo NSGA-III são então utilizadas para verificar a harmonia e o conflito entre os objetivos do problema. A Árvore de Agregação foi executada para um grupo de soluções do MOPRV e a harmonia/conflito entre os objetivos foi observado. As agregações mais convenientes são representadas graficamente em uma árvore, demonstrando informações sobre a relação

entre os objetivos. Assim, é possível visualizar de forma fácil a relação entre os objetivos do MOPRV com um custo computacional baixo. Diante deste contexto, o NSGA-II foi implementado para resolver o MOPRV reduzido, ou seja, o problema com objetivos agregados que apresentaram a maior harmonia.

Capítulo 5

NSGA-II e NSGA-III para Resolução do MOPRV

Os NSGAs (NSGA-II e NSGA-III), descritos nas Seções 4.1 e 4.2, foram adaptados a fim de solucionar o MOPRV Completo, bem como para solucionar uma versão reduzida (versão com objetivos agregados) do problema. Assim, a representação dos indivíduos da população, os métodos de geração da população inicial, métodos de seleção, métodos de cruzamento e métodos de mutação são descritos nas subseções a seguir.

5.1 Representação de uma Solução

Para representar um indivíduo (solução), os NSGAs propostos (NSGA-II e NSGA-III) utilizam uma lista que armazena os consumidores na sequência em que eles devem ser visitados pela frota de veículos. Assim um indivíduo poderia ter a seguinte configuração: [2 1 4 3 6 7 5]. Percebe-se neste exemplo que a solução proposta pelo indivíduo é que os veículos visitem o consumidor 2, 1, 4, 3, 6, 7, e por fim, o consumidor 5, nesta mesma ordem. No entanto, os veículos são obrigados a retornar ao depósito caso alguma restrição seja violada.

Esta representação é única, e uma sequência de elementos só pode ser decodificada para uma solução. É possível notar que o último cliente visitado em uma rota i é ligado ao primeiro cliente visitado na rota $i + 1$ para formar a sequência de todas as rotas envolvidas. Porém, nenhum elemento que identifique o final de uma rota é considerado até o momento. Isto porque o delimitador de rota em uma sequência restringe a factibilidade

de grande parte dos filhos produzidos pelos operadores de cruzamento. Para decodificar a sequência em configurações de rota, é necessário inserir os valores de genes em novas rotas sequencialmente. Há uma chance de que as rotas originais não sejam recuperadas após a decodificação, porém é geralmente assumido que minimizar o número de rotas ajuda a minimizar o custo total de viagem. Assim, determinar uma rota que atenda os clientes baseado na capacidade máxima dos veículos implica um potencial de solução bom como resultado (Nazif e Lee, 2012; Tan et al., 2001b; Tasan e Gen, 2012).

Deste modo, ao fazer a leitura do indivíduo os algoritmos propostos incluem o número zero (representação do depósito) no início da lista, garantindo que a primeira rota seja iniciada no depósito. Visto que o problema possui as restrições de capacidade de carga máxima, existe a possibilidade de um determinado veículo, ao atender o próximo consumidor, violar alguma restrição do problema. Caso isso ocorra, o número zero é inserido entre o último consumidor atendido pelo veículo e o consumidor subsequente da lista, garantindo que o veículo atual finalize sua rota no depósito. Por fim, o número zero também é incluso no final da lista, garantindo que o último veículo alocado para fazer o atendimento finalize sua rota no depósito central. Este processo garante que nenhuma restrição seja violada, garantindo também a factibilidade das soluções geradas.

Para o indivíduo $s = [2 \ 1 \ 4 \ 3 \ 6 \ 7 \ 5]$, por exemplo, após a leitura deste indivíduo, poderíamos ter a seguinte configuração: $s = [0 \ 2 \ 1 \ 4 \ 3 \ 0 \ 6 \ 7 \ 5 \ 0]$. Assim, o veículo parte do depósito e passa pelos consumidores 2, 1, 4 e 3, nesta ordem, e volta ao depósito por perceber que violaria alguma restrição do problema ao tentar atender o consumidor 6. No passo seguinte o veículo parte novamente do depósito atendendo aos consumidores 6, 7 e 5, e retorna ao depósito, chegando ao fim da lista de consumidores. Contudo, o depósito é inserido na lista de consumidores a fim de garantir que restrições de capacidade máxima dos veículos não sejam violadas.

5.2 Geração de uma Solução Inicial

Para gerar uma solução inicial, são utilizados dois métodos distintos. O primeiro é uma adaptação da heurística de Inserção Mais Barata (Karg e Thompson, 1964; Solomon, 1987), a qual constrói uma solução rota a rota. Inicialmente, gera-se uma rota r contendo um cliente escolhido aleatoriamente. Em seguida, calcula-se o custo de inserção c_{ij}^k de cada cliente k , que ainda não está presente na solução, entre cada par de clientes i e j da rota r . Entretanto, o custo c_{ij}^k é considerado pelos NSGAs como o somatório do cálculo

de cinco funções objetivo propostas no MOPRV para o trecho de rota formado até o momento. Assim, o custo da rota corrente é igual ao somatório dos seguintes objetivos: distância total percorrida (f_1), violação da restrição de tempo (f_3), tempo de espera dos veículos (f_4), maior rota (f_5) e balanceamento de rota (f_6). A quantidade de rotas necessárias para o atendimento (f_2) não é inserido no cálculo do custo de inserção, dado que este processo determina indiretamente o número de rotas da solução.

Deste modo, suponha que temos a seguinte rota em construção: (0 3 4 0), e que a inserção do cliente 5 entre os clientes 3 e 4 é considerada. Neste caso, calcula-se os valores das funções objetivo f_1 , f_3 , f_4 , f_5 e f_6 para trecho de rota (0 3 5 4 0). Em seguida, é feito o somatório dos valores destas funções, este somatório, por sua vez, representa o custo de inserção do consumidor 5 entre os consumidores 3 e 4. Assim, o cliente k que tiver o menor custo de inserção na posição entre os clientes i e j da rota, é adicionado a rota exatamente nesta posição. A inserção é feita desde que não viole alguma restrição do problema. Caso a inserção de qualquer cliente implique em alguma violação de restrição do problema, a rota corrente é finalizada e inicia-se a construção de uma nova rota. Esse procedimento é repetido até que todos os clientes sejam adicionados na solução.

O segundo método de construção de uma solução inicial gera soluções completamente aleatórias. Este método gera uma rota r e escolhe a cada iteração um cliente, que ainda não foi escolhido em iterações anteriores, de forma aleatória para compor rota corrente. A inserção é feita desde que não viole nenhuma restrição do problema. Caso a inserção de um cliente viole alguma restrição do problema, a rota corrente é finalizada e inicia-se a construção de uma nova rota. A solução é completamente formada quando todos os clientes são escolhidos para compor alguma rota. Este método de geração auxilia no processo de formular soluções com uma maior diversidade, evitando que o processo de otimização fique preso em ótimos locais rapidamente.

5.3 Operadores de Cruzamento

A população filha é gerada a partir da recombinação de indivíduos da população pai. Primeiramente, dois indivíduos pais são selecionados aleatoriamente e os indivíduos filhos são gerados pela aplicação de um operador de cruzamento. Para cada par de indivíduos pais selecionados existe uma probabilidade de cruzamento. Dois operadores de cruzamento são considerados para resolver o MOPRV. Estes são: Cruzamento de Mapeamento Parcial (*Partial Mapping Crossover* - PMX) e Cruzamento de Ordem (*Order Crossover*

- OX). Ambos operadores de cruzamento têm igual probabilidade de serem selecionados para geração de um par de filhos.

5.3.1 Cruzamento de Mapeamento Parcial

O operador PMX (Sivanandam e Deepa, 2007) transfere informações de ordem e de posição das rotas dos pais para as rotas dos filhos. Uma parte da sequência de um pai é mapeada e uma parte da sequência do outro pai é preservada no filho, o restante das informações é trocada entre os pais (Malaquias, 2006).

Utilizando-se como exemplo a sequência (1 2 3 4 5 6 7 8) como a rota do pai 1 e (3 7 5 1 6 8 2 4) como a rota do pai 2, o operador PMX inicialmente seleciona dois pontos de corte aleatoriamente para a divisão dos pais. Partindo do princípio que o primeiro ponto de corte está entre o terceiro e quarto elemento (gene), e o segundo ponto de corte entre o sexto e sétimo elemento, os pais são representados como segue na Figura 5.1.

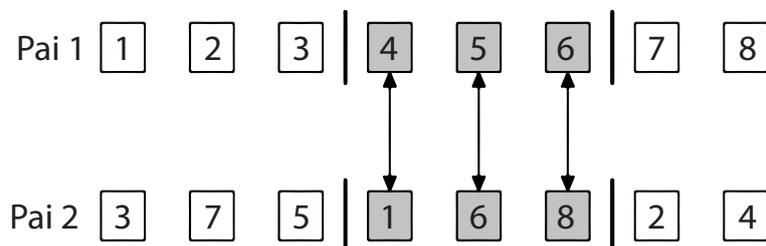


Figura 5.1: Exemplo da seção de mapeamento do PMX (Malaquias, 2006)

No passo seguinte o PMX copia os genes da segunda parte do pai 1 para a segunda parte do filho 2. Da mesma forma, copiam-se os genes da segunda parte do pai 2 para a segunda parte do filho 1. O conjunto de genes pertencentes à segunda parte dos indivíduos é denominado de seção de mapeamento. A Figura 5.2 ilustra o processo de cópia da seção de mapeamento.

Por fim, o PMX copia o restante dos genes do pai 1 para as respectivas posições do filho 1, começando pelo primeiro elemento da terceira parte. Caso o pai 1 tente inserir algum gene já existente no filho 1, o PMX verifica a posição deste gene no filho e, tenta inserir outro gene do pai correspondente a essa posição. Isso é feito até que se encontre um gene inédito para o filho.

No exemplo, o primeiro elemento da terceira seção do pai 1 é copiado para o filho 1 (cliente 7), em seguida o segundo elemento da terceira seção do filho 1 seria o cliente

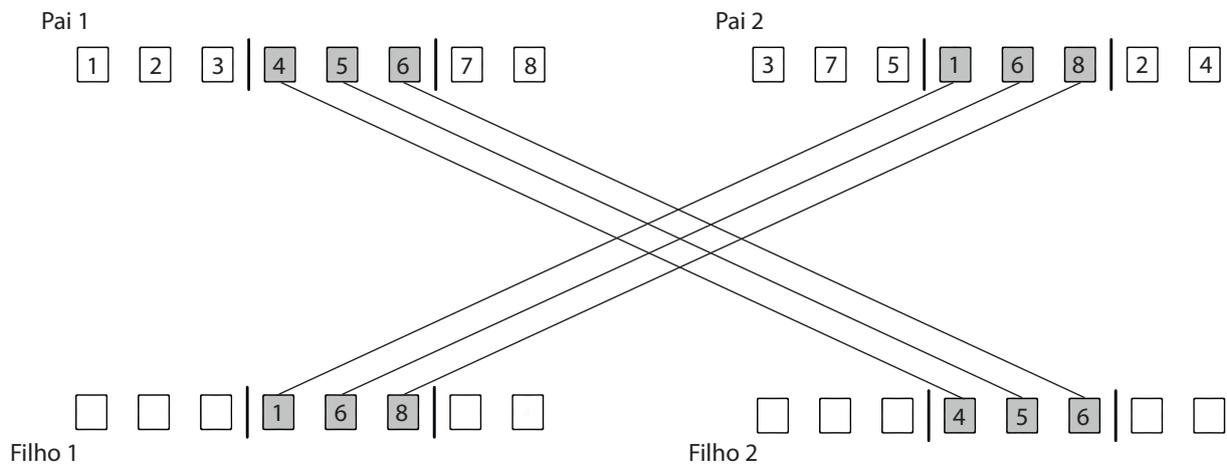


Figura 5.2: Cópia dos elementos da seção de mapeamento dos pais para os filhos (Malaquias, 2006)

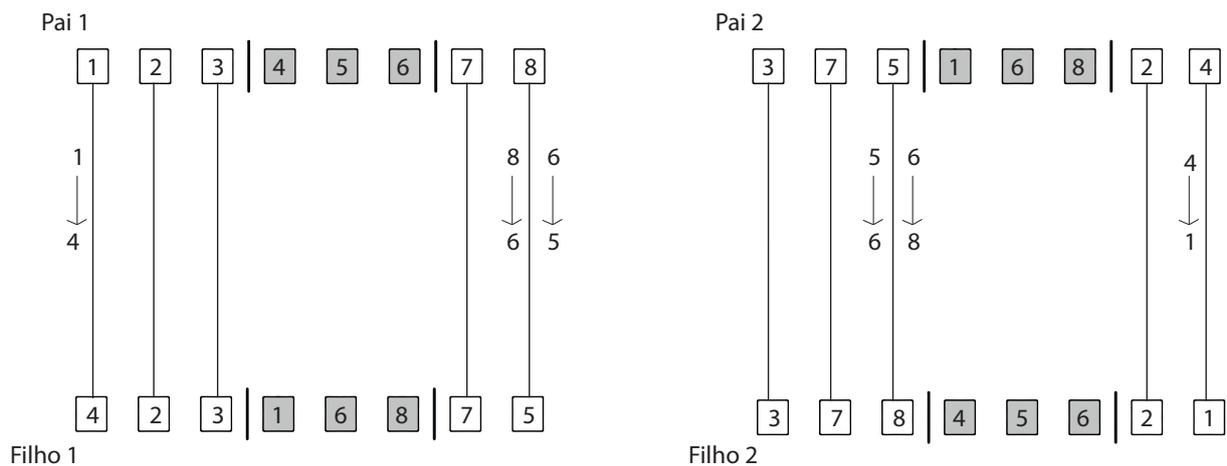


Figura 5.3: Cópia dos elementos restantes dos pais para os filhos (Malaquias, 2006)

8. Entretanto, o cliente 8 já está presente neste filho (é o sexto elemento proveniente do pai 2). Assim, usando os mapeamentos definidos, encontra-se $8 \leftrightarrow 6$, o que leva a escolher o cliente 6 para a segunda posição da terceira seção. De maneira semelhante ao cliente 8, o cliente 6 está presente no filho 1 (é o quinto elemento proveniente do pai 2). A partir disto encontra-se $6 \leftrightarrow 5$, o cliente 5 então é colocado na oitava posição do filho 1. No passo seguinte, o primeiro elemento do primeiro filho seria o cliente 1, que já está presente no mesmo. Considerando o mapeamento $1 \leftrightarrow 4$, o cliente 4 é alocado como o primeiro elemento pertencente ao filho. O segundo e o terceiro elementos do primeiro filho podem ser tomados diretamente do primeiro pai, pois estes ainda não

estão presentes no primeiro filho. Analogamente forma-se o segundo filho. A Figura 5.3 ilustra este processo.

5.3.2 Cruzamento de Ordem

O operador OX (Sivanandam e Deepa, 2007) explora a propriedade de representação do caminho, em que a ordem é importante e não a posição. Ele escolhe um subroteiro de um dos pais preservando a ordem dos elementos do outro pai. Este método constrói um filho escolhendo um subroteiro de um dos pais e preservando a ordem relativa dos consumidores do outro pai (Malaquias, 2006).

Assim como no PMX o operador OX inicia gerando dois pontos de corte aleatoriamente e copiando as seções de mapeamento dos pais para os filhos. Este processo é ilustrado na Figura 5.4.

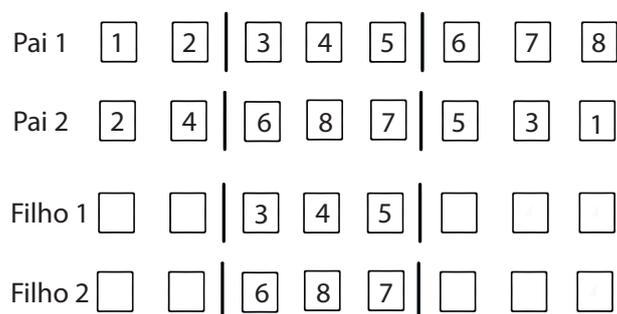


Figura 5.4: Resultado da cópia da seção de mapeamento nos filhos (Malaquias, 2006)

Assim, o operador OX copia o restante dos elementos dos pais para as respectivas posições dos filhos, começando pelo primeiro elemento da terceira parte. A diferença entre o PMX e o OX é que, caso o pai tente inserir algum elemento já existente no filho, o operador OX busca o próximo elemento a direita contido no pai, até que se encontre um elemento inédito para o filho. No exemplo, o primeiro elemento da terceira seção do filho 1 seria o cliente 5, porém este já está contido no filho 1. O próximo elemento do pai a direita do cliente 5 é o cliente 3 que também pertence ao filho. Neste caso, primeiro elemento da terceira seção do filho 1 recebe o cliente 1, dado que este ainda não está contido no filho. Em seguida os clientes 2, 6, 8 e 7 são inseridos no filho 1. Analogamente forma-se o segundo filho. A Figura 5.5 ilustra os filhos formados pelo método OX.

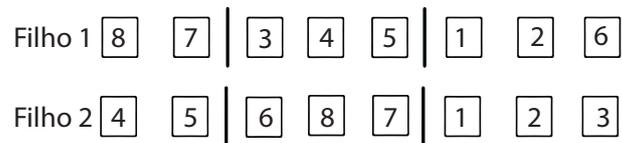


Figura 5.5: Resultado de execução do OX (Malaquias, 2006)

5.4 Operadores de Mutação

Cada indivíduo filho gerado pelos métodos de *crossover* pode sofrer mutação dada uma probabilidade. Os operadores de mutação utilizados neste trabalho são: Mutação por Inserção (*Insertion Mutation* - ISM), Mutação por Inversão Simples (*Simple Inversion Mutation* - SIM) e Mutação por Troca (*Exchange Mutation* - EM). Estes métodos têm igual probabilidade de serem selecionados para o cruzamento.

5.4.1 Mutação por Inserção

O operador ISM (Deep e Mebrahtu, 2011) escolhe aleatoriamente um elemento na sequência, remove este elemento e o insere em uma posição escolhida aleatoriamente. Tome como exemplo um indivíduo que tenha a seguinte sequência de genes (1 2 3 4 5 6 7 8) e que o quarto gene foi escolhido para ser inserido na sétima posição, o resultado da mutação é (1 2 3 5 6 7 4 8). A Figura 5.6 ilustra o processo feito pelo ISM.

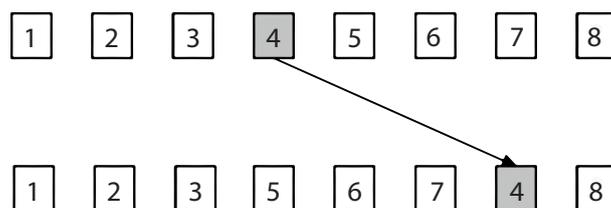


Figura 5.6: Representação do processo feito pelo ISM (Malaquias, 2006)

5.4.2 Mutação por Inversão Simples

O operador SIM (Deep e Mebrahtu, 2011) seleciona aleatoriamente dois pontos de corte e inverte os elementos do subconjunto formado a partir dos pontos de corte. Suponhamos que o indivíduo selecionado tenha a seguinte sequência de genes (1 2 3 4 5 6 7 8) e que o primeiro e o segundo ponto de corte estão entre o primeiro e o segundo gene, e o

quinto e o sexto gene, respectivamente. Invertendo o subconjunto (2 3 4 5) o resultado da mutação é (1 5 4 3 2 6 7 8). A Figura 5.7 ilustra o processo feito pelo SIM.

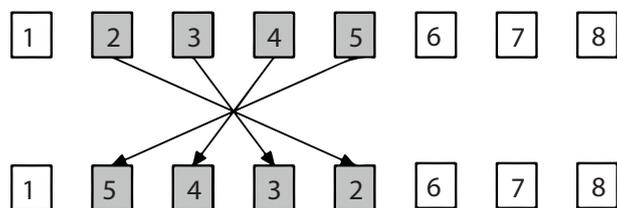


Figura 5.7: Representação do processo feito pelo SIM (Malaquias, 2006)

5.4.3 Mutação por Troca

O operador EM (Deep e Mebrahtu, 2011) seleciona aleatoriamente dois genes no indivíduo e troca suas posições. Considerando a sequência (1 2 3 4 5 6 7 8) como exemplo e que os genes da terceira e da quinta posição foram selecionados, tem-se (1 2 5 4 3 6 7 8) como resultado da mutação. A Figura 5.8 ilustra o processo feito pelo EM.

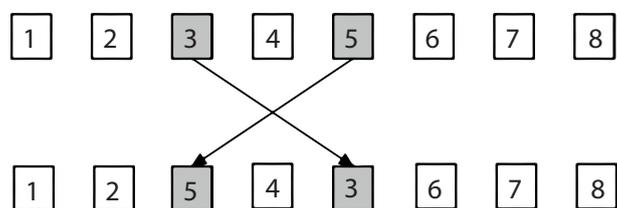


Figura 5.8: Representação do processo feito pelo EM (Malaquias, 2006)

5.5 Operador de Seleção

O NSGA-II possui dois processos que envolvem a seleção de indivíduos. O primeiro processo seleciona os indivíduos da população pai e os coloca em uma população auxiliar (seleção para reprodução). Esta seleção é executada para que as “melhores” soluções tenham maior probabilidade de passar pelo processo de cruzamento e mutação. O segundo processo de seleção de uma iteração do NSGA-II faz uso do FNDS e da Distância de multidão para formar a população da geração seguinte (seleção para sobrevivência), este processo é descrito na Seção 4.1. Assim, o método de seleção para reprodução implementado no NSGA-II, é o método de seleção por torneio de multidão. Este método escolhe

aleatoriamente dois indivíduos presentes em uma população P e os compara da seguinte forma: uma solução i , pertencente a fronteira F_i após a ordenação de dominância, é considerada melhor que a solução j , pertencente a fronteira F_j , se:

- i possui nível de não dominância melhor que j , ou seja, se $F_i \prec F_j$;
- $F_i = F_j$ e a distância de multidão de i é maior que a distância de multidão de j .

Aquele indivíduo que apresentar o melhor nível de não dominância seguido da melhor distância de multidão é copiado e inserido em uma população auxiliar. Os indivíduos comparados permanecem na população P , podendo ser escolhidos novamente para o torneio. Assim, a população Q é formada a partir dos filhos gerados pelos indivíduos da população auxiliar, podendo estes sofrer o processo de mutação. O NSGA-III não faz uso de nenhum método de seleção para reprodução explícita para gerar a população filha. Neste algoritmo, os indivíduos são escolhidos de forma aleatória. Esta população é, portanto, construída através da aplicação de operadores de cruzamento e de mutação habituais, escolhendo aleatoriamente os indivíduos da população pai.

5.6 Pontos de Referência

Como indicado anteriormente, o NSGA-III utiliza um conjunto pré-definido de pontos de referência para assegurar a diversidade em soluções obtidas. Assim, o algoritmo além de priorizar solução não dominadas, também enfatiza os membros da população que são de algum modo associados com cada ponto de referência. Uma vez que os pontos de referência são criados, estes são amplamente distribuídos em todo o hiperplano normalizado. No entanto, o NSGA-III utilizado para resolver o MOPRV adota uma abordagem sistemática proposta por Das e Dennis (1998) que insere pontos em um hiperplano normalizado ($M-1$ dimensões) que é igualmente inclinado para todos os eixos objetivo.

Assim, se p divisões são consideradas ao longo de cada eixo objetivo, o número total de pontos de referência (H) em um problema com M objetivos é dado por:

$$H = \binom{M + p - 1}{p} \quad (5.1)$$

No entanto, o algoritmo proposto verifica o valor p e divide cada eixo objetivo em p divisões uniformemente espaçadas. Sabendo-se que p é número de divisões de cada eixo objetivo, qualquer coordenada dos pontos de referência gerados recebem valores iguais ao do conjunto $PF = \{\frac{0}{p}, \dots, \frac{p}{p}\}$. Deste modo, considerando apenas os valores de PF , a abordagem de Das e Dennis (1998) gera todos os pontos possíveis que somando-se os valores das coordenadas de um ponto o resultado é igual a 1.

Tome como exemplo um problema biobjetivo com as funções f_1 e f_2 . Considere ainda que 3 divisões ($p = 3$) são consideradas em cada eixo objetivo. Temos o seguinte cálculo:

$$H = \binom{2 + 3 - 1}{3} = \binom{4}{3} = \frac{4!}{3!(4-3)!} = \frac{24}{6} = 4$$

Quatro pontos de referência são então necessários para dividir a reta que une os objetivos. Como $p = 3$, as coordenadas x e y destes pontos recebem valores iguais a $0/3 = 0$, $1/3 = 0.3333$, $2/3 = 0.6666$ ou $3/3 = 1$. As combinações de coordenadas possíveis, tal que o somatório das coordenadas de qualquer ponto se iguale a 1 são: $(1,0)$, $(0,1)$, $(0.3333,0.6666)$ e $(0.6666,0.3333)$. Este processo é ilustrado na Figura 5.9.

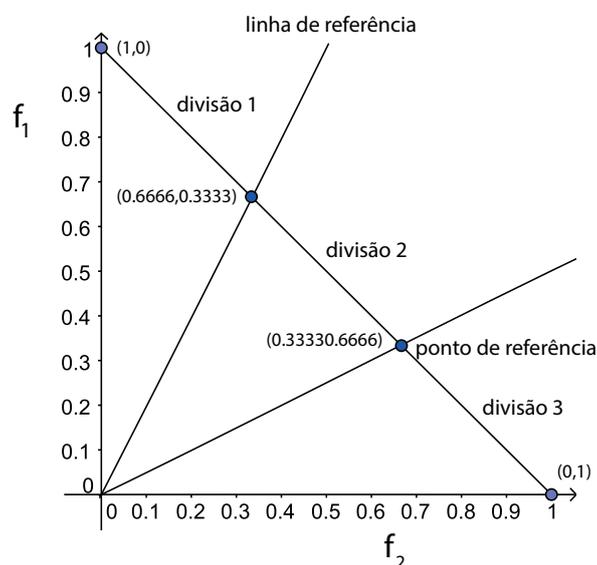


Figura 5.9: Pontos de referência em duas dimensões

Para um problema com três objetivos, f_1 , f_2 e f_3 por exemplo, os pontos de referência são criados em um triângulo formado pelos segmentos de reta que ligam os objetivos f_1 e f_2 , f_2 e f_3 , e f_1 e f_3 . Se quatro divisões ($p = 4$) são escolhidas para cada eixo objetivo,

$H = 15$, ou seja, 15 pontos de referência serão criados. As coordenadas x , y e z podem então receber os valores $0/4 = 0$, $1/4 = 0.25$, $2/4 = 0.5$, $3/4 = 0.75$ ou $4/4 = 1$. A Figura 5.10 ilustra todos os pontos de referência, para 4 divisões e 3 objetivos, que se somado os valores das coordenadas destes pontos o resultado é 1.

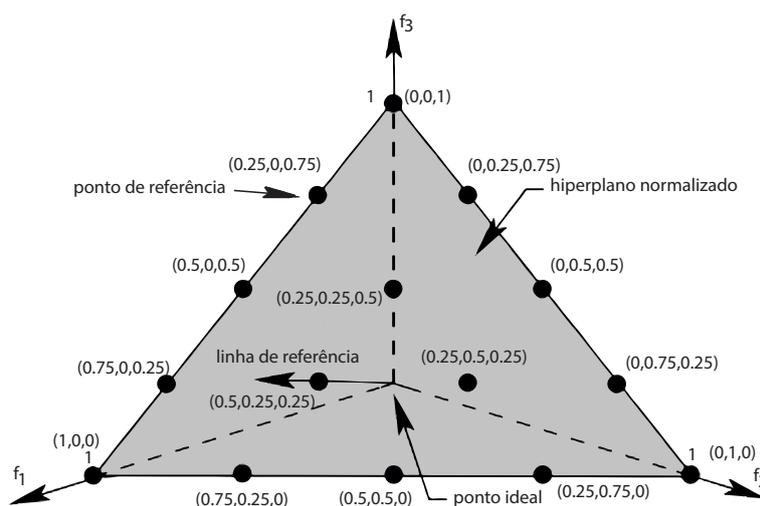


Figura 5.10: Pontos de referência em três dimensões (Deb e Jain, 2014)

Uma vez que o NSGA-III enfatiza os membros da população que são de algum modo associados com cada um destes pontos de referência, o algoritmo tende a encontrar soluções aproximadas de Pareto-Ótimo correspondentes aos pontos de referência fornecidos. Assim, o NSGA-III, do ponto de vista de uma aplicação, combina tomada de decisões e otimização de muitos objetivos. Como não temos nenhuma preferência sobre as soluções, a estratégia proposta por Das e Dennis (1998) apresenta um comportamento eficiente. Isto porque, as soluções da aproximação de Pareto-Ótimo tendem a ser bem diversificadas nesta abordagem.

5.7 Conclusão

No entanto, os NSGAs propostos consideram um indivíduo como uma lista de inteiros que representam os clientes na ordem em que eles devem ser visitados. Os métodos de cruzamento de mapeamento parcial e cruzamento de ordem, bem como os métodos de mutação por inserção, mutação por inversão simples e mutação por troca foram apresentados e detalhados. Assim, para a geração da população filha, o NSGA-II utiliza métodos de cruzamento, mutação e seleção para a reprodução, enquanto o NSGA-III

não faz uso de nenhum método de seleção para a reprodução. Deste modo, a seleção por torneio de multidão foi implementada no NSGA-II. Ainda neste capítulo, o processo que gera os pontos de referência para o NSGA-III foi descrito.

Capítulo 6

Resultados

Este capítulo apresenta os resultados dos principais algoritmos propostos neste trabalho para solucionar o problema de roteamento de veículos com muitos objetivos e janelas de tempo flexíveis. Os experimentos realizados consistem na execução dos algoritmos: *Nondominated Sorting Genetic Algorithm III* (NSGA-III), Árvore de Agregação e *Nondominated Sorting Genetic Algorithm II* (NSGA-II).

Um conjunto de 56 problemas da literatura (Solomon, 1987) foi submetido aos algoritmos propostos. Para formular uma possível redução do MOPRV Completo, um planejamento experimental foi utilizado, possibilitando identificar os objetivos com maior harmonia. Para corroborar a qualidade das soluções apresentadas, as soluções da aproximação do conjunto Pareto do NSGA-II foram comparadas às soluções da primeira frente de Pareto retornadas pelo NSGA-III.

Este capítulo está organizado em 6 seções. A primeira descreve as características das instâncias de teste utilizadas para a redução do problema completo. As medidas de distância, bem como de tempo utilizadas nos cálculos das funções objetivo são descritos na Seção 6.2. Os parâmetros dos algoritmos são apresentados na Seção 6.3. A Seção 6.4 apresenta os detalhes do planejamento experimental. A Seção 6.5 apresenta e discute os resultados obtidos pelos algoritmos propostos para resolver e reduzir o MOPRV. Por fim, as conclusões do capítulo são apresentadas na Seção 6.6.

6.1 Instâncias

Instâncias são problemas-teste criados para um determinado problema, servindo de base para a avaliação e comparação de métodos para solucioná-los. Neste trabalho, é utilizado um conjunto de instâncias propostas por Solomon (1987) contendo 100 clientes e um depósito central. As instâncias armazenam informação referente à localização dos consumidores, suas demandas de entrega, capacidade dos veículos, tempo de serviço dos veículos nos consumidores (representa o tempo de descarregamento da carga) e, ainda, as janelas de tempo dos clientes. A localização dos consumidores é dada pelas coordenadas (x, y) . As instâncias não apresentam restrição quanto ao número de veículos utilizados, sendo considerada uma frota com um número ilimitado de veículos.

As instâncias de Solomon (1987) apresentam seis conjuntos de problemas. Estas instâncias destacam vários fatores que afetam o comportamento do encaminhamento e agendamento de algoritmos de otimização. Estes fatores são: dados geográficos; o número de clientes atendidos por um veículo; e variação do tamanho e do posicionamento das janelas de tempo. Os dados geográficos, ou a localização dos clientes, são gerados aleatoriamente nos conjuntos de problemas R1 e R2, agrupados nos conjuntos de problemas C1 e C2, e uma combinação de estruturas aleatórias e agrupadas nos conjuntos de problemas RC1 e RC2. O conjunto de problemas R1, C1 e RC1 têm um curto horizonte de programação e permite que apenas alguns clientes sejam atendidos por rota. Por outro lado, os conjuntos R2, C2 e RC2 têm um horizonte de programação longo, permitindo que muitos clientes sejam atendidos pelo mesmo veículo. As Figuras 6.1(a), 6.1(b) e 6.1(c) ilustram a distribuição dos consumidores para os grupos de instância R, C e RC, respectivamente.

As coordenadas dos clientes são idênticas para todos os problemas dentro das classes R, C e RC. Os problemas se diferem no que diz respeito à largura das janelas de tempo. Algumas instâncias têm janelas de tempo muito curtas, enquanto outras têm janelas de tempo que dificilmente funcionam como restrição. Assim, os dados das instâncias podem ser representados por um grafo completo, ou seja, existe uma aresta conectando cada par de clientes do problema. As instâncias dos grupos R, C e RC somam um total de 56 instâncias testadas neste trabalho.

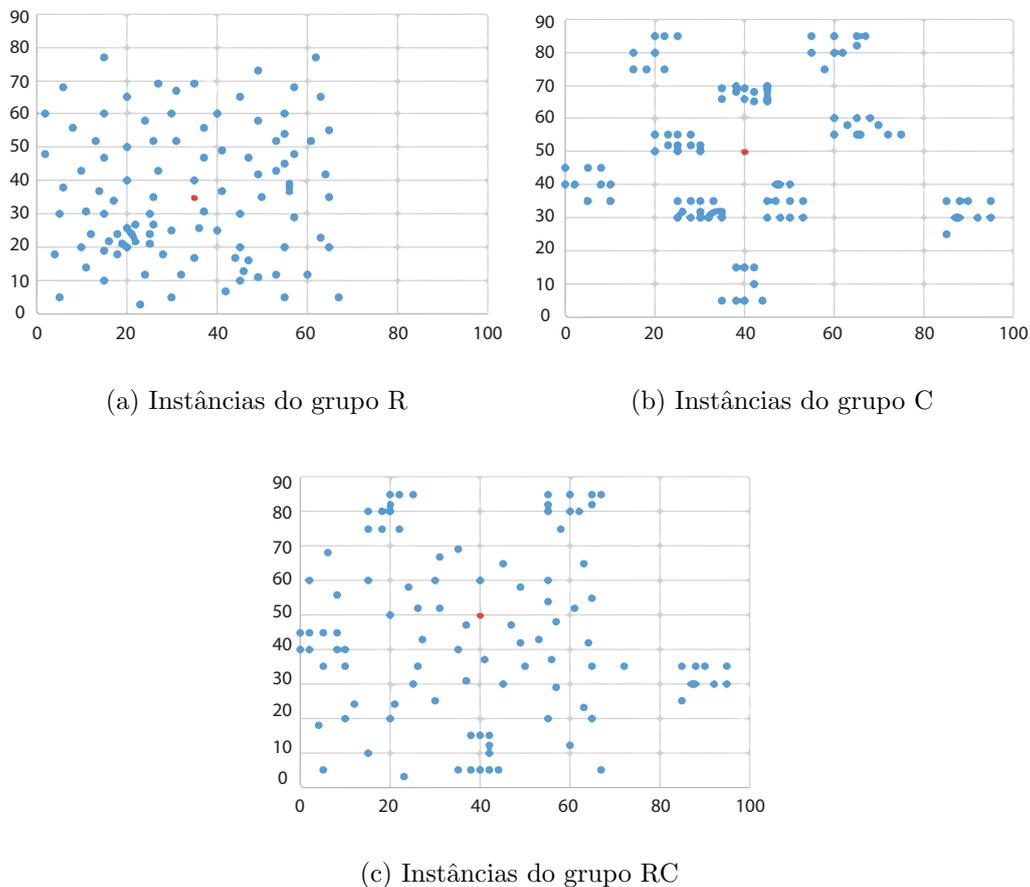


Figura 6.1: Disposição dos consumidores nos três grupos de instâncias de Solomon (1987)

6.2 Avaliação dos Objetivos

Visto que as instâncias de Solomon (1987) distribuem os clientes em um plano cuja localização dos mesmos são representadas por uma coordenada (x, y) , a distância percorrida por um veículo entre dois consumidores é dada pela distância Euclidiana entre os pontos. Entretanto, os tempos de viagem entre dois clientes se iguala às distâncias correspondentes. Isto é, o tempo de percurso entre dois clientes é proporcional à distância entre os mesmos. As instâncias de teste ainda disponibilizam as janelas de tempo de cada consumidor. Para calcular o tempo de espera dos veículos nos consumidores, basta calcular a diferença entre o tempo de início da janela de tempo e o tempo que o veículo chega ao consumidor. Assim, o somatório do tempo de espera representa a distância que os veículos poderiam percorrer, caso estes não estivessem parados. Deste modo,

o tempo necessário para finalizar uma rota é dado pela soma do tempo de espera dos veículos e a distância total percorrida. A violação da restrição de tempo é calculada de forma semelhante ao objetivo que calcula o tempo de espera dos veículos. As instâncias de teste disponibilizam o tempo de fim da janela de tempo de cada consumidor. Assim, a violação é calculada pela diferença entre o tempo que o veículo chega ao consumidor e o tempo de fim da janela de tempo.

Já o cálculo dos tamanhos das rotas para os objetivos que verificam o *makespan* e o balanceamento não consideram o tempo de espera dos veículos nos consumidores. Isto é, o tamanho de uma rota é calculado apenas sobre a distância Euclidiana entre os pontos (consumidores) de uma mesma rota. Não incluir o tempo de espera no cálculo do tamanho das rotas é justificado pelo fato de que a espera dos veículos nos clientes é um objetivo já abordado no problema. Caso o tamanho da rota considerasse este tempo de espera, estaríamos, de certa forma, agregando o objetivo f_4 (espera dos veículos) aos objetivos f_5 (*makespan*) e f_6 (balanceamento de rota) do MOPRV.

6.3 Parâmetros

Os algoritmos apresentados utilizam parâmetros que precisam ser ajustados para que o processo de busca pelas soluções seja realizado com eficiência. Os parâmetros responsáveis por esse comportamento e utilizados na implementação dos NSGAs são o tamanho da população de cada geração, critério de parada, probabilidade de cruzamento entre dois indivíduos, probabilidade de mutação de um indivíduo e a probabilidade de execução dos métodos de cruzamento e mutação.

Os parâmetros para os NSGAs são apresentados na Tabela 6.1. Como o objetivo principal é a análise do comportamento do problema proposto, seja por otimizar os objetivos separadamente ou de forma agregada, apenas uma configuração foi definida para ambos algoritmos. A única diferença entre os parâmetros dos NSGAs se refere ao tamanho da população de cada geração. Isto porque no NSGA-II o tamanho da população é um parâmetro de entrada do algoritmo, enquanto no NSGA-III o tamanho da população é o menor múltiplo de quatro maior que o número de pontos de referência gerados.

Os parâmetros foram ajustados de forma empírica após testes preliminares, com exceção do número de partições por eixo objetivo para a geração dos pontos de referência.

Entretanto, segundo Deb e Jain (2014) os pontos de referência não são parâmetros do algoritmo, pois tanto a quantidade como a localização destes pontos estão inteiramente à disposição do usuário. Assim, seis partições foram consideradas pelo NSGA-III para cada eixo objetivo do problema. O número de partições por eixo objetivo foi escolhido tendo como base o estudo feito por Deb e Jain (2014) que utilizou seis partições por eixo para um problema com cinco objetivos.

Tabela 6.1: Parâmetros

Parâmetro	NSGA-II	NSGA-III
Tamanho da População	464	464
Número máximo de gerações	500	500
Probabilidade de cruzamento	0,95	0,95
Probabilidade de mutação	0,1	0,1
Probabilidade para o PMX	0,5	0,5
Probabilidade para o OX	0,5	0,5
Probabilidade para o ISM	0,33	0,33
Probabilidade para o SIM	0,33	0,33
Probabilidade para o EM	0,33	0,33
Partições por eixo objetivo	-	6

Assim, 462 pontos de referência são gerados para cada execução do NSGA-III. Isso resulta em uma população composta por 464 indivíduos. Já a população de cada geração do NSGA-II contém 300 indivíduos. O critério de parada é o número máximo de gerações dos NSGAs. Cada par de pais escolhido para o cruzamento tem 95% de chance de gerar seus filhos, sendo que os métodos PMX e OX tem 50% de chance de serem escolhidos no processo. Apenas um método de cruzamento é executado para cada par de pais. De forma semelhante, cada indivíduo selecionado para o processo de mutação tem 10% de chance de ser mutado e apenas um método de mutação é executado para cada indivíduo selecionado. Assim, os métodos ISM, SIM e EM tem 33,33% de chance de serem escolhidos.

6.4 Planejamento Experimental

Para a execução dos experimentos, foi utilizado um conjunto de 56 instâncias propostas por Solomon (1987). O problema completo contém 6 objetivos, sendo estes relacionados ao custo de transporte, número de veículos, atraso no atendimento das demandas, espera dos veículos nos consumidores, *makespan* e balanceamento de rota, como descrito na Seção 3.7.2. Assim, o NSGA-III proposto será executado quatro vezes para cada instância testada, de modo que os seis objetivos do problema são otimizados separadamente. Cada execução do NSGA-III gera 464 soluções. Em um total de quatro execuções, 1856 soluções otimizadas serão geradas pelo NSGA-III para cada instância de Solomon (1987).

Assim, as soluções otimizadas pelo NSGA-III, bem como dez mil soluções geradas de forma aleatória são utilizadas como entrada para as Árvores de Agregação. É através destas soluções que a ferramenta analisa e propõe a agregação dos objetivos mais harmoniosos para cada instância de teste. No entanto, para cada instância testada duas árvores de agregação foram geradas. A primeira demonstrando a harmonia e conflito entre os objetivos para soluções aleatórias (soluções que não passaram pelo processo de otimização), e a segunda agregando os objetivos a partir das soluções otimizadas (soluções que passaram pelo processo de otimização através do NSGA-III). A fim de verificar a harmonia/conflito global para todas as instâncias, árvores médias foram calculadas. Neste processo, a média da matriz de harmonia do algoritmo Árvores de Agregação foi calculada e a agregação foi baseada na matriz de harmonia média.

Uma análise sobre as árvores geradas foi feita, e um padrão geral entre a harmonia dos objetivos para todas as instâncias foi observado. Assim, um modelo reduzido, com menos objetivos, do MOPRV é proposto e um NSGA-II que considera o problema reduzido é executado para todas as instâncias de Solomon (1987). Para cada instância testada o NSGA-II foi executado quatro vezes, resultando em quatro conjuntos aproximados de Pareto. O passo seguinte utiliza a métrica de Cobertura para verificar a qualidade das soluções retornadas para o problema reduzido. Esta métrica considera dois conjuntos de soluções e retorna a porcentagem de soluções dominadas de um conjunto quando comparadas com as soluções do outro conjunto.

6.5 Resultados e Discussões

Os testes foram executados em um computador com 1 processador Intel[®] Core[™] i5-3337U de 1.80 GHz, com 6 GB de memória RAM e sistema operacional Windows 8.1 de 64 bits. Os testes foram realizados a fim de verificar o conflito entre os objetivos do MOPRV, de modo que uma formulação reduzida do problema que otimiza os objetivos agregados fosse proposta com a menor perda na qualidade das soluções da frente de Pareto.

6.5.1 Agregação dos Objetivos

Como descrito anteriormente, dez mil soluções foram geradas como uma permutação aleatória de números inteiros (clientes) e 1856 soluções foram geradas com o NSGA-III para cada problema teste. As Árvores de Agregação foram executadas duas vezes para cada instância. Uma vez para as soluções aleatórias e uma vez para as soluções otimizadas. Dado que 56 instâncias são consideradas, 56 árvores resultantes de soluções aleatórias e 56 árvores resultantes de soluções otimizadas foram geradas, totalizando 112 árvores de agregação. Cada árvore de agregação, seja para soluções otimizadas ou aleatórias, apresenta o conflito e a harmonia para apenas uma instância específica. Assim, além de gerar árvores que permitem apenas a análise de uma única instância, duas outras árvores foram geradas tendo como base a média da harmonia de todas as instâncias. A primeira apresenta a árvore média para as soluções aleatórias enquanto a segunda apresenta a árvore média para soluções otimizadas.

As árvores de agregação médias permitem que a análise de um conjunto de instâncias seja feita observando apenas uma árvore de agregação que representa a média entre este conjunto de instâncias. Caso contrário, a única forma de verificar algum comportamento de agregação padrão para um conjunto de instâncias é analisar individualmente as árvores de agregação de cada uma destas instâncias. Entretanto, não desconsideramos nenhuma destas análises neste trabalho. Primeiro uma análise sobre as árvores médias é apresentada e a agregação dos objetivos nestas árvores é discutida. Em seguida, uma análise individual sobre cada instância testada é apresentada.

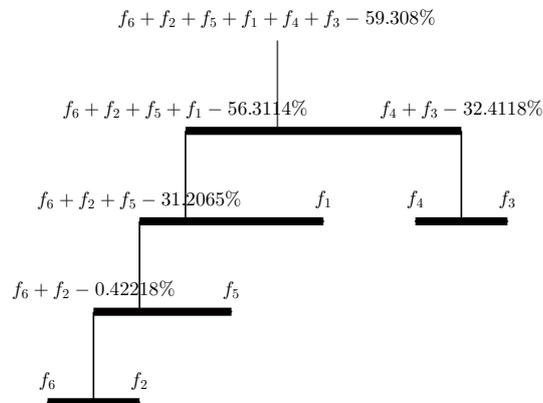
Árvores Médias

As árvores médias foram geradas a partir do cálculo da harmonia de todas as instâncias testadas. No entanto, o procedimento realizado para gerar as árvores médias considera uma matriz de harmonia média que é dada como base para a agregação dos objetivos com maior harmonia. Assim, o processo que calcula a árvore média primeiro calcula a matriz de harmonia para cada instância específica. Se temos 56 instâncias, uma iteração da árvore média calcula a matriz de harmonia das 56 instâncias separadamente. O passo seguinte faz a soma das 56 matrizes de harmonia que representam a harmonia entre os objetivos de cada uma das instâncias testadas. Por fim, cada elemento da matriz resultante desta soma é dividido pelo número de instâncias teste (56 instâncias), resultando em uma matriz que apresenta valores médios de harmonia (matriz de harmonia média) entre os objetivos do MOPRV para todas as instâncias. Assim, é considerando a matriz de harmonia média que uma iteração da árvore média agrega os objetivos do problema.

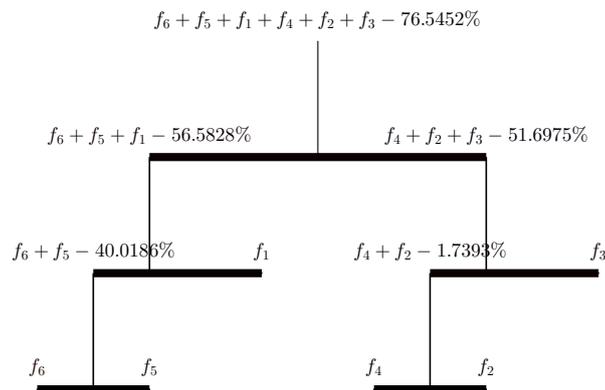
Diante disso, duas árvores médias que consideram as 56 instâncias testadas foram geradas. Estas árvores representam o resultado da harmonia média em que soluções aleatórias e otimizadas foram dadas como entrada. As árvores médias para soluções aleatórias e otimizadas são apresentadas nas Figuras 6.2(a) e 6.2(b). A visão geral das Árvores de Agregação acaba por demonstrar os principais comportamentos entre os objetivos, bem como a relação existente entre os objetivos agregados. No entanto, as árvores médias apresentaram comportamentos diferentes quando soluções otimizadas e aleatórias foram consideradas.

A árvore média gerada a partir de soluções aleatórias é formada por dois ramos principais. O ramo da esquerda forma uma subárvore que é composta pelos objetivos f_6 (diferença entre a maior e a menor rota), f_2 (número de rotas), f_5 (maior rota) e f_1 (distância percorrida). Já o ramo da direita representa uma subárvore formada pelos objetivos f_4 (espera dos veículos nos consumidores) e f_3 (violação da restrição de tempo). Assim, f_6 é agregado à f_2 na primeira iteração com um percentual muito baixo de conflito. O que significa que quase sempre o número de rotas não interfere no balanceamento de rota.

O passo seguinte, agrega f_5 ao objetivo composto $f_6 + f_2$. É possível notar que o conflito existente entre f_6 e f_2 é tão pequeno que agregar f_5 à $f_6 + f_2$, ainda assim, possui menos conflito do que agregar f_4 à f_3 , por exemplo. Diante disso e considerando esta árvore média, agregar f_5 à f_2 ou à f_6 é tão viável como agregar f_6 à f_2 . No entanto, minimizar a maior rota acaba, em muitos casos, por minimizar também a diferença entre



(a) Árvore de Agregação Média para Soluções Aleatórias



(b) Árvore de Agregação Média para Soluções Otimizadas

Figura 6.2: Árvores de Agregação Média

a maior e a menor rota ou o número de rotas. A relação entre f_5 e f_6 é de fato esperada, uma vez que minimizar a maior rota proporciona um melhor equilíbrio entre as mesmas.

O objetivo de violação da restrição de tempo (f_3) também possui pouco conflito quando comparado à espera dos veículos nos consumidores (f_4). Essa agregação reflete que quanto menos tempo um veículo perde esperando a abertura de um consumidor,

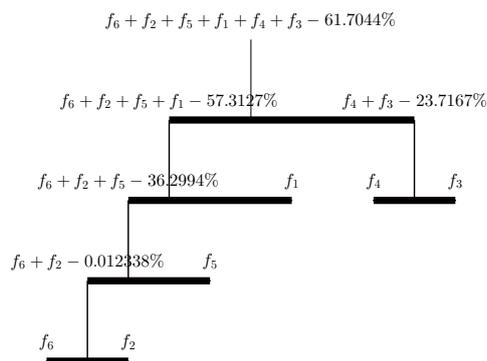
menor é a probabilidade de atender outros consumidores atrasado (depois do fim da janela de tempo). Dentre as agregações que envolvem algum objetivo original do problema (objetivo não agregado) a que possui maior conflito é a agregação entre f_1 e $f_6 + f_2 + f_5$, fazendo da distância total percorrida um dos objetivos com maior conflito quando comparado com os demais objetivos do problema. Por fim, o algoritmo agrega os objetivos compostos $f_6 + f_2 + f_5 + f_1$ e $f_4 + f_3$, resultando em um único objetivo composto que agrega todos os objetivos do problema.

A árvore média gerada a partir de soluções otimizadas apresenta uma estrutura diferente da árvore média das soluções aleatórias. A diferença para este caso está basicamente no comportamento do objetivo que considera a minimização do número de rotas. Para as soluções otimizadas f_2 possuiu uma menor média de conflito com f_4 , fazendo com que f_3 fosse agregado posteriormente e f_1 imediatamente agregado à $f_6 + f_5$. No entanto, algumas semelhanças podem ser observadas. Por exemplo, f_5 e f_6 continuam na subárvore da esquerda enquanto f_3 e f_4 permanecem na subárvore da direita. Isso enfatiza a possibilidade de que existe grande harmonia entre a violação da restrição de tempo e a espera dos veículos nos consumidores, bem como entre a maior rota e o balanceamento de rota. Por outro lado, se tratando de objetivos originais do problema, f_1 esteve envolvido na agregação com maior conflito para ambas as árvores médias. Este fato demonstra que minimizar a distância total percorrida possui pouca harmonia quando comparado aos demais objetivos do problema. Um comportamento diferente foi apresentado por f_2 para ambas árvores médias. Este, por sua vez, apresentou grande harmonia com f_6 e f_4 .

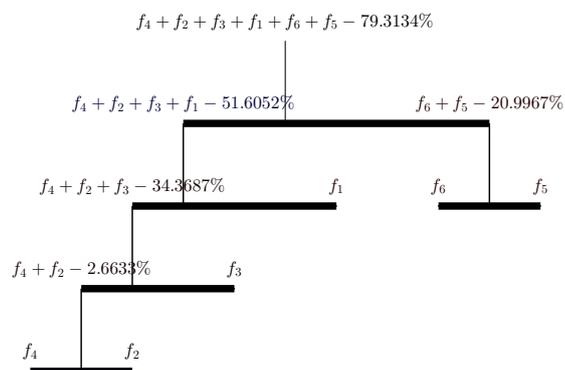
Assim, o comportamento de ambas árvores médias indicam as mesmas relações de conflito entre os grupos de objetivos. As agregações ocorrem em ordens diferentes, porém os grupos de objetivos são similares. Deste modo, pode-se inferir que o conflito/harmonia entre os objetivos do MOPRV é inerente ao espaço de busca, ou seja, os conjuntos Pareto aproximado possuem características semelhantes aos conjuntos de soluções aleatórias. A principal diferença da estrutura dos grupos de objetivos se dá pelo comportamento de f_2 , ora agregado à subárvore da esquerda (f_1 , f_5 e f_6) ora agregado a subárvore da direita (f_3 e f_4).

A Figura 6.3 apresenta árvores de agregação de instâncias específicas que obtiveram comportamento semelhante às das árvores médias. A Figura 6.3(a) ilustra a agregação dos objetivos do MOPRV no qual soluções aleatórias foram geradas para a instância C106. Esta árvore apresenta a mesma estrutura da árvore média gerada a partir de soluções aleatórias. Diversas outras árvores de instâncias específicas apresentaram a

mesma estrutura. Por outro lado, nenhuma instância específica apresentou estrutura igual à da árvore média gerada a partir de soluções otimizadas, apresentando apenas comportamentos parecidos. A árvore gerada para a instância RC105 a partir de soluções otimizadas, por exemplo, apresenta um comportamento semelhantes à da árvore média de soluções otimizadas. Esta árvore é ilustrada na Figura 6.3(b).



(a) Árvore de Agregação da Instância C106 Gerada a Partir de Soluções Aleatórias



(b) Árvore de Agregação da Instância RC105 Gerada a Partir de Soluções Otimizadas

Figura 6.3: Árvores de Agregação Específicas Semelhantes às Árvores Médias

Os gráficos polares da instância C106 gerada de soluções aleatórias e da instância RC105 gerada de soluções otimizadas são apresentadas pelas Figuras 6.4(a) e 6.4(b),

respectivamente. Cada linha dos gráficos polares representa uma solução que foi dada como entrada na Árvore de Agregação. No entanto, quanto mais externa é a linha em um determinado objetivo, melhor é o valor desta solução para este objetivo. Por outro lado, quanto mais a linha se aproxima do centro dos gráficos em um objetivo, pior é o valor desta solução para este objetivo. Assim, é possível notar através dos gráficos polares o comportamento de cada solução individualmente. Note, por exemplo, que na Figura 6.4(a) uma única linha na cor azul escuro passa no objetivo f_2 (número de rotas) no centro do gráfico polar. Isso significa que esta solução possui o pior valor para este objetivo.

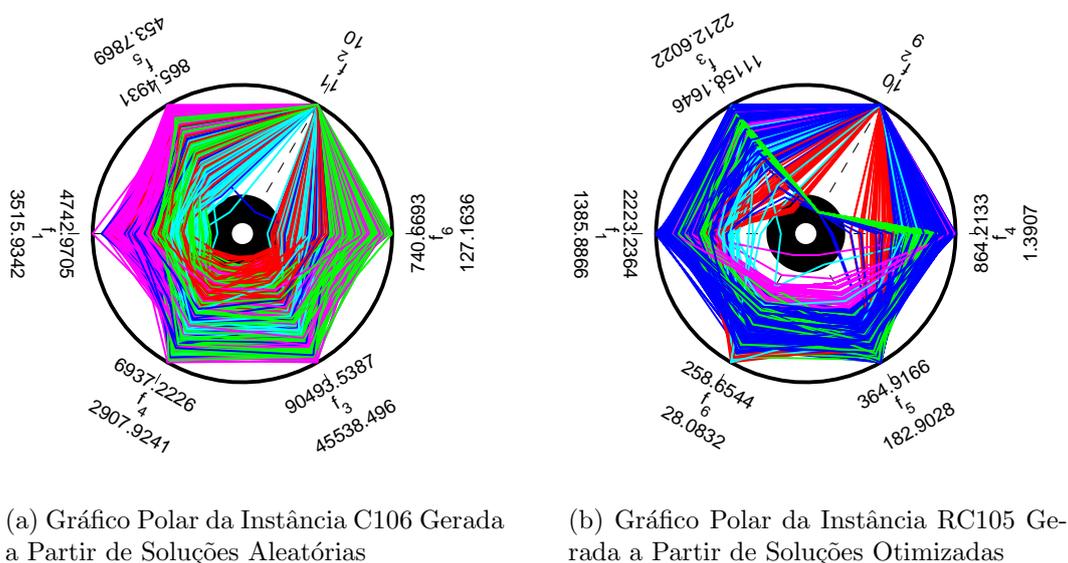


Figura 6.4: Gráficos Polares das Instâncias Semelhantes

Através das árvores médias foi possível ter uma noção sobre o comportamento dos objetivos tratados no MOPRV. Porém, para uma melhor compreensão árvores individuais de cada instância específica, bem como o comportamento de cada objetivo será analisado separadamente nas próximas seções.

Comportamento do Objetivo f_1 (distância percorrida)

Considerando 10 mil soluções geradas aleatoriamente (permutação de inteiros) o objetivo que considera a minimização da distância total percorrida apresentou diversos comportamentos distintos para as 56 instâncias propostas por Solomon (1987). Em

aproximadamente 32.1% (18 instâncias) dos casos, f_1 foi agregado ao objetivo composto $f_5+f_6+f_2$, ou seja, considerando esta subárvore a primeira agregação foi entre f_5+f_6 , o segundo passo agrega f_2 ao objetivo composto f_5+f_6 (objetivo formado por uma agregação) e por fim, f_1 é agregado à $f_5+f_6+f_2$.

Em aproximadamente 26.7% (15 instâncias) dos casos f_1 foi agregado ao objetivo composto f_5+f_6 . Para 21.4% (12 instâncias) das instâncias a agregação do objetivo f_1 foi feita na primeira iteração com o objetivo f_3 . Já para 14.5% (8 instâncias) dos casos o objetivo f_1 foi agregado ao objetivo composto f_3+f_4 . E por fim, f_1 foi agregado em aproximadamente 5.3% (3 instâncias) das instâncias com o objetivo composto $f_6+f_2+f_5$.

Agora considerando 2 mil soluções otimizadas pelo NSGA-III o objetivo f_1 apresentou, assim como para as soluções aleatórias, diversos comportamentos distintos para as 56 instâncias testadas. Em aproximadamente 30.4% (17 instâncias) dos casos f_1 foi agregado ao objetivo f_5 . Em pouco mais de 23.2% (13 instâncias) das instâncias f_1 foi agregado ao objetivo composto $f_5+f_6+f_2$. Para aproximadamente 17.8% (10 instâncias) das instâncias a agregação foi feita entre f_1 e o objetivo composto f_5+f_6 . Outros casos envolveram a agregação de f_1 com menos frequência.

Demonstrado o comportamento do objetivo que considera a minimização da distância total percorrida, fica claro que este objetivo não apresenta um comportamento padrão entre o conflito e a harmonia com os demais objetivos do problema. Assim, pode-se notar que em nenhum caso, tanto para soluções aleatórias quanto para soluções otimizadas, f_1 apresentou índice de agregação igual ou acima de 35% com qualquer outro objetivo. Outro fator importante que se pode observar é que para as soluções aleatórias, em pouco mais de 73.2% dos casos f_1 foi agregado a objetivos compostos já resultantes de duas agregações. Já para soluções otimizadas f_1 foi agregado a objetivos compostos resultantes de duas agregações em aproximadamente 51.7% das instâncias. Entretanto, é possível pensar que f_1 poderia ser agregado à f_3 ou à f_5 , dado que somando as soluções aleatórias e otimizadas estas agregações foram as que mais aconteceram (19 e 17 vezes no total de 112 execuções, respectivamente) quando considerados apenas agregações entre os objetivos originais (objetivos não agregados) do problema. Ou seria possível pensar em agregar f_1 aos maiores índices apresentados nos resultados, como por exemplo agregar f_1 à f_5+f_6 ou à $f_5+f_6+f_2$.

Porém, os resultados levam a considerar que f_1 possui pouca harmonia com os demais objetivos. Agregar f_1 à qualquer outro objetivo original do problema não faz sentido, isto porque em alguns poucos casos f_1 é agregado a estes objetivos, não podendo generalizar

este comportamento para as demais instâncias. Agregar f_1 a qualquer outro objetivo composto também não é aconselhável. Primeiro porque os objetivos mais harmoniosos são agregados nas primeiras iterações da Árvore de Agregação, fazendo com que os objetivos que são agregados com outros objetivos compostos apresentem pouca harmonia com qualquer outro objetivo. Segundo porque, mesmo que o fator anterior não interferisse na agregação, para objetivos compostos f_1 também apresenta muitas agregações com comportamentos diferentes, não podendo generalizar qualquer agregação que seja para todas as instâncias de Solomon (1987). Neste caso, o mais natural seria não agregar f_1 a qualquer outro objetivo do problema.

Comportamento do Objetivo f_2 (número de rotas)

Antes de analisar o objetivo f_2 , é importante relembrar que a Árvore de Agregação apresenta um comportamento diferenciado para objetivos que possuem o mesmo valor de harmonia para os demais objetivos do problema. Por exemplo, se um dado objetivo f_x possui um valor h de harmonia para todos os outros objetivos do problema e esta harmonia é a maior entre todas as outras, o objetivo f_x fica na reserva para que os objetivos que possuem a segunda maior harmonia sejam agregados nesta iteração. Na iteração seguinte f_x é agregado ao objetivo composto do passo anterior. Este processo garante que f_x não seja agregado arbitrariamente com outro objetivo, fazendo do resultado o mais próximo do original. Este procedimento é denominado de Critério de Desempate.

Analisando o comportamento do objetivo f_2 para as 10 mil soluções geradas aleatoriamente, foi observado que o objetivo que considera a minimização do número de rotas foi agregado, em pouco mais de 66% das instâncias, na segunda iteração da Árvore de Agregação. Nestes casos, Tal ocorrido é justificado pelo fato de que minimizar o número de rotas possui 0% de conflito com todos os outros objetivos. O valor mínimo de conflito implica em valores iguais a 100% de harmonia entre o objetivo que considera a minimização da quantidade de rotas e os demais objetivos do problema. Entretanto, valores máximos de harmonia para todos os objetivos só foram atingidos porque todas as soluções aleatórias que foram dadas como entrada para a Árvore de Agregação apresentaram valores únicos para o número rotas (número de veículos/rotas iguais para todas as soluções). Assim, a Árvore de Agregação fez uso do Critério de Desempate agregando os objetivos que apresentaram a segunda maior harmonia na primeira iteração do algoritmo.

O objetivo f_2 foi agregado em 37.5% (21 instâncias) dos casos ao objetivo composto

de classe 1 f_5+f_6 , aproximadamente 26.8% (15 instâncias) ao objetivo composto de classe 1 f_3+f_4 e em pouco mais de 1.8% (1 instância) ao objetivo composto de classe 1 f_1+f_3 , totalizando cerca de 66% das instâncias. Para as 19 instâncias restantes (aproximadamente 33.9%) f_2 foi agregado ao objetivo f_6 na primeira iteração. Para estes casos, minimizar o número de rotas (f_2) apresentou conflito muito pequeno (em média 0.4222%) quando agregado ao objetivo que visa a minimização da diferença entre a maior e a menor rota (f_6), isto porque as soluções geradas para estas instâncias apresentaram apenas 2 valores diferentes para o número de rotas.

Para as 2 mil soluções otimizadas pelo NSGA-III o comportamento de f_2 foi muito parecido com o comportamento apresentado para as soluções aleatórias. Como no caso anterior a agregação do objetivo que considera a minimização do número de rotas foi feita apenas com objetivos originais e objetivos compostos de classe 1. Para estas soluções f_2 foi agregado aos objetivos compostos f_3+f_4 , f_5+f_6 , f_5+f_1 e f_4+f_6 em aproximadamente 41.1% (23 instâncias), 28.5% (16 instâncias), 8.9% (5 instâncias) e 1.8% (1 instância) dos casos, respectivamente. Quando agregados com objetivos originais do problema, f_2 foi agregado ao objetivo f_4 , f_1 , f_6 e f_3 em aproximadamente 8.9% (5 instâncias), 5.3% (3 instâncias), 3.6% (2 instâncias) e 1.8% (1 instâncias) das instâncias, respectivamente.

Observando estes resultados, é possível notar que o número de veículos quase sempre (em aproximadamente 66% para instâncias aleatórias e 80% para instâncias otimizadas) foi agregado na segunda iteração com 0% de conflito com os demais objetivos. Para estes casos a quantidade de veículos não possui nenhuma interferência com qualquer outro objetivo, sendo viável agrega-lo a qualquer outro objetivo do problema. Porém, quando este objetivo obteve algum conflito com outros objetivos, o caso mais corrente para soluções aleatórias agregou f_2 à f_6 (19 instâncias) e, para soluções otimizadas agregou f_2 à f_4 (5 instâncias). Nestes casos, a diferença entre as soluções otimizadas e aleatórias é que, para as soluções aleatórias a única agregação que envolveu o objetivo f_2 e outro objetivo original do problema foi entre f_2 e f_6 . Já para as soluções otimizadas diversas agregações entre objetivos originais e a minimização do número de rotas foram propostas, sendo que a maior entre elas, f_2 e f_4 , ocorreu para 5 instâncias enquanto f_2 e f_6 foram agregados em 2 instâncias para estas soluções. Diante disto, pode-se notar que para soluções otimizadas a agregação entre f_2 e f_4 não obteve uma grande diferença quando comparada a agregação entre f_2 e f_6 , fazendo das soluções aleatórias um diferencial importante para esta comparação.

Diante dos resultados, o objetivo que considera a minimização do número de rotas poderia ser agregado a qualquer outro objetivo do problema. Isto porque o conflito entre

este objetivo e qualquer outro objetivo, seja ele composto ou original, é muito pequeno. No entanto, seria mais viável agregar f_2 à f_6 ou a algum objetivo composto que envolve o objetivo f_6 , como por exemplo f_5+f_6 ou f_4+f_6 . Se número de rotas quase sempre possui conflito zero com os demais objetivos e quando possui conflito é agregado ao objetivo que visa a minimização da diferença entre a maior e menor rota (balanceamento de rota), é melhor agrega-lo a qualquer seguimento da árvore de agregação que envolva o objetivo de balanceamento de rota.

Comportamento do Objetivo f_3 (violação da restrição de tempo)

Considerando 10 mil soluções geradas aleatoriamente (permutação de inteiros) o objetivo que considera o grau de violação da restrição de tempo apresentou apenas 3 comportamentos de agregação para as 56 instâncias propostas por Solomon (1987). Entre estes comportamentos o que mais se destacou foi a agregação de f_3 com o objetivo f_4 . Esta agregação aconteceu em aproximadamente 76.8% (43 instâncias) das instâncias, um índice considerável de agregação entre objetivos. Para outros casos f_3 foi agregado ao objetivo f_1 e ao objetivo composto de classe 2 $f_5+f_6+f_2$ em aproximadamente 21.4% (12 instâncias) e 1.8% (1 instâncias) das instâncias, respectivamente.

Já para as soluções otimizadas o objetivo f_3 apresentou diversos comportamentos de agregação. Todavia, um alto índice de agregação entre f_3 e f_4 também foi apresentado para as soluções otimizadas. Tal agregação ocorreu em cerca de 73.1% (41 instâncias) dos casos. Da mesma forma que para as soluções aleatórias, a segunda maior agregação que envolveu o objetivo f_3 foi com o objetivo f_1 . Esta agregação ocorreu em cerca de 12.5% (7 instâncias) das instâncias. Outros casos envolveram a agregação de f_3 com menos frequência.

Pode-se notar que minimizar o grau de violação da restrição de tempo apresentou um alto índice de agregação com o objetivo que considera a minimização da espera dos veículos nos consumidores. Assim, diferente do objetivo f_1 , o objetivo f_3 demonstrou um comportamento padrão tanto para soluções aleatórias como para soluções otimizadas. Em outras oportunidades f_3 foi agregado à objetivos originais e objetivos compostos. Todavia, o alto índice de agregação apresentado pela ocorrência do objetivo f_3+f_4 não nos faz ter dúvidas de que esta agregação é extremamente viável.

Comportamento do Objetivo f_4 (espera dos veículos)

A análise feita sobre o objetivo que considera a minimização da espera dos veículos nos consumidores (f_4) acaba por recair sobre o estudo feito para o objetivo f_3 . Isto porque em diversas instâncias a agregação entre estes dois objetivos foi feita. Desde modo, é sabido que para soluções aleatórias e soluções otimizadas, o objetivo que considera a minimização da espera dos veículos nos consumidores foi agregado ao objetivo que visa a minimização do grau de violação da restrição de tempo em 76.8% (43 instâncias) e 73.1% (41 instâncias) das instâncias, respectivamente. Outros casos envolveram a agregação de f_4 com menos frequência.

A conclusão para o objetivo f_4 acaba por ser igual à do objetivo f_3 . Isto porque a relação entre estes dois objetivos é constante para as instâncias testadas. Assim, de maneira semelhante ao objetivo f_3 , minimizar a espera dos veículos nos consumidores demonstrou um comportamento padrão tanto para soluções aleatórias como para soluções otimizadas. Em diversas outras oportunidades f_4 foi agregado a outros objetivos que não fosse f_3 . Porém, o alto índice de agregação apresentado pela ocorrência do objetivo f_3+f_4 sugere mais uma vez a agregação entre estes dois objetivos.

Comportamento do Objetivo f_5 (makespan)

Considerando 10 mil soluções geradas aleatoriamente (permutação de inteiros) o objetivo que avalia a maior rota apresentou apenas 3 comportamentos de agregação para as 56 instâncias propostas por Solomon (1987). Entre estes comportamentos o que mais se destacou foi a agregação de f_5 com o objetivo f_6 . Esta agregação aconteceu em aproximadamente 66.1% (37 instâncias) das instâncias, um índice considerável de agregação entre objetivos. Para outros casos f_5 foi agregado aos objetivos compostos f_6+f_2 e f_3+f_1 em aproximadamente 19.6% (11 instâncias) e 14.3% (8 instâncias) das instâncias, respectivamente.

Para as soluções otimizadas o objetivo f_5 também apresentou poucas variações de comportamento. A que mais se destacou foi a agregação de f_5 com o objetivo f_6 em aproximadamente 64.3% (36 instâncias) das instâncias. Outro caso que possuiu um índice de agregação razoável foi entre f_5 e f_1 com 30.3% (17 instâncias) das instâncias. Para os dois casos finais f_5 foi agregado aos objetivos compostos f_6+f_2 e $f_6+f_4+f_2$ em cerca de 3.6% (2 instâncias) e 1.8% (1 instância), respectivamente.

Dentre todos os comportamentos que o objetivo f_5 apresentou, o que mais se destacou foi a agregação deste objetivo com o objetivo f_6 . Esta agregação foi executada em 66.1% para soluções aleatórias e 64.3% para soluções otimizadas. Apesar do índice de agregação não estar dentre os maiores, esta agregação apresentou um índice razoável para as 56 instâncias propostas por Solomon (1987) tanto para soluções aleatórias como para soluções otimizadas.

Todavia, um dado importante deve ser observado. Para soluções aleatórias em 19.6% das instâncias, minimizar a maior rota foi agregado ao objetivo f_6+f_2 e para soluções otimizadas esta agregação ocorreu para cerca de 3.6% das instâncias. Como descrito anteriormente a agregação entre o objetivo de balanceamento de rota (f_6) e o objetivo que verifica o número de rotas (f_2) obteve em média 0.4222% de conflito, o que significa que o conflito entre estes objetivos é muito pequeno. Caso o objetivo f_2 tivesse 0% de conflito para todos os objetivos nesses 19.6% e 3.6% das instâncias, a primeira iteração agregaria f_5 e f_6 deixando f_2 na reserva para ser agregado na iteração seguinte. Diante deste ponto de vista, pode-se dizer que para os casos em que f_5 foi agregado ao objetivo f_6+f_2 , f_5 está tão próximo de ser agregado à f_6 na primeira iteração como 0.4222% de conflito está próximo de 0% de conflito. Um caso que ainda apresentou índices de agregação significativos foi entre f_5 e f_1 com aproximadamente 30.3% para soluções otimizadas e 0% para soluções aleatórias. Contudo, caso f_5 tivesse que ser agregado a algum objetivo, o mais natural seria que f_5 fosse agregado ao objetivo f_6 devido ao comportamento apresentado pelo mesmo.

Comportamento do Objetivo f_6 (balanceamento de rota)

A análise feita sobre o objetivo que considera a minimização da diferença entre a maior e a menor rota acaba por recair sobre o estudo feito para o objetivo f_5 . Isto porque em diversas instâncias a agregação entre estes dois objetivos foi feita. Desde modo, é sabido que para soluções aleatórias e soluções otimizadas o objetivo f_6 foi agregado ao objetivo f_5 em 66.1% (37 instâncias) e 64.3% (36 instâncias) das instâncias, respectivamente. Outros casos envolveram a agregação de f_6 com menos frequência.

Assim como para o objetivo f_5 , f_6 se destacou por ser agregado a este objetivo na maior parte das instâncias. Ainda assim, f_6 acaba por cair na mesma análise feita para o objetivo f_5 quando f_2 é considerado na agregação entre f_5 e f_6 . Para soluções aleatórias e otimizadas em 33.9% e 3.6% das instâncias, respectivamente, f_6 é agregado ao objetivo f_2 com conflito médio de 0.4222%. Deste modo, f_2 possui conflito tão baixo

com os demais objetivos que a agregação deste com qualquer outro objetivo, interferiria muito pouco nos valores de função das soluções geradas. Diante disto e desconsiderando o objetivo f_2 , a agregação entre os objetivos f_5 e f_6 apresentam índices de agregação muito superiores as outras agregações para o objetivo f_6 . Contudo, caso f_6 tivesse que ser agregado a algum objetivo, o mais natural seria que f_6 fosse agregado ao objetivo f_5 devido ao comportamento apresentado pelo mesmo.

6.5.2 Otimizando o MOPRV Completo

De forma geral, o resultado para todas as instâncias testadas, que utilizou o NSGA-III para o problema completo, são semelhantes. Considerando as soluções de cada grupo de instâncias, as soluções de Pareto para os objetivos do problema apresentam um comportamento ainda mais semelhante. Assim, as Figuras 6.5, 6.6 e 6.7 apresentam as soluções da primeira frente de Pareto de uma execução do NSGA-III em Gráficos Polares para uma instância de cada grupo (R1, R2, C1, C2, RC1 e RC2). Cada solução não dominada para uma execução do NSGA-III é representada por uma linha nos Gráficos Polares. Cada linha dos gráficos polares representa uma solução que foi dada como entrada na Árvore de Agregação. No entanto, quanto mais externa é a linha em um determinado objetivo, melhor é o valor desta solução para este objetivo. Por outro lado, quanto mais a linha se aproxima do centro dos gráficos em um objetivo, pior é o valor desta solução para este objetivo.

As soluções retornadas pelo NSGA-III para as instâncias do grupo R apresentaram o mesmo número de rotas para o objetivo f_2 (minimização do número de rotas). Assim, este objetivo não interfere no comportamento de qualquer outro objetivo do problema. Diante disso, a análise feita para o grupo de instâncias R desconsidera a relação entre o objetivo que avalia o número de rotas e os demais objetivos tratados no MOPRV. Analisando as soluções de Pareto das instâncias R101 e R201, pode-se observar que valores bons para o objetivo que considera a minimização da maior rota (f_5) acabam por retornar valores bons para o objetivo que avalia a diferença entre a maior e a menor rota (f_6). Soluções na cor verde da instância R101 apresentam este comportamento, por exemplo. De forma semelhante, quando bons valores são encontrados para o objetivo que avalia o grau de violação da restrição de tempo (f_3), bons valores também são retornados para o objetivo que considera a minimização da espera dos veículos nos consumidores (f_4). Soluções em verde da instância R201, por exemplo, apresentam valores muito bons para f_3 e f_4 , desde que valores ruins para f_5 e f_6 sejam aceitos. É possível observar que

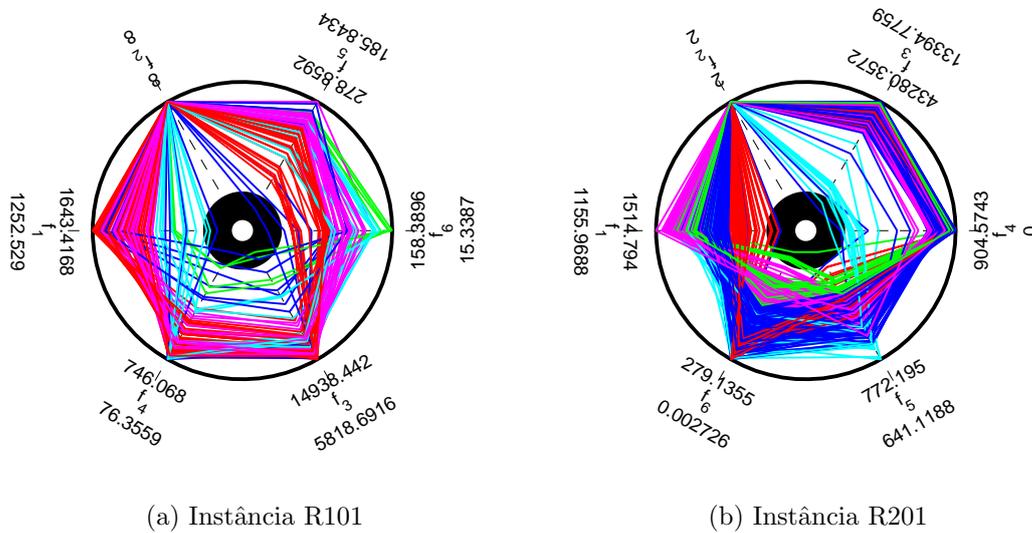


Figura 6.5: Gráficos Polares das instâncias do grupo R para o MOPRV Completo

soluções em azul claro da instância R101 demonstram um comportamento em que valores ruins e médios são apresentados para o objetivo f_1 e valores bons e médios apresentados para os demais objetivos do problema. Este fato demonstra o alto conflito apresentado pelas árvores de agregação entre o objetivo que considera a minimização da distância total percorrida (f_1) e os demais objetivos do MOPRV. No entanto, outras soluções apresentaram valores intermediários para todos os objetivos. Este comportamento é apresentado pelas soluções em rosa da instância R101 e pelas soluções na cor azul escuro da instância R201, por exemplo.

Assim como as instâncias do grupo R, as soluções de Pareto retornadas para as instâncias do grupo C apresentaram um único valor para o objetivo f_2 . Ainda assim, os demais objetivos apresentam relações que podem ser observadas através das soluções de Pareto. Por exemplo, para a instância C101 e considerando o grupo verde, as soluções apresentam valores ótimos para os objetivos f_3 e f_4 , valores razoáveis para f_5 e valores ruins para f_6 . O que acaba por demonstrar, mais uma vez, uma grande harmonia entre o objetivo que considera a minimização do grau de violação da restrição de tempo e o objetivo que calcula a espera dos veículos nos consumidores. Outra vez é possível observar, através das soluções na cor rosa da instância C101, que minimizar a maior rota da solução acaba por minimizar a diferença entre a maior e a menor rota, ou seja, valores bons para f_5 implicam em valores bons para f_6 . Outras soluções de Pareto

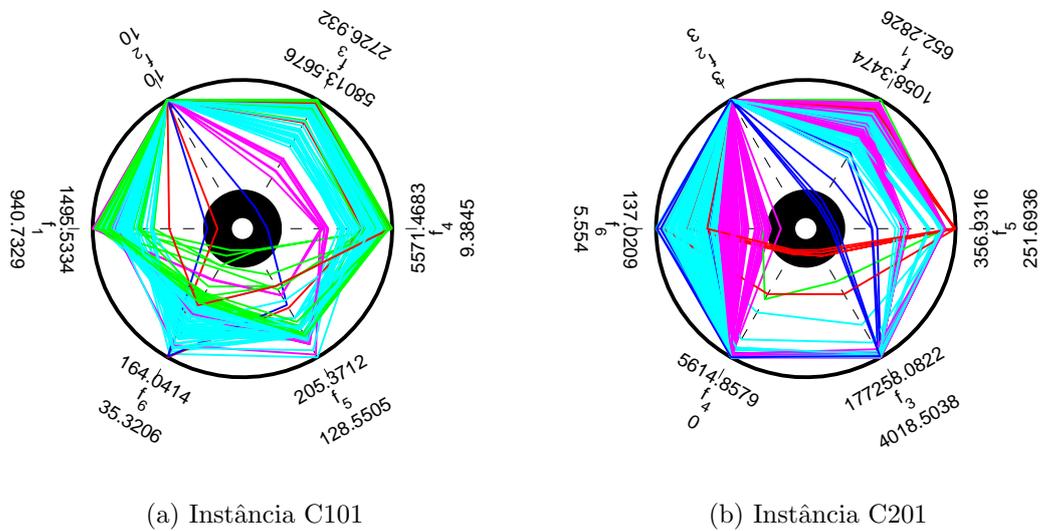


Figura 6.6: Gráficos Polares das instâncias do grupo C para o MOPRV Completo

apresentam intervalos entre valores médios e bons para todos os objetivos. Soluções em azul claro da instância C101 e da instância C201 apresentam este comportamento.

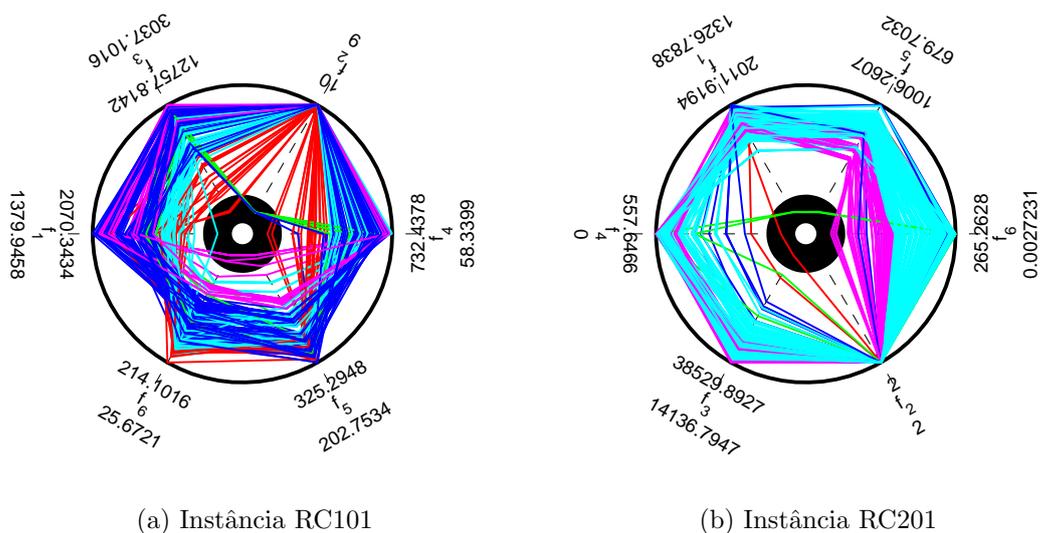


Figura 6.7: Gráficos Polares das instâncias do grupo RC para o MOPRV Completo

Se tratando do objetivo f_2 , a instância RC201 assim como as demais instâncias apre-

sentadas nesta seção, retornou um único valor para este objetivo. De forma distinta, a instância RC101 apresentou soluções com valores diferentes para o objetivo f_2 , apresentando soluções com 9 e 10 rotas. No entanto, mesmo quando o número de veículos foi diferente, o comportamento entre os demais objetivos continuou semelhante. Por exemplo, o grupo em vermelho da instância RC101 apresenta valores bons para f_5 e f_6 que implicam em valores ruins e médios para os demais objetivos do problema. Isto é, a relação de harmonia entre minimizar a maior rota e minimizar a diferença entre a maior e a menor rota continua sendo verificada. Da mesma forma, minimizar o grau da violação da restrição de tempo implica em minimizar também à espera dos veículos nos consumidores. O grupo verde da instância RC101, por exemplo, apresenta soluções com valores bons para f_3 e f_4 e valores médios e ruins para os demais objetivos.

Contudo, todas as instâncias apresentaram comportamentos semelhantes quanto às soluções geradas pelo NSGA-III. Primeiramente, é possível observar que das 6 instâncias apresentadas, para 5 instâncias o NSGA-III gerou soluções de Pareto com valores exatamente iguais para o número de rotas. Outro comportamento observado demonstra que em grande parte das instâncias quando se tinha valores bons para f_3 , bons valores também eram encontrados para f_4 . Valores ruins em f_3 também implicaram em valores ruins para f_4 . Este comportamento foi também observado para os objetivos f_5 e f_6 .

Já o objetivo f_2 quase sempre não interferiu no comportamento dos outros objetivos, isto porque, os valores para este objetivo tiveram pouca variância. Experimentos realizados por Garcia-Najera e Bullinaria (2011) demonstram que para 27 instâncias de Solomon (1987) o resultado apresentou Paretos que continham uma solução com relação ao número de rotas, para 24 instâncias as aproximações de conjuntos Pareto continham duas soluções que variavam o número de rotas, e para as outras 5 instâncias foram retornados Paretos com 3, 4 e 5 soluções com valores de rotas diferentes, somando um total de 56 instâncias. Deste modo, a natureza das instâncias propostas por Solomon (1987) não possibilitam a formação de soluções variadas para o objetivo f_2 .

Outro aspecto que pode influenciar na formação de soluções variadas para este objetivo, é que o objetivo que minimiza o número de rotas é discreto, e os algoritmos propostos inserem o número de rotas de maneira gulosa na solução, como descrito na Seção 5.1. Assim, o algoritmo não gera soluções com a mesma ordem de atendimento aos clientes com número de rotas diferentes, ou seja, se a ordem de atendimento é a mesma, o número de rotas é igual para todas estas soluções. Fazendo com que o espaço de busca para este objetivo seja reduzido.

6.5.3 Redução do MOPRV

As Árvores de Agregação apresentam o comportamento sobre o conflito e harmonia entre os objetivos de um problema de otimização. Assim, é papel do tomador de decisão avaliar e escolher quais objetivos serão agregados ou não, de acordo com seu interesse e experiência. Neste caso, faremos o papel do tomador de decisão para agregar os objetivos do Problema de Roteamento de Veículos com Muitos Objetivos e Janelas de Tempo Flexíveis.

Baseado nos resultados, os objetivos que normalmente são agregados no conjunto de instâncias testadas são $f_3 + f_4$ e $f_5 + f_6$. Assim, podemos agregar estes objetivos com uma pequena perda na qualidade das soluções da estimativa de Pareto-Ótimo do problema original. Nós então, consideramos os objetivos agregados $f_a = f_3 + f_4$ e $f_b = f_5 + f_6$. Quanto ao objetivo f_2 , apresenta baixo conflito quando comparado com todos os outros objetivos. No entanto, quando se tem valores mais diversificados no conjunto de soluções para o número de veículos (f_2), é mais comum agregar f_2 à f_6 por possuir um menor conflito. Como f_6 já está em $f_b = f_5 + f_6$, agregamos f_2 à f_b , por isso temos o objetivo composto $f_c = f_b + f_2$. Outro indicador que sugere esta agregação é que f_2 é agregado à ($f_5 + f_6$) em 37.5% (21 instâncias) para soluções aleatórias e 28.5% (16 instâncias) para soluções otimizadas.

O objetivo f_1 apresenta um comportamento indefinido a partir de um ponto de vista geral. Este objetivo é geralmente agregado em níveis mais altos da árvore com diferentes objetivos como f_3 , f_5 , f_a e f_b . Em diversos casos, f_1 apresenta um baixo nível de harmonia com todos os outros objetivos do MOPRV, não sendo interessante agregá-lo com qualquer outro objetivo. Assim, propomos uma formulação reduzida do MOPRV considerando estas agregações e reduzindo o número de objetivos no problema de seis para três.

A formulação reduzida do MOPRV é definida pelas equações 6.1 (objetivo relacionado à distância), 6.2 (violação das janelas de tempo e tempo de espera dos veículos) e 6.3 (tamanho da rota mais longa, diferença entre a maior e a menor rota e número de rotas),

$$\text{Minimize } f_1 \tag{6.1}$$

$$\text{Minimize } f_a = \alpha f_3 + \lambda f_4 \quad (6.2)$$

$$\text{Minimize } f_c = \beta f_5 + \gamma f_6 + \epsilon f_2 \quad (6.3)$$

β , γ , α , λ e ϵ são os pesos para os objetivos f_5 , f_6 , f_3 , f_4 e f_2 , respectivamente.

6.5.4 Otimizando o MOPRV Reduzido

O NSGA-II proposto foi utilizado para otimizar as 56 instâncias de Solomon (1987) no MOPRV Reduzido. Para a otimização do problema reduzido os seis objetivos originais do problema foram linearmente normalizados entre 0 e 1, sendo que o mesmo peso foi atribuído para os objetivos no somatório das funções f_a e f_c . Isto é, β , γ , α , λ e ϵ receberam $\text{peso} = 1$. Assim, dado que o objetivo f_1 não foi agregado à nenhum outro objetivo, este apresenta valores entre 0 e 1. O objetivo f_a apresenta a soma dos valores normalizados dos objetivos f_3 e f_4 , ou seja, f_a retorna valores entre 0 e 2. Por fim, o valor do objetivo f_c é igual ao somatório dos valores normalizados dos objetivos f_2 , f_5 e f_6 , resultando em soluções com valores que variam de 0 até 3.

Os resultados são apresentados em forma de gráficos em coordenadas polares. Para ilustrar os resultados, apresentamos o Gráficos Polares para 6 casos representativos, cada um de um subgrupo de instâncias (R1, R2, C1, C2, RC1 e RC2). As Figuras 6.8, 6.9, e 6.10 apresentam os resultados para as instâncias R101, R201, C101, R201, RC101 e RC201. Os resultados, no entanto, foram muito semelhantes para todas as instâncias testadas. Assim, os gráficos polares apresentaram três grupos de soluções para cada instância testada. Estes grupos são representados pelas linhas verdes, vermelhas, e azuis. Em geral, é possível notar nos gráficos que soluções com bons valores são encontradas para dois objetivos, uma vez que uma piora seja considerada no terceiro objetivo.

Por exemplo, podemos notar que para instância R101, o grupo representado em verde tem bons valores para f_a e f_c e valores ruins para f_1 . Outro exemplo pode ser visto para soluções em azul na instância R201, em que o NSGA-II otimiza bem os objetivos f_1 e f_c uma vez que piores valores para o objetivo f_a são encontrados. No entanto,

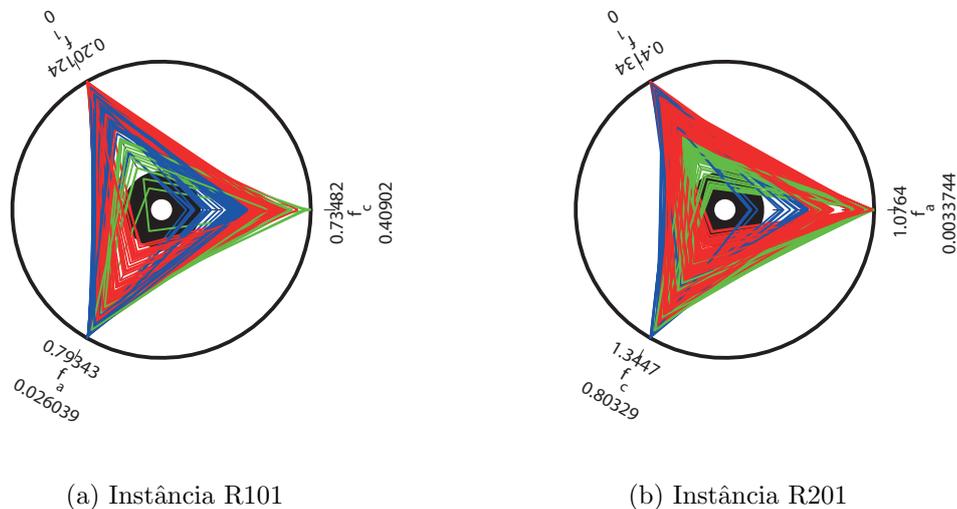


Figura 6.8: Gráficos Polares das instâncias do grupo R para o MOPRV Reduzido

existem soluções com bons valores para f_1 e f_a que apresentam valores ruins para a f_c . Soluções em vermelho apresentam diversos comportamentos, retornando valores ruins, médios e bons para todos os objetivos. Contudo, se uma análise individual for feita em cada solução do grupo de soluções em vermelho, é possível notar que, ainda que boas soluções sejam apresentadas para dois objetivos, existe uma perda no valor do terceiro objetivo. Para as instâncias do grupo R, o grupo de soluções em vermelho se destaca por obter a melhor solução para o objetivo f_1 . É possível notar ainda, que para este grupo de instâncias, a melhor solução para cada objetivo do problema pertence a grupos de soluções distintos.

Veja por exemplo, que para as instâncias C101 e C201 as soluções em azul apresentam bons resultados para f_1 e f_c com valores ruins para o objetivo f_a . Por outro lado soluções em verde apresentam valores bons para os objetivos f_a e f_c , desde que se aceite valores ruins para f_1 . Valores em vermelho possuem o mesmo comportamento descrito para instâncias do grupo R. Soluções deste grupo apresentaram valores ruins, médios e bons para todos os objetivos e retornando ainda os melhores valores para o objetivo f_1 .

Para as instâncias do grupo RC é possível notar o mesmo comportamento indicado para os grupos anteriores. Veja por exemplo, que soluções em azul têm bons valores para f_1 e f_c com pioras para o objetivo f_a . Já o grupo verde apresenta bons valores para f_1 e f_a , desde que valores ruins para f_c sejam considerados. Soluções do grupo vermelho não

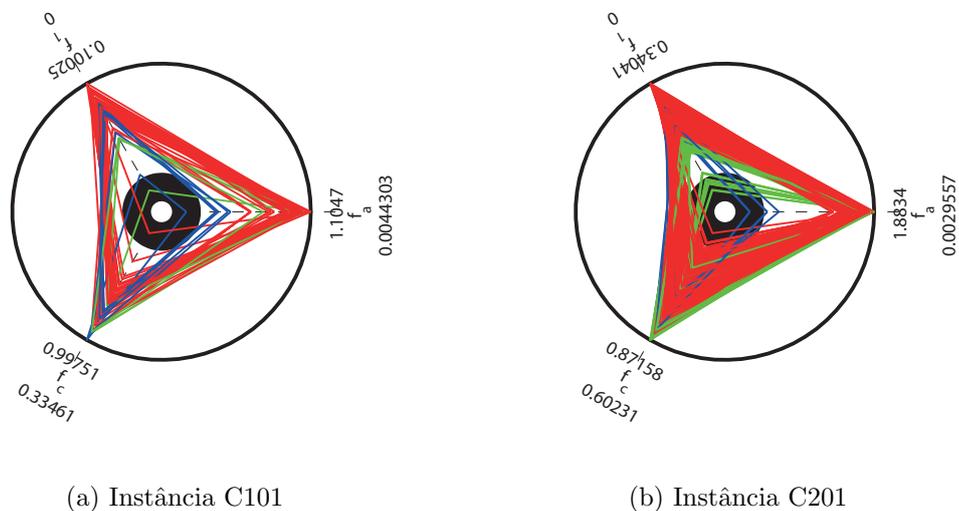


Figura 6.9: Gráficos Polares das instâncias do grupo C para o MOPRV Reduzido

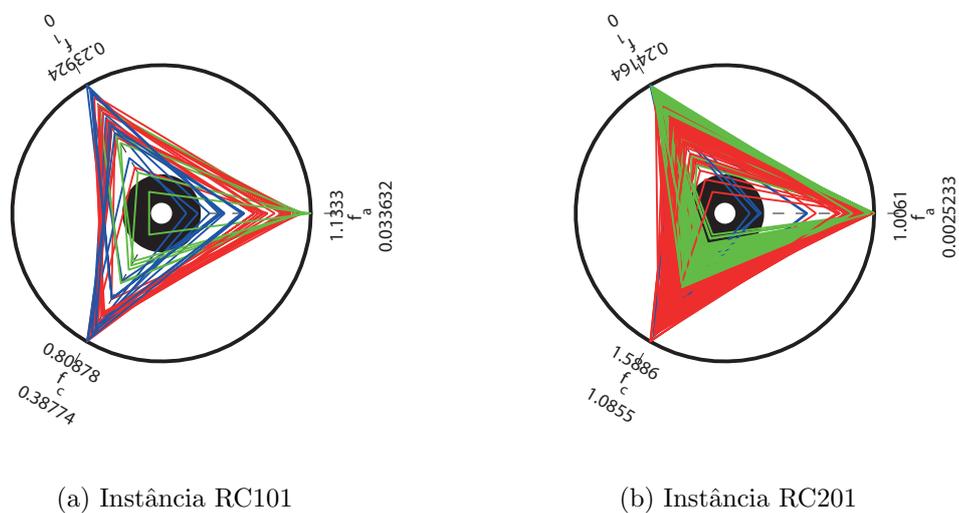


Figura 6.10: Gráficos Polares das instâncias do grupo RC para o MOPRV Reduzido

apresentaram o melhor valor para o objetivo f_1 , como ocorrido nos grupos de instâncias anteriores, porém valores muitos bons foram apresentados por soluções deste grupo.

Brevemente, foi verificado através dos Gráficos Polares que soluções com valores

muitos bons podem ser encontrados para dois objetivos, desde que valores ruins sejam considerados para o terceiro objetivo. A medida que valores médios são encontrados para dois objetivos, o valor do terceiro objetivo melhora. Assim, outras opções de soluções são apresentadas. Estas não possuem os melhores valores para algum objetivo do problema, porém, apresentam valores que se aproximam dos melhores para os três objetivos estudados. Assim, fica a disposição do decisor escolher soluções com valores muito bons para dois objetivos e valores ruins para o outro objetivo, ou escolher soluções que apresentam valores intermediários para todos os objetivos.

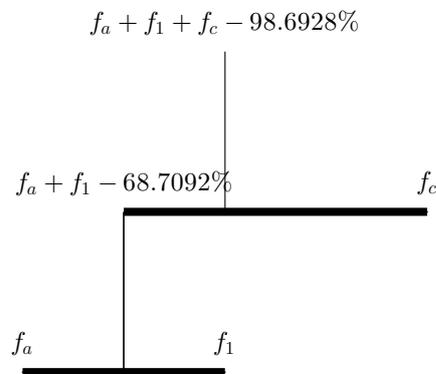


Figura 6.11: Árvore de Agregação para o MOPRV Reduzido

Entretanto, se quiséssemos reduzir o problema ainda mais, a agregação do objetivo f_1 ao objetivo f_a é o mais indicado. Isto porque a Árvore de Agregação mais gerada para o problema reduzido, considerando as instâncias testadas, é apresentada na Figura 6.11. Assim, é possível observar que o conflito encontrado na agregação destes objetivos foi extremamente alto. Este comportamento se aplica em grande parte dos problemas de teste.

6.5.5 MOPRV Completo x MOPRV Reduzido

Inicialmente o MOPRV composto por seis objetivos (problema completo) foi otimizado com o NSGA-III. Os resultados foram analisados e a aplicação do algoritmo gerou soluções otimizadas que serviram como entrada para a ferramenta Árvore de Agregação. Baseado nos resultados indicados pela árvore, uma versão reduzida do MOPRV foi pro-

posta. Esta versão reduziu o número de objetivos do problema de seis para três, de modo que, os objetivos mais harmoniosos foram agregados em um único objetivo. Considerando a versão reduzida, o NSGA-II proposto foi então utilizado para otimizar o problema. Assim, esta seção verifica através de uma métrica denominada de Cobertura (C) se as soluções de Pareto encontradas para o problema reduzido são tão boas quanto as soluções encontradas para o problema completo. Isto é, verificamos se a redução do problema completo, bem como se as soluções retornadas para o problema reduzido demonstraram alguma perda de qualidade quando comparado ao problema completo.

O propósito da métrica de cobertura é que um algoritmo com melhor desempenho retorne soluções com maior cobertura que o outro método utilizado na comparação. Assim sendo, a métrica de cobertura verifica qual o melhor desempenho entre dois conjuntos de soluções (conjunto Pareto), aquele conjunto que possui melhor desempenho é aquele que possui o menor número proporcional de soluções dominadas. Assim, dadas duas fronteiras Pareto aproximadas A e B, $C(A,B)$ calcula o número de soluções em B que são dominadas por alguma outra solução de A. Se $C(A,B)$ é igual a 1, então todas as soluções em B são dominadas por soluções da fronteira A. Por outro lado, se o valor retornado é igual a 0 isso significa que nenhuma solução de B é dominada por soluções em A. A Equação 6.4 apresenta como se obtém o valor da cobertura dadas duas fronteiras Pareto aproximadas A e B.

$$C(A, B) = \frac{|y \in B : \exists x \in A, x \preceq y|}{|B|} \quad (6.4)$$

A equação de cobertura, no entanto, apresenta a porcentagem de soluções de uma fronteira Pareto que são dominadas por outra fronteira. Por exemplo, se uma fronteira B é composta por 100 soluções, e 35 destas soluções são dominadas por alguma solução da fronteira A, obtém-se 0,35 para $C(A,B)$, ou seja, 35% das soluções da fronteira B são dominadas. Para aplicação desta métrica é necessário avaliar tanto $C(A,B)$ quanto $C(B,A)$, pois estes valores não são complementares. Assim, os resultados são apresentados em porcentagens.

Para verificar a cobertura, escolhemos aleatoriamente uma fronteira Pareto do problema completo e uma fronteira Pareto do problema reduzido para cada instância testada. Ambos problemas estão em dimensões diferentes, o problema completo apresenta soluções para 6 objetivos (6 dimensões), já o problema reduzido apresenta soluções para

3 objetivos (3 dimensões). Para que a comparação da métrica de cobertura seja feita, é necessário que cada conjunto de soluções comparado esteja na mesma dimensão. Assim, as soluções retornadas pelo NSGA-II para o problema reduzido foram mapeadas novamente para as 6 dimensões do problema completo. Deste modo, os valores originais dos objetivos f_1 , f_2 , f_3 , f_4 , f_5 e f_6 foram calculados para cada solução das fronteiras escolhidas, de forma que, através da métrica de cobertura pode-se verificar qual fronteira apresentou melhores soluções para estes objetivos. Pode-se ainda observar se as primeiras frentes de Pareto para a formulação reduzida se aproximaram das estimativas de Pareto para a formulação completa do problema.

As Tabelas 6.2, 6.3 e 6.4 apresentam a porcentagem de soluções dominadas de cada fronteira Pareto para as formulações do problema. Assim, $C(R,C)$ apresentado na coluna 2, demonstra a quantidade proporcional de soluções da fronteira Pareto gerada pelo NSGA-III para o MOPRV Completo que são dominadas por alguma solução da fronteira Pareto gerada pelo NSGA-II para o problema reduzido. Já $C(C,R)$ apresentado na coluna 3, descreve a quantidade proporcional de soluções da fronteira Pareto gerada pelo NSGA-II para o MOPRV Reduzido que são dominadas por alguma solução da fronteira Pareto gerada pelo NSGA-III para o problema completo. Na coluna 1 estão contidas as instâncias de teste.

Apresentada a cobertura entre as aproximações de Pareto de cada formulação do problema para cada instância testada, é possível verificar a qualidade das soluções destas soluções. Observe que dentre as 56 instâncias de teste, em 35 oportunidades as soluções da frente de Pareto do problema reduzido (soluções retornadas pelo NSGA-II) dominaram mais soluções do problema completo (NSGA-III) do que foram dominadas. Para 20 instâncias o inverso ocorreu, ou seja, nestas oportunidades soluções de Pareto para o problema completo (soluções retornadas pelo NSGA-III) dominaram mais soluções do problema reduzido do que foram dominadas. Em um único caso, a cobertura foi igual para ambas as aproximações de Pareto (instância RC208), nesta oportunidade nenhuma solução do problema completo, bem como do problema reduzido foi dominada. Entretanto, na maior parte das instâncias, soluções com melhor qualidade foram apresentadas para o problema reduzido. Ainda assim, para 13 instâncias de teste nenhuma solução gerada para o problema reduzido foi dominada ($C(C,R)=0\%$).

Outro dado importante pode ser verificado através da média de soluções dominadas para as aproximações de Pareto geradas pelo NSGA-II e NSGA-III. Em média, con-

Tabela 6.2: Cobertura das instâncias do grupo R

Instância	C(R,C)	C(C,R)
R101	58,06%	0%
R102	71,95%	0%
R103	77,63%	0%
R104	95,03%	0%
R105	16,36%	22,93%
R106	20,67%	37,5%
R107	46,75%	5,5%
R108	28,30%	5,6%
R109	56,17%	0%
R110	24,41%	10,38%
R111	7,44%	33,96%
R112	13%	52,94%
R201	32,37%	16,78%
R202	18,45%	34,84%
R203	30,07%	10,75%
R204	9,8%	14,58%
R205	61,16%	0%
R206	3,81%	64,84%
R207	2,29%	10,09%
R208	18,27%	4,29%
R209	44,15%	18,38%
R210	5,82%	57,31%
R211	4,04%	43,42%
Média	32,434783	19,308261
Desvio Padrão	25,999279	19,932753

siderando as 56 instâncias de teste, 33,32% das soluções de Pareto apresentadas pelo NSGA-III foram dominadas por alguma solução retornada pelo NSGA-II. Esta mesma medida foi verificada para o problema reduzido, e foi verificado que em média 15,49% das soluções retornadas pelo NSGA-II são dominadas por alguma outra solução retornada para o problema completo. Isto é, em média 84,51% das soluções apresentadas para o problema reduzido não são dominadas por nenhuma solução gerada para o problema

Tabela 6.3: Cobertura das instâncias do grupo C

Instância	C(R,C)	C(C,R)
C101	47,47%	1,35%
C102	26,9%	23,72%
C103	16,66%	27,63%
C104	26,41%	18,42%
C105	90,80%	0%
C106	51,82%	32,25%
C107	16,8%	23,43%
C108	39,02%	17%
C109	67,05%	0%
C201	2,73%	10,14%
C202	31,21%	10,16%
C203	20,38%	0%
C204	18,84%	31,12%
C205	24,03%	44,23%
C206	94,09%	1,21%
C207	61,43%	2,27%
C208	53,16%	15,25%
Média	40,517647	15,187059
Desvio Padrão	25,501743	13,271329

completo. Isso indica que a perda na qualidade das soluções da frente de Pareto é muito pequena, o que torna viável a redução do MOPRV Completo. A viabilidade da redução do problema ainda é justificada pelo fato de que em muitos casos de teste, as soluções do problema reduzido não só são não dominadas, como dominam muitas soluções do problema completo. Isso significa que o problema reduzido ainda apresenta, em muitos casos, soluções melhores para os seis objetivos do MOPRV quando comparado com o problema completo.

Tabela 6.4: Cobertura das instâncias do grupo RC

Instância	C(R,C)	C(C,R)
RC101	55,62%	2,7%
RC102	5,94%	10%
RC103	15,88%	3,33%
RC104	2,85%	38,02%
RC105	0%	31,08%
RC106	40%	0%
RC107	0%	33,33%
RC108	40%	17,39%
RC201	39,09%	8,11%
RC202	36,18%	4,05%
RC203	44,26	0,79%
RC204	61,71%	0%
RC205	52,55%	0%
RC206	30,85%	15,06%
RC207	7,5%	28,2%
RC208	0%	0%
Média	27,026875	12,003750
Desvio Padrão	21,324136	13,092385

6.6 Conclusão

Este capítulo apresentou os resultados dos principais algoritmos propostos neste trabalho para as formulações do MOPRV Completo e MOPRV Reduzido. Os experimentos realizados consistem na execução dos algoritmos: *Nondominated Sorting Genetic Algorithm III* (NSGA-III), Árvores de Agregação e *Nondominated Sorting Genetic Algorithm II* (NSGA-II).

três conjuntos de instâncias propostas por Solomon (1987) foram submetidos aos algoritmos. Estes conjuntos estão divididos em características de localização dos consumidores, número de clientes atendidos por um veículo e variação do tamanho e do posicionamento das janelas de tempo. O primeiro conjunto de instâncias R contém clientes aleatórios, o segundo conjunto de instâncias C é composto por clientes agrupados em termos de localização, e por fim o último conjunto de clientes, RC, possui ambas

características apresentadas nos grupos anteriores.

Deste modo, o NSGA-III foi executado quatro vezes para cada instância de teste, no qual o problema completo foi considerado. As soluções retornadas por este algoritmo serviram para a análise de comparação do conflito e harmonia feito pela Árvore de Agregação. Além destas, a Árvore de Agregação ainda considerou dez mil soluções geradas aleatoriamente para o MOPRV Completo. A partir das árvores geradas, o conflito, bem como a harmonia entre os objetivos do problema foram verificados. E foi observado que a agregação do objetivo que minimiza a distância percorrida não é viável devido seu alto índice de conflito. Por outro lado, verificou-se que a agregação entre os objetivos que consideram a minimização da violação da restrição de tempo e a espera dos veículos nos consumidores possuem pouco conflito, podendo ser agregados em um único objetivo. Da mesma forma, os objetivos que visam a minimização da maior rota, diferença entre a maior e a menor rota e o número de veículos foram agregados em um único objetivo, devido seu alto índice de harmonia. Assim, uma formulação do problema com objetivos agregados foi proposta.

O NSGA-II proposto foi então executado quatro vezes para cada instância de teste. Este algoritmo otimizou o problema com os objetivos agregados. Assim, os resultados apresentados pelo NSGA-II para o problema reduzido indicaram que a redução do problema foi extremamente viável. Isto porque uma porcentagem pequena de soluções do problema reduzido foi dominada por soluções do problema completo. Na maioria dos casos, o NSGA-II ainda apresentou melhores soluções do que o NSGA-III.

Capítulo 7

Considerações Finais

Este capítulo apresenta as conclusões do trabalho de dissertação. Inicialmente, ressaltase a necessidade de uma exploração de abordagem com muitos objetivos para problemas de roteamento de veículos, bem como de variações dessa classe de problemas. Posteriormente, discorre-se também acerca dos algoritmos implementados e das formulações completa e reduzida do problema de roteamento de veículos com muitos objetivos e janelas de tempo flexíveis. Por fim, são apresentadas sugestões de trabalhos futuros.

7.1 Conclusão

Diversos problemas reais de roteamento de veículos possuem uma natureza multiobjetivo, é possível verificar ainda que os PRVs envolvem muitos aspectos que são considerados no processo de logística, porém, abordagens do PRVJT que procuram otimizar muitos objetivos são ainda pouco exploradas na literatura quando comparados a problemas mono-objetivos. Dessa forma, este trabalho inicia com uma investigação dos diversos objetivos que podem ser tratados no problema de roteamento de veículos, bem como no problema de roteamento de veículos com janelas de tempo. Além disso, uma investigação ainda é feita nas diversas variações que podem ser geradas a partir do PRV clássico. Foi observado que a principal ferramenta para a resolução deste problema são os Algoritmos Evolucionários, no âmbito multiobjetivo, Algoritmos Evolucionários Multiobjetivo (MOEA) são utilizados. Entretanto, diversas pesquisas indicaram que a medida que o número de objetivos em um problema de otimização cresce, menor é a eficiência da maioria dos MOEAs. Diante deste contexto, técnicas de otimização de problemas

com muitos objetivos foram estudadas. Assim, problemas de roteamento de veículos que envolvem muitos objetivos ainda são poucos estudados na literatura.

A fim de preencher esta lacuna, este trabalho apresentou uma formulação matemática e a definição de um novo problema de roteamento de veículos com muitos objetivos denominado de: problema de roteamento de veículos com muitos objetivos e janelas de tempo flexíveis. Na formulação proposta, o problema é composto por seis objetivos referentes à distância total percorrida, número de rotas necessárias, atraso na entrega aos clientes, espera dos veículos nos consumidores, *makespan* e balanceamento de rota. Estes objetivos estão ligados ao custo de transporte, satisfação dos consumidores e satisfação dos motoristas.

Devido à complexidade computacional do problema, bem como a ineficiência de algoritmos evolucionários para resolver problemas com muitos objetivos, este trabalho apresentou um conjunto de soluções heurísticas para abordá-lo, uma vez que algoritmos exatos não seriam capazes de solucionar problemas de grande magnitude, que são, geralmente, os problemas encontrados no cotidiano das grandes empresas. Assim, este trabalho explorou as especificidades de Algoritmos Evolucionários Multiobjetivo, bem como de algoritmos de redução de dimensionalidade em objetivos redundantes. Deste modo, este trabalho apresentou três algoritmos para a resolução do MOPRV: NSGA-II, NSGA-III e Árvores de Agregação.

Uma formulação do modelo reduzido que é menos complexo de ser otimizado através de algoritmos evolucionários mais simples é proposta. A fim de alcançar este objetivo, foi utilizado um NSGA-III e Árvores de Agregação para otimizar, analisar e reduzir o número de objetivos do problema. A formulação completa foi em seguida reduzida e otimizada por um NSGA-II e a verificação do comportamento das soluções com os objetivos agregados pôde ser feita. Portanto, observou-se que em muitos casos de teste os objetivos relacionados com a violação da restrição de tempo e a espera dos veículos nos consumidores estão em harmonia e poderiam ser agregados. Do mesmo modo, o balanceamento de rota e a maior distância de uma rota foram agregados pelo seu alto índice de harmonia. Estes dois objetivos podem também ser agregados com o objetivo que minimiza o número de veículos. Embora este último objetivo apresentou uma grande harmonia com a maioria dos objetivos, o número de veículos teve a maior harmonia com o objetivo de balanceamento de rota. A distância total percorrida foi o objetivo que teve o comportamento mais inconstante, estando muitas vezes em conflito com todos os outros objetivos e sendo muitas vezes agregado com qualquer um dos objetivos. Assim, este objetivo foi isolado na formulação reduzida para que não haja uma distorção significativa

na representação das soluções do problema original.

Entretanto, experimentos demonstraram que as soluções para o problema reduzido possuem bons valores para todos os objetivos quando comparado com as soluções do problema completo. Em muitos casos, as soluções do problema reduzido foram melhores que as soluções do problema completo. Assim, conclui-se que muitos dos objetivos propostos para o MOPRV estão em harmonia e que podem ser agregados para reduzir o tamanho e a complexidade do problema. Deste modo, nossa metodologia demonstrou que é capaz de otimizar problemas com muitos objetivos, uma vez que o fardo de estudar a agregação é muitas vezes menor do que o custo de otimizar todos os objetivos simultaneamente. Em outras palavras, os resultados demonstraram que é mais vantajoso visualizar a relação entre os objetivos do MOPRV e em seguida otimizar o problema com menos objetivos do que tentar otimizar o problema considerando todos os objetivos do MOPRV. Isto é, para o problema tratado, a utilização da Árvore de Agregação, para redução dos objetivos, e do NSGA-II, para otimização do problema com menos objetivos, se fez mais eficiente do que apenas aplicar o NSGA-III para a resolução do problema com todos os objetivos. Além do ganho na qualidade das soluções apresentadas quando se abordou o problema com menos objetivos, o custo de se utilizar a Árvore de Agregação em conjunto com o NSGA-II é muito menor do que utilizar um algoritmo tão complexo como o NSGA-III.

7.2 Trabalhos Futuros

Uma vez que as técnicas utilizadas se mostraram promissoras para solucionar problemas de roteamento de veículos com muitos objetivos, trabalhos futuros poderão aplicar estas mesmas técnicas considerando probabilidade de quebra das restrições (mesmo que transformadas em objetivos) para caso de otimização robusta das janelas de tempo, trabalhar com previsões futuras de curto e longo prazo colocando custos de veículos, depreciação, empregados e capacidades, além de considerar um número finito de veículos. Pode-se ainda flexibilizar a capacidade de carga dos veículos e considerar múltiplos depósitos. Além disso, as técnicas utilizadas neste trabalho podem ser aplicadas para solucionar outros problemas de otimização combinatória com muitos objetivos.

Como visto no processo de análise realizado em cima dos resultados das Árvores de Agregação, um estudo foi feito em cima de cada árvore específica gerada para uma instância do problema e das árvores médias. Assim, trabalhos futuros podem estudar

estratégias que representem melhor árvores “médias”, ou seja, algoritmos mais inteligentes que retornem uma única árvore que apresenta o comportamento de todos os objetivos em um conjunto de instâncias. Pode-se ainda pensar em estratégias para agregação de mais de um objetivo a cada iteração se houver empate entre objetivos, bem como pensar em novos critérios de desempate.

Outro aspecto que pode ser tratado em trabalhos futuros relaciona-se às características dos NSGAs. Assim, trabalhos futuros poderão estudar novas representações de indivíduos que possibilitem uma maior variedade no número de rotas. Com a mudança da representação dos indivíduos, novos métodos de mutação e cruzamento podem ser introduzidos nos algoritmos a fim de obter melhores resultados. Pode-se ainda, em trabalhos futuros, empregar técnicas de busca local multiobjetivo para os MOEAs propostos. Todas essas mudanças possibilitariam uma comparação dos resultados apresentados neste trabalho com as novas técnicas empregadas. Além disso, existem outras métricas para avaliar uma solução para problemas de otimização com múltiplos objetivos, cada qual avalia uma determinada característica desejável para as soluções. Assim, métricas como o hipervolume, espaçamento e cardinalidade podem ser utilizadas para comparar as soluções de Pareto.

Referências Bibliográficas

- Achuthan, N. R., Caccetta, L. e Hill, S. P.: 2003, An improved branch-and-cut algorithm for the capacitated vehicle routing problem., *Transportation Science* **37**(2), 153–169.
- Agrawal, G., bloebaum, K. e Lewis, K.: 2005, Intuitive design selection using visualized n-dimensional pareto frontier, *American Institute of Aeronautics and Astronautics* .
- Alvarenga, G. B.: 2013, *Um algoritmo híbrido para o problema de roteamento de veículos estático e dinâmico com janela de tempo*, Tese - doutorado em ciência da computação, Universidade Federal de Minas Gerais.
- Associação Brasileira de Logística*: 2008.
URL: *www.abralog.org.br*
- Auger, A., Bader, J., Brockhoff, D. e Zitzler, E.: 2012, Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications, *Theoretical Computer Science* **425**, 75–103.
- Baños, R., Ortega, J., Gil, C., Fernández, A. e de Toro, F.: 2013, A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows, *Expert Systems with Applications* **40**(5), 1696 – 1707.
- Baràn, B. e Schaerer, M.: 2003, A multiobjective ant colony system for vehicle routing problem with time windows, *Proceedings of the twenty-first IASTED International Conference on Applied Informatics*, pp. 97–102.
- Batista, L. S., Campelo, F., Guimarães, F. G. e Ramírez, J. A.: 2011, Pareto cone ϵ -dominance: Improving convergence and diversity in multiobjective evolutionary algorithms., **6576**, 76–90.
- Beham, A.: 2007, Parallel tabu search and the multiobjective capacitated vehicle routing problem with soft time windows, *Proceedings of the 11th International Conference on Computer Aided Systems Theory*, EUROCAST'07, Springer-Verlag, pp. 829–836.
- Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., Mack, R. G. e Prutzman, P. J.: 1983, Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer, *Interfaces* **13**(6), 4–23.

- Bentley, P. J. e Corne, D. W. (eds): 2002, *Creative Evolutionary Systems*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Bentley, P. J. e Wakefield, J. P.: 1998, Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms, in P. K. Chawdhry, R. Roy e R. K. Pant (eds), *Soft Computing in Engineering Design and Manufacturing*, Springer-Verlag, pp. 231–240.
- Beume, N., Naujoks, B. e Emmerich, M.: 2007, Sms-emoa: Multiobjective selection based on dominated hypervolume, *European Journal of Operational Research* **181**(3), 1653–1669.
- Bhusiri, N., Qureshi, A. G. e Taniguchi, E.: 2014, The trade-off between fixed vehicle costs and time-dependent arrival penalties in a routing problem, *Transportation Research Part E: Logistics and Transportation Review* **62**(0), 1 – 22.
- Bowerman, R., Hall, B. e Calami, P.: 1995, A multiobjective optimization approach to urban school bus routing : Formulation and solution method, *Transportation Research: Part A, Policy and Practice* **29**, 107–123.
- Brockhoff, D. e Zitzler, E.: 2006, Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization, in T. Runarsson et al. (eds), *Conference on Parallel Problem Solving from Nature (PPSN IX)*, Vol. 4193 of *LNCS*, Springer, Berlin, Germany, pp. 533–542.
- Brockhoff, D. e Zitzler, E.: 2007, Dimensionality reduction in multiobjective optimization: The minimum objective subset problem, pp. 423–429.
- Brown, G. e Graves, G. W.: 1981, Real-time dispatch of petroleum tank trucks, *Management Science* **27**(1), 19–32.
- Carvalho, A. B. e Pozo, A.: 2012, Measuring the convergence and diversity of cdas multi-objective particle swarm optimization algorithms: A study of many-objective problems, *Neurocomput.* **75**(1), 43–51.
- Centro de Estudos em Logística*: 2007.
URL: www.coppead.ufrj.br
- Chiang, C.-P.: 2008, An efficiency frontier approach for the vehicle routing problem in a fluctuant demand environment, *Natural Computation* **1**, 525–530.
- Coello, C. C., Lamont, G. e van Veldhuizen, D.: 2007, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Genetic and Evolutionary Computation, 2 edn, Springer, Berlin, Heidelberg.
- Cook, S. A.: 1971, The complexity of theorem-proving procedures, *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, ACM, pp. 151–158.

- Corberan, A., Fernandez, E., Laguna, M. e Marti, R.: 2002, Heuristic solutions to the problem of routing school buses with multiple objectives, *Journal of the Operational Research Society* **53**, 427–435.
- Dantzig, G. B. e Ramser, J. H.: 1959, The truck dispatching problem, *Management Science* **6**(1), 80–91.
- Das, I. e Dennis, J. E.: 1998, Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems, *SIAM J. on Optimization* **8**(3), 631–657.
- Deb, K.: 2001, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, NY, USA.
- Deb, K.: 2008, Introduction to evolutionary multiobjective optimization, in J. Branke, K. Deb, K. Miettinen e R. Słowiński (eds), *Multiobjective Optimization*, Vol. 5252 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 59–96.
- Deb, K.: 2009, Multi-objective optimization using evolutionary algorithms.
- Deb, K. e Jain, H.: 2014, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints, *IEEE Computational Intelligence Society* **18**(4), 577–601.
- Deb, K. e Saxena, D.: 2005, On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems, *Kangal report* .
- Deb, K., Pratap, A., Agarwal, S. e Meyarivan, T.: 2002, A fast and elitist multiobjective genetic algorithm: Nsga-ii, *Trans. Evol. Comp* **6**(2), 182–197.
- Deep, K. e Mebrahtu, H.: 2011, Combined mutation operators of genetic algorithm for the travelling salesman problem, *International Journal of Combinatorial Optimization Problems and Informatics* **2**(3), 1–23.
- Dell’Amico, M., Righini, G. e Salani, M.: 2006, A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection, *Transportation Science* **40**(2), 235–247.
- Drechsler, N., Drechsler, R. e Becker, B.: 2001, Multi-objective optimisation based on relation favour, pp. 154–166.
- Eddy, J. e Lewis, K. E.: 2002, Visualization of multidimensional design and optimization data using cloud visualization, *ASME Design Engineering Technical Conferences and Computers and Information* .
- El-Sherbeny, N.: 2001, *Resolution of a vehicle routing problem with multi-objective simulated annealing method*, PhD thesis, Faculté Polytechnique de Mons, Mons, Belgique.

- Esbensen, H. e Kuh, E. S.: 1996, Explorer: an interactive floorplaner for design space exploration, pp. 356–361.
- Evans, S. e Norback, J.: 1985, The impact of a decision-support system for vehicle routing in a foodservice supply situation, *The Journal of the Operational Research Society* **36**(2), 467–472.
- F. Doerner, K. e F. Hartl, R.: 2008, The vehicle routing problem: Latest advances and new challenges, Vol. 43 of *Operations Research/Computer Science Interfaces Series*, Springer, chapter 0, pp. 527–550.
- Fang, H., Wang, Q., Tu, Y.-C. e Horstemeyer, M. F.: 2008, An efficient non-dominated sorting method for evolutionary algorithms, *Evolutionary Computation* **16**(3), 355–384.
- Fisher, M. L., Greenfield, A. J., Jaikumar, R. e Lester, J. T.: 1982, A computerized vehicle routing application, *Interfaces* **12**(4), 42–52.
- Fonseca, C. M. e Fleming, P. J.: 1993, Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 416–423.
- Freitas, A. R. R., Fleming, P. J. e Guimarães, F. G.: 2013, A non-parametric harmony-based objective reduction method for many-objective optimization, *Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC '13*, IEEE Computer Society, pp. 651–656.
- Freitas, A. R. R., Fleming, P. J. e Guimarães, F. G.: 2015, Aggregation trees for visualization and dimension reduction in many-objective optimization, *Information Sciences* **298**(0), 288 – 314.
- Garcia-Najera, A. e Bullinaria, J. A.: 2011, An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows, *Computers & Operations Research* **38**(1), 287 – 300.
- Garey, M. R. e Johnson, D. S.: 1979, Computers and intractability: A guide to the theory of np-completeness.
- Garza-Fabre, M., Pulido, G. T., Coello, C. A. e Rodriguez-Tello, E.: 2011, Effective ranking + speciation = many-objective optimization., pp. 2115–2122.
- Geiger, M.: 2003, A computational study of genetic crossover operators for multi-objective vehicle routing problem with soft time windows, in W. Habenicht, B. Scheubrein e R. Scheubrein (eds), *Multi-Criteria- und Fuzzy-Systeme in Theorie und Praxis*, Deutscher Universitätsverlag, pp. 191–207.
- Giagkiozis, I., Purshouse, R. e Fleming, P.: 2013, Generalized decomposition, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **7811 LNCS**, 428–442.

- Goldberg, D. E.: 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Golden, B. L., Assad, A. A. e Wasil, E. A.: 2001, The vehicle routing problem, Society for Industrial and Applied Mathematics, chapter Routing Vehicles in the Real World: Applications in the Solid Waste, Beverage, Food, Dairy, and Newspaper Industries, pp. 245–286.
- Golden, B. L. e Wasil, E. A.: 1987, Computerized vehicle routing in the soft drink industry, *Oper. Res.* **35**(1), 6–17.
- Horn, J., Nafpliotis, N. e Goldberg, D. E.: 1994, A niched pareto genetic algorithm for multiobjective optimization, *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, Vol. 1, pp. 82–87.
- Hughes, E. J.: 2005, Evolutionary many-objective optimisation: many once or one many?, pp. 222–227.
- Ioannou, G., Kritikos, M. e Prastacos, G.: 2003, A problem generator-solver heuristic for vehicle routing with soft time windows, *Omega - The International Journal of Management Science* **31**, 41–53.
- Ishibuchi, H., Tsukamoto, N. e Nojima, Y.: 2008, Evolutionary many-objective optimization: A short review., pp. 2419–2426.
- Jozefowicz, N., Semet, F. e Talbi, E. G.: 2008, The vehicle routing problem.
- Karabulut, K. e Tasgetiren, M. F.: 2014, A variable iterated greedy algorithm for the traveling salesman problem with time windows, *Information Sciences* **279**(0), 383 – 395.
- Karg, R. L. e Thompson, G. L.: 1964, A heuristic approach to solving travelling salesman problems, *Management Science* **10**(2), 225–248.
- Khare, V., Yao, X. e Deb, K.: 2003, Performance scaling of multi-objective evolutionary algorithms, *Proceedings of the 2Nd International Conference on Evolutionary Multi-criterion Optimization*, EMO'03, Springer-Verlag, pp. 376–390.
- Knowles, J. e Corne, D.: 2007, Quantifying the effects of objective space dimension in evolutionary multiobjective optimization, EMO'07, Springer-Verlag, pp. 757–771.
- Kong, W., Ding, J., Chai, T. e Sun, J.: 2010, Large-dimensional multi-objective evolutionary algorithms based on improved average ranking, *IEEE Decision and Control* pp. 502–507.
- Lacomme, P., Prins, C. e Sevaux, M.: 2003, Multiobjective capacitated arc routing problem., *EMO*, Vol. 2632 of *Lecture Notes in Computer Science*, Springer, pp. 550–564.

- Lacomme, P., Prins, C. e Sevaux, M.: 2006, A genetic algorithm for a bi-objective capacitated arc routing problem., *Computers & OR* **33**, 3473–3493.
- Larsen, J. e Danmarks, T.: 1999, Parallelization of the vehicle routing problem with time windows.
- Laumanns, M., Thiele, L., Deb, K. e Zitzler, E.: 2002, Combining convergence and diversity in evolutionary multiobjective optimization, *Evol. Comput.* **10**(3), 263–282.
- Lee, T.-R. e Ueng, J.-H.: 1998, A study of vehicle routing problem with load balancing, *International Journal of Physical Distribution and Logistics Management* **29**, 646–648.
- Li, H. e Lim, A.: 2003, Local search with annealing-like restarts to solve the vrptw, *European Journal of Operational Research* **150**(1), 115 – 127.
- Lin, C., Choy, K., Ho, G., Chung, S. e Lam, H.: 2014, Survey of green vehicle routing problem: Past and future trends, *Expert Systems with Applications* **41**(4, Part 1), 1118 – 1138.
- López, J., Coello, C. A. e Chakraborty, D.: 2008, Objective reduction using a feature selection technique, *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, ACM, pp. 673–680.
- López, J. e Coello, C. A.: 2009, Some techniques to deal with many-objective problems, *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, GECCO '09, ACM, pp. 2693–2696.
- Lotov, A. V.: 2005, Approximation and visualization of pareto frontier in the framework of classical approach to multi-objective optimization., *Practical Approaches to Multi-Objective Optimization*, Vol. 04461 of *Dagstuhl Seminar Proceedings*, IBFI, Schloss Dagstuhl, Germany.
- Lust, t.: 2010, *New metaheuristics for solving MOCO problems: application to the knapsack problem, the traveling salesman problem and IMRT optimization*, PhD thesis, Faculté Polytechnique de Mons, Mons, Belgique.
- Malaquias, N.: 2006, *Uso de algoritmos genéticos para otimização de rotas de distribuição*, Dissertação - mestrado em em ciências, Universidade Federal de Uberlândia.
- Mattson, C. A. e Messac, A.: 2003, Concept selection using s-pareto frontiers, *AIAA Journal* **41**(6), 1190–1198.
- Menchik, C.: 2010, Desafios do transporte na era da logística enxuta, *In Interlog* .
- Meng, Q., Lee, D. H. e Cheu, R. L.: 2005, Multiobjective vehicle routing and scheduling problem with time window constraints in hazardous material transportation, *Journal of Transportation Engineering* (9).

- Mirabi, M., Fatemi, S. e Jolai, F.: 2010, Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem, *Robotics and Computer-Integrated Manufacturing* **26**(6), 564–569. 19th International Conference on Flexible Automation and Intelligent Manufacturing Lean manufacturing and Services.
- Murata, T. e Itai, R.: 2005, Multi-objective vehicle routing problems using two-fold emo algorithms to enhance solution similarity on non-dominated solutions, pp. 885–896.
- Nazif, H. e Lee, L. S.: 2012, Optimised crossover genetic algorithm for capacitated vehicle routing problem, *Applied Mathematical Modelling* **36**(5), 2110 – 2117.
- Obayashi, S. e Sasaki, D.: 2003, Visualization and data mining of pareto solutions using self-organizing map, pp. 796–809.
- Ombuki, B. M., Nakamura, M., Osamu, M. e Beatrice, M.: 2002, A hybrid search based on genetic algorithms and tabu search for vehicle routing.
- Purshouse, R. C. e Fleming, P. J.: 2007, On the evolutionary optimization of many conflicting objectives., *IEEE Trans. Evolutionary Computation* **11**(6), 770–784.
- Rahoual, M., Kitoun, B., Mabed, M., Bachelet, V. e Benameur, F.: 2001, Multicriteria genetic algorithms for the vehicle routing problem with time windows, *Metaheuristic International Conference (MIC'2001)*, pp. 527–532.
- Rosenberg, R. S.: 1967, *Simulation of genetic populations with biochemical properties*, Tese - doutorado, University of Michigan.
- Salomon, S., Domínguez-Medina, C., Avigad, G., Freitas, A., Goldvard, A., Schütze, O. e Trautmann, H.: 2014, Psa based multi objective evolutionary algorithms, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III*, Vol. 500 of *Studies in Computational Intelligence*, Springer International Publishing, pp. 233–259.
- Sato, H., Aguirre, H. e Tanaka, K.: 2009, Local dominance moea including control of dominance area of solutions on 0//1 multiobjective knapsack problems, *Information and Media Technologies* **4**(1), 33–43.
- Saxena, D. e Deb, K.: 2007, Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding, *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization*, EMO'07, Springer-Verlag, pp. 772–787.
- Saxena, D. K., Duro, J. A., Tiwari, A., Deb, K. e Zhang, Q.: 2013, Objective reduction in many-objective optimization: Linear and nonlinear algorithms., *IEEE Trans. Evolutionary Computation* **17**(1), 77–99.
- Saxena, D. K., Ray, T., Deb, K. e Tiwari, A.: 2009, Constrained many-objective optimization: A way forward., pp. 545–552.

- Schaffer, J. D.: 1984, *Some experiments in machine learning using vector evaluated genetic algorithms*, Tese - doutorado, Vanderbilt University.
- Singh, H., Isaacs, A. e Ray, T.: 2011, A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems., *IEEE Trans. Evolutionary Computation* **15**(4), 539–556.
- Singh, S. e Singh, V.: 2011, Three-level ahp-based heuristic approach for a multi-objective facility layout problem, *International Journal of Production Research* **49**(4), 1105–1125.
- Sivanandam, S. N. e Deepa, S. N.: 2007, *Introduction to Genetic Algorithms*, 1st edn, Springer Publishing Company, Incorporated.
- Solomon, M. M.: 1987, Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operation Research* **35**(2), 254–265.
- Sousa, J. C., B. H. A. B. R. e Silveira, A.: 2011, A multi objective approach to solve capacitated vehicle routing problems with time windows using mixed integer linear programming, *International Journal of Advanced Science and Technology* pp. 1–8.
- Srinivas, N. e Deb, K.: 1994, Multiobjective optimization using nondominated sorting in genetic algorithms, *evolutionary computer* **2**(3), 221–248.
- Taillard, r., Badeau, P., Gendreau, M., Guertin, F. e Potvin, J.-Y.: 1997, A tabu search heuristic for the vehicle routing problem with soft time windows., *Transportation Science* **31**(2), 170.
- Takahashi, R., Cunha, L. e Antunes, C. A. H. d. C.: 2012, *Manual de computação evolutiva e metaheurística*, Imprensa da Universidade de Coimbra, Coimbra.
- Tan, K. C., Cheong, C. Y. e Goh, C. K.: 2007, Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation., *European Journal of Operational Research* **177**(2), 813–839.
- Tan, K., Lee, L., Zhu, K. e Ou, K.: 2001a, Heuristic methods for vehicle routing problem with time windows., *AI in Engineering* **15**, 281–295.
- Tan, K., Lee, L., Zhu, Q. e Ou, K.: 2001b, Heuristic methods for vehicle routing problem with time windows, *Artificial Intelligence in Engineering* **15**(3), 281 – 295.
- Tasan, A. S. e Gen, M.: 2012, A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries, *Computers & Industrial Engineering* **62**(3), 755 – 761. *Soft Computing for Management Systems*.
- Toth, P. e Vigo, D. (eds): 2001, *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

- Wang, T., Wu, Z. e Mao, J.: 2007, A new method for multi-objective tdma scheduling in wireless sensor networks using pareto-based pso and fuzzy comprehensive judgement., **4782**, 144–155.
- Wassan, N. A. e Osman, I. H.: 2002, Tabu search variants for the mix fleet vehicle routing problem, *Journal of the Operational Research Society* pp. 768–782.
- Yu, B., Yang, Z. Z. e Yao, B. Z.: 2011, A hybrid algorithm for vehicle routing problem with time windows, *Expert Systems with Applications* **38**(1), 435–441.
- Zitzler, E. e Kunzli, S.: 2004, Indicator-based selection in multiobjective search, pp. 832–842.
- Zitzler, E. 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., Laumanns, M. e Thiele, L.: 2001, Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization, *in* K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou e T. Fogarty (eds), *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, International Center for Numerical Methods in Engineering, pp. 95–100.