

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**HEURÍSTICAS E ALGORITMOS EVOLUTIVOS PARA
FORMULAÇÕES MONO E MULTIOBJETIVO DO
PROBLEMA DO ROTEAMENTO MULTICAST**

MARCOS LUIZ DE PAULA BUENO

Uberlândia - Minas Gerais

2010

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



MARCOS LUIZ DE PAULA BUENO

HEURÍSTICAS E ALGORITMOS EVOLUTIVOS PARA FORMULAÇÕES MONO E MULTIOBJETIVO DO PROBLEMA DO ROTEAMENTO MULTICAST

Dissertação de Mestrado apresentada à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como parte dos requisitos exigidos para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Inteligência Artificial.

Orientadora:

Prof^a. Dr^a. Gina Maira Barbosa de Oliveira

Uberlândia, Minas Gerais
2010

Dados Internacionais de Catalogação na Publicação (CIP)

Sistema de Bibliotecas da UFU, MG, Brasil.

- B928h Bueno, Marcos Luiz de Paula, 1984-
 Heurísticas e algoritmos evolutivos para formulações mono e
 multiobjetivo do problema do roteamento multicast [manuscrito] /
 Marcos Luiz de Paula Bueno. - 2010.
 131 f. : il.
- Orientadora: Gina Maira Barbosa de Oliveira.
 Dissertação (mestrado) - Universidade Federal de Uberlândia,
 Programa de Pós-Graduação em Ciência da Computação.
 Inclui bibliografia.
1. Redes de computadores - Teses. 2. Algoritmos genéticos -
 Teses. 3. Inteligência artificial - Teses. I. Oliveira, Gina Maira
 Barbosa de. II. Universidade Federal de Uberlândia. Programa de
 Pós-Graduação em Ciência da Computação. III. Título.

CDU: 681.3.02

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da dissertação intitulada “**Heurísticas e Algoritmos Evolutivos para Formulações Mono e Multiobjetivo do Problema do Roteamento Multicast**” por **Marcos Luiz de Paula Bueno** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação**.

Uberlândia, 4 de Agosto de 2010

Orientadora:

Prof^a. Dr^a. Gina Maira Barbosa de Oliveira
Universidade Federal de Uberlândia

Banca Examinadora:

Prof. Dr. Pedro Frosi Rosa
Universidade Federal de Uberlândia

Prof. Dr. Ricardo Hiroshi Caldeira Takahashi
Universidade Federal de Minas Gerais

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Data: Agosto de 2010

Autor: **Marcos Luiz de Paula Bueno**
Título: **Heurísticas e Algoritmos Evolutivos para Formulações Mono e Multiobjetivo do Problema do Roteamento Multicast**
Faculdade: **Faculdade de Computação**
Grau: **Mestrado**

Fica garantido à Universidade Federal de Uberlândia o direito de circulação e impressão de cópias deste documento para propósitos exclusivamente acadêmicos, desde que o autor seja devidamente informado.

Autor

O AUTOR RESERVA PARA SI QUALQUER OUTRO DIREITO DE PUBLICAÇÃO DESTE DOCUMENTO, NÃO PODENDO O MESMO SER IMPRESSO OU REPRODUZIDO, SEJA NA TOTALIDADE OU EM PARTES, SEM A PERMISSÃO ESCRITA DO AUTOR.

Dedicatória

Aos meus pais.

Agradecimentos

Muitas pessoas me ajudaram ao longo deste trabalho, tanto no sentido pessoal como acadêmico. Gostaria de deixar registrado um pequeno agradecimento.

Aos meus pais, Marcos e Celina, que sempre batalharam para que eu pudesse ter uma vida digna e cheia de bons momentos. Pela vida, educação, apoio, amor e compreensão, especialmente nesses dois últimos anos. As palavras nos faltam nessas horas para expressar nossos sentimentos.

Ao meu irmão Matheus, um grande companheiro nessa existência. Aos meus familiares, primos, avôs, tios, etc. e ao Felipe Arantes (Filipêras) pela convivência.

Aos amigos do mestrado, pela amizade, empenho e dedicação durante as longas horas de estudo nas disciplinas e seminários, fundamentais ao nosso crescimento acadêmico. Em especial, ao Augusto Branquinho, Allysson e Leonardo G. Marques. Um agradecimento especial ao Branquinho, que dispôs do seu tempo para discutirmos aspectos deste trabalho, contribuindo com idéias e implementações. Aos amigos do Laboratório de Inteligência Artificial (Murillo, Enrique, Leonardo, etc.).

Aos grandes amigos da “velha guarda”, dentre eles, Ayres Barcelos e Fábio Nishyama. À querida amiga Virgínia, pelo apoio e amizade.

Ao grande professor e amigo Ibrahim Andraus Gassani, com quem tive a honra e o prazer de aprender um pouco sobre Estatística e Qualidade. Que me recebeu em sua casa para agradáveis e fantásticas aulas, sempre acompanhadas de uma boa conversa.

À minha orientadora, Gina M. B. Oliveira, pela abertura para o desenvolvimento de novas idéias, dedicação e paciência com que me orientou no desenvolvimento deste trabalho. Muitos dos frutos obtidos ao longo do curso foram possíveis devido ao seu incentivo para novas idéias durante a realização deste mestrado.

Sou grato aos professores Sandra de Amo, Ilmério Reis e Flávio Oliveira, pelas diversas oportunidades que me foram oferecidas, dentre elas a monitoria, onde pude aprender um pouco mais sobre docência. Também quero agradecer à professora Márcia A. Fernandes pelas idéias e sugestões apresentadas e pela paciência no esclarecimento de conceitos importantes para o desenvolvimento deste trabalho.

Aos professores da banca por aceitarem o convite.

À UFU por me aceitar em seu programa de pós-graduação, contribuindo ao meu aprimoramento pessoal e profissional. À FACOM com seus excelentes professores, que contribuíram na minha formação acadêmica.

Aos Coordenadores da pós, Sandra e Marcelo, pelo seu empenho em auxiliar os alunos em diversos momentos, e aos secretários da pós, Maria Helena e Erisvaldo.

Ao Maurício Garcia e à Chaus Consultoria, que me deram a chance de crescimento profissional e de conhecer profissionais de alta qualidade no mundo empresarial. Aos amigos Diego Bueno e Dyêgo Figueiredo, com os quais tive o prazer de conviver e trabalhar.

À CAPES e à UFU, pelo suporte financeiro concedido durante o curso.

A Deus, por ter colocado essas e outras pessoas na minha vida.

“Não há caminho para a paz. A paz é o caminho.”
(Mahatma Gandhi)

Resumo

Neste trabalho são investigados modelos evolutivos aplicados ao Problema do Roteamento Multicast (PRM), cujo objetivo é calcular árvores multicast a partir de um grafo conectado ponderado, otimizando uma ou mais funções objetivo relacionadas a requisitos de Qualidade de Serviço e Engenharia de Tráfego. O PRM pode ser visto como uma extensão ao conhecido Problema da Árvore de Steiner, o qual sabe-se estar na classe NP-difícil. Assim, partindo de modelos baseados em Algoritmos Genéticos Clássicos e Multiobjetivo da literatura, propomos três heurísticas a serem utilizadas nos operadores de cruzamento e mutação do modelo evolutivo resultante, sendo baseadas em estratégias mais deterministas (por meio do algoritmo de Dijkstra), mais aleatórias ou uma combinação destas. A aplicação da heurística ocorre numa fase do cruzamento/mutação conhecida como reconexão de subárvores, a qual guia a geração de novas soluções combinando partes comuns aos pais a novas partes adicionadas pelo algoritmo de reconexão.

O PRM foi considerado sob uma série de formulações (uma mono-objetivo, e sete multiobjetivo usando a dominância de Pareto), com o intuito de realizar uma extensa avaliação experimental dos métodos propostos. Foram utilizados três algoritmos evolutivos conhecidos (NSGA-II, SPEA e SPEA2) como mecanismo de busca subjacente, sobre os quais duas variações de seleção de pais (Cruzamento pela Vizinhança) também foram avaliadas.

A partir de um conjunto de oito instâncias retiradas da literatura dos problemas de Roteamento e Steiner, os resultados experimentais indicaram que além de corrigir uma heurística que potencialmente gerava indivíduos inválidos, as novas heurísticas forneceram bons resultados nas métricas de convergência e diversidade consideradas. Especificamente, a heurística combinada proveu os melhores resultados na maioria dos casos, mostrando que a estratégia de combinar um menor caminho com aleatoriedade durante as reconexões foi benéfico. Por sua vez, o uso do Crossover pela Vizinhança trouxe benefícios mais nitidamente na formulação com maior número de objetivos, na qual as instâncias apresentaram conjuntos Pareto-ótimos com maior cardinalidade. Testes de significância estatística foram aplicados, confirmando o que as estimativas pontuais e por intervalo haviam evidenciado na maior parte dos casos.

Palavras chave: roteamento multicast, otimização multiobjetivo, algoritmo genético, dominância de pareto, qualidade de serviço, engenharia de tráfego.

Abstract

In this work, we investigate evolutionary models applied to the Multicast Routing Problem (MRP), in which we want to calculate multicast trees from a connected weighted graph, optimizing one or more objective functions related to Quality of Service and Traffic Engineering requirements. MRP can be seen as an extension of the well-known Steiner Tree Problem, which is in the NP-hard class of problems. Thus, starting from models based on Canonical and Multiobjective Genetic Algorithms from the literature, we propose three heuristics to be used in crossover and mutation operators of the resultant evolutionary model, which are based on more determinists strategies (using Dijkstra's algorithm), more random ones or a combination of these. The usage of such heuristic occurs in a crossover/mutation phase known as subtrees reconnection, which guides the generation of new solutions combining common parts of parents and new parts added by such reconnection algorithm.

MRP was considered under several formulations (one mono-objective and seven multi-objective ones under Pareto dominance), aiming to achieve a comprehensive experimental evaluation of the proposed methods. Three well-known algorithms (NSGA-II, SPEA and SPEA2) were used as underlying search mechanism, in which two variations of a mating selection of parents based on neighborhood (Neighborhood Crossover) were also evaluated.

From a set of eight instances taken from Routing and Steiner problems literature, the experimental results indicated that besides fixing a heuristic that could potentially generate invalid individuals, the new heuristics returned good results in the considered convergence and diversity metrics. More specifically, the combined heuristic provided the best results in most cases, showing that the strategy of combining a shortest path with randomness along the reconnections was beneficial. The use of Neighborhood Crossover, in turn, was more noticeable on the MRP formulation in which the network instances had larger Pareto-optimal sets. Statistical significance tests were performed, supporting the evidences showed by punctual and interval estimations.

Keywords: multicast routing, multiobjective optimization, genetic algorithm, pareto dominance, quality of service, traffic engineering.

Sumário

Lista de Figuras	xxi
Lista de Tabelas	xxiii
Lista de Algoritmos	xxv
Lista de Abreviaturas e Siglas	xxvii
1 Introdução	29
1.1 Visão Geral	29
1.2 Objetivos	31
1.3 Contribuições	31
1.4 Organização do Texto	32
2 Otimização Evolutiva Multiobjetivo	33
2.1 Algoritmos Genéticos	33
2.1.1 Representação de Indivíduos e População Inicial	34
2.1.2 Cálculo da Aptidão	35
2.1.3 Seleção para Cruzamento	36
2.1.4 Cruzamento	37
2.1.5 Mutação	38
2.1.6 Sobrevivência de Indivíduos (Reinserção)	38
2.2 Problemas de Otimização Multiobjetivo	39
2.3 Algoritmos Evolutivos Multiobjetivo	41
2.3.1 <i>Non-dominated Sorting Genetic Algorithm II</i>	41
2.3.2 <i>Strength Pareto Evolutionary Algorithm</i>	43
2.3.3 <i>Strength Pareto Evolutionary Algorithm 2</i>	45
3 Problema do Roteamento Multicast	49
3.1 Formulações para o Problema do Roteamento Multicast	50
3.1.1 Formulação Mono-objetivo	50
3.1.2 Formulações Multiobjetivo	51

3.1.3	PRM e a Árvore de Steiner	54
3.2	Literatura Relacionada	55
4	Modelos Anteriores para o PRM	59
4.1	Visão Geral	59
4.2	Representação dos Indivíduos e População Inicial	60
4.3	Cálculo de Aptidão	60
4.4	Cruzamento	61
4.4.1	Reconexão de Subárvores	62
4.4.2	Complexidade da Reconexão	64
4.4.3	Geração de Indivíduos Inválidos	64
4.5	Mutação	66
4.6	Filtro	67
4.7	Seleção	67
4.7.1	Seleção para Cruzamento	67
4.7.2	Reinserção	68
4.8	Roteamento Evolutivo	68
5	Novas Abordagens para o Roteamento Evolutivo	71
5.1	Heurísticas para Reconexão de Subárvores	71
5.1.1	Heurística h2	71
5.1.2	Heurística h3	74
5.1.3	Heurística h4	75
5.2	Filtros	76
5.2.1	Filtro f_{rnd}	76
5.2.2	Filtro f_{tm}	76
5.3	Cruzamento pela Vizinhança	77
5.4	A Escolha das Funções Objetivo	79
6	Resultados Experimentais	81
6.1	Instâncias e Métricas	81
6.2	Formulação Mono-objetivo (PRMMo)	84
6.3	Formulação PRMM (dois objetivos)	86
6.3.1	Análise geral das métricas	87
6.3.2	Análise das métricas com o SPEA2	87
6.3.3	Testes de hipóteses	91
6.3.4	Considerações	93
6.4	Formulação PRFM (quatro objetivos)	93
6.4.1	Análise das métricas e testes de hipóteses	94
6.4.2	Considerações	95

6.5	Formulação PRFM (cinco objetivos)	97
6.5.1	Análise das métricas	98
6.5.2	Testes de hipóteses	99
6.6	Comparação com algoritmos da literatura	100
6.6.1	Comparação com o algoritmo de Dijkstra	101
6.6.2	Comparação com a heurística Takahashi-Matsuyama	102
6.6.3	Considerações	104
7	Conclusões	107
7.1	Considerações Finais	107
7.2	Sugestões para Trabalhos Futuros	110
	Referências Bibliográficas	113
A	Resultados Complementares	119
A.1	Formulação com dois objetivos	119
B	Algorithms to Augment Diversity and Convergence in Multiobjective Multicast Flow Routing	123

Lista de Figuras

2.1	Representação das probabilidades associadas a cada indivíduo no método de seleção pela roleta.	37
2.2	Cruzamento de um ponto.	38
2.3	Cruzamento multiponto.	38
2.4	Aplicação da mutação sobre um indivíduo por meio da troca de bit. . . .	38
2.5	Soluções Pareto-ótimas em um problema de maximização de dois objetivos.	41
2.6	O cubóide para o i -ésimo indivíduo em sua fronteira não-dominada (círculos cheios são soluções da mesma fronteira não-dominada) [18].	42
2.7	Um exemplo do cálculo de aptidão para membros da população e do arquivo, em problema de maximização de dois objetivos. Os membros da população estão representados por círculos cheios, ao passo que os membros do arquivo estão marcados com o símbolo $\times$ [64].	44
2.8	Um exemplo comparativo, em um problema de maximização de dois objetivos, entre o SPEA (lado esquerdo) e o SPEA2 (lado direito). Os indivíduos dominados estão marcados com círculos cheios, ao passo que os não-dominados estão como círculos vazios. Os valores para os indivíduos indicam a aptidão (caso SPEA) e o <i>raw fitness</i> (caso SPEA2) [62].	47
3.1	Uma instância (rede 0) para o PRMMo e PRMM.	51
3.2	Instância do PRFM - rede 1.	53
3.3	Soluções Pareto-ótimas para a instância rede 0 na formulação (<i>custo, hops</i>) do PRMM (P5).	54
3.4	Soluções Pareto-ótimas para a instância rede 1 na formulação (α_{max} , <i>custo</i> , D_{max} , <i>hops</i>) do PRFM (P6).	54
4.1	Soluções para a instância rede 0, onde $D = \{1, 8, 9, 12, 13\}$	60
4.2	Etapas de cruzamento, considerando $D = \{1, 8, 11\}$. Legenda: (<i>i</i>): identificação de subárvores comuns aos pais (subárvores em negrito); (<i>iii</i>): reconexão de subárvores (vértices adicionados em negrito) ; (<i>iv</i>): poda (vértices a serem removidos estão marcados).	62

4.3	Um exemplo de conflito durante a reconexão de subárvores. Após a segunda reconexão, os vértices em negrito tornam inválidas as duas possíveis soluções.	65
4.4	Produção média de indivíduos inválidos por geração.	66
4.5	Percentual médio de indivíduos inválidos remanescentes ao final de cada geração (após reinserção).	66
4.6	Um exemplo de aplicação do operador de mutação, considerando que $\lceil 20\% \rceil$ de vértices são desconectados.	67
5.1	Um exemplo de reconexão utilizando h2. Vértices em negrito correspondem àqueles em comum entre T e s_1	73
5.2	Um exemplo de reconexão utilizando h3. Arestas aleatórias (dadas pelo conjunto R) são inseridas em T até que uma delas contenha um vértice que exista em alguma subárvore s_i não reconectada, o que é mostrado na figura pelo vértice em negrito. Neste momento, a respectiva s_i que contém tal vértice é rotacionada e unida a T	75
5.3	Tipos de Cruzamento pela Vizinhaça implementados. Aqui, $ P = 20$ e o comprimento da faixa de permutação foi definida como 30% em ambos os casos.	79
6.1	Percentual de convergência para cada heurística no PRMMo. Cada grupo de quatro experimentos corresponde aos resultados obtidos por cada heurística em cada instância ($rede, d_{max}$).	85
6.2	Médias de ganho do AG sobre o algoritmo SPT.	103
6.3	Médias de ganho do AG sobre o algoritmo TM.	105

Lista de Tabelas

2.1	Exemplo de uma população com cinco indivíduos, seus valores de aptidão e de probabilidades p_i	36
2.2	Exemplo de torneio simples.	37
6.1	Características das instâncias utilizadas neste trabalho.	81
6.2	Valores de custo ótimo (<i>custo*</i>) sujeitos a restrição de d_{max} no PRMMo. Os valores de <i>custo*</i> em negrito correspondem aos menores valores de custo possíveis em cada rede, independentemente de quão grande seja d_{max}	84
6.3	Convergência média geral obtida por cada heurística no PRMMo.	86
6.4	Cardinalidades dos conjuntos Pareto-ótimos P^* para cada instância no PRMM.	87
6.5	Médias e intervalos de confiança de 95% para a métrica <i>Error Rate</i> (<i>er</i>) utilizando o SPEA2.	89
6.6	Médias e intervalos de confiança de 95% para a métrica <i>Generational Distance</i> (<i>gd</i>) utilizando o SPEA2. Os valores estão multiplicados por 10^3 . . .	89
6.7	Médias e intervalos de confiança de 95% para a métrica <i>Spread</i> (Δ) utilizando o SPEA2. Os valores estão multiplicados por 10.	90
6.8	Médias gerais para o NSGA-II, SPEA e SPEA2 nas métricas <i>er</i> , <i>gd</i> e Δ no PRMM. Os valores de <i>gd</i> estão multiplicados por 10^3 e os de Δ por 10. . .	91
6.9	Testes de hipóteses para evidenciar diferença estatisticamente significativa entre as médias de <i>h4</i> e <i>h3</i> , considerando um nível de significância $\alpha = 10\%$ bicaudal.	92
6.10	Testes de hipóteses para evidenciar diferença estatisticamente significativa entre as médias de <i>h4</i> e <i>h2</i> , considerando um nível de significância $\alpha = 10\%$ bicaudal.	92
6.11	Cardinalidades dos conjuntos Pareto-ótimos P^* para cada instância no PRFM de quatro objetivos (P6).	94
6.12	Métricas de convergência (<i>er</i> , <i>gd</i> e <i>ps</i>) no PRFM (P6).	96
6.13	Métrica de diversidade (<i>m3</i>) no PRFM (P6).	96
6.14	Testes de hipóteses para diferença entre as médias do SPEA2, NSGA-II e SPEA (utilizando <i>h4</i>), com $\alpha = 10\%$ bicaudal.	97

6.15	Testes de hipóteses para diferença entre as médias de h4, h3 e h2 (utilizando SPEA2), com $\alpha = 10\%$ bicaudal.	97
6.16	Cardinalidades dos conjuntos Pareto-ótimos P^* para cada instância no PRFM de cinco objetivos (P7).	97
6.17	Métricas de convergência (er , gd e ps) no PRFM (P7) utilizando h4.	98
6.18	Métrica de diversidade ($m3$) no PRFM (P7) utilizando h4.	98
6.19	A métrica <i>two set coverage</i> aplicada na comparação entre SPEA2nc1 e SPEA2nc2.	99
6.20	Testes de hipóteses para diferença entre as médias do SPEA2 sem NC, com NC1 e com NC2, considerando $\alpha = 10\%$ bicaudal.	100
6.21	Percentuais mostrando os ganhos do AG sobre o algoritmo SPT, considerando valores rígidos para d_{max}^{spt} e tolerância de 10%	102
6.22	Percentuais mostrando os ganhos do AG sobre o algoritmo SPT, considerando tolerâncias de 15% e 20% para o d_{max}^{spt}	102
6.23	Percentuais mostrando os ganhos do AG sobre o algoritmo TM, considerando $c_{AG} \leq c_{TM}$ e tolerância de 10%	104
6.24	Percentuais mostrando os ganhos do AG sobre o algoritmo TM, considerando tolerâncias de 15% e 20% para c_{TM}	104
6.25	Tempo médio (em segundos) para construir uma rota multicast utilizando SPEA2.	104
7.1	Principais publicações decorrentes da pesquisa realizada.	110
A.1	Médias e intervalos de confiança de 95% para as métricas avaliadas em cada instância no NSGA-II (Problemas P1 a P3). Os valores de $\overline{gd}$ estão multiplicados por 10^3 e os de $\overline{\Delta}$ por 10.	119
A.2	Médias e intervalos de confiança de 95% para as métricas avaliadas em cada instância com NSGA-II (Problemas P4 e P5). Os valores de $\overline{gd}$ estão multiplicados por 10^3 e os de $\overline{\Delta}$ por 10.	120
A.3	Médias e intervalos de confiança de 95% para as métricas avaliadas em cada instância no SPEA (Problemas P1 a P3). Os valores de $\overline{gd}$ estão multiplicados por 10^3 e os de $\overline{\Delta}$ por 10.	120
A.4	Médias e intervalos de confiança de 95% para as métricas avaliadas em cada instância com SPEA (Problemas P4 e P5). Os valores de $\overline{gd}$ estão multiplicados por 10^3 e os de $\overline{\Delta}$ por 10.	121

List of Algorithms

1	Esquema geral de um Algoritmo Genético	34
2	Cálculo de <i>crowding distance</i> no NSGA-II (I)	42
3	Non-dominated Sorting Genetic Algorithm II (NSGA-II)	43
4	Procedimento de clusterização no SPEA (E)	45
5	Esquema geral do SPEA	46
6	Esquema geral do SPEA2	48
7	Heurística h1 para reconexão de subárvores (G, r, D, s)	63
8	AG mono-objetivo para o roteamento (Tp, Ng)	68
9	Heurística h2 para reconexão de subárvores (G, r, D, s)	72
10	Heurística h3 para reconexão de subárvores (G, r, D, s)	74
11	Filtro f_{tm} (T)	77

Lista de Abreviaturas e Siglas

AG	Algoritmo Genético
BE	<i>Best Effort</i>
CSTP	<i>Constrained Steiner Tree Problem in Graphs</i>
DVMRP	Multicast Distance Vector Routing Protocol
EMO	<i>Evolutionary Multiobjective Optimization</i>
ET	Engenharia de Tráfego
IC	Intervalo de confiança
ISP	Internet Service Provider
MOEA	<i>Multiobjective Evolutionary Algorithm</i>
MOP	<i>Multiobjective Optimization Problem</i>
MOSPF	Multicast Open Shortest Path Protocol
MRP	Multicast Routing Problem
NC	<i>Neighborhood Crossover</i>
NSGA-II	Non-dominated Sorting Genetic Algorithm II
OSPF	Open Shortest Path Protocol
PRM	Problema do Roteamento Multicast
PRMMo	Problema do Roteamento Multicast Mono-objetivo
PRMM	Problema do Roteamento Multicast Multiobjetivo (sem fluxo)
PRFM	Problema do Roteamento de Fluxo Multicast Multiobjetivo
QoS	<i>Quality of Service</i>
RIP	Routing Information Protocol
SPEA	Strength Pareto Evolutionary Algorithm
SPEA2	Strength Pareto Evolutionary Algorithm 2
SPT	Shortest Path Tree (algoritmo de Dijkstra)
STP	<i>Steiner Tree Problem in Graphs</i>
TM	Heurística de Takahashi & Matsuyama

Capítulo 1

Introdução

1.1 Visão Geral

O sucesso da Internet deve-se, em boa parte, à sua flexibilidade expressa na possibilidade de conectar máquinas com diferentes configurações de hardware e software. Assim, o custo de sua implementação pode ser relativamente baixo, oferecendo uma interessante relação de custo $\times$ benefício aos ISPs (*Internet Service Providers*) e seus usuários. Por outro lado, tal flexibilidade provoca um impacto na qualidade do serviço padrão oferecido na Internet, conhecido como “Melhor Esforço” (*Best Effort* ou BE), cujo lema principal pode ser expresso por “acesso para todos, qualidade para ninguém”.

Desta maneira, o serviço BE pode não ser o mais adequado a todos os usuários e aplicações, uma vez que estes podem necessitar de serviços que forneçam vários tipos de garantias em relação à qualidade da comunicação. Sob o ponto de vista de um usuário, o termo qualidade pode representar a garantia de certos requisitos, tais como uma comunicação rápida, confiável, com pequeno *delay* e reduzido custo. Estes e outros requisitos usualmente estão incorporados no conceito de Qualidade de Serviço (*Quality of Service* ou QoS) [49], [52], que dentre as suas diversas ramificações, está relacionado com tratamentos especiais requisitados para que um serviço possa oferecer garantias de desempenho, impactando na satisfação dos usuários. Para que isso seja possível, a rede tem de garantir o desempenho de suas aplicações, o que também requer que protocolos trabalhem nesse sentido. Em um sentido mais amplo, além de prover QoS aos usuários, a Engenharia de Tráfego (ET) [4] preocupa-se também com otimização da utilização da rede, por meio de, por exemplo, distribuição de tráfego e escolha de rotas que reduzam o gargalo e o uso médio dos enlaces.

Um importante componente, tanto no serviço BE quanto nos esquemas baseados em QoS, são os algoritmos e protocolos para roteamento, os quais, entre outras funções, efetuam o cálculo de rotas. Tais rotas podem ser usadas para entregar dados em diversos modos, como rotas unicast (uma origem a um destino) e rotas multicast (múltiplos desti-

nos). Existem protocolos para roteamento unicast na Internet, tais como o RIP (*Routing Information Protocol*) e o OSPF (*Open Shortest Path First*), os quais usam apenas uma métrica (peso de aresta) durante o cálculo de rotas, eventualmente armazenando mais de uma rota, uma para cada métrica, tais como *hops count*, custo e *delay*. No caso de protocolos para roteamento multicast, como o DVMRP (*Distance Vector Multicast Routing Protocol*) [55] e o MOSPF (*Multicast Open Shortest Path First*) [37], vários deles calculam rotas para o grupo multicast baseadas apenas em uma métrica configurável, utilizando algoritmos que apesar de eficientes, não podem garantir a otimalidade da rota em muitos casos.

Há também propostas para roteamento multicast e unicast que consideram formulações simplificadas ou heurísticas para o problema, de modo a tornar o cálculo viável. Um exemplo pode ser visto no trabalho de Wang & Crowcroft [56] em que foram propostos dois algoritmos polinomiais para as versões centralizada e distribuída do roteamento unicast com duas métricas (*delay* e largura de banda). A formulação adotada possibilitou que os algoritmos fornecessem rotas ótimas a uma complexidade de tempo baixa. No entanto, em formulações cujo objetivo seja encontrar uma rota unicast contemplando duas ou mais métricas (funções objetivo) aditivas ou multiplicativas quaisquer, os autores demonstraram se tratar de um problema NP-completo em sua versão de decisão.

O Problema do Roteamento Multicast (PRM) pode ser formalizado considerando que a topologia de rede é modelada como um grafo direcionado conectado, com vértices representando *hosts*, e arestas os enlaces de comunicação. Pesos podem ser associados com cada aresta, representando características dos enlaces da rede, tais como custo, *delay*, capacidade e tráfego atual. Assim, dado um grafo G e um novo fluxo ϕ , o problema se resume em calcular árvores enraizadas T de G para carregar ϕ em G , iniciando em um vértice fonte passando obrigatoriamente por um subconjunto de vértices de G chamados destinos (o grupo multicast). As soluções para o problema devem otimizar uma ou mais funções objetivo.

Assim, o PRM pode ser considerado como uma extensão do Problema da Árvore de Steiner em Grafos (*Steiner Tree Problem in Graphs* ou STP), problema este em que se deseja encontrar uma árvore de custo mínimo que inclua um subconjunto de vértices. Karp [31] mostrou que o STP é NP-difícil por uma redução do problema *Exact Cover*. Por esta razão, muitas técnicas têm sido utilizadas para abordar o PRM, tais como algoritmos heurísticos, de aproximação e técnicas de Inteligência Artificial [1, 2, 5, 16, 21, 22, 44, 51]. No presente trabalho, foram adotados os Algoritmos Genéticos (AGs) mono e multiobjetivos. AGs têm sido utilizados em diversos trabalhos relacionados ao roteamento com interessantes resultados [10, 15, 16, 24, 39–41, 43, 45, 60]. Dentre as suas vantagens, pode-se citar a possibilidade de calcular múltiplas rotas em uma única execução - uma vez que AGs trabalham com uma população de soluções - e a interessante relação entre a qualidade da rota e a complexidade da computação efetuada. Tais características estão diretamente

relacionadas ao PRM, especialmente no caso multiobjetivo, pois deseja-se um conjunto de soluções ótimas calculadas rapidamente.

1.2 Objetivos

De um modo genérico, o principal objetivo deste trabalho é a investigação de modelos evolutivos que possam ser aplicados com sucesso em formulações do Problema do Roteamento Multicast. Mais especificamente, os seguintes aspectos foram enfatizados na pesquisa:

- investigação de novos operadores genéticos (cruzamento e mutação) específicos para o problema: como podemos utilizar a simulação de herança genética para produzir filhos que sejam tão bem ou mais adaptados que os pais?
- pesquisa de modelos de MOEA (*Multiobjective Evolutionary Algorithms*) existentes na literatura: como as diferentes formas de utilização de dominância de Pareto impactam na convergência e diversidade no PRM?
- avaliação de formulações conhecidas e novas para o PRM: de que forma os operadores e seus modelos de MOEA subjacentes se comportam em diferentes formulações do PRM?

Para isso, adotou-se uma metodologia para avaliação dos itens propostos composta pelas seguintes etapas:

- escolha de instâncias do problema: neste trabalho, foram escolhidas instâncias anteriormente utilizadas em vários trabalhos da literatura relacionada;
- escolha de um conjunto de métricas que possam ser utilizadas para avaliar as duas metas principais em Otimização Multiobjetivo Evolutiva (convergência e diversidade);
- análises e testes de significância estatística dos resultados obtidos; e
- comparação do modelo proposto com outras abordagens utilizadas no Roteamento.

1.3 Contribuições

As investigações realizadas neste trabalho resultaram em algumas contribuições, a saber:

- detecção e correção de um problema referente à geração de indivíduos inválidos quando utilizada a heurística de cruzamento e mutação do modelo original de Ravikumar & Bajpai [45];

- desenvolvimento de três novas heurísticas para serem utilizadas nos operadores de cruzamento e mutação dos MOEAs utilizados como mecanismos de busca subjacente;
- incorporação de duas variações do conceito de Cruzamento pela Vizinhaça nos modelos de MOEA pesquisados;
- incorporação de uma métrica no contexto de Engenharia de Tráfego, mais especificamente relacionada ao uso dos enlaces ($\bar{\alpha}$); e
- desenvolvimento de dois novos filtros para promover o aumento de diversidade populacional.

Os principais resultados obtidos durante este trabalho foram publicados em [6, 8–12].

1.4 Organização do Texto

Os demais capítulos deste trabalho estão assim organizados:

- Capítulo 2: descreve os Algoritmos Genéticos clássicos e multiobjetivos, junto com as definições de otimalidade de Pareto;
- Capítulo 3: introduz e formaliza o Problema do Roteamento Multicast sob as perspectivas mono e multiobjetivo. Uma revisão da literatura relacionada é apresentada;
- Capítulo 4: apresenta os operadores genéticos com base em modelos anteriores de roteamento evolutivo. Mostra-se como o problema de soluções inválidas pode ocorrer e algumas estatísticas para quantificação das suas ocorrências são analisadas;
- Capítulo 5: apresenta os elementos propostos, como as novas heurísticas, filtros e explica como o conceito de Cruzamento pela Vizinhaça foi incorporado no modelo evolutivo. Uma discussão sobre a escolha das métricas é apresentada;
- Capítulo 6: apresentação e discussão dos resultados experimentais, utilizando métricas para avaliação de convergência e diversidade dos métodos propostos. Alguns testes de significância estatística são realizados, além de uma comparação dos resultados com algoritmos comumente usados no roteamento; e
- Capítulo 7: considerações finais e sugestões para trabalhos futuros.

Capítulo 2

Otimização Evolutiva Multiobjetivo

2.1 Algoritmos Genéticos

Algoritmos Genéticos (AGs) são considerados um dos principais métodos metaheurísticos que compõem o campo da Computação Evolutiva, sendo caracterizados pela inspiração na biologia evolutiva, por meio da simulação de mecanismos como a seleção natural e herança genética [25]. Uma das principais motivações para o uso de AGs é na resolução de muitos problemas clássicos em Inteligência Artificial, como reconhecimento de padrões e extração de regras, além de problemas NP-difíceis em otimização combinatória (p.ex., Problema do Caixeiro Viajante [14], Problema da Árvore de Steiner [31], Problema do Menor Caminho Multicritério [36], [42]). Apesar de os AGs não garantirem que a solução ótima seja encontrada na maioria dos casos, freqüentemente eles oferecem uma interessante relação entre a qualidade da solução oferecida e o esforço computacional necessário.

AGs são métodos de busca estocásticos multiponto, os quais iniciando a partir de um conjunto de soluções geradas aleatoriamente, aplicam uma série de operadores genéticos para produzir novas soluções de uma maneira iterativa. O papel de tais operadores é guiar a geração de novas soluções para regiões mais promissoras do espaço de busca. Para isso, os AGs usam uma métrica de aptidão (*fitness*), a qual está relacionada com as funções objetivo a serem otimizadas no problema estudado. Na maioria dos casos, os operadores genéticos não manipulam as soluções diretamente, mas sim, uma representação codificada de uma potencial solução para um dado problema, recebendo o nome de cromossomo ou indivíduo.

Como resultado, pedaços de duas soluções são misturados para gerar uma ou mais soluções novas, o que é geralmente feito pelo operador de cruzamento. De um modo geral, dentre os indivíduos existentes no ambiente em um dado momento, aqueles mais aptos têm tendência a sobreviver e passar seu material genético adiante. Assim, espera-se a melhoria da população de soluções ao longo das iterações (gerações, no vocabulário do AG) por meio da aplicação dos operadores genéticos. Assim, diz-se que a principal

metáfora dos AGs é a de que indivíduos mais bem adaptados em seu ambiente tendem a sobreviver e se reproduzir, na esperança de que sua prole seja mais bem adaptada que seus pais.

O fluxo principal de um AG está apresentado no Alg. 1. Como pode ser visto, após a geração e a avaliação da população inicial, o processo evolutivo é iterado um número de vezes até que um critério de término seja atingido. Um critério comum é estabelecer um número máximo de gerações. Em qualquer geração, o operador de seleção para cruzamento escolhe N_c pares de pais a serem cruzados. A aplicação do cruzamento em um par de pais retorna um ou mais indivíduos novos. A ocorrência do cruzamento em cada par é condicionada a uma probabilidade (75%, p.ex.), o que determina se ocorrerá efetivamente um cruzamento em tal par. Para prover mais diversidade para cada filho gerado, o operador de mutação é aplicado para alterar uma ou mais de suas partes. O operador de mutação também é condicionado a uma probabilidade, a qual é geralmente menor que a do cruzamento (1%, p.ex.). Sempre que um novo indivíduo é gerado, ele precisa ser avaliado, i.e., um valor de aptidão deve ser-lhe atribuído.

Após a geração dos indivíduos filhos, o AG precisa determinar quais entre pais e filhos sobreviverão para a próxima geração. Isto é feito pelo operador de reinserção, o qual também pode ter um forte impacto na convergência do AG para soluções ótimas. Uma das principais formas de reinserção em contextos Multiobjetivo é o elitismo, cuja idéia principal é que apenas os melhores indivíduos de toda a população de pais e filhos (ou um subconjunto, tal como a população de pais apenas) sobreviverão. Assim, cada iteração do AG compreendendo essas etapas (seleção para cruzamento, cruzamento, mutação e reinserção) é chamada de geração. A seguir, o conceito principal de tais operadores é brevemente descrito.

Algoritmo 1 Esquema geral de um Algoritmo Genético

- 1: Gere uma população inicial P_0
 - 2: Avalie P_0
 - 3: $t \leftarrow 0$
 - 4: **while** algum critério de término não for satisfeito **do**
 - 5: Crie pares de pais utilizando algum critério de seleção
 - 6: Aplique os operadores de cruzamento e mutação sobre os pares formados, para gerar uma população de filhos Q_t
 - 7: Avalie Q_t
 - 8: Atribua a P_{t+1} os sobreviventes de $P_t \cup Q_t$
 - 9: $t \leftarrow t + 1$
 - 10: **return** o indivíduo mais apto de P_t
-

2.1.1 Representação de Indivíduos e População Inicial

A codificação binária é uma das principais utilizadas para representar indivíduos em AGs, onde um vetor representa uma solução potencial para um problema em estudo.

Cada string é mapeada para uma solução no problema, tal como um valor nos reais ou uma seqüência de vértices no Problema do Caixeiro Viajante. Entretanto, é importante mencionar que dependendo do tipo de problema em que o AG é aplicado, outros tipos de codificações de indivíduo podem substituir a codificação binária. Em vários problemas, representações baseadas em números reais, permutações de inteiros, árvores e outros são utilizadas.

Em um exemplo, onde se deseja maximizar $f(x) = x^2$ no intervalo $[0, 31]$, cada indivíduo pode ser codificado como uma string com cinco bits sem sinal. Assim, uma string como 01101 será mapeada para $x = 13$, enquanto 10011 para $x = 19$. Esta operação é também conhecida como mapeamento do genótipo (uma codificação para soluções, adotada no AG) para o fenótipo (uma solução potencial para o problema).

Usualmente, os indivíduos que compõem a população inicial são gerados de forma aleatória, apesar de ser possível a utilização de estratégias mais deterministas como uma tentativa de acelerar a convergência. Em muitos problemas, dependendo do formato da solução, é também importante que os indivíduos gerados sejam válidos. Por exemplo, num problema que trabalha com indivíduos representando árvores geradas a partir de grafos, é necessário que cada indivíduo represente uma árvore válida (não possuir laços, por exemplo).

2.1.2 Cálculo da Aptidão

A métrica de aptidão corresponde a um valor atribuído a cada indivíduo, refletindo uma medida aproximada ou exata da sua “qualidade” no contexto de algum problema. Assim, este valor pode ser usado para comparar indivíduos, com o objetivo de determinar quais deles são os mais aptos em um dado momento da execução de um AG. Em muitos casos (incluindo as formulações do Roteamento utilizadas aqui), a aptidão está fortemente relacionada com as funções objetivo. Em geral, o cálculo de aptidão é bastante importante no contexto dos AGs, pois possui um impacto substancial para guiar a direção da busca evolutiva. Desde modo, é importante considerar como projetar a função de aptidão para guiar o AG para regiões mais promissoras durante a evolução.

No exemplo da seção anterior, para cada x corresponderá um $f(x)$, o qual neste caso será também o seu valor de aptidão, i.e., a aptidão do indivíduo 01101 será $f(13) = 169$, enquanto que 10011 resultará em $f(19) = 361$. Isto mostra que o último indivíduo é mais adaptado que o primeiro, considerando que a aptidão deve ser maximizada neste caso. Por outro lado, em outros problemas o uso direto da função objetivo como aptidão pode não ser adequado [25]. A aptidão também é utilizada para definir probabilidades em alguns métodos de seleção, de modo que indivíduos mais aptos terão maiores probabilidades de participar no cruzamento para gerar filhos que herdarão partes do seu material genético.

2.1.3 Seleção para Cruzamento

O operador de seleção para cruzamento é responsável por criar pares de pais que poderão ser cruzados na geração corrente. A principal questão nesse operador é como combinar aleatoriedade com determinismo, em termos de valores de aptidão, para selecionar pares de pais. Quando o determinismo é enfatizado, indivíduos mais aptos tendem a ter alta probabilidade de serem selecionados. Por outro lado, quando a seleção é muito aleatória, todos os indivíduos tendem a possuir a mesma chance de serem selecionados. Este critério é também conhecido como pressão seletiva e impacta na convergência do AG. Uma pressão seletiva maior pode acelerar a convergência, correndo-se o risco de levar a população à convergência prematura, i.e., quando a população é composta de muitas cópias de uma ou mais soluções sub-ótimas, congelando a evolução do AG. Por outro lado, uma pressão seletiva menor possivelmente resultará em uma convergência fraca ou lenta, requerendo mais recursos computacionais. Dois dos principais operadores para seleção são a roleta e o torneio [25].

Para exemplificação dos métodos de seleção a serem descritos, consideremos uma população P formada por cinco indivíduos, ilustrados na Tab. 2.1.

Tabela 2.1: Exemplo de uma população com cinco indivíduos, seus valores de aptidão e de probabilidades p_i .

Indivíduo	String	x	Aptidão (f_i)	p_i
1	01101	13	161	0.15
2	10011	19	361	0.33
3	10101	21	441	0.41
4	00110	6	36	0.03
5	01001	9	81	0.08
			$\sum f_i = 1088$	$\sum p_i = 1.00$

A roleta é um método de seleção em que cada indivíduo recebe “casas” numa roleta, proporcionalmente à sua aptidão. No caso da maximização de $f(x) = x^2$, quanto maior a aptidão, mais casas serão atribuídas. Subjacente a essa idéia está uma associação de probabilidades aos indivíduos, definida como

$$p_i = \frac{f_i}{\sum_{j=1}^{|P|} f_j}$$

onde p_i é a probabilidade do indivíduo i ser selecionado. A Fig. 2.1 ilustra a roleta formada de acordo com a população de indivíduos mostrada na Tab. 2.1. Para selecionar um indivíduo, sorteia-se um número entre 0 e 1, e verifica-se em que “região” da roleta tal número estará. Deste modo, para criar um par de indivíduos a serem cruzados são efetuados dois “giros” da roleta.

Por sua vez, no torneio simples primeiramente cria-se um grupo (*tour*) com dois ou

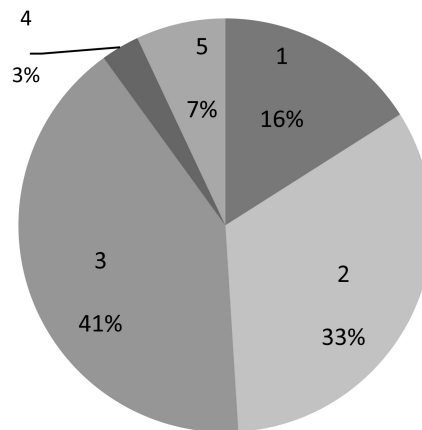


Figura 2.1: Representação das probabilidades associadas a cada indivíduo no método de seleção pela roleta.

mais indivíduos, onde cada indivíduo da população tem a mesma probabilidade de entrar no *tour*. Em seguida, o melhor indivíduo dentre aqueles do *tour* é escolhido como o vencedor do torneio. Portanto, para selecionar um par de pais para cruzamento, dois torneios são realizados. Um exemplo de torneio simples com *tour* de três indivíduos, baseado na população P , pode ser visto na Tab. 2.2.

Tabela 2.2: Exemplo de torneio simples.

<i>Tour</i>	Vencedor
2, 5, 1	2
4, 1, 3	3

2.1.4 Cruzamento

Cruzamento é o operador responsável por recombinar partes de dois pais para produzir um ou mais novos indivíduos (indivíduos filhos), simulando o processo de troca de material genético que ocorre na natureza. Tais pais foram previamente selecionados através de algum método de seleção para cruzamento (torneio simples, p.ex.). A ocorrência do cruzamento entre os pares de pais selecionados na etapa anterior é condicionada a uma probabilidade (75%, p.ex.), a qual é usada para determinar se cada par será efetivamente cruzado ou não. O papel do cruzamento é fundamental nos AGs, pois junto da mutação é responsável por fazer soluções aleatórias convergirem para soluções ótimas ou sub-ótimas ao longo do processo evolutivo. Assim, a concepção deste operador é de grande importância durante a modelagem de um problema utilizando AGs. Por exemplo, quando indivíduos são codificados como strings binárias, alguns esquemas de cruzamento são comumente usados, tais como cruzamento de um ponto e multiponto, como mostrado nas figuras 2.2 and 2.3.

Como visto, nem sempre a representação binária é utilizada, sendo possível a utilização de operadores de cruzamento diferenciados. Apesar de em muitos problemas ser possível

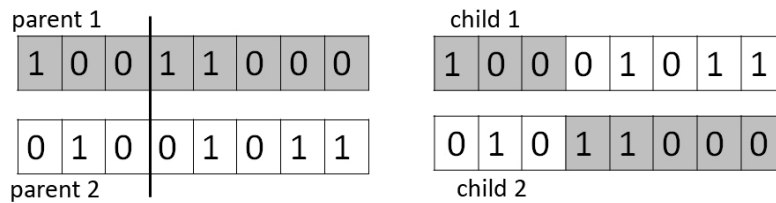


Figura 2.2: Cruzamento de um ponto.

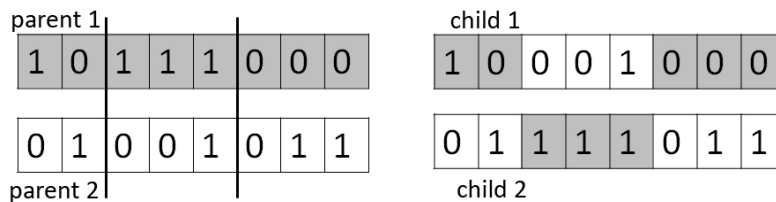


Figura 2.3: Cruzamento multiponto.

manter a representação binária, pode acontecer de uma representação não-binária ser mais intuitiva ou rápida para ser utilizada.

2.1.5 Mutação

O operador de mutação tem o objetivo de aumentar a diversidade genética da população sendo gerada, alterando aleatoriamente uma ou mais partes (genes) dos indivíduos gerados pelo cruzamento. Este operador pode ajudar o AG a explorar mais eficientemente a busca juntamente com os operadores de cruzamento e seleção. A ocorrência da mutação está condicionada uma probabilidade geralmente baixa (1%, p.ex.), podendo ser dada por indivíduo ou por gene. Em termos gerais, o funcionamento do operador de mutação depende de qual estrutura de dados é usada para representar os indivíduos do AG. Um exemplo (troca de bit) é mostrado na Figura 2.4.

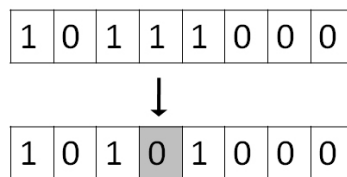


Figura 2.4: Aplicação da mutação sobre um indivíduo por meio da troca de bit.

2.1.6 Sobrevivência de Indivíduos (Reinserção)

Após a geração de filhos, o AG precisa decidir quais indivíduos entre pais e filhos sobreviverão para a próxima geração. Este processo é um tipo de seleção também conhecido como reinserção. Usualmente, os indivíduos mais adaptados tendem a sobreviver. Vários tipos de reinserção foram propostos na literatura, dentre eles:

- *Reinserção uniforme*: um procedimento de seleção (roleta, torneio, etc.) é aplicado para escolher os sobreviventes entre pais e filhos.
- *Elitismo*: um percentual dos melhores pais é mantido para a próxima geração. O restante da população é formado pelos filhos gerados no cruzamento.
- *Baseada na aptidão*: sobrevivem os melhores indivíduos na população formada pela união entre pais e filhos.

Alguns autores consideram que a reinserção baseada na aptidão é também um tipo de elitismo, como pode ser observado no método multiobjetivo NSGA-II, que será apresentado na Sec. 2.3.1.

2.2 Problemas de Otimização Multiobjetivo

Um Problema de Otimização Multiobjetivo (*Multiobjective Optimization Problem* ou MOP) é caracterizado pela otimização (minimização ou maximização) de duas ou mais funções objetivo, de modo que soluções viáveis devem satisfazer J restrições de desigualdade, K restrições de igualdade e $2n$ restrições laterais. A formulação geral para um MOP é dada por:

$$\begin{aligned} &\text{maximizar/minimizar} && f_m(x), && m = 1, 2, \dots, M \\ &\text{sujeito a} && g_j(x) \leq 0, && j = 1, 2, \dots, J \\ & && h_k(x) = 0, && k = 1, 2, \dots, K \\ & && x_i^L \leq x_i \leq x_i^U, && i = 1, 2, \dots, n \end{aligned}$$

onde x é um vetor de decisão dado por $x = (x_1, x_2, \dots, x_n)$ em um espaço de decisão n -dimensional. Cada x é mapeado para um vetor $f(x) = (f_1(x), \dots, f_m(x))$ no espaço M -dimensional de objetivos; por exemplo, em um MOP com x e $f(x)$ como vetores de números reais, haveria um mapeamento dado por $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^M$.

Em um problema sujeito à otimização de múltiplos objetivos, pode não ser possível otimizar todos os objetivos ao mesmo tempo, pois ao otimizar um objetivo, é possível que outros sejam deteriorados. Quando este é o caso, pode não existir uma solução que possua, simultaneamente, valores ótimos para todos os objetivos no MOP em questão. Devido a esse conflito, define-se a seguinte relação [47]:

Definição 1. Dominância de Pareto. *Sejam a e b dois vetores num espaço n -dimensional de decisão. Sejam $f(a) = \{f_1(a), \dots, f_m(a)\}$ e $f(b) = \{f_1(b), \dots, f_m(b)\}$ seus respectivos vetores num espaço M -dimensional de objetivos. Diz-se que a domina b , representado por $a \prec b$, sse:*

- a não é pior que b em todos os objetivos; e

- a é estritamente melhor que b em pelo menos um objetivo.

A relação de dominância de Pareto é uma relação de ordem parcial estrita, logo, as seguintes propriedades são satisfeitas [47]:

- Se $a \prec b$, então $a \neq b$ (*irreflexiva*)
- Se $a \prec b$, então $\neg b \prec a$ (*assimétrica*)
- Se $a \prec b \wedge b \prec c$, então $a \prec c$ (*transitiva*)

Uma vez que estamos lidando com conjuntos parcialmente ordenados, pode não ser possível estabelecer um ordenamento total dos vetores de decisão, a partir da avaliação dos objetivos considerados. Isto significa que, dados dois vetores a e b quaisquer, pode não ocorrer $a \prec b$ nem $b \prec a$, o que resulta na seguinte definição:

Definição 2. Conjunto não-dominado. *Seja C um conjunto de soluções. Define-se $C' \subseteq C$ como o subconjunto composto por todas as soluções não-dominadas de C , i.e., soluções em C para as quais não há outra solução em C que as domine.*

Se C for todo o espaço de decisão, C' conterá soluções não-dominadas em um sentido global. A próxima definição formaliza esta idéia.

Definição 3. Conjunto Pareto-ótimo global. *Seja P um espaço de decisão viável. Então, o conjunto não-dominado $P^* \subseteq P$ é chamado de conjunto Pareto-ótimo global.*

Conseqüentemente, toda solução x^* em P^* é chamada de solução Pareto-ótima global. Deste modo, a definição de solução ótima passa a ser conectada com a relação de dominância, permitindo a existência de múltiplas soluções eficientes nessa perspectiva.

Um exemplo de relação de dominância em um problema com uma variável de decisão ($-5.12 \leq x \leq 5.12$) e dois objetivos ($f_1 = \sin(x) + 1.1$ e $f_2 = x/10 + 1$) a serem maximizados pode ser visto na Figura 2.5. Para cada x , corresponderá um mapeamento para $f(x) = (f_1(x), f_2(x))$, tal que as soluções Pareto-ótimas são os valores de x para os quais seus objetivos $(f_1(x), f_2(x))$ são o melhor compromisso no espaço dos objetivos. Pode-se observar isso pelos segmentos em negrito em cada curva de f_1 e f_2 . No exemplo, as soluções Pareto-ótimas correspondem ao intervalo de pontos $x \in [1.57, 4.30]$ e ao ponto $x = 5.12$. O intervalo de pontos $x \in (4.30, 5.12)$ não pertence à região Pareto-ótima pois todos estes pontos são dominados por $x = 5.12$.

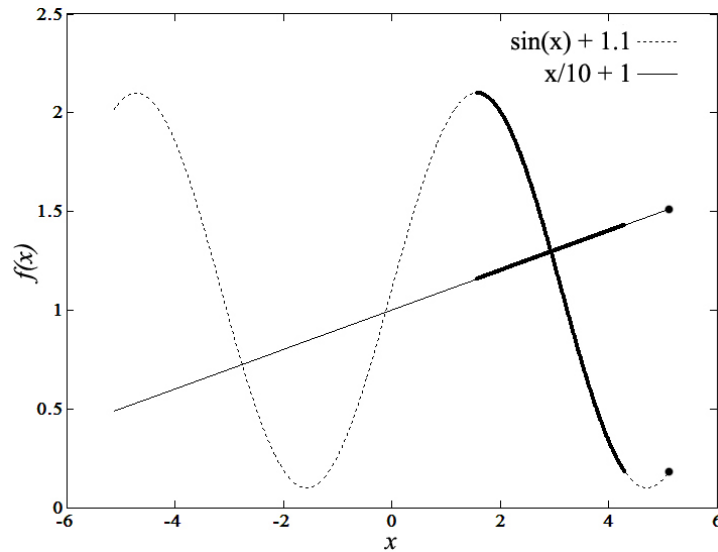


Figura 2.5: Soluções Pareto-ótimas em um problema de maximização de dois objetivos.

2.3 Algoritmos Evolutivos Multiobjetivo

Assim como os AGs, os Algoritmos Evolutivos Multiobjetivo (*Multiobjective Evolutionary Algorithms* ou MOEAs) são métodos metaheurísticos inspirados na biologia evolutiva, entretanto, são projetados especificamente para abordar problemas que incluam mais de uma função objetivo a ser otimizada. A principal diferença entre um AG clássico e um MOEA usualmente ocorre no cálculo de aptidão, onde a dominância de Pareto é usada de diferentes formas. Este trabalho lida com três algoritmos da segunda geração de MOEAs, a qual foi principalmente caracterizada pelo uso do elitismo e, conseqüentemente, vários procedimentos para lidar com soluções elite, sua preservação e como usá-las nos operadores do AG [13]. A seguir, tais MOEAs são descritos.

2.3.1 *Non-dominated Sorting Genetic Algorithm II*

O modelo *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) foi proposto por Deb et al. [18] a fim de prover diversas melhorias em relação à sua versão anterior (NSGA [50]), tais como um algoritmo mais rápido para classificação de fronteiras de não-dominância, um mecanismo mais sofisticado para preservação de diversidade, e a utilização do elitismo. O NSGA-II usa o conceito de *dominance depth* [61], em que a população corrente P é classificada em diversas fronteiras de não-dominância, de maneira que na fronteira F_1 estarão apenas os indivíduos não-dominados de P , e nas fronteiras seguintes (F_2 , F_3 , etc.) estarão soluções não-dominadas entre si, mas que sejam dominadas por pelo menos uma solução da fronteira anterior. A aptidão de um indivíduo é dada por dois valores: a sua fronteira (também chamada de *rank*) e uma estimativa de densidade (*crowding distance*).

Para atribuir a cada indivíduo o seu *rank*, no início de qualquer geração os indivíduos

da população P deverão estar classificados de acordo com fronteiras não-dominadas, como descrito. O segundo atributo (*crowding distance*) serve para estimar uma densidade do cubóide em que se encontra cada indivíduo de cada F_i . Por esta métrica, quanto mais próximo for um indivíduo em relação aos seus vizinhos imediatos de esquerda e direita (para uma dada função objetivo), menor será sua *crowding distance*, e vice-versa. Um exemplo é dado na Figura 2.6. Em tal métrica, quanto maior for a distância, melhor será considerado um indivíduo, pois ele ajuda a manter a diversidade da população.

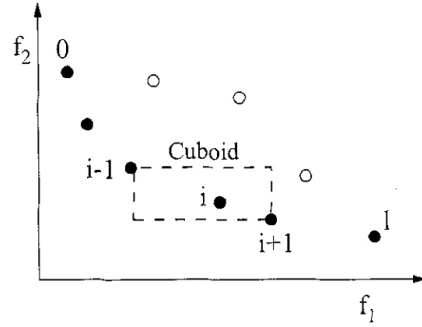


Figura 2.6: O cubóide para o i -ésimo indivíduo em sua fronteira não-dominada (círculos cheios são soluções da mesma fronteira não-dominada) [18].

O procedimento de cálculo de *crowding distance* para cada frente I de soluções não-dominadas é mostrado no Alg. 2.

Algoritmo 2 Cálculo de *crowding distance* no NSGA-II (I)

Entrada: I : um conjunto de soluções não-dominadas.

```

1:  $l \leftarrow |I|$ 
2: for cada  $i$  do
3:    $I[i]_{dist} \leftarrow 0$ 
4: for cada objetivo  $m$  do
5:    $I \leftarrow \text{ordene}(I, m)$ 
6:    $I[1]_{dist} = I[l]_{dist} = \infty$ 
7:   for  $i \leftarrow 2$  até  $(l - 1)$  do
8:      $I[i]_{dist} \leftarrow I[i]_{dist} + (I[i + 1].m - I[i - 1].m) / (f_m^{max} - f_m^{min})$ 

```

Após estes dois cálculos para toda a população, cada indivíduo na população terá os valores de *rank* e *crowding distance*, os quais são utilizados no operador de seleção (torneio binário). As comparações entre indivíduos utilizando tais valores ao longo do algoritmo dependem, então, do operador $\prec_n$ definido pelo NSGA-II:

$$\begin{aligned}
 i \prec_n j: & \ i_{rank} < j_{rank} \text{ ou} \\
 & \ i_{rank} = j_{rank} \wedge i_{dist} > j_{dist}
 \end{aligned} \tag{2.1}$$

onde i_{dist} representa a *crowding distance* para o indivíduo i . Em outras palavras, entre dois indivíduos, prefere-se aquele com menor *rank* ou aquele com maior *crowding distance* caso os *rank*s sejam iguais. Após o cruzamento e a mutação, os melhores indivíduos entre pais e filhos sobrevivem para a próxima geração. O Alg. 3 detalha o NSGA-II.

Algoritmo 3 Non-dominated Sorting Genetic Algorithm II (NSGA-II)

```

1: Gere uma população inicial  $P_0$ 
2:  $t \leftarrow 0$ 
3: while um critério de término não for satisfeito do
4:    $Q_t \leftarrow$  nova pop ( $P_t$ )
5:    $R_t \leftarrow P_t \cup Q_t$ 
6:    $F \leftarrow$  ordenação rápida por não-dominância ( $R_t$ )
7:    $P_{t+1} \leftarrow \emptyset$  e  $i \leftarrow 1$ 
8:   while  $|P_{t+1}| + |F_i| \leq Tp$  do
9:     atribua crowding distance ( $F_i$ )
10:     $P_{t+1} \leftarrow P_{t+1} \cup F_i$ 
11:     $i \leftarrow i + 1$ 
12:   ordene ( $F_i, \prec_n$ )
13:    $P_{t+1} \leftarrow P_{t+1} \cup F_i[1 : Tp - |P_{t+1}|]$ 
14:    $t \leftarrow t + 1$ 

```

A primeira vez que o torneio binário é utilizado para selecionar um pai, não se usa valores de *crowding distance*, uma vez que eles não foram calculados neste momento; portanto, serão utilizados apenas os valores de *rank*. A partir da segunda geração em diante, o torneio binário utiliza o operador $\prec_n$ para escolher os pais [18].

2.3.2 Strength Pareto Evolutionary Algorithm

O modelo *Strength Pareto Evolutionary Algorithm* (SPEA) [64] é um MOEA proposto por Zitzler & Thiele, caracterizado por manter, além da população corrente, um arquivo externo para manter as soluções não-dominadas encontradas até então. O procedimento de cálculo de aptidão difere do NSGA-II, sendo baseado nas idéias de *dominance count* e *dominance rank* [61], as quais referem-se a quantos indivíduos são dominados por um indivíduo, e quantos indivíduos o dominam, respectivamente. Tal procedimento engloba o conceito de usar uma população para avaliar outra.

No começo de cada geração, atualiza-se o arquivo E (inicialmente vazio) a partir da população corrente P e do próprio arquivo E . Assim, são mantidas em E apenas as soluções não-dominadas entre $P \cup E$. Caso o tamanho de E ultrapasse um limite previamente estabelecido, um procedimento de clusterização é aplicado (Alg. 4). Em seguida, o cálculo de aptidão é realizado em duas etapas. Primeiro, calcula-se um valor de *strength* s para cada indivíduo $i \in E$, que é dado pelo número de indivíduos $j \in P$,

tais que $i \prec j$, i.e.,

$$s(i) = \frac{|\{j : j \in P \wedge i \prec j\}|}{|P| + 1} \quad (2.2)$$

A partir desta definição, a *strength* de qualquer membro do arquivo pertencerá ao intervalo $[0, 1)$. o SPEA define que o valor de aptidão f para cada membro de E será sua própria *strength*, i.e., $f(i) = s(i), \forall i \in E$.

Em seguida, procede-se para a avaliação dos membros de P . Para cada indivíduo $j \in P$, sua aptidão será a soma das *strengths* de seus dominadores, i.e.,

$$f(j) = 1 + \sum s(i), \forall i \in E : i \prec j \quad (2.3)$$

O resultado será um valor $f(j) \geq 1, \forall j \in P$. Um exemplo pode ser visto na Fig. 2.7.

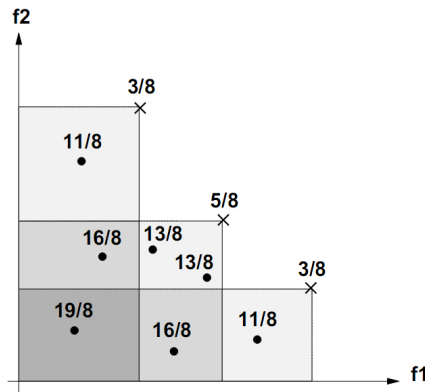


Figura 2.7: Um exemplo do cálculo de aptidão para membros da população e do arquivo, em problema de maximização de dois objetivos. Os membros da população estão representados por círculos cheios, ao passo que os membros do arquivo estão marcados com o símbolo $\times$ [64].

Sempre que E exceder um tamanho máximo estabelecido L , o SPEA aplica um procedimento de clusterização para reduzir suas dimensões. Tal procedimento foi proposto para manter as características de E [64], i.e., manter um reduzido número de soluções não-dominadas que poderiam representar o conjunto ampliado, de uma certa forma. Assim, inicialmente o procedimento de clusterização define que para cada indivíduo em E , um *cluster* unitário deve ser criado. Em seguida, *clusters* próximos começam a ser unidos, até que o número de *clusters* seja reduzido para L . Neste ponto, em cada *cluster*, deve-se escolher um de seus elementos para representá-lo; tal ponto corresponde ao centróide do *cluster*, i.e., o elemento que possuir a menor distância para os demais elementos do *cluster*. Quando o centróide de cada *cluster* tiver sido calculado, os demais elementos de cada *cluster* são removidos, resultando em L *clusters* unitários. Por fim, cada elemento destes *clusters* corresponderão aos indivíduos do arquivo E reduzido. O Alg. 4 exhibe este processo.

Após a atualização do arquivo e cálculo de aptidão, verifica-se se algum critério de

Algoritmo 4 Procedimento de clusterização no SPEA (E)

Entrada: E : conjunto de soluções não-dominadas, cujo tamanho será reduzido para L .

Saída: E : conjunto reduzido de soluções não-dominadas.

```

1:  $clusterSet \leftarrow \emptyset$ 
2: for  $i \leftarrow 1$  até  $|E|$  do
3:    $clusterSet \leftarrow clusterSet \cup \{E[i]\}$ 
4: while  $|clusterSet| > L$  do
5:    $mindist \leftarrow \infty$ 
6:   for cada par de clusters  $\{x, y\}$  de  $clusterSet$  do
7:      $curdist \leftarrow clusterDistance(x, y)$ 
8:     if  $curdist < mindist$  then
9:        $cluster1 \leftarrow x$ 
10:       $cluster2 \leftarrow y$ 
11:       $mindist \leftarrow curdist$ 
12:    $newcluster \leftarrow cluster1 \cup cluster2$ 
13:    $clusterSet \leftarrow clusterSet \setminus \{cluster1, cluster2\}$ 
14:    $clusterSet \leftarrow clusterSet \cup \{newcluster\}$ 
15:  $E \leftarrow \emptyset$ 
16: for cada  $c_i$  de  $clusterSet$  do
17:    $representative \leftarrow getCentroid(c_i)$ 
18:    $E \leftarrow E \cup \{representative\}$ 
19: return  $E$ 

```

término foi atingido; em caso positivo, o arquivo é retornado e a execução é finalizada. Caso contrário, pares de indivíduos para cruzamento são criados a partir de P e E , considerando minimização da aptidão. Então, o cruzamento e a mutação são aplicados para gerar uma população de filhos, a qual substitui a população corrente. Todo este processo é repetido por um número de gerações, como mostrado no Alg. 5.

2.3.3 *Strength Pareto Evolutionary Algorithm 2*

O modelo *Strength Pareto Evolutionary Algorithm 2* (SPEA2) [62] surgiu para melhorar seu predecessor (SPEA) principalmente em três aspectos. Primeiramente, provendo uma atribuição mais completa de aptidão, na qual a informação de dominância entre membros da população P passa a ser considerada, uma vez que no SPEA a aptidão de membros da população depende apenas da *strength* de seus dominadores, o que pode levar possivelmente a uma redução da pressão seletiva. Em segundo lugar, para cada indivíduo, uma estimativa de densidade é usada para guiar a busca mais eficientemente, especialmente quando há muitos indivíduos não-dominados na geração corrente, podendo ser usada para diferenciá-los. Finalmente, outro procedimento para reduzir a cardinalidade de E é usado, onde soluções extremas são preservadas.

A seguir são descritos os passos para cálculo da aptidão no SPEA2. Inicialmente, um

Algoritmo 5 Esquema geral do SPEA

Entrada: Tp : tamanho da população. L : tamanho máximo do arquivo.

Saída: E : um conjunto representativo de soluções não-dominadas encontradas durante a busca evolutiva.

```

1: Gere uma população inicial  $P_0$ 
2:  $E_0 \leftarrow \emptyset$ 
3:  $t \leftarrow 0$ 
4: while true do
5:   Calcule o valor de aptidão para os indivíduos de  $P_t \cup E_t$  (Eq. 2.3)
6:   Atribua a  $E_{t+1}$  os indivíduos não-dominados de  $P_t \cup E_t$ 
7:   if  $|E_{t+1}| > L$  then
8:     Reduza  $E_{t+1}$  por meio do procedimento de clusterização (Alg. 4)
9:   if um critério de término é atingido then
10:    return  $E_{t+1}$ 
11:   Faça uma seleção por torneio binário sobre  $E_{t+1} \cup P_t$  para selecionar pares para o
   cruzamento
12:   Aplique os operadores de cruzamento e de mutação nos pares a fim de gerar uma
   população de filhos  $P_{t+1}$ 
13:    $t \leftarrow t + 1$ 

```

valor de *strength* é computado para cada indivíduo de $P \cup E$. A *strength* de um indivíduo i corresponde ao número de indivíduos em tal conjunto que são dominados por i , ou seja,

$$s(i) = |\{j : j \in P \cup E \wedge i \prec j\}| \quad (2.4)$$

Em seguida, a cada indivíduo é atribuído um valor de *raw fitness*, definido como a soma das *strengths* de seus dominadores:

$$r(i) = \sum s(j), \forall j \in P \cup E : j \prec i \quad (2.5)$$

Pode-se notar que indivíduos não-dominados terão $r(i) = 0$.

Finalmente, uma estimativa de densidade $d(i)$ baseada no método do k -ésimo vizinho mais próximo é definida:

$$d(i) = \frac{1}{\sigma_i^k + 2} \quad (2.6)$$

onde σ_i^k é a k -ésima menor distância de i aos seus indivíduos, sendo $k = \sqrt{|P| + |E|}$. O valor dois é adicionado ao denominador para garantir que $d(i)$ estará no intervalo $(0, 1)$.

Finalmente, o valor de aptidão é definido como:

$$f(i) = r(i) + d(i) \quad (2.7)$$

Isto garantirá que indivíduos não-dominados sempre terão $f(i) < 1$, ao passo que indivíduos dominados terão $f(i) \geq 1$. Um exemplo de cálculo de aptidão pode ser visto

na Fig. 2.8.

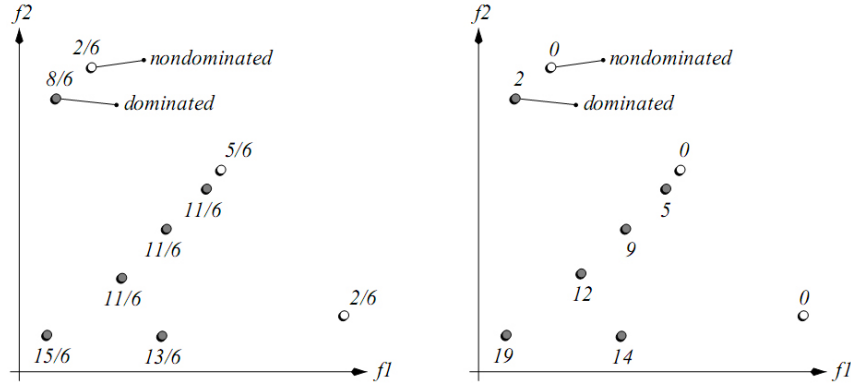


Figura 2.8: Um exemplo comparativo, em um problema de maximização de dois objetivos, entre o SPEA (lado esquerdo) e o SPEA2 (lado direito). Os indivíduos dominados estão marcados com círculos cheios, ao passo que os não-dominados estão como círculos vazios. Os valores para os indivíduos indicam a aptidão (caso SPEA) e o *raw fitness* (caso SPEA2) [62].

Após a computação de aptidão para cada membro de P_t e E_t , em E_{t+1} serão atribuídos os indivíduos não-dominados destes dois conjuntos. Diferentemente de seu predecessor, no SPEA2 o tamanho de E sempre possui um tamanho fixo L . Assim, após a atualização do arquivo, três situações podem ocorrer:

- (i) $|E_{t+1}| = L$: nada a fazer.
- (ii) $|E_{t+1}| < L$: os $L - |E_{t+1}|$ indivíduos dominados mais aptos de $E_t \cup P_t$ são copiados para E_{t+1} .
- (iii) $|E_{t+1}| > L$: um procedimento de truncamento é aplicado sobre E_{t+1} para reduzir a sua cardinalidade para L .

Como pode ser visto, é possível que existam indivíduos dominados em E , diferentemente do SPEA.

No procedimento de truncamento de arquivo proposto pelo SPEA2, no começo de cada iteração um indivíduo i é selecionado para ser removido, tal que $i \leq_d j, \forall j \in E_{t+1}$, onde

$$i \leq_d j \text{ sse: } \forall 0 < k < |E_{t+1}| : \sigma_i^k = \sigma_j^k \vee \exists 0 < k < |E_{t+1}| : [(\forall 0 < l < k : \sigma_i^l = \sigma_j^l) \wedge \sigma_i^k < \sigma_j^k] \quad (2.8)$$

em que σ_i^k denota a distância do indivíduo i ao seu k -ésimo vizinho mais próximo em E_{t+1} . Em outras palavras, após encontrar o par (a, b) de indivíduos cuja distância euclidiana $\text{dist}(a, b)$ no espaço de objetivos seja a menor dentre todos os pares de indivíduos de E_{t+1} , o procedimento removerá a ou b dependendo de qual dos dois indivíduos tiver a segunda

menor distância (observe que a primeira menor distância é a própria $dist(a, b)$). Se as segundas menores distâncias forem iguais, as terceiras menores distâncias serão verificadas e assim por diante, até que uma das condições da Eq. 2.8 seja satisfeita. Este processo mantém as soluções extremas. Indivíduos serão removidos até que se obtenha $|E_{t+1}| = L$.

Assim como no SPEA, após a avaliação dos indivíduos e atualização de E , verifica-se se algum critério de término é atingido; caso positivo, os indivíduos não-dominados do arquivo são retornados e a execução é encerrada. Caso contrário, pares para cruzamento são formados utilizando somente E , diferentemente do SPEA. Em seguida, os operadores de cruzamento e mutação são aplicados para gerar a população de filhos, a qual substitui a população corrente. Todo este processo é repetido por um número de gerações, como mostrado pelo Alg. 6.

Algoritmo 6 Esquema geral do SPEA2

Entrada: Tp : tamanho da população. L : tamanho máximo do arquivo.

Saída: E : um conjunto representativo de soluções não-dominadas encontradas durante a busca evolutiva.

- 1: Gere uma população inicial P_0
 - 2: $E_0 \leftarrow \emptyset$
 - 3: $t \leftarrow 0$
 - 4: **while** true **do**
 - 5: Calcule o valor de aptidão para cada indivíduo de $P_t \cup E_t$ (Eq. 2.7)
 - 6: Atribua a E_{t+1} os indivíduos não-dominados de $P_t \cup E_t$
 - 7: **if** $|E_{t+1}| < L$ **then**
 - 8: Preencha E_{t+1} com os $L - |E_{t+1}|$ indivíduos dominados mais aptos de $P_t \cup E_t$
 - 9: **else if** $|E_{t+1}| > L$ **then**
 - 10: Reduza E_{t+1} utilizando o procedimento de truncamento (Eq. 2.8)
 - 11: **if** algum critério de término for atingido **then**
 - 12: **return** os indivíduos não-dominados de E_{t+1}
 - 13: Faça uma seleção por torneio binário sobre E_{t+1} para criar pares de pais para cruzamento
 - 14: Aplique os operadores de cruzamento e mutação sobre os pares de pais selecionados para gerar uma população de filhos P_{t+1}
 - 15: $t \leftarrow t + 1$
-

Capítulo 3

Problema do Roteamento Multicast

Um importante componente em protocolos de roteamento são os algoritmos para cálculo de rotas, os quais usualmente consideram um ou mais requisitos que afetam a qualidade destas, com base em uma especificação do usuário ou do serviço. Protocolos para multicast, como MOSPF [37] e DVMRP [55], utilizam algoritmos eficientes tais como aqueles propostos por Dalal & Metcalfe [17] e Dijkstra [14] para calcular rotas que podem garantir a otimalidade de métricas (funções objetivos) fim-a-fim. Porém, tal otimalidade não pode ser garantida quando a métrica é relacionada ao custo da árvore, por exemplo. Neste caso, o problema se tornaria o conhecido Problema da Árvore de Steiner, o qual sabe-se estar na classe NP-difícil [31].

Além disso, tem sido observado que múltiplas métricas simultâneas podem ser necessárias para o cálculo de rotas, como custo, *delay* e *hops count*, as quais estão também relacionadas com a Qualidade de Serviço. Sob a perspectiva da Engenharia de Tráfego, também surgem requisitos relacionados à largura de banda e à otimização da utilização dos enlaces. Assim, diferentes objetivos conflitantes são estabelecidos e o cálculo de rotas pode ser visto como um problema multiobjetivo [22].

Inicialmente, diversos autores trataram alguns objetivos como restrições definidas *a priori*, e técnicas para otimização de um único objetivo foram bastante utilizadas com uma função que combinava vários objetivos, tipicamente através de uma soma ponderada [45], [60], [39], [24], [3]. Por outro lado, se um problema é formulado num contexto multiobjetivo, não se torna necessário estabelecer pesos para os objetivos *a priori*, podendo ser considerados separadamente sob a relação matemática de dominância de Pareto.

Neste capítulo são descritas as formulações mono e multiobjetivo adotadas. Ao final, apresenta-se uma revisão de trabalhos relacionados.

3.1 Formulações para o Problema do Roteamento Multicast

No contexto do Problema do Roteamento Multicast, a topologia de rede pode ser modelada como um grafo direcionado conectado ponderado $G = (V, E)$, onde V representa o conjunto de vértices ou nós (roteadores na rede, p.ex.), e E o conjunto de arestas (enlaces para comunicação entre roteadores, p.ex.). Um nó $\{r\} \subset V$ é chamado de vértice fonte ou raiz. Um conjunto de vértices $D \subset V$ é definido como o grupo multicast, no qual cada vértice é chamado de destino ou terminal. Para cada aresta e em G , definimos os pesos $c(e)$, $d(e)$, $z(e)$ e $t(e)$, correspondendo ao custo, *delay*, capacidade e tráfego corrente de cada aresta, respectivamente. Tais valores estão definidos em $\mathbb{R}$. Convém observar que, para uma dada formulação do PRM, nem todos os pesos podem ser utilizados, como será apresentado ao longo desta seção.

Dois tipos de formulação para o PRM foram adotadas neste trabalho: mono-objetivo, em que há apenas uma função objetivo a ser otimizada sujeito a uma restrição; e multiobjetivo, em que mais de uma função objetivo deve ser otimizada. A abordagem para o caso multiobjetivo segue as idéias de dominância de Pareto, apesar de outras formas serem possíveis, como uma soma ponderada dos objetivos (p.ex., Garrozi & Araujo em [24]). A razão para adotar formulações mono e multiobjetivo é para possibilitar uma avaliação experimental mais detalhada dos métodos propostos no presente trabalho.

3.1.1 Formulação Mono-objetivo

Definição 4. *Formulação do Problema do Roteamento Multicast Mono-objetivo (PRMMo).* Dado um grafo direcionado conectado $G = (V, E)$, um par de pesos $(c(e), d(e))$ para cada $e \in E$ definidos em $\mathbb{R}$, deseja-se calcular uma árvore T de G enraizada em $\{r\} \subset V$ que contenha um conjunto de vértices $D \subset V$, minimizando $\text{custo}(T)$ sujeito a $\text{delay}(t) \leq d_{\max}$ para cada $t \in D$.

Assim, uma instância para o PRMMo é dada pela tupla (G, r, D) . Uma solução ótima para qualquer formulação do PRM tem a propriedade de todos os nós-folha serem destinos. A restrição definida ($d_{\max} \in \mathbb{R}$) é um limite superior para o *delay* fim-a-fim de cada destino. A seguir, define-se a função objetivo e como se pode verificar se uma solução candidata $T = (V_T, E_T)$ atende à restrição:

$$\text{custo}(T) = \sum_{e \in E_T} c(e) \quad (3.1)$$

$$\text{delay}(t) = \sum_{e \in \text{path}(r, t)} d(e), \forall t \in D \quad (3.2)$$

5. Delay fim-a-fim máximo:

$$D_{max}(T) = \max \{delay(t_i)\}, i = 1, \dots, |D| \quad (3.6)$$

6. Hops count:

$$hops(T) = |E_T| \quad (3.7)$$

7. Utilização máxima dos enlaces (Gargalo):

$$\alpha_{max}(T) = \max \left\{ \frac{t(e) + \phi}{z(e)} \right\}, \forall e \in E_T \quad (3.8)$$

8. Utilização média dos enlaces:

$$\bar{\alpha}(T) = \frac{\sum \frac{t(e) + \phi}{z(e)}}{|E_T|}, \forall e \in E_T \quad (3.9)$$

Inicialmente, define-se um conjunto de formulações bi-objetivo sem restrição ou objetivos relacionados com fluxo:

Definição 5. Formulação do Problema do Roteamento Multicast Multiobjetivo (PRMM). Dado um grafo direcionado conectado $G = (V, E)$, um par de pesos $(c(e), d(e))$ para cada $e \in E$ definidos em $\mathbb{R}$, deseja-se calcular o conjunto Pareto-ótimo de árvores T de G enraizadas em $\{r\} \subset V$ que contenham um conjunto de vértices $D \subset V$ em cada um dos seguintes MOPs:

- P1: $\min custo(T)$ e $-D_a(T)$;
- P2: $\min custo(T)$ e $D_t(T)$;
- P3: $\min custo(T)$ e $\bar{D}(T)$;
- P4: $\min custo(T)$ e $D_{max}(T)$;
- P5: $\min custo(T)$ e $hops(T)$.

Deste modo, uma instância para o PRMM será dada por uma tupla (G, r, D) . A Figura 3.1 também mostra uma instância para o PRMM.

Em seguida, definem-se algumas formulações com restrições, a fim de considerar objetivos relacionados à Engenharia de Tráfego:

Definição 6. Formulação do Problema de Roteamento de Fluxo Multicast Multiobjetivo (PRFM). Dado um grafo direcionado conectado $G = (V, E)$, uma tupla de pesos $(c(e), d(e), z(e), t(e))$ definidos em $\mathbb{R}$ para cada $e \in E$ e uma requisição de tráfego $\phi \in \mathbb{R}$ a ser carregada na rede, deseja-se calcular o conjunto Pareto-ótimo de árvores

$T = (V_T, E_T)$ de G , enraizadas em $\{r\} \subset V$ que contenham um conjunto de vértices $D \subset V$. Cada árvore candidata T precisa satisfazer a restrição de capacidade para carregar ϕ em G , dada por: $t(e) + \phi \leq z(e), \forall e \in E_T$ (6). Os seguintes MOPs são definidos:

- P6: $\min \alpha_{\max}(T), \text{custo}(T), D_{\max}(T)$ and $\text{hops}(T)$
s. a Eq. 6.
- P7: $\min \bar{\alpha}(T), \alpha_{\max}(T), \text{custo}(T), D_{\max}(T)$ and $\text{hops}(T)$
s. a Eq. 6.

É imediata a diferença entre as formulações mono-objetivo (Def. 4) e as multiobjetivo (defs. 5 e 6): busca-se uma solução (a solução ótima) no primeiro caso, enquanto que nos outros casos busca-se um conjunto de soluções que fornecem o melhor compromisso no espaço de objetivos, i.e., o conjunto Pareto-ótimo. Tal conjunto de soluções ótimas correspondem às melhores árvores multicast T dentre todas as possíveis derivadas de G , enraizadas em r e que contenham D , de acordo com a relação de dominância de Pareto.

Diferentemente das formulações anteriores (PRMMo e PRMM), uma instância para o PRFM passa a ser dada por uma tupla (G, r, D, ϕ) . Uma instância para o PRFM pode ser vista na Figura 3.2, consistindo em uma rede com 18 vértices, onde o vértice 0 é a fonte e os vértices 6, 10, 13, 15, 17 são destinos. Para cada aresta e , os pesos custo, *delay* e tráfego corrente $(c(e), d(e), t(e))$ estão explicitamente representados. Por outro lado, a capacidade $z(e)$ é considerada fixa para todas as arestas ($z(e) = 12$ Mbps, $\forall e \in E$) e está omitida na figura por questões de visibilidade.

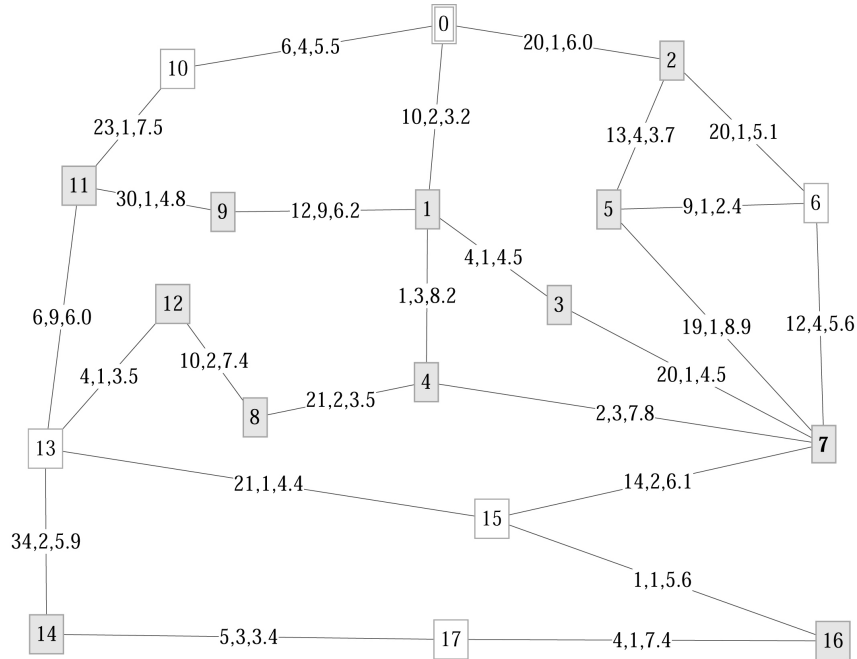


Figura 3.2: Instância do PRFM - rede 1.

Na Fig. 3.3 estão mostradas duas soluções Pareto-ótimas referentes à instância rede 0, na formulação $(\text{custo}, \text{hops})$ do PRMM (P5). Já na Fig. 3.4 são exibidas duas soluções

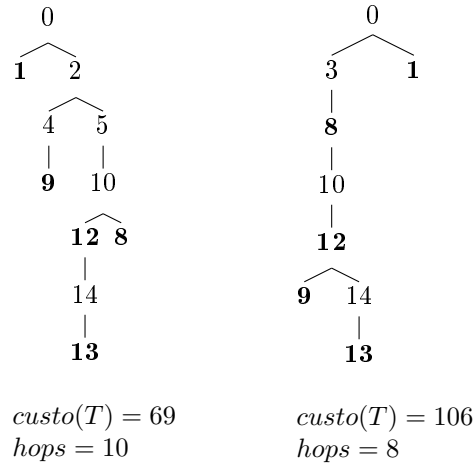


Figura 3.3: Soluções Pareto-ótimas para a instância rede 0 na formulação $(\text{custo}, \text{hops})$ do PRMM (P5).

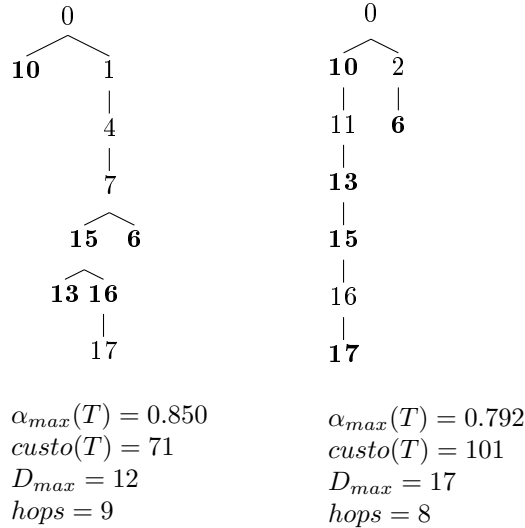


Figura 3.4: Soluções Pareto-ótimas para a instância rede 1 na formulação $(\alpha_{\max}, \text{custo}, D_{\max}, \text{hops})$ do PRFM (P6).

Pareto-ótimas para a instância rede 1, na formulação $(\alpha_{\max}, \text{custo}, D_{\max}, \text{hops})$ do PRFM (P6), considerando que um fluxo $\phi = 2.0$ Mbps deve ser carregado na rede. Em ambos os casos, há outras soluções Pareto-ótimas que não foram mostradas, mas que mantêm a relação de não-dominância entre si e entre as duas mostradas em cada caso.

3.1.3 PRM e a Árvore de Steiner

Consideremos o Problema da Árvore de Steiner em Grafos (*Steiner Tree Problem in Graphs* ou STP), que pode ser assim enunciado: dado um grafo não-direcionado conectado $G = (V, E)$, uma função de peso $w : E \rightarrow \mathbb{N}$ e um conjunto $D \subset V$, deseja-se calcular uma árvore de custo mínimo que contenha os vértices de D . Em [31], Karp mostrou que o STP está na classe de problemas NP-difíceis por meio de uma redução do problema

Exact Cover. Observemos que o STP difere do problema de calcular uma árvore geradora mínima, o qual possui algoritmos de tempo polinomial (Prim e Kruskal, p.ex. [35]), uma vez que o STP não requer que todos os vértices de V estejam na solução; os únicos vértices obrigatórios são aqueles em D . Portanto, a dificuldade em resolver o STP reside na determinação de qual subconjunto $V' \subseteq V \setminus D$ irá levar à melhor árvore de Steiner em D .

Se for considerado apenas um peso por aresta (custo, p.ex.) e uma função objetivo ($\text{custo}(T)$, p.ex.) no PRM, então basta resolver o STP com conjunto de destinos $\{r\} \cup D$. Por outro lado, se for incluída uma restrição de *delay* máximo para cada destino, como feito na formulação PRMMo (Sec. 3.1.1), então estaríamos lidando uma variante do STP chamada Árvore de Steiner com Restrições [33], a qual possui apenas uma função objetivo com uma restrição de roteamento. Em alguns contextos, quando múltiplos objetivos são considerados, o problema de calcular árvores de Steiner também é chamado de *Multiobjective Steiner Tree Problem*.

3.2 Literatura Relacionada

Devido à complexidade computacional do problema do roteamento, existe um grande número de trabalhos orientados para suas variações. Diversos protocolos e algoritmos foram projetados para transmissões multicast; um survey pode ser encontrado em [38]. Algumas propostas utilizam algoritmos de menor caminho (Dijkstra e Bellman-Ford [14], p.ex.) e algoritmos para cálculo de árvore geradora mínima (tais como Prim e Kruskal [14], p.ex.) como sub-rotinas. Por exemplo, o DVMRP (*Distance Vector Multicast Routing Protocol*) [55] é um protocolo do tipo *distance-vector* que usa um procedimento baseado no algoritmo *reverse path-forwarding* de Dalal & Metcalfe [17], [20], o qual combina uma computação de menor caminho a um mecanismo de *flooding*. Por sua vez, o MOSPF (*Multicast Open Shortest Path First*) [37] é um protocolo *link-state* que usa o algoritmo de Dijkstra para construir uma rota baseada numa árvore de menores caminhos. Estes algoritmos eficientes são elaborados para trabalhar apenas com uma métrica (função objetivo) por vez, além de serem capazes de garantir a otimalidade da rota apenas quando tal métrica é baseada em caminhos unicast (como o *delay* fim-a-fim). Entretanto, se otimizar os objetivos significa otimizar o custo da árvore, então tais algoritmos não podem garantir que as rotas serão ótimas, visto que o problema se tornaria o Problema da Árvore de Steiner em Grafos ou alguma variante sua.

O STP tem sido investigado em diversas linhas. Dreyfus & Wagner [21] propuseram um algoritmo exato para resolvê-lo em tempo $O^*(3^{|D|})$, sendo baseado em programação dinâmica e fazendo uso de um algoritmo de menor caminho para calcular subárvores de Steiner como idéia principal. Outros autores que trabalharam em abordagens exatas para o STP foram Aneja [1] e Hakimi [26]. Por outro lado, há diversos procedimentos heurísti-

cos criados para o STP. Em [2], Aragão *et al.* propõem uma metaheurística GRASP com busca local baseada na heurística de Takahashi & Matsuyama [51]. Tal heurística possui uma razão de aproximação de 2opt e utiliza o algoritmo de Dijkstra para construir uma solução a partir da união incremental do caminho mais próximo entre algum destino não inserido e a árvore corrente.

Em uma variação muito útil do STP em aplicações do roteamento, o chamado Problema da Árvore de Steiner com Restrições ou *Constrained STP* (CSTP), Rouskas & Baldine [46] propuseram uma heurística para o problema sujeito a um limite para o *delay* fim-a-fim e variação do *delay*, cuja idéia principal é a criação de soluções a partir do cálculo de sucessivos menores caminhos (primeiros, segundos, etc.). Kompella *et al.* [33], por sua vez, propuseram uma heurística baseada em programação dinâmica (heurística KPP) para abordar o CSTP considerando o custo e uma restrição de *delay* máximo. Esta formulação para o CSTP é exatamente a formulação PRMMo estudada no presente trabalho. Ravi [44] apresenta algoritmos de aproximação com otimização local para algumas variações do STP.

Diversos trabalhos propõem abordagens baseadas em algoritmos genéticos para variações do PRM. Em [45] Ravikumar & Bajpai apresentam um AG mono-objetivo com os objetivos custo e *delay* agregados em uma expressão ponderada. Tal AG foi, posteriormente, melhorado em [60], com algumas mudanças no cruzamento e aplicando elitismo na reinserção. O modelo apresentado por Araujo & Oliveira em [39] foi baseado nestes dois trabalhos, preenchendo algumas lacunas, como a escolha do algoritmo de reconexão a ser usado durante o cruzamento e a mutação. Tal trabalho também propôs um mecanismo para aumentar a diversidade populacional chamado filtro. Em uma perspectiva multi-objetivo, Oliveira & Vita [40] basearam seu modelo nos trabalhos citados [45], [60], [39], utilizando diversos MOEAs baseados no NSGA (NSGA, NSGA-II e ϵ -NSGA-II), também avaliando algumas mudanças no cruzamento.

Ainda nas formulações multiobjetivo para o MRP, Araujo & Garrozi [24], [3] trabalharam com soma ponderada ao invés de dominância de Pareto numa formulação para maximizar as arestas comuns e minimizar o tamanho da árvore. Um esquema de cruzamento como uma permutação de arestas foi proposto, permitindo uma interessante exploração do espaço de busca, comparando os resultados obtidos com os algoritmos de menor caminho e *simulated annealing*. Cui *et al.* [16] proveram um modelo com idéias do MOGA (*Multiple Objective Genetic Algorithms*), uma MOEA desenvolvido por Fonseca & Fleming [23], para cálculo de aptidão, porém incluindo uma estratégia baseada em entropia como um segundo componente da aptidão. Em tal trabalho foram considerados três objetivos relacionados com o *delay* e capacidade dos enlaces, junto de uma restrição. Em [15] Crichigno & Barán estudaram um modelo baseado no SPEA com quatro objetivos e uma restrição (similar à formulação PRFM de quatro objetivos da Sec. 3.1.2) relacionados a requisitos de QoS e ET, desenvolvendo um cruzamento e mutação baseados em menores

caminhos pré-calculados. Prieto *et al.* em [43] estudaram uma formulação com múltiplas árvores multicast considerando cinco objetivos e duas restrições, também tendo como base o SPEA, porém utilizando a uma codificação similar à usada em [45]. Um survey em que são avaliados um grande número de propostas baseadas em AGs para o PRM é apresentado em [22], onde foi proposto um modelo baseado em SPEA para onze funções objetivo. Uma análise de correlação demonstrou que dentre os objetivos considerados é possível trabalhar com um conjunto menor para evitar redundâncias.

Capítulo 4

Modelos Anteriores para o PRM

4.1 Visão Geral

Em [45], Ravikumar & Bajpai propuseram um modelo original para uma formulação mono-objetivo do MRP, considerando o objetivo custo, sujeito a um *delay* máximo fim-a-fim. Este modelo foi caracterizado por:

- (i) uma codificação de indivíduo baseada em árvores genéricas enraizadas;
- (ii) operador de cruzamento baseado nas similaridades entre duas árvores; e
- (iii) operador de mutação, em que arestas de um indivíduo mutado podem ser removidas e novas arestas inseridas.

Ainda em uma perspectiva mono-objetivo, tal modelo foi melhorado posteriormente por Zhengying *et al.* [60] e Oliveira & Araújo [39], modificando-o pela inclusão de:

- (iv) elitismo [60];
- (v) um mecanismo para reduzir a repetição de indivíduos conhecido como filtro [39]; e
- (vi) preenchimento de algumas lacunas do modelo original como a escolha do algoritmo de reconexão durante o cruzamento e a mutação, além da definição de parâmetros do AG [39].

Em um momento posterior, o modelo resultante destes estudos serviu de base para algumas formulações multiobjetivo realizadas por Oliveira & Vita [40], [41], incluindo outras mudanças no cruzamento e na mutação (especificamente, o critério de escolha do peso durante a etapa de reconexões).

O modelo e os operadores genéticos resultantes de tais investigações são descritos a seguir. Tal modelo forma a base para as proposições a serem apresentadas no Cap. 5.

4.2 Representação dos Indivíduos e População Inicial

Cada indivíduo é codificado diretamente como uma árvore genérica, sendo gerado por um algoritmo de busca aleatória, no qual vértices são inseridos na árvore até que todos os destinos estejam presentes. A seguir, uma poda é aplicada sobre a árvore, para que todo vértice folha seja um destino. Pode-se observar que este algoritmo possui o mesmo tempo de pior caso de uma busca em largura ou profundidade, ou seja, $O(|E| + |V|)$, uma vez que a única diferença é que o próximo vizinho a ser escolhido durante a expansão é feito de forma aleatória, ao invés de usar uma fila ou pilha. Três exemplos de indivíduos, extraídos da rede 0 (Fig. 3.1), podem ser vistos na Figura 4.1.

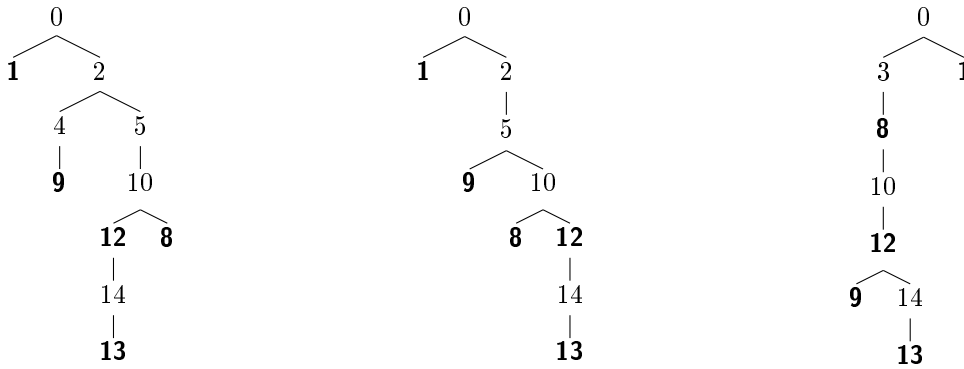


Figura 4.1: Soluções para a instância rede 0, onde $D = \{1, 8, 9, 12, 13\}$.

4.3 Cálculo de Aptidão

No PRMMo, avaliar um indivíduo corresponde a atribuir-lhe um valor de aptidão, o qual representa quão bem adaptado está o indivíduo no problema. Neste caso, a aptidão de um indivíduo é calculada a partir do valor da função objetivo (custo) de sua árvore, e a partir da consideração do seu nível de adaptação em relação à restrição d_{max} . Assim, primeiramente é calculado o custo da árvore (Eq. 3.1) e o *delay* fim-a-fim para cada destino (Eq. 3.2). Por fim, uma equação que combina esses dois componentes atribui para cada árvore um valor, dado pela Eq. 4.1.

$$F(T) = \frac{1}{custo(T)} \prod_{t \in D} (\omega(delay(t) - d_{max})) \quad (4.1)$$

onde $\omega(delay(t) - d_{max}) = \{1, \text{ if } delay(t) \leq d_{max}; 0.5 \text{ caso contrário}\}$

Nesta equação, a qualidade de uma solução é inversamente proporcional ao seu custo, também recebendo uma penalização de 50% para cada *delay* fim-a-fim não atendido. Esta estratégia tende a guiar a busca em direção a soluções que atendem d_{max} , devido à forte penalização presente.

Por outro lado, no PRMM e PRFM o cálculo de aptidão depende de procedimentos específicos definidos em cada MOEA, como descrito no capítulo anterior, e das funções objetivo que se deseja otimizar.

4.4 Cruzamento

Diferentemente do AG canônico, em que o cruzamento entre dois pais usualmente gera dois filhos, aqui apenas um filho é gerado a cada cruzamento. A partir de dois pais, um filho herda suas características seguindo o seguinte esquema:

- (i) identificação de arestas comuns aos pais, resultando numa floresta de subárvores;
- (ii) escolha de um dos pesos de G para ser usado ao longo das reconexões;
- (iii) reconexão das subárvores, resultando em uma nova árvore multicast contendo todos os destinos; e
- (iv) poda de vértices inúteis.

O resultado do passo i do cruzamento é uma floresta de G . A razão é que pode haver mais de uma subárvore comum, de forma que não haja arestas de ligando-as, uma vez que tais arestas de G não eram comuns aos pais. Pode-se observar que este passo pode ser feito em tempo $O(|V|)$, caso seja mantido para cada vértice uma referência para o seu pai. Então, basta realizar um percurso em um dos pais, verificando quais de suas arestas existem no outro pai.

O passo ii , por sua vez, depende de qual algoritmo será usado na religação de subárvores presentes na floresta. Quando a etapa de religação (iii) no PRMMo usa um algoritmo de menor caminho, uma decisão específica deve ser tomada. Para os demais casos, uma escolha aleatória é feita, como sumarizado a seguir:

PRMMo: caso $delay(t_i) \leq d_{max}, \forall t_i \in D$ em pelo menos um dos pais, o peso custo é escolhido; caso contrário, é escolhido o peso $delay$. Este é o critério original utilizado em [45] para obter uma melhor convergência de soluções que atendam a restrição d_{max} .

PRMM e PRFM: a escolha do peso é feita de forma aleatória. Este critério foi adotado em [41], pois retornou melhores conjuntos de árvores multicast na população final.

O passo iii refere-se às reconexões de subárvores da floresta; mais especificamente, consiste na adição de arestas a tais subárvores, até que ela se torne uma árvore enraizada em r contendo os vértices de D . Os algoritmos para escolher tais arestas têm um impacto decisivo na convergência do AG. No Cap. 5 são apresentadas três novas heurísticas para a

reconexão de subárvores. Como será visto ao longo desta seção, é precisamente a heurística original usada em tal processo que permite a eventual geração de indivíduos inválidos.

O passo final (*iv*) é uma poda de nós inúteis, i.e., uma operação que recursivamente remove vértices-folha que não são destinos. Este processo pode ser feito em tempo $O(|V|)$.

A complexidade do operador de cruzamento é dominada pela etapa de reconexão (*iii*), que conforme será visto, pode ser feita em $O(|V|^2)$ ou $O(|V||E| \log |V|)$, dependendo da heurística escolhida. As demais etapas (*i*, *ii* e *iv*) são limitadas por $O(|V|)$ como mostrado. Um exemplo de operação de cruzamento está ilustrado na Figura 4.2.

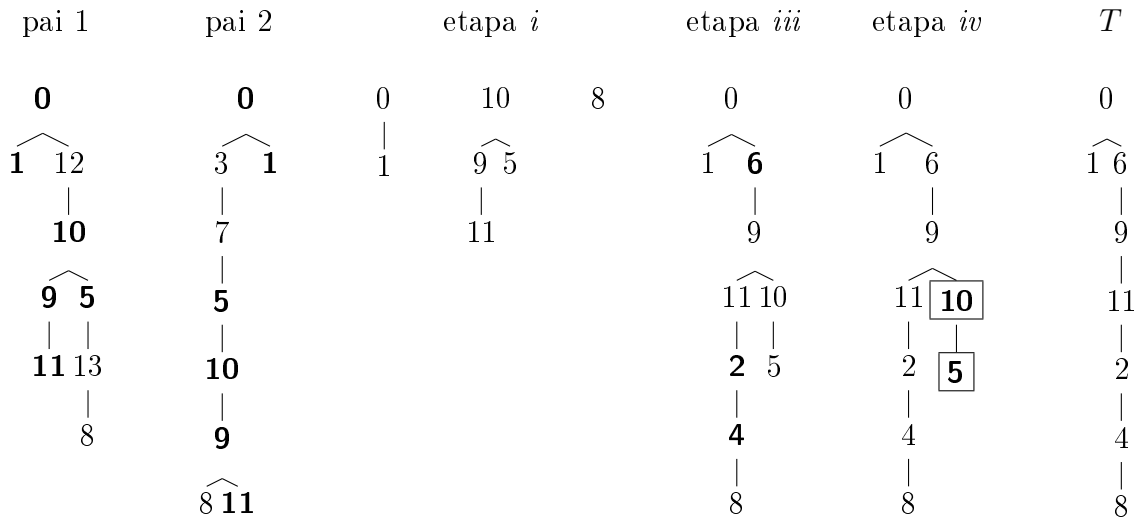


Figura 4.2: Etapas de cruzamento, considerando $D = \{1, 8, 11\}$. Legenda: (*i*): identificação de subárvores comuns aos pais (subárvores em negrito); (*iii*): reconexão de subárvores (vértices adicionados em negrito) ; (*iv*): poda (vértices a serem removidos estão marcados).

4.4.1 Reconexão de Subárvores

Na reconexão de subárvores objetiva-se construir uma nova árvore multicast T a partir de uma floresta de subárvores $s = \{s_1, s_2, \dots, s_k\}$. Cada s_i é obtida por um procedimento que calcula as arestas comuns a duas árvores. Além disso, se a raiz r ou um ou mais destinos não estiverem presentes em tais arestas comuns, subárvores isoladas são adicionadas em s (subárvores isoladas são aquelas que contêm apenas um vértice).

Assim, a partir de s , é necessário realizar $|s| - 1$ reconexões. Na primeira reconexão, duas subárvores s_i e s_j ainda não reconectadas são selecionadas, e um conjunto p de arestas é calculado, tal que $s_i \cup p \cup s_j$ seja um subgrafo conectado de G . Nas próximas reconexões, outra subárvore é escolhida e calcula-se um novo conjunto p entre ela e o subgrafo obtido na reconexão anterior.

A seguir, apresentamos a heurística original (chamada neste texto de h1) proposta

por Ravikumar & Bajpai [45], na qual um algoritmo de menor caminho é utilizado para reconectar subárvores, apesar de os autores não terem especificado qual algoritmo foi usado. Em [39] e [40] o Algoritmo de Dijkstra [14] foi usado para tal tarefa. O Algoritmo 7 descreve a heurística h1.

Algoritmo 7 Heurística h1 para reconexão de subárvores (G, r, D, s)

Entrada: $G = (V, E)$: um grafo conectado ponderado. r : nó fonte.

D : grupo multicast. s : floresta de subárvores.

Saída: $T = (V_T, E_T)$: uma árvore multicast enraizada em r , tal que $D \subset V_T$.

```

1:  $s_0 \leftarrow$  escolha elemento aleatoriamente ( $s$ )
2:  $s \leftarrow s \setminus \{s_0\}$ 
3:  $T \leftarrow s_0$ 
4: while  $s \neq \emptyset$  do
5:    $s_i \leftarrow$  escolha elemento aleatoriamente ( $s$ )
6:    $s \leftarrow s \setminus \{s_i\}$ 
7:    $p \leftarrow$  menorCaminho ( $T, s_i$ )
8:    $s'_i \leftarrow$  rotacione ( $s_i, p.end$ )
9:    $T \leftarrow T \cup p$ 
10:   $T \leftarrow T \cup s'_i$ 
11:  $T \leftarrow$  rotacione ( $T, r$ )
12: return  $T$ 

```

No Alg. 7, inicialmente é atribuída a T uma subárvore s_0 escolhida aleatoriamente. Ao entrar no laço, inicia-se o processo de reconexão, onde primeiramente outra subárvore s_i é escolhida aleatoriamente para que seja calculado o menor caminho p entre T e s_i usando o algoritmo de Dijkstra. Uma vez que G possui mais de um peso por aresta, um deles precisa ser escolhido para ser usado por este algoritmo, o que é feito como descrito na Seção 4.4.

O procedimento *rotacione*($A, novaRaiz$) altera uma árvore A recebida como argumento para que a sua raiz passe a ser *novaRaiz*. Assim, como o menor caminho p inicia em algum vértice de T e termina em algum vértice de s_i , caso o último vértice de p seja a raiz atual de s_i , nenhuma alteração será feita em s_i quando da chamada de *rotacione*. Caso contrário, s_i é rotacionada para que sua raiz passe a ser o vértice $p.end$, pois este vértice em s_i não poderá possuir um vértice pai, a fim de realizar $T \cup p \cup s_i$ como esperado. Em outras palavras, devem ser atendidas as propriedades de uma árvore enraizada: cada vértice deve possuir um pai apenas, e a raiz não deve possuir pai.

Enquanto houver alguma subárvore a ser reconectada, este processo é repetido, usando a árvore parcialmente construída T como primeira subárvore e outra subárvore s_i não incorporada, escolhida aleatoriamente, como segunda subárvore. Por fim, a árvore T obtida após a execução do laço deve ser rotacionada para r , pois é possível que a última reconexão tenha resultado em uma árvore T cuja raiz não seja r .

4.4.2 Complexidade da Reconexão

Inicialmente, observemos que o número máximo de subárvores é limitado por $O(|V|)$. Isso pode ocorrer quando D possuir o tamanho máximo $(|V| - 1)$ e não houver aresta comum entre as duas árvores submetidas ao cruzamento. Portanto, a floresta de subárvores s terá $|V|$ subárvores: uma contendo a raiz r e as demais contendo apenas um vértice de D .

Os três primeiros passos do algoritmo são limitados por $O(|V|)$. O cálculo do menor caminho entre T e s_i possui o custo do Algoritmo de Dijkstra, o que nos dá $O(|E| \log |V|)$ [14]. Para rotacionar s_i , a qual pode ter até $|V|$ vértices, pode ser necessário percorrê-la por completo, realizando operações limitadas por $O(1)$, resultando num tempo $O(|V|)$. A união entre T e p pode ser feita em $O(|V|)$, o mesmo tempo para a união entre T e s_i (ou até mesmo em tempo constante, dependendo da implementação).

Deste modo, a complexidade das operações realizadas no laço *while* é dominada por $O(|E| \log |V|)$. Como o número máximo de subárvores está em $O(|V|)$, o mesmo ocorre para o número de iterações do referido laço externo. Logo, o Algoritmo 7 possui complexidade de tempo da ordem de $O(|V||E| \log |V|)$.

4.4.3 Geração de Indivíduos Inválidos

Ao analisar o método de reconexão descrito, observou-se que indivíduos inválidos poderiam ser gerados devido a uma possível repetição de vértices. No começo de cada reconexão, o método inicia com uma floresta s de subárvores, tal que para quaisquer duas subárvores em s , não há arestas comuns. No entanto, a reconexão constrói T através da incorporação de menores caminhos de custo ou *delay* sem observar os vértices das subárvores que ainda deverão ser reconectadas. Assim, ao longo de iterações de reconexão, podem existir vértices comuns à árvore parcial T e o conjunto de subárvores ainda não reconectadas, como mostrado por um exemplo na Figura 4.3.

O rótulo “início” mostra o conjunto s , onde s_0 e s_2 foram sorteadas para compor o primeiro par a ser reconectado. Ao aplicar Dijkstra para reconectar s_0 e s_2 , obtém-se T como mostrado no rótulo “1a. reconexão”, onde $p = \langle 0, 5, 7, 9, 4 \rangle$. Na próxima iteração, T deve ser reconectada com s_1 , a única subárvore restante em s .

Ao tentar reconectar T e s_1 , há dois menores caminhos p para reconexão de menor custo (custo 0, neste caso): $\langle 5, 5 \rangle$ e $\langle 9, 9 \rangle$. Neste momento ocorre a falha, pois um ou mais vértices de T podem ficar repetidos. Assim, qualquer uma das duas escolhas leva à violação do indivíduo, como exibido na Figura 4.3. Nesta figura, pode-se ver que se for escolhida a reconexão pelo vértice 5, o resultado é uma repetição do vértice 9. Para o caso de a reconexão ser feita pelo vértice 9, será obtida uma repetição do vértice 5. Soluções inválidas violam o formato especificado pelo problema, além de possuírem um custo maior (devido aos vértices repetidos), e, para cada destino, poder existir mais de um

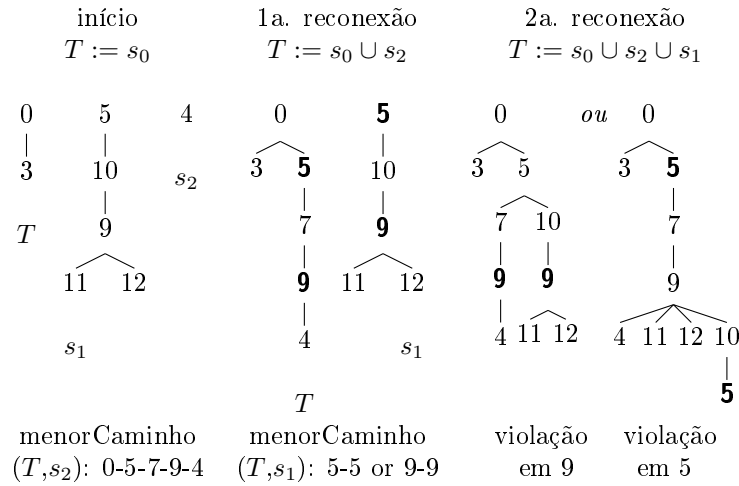


Figura 4.3: Um exemplo de conflito durante a reconexão de subárvores. Após a segunda reconexão, os vértices em negrito tornam inválidas as duas possíveis soluções.

delay fim-a-fim associado. Além disso, tais soluções não podem ser consideradas árvores de fato, pois essa repetição de vértices pode ser vista como a ocorrência de laços.

Para avaliar os efeitos desta falha, dentre as instâncias do PRM descritas no Cap. 6, foram selecionadas duas redes pequenas (redes 0 e 1, apresentadas anteriormente nas figuras 3.1 e 3.2), e uma maior (rede 2, com 33 vértices). Deste modo, executou-se 100 vezes o AG na formulação PRMMo, com parâmetros $Tp = 30$ e $Ng = 50$ (redes 0 e 1) e $Tp = 60$ e $Ng = 100$ (rede 2), a fim de obter a seguinte estatística: número médio de cruzamentos em cada geração que constroem indivíduos inválidos. Os resultados relacionados a esta estatística estão ilustrados na Figura 4.4, para cada topologia.

O gráfico indica que ao longo das gerações existe uma tendência de queda do percentual médio de filhos inválidos gerados por geração. Como nas gerações iniciais tende a existir maior diversidade populacional, é possível que exista uma maior probabilidade de que violações ocorram nas reconexões. Nas redes 0 e 1, é possível observar que o percentual de indivíduos inválidos gerados ao final de cada crossover estabiliza em um nível relativamente baixo (em torno de 10%). Entretanto, na rede 2 observa-se que mesmo após a queda inicial, o percentual de indivíduos inválidos é estabilizado em torno de 40%, um valor alto. Uma vez que 30 indivíduos são calculados em cada geração (taxa de cruzamento = 50%), em torno de 12 deles são inválidos. Deste modo, pode-se dizer que a taxa real de cruzamento é 30% e que há um desperdício considerável de esforço computacional.

Após a reinserção ao final de cada geração, obteve-se uma segunda estatística referente ao número de indivíduos inválidos remanescentes na população, para observar se tais soluções estavam sobrevivendo à seleção do AG, como mostrado na Figura 4.5. É possível ver que nas gerações iniciais ocorre uma ascensão rápida dos percentuais médios, uma vez que o AG inicia com uma população válida, ocorrendo a sobrevivência e aumento de indivíduos inválidos até a terceira/quarta gerações. A partir deste ponto, o percentual de

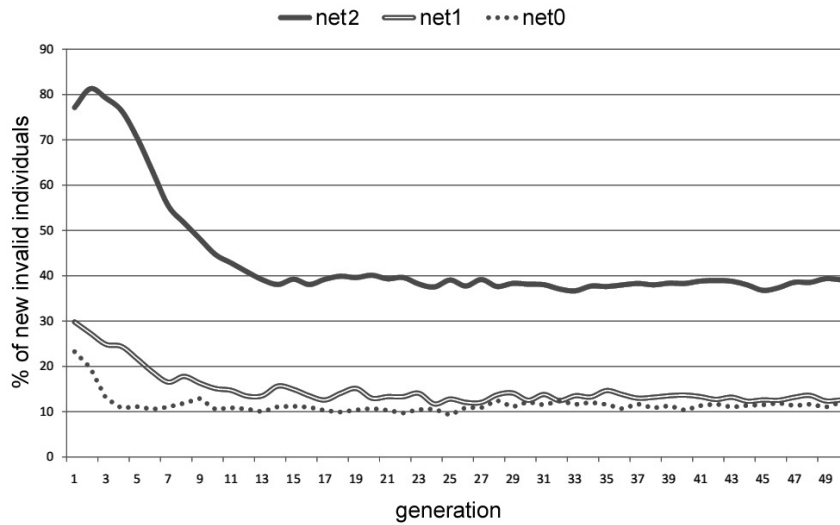


Figura 4.4: Produção média de indivíduos inválidos por geração.

indivíduos inválidos não se altera muito ao longo das gerações. O gráfico mostra que o percentual de novos e antigos indivíduos inválidos é alto, especialmente em uma rede de dimensão maior, como a rede 2. Neste caso, quase 70% da população é constituída por indivíduos inválidos, i.e., soluções inconsistentes para o roteamento.

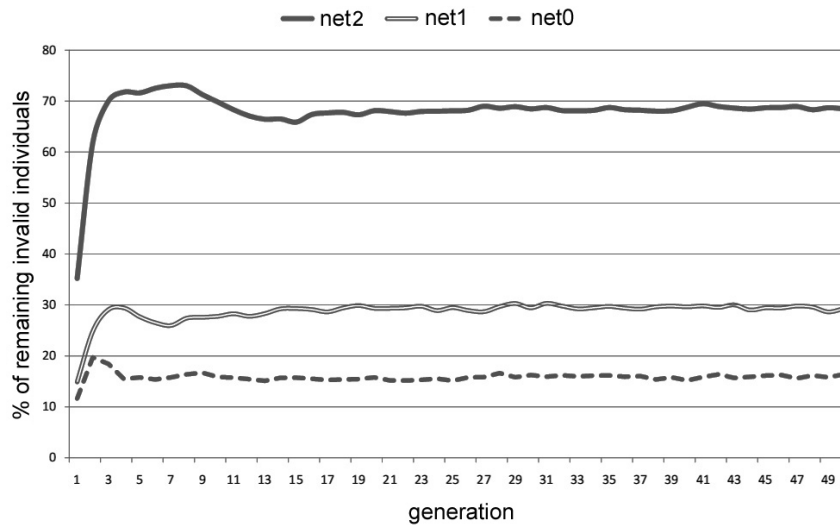


Figura 4.5: Percentual médio de indivíduos inválidos remanescentes ao final de cada geração (após reinserção).

4.5 Mutação

O operador de mutação consiste em uma escolha aleatória de um conjunto de vértices de uma árvore multicast e na posterior remoção de suas ligações, i.e., para cada vértice de tal conjunto, remover suas arestas com seu pai e com seus filhos. O resultado é uma floresta de subárvores, as quais devem ser reconectadas assim como no cruzamento. Entretanto, antes de iniciar as reconexões, o peso é escolhido aleatoriamente, mesmo no

PRMMo, como proposto em [39]. Um exemplo de execução do operador de mutação está mostrado na Figura 4.6. Os métodos de reconexão discutidos para o cruzamento podem também ser aplicados ao operador de mutação.

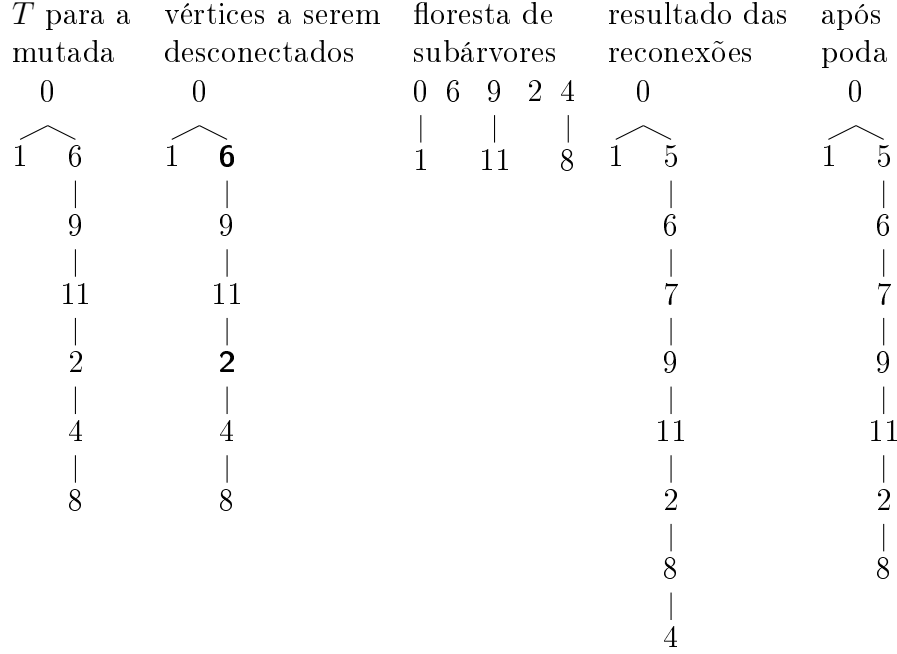


Figura 4.6: Um exemplo de aplicação do operador de mutação, considerando que $\lceil 20\% \rceil$ de vértices são desconectados.

4.6 Filtro

Um mecanismo conhecido como filtro [39] projetado para reduzir a repetição de indivíduos na população foi incorporado neste modelo. O filtro funciona aplicando uma alteração em cada filho gerado quando ele já existir na população corrente. Nos experimentos, avaliamos o desempenho do AG com e sem esse mecanismo.

4.7 Seleção

4.7.1 Seleção para Cruzamento

No PRMMo, usa-se um torneio simples com $tour = 3$.

Para os casos do PRMM e PRFM, em todos os três MOEAs, duas execuções do torneio binário simples são realizadas para selecionar um par de pares para cruzamento. No NSGA-II, vence o torneio o indivíduo que possuir o menor *rank* ou a maior *crowding distance* caso os *rank*s sejam iguais. Já no SPEA, o vencedor será o indivíduo não-dominado de acordo com o cálculo de *strength* apresentado na Sec. 2.3.2. Por sua vez,

no SPEA2 é incorporada uma métrica de densidade em conjunto com a *strength* (porém, usada de um forma diferente do SPEA), de modo que o vencedor será um indivíduo não-dominado cuja distância ao vizinho mais próximo seja maior que a dos demais não-dominados envolvidos no torneio.

4.7.2 Reinserção

No PRMMo, foi definido um tamanho de elite igual a 50%, resultando numa taxa de cruzamento igual a 50%. Portanto, ao final de cada geração, a população corrente é ordenada e os 50% piores indivíduos são substituídos pelos filhos gerados pelo cruzamento e mutação.

Para os casos do PRMM e PRFM, variações do elitismo são utilizadas nos três MOEAs. No NSGA-II, sobrevivem ao final de cada geração os melhores indivíduos entre pais e filhos. Já no SPEA e SPEA2, os melhores indivíduos dentre os filhos e os do arquivo atual são copiados para o arquivo atualizado, ou seja, indivíduos do arquivo podem ser substituídos por filhos ao longo das gerações. A diferença principal com o NSGA-II é que tanto o SPEA como o SPEA2 utilizam procedimentos de clusterização quando o arquivo excede certo limite fixado previamente, para manter as características do conjunto não-dominado encontrado até então.

4.8 Roteamento Evolutivo

A partir dos operadores definidos para o AG, apresenta-se no Alg. 8 o algoritmo para o modelo resultante para a formulação mono-objetivo no Roteamento.

Algoritmo 8 AG mono-objetivo para o roteamento (Tp, Ng)

Entrada: Tp : número de árvores multicast na população P . Ng : número de gerações.
Saída: T : a árvore mais adaptada encontrada durante a busca evolutiva.

- 1: Gere uma população inicial P_0 composta por árvores multicast aleatórias
 - 2: Avalie os indivíduos de P_0 (Eq. 4.1)
 - 3: $t \leftarrow 0$
 - 4: **while** $t < Ng$ **do**
 - 5: Selecione $Tp/2$ pares de árvores de P_t por meio de um torneio com $tour = 3$
 - 6: Aplique o cruzamento e a mutação sobre os pares de pais selecionados no estágio anterior para gerar um conjunto de filhos Q_t . Ao gerar um filho, se ele já existir em $P_t \cup Q_t$ atual, aplique um filtro sobre ele (caso o filtro esteja ativado)
 - 7: Avalie os indivíduos de Q_t (Eq. 4.1)
 - 8: Ordene P_t de acordo com a aptidão
 - 9: Substitua os 50% piores indivíduos de P_t pelos indivíduos de Q_t
 - 10: $t \leftarrow t + 1$
 - 11: **return** a árvore mais adaptada de P_t
-

Os AGs para o caso multiobjetivo não foram apresentados pois podem ser obtidos diretamente a partir dos modelos dos MOEAs apresentados nos algoritmos 3, 5 e 6.

Capítulo 5

Novas Abordagens para o Roteamento Evolutivo

Este capítulo introduz as principais inovações que foram investigadas neste trabalho. Inicialmente, apresentamos três novas heurísticas desenvolvidas para os operadores de cruzamento e mutação exibidos no capítulo anterior. Em seguida, discutimos as funções objetivo que foram adotadas e as razões para tal escolha. Finalmente, discutimos a incorporação do Cruzamento pela Vizinhança em três MOEAs, além de novos filtros para o modelo evolutivo.

5.1 Heurísticas para Reconexão de Subárvores

Uma vez que a heurística h1 (Alg. 7) utiliza as idéias de menor caminho e similaridade entre pais, as novas heurísticas possuem um direcionamento diferente, sendo baseadas na prevalência do menor caminho sobre a similaridade (heurística h2), uso exclusivo da similaridade (heurística h3) ou ainda uma escolha aleatória entre as duas anteriores (heurística h4), a fim de construir uma árvore T válida durante o cruzamento. A seguir, descrevemos as três heurísticas, incluindo exemplos e análises de complexidade.

5.1.1 Heurística h2

Seja T a árvore resultante das reconexões feitas até o momento no processo de cruzamento, e s_i a subárvore a ser reconectada na iteração corrente. Para evitar invalidar T , duas estratégias são adotadas:

- um teste para detecção de conflito após o cálculo de menor caminho entre T e s_i ;
- inserção de arestas de s_i que não invalidem T , i.e., arestas $(x, y) \in s_i$, tais que $y \notin T$.

Esta modificação no modo como arestas de s_i são inseridas implica que, quando um conflito puder ocorrer, as adjacências de seus vértices a serem efetivamente inseridos

podem ser alteradas, com o objetivo de evitar a violação do indivíduo sendo construído. Assim, ao usar a heurística h2, é possível que alguma informação de similaridade entre pais seja perdida, pela priorização da informação de menor caminho. Este método está apresentado no Algoritmo 9.

Algoritmo 9 Heurística h2 para reconexão de subárvores (G, r, D, s)

Entrada: $G = (V, E)$: um grafo conectado ponderado. r : nó fonte.

D : grupo multicast. s : floresta de subárvores.

Saída: $T = (V_T, E_T)$: uma árvore multicast enraizada em r , tal que $D \subset V_T$.

```

1:  $s_0 \leftarrow$  escolha elemento aleatoriamente ( $s$ )
2:  $s \leftarrow s \setminus \{s_0\}$ 
3:  $T \leftarrow s_0$ 
4: while  $s \neq \emptyset$  do
5:    $s_i \leftarrow$  escolha elemento aleatoriamente ( $s$ )
6:    $s \leftarrow s \setminus \{s_i\}$ 
7:    $p \leftarrow$  menorCaminho ( $T, s_i$ )
8:    $s'_i \leftarrow$  rotacione ( $s_i, p.end$ )
9:   if houver conflito em  $p$  then
10:    for cada aresta  $(x, y) \in s'_i$  do
11:      if  $y \notin V_T$  then
12:         $T \leftarrow T \cup \{(x, y)\}$ 
13:   else
14:      $T \leftarrow T \cup p$ 
15:      $T \leftarrow T \cup s'_i$ 
16:  $T \leftarrow$  rotacione ( $T, r$ )
17: return  $T$ 

```

Após a inicialização de T , o laço de reconexão é iniciado, o qual constrói T a partir de sucessivas reconexões com as demais s_i pelo cálculo de um menor caminho p . Assim, a próxima subárvore s_i é escolhida aleatoriamente, e é calculado o menor caminho p entre ela e T . O mesmo procedimento para rotacionar s_i usado em h1 é chamado.

O próximo passo consiste em um teste para verificar se ocorreria um conflito ao reconectar T e s'_i , onde verificamos se p possui uma única aresta com as mesmas pontas. Se existe um conflito, a estratégia da heurística h2 insere apenas arestas (x, y) de s'_i tais que o vértice y não exista em T . A razão é que, iniciando pela raiz, ao inserir uma aresta (x, y) de s'_i em T , o vértice correspondente ao pai (vértice x) deve existir em T , diferentemente de y , que será anexado a x de T , mantendo esta conectada. Assim, se y também existir em T , não poderemos inseri-lo em T , pois a árvore passaria a ter dois vértices y com pais diferentes, o que invalidaria a solução. Além disso, no caso de um conflito, o caminho p não precisa ser inserido, uma vez que ele contém apenas um vértice já existente em T .

Por outro lado, se não houver conflito, T é unida com p e s'_i , assim como ocorre em h1. As reconexões são feitas até que não haja mais subárvores a serem reconectadas com T . Um exemplo onde um conflito é evitado pode ser visto na Figura 5.1.

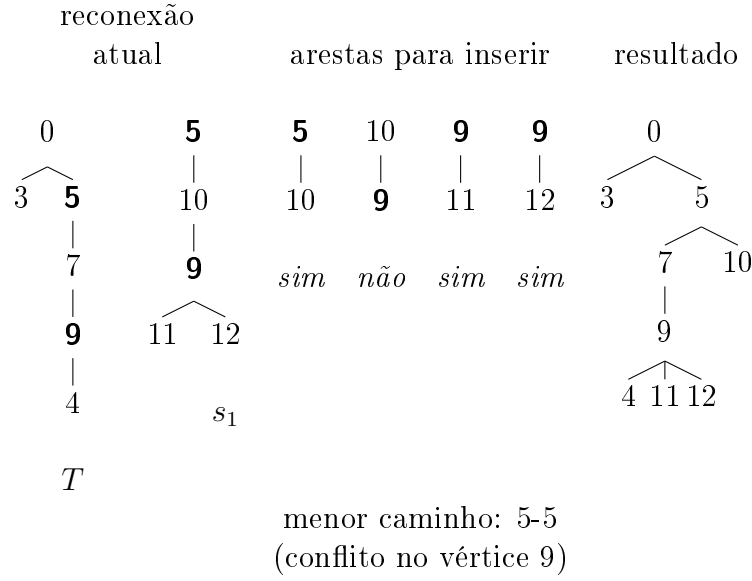


Figura 5.1: Um exemplo de reconexão utilizando h2. Vértices em negrito correspondem àqueles em comum entre T e s_1 .

A heurística h2 é similar a h1 apresentada no Alg. 7, porém preserva a validade de T ao longo das reconexões, uma vez que evita a ocorrência de conflitos, o que é feito no início do primeiro braço do condicional (linhas 9 a 12). Entretanto, como mencionado, a informação de similaridade, obtida no início do cruzamento, pode ser perdida. No exemplo da Figura 5.1, a aresta $(10, 9)$ está em s_1 pois foi identificada como uma aresta comum aos pais. Por outro lado, é fácil ver que a aresta $(7, 9)$ somente pôde ser adicionada em T em alguma reconexão anterior devido a algum cálculo de menor caminho. Porém, como há um conflito entre a informação de similaridade (aresta $(10, 9)$ em s_1) e a informação de menor caminho obtida em iterações anteriores (aresta $(7, 9)$ em T), o método h2 mantém a informação de menor caminho (aresta $(7, 9)$) e perde a aresta de similaridade $(10, 9)$. Assim, é possível dizer que h2 tenta usar ambas informações - similaridade e menor caminho - heurísticamente ao construir uma nova árvore. Entretanto, quando há um conflito, a heurística prioriza a informação de menor caminho.

A complexidade do Alg. 9 é $O(|V||E| \log |V|)$. Inicialmente, observemos que a complexidade dos quatro comandos localizados no laço while que precedem o primeiro condicional if-else é dominada por $O(|E| \log |V|)$, que corresponde à complexidade do cálculo do menor caminho entre T e s_i . No primeiro braço do condicional, o teste é $O(1)$, uma vez que precisamos verificar apenas se p possui somente uma aresta com as mesmas pontas. Os comandos no interior do condicional são dominados por $O(|V|)$, pois $|V|$ é o número máximo de vértices em s_i ; conseqüentemente, o número máximo de arestas é $|V| - 1$. Analogamente, a complexidade do segundo braço do condicional é dominada por $O(|V|)$, considerando que p nunca possui mais que $|V|$ vértices. Assim, os comandos do laço externo são dominados por $O(|E| \log |V|)$. Como o número de execuções do laço externo é

limitado por $O(|V|)$, a complexidade do Algoritmo 9 é $O(|V||E| \log |V|)$.

5.1.2 Heurística h3

Diferentemente de h2, em h3 a informação de similaridade entre pais é mantida durante a reconexão de subárvores. O cálculo de menor caminho não é utilizado aqui. O Alg. 10 formaliza este processo.

Na heurística h3, após a inicialização de T , esta é preenchida aleatoriamente utilizando a mesma idéia do algoritmo de busca aleatória usado na geração da população inicial (ver Seção 4.2), até que apareça um vértice existente em alguma subárvore s_i não reconectada ainda. Neste momento, tal s_i é rotacionada e unida a T . O algoritmo repete esses passos enquanto houver alguma subárvore a ser reconectada. Um exemplo pode ser visto na Figura 5.2.

Algoritmo 10 Heurística h3 para reconexão de subárvores (G, r, D, s)

Entrada: $G = (V, E)$: um grafo conectado ponderado. r : nó fonte.

D : grupo multicast. s : floresta de subárvores.

Saída: $T = (V_T, E_T)$: uma árvore multicast enraizada em r , tal que $D \subset V_T$.

```

1:  $s_0 \leftarrow$  escolha elemento aleatoriamente ( $s$ )
2:  $s \leftarrow s \setminus \{s_0\}$ 
3:  $T \leftarrow s_0$ 
4: while  $s \neq \emptyset$  do
5:    $u \leftarrow$  escolhaPai ( $T$ )
6:    $v \leftarrow$  escolhaFilho ( $u, T, G$ )
7:    $T \leftarrow T \cup \{(u, v)\}$ 
8:   if  $v$  existir em alguma  $s_i$  não reconectada ainda then
9:      $s \leftarrow s \setminus \{s_i\}$ 
10:     $s'_i \leftarrow$  rotacione ( $s_i, v$ )
11:     $T \leftarrow T \cup s'_i$ 
12:  $T \leftarrow$  rotacione ( $T, r$ )
13: return  $T$ 
```

Neste método, a validade de T é preservada. Durante o preenchimento aleatório de T corrente, sempre que um vértice é inserido, caso ele exista em alguma s_i não incorporada ainda, tal subárvore é imediatamente rotacionada e unida a T , garantindo a sua consistência.

A heurística h3 é bastante diferente de h1 e h2, pois a informação de menor caminho não é utilizada em nenhum momento. Por outro lado, é possível dizer que a informação de similaridade entre pais é completamente preservada na árvore final T , ao usar este método. Portanto, os métodos h2 e h3 garantem a validade da árvore final; no entanto, o primeiro prioriza a informação de menor caminho em detrimento da informação de similaridade, enquanto o segundo método usa apenas a informação de similaridade.

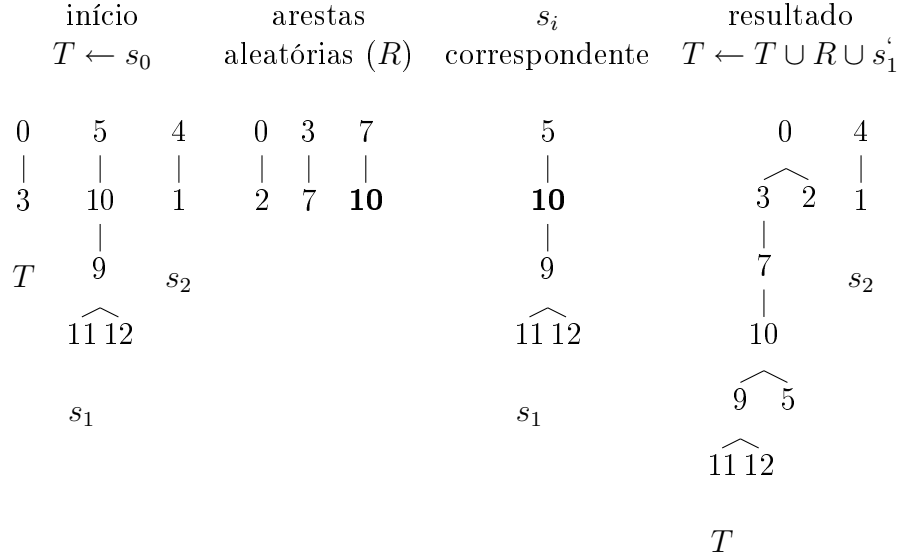


Figura 5.2: Um exemplo de reconexão utilizando h3. Arestas aleatórias (dadas pelo conjunto R) são inseridas em T até que uma delas contenha um vértice que exista em alguma subárvore s_i não reconectada, o que é mostrado na figura pelo vértice em negrito. Neste momento, a respectiva s_i que contém tal vértice é rotacionada e unida a T .

A complexidade do Algoritmo 10 é $O(|V|^2)$. O número máximo de execuções do primeiro laço é $O(|V|)$, cujo teste de saída é $O(1)$. Dentro deste laço, os três comandos que precedem o condicional if-else são $O(|V|)$. Por sua vez, o teste condicional e seus comandos são $O(|V|)$. Portanto, em uma análise de pior caso, o Algoritmo 10 possui complexidade de tempo $O(|V|^2)$. Deste modo, a heurística h3 possui uma complexidade de ordem inferior à heurística h2 (que é $O(|V||E| \log |V|)$), portanto, espera-se um tempo de execução menor para h3.

5.1.3 Heurística h4

Experimentos iniciais com algumas instâncias do problema indicaram que, em alguns casos, o uso da heurística h2 retornou melhor convergência do AG. Entretanto, em outros casos, tal comportamento foi obtido com a heurística h3. Assim, a heurística h4 propõe uma combinação entre as heurísticas h2 e h3 da seguinte forma: no início de cada cruzamento, após o cálculo das subárvores comuns e a escolha do peso, é feito um sorteio para decidir qual das duas heurísticas será usada para realizar as reconexões das subárvores do conjunto s . Portanto, a heurística h4 é uma escolha aleatória entre as heurísticas h2 e h3 no início de cada cruzamento.

Observe que, assim como as duas heurísticas usadas em sua forma original, a validade de T final é preservada pela heurística h4. A razão é que neste esquema cada cruzamento pode ser feito usando a heurística h2 ou h3, as quais são válidas, como discutido previamente.

O pior caso relativo à complexidade deste método ocorre quando h_2 é sempre escolhida, o que resulta em uma complexidade $O(|V||E| \log |V|)$. Porém, se considerarmos que o sorteio para decisão entre h_2 e h_3 usará uma distribuição uniforme de probabilidade, então cada heurística será usada em aproximadamente metade dos casos. Logo, é esperado que o AG que utiliza a heurística h_4 gaste um tempo de execução situado entre os tempos de h_2 e h_3 .

5.2 Filtros

Considerando que uma das principais dificuldades para convergência do AG é a falta de diversidade populacional, Oliveira & Araújo propuseram em [39] o mecanismo filtro (Sec. 4.6), como uma tentativa de aumentar a diversidade da prole gerada durante as gerações. O funcionamento do filtro consiste em alterar cada filho gerado se ele já existir na população corrente. Três algoritmos para o filtro foram avaliados neste trabalho:

f_{rav} proposto por Ravikumar & Bajpai [45], correspondendo ao algoritmo usado na mutação (Sec. 4.5);

f_{rnd} substitui o filho repetido por um novo, gerado aleatoriamente;

f_{tm} baseado nas idéias da heurística Takahashi-Matsuyama [51].

A seguir, são detalhados os dois novos filtros (f_{rnd} e f_{tm}).

5.2.1 Filtro f_{rnd}

Este filtro substitui um filho repetido por um novo indivíduo gerado aleatoriamente. Para isso, usa-se o mesmo algoritmo de geração da população inicial. Portanto, o tempo de f_{rnd} é $O(|E| + |V|)$.

Crichigno & Baran usaram uma idéia similar para substituir indivíduos repetidos presentes na população corrente [15]. No entanto, em tal trabalho, os indivíduos repetidos são substituídos no começo de cada geração do AG, e não no momento em que são gerados, como é o caso dos filtros apresentados aqui.

5.2.2 Filtro f_{tm}

O filtro f_{tm} é baseado nas idéias da heurística Takahashi-Matsuyama (TM) [51], que pode ser resumidamente descrita da seguinte forma: iniciando com uma árvore contendo apenas a raiz, iterativamente adicione o destino mais próximo por meio de um menor caminho, até que não haja mais destinos a serem inseridos. No contexto do filtro, o indivíduo a sofrer a mutação já está completo (i.e., possui todos os destinos); deste modo,

a aplicação do filtro inclui um passo inicial adicional referente à poda de um número de destinos, para depois religá-los à sua árvore por meio de um menor caminho, analogamente à heurística TM. O Alg. 11 apresenta este filtro.

Algoritmo 11 Filtro f_{tm} (T)

Entrada: $T = (V_T, E_T)$: uma árvore multicast com conjunto de destinos D_T .

Saída: $T = (V_T, E_T)$: uma nova árvore multicast com conjunto de destinos D_T .

```

1:  $D' \leftarrow$  escolha aleatoriamente  $\lceil 20\% \rceil$  de vértices de  $D_T$  que sejam folhas
2:  $D_T \leftarrow D_T \setminus D'$ 
3: poda ( $T$ )
4: while  $D' \neq \emptyset$  do
5:    $p \leftarrow$  menorCaminho ( $V_T, D'$ )
6:    $T \leftarrow T \cup p$ 
7:    $D' \leftarrow D' \setminus \{p.end\}$ 
8: return  $T$ 

```

O algoritmo proposto f_{tm} inicia escolhendo aleatoriamente um conjunto D' de destinos que sejam folhas para serem desconectados; neste trabalho, o tamanho de D' foi definido como $\lceil 20\% \rceil$ de D_T . Em seguida, altera-se a definição de D_T , de forma a permitir a aplicação de um procedimento de poda sobre T para remover os elementos de D' e eventualmente outros vértices não-destinos que possam aparecer durante o procedimento recursivo. Agora, é preciso reconectar os destinos de D' com T , o que é feito usando a mesma idéia da heurística TM: no começo de cada iteração do laço, seleciona-se o destino de D' mais próximo a T , adicionando tal menor caminho a T e removendo tal destino de D' . Este processo continua até que todos os destinos de D' sejam reconectados a T .

O algoritmo f_{tm} possui complexidade $O(|D||E| \log |V|)$. O gasto de tempo é dominado pelo cálculo de menor caminho ($O(|E| \log |V|)$), o qual é, por sua vez, executado $|D'|$ vezes, no máximo, dentro do laço while (observemos que $|D'|$ é limitado por $|D|$).

5.3 Cruzamento pela Vizinhaça

A principal idéia do Cruzamento pela Vizinhaça (*Neighborhood Crossover* ou NC) é cruzar indivíduos próximos, com o objetivo de auxiliar a geração de indivíduos tão bons ou melhores que os pais. De um modo geral, as noções de proximidade e vizinhaça foram exploradas em vários trabalhos na literatura de Otimização Evolutiva Multiobjetivo [57], [32], [59], [27], [28], [58].

Watanabe *et al.* [57] propuseram um MOEA baseado no SPEA2 com Cruzamento pela Vizinhaça como estratégia de seleção para cruzamento, chamado *Neighborhood Cultivation Genetic Algorithm* (NCGA). No NCGA, no início de cada geração, uma das funções objetivo é selecionada e a população é ordenada de acordo com tal objetivo. O objetivo enfatizado é alterado em cada geração, de forma seqüencial. Após a ordenação, pares

seqüenciais de pais são formados, os quais serão os pares a serem cruzados - procedimento chamado *local cultivate*, como mencionam os autores.

Como pode ser observado, esta estratégia difere da seleção para cruzamento usual, na qual pares de pais são formados de maneira aleatória (através de um torneio binário ou uma roleta, p.ex.). Apesar de os resultados experimentais de tal trabalho sobre MOPs de dois objetivos terem mostrado que o NCGA retornou melhores resultados que o NSGA-II e o SPEA2 nas métricas utilizadas, estudos posteriores [32], [59] mostraram que tal determinismo presente na seleção para cruzamento no NCGA poderia levar a uma redução da habilidade do AG em explorar o espaço de busca. Isso porque, este procedimento, assim expresso, pode resultar em falta de diversidade, pois não se pode garantir que indivíduos similares irão resultar em melhores filhos sempre.

Assim, para evitar o forte determinismo presente no NCGA, Kim et al. [32] propuseram o MOEA SPEA2+, o qual incorpora a idéia de NC junto com outras idéias (p.ex., um arquivo adicional para armazenar soluções elite no espaço de decisão). Além disso, um procedimento que mistura a população ordenada entra em ação para aliviar o forte determinismo presente no NCGA. Este procedimento de embaralhamento (*shuffling*) consiste em aplicar uma permutação em indivíduos localizados em um pequeno intervalo (p.ex., algo em torno de 10% a 20%). Outra diferença é que no começo de cada geração, o objetivo a ser enfatizado é selecionado aleatoriamente, e não mais seqüencialmente. Os resultados experimentais mostraram que o SPEA2+ levou a resultados melhores que SPEA2 e NSGA-II. Este estudo foi seguido por Yoshii et al. [59], trabalho no qual o NC foi aplicado, desta vez tendo como base o MOEA NSGA-II, avaliando diferentes tamanhos de permutação da população.

A partir das idéias presentes em [32] e [59], desenvolvemos duas implementações para o NC, a saber:

- **NC1:** Os indivíduos dentro de um intervalo contíguo de 20% do tamanho da população são permutados, de modo que o começo deste intervalo é definido aleatoriamente. A população de filhos é gerada aplicando cruzamento e mutação em cada par formado pelo i -ésimo e $(i + 1)$ -ésimo indivíduo da população permutada. O último indivíduo cruza com o primeiro.
- **NC2:** Cada par de pais é composto pelo i -ésimo indivíduo e algum outro no intervalo $[i + 1, i + \text{rand}(1 \dots 10\% \times |P|)]$, i.e., o “vizinho” será um indivíduo escolhido aleatoriamente dentro da faixa permitida (10% do tamanho da população).

Os valores de comprimento da permutação para NC1 e NC2 (20% e 10%, respectivamente) foram determinados empiricamente e seguindo alguns resultados obtidos em [59]. Na Fig. 5.3 estes dois tipos NC são ilustrados, na qual cada círculo representa um indivíduo da população.

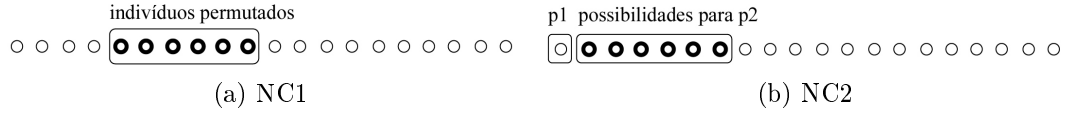


Figura 5.3: Tipos de Cruzamento pela Vizinhança implementados. Aqui, $|P| = 20$ e o comprimento da faixa de permutação foi definida como 30% em ambos os casos.

5.4 A Escolha das Funções Objetivo

As formulações de dois objetivos (MMRP) foram originalmente propostas por Oliveira & Vita em [40] e [41]. Nestes casos, apesar de não terem se mostrado difíceis de convergir na maior parte das vezes, quando um dos objetivos é fixado (*custo*), é possível observar que as diferentes formulações de dois objetivos apresentam dificuldades diferentes para as heurísticas, conforme os resultados a serem apresentados no Cap. 6. Isso também nos mostra o efeito que diferentes objetivos podem ter, gerando conjuntos P^* com diferentes formatos e dificuldades para serem obtidos.

Em um momento posterior, o fator fluxo foi incorporado explicitamente, seguindo as linhas de investigação de Crichigno & Barán [15], Prieto *et al.* [43] e Fabregat *et al.* [22]. Tomando por base tais formulações, foi desenvolvida a formulação com quatro objetivos e uma restrição (P6). Esta formulação (MFRP) têm como principal diferença em comparação àquelas do MMRP a presença do objetivo α_{max} (Eq. 3.8), o qual também possui uma relação com requisitos de Engenharia de Tráfego. Este requisito dá uma informação sobre o gargalo de cada rota em relação aos recursos que ela utilizaria, caso fosse carregada na rede. A inclusão da restrição de capacidade de tráfego por aresta torna o modelo mais realista, o que também torna mais difícil a busca dependendo do ϕ submetido, pois achar rotas viáveis pode ser mais difícil nessas circunstâncias.

Uma inovação em relação aos modelos pesquisados na literatura foi a inclusão do objetivo $\bar{\alpha}$ (Eq. 3.9), permitindo uma avaliação complementar junto ao objetivo α_{max} . Experimentalmente, é possível observar que esta inclusão aumenta o leque de possíveis rotas, oferecendo um interessante mecanismo para os decisores e administradores de rede. Por exemplo, consideremos duas soluções Pareto-ótimas cujos objetivos sejam dados por $s_1 = (0.3898; 0.5423; 81.3; 63; 23)$ e $s_2 = (0.4483; 0.5423; 75.9; 42; 19)$, onde cada tupla é denotada por $(\bar{\alpha}; \alpha_{max}; \phi \times \text{custo}; D_{max}; hops)$. Como todos os objetivos devem ser minimizados, claramente s_1 seria dominada por s_2 se o objetivo $\bar{\alpha}$ fosse descartado. No entanto, quando $\bar{\alpha}$ é considerado, s_1 apareceria em P^* e poderia ser uma solução interessante para um determinado momento na rede, uma vez que ela apresenta o mesmo gargalo que s_2 porém percorreria links menos carregados, de um modo geral.

Finalmente, o efeito provocado pela inclusão de certas métricas que admitem valores reais, tais como $\bar{D}$ e $\bar{\alpha}$, é percebido pelo aumento da cardinalidade dos conjuntos P^* , especialmente quando o número de objetivos também é aumentado. Isso traz um benefício do ponto de vista de oferecer mais opções à escolha de rotas, permitindo flexibilidade para

diferentes momentos e requisitos. No entanto, o problema torna-se mais desafiador para ser resolvido, pois requer do MOEA uma habilidade para refinar a busca a fim de conseguir encontrar mais pontos ótimos.

Capítulo 6

Resultados Experimentais

6.1 Instâncias e Métricas

O desempenho das heurísticas propostas (h2, h3 e h4) foi avaliado experimentalmente em cada formulação do PRM sobre 8 instâncias retiradas da literatura de Roteamento e Steiner, listadas na Tab. 6.1. Em suas definições originais, as redes 0, 1 e 2 têm valores de custo e *delay*, ao passo que as redes 3 a 7 possuem apenas valores de custo. Assim, foram gerados valores aleatórios de *delay* para as redes 3 a 7, distribuídos uniformemente no intervalo $[1, 15]$. Além disso, como também são necessários os pesos capacidade e tráfego corrente, valores aleatórios foram gerados no intervalo $[1000, 1500]$ Kbps e $[0, 500]$ Kbps, respectivamente (neste caso, para as redes 0 a 7). Para as formulações do PRFM, foi definido um fluxo $\phi = 300$ Kbps para cada instância. Em resumo:

- **PRMMo e PRMM**: cada aresta e possui um par de pesos $(c(e), d(e))$; no caso das redes 3 a 7, valores de $d(e)$ foram gerados aleatoriamente, pois não constavam da sua definição original.
- **PRFM**: cada aresta e possui uma tupla de pesos $(c(e), d(e), z(e), t(e))$; para todas as redes, os valores de $z(e)$ e $t(e)$ foram gerados aleatoriamente, pela mesma razão anterior.

Tabela 6.1: Características das instâncias utilizadas neste trabalho.

Rede	$ V $	$ E $	$ D $	Origem
0	15	44	5	[60], [45]
1	18	50	5	[60], [45]
2	33	106	10	[54], [10]
3	50	126	12	[5]
4	75	188	19	[5]
5	75	188	38	[5]
6	75	300	13	[5]
7	100	250	17	[5]

De acordo com a formulação ou a técnica a ser enfatizada, um subconjunto dessas redes foi selecionado, para que a análise dos dados pudesse ficar mais focada. Dois tipos principais de experimentos foram feitos neste trabalho: o primeiro avalia o desempenho de heurísticas, MOEAs e o Cruzamento pela Vizinhança por meio de métricas de convergência e diversidade; o segundo grupo de experimentos compara o melhor modelo obtido nas etapas anteriores com algoritmos conhecidos da literatura (Dijkstra [14] e Takahashi-Matsuyama [51]), com o objetivo de destacar as vantagens ao se utilizar uma abordagem multiobjetivo no PRM.

Por fins de clareza, os resultados com os filtros propostos foram deslocados para o Apêndice B, onde mostra-se que o filtro f_{tm} levou a resultados melhores que os demais filtros em formulações com 3 e 4 objetivos no PRFM, utilizando o SPEA2 e a heurística $h4$.

Ao realizar experimentos para avaliação de desempenho de modelos em MOPs nesse primeiro grupo de experimentos, duas metas principais em otimização multiobjetivo devem ser consideradas: convergência e diversidade. A primeira meta avalia a qualidade da aproximação de um conjunto de soluções P retornadas por algum algoritmo em relação ao conjunto ótimo de Pareto (P^*) de referência. Por outro lado, as soluções também devem estar bem distribuídas ao longo do espaço dos objetivos, o que nos leva a indicadores como a uniformidade de P e qual cobertura deste espaço é atingida (em termos de soluções extremas, por exemplo). Assim, várias métricas foram descritas na literatura para avaliar aspectos diferentes dessas duas metas, a fim de fornecer um conjunto de valores que possa ser utilizado na tomada de decisão.

A convergência de um conjunto de soluções P retornadas por algum método evolutivo para um conjunto de referência P^* pode ser avaliada por várias formas, tais como a contagem do número de soluções em P , o número de soluções comuns aos dois conjuntos, a distância (p.ex., Euclidiana) entre os dois conjuntos, etc. Neste trabalho, as seguintes métricas para avaliar convergência foram adotadas:

- **Error Rate (er)**: corresponde ao percentual de soluções de P que não são soluções Pareto-ótimas [53]. Em símbolos:

$$er = \frac{\sum_{i=1}^{|P|} e_i}{|P|}$$

onde $e_i = 1$ se a solução $i \in P$ for dominada por alguma solução de P^* ; caso contrário, $e_i = 0$. Quanto maior o er , pior será a convergência relativa, i.e., o percentual de soluções de P que não são soluções Pareto-ótimas. Logo, esta métrica deve ser *minimizada*.

- **Generational Distance (gd)**: corresponde à distância média entre os pontos de

P e P^* [53]. Em símbolos:

$$gd = \frac{\sqrt{\sum_{i=1}^{|P|} d_i^2}}{|P|}$$

onde $d_i = \min_{j \in P^*} d(i, j)$, i.e., para cada ponto $i \in P$, encontramos o ponto $j \in P^*$ que lhe é mais próximo em termos de distância Euclidiana. Quanto menor a gd , mais próximo de P^* estará o conjunto P . Logo, esta métrica deve ser *minimizada*.

- **Pareto Subset** (ps): corresponde ao número absoluto de soluções de P que são, de fato, soluções Pareto-ótimas:

$$ps = (1 - er)|P|$$

Esta métrica complementa a informação fornecida por er , porém na direção oposta (número de soluções que *são* Pareto-ótimas, ao invés do percentual de soluções que *não são* Pareto-ótimas) e em um sentido absoluto (er fornece uma medida relativa). Esta métrica deve ser *maximizada*.

Para avaliação da diversidade das soluções, aspectos como o espalhamento das soluções e a cobertura do espaço de objetivos podem ser observados. As seguintes métricas foram utilizadas:

Spread (Δ): tem o objetivo de avaliar quão uniformemente estão distribuídas as soluções de P ao longo do espaço de objetivos. Δ deve ser *minimizada*, alcançando o valor zero quando as soluções de P estiverem uniformemente distribuídas. Seguindo a definição de Deb *et al.* [18], Δ é definido como:

$$\Delta = \frac{\sum_{i=1}^M d_i^e + \sum_{i=1}^{|P|} |d_i - \bar{d}|}{\sum_{i=1}^M d_i^e + |P|\bar{d}}$$

onde d_i corresponde à distância Euclidiana entre soluções consecutivas de P ; $\bar{d}$ é a média de tais distâncias; d_i^e corresponde à distância entre as soluções extremas de P e P^* no i -ésimo objetivo.

Maximum Spread ($m3$): corresponde à soma da extensão máxima obtida em cada objetivo por um conjunto P [63]. Formalmente,

$$m3 = \sqrt{\sum_{i=1}^M (\max f_i - \min f_i)^2}$$

Por fim, para comparar diretamente dois conjuntos de soluções P e Q em termos de dominância, a seguinte métrica [63] foi utilizada:

Two Set Coverage (sc):

$$sc(P, Q) = \frac{|\{j \in Q; \exists i \in P : i \prec j\}|}{|Q|}$$

Esta métrica não é simétrica para dois conjuntos P e Q , logo, é necessário o cálculo de $sc(P, Q)$ e $sc(Q, P)$. Um valor $sc(P, Q) = 1$ indica que todas as soluções de Q são dominadas por pelo menos uma solução de P , ao passo que $sc(P, Q) = 0$ indicaria que nenhuma solução de Q seria dominada por alguma de P .

6.2 Formulação Mono-objetivo (PRMMo)

Para avaliar as heurísticas propostas na formulação mono-objetivo do PRM, é necessário um valor de d_{max} . Mais especificamente, é possível definir vários valores para d_{max} em cada rede, de maneira que para cada um, irá corresponder um custo ótimo ($custo^*$), como mostrado na Tab. 6.2. Tal $custo^*$ é exatamente o custo ótimo que qualquer solução pode atingir ao atender um d_{max} específico em uma dada rede. Assim, foram definidos dois valores para esta restrição em cada rede, como mostra a tabela.

Os parâmetros do AG neste caso foram definidos como: probabilidade de mutação igual a 10%, taxa de desconexão de vértices na mutação de 20%, $Tp = 30$, $Ng = 50$ para as redes 0 e 1, e $Tp = 60$ e $Ng = 100$ para as demais redes, onde Tp e Ng denotam o tamanho da população e o número de gerações, respectivamente. Cada experimento foi executado 30 vezes.

Tabela 6.2: Valores de custo ótimo ($custo^*$) sujeitos a restrição de d_{max} no PRMMo. Os valores de $custo^*$ em negrito correspondem aos menores valores de custo possíveis em cada rede, independentemente de quão grande seja d_{max} .

Rede	d_{max}	$custo^*$
0	24	69
	20	90
1	12	71
	8	92
2	38	169
	21	262
3	77	83
	46	125
4	61	104
	38	127
5	84	220
	52	227

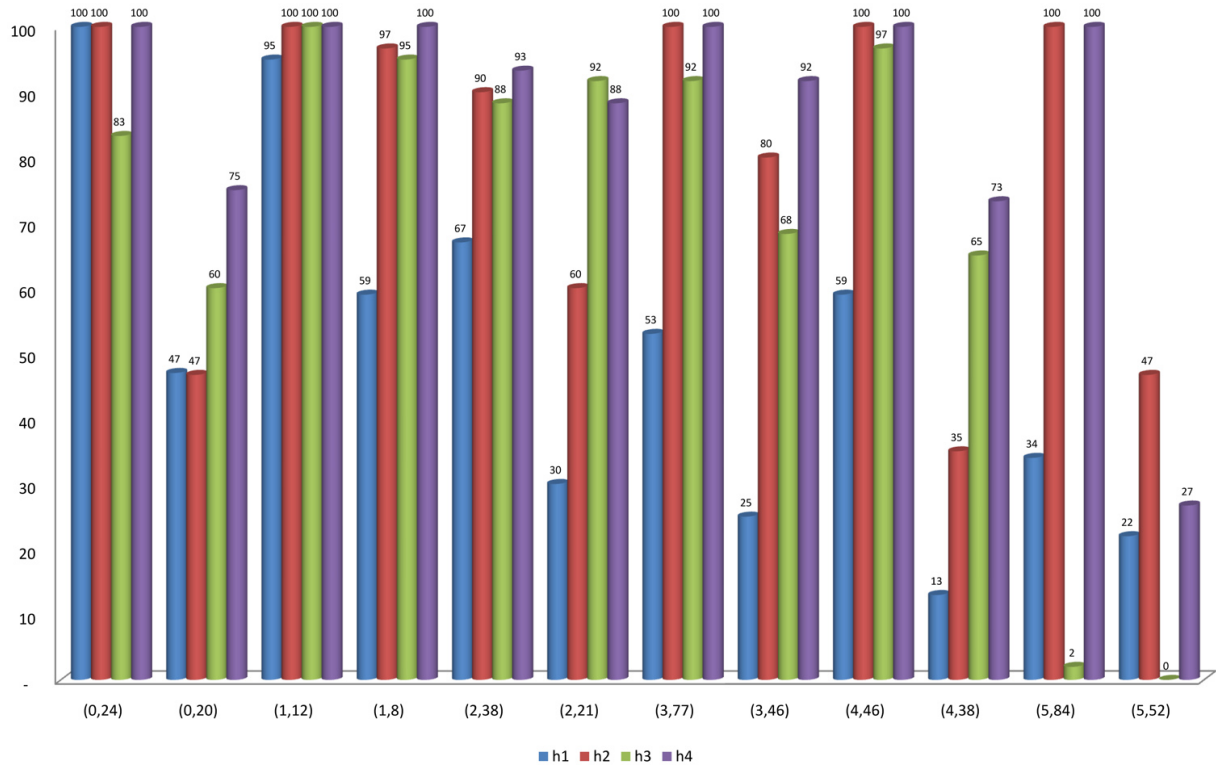


Figura 6.1: Percentual de convergência para cada heurística no PRMMo. Cada grupo de quatro experimentos corresponde aos resultados obtidos por cada heurística em cada instância ($rede, d_{max}$).

Os resultados de convergência obtidos estão ilustrados na Fig. 6.1. Analisando-os, a heurística h1 de Ravikumar & Bajpai [45] retornou os piores resultados em quase todos os cenários. Apenas nas redes menores o efeito das violações não prejudicou a convergência do AG, levando a heurística h1 a atingir resultados satisfatórios. Este fato era esperado, de uma certa forma, pois nessas redes menores a quantidade média de indivíduos inválidos produzidos foi bem menor do que em uma rede maior (como a rede 2, p.ex.), como mostrado na Sec. 4.4.3. A comparação mais próxima a h1 seria h2, a qual possui a mesma idéia central, porém preserva a validade das soluções, alcançando resultados muito superiores nas redes maiores, quando usada no lugar de h1.

Por outro lado, ao observar os resultados das heurísticas h2 e h3, em alguns cenários há uma alternância entre qual das duas superou a outra. Isso evidencia que o “formato” da solução ótima depende do valor de d_{max} utilizado, ou seja, dependendo do valor, fica mais “fácil” o trabalho de uma heurística mais determinística ou de uma mais aleatória, como é o caso de h2 e h3, respectivamente. Ainda nesse sentido, é possível observar que uma mesma heurística pode dar um bom resultado em um d_{max} e ruim em outro numa mesma rede (p.ex., h2 na rede 0 com $d_{max} = 24$ e $d_{max} = 20$). Este fato fica mais reforçado ao constatar as taxas de convergência obtidas com h4: ela obtém resultados melhores que as outras três heurísticas na grande maioria dos cenários. Assim, pode-se intuir que h4 tenta construir uma população que combina “formatos” de indivíduos gerados por h2 e

por h3, o que se mostrou benéfico para melhorar a convergência nos cenários testados.

A formulação resolvida no PRMMo corresponde ao chamado *Constrained Steiner Tree Problem*, devido à presença de d_{max} , como discutido na Sec. 3.2. Dentre os dois valores de d_{max} em cada rede, um deles corresponde ao *delay* que permite solução de custo ótimo independente da restrição de *delay* aplicada. Ou seja, se aplicássemos um algoritmo exato para encontrar o custo ótimo de qualquer uma das redes discutidas (sem considerar nenhuma restrição de *delay*), os menores valores possíveis de custo seriam exatamente aqueles mostrados em negrito na Tab. 6.2. Isso significa que nesses cenários, resolver o PRMMo também significa resolver o clássico Problema da Árvore de Steiner, onde o AG+h4 se saiu bem em todos os casos - são seis cenários - alcançando uma média geral próxima a 90%, como mostra a Tab. 6.3.

Portanto, de um modo geral, as heurísticas propostas apresentaram um desempenho melhor que a heurística original, como mostra a Tab. 6.3, que sumariza os percentuais médios obtidos a partir da Fig. 6.1. Como discutido, o método combinado (h4) apresentou a melhor convergência geral. Além disso, as taxas de convergência obtidas com as novas heurísticas (h2, h3 e h4) também superaram os resultados apresentados em [39, 45, 54, 60], cujas médias para as redes 0 e 1 estiveram situadas em torno de 62.5% contra 93.75% obtida por h4.

Tabela 6.3: Convergência média geral obtida por cada heurística no PRMMo.

Heurística	Convergência média
<i>h1</i>	50.3%
<i>h2</i>	79.6%
<i>h3</i>	70.2%
<i>h4</i>	87.4%

6.3 Formulação PRMM (dois objetivos)

Na Tab. 6.4 estão apresentadas as cardinalidades dos conjuntos P^* referentes às instâncias avaliadas nos cinco problemas (P1 a P5, definidos na Sec. 3.1.2) correspondentes à formulação PRMM. Os conjuntos de tabelas A.1, A.2 (NSGA-II), A.3 e A.4 (SPEA), 6.5, 6.6 e 6.7 (SPEA2) apresentam os resultados obtidos com cada heurística em cada problema/instância. Por razões de clareza, as tabelas correspondentes aos resultados do NSGA-II e SPEA foram deslocadas para o Apêndice A.

As métricas usadas para avaliar os resultados obtidos nesta formulação foram duas para convergência (*error rate* e *generational distance*) e uma para diversidade (*spread*). Os parâmetros do AG nesta formulação foram: $Tp = 30$ e $Ng = 50$ para as redes 0 e 1, e $Tp = 60$ e $Ng = 100$ para as demais redes.

Tabela 6.4: Cardinalidades dos conjuntos Pareto-ótimos P^* para cada instância no PRMM.

Rede	Cardinalidade de P^*				
	P1	P2	P3	P4	P5
0	3	6	7	4	3
1	2	5	6	2	2
2	7	11	29	24	5
3	6	12	20	9	3
4	7	8	13	9	3
5	7	21	19	6	5

6.3.1 Análise geral das métricas

Inicialmente, pode-se observar que os resultados das tabelas citadas indicam que as redes 0 e 1 parecem ser instâncias relativamente fáceis para os três MOEAs nas métricas er e gd , uma vez que na grande maioria dos casos seus valores estiveram próximos ou iguais a zero. Por outro lado, a métrica Δ apresentou valores maiores que zero em vários casos, mesmo quando a er é zero. Essa situação pode ocorrer devido ao formato da fronteira de P^* no espaço dos objetivos, que pode não ser uniformemente distribuída, o que já foi constatado por outros autores, como Veldhuizen [53]. Já nas demais redes (2 a 5), os resultados variam conforme o problema e a heurística.

Especificamente, ao comparar os resultados do NSGA-II, SPEA e SPEA2, pode-se notar que, de um modo geral, o SPEA2 provê melhorias para as heurísticas na maioria dos cenários, tanto nas estimativas por médias, quanto naquelas por intervalos de confiança, fornecendo intervalos mais curtos. Em alguns casos, é possível observar essa melhoria de forma mais clara, como por exemplo, nas redes 2 (P1 e P4) e 5 (P1, P3 e P4). Considerando todos os cenários, estão apresentadas na Tab. 6.8 as médias gerais para os três MOEAs em cada heurística, reforçando a observação de que o SPEA2 melhora todas as heurísticas em todas as três métricas. Por sua vez, o NSGA-II obtém resultados melhores que o SPEA.

6.3.2 Análise das métricas com o SPEA2

Convergência

Analisando os resultados referentes ao SPEA2, primeiramente concentramos nossa atenção na métrica de convergência er (Tab. 6.5), que deve ser minimizada, pois quando $er = 0$, então $P \subseteq P^*$, i.e., toda solução de P (conjunto de soluções retornadas pelo AG) é também uma solução Pareto-ótima. Por outro lado, se er é alto, P conterá muitas soluções dominadas por pelo menos uma solução de P^* . Assim, pode-se notar que ao comparar os valores de er obtidos com h2 e h3, h2 obtém melhores resultados na maioria dos casos. Os intervalos de confiança (ICs) para h2 são também menores do que os de h3, o que é refletido por menores desvios-padrão. Estes fatos indicam que um algoritmo

de menor caminho para religar subárvores no cruzamento/mutação no PRMM parece ser mais adequado do que um algoritmo que trabalha aleatoriamente nessas religiões.

Entretanto, existe uma alternância de bons resultados entre h2 e h3, como por exemplo, em P3 e P4, de forma análoga ao que foi observado na formulação mono-objetivo (PRMMo). Isto significa que uma combinação de indivíduos produzidos por h2 e h3 poderia melhorar tais resultados, que é exatamente o que acontece quando a heurística h4 é usada: ela melhora a *er* de h2 e h3 na grande maioria dos casos, especialmente quando a escolha individual de h2 ou h3 levaria a resultados bons em um cenário, e ruins em outros. Mais especificamente, h4 obtém melhores valores para esta métrica na maioria das instâncias/MOPs, especialmente em P1, P4 e P5. Também podemos citar os ganhos expressivos obtidos nas redes 2 e 5, nos problemas de P1 a P5.

A utilização de h4 forneceu uma *er* menor que 10% em todos os casos, exceto na rede 2 em P1 e P3, o que não a impediu de superar as outras heurísticas mesmo nesse caso. Na média geral, h4 obteve *er* = 2.9% contra 7.1% e 13.5% de h2 e h3, respectivamente, como mostrado na Tab. 6.8. Isso indica que, considerando todas as redes, em torno de 97% das soluções calculadas por h4 foram, de fato, Pareto-ótimas.

Dentre todas as heurísticas, h4 foi a que obteve o maior número de ICs curtos, os quais estiveram situados abaixo de 3.5%. Por outro lado, os demais métodos obtiveram ICs maiores de um modo geral, em alguns casos atingindo valores próximos ou maiores que 10%. Essa característica de menor dispersão apresentada por h4 é importante, pois indica uma maior estabilidade do método, fruto de um desvio-padrão reduzido.

Por sua vez, a métrica de convergência *gd* fornece uma medida de proximidade entre os conjuntos P e P^* no espaço dos objetivos, i.e., entre as fronteiras não-dominadas de tais conjuntos (alguns autores chamam a imagem de P e P^* no espaço dos objetivos de FP e FP^* , respectivamente). Quando uma solução $x \in P$ também está em P^* , então $dist(x, P^*) = 0$; por outro lado, esta distância será diferente de 0 quando $x \notin P^*$. Ou seja, esta métrica pode ser usada para refletir a proximidade (em termos de distância Euclidiana) das soluções dominadas de P ao conjunto P^* . Deste modo, quando $gd = 0$ não for possível de ser obtida, mas sim um valor próximo a esse, significa que as soluções de P têm uma boa convergência para P^* .

Os resultados desta segunda métrica de convergência (Tab. 6.6) indicam que dentre as heurísticas, h4 foi a que retornou conjuntos P mais próximos de P^* no espaço dos objetivos, de um modo geral. Como consequência, ao observarmos os valores de *gd* como um todo, h4 obteve valores normalizados menores; na média geral, os valores obtidos por h2 e h3 foram em torno de 2 e 7 vezes maiores que os de h4 nessa métrica, respectivamente. Assim como em *er*, h4 obteve os melhores ICs, especificamente com valores pequenos e próximos de 0, ao contrário de h2 e h3, que obtiveram vários ICs maiores.

Tabela 6.5: Médias e intervalos de confiança de 95% para a métrica *Error Rate* (*er*) utilizando o SPEA2.

Rede	h	P1	P2	P3	P4	P5
0	2	0.0 $\pm$ 0.0	0.6 $\pm$ 0.9	3.5 $\pm$ 2.7	2.5 $\pm$ 2.3	0.0 $\pm$ 0.0
	3	1.7 $\pm$ 2.8	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	2.2 $\pm$ 2.5
	4	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
1	2	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	1.7 $\pm$ 2.8
	3	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
	4	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
2	2	23.3 $\pm$ 4.9	0.0 $\pm$ 0.0	15.8 $\pm$ 1.4	9.3 $\pm$ 1.9	0.0 $\pm$ 0.0
	3	57.9 $\pm$ 3.1	20.9 $\pm$ 2.2	26.6 $\pm$ 3.7	23.3 $\pm$ 6.1	8.6 $\pm$ 4.4
	4	19.0 $\pm$ 3.3	0.0 $\pm$ 0.0	12.5 $\pm$ 1.4	5.2 $\pm$ 1.9	0.0 $\pm$ 0.0
3	2	0.6 $\pm$ 0.9	0.0 $\pm$ 0.0	10.9 $\pm$ 2.6	4.1 $\pm$ 2.2	0.0 $\pm$ 0.0
	3	23.8 $\pm$ 4.9	1.9 $\pm$ 1.3	1.5 $\pm$ 0.9	25.3 $\pm$ 4.6	20.0 $\pm$ 5.0
	4	1.7 $\pm$ 1.5	0.0 $\pm$ 0.0	0.5 $\pm$ 0.5	2.6 $\pm$ 2.1	0.0 $\pm$ 0.0
4	2	5.2 $\pm$ 2.1	0.0 $\pm$ 0.0	4.6 $\pm$ 1.8	6.1 $\pm$ 2.5	0.0 $\pm$ 0.0
	3	2.9 $\pm$ 1.8	0.0 $\pm$ 0.0	1.9 $\pm$ 1.2	10.2 $\pm$ 2.7	8.9 $\pm$ 5.6
	4	5.7 $\pm$ 2.1	0.0 $\pm$ 0.0	2.1 $\pm$ 1.1	5.8 $\pm$ 2.5	0.0 $\pm$ 0.0
5	2	24.8 $\pm$ 2.5	3.4 $\pm$ 1.0	27.2 $\pm$ 3.1	58.6 $\pm$ 4.8	11.3 $\pm$ 3.0
	3	37.7 $\pm$ 9.3	18.8 $\pm$ 5.7	26.7 $\pm$ 4.6	53.2 $\pm$ 8.4	30.6 $\pm$ 11.6
	4	7.7 $\pm$ 3.5	2.8 $\pm$ 1.2	10.0 $\pm$ 2.8	10.1 $\pm$ 2.5	0.0 $\pm$ 0.0

Tabela 6.6: Médias e intervalos de confiança de 95% para a métrica *Generational Distance* (*gd*) utilizando o SPEA2. Os valores estão multiplicados por 10^3 .

Rede	h	P1	P2	P3	P4	P5
0	2	0.0 $\pm$ 0.0	0.2 $\pm$ 0.3	1.3 $\pm$ 1.0	1.1 $\pm$ 1.1	0.0 $\pm$ 0.0
	3	2.0 $\pm$ 3.2	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	1.6 $\pm$ 1.9
	4	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
1	2	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	2.1 $\pm$ 3.5
	3	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
	4	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
2	2	4.1 $\pm$ 1.0	0.0 $\pm$ 0.0	1.0 $\pm$ 0.1	2.2 $\pm$ 0.4	0.0 $\pm$ 0.0
	3	12.7 $\pm$ 0.7	6.8 $\pm$ 0.6	1.8 $\pm$ 0.3	4.7 $\pm$ 1.0	2.1 $\pm$ 1.1
	4	3.2 $\pm$ 0.9	0.0 $\pm$ 0.0	0.8 $\pm$ 0.1	1.1 $\pm$ 0.4	0.0 $\pm$ 0.0
3	2	0.0 $\pm$ 0.1	0.0 $\pm$ 0.0	0.6 $\pm$ 0.1	0.6 $\pm$ 0.3	0.0 $\pm$ 0.0
	3	3.8 $\pm$ 1.4	0.3 $\pm$ 0.2	0.1 $\pm$ 0.1	2.8 $\pm$ 0.4	2.6 $\pm$ 1.0
	4	0.2 $\pm$ 0.2	0.0 $\pm$ 0.0	0.1 $\pm$ 0.0	0.3 $\pm$ 0.2	0.0 $\pm$ 0.0
4	2	0.9 $\pm$ 0.4	0.0 $\pm$ 0.0	0.5 $\pm$ 0.2	0.8 $\pm$ 0.3	0.0 $\pm$ 0.0
	3	0.5 $\pm$ 0.3	0.0 $\pm$ 0.0	0.4 $\pm$ 0.2	1.0 $\pm$ 0.3	0.7 $\pm$ 0.4
	4	1.1 $\pm$ 0.4	0.0 $\pm$ 0.0	0.4 $\pm$ 0.2	0.8 $\pm$ 0.4	0.0 $\pm$ 0.0
5	2	0.8 $\pm$ 0.1	0.1 $\pm$ 0.0	1.0 $\pm$ 0.1	1.3 $\pm$ 0.1	0.5 $\pm$ 0.1
	3	3.8 $\pm$ 1.3	0.7 $\pm$ 0.2	1.9 $\pm$ 0.5	12.3 $\pm$ 3.3	1.5 $\pm$ 0.7
	4	0.3 $\pm$ 0.1	0.1 $\pm$ 0.1	0.5 $\pm$ 0.2	0.5 $\pm$ 0.1	0.0 $\pm$ 0.0

Diversidade

A métrica *spread* (Δ) é utilizada para indicar quão uniformemente distribuído está um conjunto de soluções ao longo do espaço dos objetivos. Convém mencionar que à medida

que P se aproxima de P^* , e mais especificamente quando P for o próprio P^* , então Δ_P irá tender a Δ_{P^*} . Por essa razão, o valor de Δ_P pode não ser zero mesmo quando $P = P^*$, pois esta métrica depende do formato da fronteira de P^* , cujas soluções podem não estar uniformemente distribuídas. Assim, sem uma análise do tamanho e do formato do conjunto P^* em cada MOP/instância, não se pode dizer que um MOEA/heurística falhou ao retornar um conjunto P com $\Delta_P > 0$.

Com base nos resultados da Tab. 6.7, pode-se notar que este fato ocorreu nitidamente em alguns cenários, tais como nas redes 0 em P1, e nas redes 2, 3 e 4 em P2, onde h4 atingiu $er = 0$ com $\Delta > 0$. Observando as cardinalidades dos conjuntos P e P^* , foi possível constatar que não somente $P \subseteq P^*$, mas como $P = P^*$, ou seja, de fato as soluções do Pareto-ótimo não estavam uniformemente distribuídas. Uma observação mais geral irá mostrar que, na verdade, a grande maioria dos conjuntos P^* não estão uniformemente distribuídos, pois em muitos casos h4 obteve uma er próxima ou igual a zero com um $\Delta > 0$, novamente tomando o cuidado ao verificar se $|P| = |P^*|$ (tais valores foram calculados mas não foram mostrados nas tabelas de resultados pois os conjuntos Pareto são relativamente pequenos no PRMM). As exceções foram somente nos casos em que P^* era constituído por apenas duas soluções, onde de fato observou-se $\Delta = 0$. Por fim, h4 tende a “acompanhar” a heurística que obtém melhor uniformidade de soluções, às vezes conseguindo valores melhores. Na média geral (Tab. 6.8), h4 forneceu soluções tão bem distribuídas quanto h2, porém mais bem distribuídas do que as de h3.

Tabela 6.7: Médias e intervalos de confiança de 95% para a métrica *Spread* (Δ) utilizando o SPEA2. Os valores estão multiplicados por 10.

Rede	h	P1	P2	P3	P4	P5
0	2	1.3 $\pm$ 0.0	3.9 $\pm$ 0.0	3.2 $\pm$ 0.1	3.4 $\pm$ 0.2	0.7 $\pm$ 0.0
	3	1.3 $\pm$ 0.1	3.9 $\pm$ 0.1	3.0 $\pm$ 0.0	3.2 $\pm$ 0.0	0.8 $\pm$ 0.2
	4	1.3 $\pm$ 0.0	3.9 $\pm$ 0.0	3.0 $\pm$ 0.0	3.2 $\pm$ 0.0	0.7 $\pm$ 0.0
1	2	0.0 $\pm$ 0.0	1.8 $\pm$ 0.0	3.5 $\pm$ 0.2	0.0 $\pm$ 0.0	0.1 $\pm$ 0.1
	3	0.0 $\pm$ 0.0	1.8 $\pm$ 0.0	3.8 $\pm$ 0.1	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
	4	0.0 $\pm$ 0.0	1.8 $\pm$ 0.0	3.8 $\pm$ 0.1	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
2	2	0.6 $\pm$ 0.1	4.2 $\pm$ 0.0	4.8 $\pm$ 0.1	3.2 $\pm$ 0.2	1.0 $\pm$ 0.0
	3	1.7 $\pm$ 0.2	4.8 $\pm$ 0.1	5.3 $\pm$ 0.2	3.5 $\pm$ 0.2	4.5 $\pm$ 0.1
	4	0.7 $\pm$ 0.1	4.2 $\pm$ 0.0	4.8 $\pm$ 0.1	3.3 $\pm$ 0.2	1.0 $\pm$ 0.0
3	2	2.9 $\pm$ 0.0	2.5 $\pm$ 0.0	5.7 $\pm$ 0.1	6.8 $\pm$ 0.1	0.8 $\pm$ 0.0
	3	3.9 $\pm$ 0.2	3.2 $\pm$ 0.3	5.4 $\pm$ 0.1	6.9 $\pm$ 0.1	1.6 $\pm$ 0.2
	4	2.9 $\pm$ 0.0	2.5 $\pm$ 0.0	5.6 $\pm$ 0.0	7.0 $\pm$ 0.0	0.8 $\pm$ 0.0
4	2	2.8 $\pm$ 0.1	1.7 $\pm$ 0.0	7.3 $\pm$ 0.0	3.2 $\pm$ 0.1	4.6 $\pm$ 0.2
	3	3.0 $\pm$ 0.2	2.0 $\pm$ 0.2	6.8 $\pm$ 0.1	2.4 $\pm$ 0.2	6.1 $\pm$ 0.4
	4	2.8 $\pm$ 0.1	1.7 $\pm$ 0.0	7.3 $\pm$ 0.1	3.2 $\pm$ 0.1	5.0 $\pm$ 0.4
5	2	8.0 $\pm$ 0.0	4.4 $\pm$ 0.2	7.8 $\pm$ 0.1	8.6 $\pm$ 0.1	2.7 $\pm$ 0.1
	3	7.1 $\pm$ 0.1	5.3 $\pm$ 0.2	7.5 $\pm$ 0.2	7.2 $\pm$ 0.2	3.1 $\pm$ 0.1
	4	7.8 $\pm$ 0.1	4.6 $\pm$ 0.1	8.0 $\pm$ 0.1	8.7 $\pm$ 0.1	2.5 $\pm$ 0.0

Tabela 6.8: Médias gerais para o NSGA-II, SPEA e SPEA2 nas métricas er , gd e Δ no PRMM. Os valores de gd estão multiplicados por 10^3 e os de Δ por 10.

h	Error Rate			Generational Distance			Spread		
	NSGA-II	SPEA	SPEA2	NSGA-II	SPEA	SPEA2	NSGA-II	SPEA	SPEA2
2	13.6%	16.3%	7.1%	4.0	4.0	0.6	3.7	3.6	3.4
3	15.3%	24.9%	13.5%	3.0	5.0	2.1	4.0	3.8	3.6
4	5.1%	8.5%	2.9%	1.0	1.0	0.3	3.5	3.4	3.4

6.3.3 Testes de hipóteses

Uma vez que os experimentos realizados neste trabalho consistem de 30 execuções, podemos realizar testes de hipóteses (paramétricos ou não-paramétricos, p.ex.) para estimar um valor populacional (média, p.ex.) a partir da avaliação de uma amostra. Neste caso, uma amostra corresponde a um conjunto de execuções em uma dada configuração (h4+SPEA2, p.ex.). Como queremos comparar o desempenho de vários métodos (heurísticas, MOEAs, etc.), então utilizamos um teste de hipóteses para inferir se deve existir diferença significativa entre as médias populacionais μ_1 e μ_2 a partir da avaliação de suas médias amostrais $\bar{x}_1$ e $\bar{x}_2$ e seus respectivos desvios s_1 e s_2 , considerando um dado nível de significância α . Neste trabalho, os parâmetros populacionais a serem inferidos são as métricas utilizadas para avaliar convergência e diversidade, quando utilizada uma heurística/MOEA em algum cenário (MOP/instância). Assim, as seguintes hipóteses foram formuladas:

$$H_0: \mu_{h4} = \mu_{hx}.$$

$$H_1: \mu_{h4} \neq \mu_{hx}.$$

onde μ pode corresponder a qualquer uma das métricas (er , gd , etc.) e hx à heurística h2 ou h3. O nível de significância escolhido foi $\alpha = 10\%$ bicaudal, o que resulta num nível de confiança de 90%. Convém observar que o Teorema do Limite Central afirma que para amostras com tamanho igual ou superior a 30, extraídas de uma população caracterizada por uma distribuição de probabilidade arbitrária, a população de médias amostrais poderá ser aproximada por uma distribuição Normal. Logo, o teste paramétrico acima poderá ser feito utilizando valores conhecidos para a variável padronizada z , considerando $\alpha = 10\%$ (5% em cada cauda). O tamanho da amostra utilizado neste trabalho é 30, correspondendo ao número de vezes que o AG é executado com diferentes sementes, para obter as médias de cada métrica nos cenários avaliados.

A partir das hipóteses definidas, quando H_1 é aceita, uma das duas conclusões abaixo pode ocorrer, dependendo do valor da estatística de teste (z_t) em relação ao valor $z_{\alpha/2}$ da região crítica, a saber:

- $z_t < -z_{\alpha/2}$: implica que $\mu_{h4} < \mu_{hx}$.
- $z_t > z_{\alpha/2}$: implica que $\mu_{h4} > \mu_{hx}$.

Assim, é possível saber especificamente qual desigualdade será inferida a partir da rejeição da hipótese nula. Como o objetivo do teste é estimar se há melhoria significativa de h_4 em relação às demais, foram escolhidos os resultados com o SPEA2, pois este MOEA superou o NSGA-II e SPEA, de um modo geral.

Os resultados estão apresentados nas tabelas 6.9 e 6.10, onde o sinal “=” significa que a hipótese nula (H_0) foi aceita, ou seja, não foi evidenciada diferença significativa entre os métodos comparados. Por outro lado, os sinais “<” e “>” representam a aceitação da hipótese alternativa (H_1), ou seja, existe evidência de que $\mu_{h_4} < \mu_{h_x}$ e $\mu_{h_x} < \mu_{h_4}$, respectivamente.

Tabela 6.9: Testes de hipóteses para evidenciar diferença estatisticamente significativa entre as médias de h_4 e h_3 , considerando um nível de significância $\alpha = 10\%$ bicaudal.

Rede	Error Rate					Generational Distance					Spread				
	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5
0	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=
1	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=
2	<	<	<	<	<	<	<	<	<	<	<	<	<	=	<
3	<	<	=	<	<	<	<	=	<	<	<	<	>	=	<
4	=	=	=	<	<	>	=	=	=	<	<	<	>	>	<
5	<	<	<	<	<	<	<	<	<	<	<	<	>	>	<

Nas comparações entre h_4 e h_3 mostradas na Tab. 6.9, os testes para *er* indicaram que a melhoria (redução) fornecida por h_4 foi significativa na maioria dos casos (16/30); nos outros 14 casos, não houve diferenças significativas entre h_3 e h_4 . Na métrica *gd*, resultados similares foram obtidos, onde h_4 foi melhor em 15/30 casos, ao passo que h_3 foi melhor em apenas 1/30 casos. Na métrica Δ , apesar de h_3 ter atingido reduções significativas em 5/30 casos, h_4 ainda a superou, obtendo melhorias em 13/30 testes. Em todas estas métricas, os resultados foram indiferentes em termos estatísticos nas redes 0 e 1, consideradas instâncias relativamente fáceis para h_4 e h_3 .

Tabela 6.10: Testes de hipóteses para evidenciar diferença estatisticamente significativa entre as médias de h_4 e h_2 , considerando um nível de significância $\alpha = 10\%$ bicaudal.

Rede	Error Rate					Generational Distance					Spread				
	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5
0	=	=	<	<	=	=	=	<	<	=	=	=	<	<	=
1	=	=	=	=	=	=	=	=	=	=	=	=	>	=	=
2	=	=	<	<	=	=	=	<	<	=	=	=	=	=	=
3	=	=	<	=	=	=	=	<	=	=	=	=	=	>	=
4	=	=	<	=	=	=	=	=	=	=	=	=	=	=	>
5	<	=	<	<	<	<	=	<	<	<	<	=	>	=	<

Por sua vez, a comparação entre h_4 e h_2 (Tab. 6.10) mostrou uma disputa mais apertada entre ambas as heurísticas. Os resultados obtidos em *er* e *gd* foram similares, onde h_4 superou h_2 em 10/30 e 9/30 testes, respectivamente; nos demais testes não ocorreram diferenças. Já na métrica Δ , não foi possível ter uma distinção clara entre h_2 e h_4 , pois houve um empate (4/10 testes para cada heurística).

Um interessante ponto a ser observado é que as melhorias de h4 foram mais substanciais, do ponto vista estatístico, nas redes maiores (2 a 5). Isso mostra a sua habilidade em encontrar melhores soluções em espaços de busca maiores.

6.3.4 Considerações

As heurísticas que utilizam o algoritmo de Dijkstra nas reconexões (sempre - h2, ou por meio de um sorteio - h4) obtiveram melhores resultados na média geral, mostrando que esse algoritmo é importante nestas formulações de dois objetivos (PRMM). Por outro lado, a utilização de um algoritmo que trabalha aleatoriamente parece ajudar o AG em certos momentos, quando combinado com o Dijkstra, que é justamente a idéia de h4. Acreditamos que isso evidencie que a idéia de sempre buscar o menor caminho pode ser boa no PRMM, mas a mistura com um pouco mais de aleatoriedade pode dar mais diversificação ao AG e um melhor resultado, fugindo um pouco do determinismo de tal idéia.

Na maioria dos casos, pode-se notar que a heurística h4 “acompanha” a heurística que obtém os melhores resultados, muitas vezes sendo capaz de melhorá-los. Os resultados dos testes de hipóteses evidenciaram que a heurística h4 obteve melhorias significativas nas métricas *er*, *gd* e Δ em relação a h2 e h3, com exceção da comparação com h2 no Δ , caso em que não foram noticiadas diferenças significativas. Analisando as instâncias como um todo, no saldo geral em nenhuma métrica h4 obteve resultados piores que h2 e h3. Isso reforça estatisticamente os dados detalhados das tabelas 6.5, 6.6 e 6.7, evidenciando alguns resultados da configuração h4+SPEA2:

- (i) seus percentuais de soluções não-dominadas que são, de fato, Pareto-ótimas são maiores;
- (ii) suas soluções não-dominadas que não são Pareto-ótimas estão mais próximas de P^* no espaço dos objetivos; e
- (iii) as soluções retornadas estão distribuídas de forma mais uniforme ao longo do espaço dos objetivos.

6.4 Formulação PRFM (quatro objetivos)

Nesta formulação, para avaliação de convergência foram adotadas as métricas *er* e *gd* como antes, e a métrica *ps*, que fornece uma informação complementar a *er* num sentido de quantidades absolutas. Para avaliação de diversidade foi escolhida *m3* para dar um indicativo da extensão máxima alcançada nos objetivos por cada método. Outras métricas que são comumente usadas em problemas de dois objetivos, tais como o *spread* (Δ) e o *spacing* [48], podem ter uma interpretação mais difícil ou problemática em problemas com

mais de 2 objetivos, além de poderem necessitar de outras técnicas para serem calculados (diagramas de Voronoi para o cálculo do Spread [18]), como discutido por alguns autores (p.ex., Deb *et al.* em [19]).

Na Tab. 6.11 estão apresentadas as cardinalidades dos conjuntos P^* referentes às instâncias avaliadas na formulação PRFM de quatro objetivos (P6 da Sec. 3.1.2). Os parâmetros para o AG neste caso são: $Tp = 30, Ng = 50$ para as redes 0 e 1, e $Tp = 90, Ng = 100$ para as demais redes.

Tabela 6.11: Cardinalidades dos conjuntos Pareto-ótimos P^* para cada instância no PRFM de quatro objetivos (**P6**).

Rede	$ P^* $
0	27
1	7
2	122
3	40
4	72
5	60

6.4.1 Análise das métricas e testes de hipóteses

Observando os resultados mostrados nas tabelas 6.12 e 6.13, de um modo geral, como aconteceu na formulação anterior, o SPEA2 forneceu valores de *er* menores em comparação ao NSGA-II e SPEA. Por sua vez, o NSGA-II obteve resultados melhores que o SPEA observando os valores de médias gerais. Porém, a métrica *ps* no SPEA fornece um maior número absoluto de soluções ótimas do que no NSGA-II, especialmente em h4. Além disso, em vários cenários a escolha entre os dois MOEAs seria difícil, pois há de fato uma alternância entre qual dos dois supera o outro (p.ex., o NSGA-II foi melhor nas redes 2, 4 e 5, ao passo que o SPEA o foi nas redes 0 e 3). Por outro lado, o SPEA2 supera os dois MOEAs em todos os casos, exceto nas redes pequenas, onde, apesar disso, a sua *er* foi próxima a zero. O SPEA2 também apresentou valores de *er* menos dispersos, como pode ser visto pelos ICs, especialmente os de h4.

Em relação aos testes de hipóteses exibidos nas tabelas 6.14a e 6.14b, onde se compara o SPEA2 com o NSGA-II e com o SPEA, quando o resultado do teste for “<” ou “>” nas métricas a serem minimizadas (*er* e *gd*), significa que o SPEA2 resultou em diferenças significativas, para melhor ou para pior, respectivamente. Nas métricas a serem maximizadas (*ps* e *m3*), a lógica dos sinais de desigualdade é invertida. Assim, observando-se primeiramente os resultados referentes a *er* e *ps* (colunas 1 e 3, respectivamente), pode-se evidenciar estatisticamente as melhorias citadas que o SPEA2 provê, pois em sua maioria, os testes resultaram na aceitação da hipótese alternativa, especificamente indicando que o SPEA2 melhora estas duas métricas.

Já nos testes realizados para comparar h4 com h3 utilizando o SPEA2, a Tab. 6.15b mostra que apesar de os valores médios de *er* obtidos por h4 nas redes maiores terem sido melhores que os de h3, na verdade não foi possível concluir que tais mudanças foram tão significativas, pois em apenas um dos testes foi evidenciada uma diferença para melhor. Nas demais situações não foi possível evidenciar diferença estatística. Por outro ângulo, isso significa que h4 foi pelo menos tão bem quanto h3 em *er*. Já em *ps* houve uma vantagem maior. Por sua vez, na comparação de h4 com h2 (Tab. 6.15a), a melhoria foi mais significativa, tanto em *er* como em *ps*.

Passando para a análise da métrica *gd*, o SPEA2 continuou fornecendo melhores resultados que os demais MOEAs, como pode ser visto pelas médias gerais e pelos testes realizados (segunda coluna das tabelas 6.14a e 6.14b). Na comparação de médias gerais obtidas pelas heurísticas utilizando o SPEA2, diferentemente do que havia sido observado nos resultados de *er*, onde h4 foi melhor, os resultados referentes a *gd* indicaram que h3 foi melhor na média geral, o que foi bastante influenciado pelo resultado ruim obtido por h4 na rede 1 com SPEA2. Assim, testes de hipóteses foram realizados para evidenciar a significância destas comparações (segunda coluna das tabelas 6.15a e 6.15b), os quais indicaram que, na verdade, h4 superou não somente h2, mas como h3, na maioria dos cenários. Por sua vez, h3 não superou h4 em nenhum caso, havendo no melhor dos casos empatado. Já h2 obteve resultados piores que as duas tanto na análise individual das redes como na média.

Na análise de diversidade por *m3*, os três MOEAs obtiveram desempenho competitivo na média geral, especialmente a comparação entre SPEA e SPEA2. A heurística h3 foi superada por h2 e h4 neste quesito. Os testes de significância mostraram que h4 superou h3, mas não h2, onde empatou com esta praticamente. No entanto, os ICs de h4 são melhores que os de h2 na maioria dos casos.

6.4.2 Considerações

As estimativas pontuais, por intervalo e os testes de hipóteses indicaram que a combinação h4+SPEA2 foi melhor que as demais opções (h2, h3, SPEA e NSGA-II) nas métricas de convergência (*er*, *gd* e *ps*) na maioria dos cenários. Por outro lado, no quesito diversidade (*m3*), h4 foi bastante competitiva com h2, nos três MOEAs. Por sua vez, h3 foi competitiva com h4 em *er*, principalmente quando utilizado o SPEA2. Assim, contrabalanceando estes aspectos analisados, pode-se dizer que a combinação SPEA2+h4 seria a melhor escolha na formulação de quatro objetivos do PRFM, pois ela “supera” as fraquezas das demais heurísticas (h2 em *gd* e h3 em *m3*, por exemplo).

A avaliação experimental das heurísticas nesta formulação (PRFM) mostrou que é possível acreditar que a heurística que utiliza apenas o menor caminho nas reconexões (h2) pode reduzir a convergência quando um número maior de objetivos é usado. Nos

Tabela 6.12: Métricas de convergência (*er*, *gd* e *ps*) no PRFM (**P6**).

Rede	h	Error Rate (%)			Generational Distance ($\times 10^3$)			Pareto Subset		
		NSGA-II	SPEA	SPEA2	NSGA-II	SPEA	SPEA2	NSGA-II	SPEA	SPEA2
0	2	4.0 $\pm$ 1.0	2.4 $\pm$ 0.9	1.2 $\pm$ 0.6	3.5 $\pm$ 0.9	2.1 $\pm$ 0.8	1.2 $\pm$ 0.7	16.2 $\pm$ 0.4	21.2 $\pm$ 0.4	21.4 $\pm$ 0.5
	3	2.4 $\pm$ 1.1	3.4 $\pm$ 1.6	0.5 $\pm$ 0.4	1.9 $\pm$ 0.9	2.7 $\pm$ 1.0	0.7 $\pm$ 0.6	19.2 $\pm$ 0.5	24.7 $\pm$ 0.4	25.6 $\pm$ 0.3
	4	1.6 $\pm$ 0.7	0.1 $\pm$ 0.2	0.7 $\pm$ 0.5	1.3 $\pm$ 0.6	0.1 $\pm$ 0.1	0.5 $\pm$ 0.3	19.1 $\pm$ 0.3	25.2 $\pm$ 0.3	25.1 $\pm$ 0.3
1	2	7.6 $\pm$ 2.2	7.6 $\pm$ 2.2	4.8 $\pm$ 2.1	31 $\pm$ 11	32 $\pm$ 11	19 $\pm$ 9.3	6.0 $\pm$ 0.1	6.1 $\pm$ 0.1	6.4 $\pm$ 0.1
	3	0.0 $\pm$ 0.0	2.3 $\pm$ 1.8	0.5 $\pm$ 0.8	0.0 $\pm$ 0.0	5.2 $\pm$ 4.1	1.2 $\pm$ 2.0	7.0 $\pm$ 0.0	6.9 $\pm$ 0.1	7.0 $\pm$ 0.1
	4	0.0 $\pm$ 0.0	4.8 $\pm$ 2.8	1.9 $\pm$ 1.5	0.0 $\pm$ 0.0	12.3 $\pm$ 7.3	4.9 $\pm$ 3.8	7.0 $\pm$ 0.0	6.8 $\pm$ 0.1	6.9 $\pm$ 0.1
2	2	24.6 $\pm$ 1.2	24.5 $\pm$ 1.4	21.2 $\pm$ 1.6	2.7 $\pm$ 0.2	2.5 $\pm$ 0.1	2.1 $\pm$ 0.1	44.3 $\pm$ 0.7	49.5 $\pm$ 1.2	63.0 $\pm$ 1.1
	3	13.8 $\pm$ 1.4	19.1 $\pm$ 1.7	11.6 $\pm$ 1.3	2.1 $\pm$ 0.2	3.3 $\pm$ 0.2	1.7 $\pm$ 0.2	55.2 $\pm$ 1.3	52.7 $\pm$ 1.0	68.3 $\pm$ 1.9
	4	18.4 $\pm$ 1.5	15.4 $\pm$ 0.9	10.4 $\pm$ 1.1	2.2 $\pm$ 0.2	2.2 $\pm$ 0.2	1.3 $\pm$ 0.1	53.6 $\pm$ 1.1	70.8 $\pm$ 1.0	80.5 $\pm$ 0.9
3	2	31.0 $\pm$ 2.0	45.5 $\pm$ 2.4	24.0 $\pm$ 2.3	6.3 $\pm$ 0.6	8.5 $\pm$ 0.6	4.6 $\pm$ 0.5	19.2 $\pm$ 0.7	15.4 $\pm$ 0.8	25.0 $\pm$ 0.7
	3	7.9 $\pm$ 2.3	7.3 $\pm$ 1.6	6.7 $\pm$ 1.3	1.9 $\pm$ 0.5	3.5 $\pm$ 1.2	1.3 $\pm$ 0.5	21.3 $\pm$ 1.1	21.3 $\pm$ 0.5	28.1 $\pm$ 0.9
	4	11.7 $\pm$ 2.3	14.0 $\pm$ 1.4	5.3 $\pm$ 1.4	1.4 $\pm$ 0.4	2.9 $\pm$ 0.4	0.4 $\pm$ 0.2	26.7 $\pm$ 1.0	25.7 $\pm$ 0.5	32.7 $\pm$ 0.7
4	2	18.8 $\pm$ 2.2	18.1 $\pm$ 1.8	7.1 $\pm$ 1.1	1.5 $\pm$ 0.2	1.8 $\pm$ 0.2	0.7 $\pm$ 0.1	32.5 $\pm$ 1.3	35.7 $\pm$ 1.0	46.4 $\pm$ 0.9
	3	10.2 $\pm$ 1.6	22.0 $\pm$ 2.5	14.8 $\pm$ 2.0	1.2 $\pm$ 0.3	2.5 $\pm$ 0.3	1.7 $\pm$ 0.3	27.9 $\pm$ 0.6	23.9 $\pm$ 1.0	31.8 $\pm$ 1.2
	4	13.9 $\pm$ 1.4	13.7 $\pm$ 1.8	7.2 $\pm$ 1.3	1.1 $\pm$ 0.1	1.0 $\pm$ 0.1	0.6 $\pm$ 0.1	39.6 $\pm$ 1.2	40.9 $\pm$ 1.0	50.6 $\pm$ 1.0
5	2	21.6 $\pm$ 1.5	32.2 $\pm$ 2.5	10.2 $\pm$ 1.7	3.0 $\pm$ 0.3	3.5 $\pm$ 0.4	1.2 $\pm$ 0.2	29.8 $\pm$ 0.7	30.2 $\pm$ 0.9	44.0 $\pm$ 0.5
	3	10.2 $\pm$ 3.1	10.7 $\pm$ 1.9	3.8 $\pm$ 3.7	1.0 $\pm$ 0.2	1.2 $\pm$ 0.3	0.2 $\pm$ 0.1	27.6 $\pm$ 1.3	27.8 $\pm$ 1.1	45.9 $\pm$ 2.6
	4	7.4 $\pm$ 1.9	8.6 $\pm$ 2.0	0.6 $\pm$ 0.3	1.1 $\pm$ 0.4	0.8 $\pm$ 0.1	0.1 $\pm$ 0.1	36.0 $\pm$ 0.7	44.9 $\pm$ 1.2	55.1 $\pm$ 0.8
$\bar{x}$	2	17.9%	21.7%	11.4%	7.97	8.48	4.78	24.68	26.34	34.34
	3	7.4%	10.8%	6.3%	1.35	3.07	1.13	26.37	26.23	34.44
	4	8.8%	9.4%	4.3%	1.18	3.22	1.30	30.32	35.71	41.79

Tabela 6.13: Métrica de diversidade (*m3*) no PRFM (**P6**).

Rede	h	Maximum Spread ($\times 10$)		
		NSGA-II	SPEA	SPEA2
0	2	8.1 $\pm$ 0.1	8.1 $\pm$ 0.1	8.1 $\pm$ 0.2
	3	8.0 $\pm$ 0.1	8.1 $\pm$ 0.1	8.1 $\pm$ 0.1
	4	8.0 $\pm$ 0.0	8.0 $\pm$ 0.1	8.1 $\pm$ 0.1
1	2	8.8 $\pm$ 0.4	8.9 $\pm$ 0.4	8.4 $\pm$ 0.3
	3	7.9 $\pm$ 0.0	8.1 $\pm$ 0.1	8.0 $\pm$ 0.0
	4	7.9 $\pm$ 0.0	8.3 $\pm$ 0.2	8.1 $\pm$ 0.1
2	2	8.6 $\pm$ 0.1	8.2 $\pm$ 0.1	8.6 $\pm$ 0.1
	3	8.2 $\pm$ 0.1	8.2 $\pm$ 0.1	8.2 $\pm$ 0.1
	4	8.7 $\pm$ 0.0	8.8 $\pm$ 0.1	8.8 $\pm$ 0.1
3	2	7.6 $\pm$ 0.1	7.8 $\pm$ 0.2	7.5 $\pm$ 0.1
	3	6.3 $\pm$ 0.3	6.5 $\pm$ 0.4	7.1 $\pm$ 0.2
	4	7.3 $\pm$ 0.0	7.5 $\pm$ 0.1	7.3 $\pm$ 0.0
4	2	7.9 $\pm$ 0.1	8.0 $\pm$ 0.1	8.1 $\pm$ 0.1
	3	7.5 $\pm$ 0.1	7.6 $\pm$ 0.2	7.7 $\pm$ 0.1
	4	8.1 $\pm$ 0.1	8.0 $\pm$ 0.0	8.2 $\pm$ 0.0
5	2	5.4 $\pm$ 0.0	5.5 $\pm$ 0.1	5.3 $\pm$ 0.0
	3	3.5 $\pm$ 0.1	3.6 $\pm$ 0.2	3.6 $\pm$ 0.1
	4	5.2 $\pm$ 0.2	5.1 $\pm$ 0.2	5.3 $\pm$ 0.0
$\bar{x}$	2	7.73	7.75	7.67
	3	6.90	7.02	7.12
	4	7.53	7.62	7.63

testes de hipóteses efetuados na formulação anterior (PRMM), h2 foi mais competitiva com h4; no PRFM recém analisado, pode-se dizer que a concorrência maior ocorreu entre h3 e h4. Assim, uma estratégia mais aleatória do que o menor caminho exclusivo parece dar maior habilidade ao AG para convergir para a fronteira de Pareto nesta situação.

Rede	<i>er</i>	<i>gd</i>	<i>ps</i>	<i>m3</i>	Rede	<i>er</i>	<i>gd</i>	<i>ps</i>	<i>m3</i>
0	<	<	>	=	0	>	>	=	>
1	>	>	<	>	1	=	=	=	<
2	<	<	>	=	2	<	<	>	=
3	<	<	>	>	3	<	<	>	<
4	<	<	>	=	4	<	<	>	>
5	<	<	>	=	5	<	<	>	>

(a) SPEA2 e NSGA-II

(b) SPEA2 e SPEA

Tabela 6.14: Testes de hipóteses para diferença entre as médias do SPEA2, NSGA-II e SPEA (utilizando h4), com $\alpha = 10\%$ bicaudal.

Rede	<i>er</i>	<i>gd</i>	<i>ps</i>	<i>m3</i>	Rede	<i>er</i>	<i>gd</i>	<i>ps</i>	<i>m3</i>
0	=	<	>	=	0	=	=	>	=
1	<	<	>	<	1	=	=	=	=
2	<	<	>	>	2	=	<	>	>
3	<	<	>	<	3	=	<	>	>
4	=	=	>	=	4	<	<	>	>
5	<	<	>	<	5	=	=	>	>

(a) h4 e h2

(b) h4 e h3

Tabela 6.15: Testes de hipóteses para diferença entre as médias de h4, h3 e h2 (utilizando SPEA2), com $\alpha = 10\%$ bicaudal.

6.5 Formulação PRFM (cinco objetivos)

Nesta formulação, continuou-se com o mesmo conjunto de métricas para avaliação de convergência (*er*, *gd* e *ps*) e diversidade (*m3*) da formulação anterior. Na Tab. 6.16 estão apresentadas as cardinalidades dos conjuntos P^* referentes às instâncias avaliadas na formulação PRFM de 5 objetivos (P7 da Sec. 3.1.2). Os parâmetros para o AG que mudaram nesta formulação foram: $Tp = 90$ e $Ng = 100$ para todas as redes.

Consideramos que as heurísticas h2 e h3 foram superadas por h4, com base nos experimentos anteriores. Analogamente, o SPEA2 superou o NSGA-II e o SPEA. Assim, os resultados desta seção utilizam apenas a heurística h4 e o SPEA2, para que seja melhor evidenciado o efeito do Crossover pela Vizinhança. Algumas instâncias menores (redes 0 e 1) foram substituídas por instâncias maiores (redes 6 e 7) devido à mesma razão.

Tabela 6.16: Cardinalidades dos conjuntos Pareto-ótimos P^* para cada instância no PRFM de cinco objetivos (P7).

Rede	$ P^* $
2	424
3	349
4	326
5	551
6	624
7	186

6.5.1 Análise das métricas

Observando os resultados das tabelas 6.17 e 6.18, pode-se notar que o NC reduz a *er* em todas as instâncias, exceto na rede 7. Há uma certa alternância de melhores resultados entre as técnicas NC1 e NC2 a respeito de qual delas reduz esta métrica em cada instância. Relacionando a *er* com *ps*, esta afirmação pode ser reforçada, pois além de reduzir o percentual de soluções dominadas indicado por *er*, o SPEA2 com NC1 ou NC2 proveu mais soluções Pareto-ótimas em termos de quantidades absolutas, como pode ser visto nas médias gerais nas colunas relativas a *ps* da Tab. 6.17. Apesar dessa melhoria, algumas instâncias (5 e 6) se mostraram difíceis de convergir, pois em torno de 50% de suas soluções não são Pareto-ótimas.

Na média geral, os valores de *er* obtidos pelo SPEA2nc1 e SPEA2nc2 foram de 29.5% e 29.6%, respectivamente, contra 32.1% do SPEA2, o que significa que os dois primeiros calcularam em torno de 70.5% e 70.6% de soluções Pareto-ótimas, contra o percentual de 67.9% obtido pelo SPEA2, em uma formulação do PRM que se mostrou difícil de convergir, talvez devido à maior dimensionalidade do espaço dos objetivos.

Uma análise da métrica *gd* mostra que a presença do NC também levou a melhores valores na maioria dos casos, aproximando mais as soluções dominadas à fronteira de Pareto. Além disso, as médias de *gd* evidenciam que o NC também trouxe melhorias ao SPEA2.

Na avaliação de diversidade pela métrica *m3*, os três modelos foram bastante competitivos. Há uma alternância entre qual dos três cobriu uma extensão maior do espaço dos objetivos. Uma análise estatística a ser mostrada mais adiante irá reforçar essas médias.

Tabela 6.17: Métricas de convergência (*er*, *gd* e *ps*) no PRFM (**P7**) utilizando h4.

Rede	Error Rate (%)			Generational Distance ($\times 10^3$)			Pareto Subset		
	SPEA2	SPEA2nc1	SPEA2nc2	SPEA2	SPEA2nc1	SPEA2nc2	SPEA2	SPEA2nc1	SPEA2nc2
2	25.7% $\pm$ 1.8%	20.6% $\pm$ 1.2%	22.3% $\pm$ 1.4%	2.4 $\pm$ 0.1	2.1 $\pm$ 0.1	2.3 $\pm$ 0.1	66.9 $\pm$ 1.6	71.4 $\pm$ 1.1	69.9 $\pm$ 1.2
3	22.3% $\pm$ 1.2%	20.6% $\pm$ 1.1%	19.6% $\pm$ 0.9%	2.2 $\pm$ 0.1	2.3 $\pm$ 0.1	2.0 $\pm$ 0.1	69.9 $\pm$ 1.1	71.4 $\pm$ 1.0	72.4 $\pm$ 0.8
4	15.1% $\pm$ 1.4%	11.5% $\pm$ 0.9%	13.0% $\pm$ 1.7%	1.5 $\pm$ 0.1	1.4 $\pm$ 0.1	1.4 $\pm$ 0.1	76.4 $\pm$ 1.3	79.6 $\pm$ 0.8	78.3 $\pm$ 1.5
5	54.8% $\pm$ 2.0%	49.9% $\pm$ 1.9%	49.9% $\pm$ 2.4%	2.9 $\pm$ 0.2	2.7 $\pm$ 0.1	2.8 $\pm$ 0.2	40.7 $\pm$ 1.8	45.1 $\pm$ 1.8	45.1 $\pm$ 2.2
6	61.1% $\pm$ 1.8%	57.6% $\pm$ 1.7%	56.6% $\pm$ 1.8%	3.4 $\pm$ 0.1	3.1 $\pm$ 0.1	3.1 $\pm$ 0.2	35.0 $\pm$ 1.6	38.2 $\pm$ 1.5	39.1 $\pm$ 1.6
7	13.4% $\pm$ 1.3%	16.7% $\pm$ 1.7%	16.0% $\pm$ 1.3%	1.5 $\pm$ 0.2	1.7 $\pm$ 0.2	1.8 $\pm$ 0.3	77.9 $\pm$ 1.2	75.0 $\pm$ 1.5	75.6 $\pm$ 1.2
$\bar{x}$	32.1%	29.5%	29.6%	2.32	2.22	2.23	61.14	63.47	63.39

Tabela 6.18: Métrica de diversidade (*m3*) no PRFM (**P7**) utilizando h4.

Rede	Maximum Spread ($\times 10$)		
	SPEA2	SPEA2nc1	SPEA2nc2
2	9.1 $\pm$ 0.1	9.2 $\pm$ 0.1	9.3 $\pm$ 0.1
3	9.6 $\pm$ 0.1	9.6 $\pm$ 0.0	9.5 $\pm$ 0.0
4	8.3 $\pm$ 0.1	8.3 $\pm$ 0.1	8.3 $\pm$ 0.1
5	8.7 $\pm$ 0.2	8.5 $\pm$ 0.2	8.5 $\pm$ 0.2
6	11.1 $\pm$ 0.1	11 $\pm$ 0.1	11.1 $\pm$ 0.1
7	8.5 $\pm$ 0.1	8.5 $\pm$ 0.1	8.5 $\pm$ 0.1
$\bar{x}$	9.22	9.18	9.20

Como uma tentativa de refinar mais as diferenças entre o SPEA2nc1 e o SPEA2nc2, usou-se a métrica *two set coverage* (*sc*), com o intuito de comparar diretamente os resultados dos dois métodos em termos de dominância. Os resultados estão apresentados na

Tab. 6.19. Como pode ser visto, na média, o SPEA2nc2 domina um percentual maior de soluções do SPEA2nc1 do que o contrário.

Tabela 6.19: A métrica *two set coverage* aplicada na comparação entre SPEA2nc1 e SPEA2nc2.

Rede	$sc(nc1, nc2)$	$sc(nc2, nc1)$
2	0.100	0.067
3	0.011	0.056
4	0.067	0.022
5	0.089	0.222
6	0.111	0.133
7	0.056	0.122
$\bar{x}$	7.2%	10.4%

6.5.2 Testes de hipóteses

Inicialmente, foram realizados testes de hipóteses para verificar se as melhorias ocasionadas pela incorporação do NC no SPEA2 foram significativas. Novamente, foi considerado um nível de significância $\alpha = 10\%$ bicaudal. Assim, as seguintes hipóteses foram formuladas:

H_0 : a inclusão do NC ao modelo não resulta em diferença significativa para alguma métrica μ , i.e., $\mu_{+nc} = \mu_{-nc}$.

H_1 : a inclusão do NC resulta em diferença significativa para alguma métrica μ , i.e., $\mu_{+nc} \neq \mu_{-nc}$.

A Tab. 6.20 apresenta os resultados dos testes em cada uma das quatro métricas, comparando o SPEA2nc1 com SPEA2, o SPEA2nc2 com o SPEA2, e por fim, o SPEA2nc1 com o SPEA2nc2.

Os resultados das tabelas 6.20a e 6.20b mostram que a incorporação das duas variações do mecanismo de NC levou a melhorias similares sobre o SPEA2 no aspecto de convergência do AG, especialmente nas métricas *er* e *ps*. Ou seja, o NC pode auxiliar o AG a prover mais soluções Pareto-ótimas, tanto no sentido relativo (*er*) quanto absoluto (*ps*). Por sua vez, os testes para *gd* evidenciaram que o NC não piora nenhum cenário, ainda sendo capaz de melhorar alguns casos. Logo, de um modo geral, existe evidência de que os conjuntos P retornados com o uso do NC estão mais próximos de P^* .

Na análise de diversidade, não foi possível observar nenhuma mudança significativa pelo NC1 em nenhum caso e em apenas um pelo NC2. Apesar disso, nenhum NC piorou os resultados, mostrando que as três propostas foram competitivas nesta métrica, como havia sido evidenciado nas estimativas pontuais comentadas anteriormente.

Um terceiro conjunto de testes de hipóteses foi feito para verificar se existe evidência de melhor desempenho de um NC sobre o outro. Os resultados apresentados na Tab. 6.20c

mostram que em apenas 4/24 testes houve diferenças observáveis, mas mesmo nesses casos houve um empate entre NC1 e NC2.

Rede	<i>er</i>	<i>gd</i>	<i>ps</i>	<i>m3</i>	Rede	<i>er</i>	<i>gd</i>	<i>ps</i>	<i>m3</i>
2	<	<	>	=	2	<	=	>	>
3	<	=	>	=	3	<	<	>	=
4	<	=	>	=	4	=	=	=	=
5	<	<	>	=	5	<	=	>	=
6	<	<	>	=	6	<	<	>	=
7	>	=	<	=	7	>	=	<	=

(a) SPEA2nc1 e SPEA2

Rede	<i>er</i>	<i>gd</i>	<i>ps</i>	<i>m3</i>
2	=	>	=	=
3	=	<	=	<
4	=	=	=	=
5	=	=	=	=
6	=	=	=	>
7	=	=	=	=

(b) SPEA2nc2 e SPEA2

Rede	<i>er</i>	<i>gd</i>	<i>ps</i>	<i>m3</i>
2	=	>	=	=
3	=	<	=	<
4	=	=	=	=
5	=	=	=	=
6	=	=	=	>
7	=	=	=	=

(c) SPEA2nc1 e SPEA2nc2

Tabela 6.20: Testes de hipóteses para diferença entre as médias do SPEA2 sem NC, com NC1 e com NC2, considerando $\alpha = 10\%$ bicaudal.

6.6 Comparação com algoritmos da literatura

Nesta seção, procede-se para um segundo tipo de experimento, em que o modelo obtido nas etapas anteriores (SPEA2+h4) é comparado a dois conhecidos algoritmos da literatura dos problemas de Roteamento e Steiner: o algoritmo de Dijkstra (também conhecido como *Shortest Path Tree* ou SPT) [14] e a heurística de Takahashi & Matsuyama [51]. Primeiramente compara-se o AG com o SPT, o qual foi executado para minimizar o *delay* fim-a-fim em cada instância. A principal característica do algoritmo SPT que será relevante aqui é que ele garante o *delay* fim-a-fim ótimo do caminho entre dois vértices quaisquer em um grafo conectado. Deste modo, o AG foi definido para minimizar o objetivo D_{max} junto de *custo*, *hops*, D_t , α_{max} e $\bar{\alpha}$, a fim de observar:

- (i) em quantas execuções o AG retornou soluções com D_{max} que atendia ao *delay* ótimo fornecido pelo SPT (i.e., $D_{max} = d_{max}^{spt}$);
- (ii) quando atendido, quão melhor foram as soluções do AG em respeito aos outros objetivos mencionados; e
- (iii) se for definida uma tolerância para d_{max}^{spt} , se haveria aumento dos ganhos obtidos em comparação ao uso de um *delay* rígido d_{max}^{spt} .

Observemos que na comparação entre o AG e o SPT não é possível melhorar o objetivo $\overline{D}$, pois ele corresponde à média dos *delays* fim-a-fim, para os quais o SPT fornece a solução ótima, como discutido. Um argumento análogo vale para D_a .

A seguinte figura de comparação foi usada para determinar os ganhos/perdas do AG sobre os dois métodos:

$$g = \frac{f_i^X - f_i^{AG}}{f_i^{AG}}$$

onde f_i^X e f_i^{AG} correspondem aos valores do objetivo i obtidos pelo algoritmo X (SPT ou TM) e pelo AG, para uma dada solução (p.ex., aquela que atende d_{max}^{spt}), respectivamente. Logo, quando $g > 0$, a métrica fornece o quanto a solução de X é pior em relação àquela do AG; quando $g < 0$, a solução do AG terá sido pior que a do método X .

6.6.1 Comparação com o algoritmo de Dijkstra

Os resultados relacionados à comparação entre as soluções retornadas pelo AG e as soluções retornadas pelo SPT estão mostrados nas tabelas 6.21 e 6.22. Considerando o caso em que o AG deve atender ao *delay* rígido d_{max}^{spt} do SPT, a tabela mostra que o AG foi capaz de encontrar soluções que atendessem a tal *delay* em todos os casos. Além disso, pode-se notar que o AG também foi capaz de obter soluções que melhorassem os outros objetivos (*custo*, *hops*, D_t , α_{max} e $\overline{\alpha}$) em quase todas as redes. As métricas α_{max} e $\overline{\alpha}$ se mostraram mais difíceis de serem melhoradas, com uma melhoria mais evidente nas redes maiores (2 a 7). Em alguns casos não ocorreram ganhos, tais como nos objetivos *hops* e D_t na rede 3; de fato, não havia soluções em P^* com valores menores para esses objetivos que atendessem d_{max}^{spt} . As médias considerando todas as redes estão apresentadas na Fig. 6.2, mostrando que as soluções do SPT foram piores que as do AG principalmente em termos do custo. No geral, foram obtidos ganhos de 17.9%, 9.5%, 11.9%, 2.3% e 3.6% para o *custo*, *hops*, D_t , α_{max} e $\overline{\alpha}$, respectivamente.

Considerando os cenários em que tolerâncias de 10%, 15% e 20% para o *delay* d_{max} do SPT são permitidas, à medida que o valor de *tol* aumenta, os ganhos nos cinco objetivos aumentam (com exceção de D_t que fica estável após $tol = 10\%$). O objetivo que fornece o melhor ganho é o custo. Ainda vale observar que objetivos relacionados ao uso da rede (p.ex., α_{max} e $\overline{\alpha}$) têm um bom crescimento em relação aos seus valores iniciais (sem tolerância), especialmente α_{max} que sai de 2.3% para 15.4% ao ser permitida uma pequena tolerância de 10%. Por outro lado, alguns objetivos aumentam pouco ou nada mesmo com o aumento da tolerância, o que na maioria dos casos ocorre por, de fato, não existirem soluções melhores nessa faixa de tolerância. Ou seja, para que outras soluções melhores aparecessem (caso existissem), seria necessário aplicar um valor mais alto de tolerância. A observação dos conjuntos P^* permite essa constatação.

Por exemplo, consideremos que num MOP com 4 objetivos (rede 3) a solução que

dá o menor D_{max} tenha a tupla $a = (0.690, 127, 46, 22)$ de objetivos, supondo que cada tupla seja dada por $(\alpha_{max}, custo, D_{max}, hops)$. Para que um α_{max} sofresse uma redução de aproximadamente 15%, D_{max} teria que ser aumentado de 46 para pelo menos 71, ou seja, seria necessária uma tolerância de aproximadamente 55%. A solução que propiciaria tal redução de α_{max} com o menor D_{max} seria $b = (0.589, 131, 71, 25)$. Esse exemplo mostra que uma pequena melhoria num objetivo pode, às vezes, requerer uma aumento maior de outro, portanto, os ganhos apresentados pelo AG podem ser considerados bons percentuais.

Tabela 6.21: Percentuais mostrando os ganhos do AG sobre o algoritmo SPT, considerando valores rígidos para d_{max}^{spt} e tolerância de 10%.

Rede	$D_{max} = d_{max}^{spt}$						$tol = 10\%$				
	d_{max}^{spt} at.	$custo$	$hops$	D_t	α_{max}	$\bar{\alpha}$	$custo$	$hops$	D_t	α_{max}	$\bar{\alpha}$
0	100%	27.4%	25.0%	26.9%	0.0%	0.0%	31.1%	25.0%	26.9%	35.0%	9.8%
1	100%	30.4%	11.1%	15.4%	0.0%	1.8%	30.4%	11.1%	15.4%	0.0%	1.8%
2	100%	22.1%	16.7%	12.8%	2.1%	5.3%	39.1%	16.7%	29.3%	28.8%	9.5%
3	100%	1.5%	0.0%	0.0%	0.0%	7.2%	35.1%	4.8%	6.6%	0.0%	7.4%
4	100%	23.2%	14.3%	17.6%	0.0%	-0.3%	35.3%	23.1%	25.1%	14.7%	5.7%
5	100%	15.2%	2.2%	3.8%	16.3%	6.2%	17.5%	4.4%	7.1%	16.3%	6.7%
6	100%	8.7%	3.7%	16.3%	0.0%	3.4%	33.8%	21.7%	18.8%	23.8%	13.1%
7	100%	14.4%	3.2%	2.7%	0.0%	5.3%	17.0%	3.2%	2.7%	5.0%	7.5%
$\bar{x}$	100%	17.9%	9.5%	11.9%	2.3%	3.6%	29.9%	13.8%	16.5%	15.4%	7.7%

Tabela 6.22: Percentuais mostrando os ganhos do AG sobre o algoritmo SPT, considerando tolerâncias de 15% e 20% para o d_{max}^{spt} .

Rede	$tol = 15\%$					$tol = 20\%$				
	$custo$	$hops$	D_t	α_{max}	$\bar{\alpha}$	$custo$	$hops$	D_t	α_{max}	$\bar{\alpha}$
0	50.0%	25.0%	26.9%	35.0%	19.2%	50.0%	25.0%	26.9%	35.0%	19.2%
1	30.4%	25.0%	15.4%	0.0%	3.3%	30.4%	25.0%	15.4%	0.0%	3.3%
2	49.5%	23.5%	29.3%	31.1%	11.1%	49.5%	23.5%	29.3%	33.0%	13.6%
3	38.0%	10.0%	6.6%	0.0%	8.9%	46.0%	10.0%	6.6%	5.9%	11.7%
4	35.3%	23.1%	25.1%	14.7%	6.5%	38.9%	23.1%	25.1%	14.7%	10.9%
5	17.5%	4.4%	7.1%	16.3%	6.9%	18.5%	4.4%	7.1%	16.3%	7.3%
6	34.2%	21.7%	18.8%	23.8%	14.3%	34.2%	21.7%	18.8%	23.8%	14.4%
7	17.0%	3.2%	2.7%	5.2%	7.8%	17.0%	3.2%	2.7%	5.8%	9.0%
$\bar{x}$	34.0%	17.0%	16.5%	15.7%	9.7%	35.6%	17.0%	16.5%	16.8%	11.2%

6.6.2 Comparação com a heurística Takahashi-Matsuyama

Em seguida, compara-se o AG e a heurística de Takahashi & Matsuyama [51], a qual provê um fator de aproximação de 2opt quando aplicada para calcular o custo de uma árvore multicast. Neste caso, busca-se verificar a capacidade do AG em retornar soluções que atendam ou reduzam o custo do TM, i.e., verificar se $c_{AG} \leq c_{TM}$, e quais os ganhos obtidos nos outros objetivos quando tal condição é satisfeita. As análises de tolerâncias também são aplicadas.

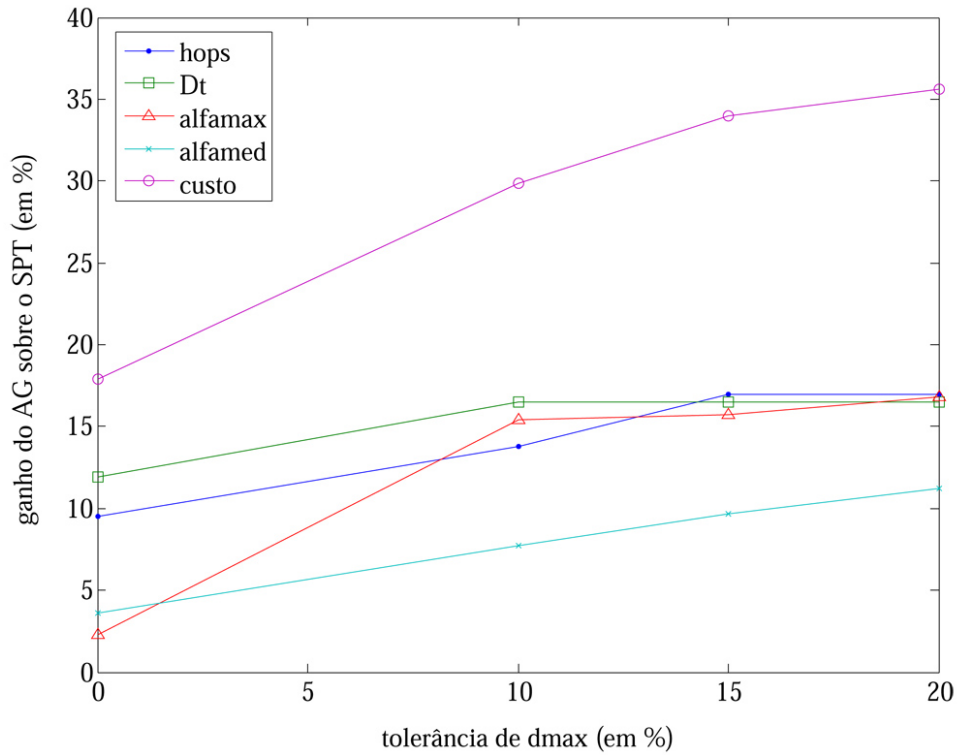


Figura 6.2: Médias de ganho do AG sobre o algoritmo SPT.

Com base nos resultados apresentados nas tabelas 6.23 e 6.24, pode-se ver que apesar do TM ter obtido valores ótimos de custo nas redes menores (rede 0 e rede 1), em várias redes maiores isso não ocorreu (p.ex., 2, 6 e 7), ao contrário do AG, que conseguiu encontrar o ótimo em praticamente todos os casos, com exceção da rede 5, mas por uma diferença muito pequena. Isto significa que, além de prover soluções que atendessem ao custo do TM, o AG foi capaz de melhorá-lo quando possível (i.e., quando o TM não encontrou o custo ótimo). Além disso, observando os conjuntos Pareto-ótimos das instâncias em que o TM encontrou o custo ótimo, torna-se visível o porquê de não ter sido possível ao AG melhorar os demais objetivos (D_{max} , $hops$, etc.): de fato, não havia outras soluções com custo ótimo que tivessem valores reduzidos para tais objetivos. Assim, os ganhos médios referentes ao custo do TM sem tolerância são relativamente baixos para alguns objetivos ($hops$, α_{max} e $\bar{\alpha}$), como mostrado na Fig. 6.3. Entretanto, quando o custo do TM é flexibilizado, ganhos mais expressivos são obtidos, especialmente nos objetivos relacionados ao *delay* (D_{max} , D_t e $\bar{D}$). Em tais casos ($tol \geq 10$), as melhorias do AG sobre o TM são até mesmo maiores que aquelas obtidas sobre o algoritmo SPT, nos experimentos anteriores.

O tempo médio ao executar cada instância com cada MOEA/heurística está mostrado na Tab. 6.25, baseado em uma implementação em C++ executada em uma configuração Intel Quad Core 2.00 GHz com 4GB. Especificamente, o tempo para construir uma rota multicast para as redes menores (0 e 1) utilizando SPEA2+h4 foi em torno de 0.005 s, enquanto que para as redes maiores (2 a 7), foi em torno de 0.056 s. Foi possível observar a relação de consumo de tempo $h3 < h4 < h2$; h4 foi em torno de 13% mais rápida que

Tabela 6.23: Percentuais mostrando os ganhos do AG sobre o algoritmo TM, considerando $c_{AG} \leq c_{TM}$ e tolerância de 10%.

Rede	vals. absolutos			$c_{GA} \leq c_{TM}$							$tol = 10\%$					
	c^*	c_{TM}	c_{AG}	$custo$	D_{max}	$hops$	D_t	$\bar{D}$	α_{max}	$\bar{\alpha}$	D_{max}	$hops$	D_t	$\bar{D}$	α_{max}	$\bar{\alpha}$
0	69	69	69	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.2%	0.0%	0.0%	0.0%
1	71	71	71	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
2	149	163	149	9.4%	2.7%	5.0%	28.8%	0.0%	0.0%	3.2%	40.6%	23.5%	54.5%	15.9%	29.9%	8.2%
3	83	84	83	1.2%	0.0%	0.0%	0.6%	0.0%	0.0%	2.9%	11.5%	10.0%	8.0%	21.8%	10.1%	8.0%
4	104	104	104	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	32.6%	3.8%	14.0%	39.0%	8.0%	3.4%
5	220	220	220.1	0.0%	0.0%	0.0%	0.0%	1.6%	0.0%	0.0%	95.3%	8.9%	23.0%	69.2%	16.3%	7.8%
6	86	91	86	5.8%	17.1%	13.6%	16.2%	38.4%	3.1%	3.8%	44.4%	18.5%	29.9%	75.4%	3.1%	10.5%
7	165	178	165	7.9%	57.8%	25.8%	50.9%	61.3%	10.5%	2.8%	65.1%	25.8%	50.9%	73.0%	16.9%	6.7%
$\bar{x}$	-	-	-	3.0%	9.7%	5.6%	12.1%	12.7%	1.7%	1.6%	36.2%	11.3%	22.8%	36.8%	10.5%	5.6%

Tabela 6.24: Percentuais mostrando os ganhos do AG sobre o algoritmo TM, considerando tolerâncias de 15% e 20% para c_{TM} .

Rede	$tol = 15\%$						$tol = 20\%$					
	D_{max}	$hops$	D_t	$\bar{D}$	α_{max}	$\bar{\alpha}$	D_{max}	$hops$	D_t	$\bar{D}$	α_{max}	$\bar{\alpha}$
0	0.0%	11.1%	35.3%	0.0%	0.0%	0.0%	0.0%	11.1%	35.3%	0.0%	0.0%	0.0%
1	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.7%	25.6%	8.9%
2	40.6%	23.5%	66.7%	15.9%	31.2%	9.8%	50.0%	31.3%	66.7%	27.1%	31.6%	12.1%
3	18.4%	10.0%	9.5%	24.1%	10.1%	10.7%	18.4%	10.0%	12.5%	28.4%	10.1%	11.1%
4	45.2%	3.8%	23.3%	41.3%	8.0%	6.5%	45.2%	3.8%	24.0%	41.3%	8.4%	6.5%
5	95.3%	8.9%	27.1%	70.6%	16.3%	9.5%	95.3%	8.9%	29.1%	71.5%	16.3%	10.2%
6	44.4%	18.2%	38.0%	93.6%	3.1%	12.1%	73.3%	18.2%	43.5%	105.2%	3.1%	13.7%
7	65.1%	25.8%	50.9%	73.0%	16.9%	7.6%	65.1%	25.8%	50.9%	76.2%	16.9%	8.0%
$\bar{x}$	38.6%	12.7%	31.3%	39.8%	10.7%	7.0%	43.4%	13.6%	32.7%	43.9%	14.0%	8.8%

h2, a em torno de 17% mais lenta que h3, na média. Esses resultados mostram também que se uma heurística mais rápida é necessária, é possível utilizar h3, especialmente se uma formulação com quatro ou cinco objetivos for considerada, podendo ser considerada uma boa alternativa para o roteamento.

Tabela 6.25: Tempo médio (em segundos) para construir uma rota multicast utilizando SPEA2.

Rede	h2	h3	h4
0	0.007	0.006	0.007
1	0.006	0.005	0.005
2	0.009	0.007	0.008
3	0.009	0.006	0.008
4	0.049	0.041	0.046
5	0.131	0.111	0.121
6	0.104	0.077	0.091
7	0.107	0.096	0.101
$\bar{x}$	0.068	0.056	0.062

6.6.3 Considerações

Os estudos sobre flexibilização de d_{max} ou custo da árvore podem ser uma interessante ferramenta para a tomada de decisão, administração de rede, dentre outros. Por exemplo, em diferentes momentos de operação, políticas que determinam o valor do *delay* podem ser usadas para escolher soluções mais adequadas na lista de soluções não-dominadas oferecidas pelo AG. Dependendo dos valores dos parâmetros negociados, um leque de

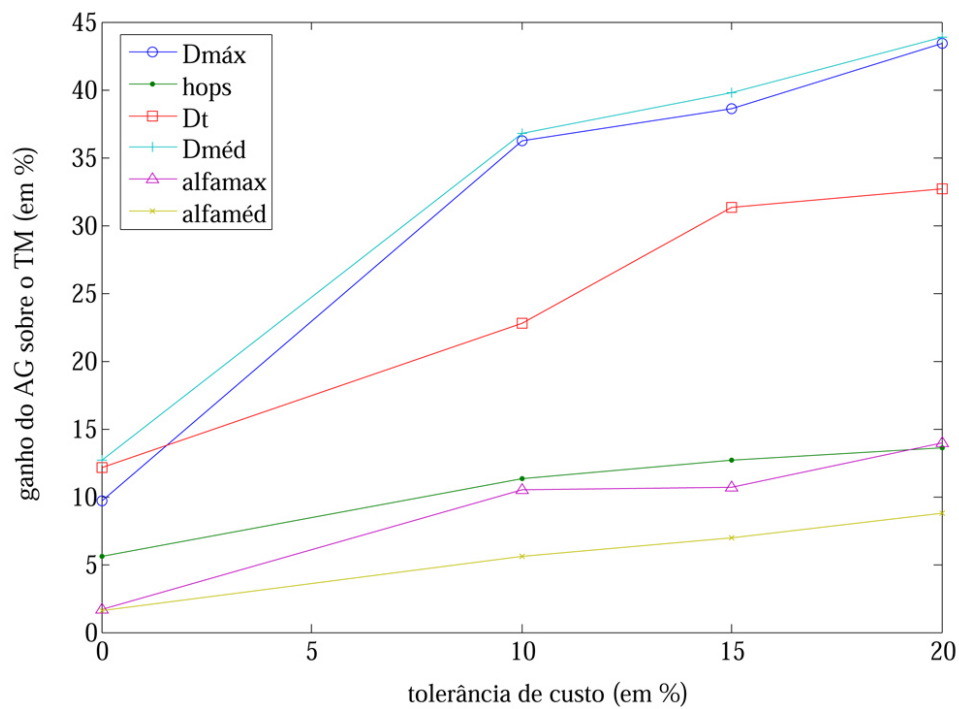


Figura 6.3: Médias de ganho do AG sobre o algoritmo TM.

soluções estará disponível para atender um ou mais requisitos. Esse tipo de abordagem também pode facilitar o re-roteamento rápido para escolha de rotas alternativas, evitando a repetição de cálculos de novas rotas. Nesse caso, é possível que alterações no algoritmo (p.ex., SPT) ou na topologia da rede (para atualizar o *status* de enlaces com problemas, por exemplo) tivessem de ser feitas, para que rotas alternativas pudessem ser calculadas. Esse tipo de situação pode ser provida por um AG de forma natural, devido à sua característica de método multiponto.

Capítulo 7

Conclusões

7.1 Considerações Finais

Este trabalho investigou a aplicação de algoritmos evolutivos multiobjetivos (MOEAs) no Problema do Roteamento Multicast. Algumas formulações multiobjetivo do PRM com dois, quatro e cinco objetivos foram consideradas, além de uma mono-objetivo. Foram consideradas funções objetivo que podem ser utilizadas para atender a certos requisitos de QoS e Engenharia de Tráfego ao calcular rotas multicast, tais como redução do custo, do *hops count*, do *delay* fim-a-fim, da utilização média da largura de banda e minimização do gargalo da rede. Dentre essas funções objetivo, propomos aquela relacionada com a utilização média dos enlaces, a qual não havia sido encontrada nos trabalhos pesquisados; juntamente com o gargalo, elas podem prover um interessante par de objetivos que podem ser usados durante a escolha de rotas por um decisor ou administrador de rede, buscando, por exemplo, a otimização de recursos da rede no momento do carregamento de uma nova rota e seus dados na rede.

No processo de busca evolutiva desenvolvida foram utilizados MOEAs conhecidos da literatura (NSGA-II, SPEA, SPEA2) junto de um conjunto de operadores anteriormente propostos por Ravikumar & Bajpai [45], Zhengying *et al.* [60], Araujo & Oliveira [39], e Oliveira & Vita [40]. No cruzamento, a geração de um filho é guiada pela herança de partes comuns aos pais - subárvores - que deverão ser reconectadas via menor caminho para resultar numa árvore multicast. Porém, a investigação desses operadores mostrou que existia uma potencial falha durante o cruzamento e a mutação, podendo levar à geração de indivíduos inválidos (ocorrência de laços em árvores), como mostrado através de exemplos. Essas falhas foram quantificadas em algumas instâncias de teste, mostrando que a convergência do AG ficou comprometida, especialmente em redes maiores. Assim, novas heurísticas para realizar as reconexões nos operadores citados foram propostas para corrigir esta falha: a primeira mantendo o uso de menor caminho durante reconexões mas evitando a ocorrência de laços; a segunda, realizando as reconexões de forma aleatória;

e a terceira, como uma escolha aleatória entre das anteriores. Um mecanismo de Cruzamento pela Vizinhança (NC) que efetua a seleção de pais similares também foi avaliado, aumentando o determinismo da seleção para cruzamento do AG.

A avaliação experimental foi realizada utilizando-se oito instâncias de dimensões variadas originadas na literatura de Steiner e Roteamento. Para avaliação de convergência, foram utilizadas as métricas *error rate* (*er*), *generational distance* (*gd*) e *pareto subset* (*ps*). Por sua vez, a diversidade foi avaliada através do *spread* (Δ) e *maximum spread* (*m3*). A métrica *two set coverage* (*sc*) foi utilizada para comparar dois conjuntos de soluções (p.ex., duas heurísticas diferentes) em termos de dominância. Como o AG é um método estocástico, várias execuções foram feitas para que médias e intervalos de confiança pudessem ser calculados. Além disso, para evidenciar estatisticamente as comparações entre heurísticas/MOEAs/NCs foram utilizados testes de hipóteses.

Os primeiros experimentos foram realizados na formulação mono-objetivo (PRMMo), os quais indicaram que a heurística do modelo original (h1) teve sua convergência bastante prejudicada devido à presença das violações mencionadas. Já as novas heurísticas forneceram bons resultados, especialmente a heurística combinada (h4), que teve uma convergência média de 87.4%. A heurística mais determinista (h2) foi a segunda melhor, com convergência de 79.6%. Esses resultados evidenciam que no PRMMo é possível crer que a utilização de um menor caminho guie o AG para regiões mais promissoras de busca. Apesar disso, a heurística que adiciona arestas aleatórias durante a reconexão (h3) teve um desempenho razoável, obtendo um percentual de convergência em torno de 70%. Esses dados também mostram que foi importante ter realizado essa correção na heurística original, tanto do ponto de vista da melhoria da convergência, quanto da validade das soluções.

Os experimentos na formulação de dois objetivos (PRMM) novamente mostraram que h4 foi melhor que as outras duas novas heurísticas. Relativamente aos três MOEAs, foi possível ver que as melhorias do SPEA2 propiciaram uma melhor exploração do espaço de busca do que o seu antecessor (SPEA) e o NSGA-II. Na média geral das redes, o SPEA2 melhorou todas métricas (*er*, *gd* e Δ) qualquer uma das três heurísticas. Porém, quando combinado com h4, os resultados foram os melhores destes cenários, onde foram obtidos valores de *er* menores que 10% e distância ao fronte próxima a zero, além de ter alcançado uma distribuição melhor das soluções ao longo do espaço dos objetivos. Testes de hipóteses evidenciaram que, de fato, estatisticamente foi significativa a melhoria de h4 sobre h2 e h3 em praticamente todas as métricas (com exceção de um empate entre h4 e h2 em Δ). De um modo geral, novamente h2 saiu melhor que h3.

Na formulação do PRFM com quatro objetivos, tornou-se visível que h3 passou a superar h2 na convergência (métricas *er*, *gd* e *ps*) na maior parte dos casos. Assim, talvez nestes cenários seja possível evidenciar uma limitação ao se utilizar reconexões unicamente pelo menor caminho, possivelmente pelo maior número de dimensões do espaço de busca.

Novamente, h4+SPEA2 foi a combinação que deu o melhor *trade-off* no conjunto de métricas avaliadas (*er*, *gd* e *ps* para convergência, e *m3* para diversidade).

Na formulação de cinco objetivos do PRFM foram aplicadas duas implementações do mecanismo de Cruzamento pela Vizinhaça (NC) ao SPEA2. Foi possível verificar uma melhoria significativa do NC1/NC2 sobre o SPEA2 sem NC. No entanto, não foram muito distinguível a comparação entre as versões NC1 e NC2, apesar de o NC2 ter se saído um pouco melhor na métrica de cobertura (*sc*). O NC foi testado sobre a formulação de quatro objetivos, porém, não resultou em diferença notável, talvez devido à cardinalidade menor dos conjuntos de Pareto-ótimos em relação à formulação com cinco objetivos. Talvez o NC tenha maior efeito em problemas de alta dimensionalidade no espaço dos objetivos, pois este mecanismo foi testado originalmente sobre problemas cuja fronteira de Pareto tem infinitos ou um número muito grande de pontos.

Por fim, pode-se dizer que, dentre as quatro heurísticas avaliadas, a heurística h4 foi aquela que retornou os melhores resultados nas métricas avaliadas nas formulações do PRM discutidas acima. A idéia de h4 é compor indivíduos que misturam os “formatos” de h2 e h3, como uma tentativa de superar as limitações de cada uma quando usadas isoladamente. Experimentalmente, foi possível perceber que na maior parte das vezes h4, de fato, acompanha ou melhora os resultados da melhor das opções em cada cenário. Ao ser combinada com o SPEA2, resultou em bons resultados na maioria dos casos, fornecendo uma opção “segura” com algum grau de generalização, que foi um dos objetivos da experimentação em mais de uma formulação do PRM com várias instâncias. Por sua vez, a utilização do NC pode melhorar o desempenho deste AG no caso de se trabalhar com cinco dimensões. Experimentos futuros podem indicar o seu efeito em outras especificações (dois, três objetivos, etc.), apesar de termos realizado alguns experimentos com quatro objetivos e não ter sido notada diferença para tal caso.

Na Tab. 7.1 encontra-se um resumo dos principais artigos publicados que resultaram desta dissertação.

Tabela 7.1: Principais publicações decorrentes da pesquisa realizada.

Veículo	Formulação do PRM	Heurísticas	MOEAs	Instâncias	Avaliação experimental	Comparações
Journal of Heuristics [9]	PRMM (P1 ao P5) e PRMMo	h1, h2, h3 e h4	NSGA-II e AG clássico	0, 1, 2, 3, nsf [15]	Métrica de convergência [40] e <i>Spread</i> . Intervalos de confiança individuais e para comparação entre duas médias.	Tabelas e gráficos para comparação com SPT e TM. Comparações com o filtro f_{mut} .
SMC [8]	PRFM 5-obj. (P7)	h4	SPEA2, nc1, nc2	2, 3, 4, 5, 6, 7	<i>er, gd, ps, m3, sc</i> . Testes de hipóteses para comparação de duas médias.	Tabelas para comparação com SPT e TM.
SBRN [7]	PRFM 3-obj. (não estudado) e PRFM 4-obj. (P6)	h4, filtros novos	SPEA2	2, 3, 4, 5, 6, 7	<i>er, gd, ps, m3, sc</i> .	Comparações dos filtros novos f_{rnd} e f_{tm} com o original f_{mut} .
CIT [11]	PRFM 4-obj. (P6)	h4	SPEA, SPEA2	0, 1, 2, 3, 4, 5	<i>er, gd, spacing, m3</i>	SPT.
NCA [12]	PRFM 4-obj. (não estudado; troca <i>hops</i> por $\bar{D}$)	h2, h3	SPEA2	0, 1, 2, 3, 4, nsf	<i>er, gd, spacing</i>	SPT.
CEC [10]	PRFM 4-obj. (P6)	h2, h3, h4	NSGA-II, SPEA, SPEA2	0, 1, 2, 3, 4, nsf	<i>er, gd, spacing, m3</i>	-
CBRN [6]	PRMMo	h1, h2, h3	-	0, 1, 2	Convergência para o ótimo.	-

7.2 Sugestões para Trabalhos Futuros

- avaliação dos algoritmos propostos por meio de modelagem e simulação de modelos de tráfego estocásticos (utilizando, por exemplo, o Network Simulator), considerando aspectos como Controle de Admissão de Chamadas e Escalonamento.
- estudar abordagens *many-objectives* [29], [30], [34] para melhorar a aproximação aos conjuntos ótimos de Pareto, reduzindo a deteriorização durante o processo evolutivo de busca, por exemplo.
- desenvolvimento de um MOEA memético através da aplicação de técnicas como busca local.
- outras análises estatísticas para comparar os métodos podem ser feitas, como testes não paramétricos (p.ex., Mann-Whitney e Kruskal-Wallis) e análise de regressão.
- avaliar a aplicação dinâmica do NC, onde ele poderia ser “ligado/desligado” em momentos distintos da busca evolutiva, de acordo com momentos adequados pré-estabelecidos (utilizando métricas como *rate convergence* [53] ou similares que estimam como um algoritmo converge ao longo das gerações).
- investigar outras noções de NC, como pode ser visto no estudo de Yijie & Gongzhang [58], onde a distância entre pais também foi investigada no sentido de indivíduos distantes (min-max).

- uso de outras métricas para avaliação de diversidade em problemas com mais de 2 objetivos, tais como o desvio χ^2 da contagem de nichos σ [19].

Referências Bibliográficas

- [1] Y P Aneja. An integer programming approach to the steiner problem in graphs. *Networks*, (10):167–178, 1980.
- [2] Marcus Poggi Aragão, Celso C. Ribeiro, Eduardo Uchoa, and Renato F. Werneck. Hybrid local search for the steiner problem in graphs. In *in Extended Abstracts of the 4th Metaheuristics International Conference*, pages 429–433, 2001.
- [3] Aluizio Fausto Ribeiro Araújo and Cícero Garrozi. Mulroga: A multicast routing genetic algorithm approach considering multiple objectives. *Applied Intelligence*, 2008.
- [4] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and Principles of Internet Traffic Engineering, 2002.
- [5] J.E. Beasley. OR-Library: Distributing test problems by electronic mail. *Journal of the Op. Research Society*, 41:1069–1072, 1990.
- [6] Marcos L. P. Bueno and Gina M. B. Oliveira. New approaches for subtree reconnection in genetic algorithm-based multicast routing. In *Anais do IX Congresso Brasileiro de Redes Neurais e Inteligência Computacional (IX CBRN / IC)*, pages 1–5, Ouro Preto, Brasil, 2009.
- [7] Marcos L. P. Bueno and Gina M. B. Oliveira. Algorithms to augment diversity and convergence in multiobjective multicast flow routing. In *11th IEEE Brazilian Symposium on Neural Networks (SBRN '10)*, pages 158–163, São Bernardo do Campo, Brasil, 2010.
- [8] Marcos L. P. Bueno and Gina M. B. Oliveira. Analyzing the effects of neighborhood crossover in multiobjective multicast flow routing problem. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC'10)*, pages 1–8, Istambul, Turkey, 2010.
- [9] Marcos L. P. Bueno and Gina M. B. Oliveira. Crossover heuristics for evolutionary models to solve mono-objective and pareto-based formulations of multicast routing problem. *Submitted to Journal of Heuristics*, pages 1–21, 2010.
- [10] Marcos L. P. Bueno and Gina M. B. Oliveira. Multicast flow routing: Evaluation of heuristics and multiobjective evolutionary algorithms. In *IEEE Congress on Evolutionary Computation (CEC'10)*, pages 1–8, Barcelona, Spain, 2010.
- [11] Marcos L. P. Bueno and Gina M. B. Oliveira. Multiobjective evolutionary algorithms and a combined heuristic for route reconnection applied to multicast flow routing. In

- 10th IEEE International Conference on Computer and Information Technology (CIT '10)*, pages 464–471, Bradford, UK, 2010.
- [12] Marcos L. P. Bueno and Gina M. B. Oliveira. Pareto-based optimization of multicast flows with qos and traffic engineering requirements. In *9th IEEE International Symposium on Network Computing and Applications (NCA '10)*, pages 257–260, Cambridge, USA, 2010.
- [13] Carlos A. Coello Coello. Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine, IEEE*, 1(1):28–36, 2006.
- [14] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.
- [15] Jorge Crichigno and Benjamín Barán. Multiobjective multicast routing algorithm for traffic engineering. In Ronald P. Luijten, Luiz A. DaSilva, and Antonius P. J. Engbersen, editors, *ICCCN*, pages 301–306. IEEE, 2004.
- [16] Xunxue Cui, Chuang Lin, and Yaya Wei. A multiobjective model for qos multicast routing based on genetic algorithm. In *ICCNMC '03: Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing*, page 49, Washington, USA, 2003. IEEE Computer Society.
- [17] Yogen K. Dalal and Robert M. Metcalfe. Reverse path forwarding of broadcast packets. *Commun. ACM*, 21(12):1040–1048, 1978.
- [18] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
- [19] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, pages 105–145. Springer Berlin Heidelberg, 2005. 10.1007/1-84628-137-7_6.
- [20] Stephen E. Deering and David R. Cheriton. Multicast routing in datagram internet-networks and extended lans. *ACM Trans. Comput. Syst.*, 8(2):85–110, 1990.
- [21] S E Dreyfus and R A Wagner. The steiner problem in graphs. *Networks*, (1):195–207, 1971.
- [22] Ramon Fabregat, Yezid Donoso, Benjamín Baran, Fernando Solano, and Jose L. Marzo. Multi-objective optimization scheme for multicast flows: a survey, a model and a MOEA solution. In *LANC '05: Proc. of the 3rd IFIP Latin American conf. on Networking*, pages 73–86, New York, USA, 2005. ACM.
- [23] C.M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

- [24] Cícero Garrozi and Aluizio Fausto Ribeiro Araújo. Multiobjective Genetic Algorithm for Multicast Routing. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 2513–2520, Vancouver, BC, Canada, July 2006. IEEE.
- [25] David E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, Massachusetts, 1989.
- [26] S L Hakimi. Steiner's problem in graphs and its implications. *networks*, (1):113–131, 1971.
- [27] Hisao Ishibuchi and Kaname Narukawa. Recombination of similar parents in emo algorithms. In *Proc. of Third International Conference in Evolutionary Multi-Criterion Optimization*, volume 3410 of *Lecture Notes in Computer Science*, pages 265–279, Guanajuato, Mexico, January 2005. Springer Berlin / Heidelberg.
- [28] Hisao Ishibuchi, Yuji Sakane, Noritaka Tsukamoto, and Yusuke Nojima. Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization. In *CEC'09: Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, pages 2508–2515, Piscataway, NJ, USA, 2009. IEEE Press.
- [29] Hisao Ishibuchi, Yuji Sakane, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization by nsga-ii and moea/d with large populations. In *SMC'09: Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics*, pages 1758–1763, Piscataway, NJ, USA, 2009. IEEE Press.
- [30] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *CEC'08: 2008 IEEE Congress on Evolutionary Computation*, pages 2424–2431. IEEE Press, 2008.
- [31] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. 1972.
- [32] Mifa Kim, Tomoyuki Hiroyasu, Mitsunori Miki, and Shinya Watanabe. Spea2+: Improving the performance of the strength pareto evolutionary algorithm 2. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 742–751, 2004.
- [33] Vachaspathi P. Kompella, Joseph C. Pasquale, and George C. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Trans. Netw.*, 1(3):286–292, 1993.
- [34] Mario Köppen and Kaori Yoshida. Substitute distance assignments in nsga-ii for handling many-objective optimization problems. In *EMO'07: Proceedings of the 4th international conference on Evolutionary multi-criterion optimization*, pages 727–741, Berlin, Heidelberg, 2007. Springer-Verlag.
- [35] Udi Manber. *Introduction to Algorithms: A Creative Approach*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
- [36] Ernesto Queiros Vieira Martins. On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2):236–245, May 1984.
- [37] J. Moy. Multicast Extensions to OSPF. RFC 1584 (Historic), March 1994.

- [38] Carlos A. S. Oliveira and Panos M. Pardalos. A survey of combinatorial optimization problems in multicast routing. *Comput. Oper. Res.*, 32(8):1953–1981, 2005.
- [39] Gina M. B. Oliveira and Paulo T. Araújo. Determining multicast routes with qos and traffic engineering requirements based on genetic algorithm. In *Proc. of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, volume 1, pages 665 – 669, Singapore, Dec 2004.
- [40] Gina M. B. Oliveira and Stéfano S. B. V. Vita. A Multi-Objective Evolutionary Algorithm with ϵ -dominance to Calculate Multicast Routes with QoS Requirements. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*, pages 1–9, Norway, May 2009.
- [41] Gina M. B. Oliveira and Stéfano S. B. V. Vita. Multi-objective multicast environments for qos routing and a new crossover with no maximum delay. In *Proceedings of IASTED International Conference on Artificial Intelligence and Applications (AIA2009)*, pages 140–145, Austria, February 2009.
- [42] José Maria A. Pangilinan and Gerrit K. Janssens. Evolutionary algorithms for the multiobjective shortest path problem. *World Academy of Science, Engineering and Technology*, (25):205–210, Jan 2007.
- [43] Joel Prieto, Benjamín Barán, and Jorge Crichigno. Multitree-multiobjective multicast routing for traffic engineering. In Max Bramer, editor, *IFIP AI*, volume 217 of *IFIP*, pages 247–256. Springer, 2006.
- [44] Ramamurthy Ravi. *Steiner trees and beyond: approximation algorithms for network design*. PhD thesis, Departament of Computer Science – Brown University, Providence, Rhode Island, September 1993.
- [45] C. P. Ravikumar and Rajneesh Bajpai. Source-based delay-bounded multicasting in multimedia networks. *Computer Communications*, 21(2):126–132, 1998.
- [46] George N. Rouskas and Ilia Baldine. Multicast routing with end-to-end delay and delay variation constraints. In *Selected Areas in Communications, IEEE Journal on Communications*, volume 15 of 3, pages 346–356, USA, April 1997. IEEE.
- [47] Günter Rudolph. Evolutionary search under partially ordered fitness sets. In *In Proceedings of the International Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent Systems (ISI 2001)*, pages 818–822. ICSC Academic Press, 1999.
- [48] Jason R. Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Massachusetts Institute of Technology, Massachusetts, USA, May 1995.
- [49] Eric D. Siegel. *Designing Quality of Service Solutions for the Enterprise*. John Wiley & Sons, Inc., New York, NY, USA, 1999.
- [50] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

- [51] H Takahashi and Matsuyama. An approximate solution for the steiner problem in graphs. *Mathematical Japonica*, (24):573–577, 1980.
- [52] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, 4th edition, 2002.
- [53] David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Wright-Patterson AFB, OH, 1999.
- [54] Stéfano S. B. V. Vita. Multiobjective genetic algorithms applied to multicast routing with quality of service. Master’s thesis, Universidade Federal de Uberlândia - Programa de Pós-Graduação em Ciência da Computação, 2009.
- [55] D. Waitzman, C. Partridge, and S.E. Deering. Distance Vector Multicast Routing Protocol. RFC 1075, November 1988.
- [56] Zheng Wang and Jon Crowcroft. Quality of service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 14:1228–1234, 1996.
- [57] Shinya Watanabe, Tomoyuki Hiroyasu, and Mitsunori Miki. Ncga: Neighborhood cultivation genetic algorithm for multi-objective optimization problems. In *GECCO Late Breaking Papers*, pages 458–465, 2002.
- [58] Sun Yijie and Shen Gongzhang. Improved nsga-ii multi-objective genetic algorithm based on hybridization-encouraged mechanism. *Chinese Journal of Aeronautics*, 21(6):540 – 549, 2008.
- [59] Kengo Yoshii, Tomoyuki Hiroyasu, Mitsunori Miki, and Akira Hashimoto. Effectiveness of neighborhood crossover in multiobjective genetic algorithm. In *Computational Intelligence*, pages 222–227, 2006.
- [60] Wang Zhengying, Shi Bingxin, and Zhao Erdun. Bandwidth-delay-constrained least-cost multicast routing based on heuristic ga. *Computer Communications*, 24(7-8):685–692, 2001.
- [61] E. Zitzler, M. Laumanns, and S. Bleuler. A Tutorial on Evolutionary Multiobjective Optimization. In X. Gandibleux et al., editors, *Metaheuristics for Multiobjective Optimisation*, Lecture Notes in Economics and Mathematical Systems. Springer, 2004.
- [62] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In *EUROGEN 2001*.
- [63] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8:173–195, 2000.
- [64] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, (3):257–271, 1999.

Apêndice A

Resultados Complementares

A.1 Formulação com dois objetivos

Tabela A.1: Médias e intervalos de confiança de 95% para as métricas avaliadas em cada instância no NSGA-II (Problemas **P1** a **P3**). Os valores de $\overline{gd}$ estão multiplicados por 10^3 e os de $\overline{\Delta}$ por 10.

P1			P2			P3		
$\overline{er}$	$\overline{gd}$	$\overline{\Delta}$	$\overline{er}$	$\overline{gd}$	$\overline{\Delta}$	$\overline{er}$	$\overline{gd}$	$\overline{\Delta}$
0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	1.3 $\pm$ 0.0	2.2% $\pm$ 1.7%	0.6 $\pm$ 0.0	3.8 $\pm$ 0.1	21.1% $\pm$ 3.8%	9.7 $\pm$ 2.5	4.4 $\pm$ 0.2
4.7% $\pm$ 5.4%	3.3 $\pm$ 3.8	1.3 $\pm$ 0.1	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	3.9 $\pm$ 0.2	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	3.0 $\pm$ 0.0
0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	1.3 $\pm$ 0.0	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	3.9 $\pm$ 0.1	1.7% $\pm$ 1.9%	0.6 $\pm$ 0.7	3.1 $\pm$ 0.1
0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	1.8 $\pm$ 0.0	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	3.4 $\pm$ 0.2
0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	2.0 $\pm$ 0.2	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	3.3 $\pm$ 0.2
0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	1.8 $\pm$ 0.0	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	3.3 $\pm$ 0.2
49.5% $\pm$ 3.7%	12 $\pm$ 1.2	1.5 $\pm$ 0.1	1.0% $\pm$ 0.9%	0.6 $\pm$ 0.1	3.9 $\pm$ 0.1	13.2% $\pm$ 1.1%	1.1 $\pm$ 0.1	5.4 $\pm$ 0.1
68.3% $\pm$ 4.0%	20 $\pm$ 2.1	3.0 $\pm$ 0.3	14.1% $\pm$ 3.4%	5.6 $\pm$ 1.2	4.9 $\pm$ 0.2	32.4% $\pm$ 6.0%	2.9 $\pm$ 0.6	6.1 $\pm$ 0.2
41.4% $\pm$ 4.4%	12 $\pm$ 2.0	1.5 $\pm$ 0.2	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	4.2 $\pm$ 0.1	7.2% $\pm$ 1.8%	0.6 $\pm$ 0.1	5.4 $\pm$ 0.1
7.2% $\pm$ 2.5%	1.2 $\pm$ 0.4	3.1 $\pm$ 0.1	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	2.6 $\pm$ 0.1	28.3% $\pm$ 3.7%	1.8 $\pm$ 0.2	5.8 $\pm$ 0.2
26.2% $\pm$ 4.4%	4.5 $\pm$ 1.3	4.2 $\pm$ 0.2	2.9% $\pm$ 2.7%	0.3 $\pm$ 0.3	4.3 $\pm$ 0.2	1.3% $\pm$ 0.8%	0.1 $\pm$ 0.1	5.7 $\pm$ 0.1
12.7% $\pm$ 3.5%	1.9 $\pm$ 0.6	3.2 $\pm$ 0.0	0.3% $\pm$ 0.5%	0.0 $\pm$ 0.1	2.8 $\pm$ 0.1	7.1% $\pm$ 3.0%	0.4 $\pm$ 0.2	5.6 $\pm$ 0.1
15.8% $\pm$ 3.5%	4.9 $\pm$ 1.4	3.6 $\pm$ 0.2	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	1.7 $\pm$ 0.0	14.9% $\pm$ 2.9%	1.8 $\pm$ 0.4	6.9 $\pm$ 0.1
8.7% $\pm$ 2.3%	4.0 $\pm$ 2.8	4.0 $\pm$ 0.3	1.6% $\pm$ 1.5%	0.2 $\pm$ 0.2	2.8 $\pm$ 0.2	0.4% $\pm$ 0.6%	0.1 $\pm$ 0.1	6.7 $\pm$ 0.1
7.1% $\pm$ 2.2%	5.1 $\pm$ 3.4	3.8 $\pm$ 0.3	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	1.8 $\pm$ 0.1	0.7% $\pm$ 1.2%	0.1 $\pm$ 0.1	6.9 $\pm$ 0.1
34.8% $\pm$ 2.5%	3.5 $\pm$ 1.1	7.7 $\pm$ 0.1	5.9% $\pm$ 0.9%	0.1 $\pm$ 0.1	4.2 $\pm$ 0.1	30.1% $\pm$ 4.5%	1.0 $\pm$ 0.2	8.0 $\pm$ 0.2
66.4% $\pm$ 10.8%	3.6 $\pm$ 0.6	7.1 $\pm$ 0.2	17.6% $\pm$ 5.4%	0.1 $\pm$ 0.2	6.0 $\pm$ 0.2	35.2% $\pm$ 6.0%	2.1 $\pm$ 0.4	8.0 $\pm$ 0.1
16.9% $\pm$ 5.4%	0.7 $\pm$ 0.3	7.5 $\pm$ 0.2	3.7% $\pm$ 1.4%	0.0 $\pm$ 0.1	4.9 $\pm$ 0.1	16.7% $\pm$ 4.6%	0.6 $\pm$ 0.2	8.3 $\pm$ 0.2

Tabela A.2: Médias e intervalos de confiança de 95% para as métricas avaliadas em cada instância com NSGA-II (Problemas **P4** e **P5**). Os valores de $\overline{gd}$ estão multiplicados por 10^3 e os de $\overline{\Delta}$ por 10.

P4			P5		
$\overline{er}$	$\overline{gd}$	$\overline{\Delta}$	$\overline{er}$	$\overline{gd}$	$\overline{\Delta}$
11.9% $\pm$ 3.9%	5.5 $\pm$ 1.8	4.5 $\pm$ 0.4	6.7% $\pm$ 4.1%	7.7 $\pm$ 4.7	1.2 $\pm$ 0.3
12.3% $\pm$ 6.1%	7.5 $\pm$ 4.1	3.4 $\pm$ 0.5	6.7% $\pm$ 4.1%	4.9 $\pm$ 3.0	2.0 $\pm$ 0.3
1.1% $\pm$ 1.8%	0.5 $\pm$ 0.8	2.9 $\pm$ 0.2	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.7 $\pm$ 0.0
0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	28.3% $\pm$ 7.6%	0.4 $\pm$ 9.6	0.9 $\pm$ 0.3
0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
15.2% $\pm$ 2.2%	5.1 $\pm$ 0.8	3.5 $\pm$ 0.2	17.3% $\pm$ 3.8%	6.8 $\pm$ 1.6	2.1 $\pm$ 0.3
45.6% $\pm$ 8.7%	0.1 $\pm$ 1.8	4.6 $\pm$ 0.3	4.4% $\pm$ 3.5%	1.6 $\pm$ 1.3	5.2 $\pm$ 0.2
7.9% $\pm$ 2.1%	3.6 $\pm$ 0.9	3.8 $\pm$ 0.1	0.7% $\pm$ 1.1%	0.5 $\pm$ 0.8	1.0 $\pm$ 0.0
13.8% $\pm$ 3.4%	2.1 $\pm$ 0.4	7.0 $\pm$ 0.1	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.8 $\pm$ 0.0
14.3% $\pm$ 4.8%	1.9 $\pm$ 0.5	7.1 $\pm$ 0.1	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.9 $\pm$ 0.2
8.6% $\pm$ 3.2%	1.4 $\pm$ 0.5	6.9 $\pm$ 0.1	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	0.8 $\pm$ 0.0
10.8% $\pm$ 2.1%	2.7 $\pm$ 0.7	4.0 $\pm$ 0.2	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	6.5 $\pm$ 0.2
4.3% $\pm$ 1.9%	0.7 $\pm$ 0.3	1.6 $\pm$ 0.1	6.7% $\pm$ 5.2%	0.5 $\pm$ 0.4	6.9 $\pm$ 0.2
4.2% $\pm$ 1.7%	0.6 $\pm$ 0.3	3.5 $\pm$ 0.1	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	6.2 $\pm$ 0.4
59.8% $\pm$ 3.8%	6.9 $\pm$ 2.5	8.1 $\pm$ 0.2	18.7% $\pm$ 2.2%	0.9 $\pm$ 0.2	2.8 $\pm$ 0.1
59.9% $\pm$ 7.6%	0.1 $\pm$ 3.8	7.5 $\pm$ 0.2	25.0% $\pm$ 10.2%	1.9 $\pm$ 0.8	3.8 $\pm$ 0.2
13.9% $\pm$ 1.9%	1.3 $\pm$ 0.3	8.6 $\pm$ 0.1	0.0% $\pm$ 0.0%	0.0 $\pm$ 0.0	2.5 $\pm$ 0.1

Tabela A.3: Médias e intervalos de confiança de 95% para as métricas avaliadas em cada instância no SPEA (Problemas **P1** a **P3**). Os valores de $\overline{gd}$ estão multiplicados por 10^3 e os de $\overline{\Delta}$ por 10.

P1			P2			P3		
$\overline{er}$	$\overline{gd}$	$\overline{\Delta}$	$\overline{er}$	$\overline{gd}$	$\overline{\Delta}$	$\overline{er}$	$\overline{gd}$	$\overline{\Delta}$
0.0% $\pm$ 0.0%	0 $\pm$ 0	1.3 $\pm$ 0	0.0% $\pm$ 0.0%	0 $\pm$ 0	3.9 $\pm$ 0	18.6% $\pm$ 3.5%	7.1 $\pm$ 1.3	4 $\pm$ 0.2
9.7% $\pm$ 6.8%	9.1 $\pm$ 6.3	1.4 $\pm$ 0.1	0.0% $\pm$ 0.0%	0 $\pm$ 0	3.8 $\pm$ 0	0.5% $\pm$ 0.8%	0.3 $\pm$ 0.5	3 $\pm$ 0
0.0% $\pm$ 0.0%	0 $\pm$ 0	1.3 $\pm$ 0	0.0% $\pm$ 0.0%	0 $\pm$ 0	3.9 $\pm$ 0	0.0% $\pm$ 0.0%	0 $\pm$ 0	3 $\pm$ 0
0.0% $\pm$ 0.0%	0 $\pm$ 0	0 $\pm$ 0	0.0% $\pm$ 0.0%	0 $\pm$ 0	1.8 $\pm$ 0	0.0% $\pm$ 0.0%	0 $\pm$ 0	3.2 $\pm$ 0.2
0.0% $\pm$ 0.0%	0 $\pm$ 0	0 $\pm$ 0	0.0% $\pm$ 0.0%	0 $\pm$ 0	2 $\pm$ 0.1	0.0% $\pm$ 0.0%	0 $\pm$ 0	3.2 $\pm$ 0.2
0.0% $\pm$ 0.0%	0 $\pm$ 0	0 $\pm$ 0	0.0% $\pm$ 0.0%	0 $\pm$ 0	1.8 $\pm$ 0	0.0% $\pm$ 0.0%	0 $\pm$ 0	3.2 $\pm$ 0.2
57.1% $\pm$ 3.6%	10.8 $\pm$ 0.6	1.4 $\pm$ 0.1	4.7% $\pm$ 2.4%	1.8 $\pm$ 0.9	3.8 $\pm$ 0.1	17.6% $\pm$ 1.7%	1.5 $\pm$ 0.1	5.1 $\pm$ 0.1
76.1% $\pm$ 2.4%	20.5 $\pm$ 1.3	2.7 $\pm$ 0.2	28.6% $\pm$ 3.4%	7.9 $\pm$ 0.5	5 $\pm$ 0.2	33.6% $\pm$ 4.4%	3.3 $\pm$ 0.7	5.8 $\pm$ 0.2
41.0% $\pm$ 4.0%	7.8 $\pm$ 1.1	1 $\pm$ 0.1	0.0% $\pm$ 0.0%	0 $\pm$ 0	4.1 $\pm$ 0	8.5% $\pm$ 1.8%	0.8 $\pm$ 0.2	5.2 $\pm$ 0.1
12.9% $\pm$ 3.4%	2.2 $\pm$ 0.6	3.1 $\pm$ 0.1	0.0% $\pm$ 0.0%	0 $\pm$ 0	2.6 $\pm$ 0.1	42.9% $\pm$ 2.9%	2.8 $\pm$ 0.2	5.9 $\pm$ 0.2
32.7% $\pm$ 3.5%	5.4 $\pm$ 0.6	3.7 $\pm$ 0.2	31.7% $\pm$ 5.0%	2.8 $\pm$ 0.4	4.4 $\pm$ 0.2	3.8% $\pm$ 1.4%	0.3 $\pm$ 0.1	5.8 $\pm$ 0.1
8.4% $\pm$ 3.2%	1.2 $\pm$ 0.4	3.1 $\pm$ 0.1	0.0% $\pm$ 0.0%	0 $\pm$ 0	2.6 $\pm$ 0.1	3.7% $\pm$ 1.3%	0.3 $\pm$ 0.1	5.6 $\pm$ 0.1
16.3% $\pm$ 3.1%	6.4 $\pm$ 1.5	3.7 $\pm$ 0.2	0.0% $\pm$ 0.0%	0 $\pm$ 0	1.7 $\pm$ 0	18.5% $\pm$ 2.4%	2.3 $\pm$ 0.4	6.8 $\pm$ 0.1
13.4% $\pm$ 1.7%	4.3 $\pm$ 1.2	3.9 $\pm$ 0.2	0.0% $\pm$ 0.0%	0 $\pm$ 0	2.4 $\pm$ 0.2	10.8% $\pm$ 2.8%	1.6 $\pm$ 0.6	6.6 $\pm$ 0.1
13.3% $\pm$ 1.1%	3.3 $\pm$ 0.4	3.3 $\pm$ 0.1	0.0% $\pm$ 0.0%	0 $\pm$ 0	1.7 $\pm$ 0	13.1% $\pm$ 2.4%	1.6 $\pm$ 0.5	6.7 $\pm$ 0.1
42.2% $\pm$ 2.4%	11.3 $\pm$ 2.1	7.2 $\pm$ 0.1	1.6% $\pm$ 0.7%	0.1 $\pm$ 0	4.1 $\pm$ 0.1	69.3% $\pm$ 4.3%	3.6 $\pm$ 0.4	8.1 $\pm$ 0.1
99.2% $\pm$ 1.4%	14.1 $\pm$ 2.8	7.2 $\pm$ 0.3	38.4% $\pm$ 4.5%	1.4 $\pm$ 0.2	5.9 $\pm$ 0.2	58.1% $\pm$ 7.2%	6.8 $\pm$ 3.1	8.3 $\pm$ 0.2
41.7% $\pm$ 4.5%	3.4 $\pm$ 0.5	7.9 $\pm$ 0.1	0.3% $\pm$ 0.4%	0 $\pm$ 0	4.1 $\pm$ 0.1	41.8% $\pm$ 6.0%	1.5 $\pm$ 0.2	8.4 $\pm$ 0.2

Tabela A.4: Médias e intervalos de confiança de 95% para as métricas avaliadas em cada instância com SPEA (Problemas **P4** e **P5**). Os valores de $\overline{gd}$ estão multiplicados por 10^3 e os de $\overline{\Delta}$ por 10.

P4			P5		
$\overline{er}$	$\overline{gd}$	$\overline{\Delta}$	$\overline{er}$	$\overline{gd}$	$\overline{\Delta}$
17.5% $\pm$ 3.5%	8 $\pm$ 1.6	5 $\pm$ 0.4	5.6% $\pm$ 3.8%	6.4 $\pm$ 4.4	1.1 $\pm$ 0.3
18.0% $\pm$ 6.9%	15.1 $\pm$ 5.8	3.7 $\pm$ 0.3	6.7% $\pm$ 4.1%	4.9 $\pm$ 3	1.6 $\pm$ 0.3
3.3% $\pm$ 2.6%	1.5 $\pm$ 1.2	3.5 $\pm$ 0.3	0.0% $\pm$ 0.0%	0 $\pm$ 0	0.7 $\pm$ 0
0.0% $\pm$ 0.0%	0 $\pm$ 0	0 $\pm$ 0	20.0% $\pm$ 7.5%	25.3 $\pm$ 9.5	0.6 $\pm$ 0.2
0.0% $\pm$ 0.0%	0 $\pm$ 0	0 $\pm$ 0	0.0% $\pm$ 0.0%	0 $\pm$ 0	0 $\pm$ 0
0.0% $\pm$ 0.0%	0 $\pm$ 0	0 $\pm$ 0	0.0% $\pm$ 0.0%	0 $\pm$ 0	0 $\pm$ 0
16.1% $\pm$ 2.0%	3.8 $\pm$ 0.6	3.6 $\pm$ 0.2	21.3% $\pm$ 2.2%	11.8 $\pm$ 1.5	2.8 $\pm$ 0.2
55.8% $\pm$ 6.5%	12 $\pm$ 1.3	4.5 $\pm$ 0.3	13.9% $\pm$ 5.3%	4.6 $\pm$ 2.5	4.7 $\pm$ 0.2
8.0% $\pm$ 2.7%	1.6 $\pm$ 0.5	3.5 $\pm$ 0.2	0.0% $\pm$ 0.0%	0 $\pm$ 0	1 $\pm$ 0
18.6% $\pm$ 3.4%	2.5 $\pm$ 0.4	7 $\pm$ 0.1	2.2% $\pm$ 2.5%	0.2 $\pm$ 0.3	0.8 $\pm$ 0.1
35.8% $\pm$ 3.8%	4 $\pm$ 0.4	6.8 $\pm$ 0.1	2.2% $\pm$ 2.5%	0.2 $\pm$ 0.3	0.8 $\pm$ 0.1
17.6% $\pm$ 3.6%	2 $\pm$ 0.3	6.9 $\pm$ 0.1	0.0% $\pm$ 0.0%	0 $\pm$ 0	0.8 $\pm$ 0
13.3% $\pm$ 2.0%	4.7 $\pm$ 1.4	4 $\pm$ 0.2	0.0% $\pm$ 0.0%	0 $\pm$ 0	5.8 $\pm$ 0.4
15.9% $\pm$ 3.6%	3.3 $\pm$ 0.9	2 $\pm$ 0.2	5.6% $\pm$ 3.8%	0.4 $\pm$ 0.3	3.5 $\pm$ 0.5
13.1% $\pm$ 2.3%	2.3 $\pm$ 0.4	3.4 $\pm$ 0.2	0.0% $\pm$ 0.0%	0 $\pm$ 0	5.2 $\pm$ 0.4
61.4% $\pm$ 2.4%	13.4 $\pm$ 2.3	7.7 $\pm$ 0.2	11.3% $\pm$ 3.0%	0.5 $\pm$ 0.1	2.7 $\pm$ 0.1
97.2% $\pm$ 2.6%	25.9 $\pm$ 4	7.1 $\pm$ 0.4	60.0% $\pm$ 6.9%	6 $\pm$ 0.7	4.5 $\pm$ 0.3
40.0% $\pm$ 4.1%	3.6 $\pm$ 0.8	8.4 $\pm$ 0.1	0.0% $\pm$ 0.0%	0 $\pm$ 0	2.5 $\pm$ 0

Apêndice B

Algorithms to Augment Diversity and Convergence in Multiobjective Multicast Flow Routing

Neste apêndice está apresentado o artigo *Algorithms to Augment Diversity and Convergence in Multiobjective Multicast Flow Routing* [7], aceito para publicação no 11º Simpósio Brasileiro de Redes Neurais (SBRN), a ser realizado em Outubro de 2010 em São Bernardo do Campo - SP.

Algorithms to Augment Diversity and Convergence in Multiobjective Multicast Flow Routing

Marcos L. P. Bueno
*Faculty of Computer Science
 Federal University of Uberlandia
 Uberlandia, Brazil
 marcoslbueno@gmail.com*

Gina M. B. Oliveira
*Faculty of Computer Science
 Federal University of Uberlandia
 Uberlandia, Brazil
 gina@facom.ufu.br*

Abstract—Multicast transmission corresponds to send data to several destinations, often involving requirements of Quality of Service (QoS) and Traffic Engineering (TE). These multiple requirements lead to the need of optimizing a set of conflicting objectives subject to constraints. Starting from the well-known evolutionary algorithm SPEA2, two formulations for the Routing problem were considered, minimizing four objectives - maximum link utilization, total cost, maximum end-to-end delay and hops count - subject to a link capacity constraint. The key investigation performed here is about the incorporation of a mechanism to reduced repeated individuals along the population, evaluating three algorithms for such task. One of them (f_{tm}) is proposed on this study, whose results in six instances of the problem with five metrics to evaluate convergence and diversity goals, indicates that the evolutionary model based on SPEA2 using our algorithm had returned the better results.

Keywords—multicast routing, genetic algorithm, evolutionary multiobjective optimization, steiner tree problem, path relinking.

I. INTRODUCTION

In computer networks, a multicast transmission corresponds to send data to a selective number of destinations. Due to the increment in the last years in the use of new point-to-multipoint applications, such as video, voice and online games, multicast routing algorithms have recently been intensively investigated [1], [2]. Often, such type of routing also involves the concept of Quality of Service (QoS) [1], since point-to-multipoint applications can have different requirements as maximum delay, minimum total cost and minimum bandwidth. On the other hand, Traffic Engineering (TE) is concerned with optimizing the network utilization aiming to reduce congestion bottlenecks, improve resource use and also to provide adequate QoS for final users [3]. When QoS and TE requirements are also being considered in multicast routing, it leads to the need of optimizing a set of objectives subject to some constraints.

In Multicast Flow Routing Problem (MFRP), given a connected undirected graph, with vertices representing hosts and edges representing network links, functions representing characteristics of the network links (cost, delay, capacity and

current traffic), and a new multicast flow ϕ , we want to calculate a set of Pareto-optimal rooted trees T of G to carry ϕ into G , starting from a source vertex and passing by a subset of vertices called terminals. The kind of multicast route we want to produce is affected by QoS and TE requirements, which establish objectives to minimize and/or maximize and constraints to be attended. Two formulations of MFRP were considered in this work, minimizing the following objective functions: maximum link utilization, total cost, maximum end-to-end delay and hops count, subject to a link capacity constraint.

The MFRP can be considered as an extension of the Steiner Tree Problem (STP), a problem to find a tree with minimum cost spanning a subset of vertices, which Karp [4] showed to be NP-hard. Thus, many techniques have been used to tackle MFRP, such as heuristics, approximation algorithms and various artificial intelligence techniques [5], [6], [7], [8], [9], [10], [11]. Evolutionary algorithms represent one of the most investigated artificial intelligence techniques to handle this problem. In this work, we use multiobjective evolutionary algorithms (MOEAs) to approach MFRP, since they have been used in several works ([12], [13], [14], [15], [16], [17], [18], [19], [20]) with promising results.

In this work, we present and evaluate three algorithms for a mechanism called filter [15] to increase population diversity and convergence. Two of them are from literature [12], [16], while the third one is an algorithm we propose, based on the well-known Takahashi-Matsuyama heuristic [10]. Experimentally, using five metrics for assess convergence and diversity on six network instances of MFRP, we were able to observe that filter mechanism can bring advantages to our SPEA2-based model.

The remainder of this work is organized as follows: Section II presents the formulations of multicast routing problem. Section III present the proposed evolutionary model and its operators, while Section IV exposes the three algorithms for filter. The experimental results are discussed in Section V, while the final remarks and future research are presented on Section VI.

II. MULTICAST FLOW ROUTING PROBLEM

A. Basic Concepts

In this work, routing with QoS and TE requirements is considered under a multiobjective optimization view. In that case, it may not be possible to optimize all objectives simultaneously, because when optimizing one or more objectives, we could deteriorate another(s). If that is the case, it may not exist a solution that has, simultaneously, optimal values for each objective. Due to this conflict, we use a strict partial order relation known as **Pareto dominance**: given two n -dimensional decision vectors a and b and their respective objective vectors $f(a) = \{f_1(a), \dots, f_m(a)\}$ and $f(b) = \{f_1(b), \dots, f_m(b)\}$, we say a dominates b ($a \prec b$, in symbols) iff: (i) a is not worse than b in any objective; (ii) a is strictly better than b in at least one objective.

In a Pareto dominance relation, for any two points a and b , may not occur $a \prec b$ nor $b \prec a$. Inside a set of solutions, those for which there is no other that dominates them are called nondominated solutions. Thus, in a global sense, if P is the entire decision space, then a feasible solution $x^* \in P$ is called Pareto-optimal solution iff $\nexists x \in P, x \prec x^*$. The set composed by all Pareto-optimal solution is called global Pareto-optimal set, denoted by P^* .

Multiobjective Evolutionary Algorithms (MOEAs) are evolutionary algorithms designed to approach problems that includes more than one objective to optimize, usually considering the Pareto dominance relation. The main difference between a classic genetic algorithm (GA) and a MOEA usually lies on fitness assignment, where Pareto dominance is commonly used in different ways. This work deals with SPEA2, an algorithm from the second generation of MOEAs, which was mainly characterized by the use of elitism.

Starting from the standard GA [21], a lot of innovations and improvements have been considered over this evolutionary paradigm. This work evaluates two algorithms from literature and proposes a new one for a mechanism called filter to provide more diversity to the population. The filter main idea is that, after crossover and mutation, if a generated child is already present in the current population, an algorithm applies changes on it according to some heuristic. As we are going to expose in Section IV, the algorithms for filter are used to perform small changes in the repeated child (except the filter which replaces it by a random one), i.e., it follows a more local search sense.

B. Problem Definition

Considering a multicast context, the network topology is modeled as a connected undirected graph $G = (V, E, W)$, such that V represents a set of nodes or vertices, E a set of edges and W a tuple of weights. We define $r \in V$ as the source or root node and a set $D \subset V$ as the multicast group, in which each node is called destination or terminal. For each

edge e in G , there is a tuple $W = (c(e), d(e), z(e), t(e))$, corresponding to an edge's cost, delay, capacity and current traffic, respectively, which are defined on $\mathbf{R}$. Since each edge has a capacity, we have the following restriction over the traffic passing by it at any moment: $t(e) \leq z(e), \forall e \in E$. Finally, there is a traffic request $\phi \in \mathbf{R}$ that we want to load on G .

Thus, in Multiobjective Multicast Flow Routing Problem (MFRP), given an instance (G, r, D, ϕ) , we want to calculate trees $T = (V_T, E_T)$ of G , rooted in r , with $\{r\} \cup D \subseteq V_T$, such that T will allow the loading of ϕ into G . In order to load ϕ into G , every feasible candidate T (also called multicast tree) must attend the constraint $t(e) + \phi \leq z(e), \forall e \in E_T$. We also want to minimize four objective functions for each T , namely: maximum link utilization or α , total cost, maximum end-to-end delay and hops count, given by Eqs. 1, 2, 3 and 4, respectively, and a capacity constraint given by Eq. 5.

Minimize

$$\alpha(T) = \max \left\{ \frac{t(e) + \phi}{z(e)} \right\} \quad (1)$$

$$\text{cost}(T) = \phi \sum c(e) \quad (2)$$

$$d_m(T) = \max \{ \text{delay}(t_1), \dots, \text{delay}(t_{|D|}) \} \quad (3)$$

$$\text{hops}(T) = |E_T| \quad (4)$$

Subject to

$$t(e) + \phi \leq z(e) \quad (5)$$

where $e \in E_T$ and $\text{delay}(t_i) = \sum_{e \in \text{path}(r, t_i)} d(e), \forall t_i \in D$. Thus, given the two formulations for MFRP below (Multiobjective Optimization Problems or MOPs), we want to calculate the Pareto-optimal set of solutions, such that:

mop3: minimize Eqs. 1, 2 and 3, subject to Eq. 5.

mop4: minimize Eqs. 1, 2, 3 and 4, subject to Eq. 5.

An instance of MFRP is shown on Figure 1, consisting on a network with 18 nodes, in which node 0 is the source and nodes 6, 10, 13, 15, 17 are the terminal ones. For each edge e , the weights cost, delay and current traffic $(c(e), d(e), t(e))$ are explicit presented. On the other hand, capacity $z(e)$ is considered fixed for all the edges ($z(e) = 12$ Mbps, $\forall e \in E$) and it is omitted in the figure. A Pareto-optimal for such G , D , r and $\phi = 2.0$ Mbps on *mop3* and *mop4* is showed by the tree on right side of the figure, along with its objective values.

III. EVOLUTIONARY MODELS FOR MFRP

A. Initial population

Each individual is codified directly as generic rooted tree as shown on the right side of Figure 1, being generated by a random search algorithm, in which nodes are inserted in the

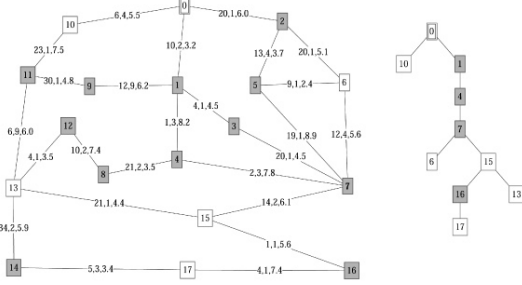


Figure 1. A MFRP instance on left ($\phi = 2.0$), and a Pareto-optimal solution on right, whose objectives are $\alpha(T) = 0.85$, $cost(T) = 71 * 2.0$, $d_m(T) = 12$ and $hops(T) = 9$.

tree in a random way until all destination nodes are present. Next, a pruning is applied over the tree, so that every leaf node is a destination. Notice that this algorithm has the same worst case complexity of breadth-first or depth-first search, i.e., $O(|E| + |V|)$ since the only difference is that we choose the next neighbor to expand in a random way, instead of using a queue or a stack.

B. Crossover

The crossover operator presented here is based on the general scheme proposed in [12] and [18]. From two parents, a child inherits their features following this idea: (i) identification of common edges between two parents, resulting in a forest of subtrees to be reconnected; (ii) random choice of G weight to be used along reconnections; (iii) reconnection of subtrees of the forest, resulting on a new multicast tree containing all destination nodes; (iv) pruning of useless nodes.

In subtrees reconnection we aim to build a new multicast tree T from a forest of subtrees $s = \{s_1, s_2, \dots, s_k\}$. Each s_i is obtained by a procedure that calculates the common edges of two trees. Moreover, if the root r or any destination is not in those common edges, we build and add isolated subtrees, i.e., subtrees that contains only one vertex, into s . Thus, given s , we have to make $|s| - 1$ reconnections. In the first reconnection, we select two subtrees s_i and s_j not yet reconnected and calculate a set p of edges such that $s_i \cup p \cup s_j$ will become a connected subgraph of G . In the following reconnections, we select another subtree and calculate a new set p between it and the subgraph obtained on the previous reconnection. An important decision that affects algorithm convergence to optimal solutions is how to choose p edges; we explored this aspect proposing initially two heuristics (h_{dij} and h_{rnd}) in a previous study [20], and a third heuristic (h_{mix}) in [22], which combines the features of the two formers.

In this work, we have adopted heuristic h_{mix} to be used on crossover, since in a previous work [22] was showed that it returned better results than the formers in various exper-

iments with $mop4$. Thus, h_{mix} is a composition of h_{dij} and h_{rnd} in the following way: at the beginning of a crossover, after calculation of common subtrees and random choice of edge's weight (cost, e.g.), we randomly choose which of the two heuristics will be used to perform reconnections of subtrees forest s . If h_{dij} is selected to reconnect the subtrees in s , it will be done using Dijkstra's algorithm in a way that no loops can occur (as showed on [20], [22]). On the other hand, if h_{rnd} is picked, the algorithm add random edges to reconnect subtrees in s , having as final result a connected subgraph representing a valid solution to the problem. As showed [20], [22], h_{mix} have $O(|V||E| \log |V|)$ complexity.

C. Mutation

Mutation operator consists in a random pick of a set of nodes from a multicast tree and remove their connections, i.e., for each node in such set, remove its edge with its parent and its edges with its children. The result will be a forest of subtrees, which ones must be reconnected. In this work, we used the same reconnection algorithm (h_{mix}) used in crossover operator.

IV. ALGORITHMS TO AUGMENT DIVERSITY AND CONVERGENCE

Since one of the main difficulties to GAs converge is the lack of diversity on the population, Oliveira & Araujo proposed in [15] a mechanism called filter as an attempt to augment the diversity of the generated offspring along the generations. It works changing each generated child whenever it is already present on the population; in SPEA2, we considered such comparison with individuals from E and the current children in P' . We evaluate three algorithms for the filter: two of them from the literature - f_{mut} algorithm, proposed by Ravikumar & Bajpai [12], which is the same one used for disconnect nodes in mutation, and f_{rnd} algorithm, which simply swaps a repeated child with a random one (such strategy was used in a different way in [16]). The third one - f_{tm} algorithm - is proposed on the present work, being based on Takahashi-Matsuyama heuristic [10]. Alg. 1 describes f_{tm} .

Algorithm 1: Terminals relinking based on TM algorithm

Input: $T = (V_T, E_T)$: a multicast tree with terminal set D_T .
Output: $T = (V_T, E_T)$: a new multicast tree with terminal set D_T .

- 1 $D' :=$ randomly pick 20% of leaf nodes in D_T ;
- 2 $D_T := D_T \setminus D'$;
- 3 $prune(T)$;
- 4 **while** $D' \neq \emptyset$ **do**
- 5 $sp := \text{shortestPath}(V_T, D')$;
- 6 $T := T \cup sp$;
- 7 $D' := D' \setminus \{sp.end\}$;
- 8 **return** T ;

The proposed f_{tm} algorithm starts randomly picking a set D' of nodes that are terminals and leaf nodes to be

disconnected; we set that amount as 20% of D_T here. Then, we change the terminal set definition of T to exclude D' from D_T , in order to apply a prune procedure over T to remove the terminals of D' and eventually other non-terminal nodes that may appear as leaf nodes after that prune. Now, we need to reconnect the terminals of D' with T , which is done using the same idea of TM heuristic: at each iteration of the loop, we select the closest terminal in D' to T , adding such shortest path to T and removing that terminal from D' . This process continues until all terminals from D' are reconnected to T . Algorithm f_{tm} has $O(|D||E|\log|V|)$ time, since it is dominated by shortest path calculation ($O(|E|\log|V|)$), which is executed $|D'|$ times at most inside the while loop (note that $|D'|$ is bounded by $|D|$).

V. EXPERIMENTAL ANALYSIS

Six instances of MFRP previously used on several works regarding Routing and Steiner problems were used to evaluate the effects of the investigated filters. They are described on Table I: net A comes from [20] while nets B to F are from OR-Library [6]. Net A has fixed fixed values for cost and delay on each edge, while OR-Library instances have only costs defined in [6]; we randomly generated values for delay on those nets in the range of $[1, 15]$. As four weights were used per edge, random values were generated for the other two into the intervals $[1000; 1500]$ Kbps and $[0; 500]$ Kbps, for capacity and current traffic flowing through each edge, respectively. Since an instance for MFRP is given by a tuple (G, r, D, ϕ) , the multicast flow ϕ was set to 300 Kbps for every instance.

Table I
MFRP INSTANCES USED IN THIS WORK.

net	$ V $	$ E $	$ D $	$ P^* $ in $mop3$	$ P^* $ in $mop4$
A	33	106	10	48	122
B	50	126	13	23	40
C	75	188	19	34	72
D	75	188	38	16	60
E	75	300	13	27	42
F	100	250	17	11	13

Considering that P denotes the set of solutions obtained by an algorithm, and P^* the global Pareto set, three metrics to assess convergence were used: *error rate* (er) [23], *generational distance* (gd) [23] and *pareto subset* (ps , which is $(1 - er) \times |P|$). The maximum spread metric ($m3$) [24] was used to evaluate diversity. To directly compare two sets of solutions in terms of dominance, the two set coverage metric (cs) [24] was used and will be defined in proper moment along this section.

The GA parameters were set to: mutation probability of 10%, node disconnection rate in mutation and f_{mut} of 20%, and $N_p = 60, N_g = 100$, where N_p and N_g denotes population size and number of generations, respectively. In each case, 100 executions were done to obtain the averages of each metric, whose results are summarized in Tab. II. The

four variations of SPEA2 (SPEA2 without filter or SPEA2-f, SPEA2+ f_{mut} , SPEA2+ f_{tm} , SPEA2+ f_{rnd}) are grouped in two major groups (one for each mop).

Looking at the four metrics, the best behavior on them, i.e., first and second best means on each metric/net, is shifted to the regions of the filters in almost all cases, for both mops. This shows that filter mechanism can bring improvements on convergence and diversity on those metrics. Among the three filters, f_{tm} reaches the best behavior on er and gd metrics in a case-to-case analysis. Looking at the averages for er , f_{tm} also reaches the best values on $mop3$ and $mop4$, situated around 18% and 10%, respectively, against 22% and 13% obtained by SPEA2-f. In other words, among the non-dominated solutions obtained with SPEA2+ f_{tm} , 82% ($mop3$) and 90% ($mop4$) were indeed Pareto-optimal ones, while that percents were 78% and 87% for SPEA2-f. Such results on er and gd means that: (i) the nondominated solutions returned by f_{tm} are more likely to be Pareto-optimal ones; (ii) the nondominated solutions which are not Pareto-optimal ones are, in turn, “closer” to P^* in the sense of distance in the objective space, which is desirable when $gd(P) = 0$ (or, equivalently, $er = 0$ or $P \subseteq P^*$) was not achieved. This argument can be reinforced by ps metric we provided on this study, showing that besides f_{tm} return more Pareto-optimal solutions in percent terms, the absolute quantities for that was also higher, on most cases. Looking at the data fluctuation around the mean values, the best confidence intervals (CIs) are also located on f_{tm} and f_{mut} regions (primarily, on f_{tm} one), reinforcing the good performance of the filter mechanism, specially f_{tm} .

One may find awkward that er values in $mop3$ were higher than those in $mop4$. However, it should be pointed out the differences between the sizes of non-dominated sets returned on each mop for the same instance, which were close to the size of Pareto-optimal sets too. Thus, since the size of Pareto sets in $mop3$ are smaller than those in $mop4$, whenever an algorithm returns a small number of dominated solutions, it can lead to larger er values; for instance, on net B f_{tm} returned approximately 3 dominated solutions giving an $er = 14.3\%$ in $mop3$, while that value in $mop4$ was 6.5% when f_{tm} returned almost 4 dominated solutions. ps metric can help providing more info along with er to clarify such situations.

Analyzing the diversity metric $m3$, clearly there is a trade-off between the four MOEAs, since they reached similar values on each net. However, it seems f_{tm} did well on both mops, specially on $mop4$, differently than f_{mut} , which seemed to achieve not so good results on $mop3$. Thus, it means that, in general, f_{tm} covered more extension of the objective space. The same previous reasoning about the dispersion of the means can be applied here: f_{tm} reached slightly better results than f_{mut} on CIs.

The last metric to be analyzed - two set coverage - is designed to directly compare two sets of solutions in terms

Table II
MEANS AND 95% CONFIDENCE INTERVALS FOR THE METRICS ON *mop3* AND *mop4*. *gd* VALUES ARE MULTIPLIED BY 10^3 , WHILE THOSE FROM *m3* ARE BY 10.

filter	net	<i>mop3</i>				<i>mop4</i>			
		<i>er</i>	<i>gd</i>	<i>ps</i>	<i>m3</i>	<i>er</i>	<i>gd</i>	<i>ps</i>	<i>m3</i>
$-f$	A	24.0% $\pm$ 2.1%	2.5 $\pm$ 0.2	32.0 $\pm$ 0.9	8.1 $\pm$ 0.2	15.5% $\pm$ 1.5%	2.5 $\pm$ 0.3	50.7 $\pm$ 0.9	8.7 $\pm$ 0.1
	B	19.6% $\pm$ 3.4%	1.3 $\pm$ 0.2	16.6 $\pm$ 0.7	6.9 $\pm$ 0.0	7.7% $\pm$ 1.9%	0.7 $\pm$ 0.2	31.2 $\pm$ 0.9	7.3 $\pm$ 0.0
	C	20.6% $\pm$ 3.2%	3.9 $\pm$ 0.6	19.1 $\pm$ 0.9	8.0 $\pm$ 0.2	10.7% $\pm$ 1.5%	0.9 $\pm$ 0.1	45.5 $\pm$ 1.0	8.0 $\pm$ 0.0
	D	13.2% $\pm$ 2.7%	0.6 $\pm$ 0.1	13.8 $\pm$ 0.4	5.2 $\pm$ 0.0	1.4% $\pm$ 0.5%	0.2 $\pm$ 0.1	52.7 $\pm$ 0.8	5.1 $\pm$ 0.2
	E	35.0% $\pm$ 2.3%	4.7 $\pm$ 0.6	14.4 $\pm$ 0.5	8.7 $\pm$ 0.1	25.5% $\pm$ 2.0%	4.2 $\pm$ 0.5	21.8 $\pm$ 0.6	8.9 $\pm$ 0.1
	F	15.1% $\pm$ 3.2%	2.9 $\pm$ 0.9	9.7 $\pm$ 0.3	6.3 $\pm$ 0.0	16.0% $\pm$ 3.3%	3.2 $\pm$ 1.1	10.3 $\pm$ 0.3	6.4 $\pm$ 0.1
f_{mut}	A	20.9% $\pm$ 1.8%	2.0 $\pm$ 0.2	34.5 $\pm$ 0.8	8.2 $\pm$ 0.1	16.6% $\pm$ 1.3%	2.4 $\pm$ 0.2	50.0 $\pm$ 0.8	8.7 $\pm$ 0.1
	B	17.6% $\pm$ 2.8%	1.2 $\pm$ 0.2	17.1 $\pm$ 0.6	6.9 $\pm$ 0.0	5.6% $\pm$ 1.3%	0.5 $\pm$ 0.2	32.3 $\pm$ 0.7	7.3 $\pm$ 0.0
	C	13.9% $\pm$ 2.2%	3.0 $\pm$ 0.5	22.3 $\pm$ 0.7	7.9 $\pm$ 0.2	7.1% $\pm$ 1.0%	0.8 $\pm$ 0.3	50.6 $\pm$ 1.1	8.2 $\pm$ 0.1
	D	10.9% $\pm$ 2.9%	0.5 $\pm$ 0.1	14.4 $\pm$ 0.4	5.2 $\pm$ 0.0	1.2% $\pm$ 0.7%	0.2 $\pm$ 0.1	53.0 $\pm$ 0.7	5.3 $\pm$ 0.0
	E	35.1% $\pm$ 2.6%	4.2 $\pm$ 0.5	14.6 $\pm$ 0.6	8.6 $\pm$ 0.1	21.3% $\pm$ 2.0%	4.0 $\pm$ 0.7	23.8 $\pm$ 0.5	9.0 $\pm$ 0.1
	F	11.9% $\pm$ 3.3%	2.7 $\pm$ 0.9	10.0 $\pm$ 0.3	6.3 $\pm$ 0.0	11.4% $\pm$ 3.0%	1.6 $\pm$ 0.5	10.8 $\pm$ 0.3	6.4 $\pm$ 0.0
f_{tm}	A	17.6% $\pm$ 1.3%	2.1 $\pm$ 0.1	34.8 $\pm$ 0.6	8.0 $\pm$ 0.2	14.8% $\pm$ 1.3%	2.4 $\pm$ 0.2	51.1 $\pm$ 0.8	8.7 $\pm$ 0.1
	B	14.3% $\pm$ 1.6%	1.0 $\pm$ 0.2	17.7 $\pm$ 0.4	6.9 $\pm$ 0.0	6.4% $\pm$ 1.6%	0.7 $\pm$ 0.3	32.6 $\pm$ 0.7	7.4 $\pm$ 0.0
	C	10.5% $\pm$ 1.4%	2.3 $\pm$ 0.4	24.5 $\pm$ 0.6	7.9 $\pm$ 0.1	6.5% $\pm$ 1.0%	0.7 $\pm$ 0.1	51.8 $\pm$ 0.6	8.2 $\pm$ 0.0
	D	13.9% $\pm$ 3.9%	0.6 $\pm$ 0.1	13.8 $\pm$ 0.6	5.2 $\pm$ 0.0	0.6% $\pm$ 0.4%	0.1 $\pm$ 0.1	53.7 $\pm$ 0.5	5.3 $\pm$ 0.0
	E	31.3% $\pm$ 2.8%	4.1 $\pm$ 0.5	15.0 $\pm$ 0.5	8.6 $\pm$ 0.1	20.8% $\pm$ 2.3%	4.0 $\pm$ 0.5	24.1 $\pm$ 0.7	9.0 $\pm$ 0.1
	F	15.8% $\pm$ 3.3%	2.4 $\pm$ 0.7	9.6 $\pm$ 0.3	6.2 $\pm$ 0.0	10.8% $\pm$ 2.7%	2.7 $\pm$ 1.0	11.0 $\pm$ 0.3	6.4 $\pm$ 0.0
f_{rnd}	A	23.3% $\pm$ 1.8%	2.2 $\pm$ 0.2	32.3 $\pm$ 0.7	8.3 $\pm$ 0.2	16.8% $\pm$ 1.2%	2.5 $\pm$ 0.2	49.9 $\pm$ 0.7	8.7 $\pm$ 0.1
	B	19.8% $\pm$ 2.8%	1.4 $\pm$ 0.2	16.8 $\pm$ 0.6	6.9 $\pm$ 0.0	6.5% $\pm$ 1.7%	0.6 $\pm$ 0.3	31.8 $\pm$ 0.8	7.3 $\pm$ 0.0
	C	23.0% $\pm$ 3.4%	3.6 $\pm$ 0.6	19.1 $\pm$ 1.0	7.9 $\pm$ 0.2	13.2% $\pm$ 1.8%	0.9 $\pm$ 0.1	44.2 $\pm$ 1.3	8.1 $\pm$ 0.1
	D	11.8% $\pm$ 2.4%	0.5 $\pm$ 0.1	14.1 $\pm$ 0.4	5.2 $\pm$ 0.0	3.8% $\pm$ 4.1%	0.2 $\pm$ 0.1	50.6 $\pm$ 2.4	5.1 $\pm$ 0.2
	E	34.8% $\pm$ 2.8%	4.3 $\pm$ 0.6	14.5 $\pm$ 0.6	8.6 $\pm$ 0.1	26.2% $\pm$ 1.6%	4.0 $\pm$ 0.4	21.8 $\pm$ 0.5	8.8 $\pm$ 0.1
	F	10.2% $\pm$ 3.4%	1.5 $\pm$ 0.9	10.0 $\pm$ 0.3	6.3 $\pm$ 0.0	11.2% $\pm$ 3.7%	2.2 $\pm$ 1.0	10.9 $\pm$ 0.4	6.4 $\pm$ 0.0

of dominance. Given two sets of solutions X and Y , then $cs(X, Y)$ will return the percent of solutions from Y that are dominated by at least one solution from X . Formally, $cs(X, Y) := |\{y \in Y : \exists x \in X \wedge x \prec y\}|/|Y|$. Note that $cs(X, Y)$ is not symmetric to $cs(Y, X)$. To emphasize the usage of filter, we used set cover to compare SPEA2- f (set X) and SPEA2+ f_{tm} (set Y) on each instance, resulting in averages of $cs(X, Y) = 0.103$ and $cs(Y, X) = 0.104$ in *mop3*, and $cs(X, Y) = 0.037$ and $cs(Y, X) = 0.081$ in *mop4*. Such results reinforce the advantages of f_{tm} , showing that SPEA2+ f_{tm} dominated more solutions from SPEA2- f than SPEA2- f did on SPEA2+ f_{tm} .

VI. FINAL REMARKS

This work investigates multiobjective evolutionary models for two formulations of multicast flows routing with QoS and TE requirements. We evaluated the effects of three algorithms for a filter mechanism to reduce repeated individuals on the population. Upon five metrics for convergence and diversity, the results indicated that SPEA2 with filter (specifically, SPEA2+ f_{tm} - the proposed algorithm) reached the best trade-off behavior compared to the other options, resulting on good improvements on convergence metrics with a good compromise on diversity ones. Thus, this paper contributes showing that promoting diversity along

the individuals of the population can be positive, leading to better convergence and, in some cases, diversity values in a multiobjective context.

There is room to improvements on some evaluated instances, specially on those nets where convergence and confidence intervals returned an average behavior. Thus, we intend to investigate mechanisms to improve the evolutionary models investigated, such as memetic algorithms, ϵ -dominance and other heuristics for crossover and mutation operators, as an attempt to improve convergence and diversity in such instances.

ACKNOWLEDGMENTS

MLPB is grateful for CAPES and UFU grants and financial support during this research. GMBO is grateful for CNPq and Fapemig for financial support.

REFERENCES

- [1] E. D. Siegel, *Designing Quality of Service Solutions for the Enterprise*. New York, NY, USA: John Wiley & Sons, Inc., 1999.
- [2] A. S. Tanenbaum, *Computer Networks*, 4th ed. Prentice Hall, 2002.
- [3] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," United States, 2002.

-
- [4] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds., 1972, pp. 85–103.
 - [5] Y. P. Aneja, "An integer programming approach to the steiner problem in graphs," *Networks*, no. 10, pp. 167–178, 1980.
 - [6] J. Beasley, "OR-Library: Distributing test problems by electronic mail," *Journal of the Op. Research Society*, vol. 41, pp. 1069–1072, 1990.
 - [7] S. E. Dreyfus and R. A. Wagner, "The steiner problem in graphs," *Networks*, no. 1, pp. 195–207, 1971.
 - [8] R. Fabregat, Y. Donoso, B. Baran, F. Solano, and J. L. Marzo, "Multi-objective optimization scheme for multicast flows: a survey, a model and a MOEA solution," in *LANC '05: Proc. of the 3rd IFIP Latin American conf. on Networking*. New York, USA: ACM, 2005, pp. 73–86.
 - [9] R. Ravi, "Steiner trees and beyond: approximation algorithms for network design," Ph.D. dissertation, Department of Computer Science – Brown University, Providence, Rhode Island, September 1993.
 - [10] H. Takahashi and Matsuyama, "An approximate solution for the steiner problem in graphs," *Mathematical Japonica*, no. 24, pp. 573–577, 1980.
 - [11] M. P. Aragão, C. C. Ribeiro, E. Uchoa, and R. F. Werneck, "Hybrid local search for the steiner problem in graphs," in *Extended Abstracts of the 4th Metaheuristics International Conference*, 2001, pp. 429–433.
 - [12] C. P. Ravikumar and R. Bajpai, "Source-based delay-bounded multicasting in multimedia networks," *Computer Communications*, vol. 21, no. 2, pp. 126–132, 1998.
 - [13] W. Zhengying, S. Bingxin, and Z. Erdun, "Bandwidth-delay-constrained least-cost multicast routing based on heuristic ga," *Computer Communications*, vol. 24, no. 7-8, pp. 685–692, 2001.
 - [14] X. Cui, C. Lin, and Y. Wei, "A multiobjective model for qos multicast routing based on genetic algorithm," in *ICCNMC '03: Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing*. Washington, USA: IEEE Computer Society, 2003, p. 49.
 - [15] G. M. B. Oliveira and P. T. Araújo, "Determining multicast routes with qos and traffic engineering requirements based on genetic algorithm," in *Proc. of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, vol. 1, Singapore, Dec 2004, pp. 665 – 669.
 - [16] J. Crichigno and B. Barán, "Multiobjective multicast routing algorithm for traffic engineering," in *ICCCN*, R. P. Luijten, L. A. DaSilva, and A. P. J. Engbersen, Eds. IEEE, 2004, pp. 301–306.
 - [17] J. Prieto, B. Barán, and J. Crichigno, "Multitree-multiobjective multicast routing for traffic engineering," in *IFIP AI*, ser. IFIP, M. Bramer, Ed., vol. 217. Springer, 2006, pp. 247–256.
 - [18] G. M. B. Oliveira and S. S. B. V. Vita, "A Multi-Objective Evolutionary Algorithm with ϵ -dominance to Calculate Multicast Routes with QoS Requirements," in *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*, Norway, May 2009, pp. 1–9.
 - [19] M. L. P. Bueno and G. M. B. Oliveira, "New approaches for subtree reconnection in genetic algorithm-based multicast routing," in *Anais do IX Congresso Brasileiro de Redes Neurais e Inteligência Computacional (IX CBRN / IC)*, Ouro Preto, Brasil, 2009, pp. 1–5.
 - [20] —, "Multicast flow routing: Evaluation of heuristics and multiobjective evolutionary algorithms," in *IEEE Congress on Evolutionary Computation (CEC'10)*, Barcelona, Spain, 2010, pp. 1–8.
 - [21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*. Massachusetts: Addison-Wesley, 1989.
 - [22] M. L. P. Bueno and G. M. B. Oliveira, "Multiobjective evolutionary algorithms and a combined heuristic for route reconnection applied to multicast flow routing," in *10th IEEE International Conference on Computer and Information Technology (CIT '10)*, Bradford, UK, 2010.
 - [23] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations," Ph.D. dissertation, Wright-Patterson AFB, OH, 1999. [Online]. Available: cite-seer.ist.psu.edu/vanveldhuizen99multiobjective.html
 - [24] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, pp. 173–195, 2000.