

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

**Metaheurísticas multiobjetivo para o
problema de restauração do serviço em
redes de distribuição de energia elétrica**

Vinícius Jacques Garcia

Tese apresentada à Faculdade de Engenharia
Elétrica e de Computação, UNICAMP, como
requisito parcial para a obtenção do título de
Doutor em Engenharia Elétrica.

Área de Concentração: Automação.

Orientador: Prof.Dr. Paulo Morelato França.

Campinas, SP

2005

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

**Metaheurísticas multiobjetivo para o
problema de restauração do serviço em
redes de distribuição de energia elétrica**

Vinícius Jacques Garcia

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação, UNICAMP, como requisito parcial para a obtenção do título de Doutor em Engenharia Elétrica.

Área de Concentração: Automação.

Aprovação: 11/11/2005.

Banca Examinadora:

- Paulo Morelato França (Orientador)
- Christiano Lyra Filho - DENSIS - FEEC - UNICAMP
- Luiz Carlos Pereira da Silva - DSCE - FEEC - UNICAMP
- Carlos Alberto Castro Júnior - DSEE - FEEC - UNICAMP
- José Roberto Sanches Mantovani - DEE - UNESP Ilha Solteira
- Felipe Martins Müller - DELC - CT - UFSM

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

G165m	Garcia, Vinícius Jacques Metaheurísticas multiobjetivo para o problema de restauração do serviço em redes de distribuição de energia elétrica. / Vinícius Jacques Garcia. – Campinas, SP: [s.n.], 2005. Orientador: Paulo Morelato França. Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. 1. Sistemas de energia elétrica. 2. Heurística. 3. Otimização combinatória. 4. Algoritmos genéticos. I. França, Paulo Morelato. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título
-------	--

Título em Inglês:	Multiobjective metaheuristics for service restoration in electric power distribution networks
Palavras-chave em Inglês:	Metaheuristics, Power systems, Distribution systems, Service restoration, Multiobjective optimization, Genetic algorithms, Tabu search
Área de concentração:	Automação
Titulação:	Doutor em Engenharia Elétrica
Banca Examinadora:	Christiano Lyra Filho, Luiz Carlos Pereira da Silva, Carlos Alberto Castro Júnior, José Roberto Sanches Mantovani e Felipe Martins Müller
Data da defesa:	11/11/2005

Metaheurísticas multiobjetivo para o problema de restauração do serviço em redes de distribuição de energia elétrica

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Vinícius Jacques Garcia e aprovada pela Banca Examinadora.

Campinas, 11 de novembro de 2005.

Paulo Morelato França (Orientador)

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação, UNICAMP, como requisito parcial para a obtenção do título de Doutor em Engenharia Elétrica.

Este trabalho teve o suporte financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo - (FAPESP) e da Coordenação de Aperfeiçoamento Pessoal de Nível Superior - (CAPES)

Resumo

Depois da regulamentação do setor elétrico brasileiro, a qualidade no fornecimento de energia ganhou maior importância por parte das concessionárias. Neste contexto, o problema de restauração do serviço tem particular relevância pela relação com a frequência e duração das interrupções no fornecimento: através de alterações na configuração original da rede, busca-se reduzir a carga não atendida sem deixar de observar as restrições de capacidade dos alimentadores, de queda de tensão nas barras de carga e de radialidade da rede. Considerando o caráter temporário destas manobras, torna-se desejável reduzir o grau de intervenção de modo a facilitar a restauração da configuração original. Nesta tese é considerado o problema multiobjetivo de restauração do serviço que compreende a minimização da carga sem fornecimento e do número de chaves manipuladas. Depois da definição matemática do problema, da revisão da literatura especializada e da descrição de um "framework" para problemas relacionados, são descritas duas heurísticas, uma construtiva e outra de melhoramento. A seguir, apresentam-se duas metaheurísticas para o problema, uma Busca Tabu e um Algoritmo Evolutivo, ambas baseadas em otimização de Pareto. Por fim, por meio de estudos práticos com sistemas de distribuição brasileiros, avalia-se experimentalmente a aplicabilidade das abordagens propostas.

Palavras-chave: Metaheurísticas, Sistemas de potência, Sistemas de distribuição, Restauração do serviço, Otimização multiobjetivo, Algoritmos evolutivos, Busca tabu.

Abstract

After the Brazilian electric power market regulation, quality of service became a crucial concern of utilities. In fact, the service restoration has a particular importance since it is closely related to frequency and duration of service interruption: through network reconfigurations, one aims to reduce the non supplied load while respecting constraints like feeder and voltage limits as well as the maintenance of a radial structure. Considering that this emergency state is transitory existing only until the fault is eliminated, it is convenient to reduce the number of switching operations in order to make the return back to the original configuration easy. This work considers the multiobjective service restoration to minimize both the load not supplied and the number of switching operations. After defining the mathematical formulation proposed and presenting the bibliographical survey with the description of a new framework to related problems, two new heuristics are presented, one for constructive search and another one for neighborhood search. Next, two metaheuristics especially developed for the referred problem are described, both based on Pareto optimization. Finally, the effectiveness of these proposed methods are proved in a set of five systems, three of them referring to actual Brazilian systems.

Keywords: Metaheuristics, Power systems, Distribution systems, Service restoration, Multiobjective optimization, Genetic algorithms, Tabu search.

Dedicatória

A Vanessa com todo o meu amor.

Agradecimentos

Ao meu orientador, Paulo Morelato França, minha eterna gratidão pela excepcional formação proporcionada ao longo dos quase 4 anos de trabalho. Sua amizade, paciência, perspicácia e experiência muito contribuíram para que esta tese chegasse ao final. Obrigado Amigo!

Ao professor Mesut Baran, da “North Carolina State University”, pela acolhida gentil e amável e também pela experiência profissional durante os três meses que passei nos EUA. Aos meus colegas de laboratório Jinsang e Yi e ao eficiente e atencioso Jerry, meu sincero agradecimento pela atenção com que me receberam em Raleigh.

Ao professor André Luiz Morelato França, pelas valiosas discussões sobre o problema de restauração do serviço.

Ao Celso Cavellucci e ao Prof. Christiano Lyra Filho, pela ajuda fundamental em temas técnicos e pela inestimável amizade.

Aos meus colegas e amigos José Federico Vizcaino González e Edilson Aparecido Bueno, pela fiel e indispensável colaboração.

Aos demais colegas e amigos do laboratório de Redes de Distribuição: Leonardo, Paulo, Joselaine, Marta e Daniela, obrigado pelo carinho e consideração de vocês.

Aos companheiros e amigos do DENSIS: Luiz, Marcelo, Olinto, André, Anibal, Eduardo, Cláudio, Jeanne, meu sincero reconhecimento pela convivência e pelo ambiente fraterno que cultivam.

Ao Prof. Felipe Martins Müller, pela confiança e incentivo.

Aos padrinhos Jorge e Clélia, a Lulu e ao Victor.

A todos os amigos, em particular aos residentes em Campinas: Glaudson, Paula, Fabiano, Cássia, Mário, Tatiane, Jeferson, Emilene, Vinícius, Fábio, Adriane, Leonardo, Daniela, Hudo, Camila, Amanda, e a todos os demais que, de alguma forma, me proporcionaram momentos agradáveis e descontraídos.

Ao CTT[®] (Conselho Tutorial do Trago) pelo companheirismo, pela cachaça e pela união.

Aos meus pais, incansáveis e fiéis incentivadores, aos meus irmãos Rômulo e Laura,

ao meu cunhado Rafael, aos meus queridos sobrinhos Henrique e Victor, pelos momentos inesquecíveis que passamos juntos. Obrigado pelo amor de vocês.

A Neuza, Acioli (*in memoriam*), Michele e Vinícius, pelo carinho e pela amizade incondicional de vocês.

A Vanessa, minha querida esposa, sempre presente e cuidadosa, meu carinho, meu amor e meu profundo reconhecimento pela dedicação e incentivo.

A Deus pela inspiração e alegria de viver.

Sumário

Resumo	vii
Abstract	ix
Dedicatória	xi
Agradecimentos	xiii
Lista de Tabelas	xix
Lista de Figuras	xxi
Lista de Abreviaturas	xxv
1 Introdução	1
1.1 Sistemas de energia elétrica	3
1.1.1 Sistema de distribuição	4
1.1.2 Estados do sistema de distribuição	6
1.1.3 Escopo de aplicação do problema de restauração do serviço	7
1.2 Revisão bibliográfica	8
1.2.1 Abordagens baseadas em Inteligência Artificial	9
1.2.2 Abordagens baseadas em programação matemática	10
1.2.3 Abordagens heurísticas	10
1.2.4 Abordagens evolutivas	11
1.2.5 Abordagens multicritério	12
2 Otimização Combinatória	15
2.1 Metaheurísticas	16
2.1.1 Conceitos básicos	17

2.1.2	<i>Simulated Annealing</i>	21
2.1.3	Busca Tabu	24
2.1.4	Algoritmos evolutivos	25
3	Otimização combinatória multiobjetivo	29
3.1	Problemas de otimização multiobjetivo	29
3.1.1	Definições	31
3.2	Classificação de métodos para avaliação multiobjetivo	34
3.3	Métodos clássicos de otimização multiobjetivo	35
3.3.1	Método da soma ponderada	36
3.3.2	Método do ε -restrito	37
3.4	Metaheurísticas para POCM	38
3.4.1	<i>Simulated Annealing</i>	39
3.4.2	Busca Tabu	40
3.4.3	Algoritmos evolutivos	41
3.5	Métodos de avaliação de heurísticas multiobjetivo	50
3.5.1	Métodos que consideram o conjunto Pareto-ótimo	51
3.5.2	Medidas baseadas na preferência do decisor	52
3.5.3	Escolha do método para avaliação das heurísticas multiobjetivo	56
4	Definição do problema de restauração do serviço	59
4.1	Metodologia para resolução	64
4.1.1	Simplificação adotada	66
4.1.2	Fluxo de carga utilizado	67
5	Modelagem computacional	71
5.1	Diagrama de classes	72
5.1.1	Módulo Entrada e Saída	73
5.1.2	Módulo Estruturas de Dados Auxiliares	74
5.1.3	Módulo Instância	75
5.1.4	Módulo Solução	76
5.1.5	Módulo Algoritmo	77
5.2	Estruturas de dados	77
5.2.1	Estrutura de dados para a Instância	78
5.2.2	Estrutura de dados para a Solução	81
5.2.3	Estrutura de dados para o Conjunto de Pareto	84

6	Algoritmos heurísticos multiobjetivo para o PRS	87
6.1	Algoritmo construtivo	88
6.1.1	Algoritmo construtivo segundo o critério de máxima carga restaurada	90
6.1.2	Algoritmo construtivo segundo o critério de mínimo chaveamento .	94
6.2	Busca local	98
6.2.1	Redução por agrupamento (<i>clustering</i>)	100
6.2.2	Vizinhança desenvolvida	103
7	Algoritmo evolutivo multiobjetivo para o PRS	107
7.1	Representação dos indivíduos	109
7.2	Função de adaptação	114
7.3	Mecanismo de seleção	116
7.4	Operador de recombinação	117
7.5	Operador de mutação	118
7.6	Redução da população	121
8	Busca Tabu multiobjetivo para o PRS	123
8.1	Algoritmo proposto	124
8.1.1	Geração da vizinhança e memória adaptativa	127
8.1.2	Seleção de soluções dominadas	131
8.1.3	Mecanismo de diversificação	132
9	Resultados computacionais	135
9.1	Resultados para o algoritmo evolutivo	141
9.2	Resultados para a busca tabu	143
9.3	Comparação dos métodos propostos	149
10	Conclusões	159
	Referências Bibliográficas	163
A	Conceitos básicos da teoria dos grafos	173
B	Resultados para o algoritmo evolutivo proposto	175
C	Resultados para a busca tabu proposta	187

Lista de Tabelas

4.1	Nomenclatura.	61
5.1	Listas de vértices e arestas do grafo da Figura 5.2.	79
6.1	Caminhos de interligação entre todos os nós da área escura e todos os nós-fonte, obtidos dos grafos das Figuras 6.11(c)-6.11(g).	105
9.1	Redes de distribuição empregadas nos testes computacionais.	136
9.2	Parâmetros utilizados para o AE e os respectivos valores associados.	142
9.3	Resultados da aplicação do AE com uma única configuração de parâmetros para todas as redes da Tabela 9.1.	142
9.4	Parâmetros utilizados para a BT com os respectivos valores associados.	148
9.5	Resultados da aplicação do BT com uma única configuração de parâmetros para todas as redes da Tabela 9.1.	148
9.6	Comparação de desempenho entre o AE e a BT com as redes da Tabela 9.1.	155
B.1	Primeira parte dos resultados da aplicação do AE para a rede 1 da Tabela 9.1.	176
B.2	Segunda parte dos resultados da aplicação do AE para a rede 1 da Tabela 9.1.	177
B.3	Primeira parte dos resultados da aplicação do AE para a rede 2 da Tabela 9.1.	178
B.4	Segunda parte dos resultados da aplicação do AE para a rede 2 da Tabela 9.1.	179
B.5	Primeira parte dos resultados da aplicação do AE para a rede 3 da Tabela 9.1.	180
B.6	Segunda parte dos resultados da aplicação do AE para a rede 3 da Tabela 9.1.	181
B.7	Primeira parte dos resultados da aplicação do AE para a rede 4 da Tabela 9.1.	182
B.8	Segunda parte dos resultados da aplicação do AE para a rede 4 da Tabela 9.1.	183
B.9	Primeira parte dos resultados da aplicação do AE para a rede 5 da Tabela 9.1.	184
B.10	Segunda parte dos resultados da aplicação do AE para a rede 5 da Tabela 9.1.	185
C.1	Primeira parte dos resultados da aplicação da BT para a rede 1 da Tabela 9.1.	188
C.2	Segunda parte dos resultados da aplicação da BT para a rede 1 da Tabela 9.1.	189
C.3	Primeira parte dos resultados da aplicação da BT para a rede 2 da Tabela 9.1.	190
C.4	Segunda parte dos resultados da aplicação da BT para a rede 2 da Tabela 9.1.	191
C.5	Primeira parte dos resultados da aplicação da BT para a rede 3 da Tabela 9.1.	192
C.6	Segunda parte dos resultados da aplicação da BT para a rede 3 da Tabela 9.1.	193
C.7	Primeira parte dos resultados da aplicação da BT para a rede 4 da Tabela 9.1.	194

- C.8 Segunda parte dos resultados da aplicação da BT para a rede 4 da Tabela 9.1.195
- C.9 Primeira parte dos resultados da aplicação da BT para a rede 5 da Tabela 9.1.196
- C.10 Segunda parte dos resultados da aplicação da BT para a rede 5 da Tabela 9.1.197

Lista de Figuras

1.1	Representação de um sistema de energia elétrica.	4
1.2	Organização do sistema de distribuição de energia elétrica.	5
1.3	Estados de operação da rede de distribuição.	7
2.1	O espaço de busca S , uma possível solução x e sua vizinhança $N(x)$ representada pelo interior do círculo que contém x	19
2.2	Pseudo-código de um algoritmo de <i>Simulated Annealing</i>	23
2.3	Pseudo-código de um algoritmo de <i>Busca Tabu</i>	26
2.4	Algoritmo Evolutivo.	27
3.1	Interpretação gráfica do método da soma ponderada (Fonte: Arroyo (2002)).	37
3.2	Interpretação gráfica do método ε -restrito (Fonte: Arroyo (2002)).	38
3.3	A aproximação A é visivelmente melhor que B, embora a medida Q_1 indique que ambas são equivalentes.	52
3.4	Exemplos das relações descritas: $B O_W C$ e $A O_C C$	53
3.5	A medida Q_3 favorece a aproximação A em relação à B.	54
3.6	Aproximações incomparáveis com a utilização da Definição 3	55
3.7	Aproximações comparadas pela medida Q_6	57
4.1	Grafo resultante após isolamento do defeito para uma rede hipotética.	65
4.2	Subgrafo obtido a partir da simplificação do grafo da Figura 4.1.	66
4.3	Redução da rede através da criação de <i>blocos de carga</i>	67
4.4	Diagrama simplificado de uma rede de distribuição radial.	68
4.5	Fluxo de carga <i>Varredura VaiEVem</i>	69
5.1	Diagrama de classes do sistema computacional.	72
5.2	Modelo de grafo para uma rede de distribuição.	79
5.3	Representação para o grafo da Figura 5.2.	80
5.4	Subgrafo e estrutura de dados para o setor 1.	81
5.5	Subgrafo e estrutura de dados para o setor 2.	82
5.6	Subgrafo e estrutura de dados para o setor 3.	82
5.7	Duas possíveis árvores para o setor 2, escolhendo-se os vértices 7 e 9 como raiz.	83
5.8	Grafo e estrutura de dados para uma solução.	84

5.9	Lista linear empregada como estrutura de dados para manter o conjunto de Pareto.	85
6.1	Redes antes e depois do isolamento do defeito e a rede em operação ⁰	89
6.2	Soluções obtidas ⁰ com o algoritmo construtivo segundo o critério de máxima carga restaurada, a partir do nó-fonte 4.	91
6.3	Soluções obtidas ⁰ com o algoritmo construtivo, segundo o critério de máxima carga restaurada, a partir do nó-fonte 6.	93
6.4	Pseudo-código do algoritmo construtivo, segundo o critério de máxima carga restaurada.	94
6.5	Soluções obtidas ⁰ com o algoritmo construtivo, segundo o critério de mínimo chaveamento, a partir do nó-fonte 4.	96
6.6	Soluções obtidas ⁰ com o algoritmo construtivo, segundo o critério de mínimo chaveamento, a partir do nó-fonte 6.	97
6.7	Pseudo-código do algoritmo construtivo, segundo o critério de mínimo chaveamento.	98
6.8	Pseudo-código da busca local multiobjetivo.	99
6.9	Pseudo-código do método de redução por agrupamento.	101
6.10	Resultado da redução de um conjunto pelo método de Morse (1980).	102
6.11	Vizinhança para o PRS ⁰	104
6.12	Soluções vizinhas da solução x com a utilização dos caminhos oriundos do nó 7 até os nós-fonte alcançáveis ⁰ , definidos na Tabela 6.1.	106
7.1	Algoritmo evolutivo desenvolvido para o PRS.	109
7.2	Rede pós-falha ⁰ com os caminhos de interligação entre os nós da área escura e os respectivos nós-fonte alcançáveis.	111
7.3	Representação de um indivíduo no AE desenvolvido para o PRS e as etapas de decodificação ⁰	112
7.4	Representação da permanência do nó 9 na área escura ⁰	113
7.5	Estrutura de dados para a solução da Figura 7.3(f), obtida depois da decodificação do indivíduo representado pelo cromossomo da Figura 7.3(a).	113
7.6	Comparação entre soluções quando a seqüência de caminhos nos respectivos cromossomos é alterada ⁰	115
7.7	Exemplo de aplicação do operador de recombinação ⁰	119
7.8	Exemplo de aplicação do operador de mutação ⁰	120
8.1	Pseudo-código da Busca Tabu multiobjetivo desenvolvida.	128
8.2	Memória adaptativa utilizada na BT desenvolvida: uma lista tabu (LT) e uma lista de frequência (LF) para cada uma das N_{sol} trajetórias de busca.	130
8.3	Pseudo-código da procedimento “GeraVizinhança” do passo 14 da Figura 8.1.	131
8.4	Pseudo-código da fase de diversificação empregada na BT da Figura 8.1.	133
9.1	Topologia da rede de distribuição 1.	137

9.2	Topologia da rede de distribuição 2.	137
9.3	Topologia da rede de distribuição 3.	138
9.4	Topologia da rede de distribuição 4.	139
9.5	Topologia da rede de distribuição 5.	140
9.6	Fronteira inicial e final obtidas pela execução do AE para a rede 1 com os parâmetros da Tabela 9.3.	143
9.7	Fronteira inicial e final obtidas pela execução do AE para a rede 2 com os parâmetros da Tabela 9.3.	144
9.8	Fronteira inicial e final obtidas pela execução do AE para a rede 3 com os parâmetros da Tabela 9.3.	145
9.9	Fronteira inicial e final obtidas pela execução do AE para a rede 4 com os parâmetros da Tabela 9.3.	146
9.10	Fronteira inicial e final obtidas pela execução do AE para a rede 5 com os parâmetros da Tabela 9.3.	147
9.11	Fronteira inicial e final obtidas pela execução da BT para a rede 1 com os parâmetros da Tabela 9.5.	149
9.12	Fronteira inicial e final obtidas pela execução da BT para a rede 2 com os parâmetros da Tabela 9.5.	150
9.13	Fronteira inicial e final obtidas pela execução da BT para a rede 3 com os parâmetros da Tabela 9.5.	151
9.14	Fronteira inicial e final obtidas pela execução da BT para a rede 4 com os parâmetros da Tabela 9.5.	152
9.15	Fronteira inicial e final obtidas pela execução da BT para a rede 5 com os parâmetros da Tabela 9.5.	153
9.16	Fronteiras obtidas pelo AE e pela BT para a rede 3 da Tabela 9.5.	155
9.17	Fronteiras obtidas pelo AE e pela BT para a rede 4 da Tabela 9.5.	156
9.18	Fronteiras obtidas pelo AE e pela BT para a rede 5 da Tabela 9.5.	157
A.1	Exemplo de grafo não-orientado e orientado.	174

Lista de Abreviaturas

DNAEE	Departamento Nacional de Águas e Energia Elétrica	1
ANEEL	Agência Nacional de Energia Elétrica	1
DEC	Duração Equivalente de interrupção por unidade Consumidora	1
FEC	Frequência Equivalente de interrupção por unidade Consumidora	1
DIC	Duração de Interrupção por unidade Consumidora	1
DMIC	Duração Máxima de Interrupção por unidade Consumidora	1
FIC	Frequência de Interrupção por unidade Consumidora	1
PRS	Problema de Restauração do Serviço	2
NSGA	<i>Non-dominated Sorting Genetic Algorithm</i>	14
BT	Busca Tabu	24
AE	Algoritmo Evolutivo	25
POCM	Problema de Otimização Combinatória Multiobjetivo	29
TDM	Tomada de decisão multicritério	30
SA	<i>Simulated Annealing</i>	39
MOSA	Multiple Objective Simulated Annealing	39
AEM	Algoritmo Evolutivo Multiobjetivo	41
HLGA	<i>Hajela's Lin's Genetic Algorithm</i>	42
VEGA	<i>Vector Evaluated Genetic Algorithm</i>	43
MOGA	<i>Multiple Objective Genetic Algorithm</i>	45
NSGA	<i>Non-Dominated Sorting Genetic Algorithm</i>	46
NPGA	<i>Niched Pareto Genetic Algorithm</i>	46
PAES	<i>Pareto Archived Evolution Strategy</i>	47

SPEA	<i>Strength Pareto Evolutionary Algorithm</i>	47
PESA	<i>Pareto Envelope-based Selection Algorithm</i>	48

Capítulo 1

Introdução

Diante de situações de contingência provocadas por intempéries eletromagnéticas e/ou climáticas que afetam a rede de distribuição de energia elétrica, é bastante provável que haja certas regiões sem fornecimento devido à atuação de equipamentos de proteção. Ainda que o defeito seja bem restrito e já esteja devidamente isolado, áreas vizinhas também são afetadas, prejudicando os índices que medem a qualidade do serviço de distribuição, além de provocar prejuízo financeiro direto às companhias de distribuição pela ausência de consumo.

A preocupação com os aspectos de continuidade no sistema de distribuição não é atual, data da publicação da portaria 46 de 1978, do já extinto Departamento Nacional de Águas e Energia Elétrica (DNAEE). Depois da regulamentação do sistema elétrico brasileiro, a Agência Nacional de Energia Elétrica (ANEEL) publicou a resolução 24 em 28 de janeiro de 2000, revogando a portaria 46/1978 do DNAEE e estabelecendo novos índices para medir a qualidade no fornecimento de energia. Houve, de fato, a manutenção dos índices para medir a duração (DEC¹) e a frequência (FEC²) de interrupção por conjunto de unidades consumidoras, com a inclusão de outros índices para medir a duração (DIC³ e DMIC⁴) e a frequência (FIC⁵) de interrupção individualmente para cada unidade consumidora. Exatamente essa duração de interrupção pode ser significativamente reduzida quando da utilização de procedimentos de restauração do serviço, que exploram as características da rede de distribuição na tentativa de restabelecer o fornecimento para as regiões afetadas pela contingência.

¹Duração Equivalente de interrupção por unidade Consumidora.

²Frequência Equivalente de interrupção por unidade Consumidora.

³Duração de Interrupção por unidade Consumidora.

⁴Duração Máxima de Interrupção por unidade Consumidora.

⁵Frequência de Interrupção por unidade Consumidora.

O principal objetivo da restauração do serviço é minimizar o número de consumidores sem fornecimento de energia, transferindo-os para alimentadores de suporte do sistema de distribuição. Devem ser preservadas as condições mínimas para a operação do sistema, como radialidade da rede, limites dos equipamentos e quedas de tensão dentro dos padrões definidos.

No cotidiano das companhias de distribuição, técnicos utilizam um conjunto de regras empíricas e procedimentos de emergência para tentar restabelecer o fornecimento de energia aos consumidores afetados. As limitações inerentes ao processo ainda têm um agravante: o tempo. Como são condições anormais, é preciso que a resposta seja rápida e precisa. De fato, como a restauração do serviço é um problema de otimização combinatoria não-linear e muito restrito, a sua complexidade levanta questionamentos sobre a eficácia de alternativas baseadas em regras e procedimentos pré-estabelecidos.

Neste trabalho são propostas duas metaheurísticas para resolução do Problema de Restauração do Serviço (PRS), considerando uma abordagem multiobjetivo. Ainda que o procedimento de restauração da rede tenha como principal objetivo restaurar a máxima carga possível, as alterações propostas através da troca do estado de chaves devem ser mínimas, justamente porque se trata de uma solução temporária. Com isso, vislumbram-se os dois objetivos considerados: minimizar a carga sem fornecimento e minimizar o número de chaveamentos requeridos.

A seguir é descrita a organização desta tese.

- Na Seção 1.1 são introduzidos alguns conceitos dos sistemas de energia elétrica que permitem definir e enquadrar o PRS, assim como o escopo de aplicação do problema. Ainda neste capítulo é apresentado o levantamento bibliográfico realizado para o PRS, Seção 1.2;
- O Capítulo 2 define otimização combinatoria, apresenta métodos heurísticos e também algumas metaheurísticas utilizadas na resolução de problemas mono-objetivo;
- O Capítulo 3 apresenta os conceitos e técnicas fundamentais de otimização combinatoria multiobjetivo, desenvolvendo ainda alguns métodos para comparação de heurísticas multiobjetivo;
- A definição do problema, com o respectivo modelo matemático proposto, é apresentada no Capítulo 4;
- O Capítulo 5 contém a modelagem computacional desenvolvida, o diagrama de classes e uma breve descrição de cada uma delas;

- No Capítulo 6, antes da descrição das metaheurísticas propostas para o PRS, são apresentados os algoritmos heurísticos multiobjetivo constituintes destas técnicas;
- O Algoritmo Evolutivo proposto para o PRS é apresentado no Capítulo 7;
- No Capítulo 8 é apresentada a Busca Tabu multiobjetivo desenvolvida para o PRS;
- Os resultados dos testes da aplicação realizados com sistemas de distribuição brasileiros são apresentados no Capítulo 9;
- Por fim são apresentadas as conclusões e as contribuições desta tese no Capítulo 10.

1.1 Sistemas de energia elétrica

Mesmo depois da última reestruturação do setor elétrico brasileiro (lei nº 10.848/2004) (Fortunato, Neto, Albuquerque e Pereira 1990), é possível identificar três setores fundamentais nos sistemas de energia elétrica:

- **Geração:** corresponde à produção de energia elétrica propriamente dita. No Brasil, vários tipos de usinas geradoras são encontradas: usinas hidrelétricas, térmicas, nucleares, eólicas, biogás, biomassa, solares, etc;
- **Transmissão/subtransmissão:** compreende as linhas de transmissão, desde as usinas até as subestações. Estas últimas também fazem parte do sistema de transmissão de energia elétrica. A subtransmissão pode ser considerada uma parte da transmissão destinada a transferência de energia em trechos com menor distância e com níveis de tensão menores que a transmissão propriamente dita;
- **Distribuição:** esse sistema refere-se à parte que interliga as subestações aos consumidores finais, sejam residenciais, comerciais ou industriais. Ainda é preciso citar produtores independentes e a geração distribuída como atores que também se fizeram presentes na distribuição, especialmente depois da reestruturação do setor elétrico.

Como não há meios de armazenagem de energia elétrica para grandes quantidades, toda energia produzida pela geração deve ser imediatamente transportada até os centros urbanos, de forma a ser distribuída e consumida. Considerando o cenário da geração distribuída, o transporte de energia é muito favorecido justamente porque os geradores estão próximos dos locais de consumo. Um benefício notável desta localização estratégica

é a redução da energia dissipada nos condutores da rede, evitando, com isso, as chamadas perdas elétricas.

O diagrama da Figura 1.1 apresenta os principais componentes da geração, transmissão, subtransmissão e distribuição de energia elétrica (Pansini 1983). Logo na usina, onde a energia é gerada, a tensão elétrica é elevada para o nível da transmissão, da ordem de centenas de kV. As subestações nos centros urbanos se encarregam de rebaixar essa tensão para os padrões do sistema de distribuição, da ordem de dezenas de kV.

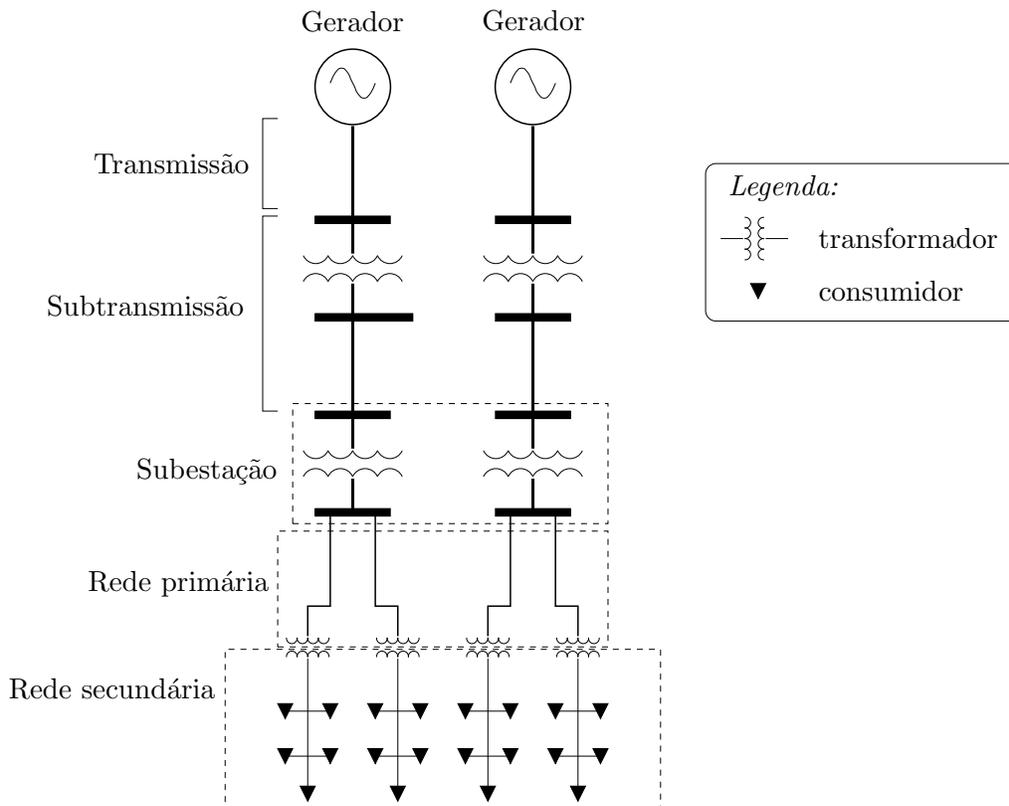


Figura 1.1: Representação de um sistema de energia elétrica.

1.1.1 Sistema de distribuição

O sistema de distribuição, por sua vez, também subdivide-se em dois: rede primária e rede secundária, conforme o diagrama da Figura 1.2. Faz parte da rede primária o barramento da subestação de distribuição e a rede até os transformadores de distribuição. A partir destes encontra-se a rede secundária. Os penúltimos equipamentos do sistema

são os medidores, que determinam o consumo de energia a ser pago. Depois deles existem somente as cargas propriamente ditas.

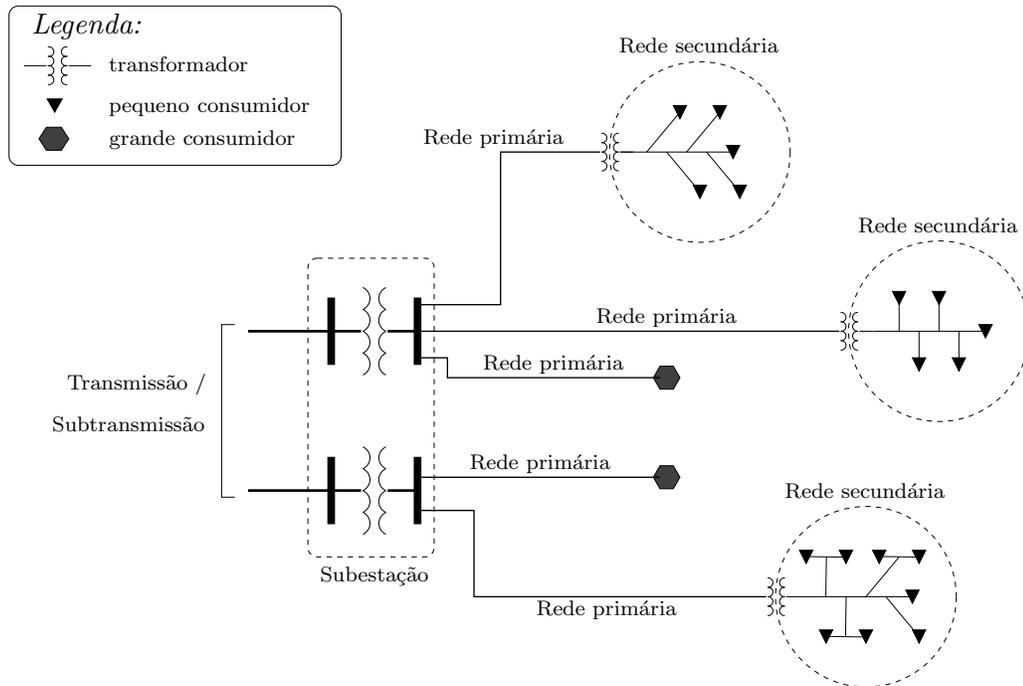


Figura 1.2: Organização do sistema de distribuição de energia elétrica.

As redes primárias compreendem todos os equipamentos entre os transformadores das subestações de distribuição até os transformadores de distribuição, geralmente localizados nos postes das ruas e que fazem a redução do nível de tensão para valores encontrados nos consumidores. A energia é conduzida por meio de alimentadores, que incluem condutores partindo das subestações até os transformadores de distribuição, e de equipamentos adequadamente arranjados, tais como equipamentos de manobra (chaves), de proteção (disjuntores e religadores), de correção (reguladores de tensão e bancos de capacitores) e de transformação.

Os equipamentos de manobra permitem a alteração da configuração dos circuitos dos alimentadores para gerenciamento da rede. Os equipamentos de proteção atuam quando da ocorrência de um defeito, de forma coordenada e com o intuito de proteger os outros equipamentos da rede, bem como o consumidor. Equipamentos de correção servem para manter certos indicadores de desempenho da rede dentro dos níveis regulamentados. Por fim, os equipamentos de transformação encarregam-se de reduzir a tensão de distribuição para níveis adequados aos consumidores.

As redes secundárias têm início nos transformadores de distribuição e terminam nos pontos de entrega aos consumidores. Incluem condutores e medidores e apresentam níveis baixos de tensão, tipicamente 220V.

De especial importância nos sistemas de distribuição, a qualidade no fornecimento de energia elétrica é garantida pelos padrões estabelecidos pela ANEEL. Entretanto, ainda que as companhias de distribuição observem as normas da agência, certos equipamentos eletro-eletrônicos interligados à rede pelo consumidor podem ocasionar perturbações indesejadas, tais como distorções harmônicas, e contribuir para a dissipação dessa “poluição elétrica”.

A observância das normas da ANEEL, que incluem protocolos de medição, metodologias de cálculo relacionadas aos equipamentos de medição, além de indicadores de qualidade, permitem a avaliação do desempenho da rede e eventuais levantamentos relativos às responsabilidades decorrentes do mau uso ou mau funcionamento. Em geral, os indicadores de qualidade definidos pela ANEEL relacionam frequência e duração da interrupção no fornecimento (FEC e DEC), tensões e limites de fornecimento, fator de potência das instalações, cintilação (*flicker*) e distorções harmônicas.

Com isso, torna-se imperativo o investimento por parte das concessionárias em equipamentos e procedimentos de modo a atender as exigências da ANEEL e evitar eventuais prejuízos decorrentes da redução no faturamento, ocasionado pela ausência de consumo, e de eventuais penalizações financeiras.

A seção a seguir trata dos estados de operação do sistema de distribuição de energia elétrica.

1.1.2 Estados do sistema de distribuição

O sistema de distribuição tem por objetivo fornecer energia elétrica de forma confiável e econômica, respeitando restrições operacionais e de carga. As restrições operacionais impõem limites de tensão e corrente elétrica para que os equipamentos funcionem de forma segura e eficiente, enquanto que as restrições de carga determinam que a demanda total dos consumidores deve ser atendida. Diante disso, a operação do sistema de distribuição pode ser dividida em três estados distintos (Murphy e Wu 1990): normal, de emergência e de restauração. A Figura 1.3 apresenta o relacionamento entre esses estados.

No estado normal ambas as restrições, operacionais e de carga, são satisfeitas, sendo as únicas atribuições da operação: a manutenção da confiabilidade e a redução dos custos operacionais. O estado de emergência caracteriza-se pela ocorrência de alguma falha na rede que provoque a violação das restrições operacionais. Depois de identificado e

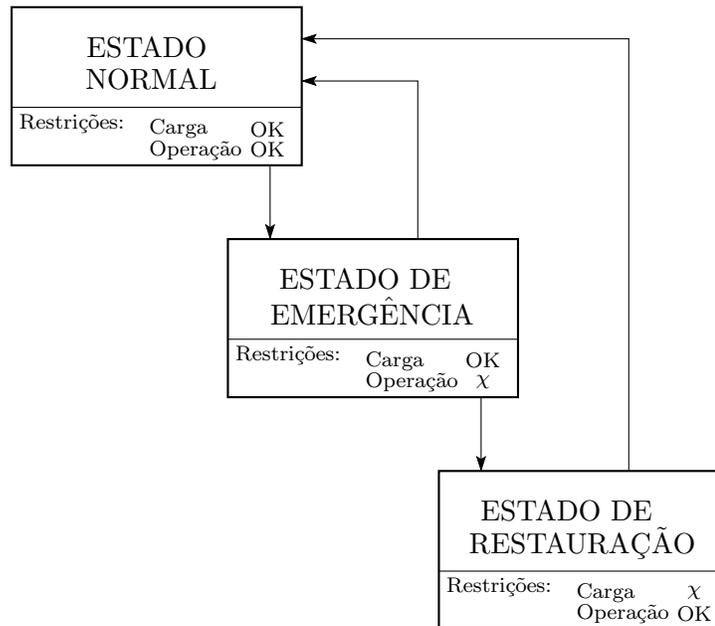


Figura 1.3: Estados de operação da rede de distribuição.

isolado o defeito, a rede passa para o estado de restauração, com as restrições operacionais satisfeitas.

1.1.3 Escopo de aplicação do problema de restauração do serviço

O PRS abordado neste trabalho encarregar-se-á apenas de uma parte do sistema de energia elétrica. Conforme dito anteriormente, ele é um problema relativo ao sistema de distribuição.

A parte da rede elétrica abordada neste estudo compreende desde as subestações rebaixadoras, as quais são a fronteira entre o sistema de transmissão e o sistema de distribuição, até os transformadores rebaixadores, encarregados de fornecer a energia com o nível de tensão adequado ao tipo de consumidor (industrial, comercial ou residencial). De outra forma, a parte do sistema de distribuição que será objeto de estudo corresponde apenas à rede primária.

O restabelecimento parcial ou total do fornecimento de energia elétrica, ocasionado por uma eventual contingência, é realizado por meio dos equipamentos de manobra existentes na rede primária. Como a grande maioria dos sistemas de distribuição operam de forma radial, o que significa que há somente um único caminho energizado entre um transforma-

dor de subestação e cada um dos consumidores, as manobras realizadas deverão manter este estado, na medida em que os consumidores têm o seu fornecimento restabelecido.

Com a evolução dos níveis de automação das concessionárias, algumas dessas manobras podem ser realizadas remotamente. Há também aquelas manuais, efetuadas pelas equipes de manutenção. Considerando todos os equipamentos de manobra (chaves) como variáveis de decisão de um problema de otimização para redução da área sem fornecimento, o número de configurações possíveis está relacionado com o número de combinações de estado destas chaves. Devido ao grande número de chaves existentes, caracteriza-se um problema de otimização combinatória altamente complexo, envolvendo funções objetivo e restrições não-lineares. Ainda que os avanços tecnológicos tenham permitido o aumento considerável na capacidade de processamento dos computadores, a enumeração completa de todos os estados possíveis é inviável, até mesmo para redes com poucas dezenas de barras de carga.

No Brasil, grande parte das redes de distribuição apresentam poucas chaves manipuláveis pela operação. Como conseqüência, a confiabilidade da rede também é prejudicada, dada que esta tem relação direta com o número de chaves justamente pelas alternativas de alimentação para uma mesma carga (alimentadores laterais). Entretanto, mesmo com esta pequena quantidade de chaves manipuláveis, o problema decorrente dessas redes ainda é difícil em decorrência da natureza combinatória.

1.2 Revisão bibliográfica

Especialmente desde o final dos anos 80, cresceu o número de trabalhos desenvolvidos propondo técnicas e abordagens para resolução do PRS (Curcic, Ozveren, Crowe e Lo 1996). As primeiras propostas procuravam aproveitar ao máximo o conhecimento dos operadores, utilizando heurísticas que guardam bastante semelhança com as regras e procedimentos utilizados por eles, fossem heurísticas (Aoki, Kuwabara, Satoh e Kanezashi 1987; Aoki, Satoh, Itoh, Kuwabara e Kanezashi 1988; Aoki, Nara, Itoh, Satoh e Kuwabara 1989; Morelato e Monticelli 1989) ou baseadas em Inteligência Artificial (sistemas especialistas) (Liu, Lee e Venkata 1988; Cavellucci 1989).

Na década seguinte, 1990, técnicas heurísticas para o PRS continuaram tendo relativa importância (Shirmohammadi 1992; Nahman e Strbac 1994; Miu, Chiand, Yuan e Darling 1998). Ainda nesta década, houve trabalhos que já empregavam metaheurísticas com relativo sucesso (Mori e Takeda 1994; Fukuyama, Chiang e Miu 1996; Toune, Fudo, Genji, Fukuyama e Nakanishi 1998; Matos e Melo 1999). Houve também abordagens que empregaram lógica nebulosa (Hsu e Kuo 1994; Augugliaro, Dusonchet e Sanseverino 2000).

Mais recentemente, avançando na consideração de aspectos práticos envolvendo uma mais adequada abordagem para o PRS, têm ganho importância abordagens que consideram o aspecto multicritério do problema, utilizando lógica nebulosa (Huang 2003) ou a combinação desta com algoritmos genéticos (Hsiao e Chien 2000; Augugliaro, Dusonchet e Sanseverino 2001). A utilização de heurísticas (Miu, Chiang e McNulty 2000; Toune, Fudo, Genji e Fukuyama 2002), ainda que combinada com métodos de programação matemática inteira-mista (Ciric e Popovic 2000), comprova que métodos aproximados tornam-se ainda mais necessários quando são considerados vários objetivos.

A seguir detalham-se algumas abordagens, classificando-as quanto às técnicas utilizadas, às restrições consideradas e ao número de objetivos considerados.

1.2.1 Abordagens baseadas em Inteligência Artificial

O trabalho precursor de Morelato e Monticelli (1989) propõe a representação de uma solução para o PRS através de um conjunto de estados de chaves. O estado de cada uma pode ser aberto (0) ou fechado (1). O conhecimento específico do problema guia uma busca heurística, que percorre uma árvore de decisão binária. Cada nível da árvore compreende a definição de uma variável binária, sendo os estados consequência do caminho escolhido para alcançar as folhas. Estas caracterizam-se pela decisão de todas as variáveis, com seus respectivos estados.

A busca heurística utilizada compreende uma busca em profundidade na árvore de decisão até encontrar uma folha. Depois é testada a factibilidade da solução encontrada. Caso a mesma seja infactível, um procedimento de *backtracking* é iniciado de modo a obter uma solução alternativa. Os resultados reportados contemplam apenas uma rede com 20 cargas, 33 chaves e 4 alimentadores.

Cavellucci (1989) propõe em sistema integrado para visualização e determinação de planos de manobra quando da ocorrência de uma contingência. Esses planos são determinados a partir de um sistema especialista, o qual é baseado nos procedimentos empregados nos centros de controle dos sistemas de distribuição. Para cada plano de manobra sugerido pelo sistema especialista, todos os requisitos elétricos são verificados.

No trabalho de Kim, Ko e Jung (1992), uma estratégia de busca do tipo *best-first* é desenvolvida. Os autores afirmam que não há garantia de obtenção da solução ótima. O procedimento adotado tem por objetivo transferir cargas sistematicamente, desde as áreas afetadas pelas cargas até os alimentadores em perfeita operação. Regras heurísticas, obtidas dos operadores de companhias e de engenheiros de órgãos controladores do sistema interligado, são incorporadas ao algoritmo. Além dessas regras, outras dependentes da

estrutura topológica da rede são geradas para reduzir o número de possíveis chaveamentos. Os resultados obtidos, embora restritos a pequenos problemas, demonstram que o algoritmo desenvolvido encontra soluções muito próximas da ótima.

1.2.2 Abordagens baseadas em programação matemática

Abordagens estritamente baseadas em métodos matemáticos exatos de otimização não são encontradas, salvo algumas exceções, justamente por se tratar de um problema em geral de grande porte e bastante restrito (Curcic et al. 1996). Propostas de solução que contemplam modelos matemáticos são, em geral, compostas por heurísticas inspiradas em métodos clássicos de otimização (Aoki et al. 1989).

O trabalho de Aoki et al. (1989) propõe um modelo de programação inteira, inspirado no problema da mochila. Isso é possível através da seguinte metodologia: resolve-se o problema da transferência de carga entre os transformadores, considerando-se particularmente dois a dois. O método aproximado faz uso do gradiente efetivo, considerando restrições de limite de fornecimento dos alimentadores e de fluxo nas linhas, quedas de tensão nos consumidores, assim como radialidade. A função objetivo maximiza unicamente a área restaurada. Os resultados computacionais contemplam uma rede média, com 86 nós consumidores e 577 chaves seccionadoras.

Recentemente, Ciric e Popovic (2000) propõem um método híbrido, combinando técnicas heurísticas e programação inteira-mista. O espaço de soluções é reduzido por meio da montagem de uma lista contendo todas as possíveis soluções que sejam alcançáveis através do fechamento de apenas uma chave. Um procedimento de sub-otimização, que usa programação inteira-mista, é utilizado quando não há solução na lista anterior. As restrições consideradas são: capacidade dos alimentadores, limites de fluxo nas linhas e quedas de tensão adequada nos nós consumidores. Ainda é possível escolher um objetivo a otimizar: minimizar o número de chaveamentos, minimizar os desvios das quedas de tensão adequadas, maximizar a capacidade de reserva dos alimentadores e minimizar as perdas elétricas. Também é possível optar pela combinação de vários objetivos, somando-os de forma ponderada na função objetivo. Os resultados reportados, embora adotem uma rede com 36 nós consumidores, indicam o potencial e a flexibilidade do método.

1.2.3 Abordagens heurísticas

Conforme já dito, abordagens empíricas inspiraram o desenvolvimento dos primeiros métodos de resolução para o PRS. Eles usam o conhecimento dos operadores das companhias de distribuição de energia elétrica, que, diante de uma situação de emergência,

adotam certos procedimentos de forma a restabelecer, pelo menos em parte, o fornecimento de energia aos consumidores afetados.

No trabalho de Shirmohammadi (1992) regras heurísticas são utilizadas para determinar o número mínimo de manipulações de chaves para restaurar o serviço em uma área afetada por uma falha. O procedimento baseia-se primeiro na criação de uma rede malhada, por meio do fechamento de todas as chaves. A seguir executa-se um fluxo de carga para redes desse tipo, classificando as chaves quanto ao fluxo de potência que as atravessa. Busca-se então abrir a chave que apresenta o menor fluxo, repetindo-se esse procedimento até encontrar uma rede radial.

Numa abordagem multiobjetivo, Miu et al. (1998) resolvem o PRS considerando a maximização da carga restaurada, a maximização da carga prioritária restaurada e a minimização do número de manipulações de chaves. As restrições consideradas são: radialidade, limites de fluxo, quedas de tensão e capacidade dos alimentadores. Com a criação de índices para a classificação das chaves, as cargas são restauradas sempre orientadas por um eficiente algoritmo de fluxo de potência. Uma importante observação é que o algoritmo é executado de forma independente para cada área sem fornecimento, o que pode levar a erros. A rede utilizada nos testes possui 208 nós consumidores e 83 chaves.

Recentemente, Toune et al. (2002) compararam quatro metaheurísticas para o PRS: algoritmos genéticos (Fukuyama et al. 1996), *simulated annealing* (Mori e Takeda 1994), Busca Tabu e Busca Tabu reativa (Toune et al. 1998). É estabelecida a mesma representação da solução e metodologia para criação de soluções iniciais. A função objetivo corresponde à soma ponderada entre o balanço de carga dos alimentadores e o número de cargas restauradas, considerando a restrição de radialidade, de limites de fluxo, de quedas de tensão e de capacidade dos alimentadores. As conclusões elegem a Busca Tabu reativa como o método mais eficiente, quanto ao número de cargas restauradas e ao tempo gasto.

1.2.4 Abordagens evolutivas

As abordagens evolutivas para o PRS têm sido empregadas como uma alternativa poderosa para tratar sistemas de grande porte, inclusive incorporando restrições muitas vezes desconsideradas, como por exemplo limite de queda de tensão.

Mesmo não tratando especificamente do PRS, o trabalho de Nara, Shiose, Kitagawa e Ishihara (1992) tem grande relevância pelas contribuições apresentadas. O problema tratado é o de reconfiguração para mínimas perdas elétricas, utilizando algoritmos genéticos como método de resolução. Os indivíduos correspondem às posições das chaves envolvidas na reconfiguração. A função objetivo descrita contempla as perdas elétricas de toda

a rede, com as restrições de limites de queda de tensão e capacidade de fluxo incluídas como penalidades na função de avaliação (função de *fitness*). Os resultados obtidos, com redes reais de até 1692 chaves, demonstram que o método é bastante aplicável.

O trabalho de Fukuyama et al. (1996) trata especificamente do PRS e emprega um algoritmo genético paralelo, conforme o modelo de ilhas (Cantú-Paz 1997). Assim como o trabalho de Nara et al. (1992), Fukuyama et al. (1996) também tratam as restrições de limite de capacidade dos alimentadores, de queda de tensão e de fluxo nas linhas como penalidades na função de avaliação dos indivíduos. Um operador de reparação também é proposto, para situações nas quais um indivíduo gerado não corresponde a uma rede radial. Os resultados apresentados indicam a eficiência do método quanto à redução do tempo computacional gasto em relação ao tamanho da rede, ainda que estejam restritos a redes com no máximo 69 nós consumidores.

Um algoritmo genético modificado, com estratégias evolutivas diferenciadas, foi proposto por Augugliaro, Dusonchet e Sanseverino (1998). Entre as duas variações de operadores, uma delas é escolhida baseado na relação entre a capacidade disponível dos alimentadores e a demanda dos nós sem fornecimento. A função objetivo assume certas variações justamente por esses critérios mencionados, ora ressaltando o caráter de minimização das perdas, ora buscando restaurar o serviço para o maior número de consumidores. Todas as restrições são tratadas como penalidades associadas à função de avaliação. São elas: limites de fluxo, queda de tensão e de capacidade dos alimentadores, assim como radialidade da rede proposta como solução. A aplicação do método desenvolvido é realizada em uma rede com 90 nós consumidores e 98 chaves, obtendo-se tempos da ordem de centenas de segundos para resolução em um *mainframe* IBM 3090.

1.2.5 Abordagens multicritério

A procura por uma solução para o PRS corresponde a um compromisso entre critérios conflitantes, como mínima intervenção e maior número de cargas restauradas. Mínima intervenção porque é desejo alterar o mínimo possível a rede, dado que toda alteração realizada precisará ser desfeita quando a rede passar do estado de restauração para o estado normal de operação. O outro objetivo, restaurar o maior número possível de cargas, é, fundamentalmente, o que justifica o problema e está em pouca oposição ao primeiro critério.

Com a adoção de técnicas baseadas em lógica nebulosa, Hsu e Kuo (1994) propuseram uma abordagem que trata variáveis linguísticas para descrever características desejáveis da solução procurada. Embora considerem apenas um objetivo, eles relatam

que a técnica pode ser estendida de forma a contemplar vários. Procurando utilizar um número **razoável** de manipulações de chaves para restaurar as cargas afetadas, são evitadas, ainda que não impedidas, sobrecargas nos alimentadores. Da mesma forma são tratadas restrições como limites de fluxos nas linhas e limites de quedas de tensão nos nós consumidores.

Outra abordagem multicritério é proposta por Lee, Lim e Ahn (1998). Primeiramente eles constroem um conjunto de soluções candidatas, portanto factíveis, obtidas pela aplicação de uma série de esquemas de restauração. A seguir essas soluções são avaliadas baseado nos seguintes critérios: capacidade de restauração⁶, capacidade de reserva, número de chaves manipuladas, número de cargas restauradas considerando sua importância relativa. A solução candidata que obtiver o melhor índice será escolhida como a solução a ser proposta pelo método.

Também utilizando índices, Miu et al. (2000) desenvolveram um método que trata cada área da rede sem fornecimento de energia elétrica separadamente. Para cada uma em particular é executado um algoritmo, o qual contempla, nos seus passos, os índices que irão incorporar os critérios estabelecidos: maximizar a área total restaurada, maximizar a área total prioritária restaurada, minimizar o número de manipulações de chaves, minimizar a distância geográfica e ainda minimizar o número de manipulações nos capacitores. As restrições operacionais tratadas são: limites de fluxo, de queda de tensão e de capacidade dos alimentadores. Os resultados obtidos com uma rede com 201 nós consumidores e 76 chaves demonstram a aplicabilidade da proposta desenvolvida.

Considerando como objetivos a minimização da área sem fornecimento de energia, do número de chaves manipuladas, da corrente das linhas, do desvio em relação ao limite de quedas de tensão aceitáveis e da carga dos transformadores, Hsiao e Chien (2000) propõem um método multiobjetivo que utiliza conjuntos nebulosos e algoritmos genéticos, correspondendo a um método de satisfação iterativo. A rede empregada nos testes é composta de 102 nós consumidores e 217 chaves, com tempos de resolução da ordem de 50 segundos, obtidos com um computador Pentium-CELERON 300A PC.

A utilização de estratégias evolutivas e conjuntos nebulosos é proposta por Augugliaro et al. (2001). A abordagem considera dois objetivos: a maximização da carga restaurada e a minimização das perdas elétricas da configuração proposta. As restrições consideradas compreendem radialidade, limites de fluxo nas linhas, de queda de tensão e de capacidade dos alimentadores. O método de resolução proposto é a combinação de outros dois: estratégias evolutivas *fuzzy* (Augugliaro et al. 2000) e o algoritmo genético multicritério

⁶Definida pelos autores como sendo a capacidade de restauração de um alimentador quando da ocorrência de uma falha depois do processo de restauração.

Non-dominated Sorting Genetic Algorithm (NSGA) (Srinivas e Deb 1994). A mesma rede utilizada em Augugliaro et al. (1998) e Augugliaro et al. (2000) é empregada nos testes computacionais, os quais também compreendem uma comparação entre as três técnicas descritas no artigo (cada um dos dois métodos utilizados e a combinação deles).

Recentemente, Huang (2003) propôs a adoção de redes *fuzzy* causa-efeito para resolução do PRS considerando como critérios e restrições um conjunto de regras obtidas de operadores de sistemas de distribuição de energia elétrica.

Capítulo 2

Otimização Combinatória

Problemas de otimização combinatórios são caracterizados pela consideração de uma seleção ou permutação de um conjunto discreto de itens, ou ainda pela designação que os inclui. Eles têm a forma:

$$\begin{aligned} \textit{Minimizar} \quad & f(x) \\ \textit{sujeito a} \quad & x \in S \end{aligned} \tag{2.1}$$

onde x é uma solução que deve pertencer ao conjunto discreto de soluções S , também chamado de *espaço de soluções* ou *espaço de decisão*. Para muitos problemas combinatórios, o conjunto S é finito. A função f mapeia S para \Re e é chamada de *função objetivo*, descrevendo, portanto, o valor da função objetivo para a solução x .

Problemas combinatórios podem ser computacionalmente intratáveis (Garey e Johnson 1979) e este é o caso se o problema pertencer à classe NP . Esta classe compreende aqueles problemas para os quais não existe um algoritmo que resolva todas as instâncias do problema de forma ótima em um tempo computacional que cresça segundo uma função polinomial do tamanho da instância. A existência deste algoritmo incluirá o problema na classe P .

Problemas NP são definidos através dos problemas de decisão NP -Completo. Um problema de decisão é NP -Completo se o tempo computacional requerido para encontrar a solução para uma instância de pior caso cresce mais rápido com o tamanho dessa instância que qualquer função polinomial. De outra forma, significa responder “sim” ou “não” para a seguinte pergunta:

$$\text{Para um valor } z, \text{ existe uma solução } x \in S \text{ tal que } f(x) \geq z ? \tag{2.2}$$

Resolvendo o problema 2.2 na otimalidade ou respondendo à questão 2.2 será necessário, para problemas *NP-Completo*, um tempo computacional que cresce de forma exponencial com o tamanho do problema. Algoritmos sofisticados podem ser desenvolvidos de forma que requeiram um baixo tempo computacional e isso é já feito para vários problemas *NP*. Entretanto, o tempo sempre crescerá exponencialmente com o tamanho do problema, a menos que $P = NP$.

Um algoritmo que obtenha uma solução ótima para o problema 2.2 é chamado de *algoritmo exato*. Caso não exista necessidade de uma solução ótima, um *algoritmo heurístico* pode ser usado. Este último caracteriza-se por buscar “boas” soluções, não necessariamente ótimas, com um tempo computacional “razoável”.

Quando do desenvolvimento de algoritmos, surgem várias questões práticas, tais como: eficiência, rapidez, legibilidade, de fácil implementação, execução e manutenção, e que forneçam soluções satisfatórias. Estes aspectos justificam o emprego de algoritmos heurísticos, ao invés de algoritmos exatos, em um grande número de casos, correspondendo, inclusive, à única alternativa para certos problemas.

A seção a seguir apresenta certos tipos de algoritmos heurísticos conhecidos como metaheurísticas.

2.1 Metaheurísticas

Uma técnica freqüentemente empregada para resolver problemas de otimização é o uso de regras simples, geralmente associadas ao conhecimento específico do problema que se quer resolver. Um exemplo seria o caso de uma empresa de entrega de mercadorias que utiliza o mesmo veículo para clientes com o mesmo código postal. Essas regras, portanto, descrevem uma abordagem simplificada para um problema até então de difícil solução e são chamadas de *estratégias heurísticas* (Silver, Vidal e Werra 1980).

Estratégias heurísticas podem ser empregadas para desenvolver algoritmos para problemas de otimização. Um algoritmo heurístico construído através de princípios heurísticos é conhecido como *procedimento heurístico* ou simplesmente como uma *heurística*. A origem do termo *heurística* está associada a palavra grega *heuriskein*, que significa encontrar ou descobrir, no entanto seu significado mais comum em otimização é “como buscar boas soluções”.

Enquanto as heurísticas têm sido amplamente utilizadas em otimização para solução de problemas específicos, nos últimos anos houve um maior e crescente interesse nas *metaheurísticas*, ou seja, princípios heurísticos que não sejam específicos mas possíveis de serem empregados a uma grande gama de problemas. A bibliografia relacionada em

Osman e Laporte (1996) apresenta várias referências que abordam tanto aspectos teóricos como práticos das metaheurísticas.

O termo “metaheurísticas” foi citado pela primeira vez por Glover (1986), designando procedimentos iterativos para resolver problemas de otimização combinatória que incluem heurísticas tradicionais como sub-procedimentos. Heurísticas, por sua vez, podem ser definidas, segundo Greenberg (1999), como um procedimento que busca uma solução para um problema, sem a garantia de obter a solução ótima.

Outra importante característica que torna as metaheurísticas técnicas bem diferenciadas das heurísticas tradicionais é a aceitação tanto de movimentos que ocasionem soluções melhores como também daqueles que geram soluções piores que a solução atual (Pardalos e Resende 2001). Quando da resolução de problemas que têm muitos ótimos locais, mas que se busca um único ótimo global (funções multimodais), essa potencialidade torna-se fundamental para superar a principal limitação que as heurísticas tradicionais apresentam, qual seja, de convergir para um ótimo local.

As metaheurísticas apresentam um compromisso fundamental que as difere de outras técnicas para resolução de problemas de otimização combinatória: o compromisso entre exploração (diversificação) e exploração (intensificação). Esta primeira característica se refere à amplitude da busca, ou seja, cobrir a maior parte possível do espaço. A última é justamente a capacidade de pesquisar completamente uma região restrita do espaço de busca. Nos métodos descritos nas seções seguintes serão melhor especificadas essas características.

Para um perfeito entendimento das metaheurísticas que serão apresentadas a seguir, alguns conceitos fundamentais relativos a essas técnicas são relacionados.

2.1.1 Conceitos básicos

Três conceitos básicos são comuns para qualquer técnica algorítmica para resolução de problemas (Michalewicz e Fogel 2000): (1) a representação, (2) o objetivo e (3) a função de avaliação. A representação codifica soluções candidatas alternativas para manipulação, o objetivo descreve os requisitos que devem ser atendidos e a função de avaliação retorna um valor específico que indica a qualidade de uma determinada solução, a partir da representação.

A seguir considera-se cada um destes três aspectos com mais detalhes, além dos conceitos de vizinhança, ótimo local e busca local.

2.1.1.1 Representação

Para cada problema, a representação de uma solução e a sua respectiva interpretação determinam o espaço de busca. Esse espaço corresponderá ao conjunto de todas as soluções possíveis de serem encontradas.

Definir o espaço de busca mais correto é de fundamental importância para o sucesso de qualquer proposta de resolução de um problema. Caso o domínio mais adequado não seja selecionado para iniciar a busca, várias soluções inviáveis ou até mesmo duplicadas podem fazer parte da lista de possibilidades, prejudicando a eficiência da busca ou ainda a correteza da solução encontrada.

Como exemplo, considera-se o problema do caixeiro viajante (PCV), conceitualmente muito simples: o caixeiro viajante deve visitar todas as cidades de seu território exatamente uma vez, retornando a sua cidade de origem e percorrendo a menor distância possível. Supondo um conjunto de n cidades no território do caixeiro, uma possível representação da solução para este problema seria a permutação dos números naturais $1, \dots, n$ onde cada número corresponde a uma cidade. Dessa forma, o espaço de busca consiste de todas as permutações, da ordem de $(n - 1)!$. Caso seja considerado o problema simétrico, onde o custo de viajar da cidade i para a cidade j é o mesmo em qualquer direção, o espaço de busca pode ser reduzido pela metade. Ainda nesse caso, o circuito será o mesmo, independente da cidade que se partir, reduzindo o espaço de busca por um fator n . Conseqüentemente, o tamanho real do espaço de busca é reduzido para $(n - 1)!/2$, embora este número seja extremamente grande até mesmo para problemas “médios”.

2.1.1.2 Objetivo

A partir da definição do espaço de busca, é preciso definir o que se busca com uma solução para o problema em questão. O objetivo do problema é uma sentença matemática que expressa a tarefa a ser alcançada, correspondendo a uma expressão e não a uma função.

Considerando-se o exemplo do PCV, o objetivo típico corresponde a minimização da distância total percorrida pelo caixeiro, podendo ser escrito em termos matemáticos da seguinte forma: $\min \sum dist(x, y)$, onde $dist$ é a distância euclideana entre as cidades x e y , pertencentes à rota.

2.1.1.3 Função de avaliação

O objetivo não corresponde exatamente à função de avaliação. A última representa tipicamente um mapeamento do espaço de soluções candidatas possíveis para um número (por exemplo real), onde à cada elemento desse espaço é atribuído um valor que indica

sua qualidade. A função de avaliação permite comparar o valor de soluções alternativas conforme a modelagem adotada. Algumas funções criam uma classificação de todas as soluções possíveis, outras informam não só a ordem das soluções em termos da sua qualidade como também o grau desta qualidade.

O objetivo no PCV é minimizar a soma das distâncias da rota percorrida, respeitando as restrições do problema. Uma possível função de avaliação pode mapear cada rota para a sua correspondente distância total. Poder-se-iam comparar duas rotas e saber não somente que uma é melhor que outra, mas também exatamente quão melhor. Por razões de eficiência computacional, é desejável que uma função de avaliação considere tão somente duas soluções, retornando qual das duas é a mais desejável.

2.1.1.4 Vizinhaça e ótimo local

Caso seja escolhida uma certa região do espaço de busca que seja “perto” de um determinado ponto, pode-se dizer que se está verificando a vizinhaça desse ponto. Pela Figura 2.1 descreve-se o espaço de busca S , com um ponto $x \in S$. Uma vizinhaça $N(x)$ de x é um conjunto de todos os pontos do espaço de busca S que estão “próximos” do ponto x , quando uma certa medida é considerada.

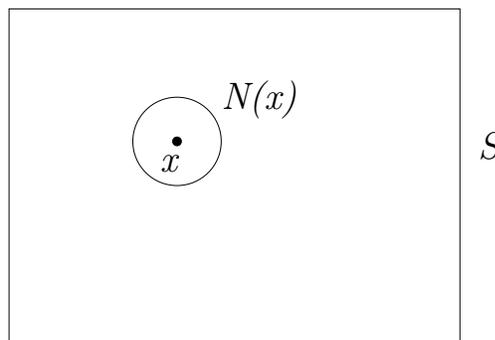


Figura 2.1: O espaço de busca S , uma possível solução x e sua vizinhaça $N(x)$ representada pelo interior do círculo que contém x .

A definição da proximidade de um ponto pode ser feita basicamente de duas formas:

- Através de uma função de distância no espaço de busca S :

$$dist: S \times S \rightarrow \mathfrak{R} ,$$

e então definir a vizinhaça $N(x)$ como:

$$N(x) = \{y \in S : \text{dist}(x, y) \leq \epsilon\},$$

para algum $\epsilon \geq 0$.

- Definindo um mapeamento m no espaço de busca S como

$$m: S \rightarrow 2^S,$$

e este mapeamento define a vizinhança para qualquer ponto $x \in S$. Considerando-se o exemplo do PCV, pode-se definir um mapeamento que corresponda à troca de posição de duas cidades da rota como sendo a vizinhança de qualquer solução x do espaço S , conforme o exemplo a seguir:

15-3-11-19-17-2-...-6-15 (solução original),

15-3-17-19-11-2-...-6-15 (solução vizinha após a troca entre a terceira e quinta cidades).

Dessa forma, para cada solução candidata, existem $\frac{n(n-1)}{2}$ vizinhos, portanto uma vizinhança $O(n^2)$.

Com a noção de vizinhança é possível enunciar o conceito de *mínimo local*, ou *ótimo local*. Diz-se que uma solução $x \in S$ é um ótimo local com respeito à vizinhança N se, e apenas se, $eval(y) \geq eval(x)$, para todo $y \in N(x)$ (assumindo o critério de minimização), sendo $eval(\cdot)$ uma função de avaliação capaz de avaliar cada elemento do conjunto $N(x)$.

Muitos métodos de busca são baseados em vizinhanças de pontos. Ou seja, a seqüência de pontos que estes métodos geram enquanto buscam a melhor solução possível baseia-se na informação de ótimos locais a cada iteração. Eles são conhecidos como estratégias de busca *local* ou em *vizinhança*, e são tratados na próxima seção.

2.1.1.5 Métodos de busca local

Um método de busca local compreende quatro passos básicos:

1. Tome uma solução do espaço de busca e avalie-a. Defina-a como solução incumbente.
2. Aplique transformações na solução incumbente para gerar uma nova solução, avaliando-a a seguir.
3. Caso a nova solução seja melhor que a solução incumbente, substitua esta solução pela nova. Caso contrário descarte a nova solução.
4. Repita os passos 2 e 3 até que nenhuma transformação melhore a solução incumbente.

A transformação provocada na solução incumbente no passo 2 corresponde ao mapeamento que define a vizinhança dessa solução no espaço de busca. A partir da geração de todas as soluções vizinhas à solução incumbente, toma-se a melhor para torná-la a solução incumbente da próxima iteração. Caso não seja encontrada nenhuma solução melhor que a incumbente, pára-se a busca e diz-se que um mínimo local foi obtido.

Estes métodos caracterizam-se pela existência de um compromisso muito interessante entre o tamanho da vizinhança $N(x)$ e a eficiência da busca. Caso o tamanho da vizinhança seja pequeno, o método será capaz de verificar rapidamente essa vizinhança. Apenas poucas soluções potenciais serão avaliadas antes da tomada de decisão que elegerá a nova solução para o próximo passo. No entanto, essa visibilidade limitada aumenta a chance de levar a busca rapidamente para um mínimo local. Isto sugere o uso de vizinhanças maiores: uma maior visibilidade será de grande valia para tomar decisões melhores. O número de avaliações, entretanto, pode se tornar muito elevado, o que inviabiliza o algoritmo quando considerado o tempo computacional. Cabe salientar que o conceito de ótimo local depende da definição da vizinhança.

Por isso, a definição do tamanho da vizinhança é um dos maiores desafios de um método de busca local: aliar eficiência com eficácia. Para tanto, deve-se usar todo o conhecimento a respeito da representação adotada para o problema, assim como da função de avaliação escolhida.

Nas próximas seções serão descritas algumas técnicas que fazem uso de métodos de busca local, as quais têm o propósito de buscar outras regiões do espaço de busca a partir dos eventuais mínimos locais encontrados.

2.1.2 *Simulated Annealing*

Simulated Annealing (SA) (Aarts e Korst 1989) é uma técnica estocástica de busca local para encontrar soluções ótimas ou quase ótimas. Ela é baseada no processo físico de cristalização de materiais sólidos. Este processo consiste em dois passos básicos: primeiramente os materiais são aquecidos até derreterem; depois a temperatura é reduzida cuidadosamente até que as moléculas do sólido derretido se arranjam e formem o estado base deste sólido, também conhecido como estado de mínima energia. Do estado líquido para o estado sólido, o arranjo das moléculas do sistema passa de totalmente aleatório até ao nível de maior estruturação, com a conseqüente redução da energia na medida em que a estruturação aumenta e a temperatura diminui.

O processo físico de cristalização pode ser modelado facilmente em um ambiente computacional utilizando-se métodos de busca local. O algoritmo computacional baseia-se na

geração de uma seqüência de estados para o sólido da seguinte forma: dado um estado atual i de um sólido com energia E_i , um estado j subsequente é gerado aplicando-se um mecanismo de perturbação (busca em vizinhança). Caso a diferença de energia, $E_j - E_i$, seja menor que 0, o estado j é aceito como novo estado atual. Caso a diferença de energia seja maior ou igual a 0, o estado j é aceito com uma probabilidade dada por:

$$\exp\left(\frac{E_i - E_j}{k_b T}\right),$$

onde T corresponde à temperatura do sistema e k_b a uma constante conhecida como constante de *Boltzmann*. Este critério corresponde à regra de aceitação do próximo estado do sistema.

Considerando-se o método de SA, pode-se aplicar a regra de aceitação para gerar uma seqüência de soluções para o problema de otimização combinatória. Para tanto, assume-se uma analogia entre o sistema físico e o problema de otimização combinatória cuja função objetivo seja de minimização e com as seguintes equivalências:

- Soluções do problema de otimização combinatória são equivalentes aos estados do sistema físico.
- O valor de uma solução é equivalente à energia de um estado.
- Soluções vizinhas, obtidas pelo mecanismo de perturbação, correspondem aos estados subsequentes.

Com isso pode-se deixar a metáfora inspiradora (o processo físico) e formular o método de SA como um algoritmo baseado em busca local. A Figura 2.2 apresenta um pseudo-código de um algoritmo de SA. O contador de iterações t e a temperatura T são inicializadas nos passos 1 e 2, respectivamente. A seleção de uma solução aleatória é realizada no passo 3, seguida da sua respectiva avaliação, passo 4.

A seguir iniciam-se dois laços: o mais externo, entre os passos 5 e 15, irá determinar o número de iterações do algoritmo, sendo encerrado quando o critério de parada for satisfeito; o mais interno, entre os passos 7 e 13, representa a busca local propriamente dita, que atualiza a solução atual Sc conforme a regra de aceitação estabelecida, sendo encerrado quanto a condição de saída for satisfeita.

A substituição da solução atual pela nova solução é realizada nos passos 11 e 13. No passo 11 será utilizado o critério do melhor custo, enquanto que no passo 13 será gerado um número aleatório e este será comparado com a probabilidade Pa de aceitação da nova solução, dada pela seguinte equação:

$$Pa = e^{\frac{eval(Sn) - eval(Sc)}{T}}$$

A redução da temperatura é determinada por uma função G , passo 14, que toma como parâmetros a temperatura atual do sistema e o número da iteração atual do algoritmo, atualizada no passo 15. Por fim, no passo 16 é retornada a melhor solução encontrada.

```

SIMULATED ANNEALING()
1   $t \leftarrow 0$ ;
2  INICIALIZA( $T$ );
3   $Sc \leftarrow$  SELECIONAALEATORIO();
4  AVALIA( $Sc$ )
5  enquanto  $criterioDeParada == falso$ 
6  faça
7      enquanto  $condicaoDeSaida == falso$ 
8      faça
9           $Sn \leftarrow$  SELECIONAVIZINHO( $Sc$ )
10         se  $eval(Sn) < eval(Sc)$ 
11             então  $Sc \leftarrow Sn$ ;
12         senão se  $random[0, 1) < Pa$ 
13             então  $Sc \leftarrow Sn$ ;
14      $T \leftarrow G(T, t)$ ;
15      $t \leftarrow t + 1$ ;
16 retorna  $Sc$ 

```

Figura 2.2: Pseudo-código de um algoritmo de *Simulated Annealing*.

A idéia fundamental do algoritmo é realizar iterativamente várias buscas locais que fornecem novas soluções, eventualmente adotadas como solução atual sempre que aquelas forem melhores que esta última ou quando a probabilidade de aceitação for maior que um número gerado aleatoriamente. A redução da temperatura afeta justamente essa probabilidade de aceitação, tornando o algoritmo mais seletivo porque diminui as chances de deterioração da solução até um nível em que somente são aceitas soluções que tiverem um custo melhor que a solução atual.

Nesta técnica, portanto, não há memória para armazenar a melhor solução encontrada ao longo do processo de busca. Espera-se que a melhor solução não seja perdida pela aceitação de um novo estado. Entretanto, algumas abordagens que utilizam *Simulated Annealing* armazenam essa melhor solução.

2.1.3 Busca Tabu

A palavra tabu tem origem na Polinésia e era usada pelos aborígenas da ilha de Tongan para indicar coisas intocáveis pelo seu caráter sagrado. No dicionário Aurélio da Língua Portuguesa, a palavra tabu tem o seguinte significado: “proibição convencional imposta por tradição ou costume a certos atos, palavras, temas, etc.”. A associação importante com o uso tradicional da palavra vem do fato de que o caráter tabu pode ser modificado ou adaptado ao longo do tempo, justamente pela relação direta com o aspecto social das relações humanas.

Dessa forma, a ligação fundamental criada entre a origem da palavra tabu e a técnica de *Busca Tabu* (BT) (Glover (1989); Glover (1990); Glover e Laguna (1997)) corresponde ao uso de memória adaptativa, modificável ao longo do processo de busca, que caracterizará movimentos proibidos (tabus). O caráter adaptativo dessa memória permite percorrer o espaço de busca de forma econômica e efetiva, combinando informação local e história de busca.

A estrutura de memória adaptativa utilizada apresenta quatro dimensões principais: recência, frequência, qualidade e influência. As dimensões de recência e frequência são complementares e têm como objetivo a manutenção de atributos de soluções que foram modificados considerando-se um passado próximo. A dimensão de qualidade mantém atributos que foram determinantes quando da obtenção de boas soluções. A última dimensão, influência, considera o impacto que as escolhas feitas durante a busca tiveram não apenas na qualidade como também na estrutura das soluções encontradas.

O conteúdo da memória utilizada pela BT é tanto explícito quanto atributivo. A parte explícita armazena soluções completas, tipicamente soluções de elite visitadas durante a busca. Uma extensão dessa memória explícita armazena vizinhos promissores dessas soluções de elite que não foram explorados. A parte atributiva mantém atributos de soluções que mudaram quando passou-se de uma solução para outra. Considerando o exemplo do PCV, um possível atributo seria a posição de uma cidade na rota.

A Figura 2.3 apresenta um pseudo-código de um algoritmo de BT genérico. A memória utilizada é bem simplificada, reduzida à lista tabu que armazena movimentos proibidos.

O algoritmo da Figura 2.3 é composto de dois laços: o primeiro, mais externo, entre os passos 3 e 19, se ocupa em gerar várias soluções iniciais (passo 6) para a busca local; o segundo laço, mais interno, entre os passos 8 e 16, realiza a busca local guiada, ou seja, com os movimentos fornecidos pela busca local e com a lista tabu é possível obter soluções nem sempre melhores que a solução atual. No passo 10 é criado um conjunto (*MOV*) com os melhores movimentos da vizinhança da solução atual *Sol*. Dentre estes

movimentos será selecionado apenas um (*mov*) que não for proibido pela lista tabu e não satisfizer o critério de aspiração, passo 11. A realização do movimento selecionado é feita no passo 12, com a posterior atualização da lista tabu, passo 13. A melhor solução obtida durante a busca local guiada é armazenada nos passos 14 e 15, e a melhor solução de toda a busca nos passos 17 e 18. O passo 20 encerra o algoritmo, devolvendo a melhor solução encontrada.

Percebe-se que há uma preocupação em garantir que a melhor solução encontrada pela busca seja mantida ao longo do processo. A definição dos seguintes parâmetros, ainda que seja uma tarefa muito difícil, terá grande influência na eficiência do algoritmo desenvolvido:

- Tamanho da lista tabu;
- Duração tabu: quantas iterações um movimento permanecerá na lista;
- Condição de saída da busca local guiada (*MAX_ITERACOES*);
- Condição de parada do algoritmo (*MAX_TENTATIVAS*).

2.1.4 Algoritmos evolutivos

Algoritmos evolutivos (AE) são uma classe de algoritmos de otimização estocástica que simulam o processo de evolução natural. Desde os anos 70, várias metodologias de resolução foram propostas, a maioria pertencente a uma destas três classes: algoritmos genéticos, programação evolutiva e estratégias evolutivas (Fogel (1995); Bäck, Fogel e Michalewicz (2000)). Uma das características principais destas técnicas é o uso de um conjunto de soluções candidatas (população), as quais são modificadas aplicando-se dois princípios básicos: seleção e perturbação. A seleção imita o processo de competição para perpetuar a espécie e para dispor de recursos limitados no ambiente evolutivo. O outro princípio, perturbação, corresponde a capacidade natural de criar novos indivíduos através de recombinação e mutação.

A seguir são apresentados os conceitos associados a estas técnicas e sua relação com a metáfora inspiradora da evolução natural:

- *Indivíduos*: representam organismos genéticos que farão parte do processo evolutivo;
- *Seleção*: representa a máxima em evolução natural, onde somente os mais adaptados sobrevivem e se perpetuam;

```
BUSCATABU()
1   $t \leftarrow 0$ ;
2   $MelhorSol \leftarrow 0$ ;
3  enquanto  $t < MAX\_TENTATIVAS$ 
4  faça
5       $MelhorSolLocal \leftarrow 0$ ;
6       $Sol \leftarrow GERASOLUCAO()$ ;
7       $contador \leftarrow 0$ ;
8      enquanto  $contador < MAX\_ITERACOES$ 
9      faça
10          $MOV \leftarrow IDENTIFICAMOVIMENTOS(Sol)$ ;
11          $mov \leftarrow SELECIONAMELHORMOVIMENTOADMISSIVEL(MOV)$ ;
12          $Sol \leftarrow REALIZAMOVIMENTO(mov, Sol)$ ;
13          $ATUALIZALISTATABU(mov, Sol)$ ;
14         se  $eval(Sol) < eval(MelhorSolLocal)$ 
15             então  $MelhorSolLocal \leftarrow Sol$ ;
16          $contador \leftarrow contador + 1$ ;
17     se  $eval(MelhorSolLocal) < eval(MelhorSol)$ 
18         então  $MelhorSol \leftarrow MelhorSolLocal$ ;
19      $t \leftarrow t + 1$ ;
20 retorna  $MelhorSol$ 
```

Figura 2.3: Pseudo-código de um algoritmo de *Busca Tabu*.

- *Recombinação*: é o mecanismo natural para que uma espécie seja perpetuada, o qual compreende a transferência de material genético dos ancestrais para seus descendentes;
- *Mutação*: representa a garantia de que erros ocorrem durante a fase de transferência de material genético.

Embora existam as várias classes de AEs, é possível descrever um algoritmo básico comum a todas, conforme ilustra a Figura 2.4 (Michalewicz 1996). No primeiro passo, a população é inicializada, geralmente é utilizada aleatoriedade de modo a permitir maior diversidade. A seguir, no passo 2, ocorre a avaliação dos indivíduos. O laço entre os passos 4 e 9 corresponde às etapas incluídas em cada geração do algoritmo: no primeiro, passo 5, a população é perturbada; no segundo, passo 6, a população é avaliada; finalmente no passo 7 a população sofre uma seleção com o intuito de obter somente aqueles indivíduos mais adaptados segundo a função de avaliação; no último passo do laço, passo 8, será realizado o incremento do contador de iterações. Esta seqüência será repetida até que o critério de parada seja satisfeito.

```
ALGÉVOLUTIVO()
1   $P \leftarrow \text{CONSTROIPOPINICIAL}()$ ;
2   $\text{AVALIE}(P)$ ;
3   $\text{geracoes} \leftarrow 0$ ;
4  enquanto  $\text{Criteriodeparada} = \text{falso}$ 
5  faça  $\text{ALTERE}(P)$ ;
6       $\text{AVALIE}(P)$ ;
7       $\text{SELECIONE}(P)$ ;
8       $\text{geracoes} \leftarrow \text{geracoes} + 1$ ;
9  retorna  $P$ ;
```

Figura 2.4: Algoritmo Evolutivo.

Observa-se na literatura especializada do PRS um grande número de publicações que empregam algoritmos evolutivos. Tal fato se justifica pela alta complexidade associada aos problemas de otimização combinatória, limitando ou até mesmo impedindo o uso de métodos exatos (Lewis e Papadimitriou 1981). Além disso, há casos em que o problema a ser resolvido apresenta características como não-linearidade e não-diferenciabilidade, impedindo a aplicação de métodos tradicionais (método do gradiente, técnicas de programação linear, etc.). Neste cenário, algoritmos evolutivos apresentam-se como uma boa

alternativa para obtenção de soluções aproximadas de problemas com grande complexidade associada.

Capítulo 3

Otimização combinatória multiobjetivo

Um Problema de Otimização Combinatória Multiobjetivo (POCM) consiste em minimizar (ou maximizar) simultaneamente um conjunto de critérios (objetivos), satisfazendo um conjunto de restrições. Para estes problemas não existe uma única solução que seja ótima com relação a todos os objetivos. Há, portanto, um conjunto de soluções eficientes no qual nenhuma é melhor que outra com relação a todos os objetivos. Ao decisor cabe escolher uma ou mais soluções desse conjunto que melhor o satisfaça, considerando, por exemplo, os objetivos gerais por ele escolhidos ou uma função utilidade pré-definida.

Esta seção apresenta formalmente os problemas de otimização multiobjetivo, com seus respectivos conceitos básicos. Também são apresentados métodos clássicos que obtêm soluções Pareto-ótimas bem como alguns métodos para avaliação de heurísticas multiobjetivo.

3.1 Problemas de otimização multiobjetivo

O desenvolvimento de um modelo matemático para problemas práticos de decisão pode envolver grandes dificuldades justamente na formulação ou na escolha da função objetivo. Vários atributos que denotam maior ou menor qualidade da solução podem não ser facilmente combinados em uma única função objetivo. Neste caso, procede-se de duas formas:

1. Construir uma função objetivo de agregação que combine esses atributos;
2. Considerar os vários critérios de forma explícita no modelo desenvolvido.

A segunda opção, que prevê a modelagem explícita de várias funções objetivo, corresponde à área da *tomada de decisão multicritério* (TDM). Neste trabalho aborda-se somente a parte matemática da TDM, também chamada de *otimização multiobjetivo*. O problema a ser considerado apresenta-se da seguinte forma:

$$\begin{aligned} \text{Minimizar } f(x) &= \left\{ \begin{array}{c} f_1(x) \\ f_2(x) \\ \dots \\ f_n(x) \end{array} \right\} & (3.1) \\ \text{Sujeito a } x &\in D \end{aligned}$$

onde n é o número de funções objetivo e x um vetor com as n variáveis de decisão. Dependendo da natureza das funções incluídas, assim como das restrições, o problema 3.1 pode pertencer a diferentes classes. Um modelo com variáveis de decisão reais e funções e restrições lineares corresponderá a um problema multiobjetivo linear. Ele será um problema multiobjetivo não-linear se as funções e/ou as restrições forem não-lineares. Caso as variáveis de decisão sejam inteiras, o problema será um problema multiobjetivo inteiro, podendo ser ainda linear ou não-linear.

Uma notação mais compacta pode incluir o espaço objetivo $z = (z_1, z_2, \dots, z_n)$, onde $z_j = f_j(x)$. Seja Z a imagem de D no espaço objetivo, ou seja, $z \in Z \leftrightarrow \exists x \in D f(x) = z$. O problema 3.1 pode ser reescrito da seguinte forma:

$$\begin{aligned} \text{Minimizar } z &= \left\{ \begin{array}{c} z_1 \\ z_2 \\ \dots \\ z_n \end{array} \right\} & (3.2) \\ \text{Sujeito a } z &\in Z \end{aligned}$$

Um importante conceito é a *relação de dominância*, que se refere ao espaço objetivo. Um ponto z^1 domina z^2 ($z^1 \succ z^2$) se $z_j^1 \leq z_j^2$, para todos os objetivos $j \in \{1, 2, \dots, n\}$ e se $z^1 \neq z^2$. Quando um ponto não é dominado por nenhum outro em Z , este ponto é dito *não-dominado*. Uma solução x que seja mapeada por f para um ponto não-dominado é dita solução *Pareto ótima*, *eficiente* ou solução *não-inferior*. Estas soluções serão as relevantes quando da otimização multiobjetivo (Steuer 1986).

Quando são considerados problemas de otimização combinatória (apresentados no capítulo 2) com várias funções objetivo, eles são ditos *problemas de otimização combinatória multiobjetivo* (POCM), podendo ser modelados como problemas inteiros com

múltiplos objetivos e formulados como o problema 3.1, com o conjunto S sendo discreto. Como a cardinalidade de Z nunca pode ser maior que a cardinalidade do conjunto S , o conjunto Z será finito sempre que S o for.

Serafini (1987) provou a *NP-Compleitude* para os POCM, cuja definição corresponde a um problema de decisão mono-objetivo descrito em 2.2 com a adequada reinterpretação da desigualdade, conforme segue:

$$\text{Para um ponto } z, \text{ existe uma solução } x \in D \text{ tal que } f(x) \geq z? \quad (3.3)$$

Um POCM é dito ser um problema *NP* se a determinação da dominância ou não de um ponto não possa ser feita em um tempo polinomial para todas as instâncias do problema. O POCM correspondente a um problema de otimização combinatória mono-objetivo 2.2 pode ser definido com o mesmo conjunto D de soluções factíveis, sendo cada função objetivo do POCM com a mesma estrutura da função descrita em 2.2. Segundo a definição 3.3, um POCM será um problema *NP* se o seu POCM correspondente a um problema de otimização combinatória mono-objetivo for *NP*.

A seguir são apresentadas algumas definições considerando que os problemas são de minimização.

3.1.1 Definições

O problema multiobjetivo é formulado genericamente da seguinte forma:

$$\begin{aligned} \text{Minimizar } \{f_1(x) = z_1, \dots, f_n(x) = z_n\} \\ \text{Sujeito a } x \in D, \end{aligned} \quad (3.4)$$

onde uma solução $x = [x_1, \dots, x_I]$ é um vetor com as variáveis de decisão e D é o conjunto das soluções factíveis. O interesse particular deste trabalho encontra-se nos problemas de otimização combinatória multiobjetivo, em que as variáveis de decisão são binárias e o conjunto D apresenta uma estrutura específica.

A imagem de uma solução x no espaço objetivo é um ponto. $z^x = [z_1^x, \dots, z_n^x] = f(x)$, tal que $z_j^x = f_j(x)$, $j = 1, \dots, n$.

Solução Pareto-ótima, conjunto Pareto-ótimo e fronteira de Pareto

Uma solução $x \in D$ é *Pareto-ótima* (eficiente) se não existe uma solução $x' \in D$ que domine x . Um ponto que é a imagem de uma solução Pareto-ótima é chamado

não-dominado. O conjunto PO com todas as soluções Pareto-ótimas é dito *conjunto Pareto-ótimo*. A imagem ND do conjunto Pareto-ótimo no espaço objetivo é chamado *conjunto não-dominado* ou *fronteira de Pareto*.

Uma aproximação do conjunto não-dominado é um conjunto de pontos A tal que $\nexists z^1, z^2 \in A$ tal que $z^1 \succ z^2$, ou seja, o conjunto A é composto de pontos mutuamente não-dominados.

Ponto ideal

Um ponto z^* composto dos melhores valores para cada função objetivo é dito *ponto ideal*.

$$z_j^* = \min \{f_j(x) | x \in D\} \quad j = 1, \dots, n. \quad (3.5)$$

Ponto nadir

Um ponto z_* composto dos piores valores para cada função objetivo é dito *ponto nadir*.

$$z_*^j = \max \{f_j(x) | x \in D\} \quad j = 1, \dots, n. \quad (3.6)$$

Fatores normalizadores

Fatores normalizadores (Steuer 1986, cap. 8.4.2) são definidos da seguinte forma:

$$\pi_j = \frac{1}{R_j}, \quad j = 1, \dots, n \quad (3.7)$$

onde R_j é o domínio do objetivo j no conjunto D , ND ou A . Valores de função objetivo multiplicados pelos fatores de normalização são chamados *valores de função objetivo normalizados*.

Função utilidade

Uma *função utilidade* $u : \mathfrak{R}^n \rightarrow \mathfrak{R}$ é uma representação das preferências do decisor que mapeia cada ponto do espaço objetivo a um valor de utilidade. Embora seja usual a maximização da função utilidade, assume-se, nesta tese, a minimização dessa função.

Diz-se que uma função utilidade u é compatível com a relação de dominância se, e somente se, $\forall z^1, z^2 \in \mathfrak{R}^n$ $z^1 \succ z^2 \Rightarrow u(z^1) \leq u(z^2)$. O conjunto de todas as funções utilidade que são compatíveis com a relação de dominância é descrito por U_c .

Uma função utilidade u é estritamente compatível com a relação de dominância se, e somente se, $\forall z^1, z^2 \in \mathfrak{R}^n$ $z^1 \succ z^2 \Rightarrow u(z^1) < u(z^2)$. O conjunto de todas as funções

utilidade que são estritamente compatíveis com a relação de dominância é descrito por U_{sc} .

Pode-se definir um conjunto de funções utilidade através do uso de funções utilidade paramétricas $u(z, r)$, $r \in D(r) \subseteq \mathfrak{R}^n$, onde r é um vetor de parâmetros e $D(r)$ o seu respectivo domínio. Dessa forma:

$$U(r) = \{ u(z, r) \mid r \in D(r) \} \quad (3.8)$$

Funções lineares ponderadas

Funções lineares ponderadas são definidas da seguinte forma:

$$s_1(z, \lambda) = \sum_{j=1}^n \lambda_j z_j = \sum_{j=1}^n \lambda_j f_j(x) \quad (3.9)$$

onde $\lambda = [\lambda_1, \dots, \lambda_n]$ é um vetor de pesos tal que $\lambda_j \geq 0 \forall j$ e $\exists j \mid \lambda_j > 0$. Cada função linear ponderada tem um único mínimo (máximo) global pertencente ao conjunto de soluções Pareto-ótimo (Steuer 1986). Uma solução Pareto-ótima que é um mínimo global para uma função linear ponderada é chamada de solução Pareto-ótima *de apoio* e sua imagem no espaço objetivo é chamada de *ponto de apoio* (Ulungu e Teghem 1994). No caso de um conjunto D não-convexo, por exemplo em problemas de Otimização Combinatória Multiobjetivo, existem soluções Pareto-ótimas de suporte que não são ótimos globais de $s_1(z, \lambda)$ para algum λ que atenda as restrições anteriores.

Funções Tchebycheff ponderadas

Funções Tchebycheff ponderadas são definidas da seguinte forma:

$$s_\infty(z, z^0, \lambda) = \max_j \{ \lambda_j (z_j - z_j^0) \} = \max_j \{ \lambda_j (f_j(x) - z_j^0) \}, \quad (3.10)$$

onde z^0 é um ponto de referência, $\lambda = [\lambda_1, \dots, \lambda_n]$ é um vetor de pesos tal que $\lambda_j \geq 0 \forall j$ e $\exists j \mid \lambda_j > 0$. Cada função Tchebycheff ponderada tem ao menos um ótimo global (mínimo) pertencente ao conjunto de soluções Pareto-ótimas. Para cada solução Pareto-ótima x existe uma função Tchebycheff ponderada s tal que x é um ótimo global (mínimo) de s (Steuer 1986, cap. 14.8).

3.2 Classificação de métodos para avaliação multiobjetivo

Quando consideram-se problemas de otimização multiobjetivo, qualquer método sempre restringe-se a encontrar um conjunto de soluções eficientes. O método, portanto, nada mais faz do que sugerir um conjunto de soluções ao decisor que deverá ponderar os aspectos que julgue mais importantes com relação aos objetivos conflitantes.

Assim, identificam-se duas etapas básicas: a busca de soluções (responsabilidade do método) e a tomada de decisões (responsabilidade do decisor). A primeira etapa corresponde à busca de soluções na região factível em direção à fronteira Pareto-ótima, tarefa essa que pode se tornar penosa devido ao tamanho e complexidade do espaço de busca. A segunda etapa, tão importante quanto a primeira, corresponde à escolha de um critério para determinar qual das várias soluções eficientes deverá ser eleita para ser implantada.

A seguir identificam-se as três categorias de métodos considerando-se o ponto-de-vista do decisor.

Métodos a-priori

Métodos desse tipo caracterizam-se pela participação do decisor antes de resolver o problema. Há duas abordagens comumente empregadas:

1. *Combinação*: essa abordagem emprega a combinação dos objetivos envolvidos no problema de forma a deixá-lo como um problema mono-objetivo. Para tanto, é preciso determinar a importância de cada objetivo através da atribuição de seus respectivos pesos associados.
2. *Decomposição*: nesse caso os objetivos são elencados de acordo com as respectivas prioridades. A seguir o problema é resolvido para cada um deles, independentemente. No entanto, para a resolução do segundo objetivo, por exemplo, é imposta uma restrição que se refere ao valor ótimo encontrado no primeiro. Considerando-se um exemplo com apenas dois objetivos, tem-se:

$$\begin{aligned}
 & \text{Minimizar} && f_2(x) \\
 & \text{Sujeito a :} && \\
 & && f_1(x) \leq f_1^* \\
 & && x \in X^*
 \end{aligned} \tag{3.11}$$

onde f_1^* é a solução do problema:

$$\begin{aligned} & \text{Minimizar} && f_1(x) \\ & \text{Sujeito a :} && \\ & && x \in X^* \end{aligned}$$

Métodos a-posteriori

Para esses métodos, a ordem “natural” entre as etapas de busca e de decisão é mantida, ou seja, é realizada primeiramente a busca por soluções Pareto-ótimas, seguida do processo de decisão, responsável pela escolha da solução mais adequada dentre as soluções do conjunto obtido no processo anterior. Uma possibilidade seria a definição de uma função de utilidade por parte do decisor.

Métodos iterativos

Esses métodos caracterizam-se por eliminar a separação entre as etapas de busca e de decisão. O processo de busca é guiado para regiões potencialmente promissoras através da determinação de características desejáveis de uma solução.

3.3 Métodos clássicos de otimização multiobjetivo

Como já foi discutido anteriormente, a dificuldade inerente à otimização multiobjetivo é a existência de objetivos conflitantes. Ainda que as soluções ótimas para cada objetivo, tomado separadamente, sejam diferentes, o decisor precisa de soluções que apresentem certas características com relação aos objetivos considerados.

Os métodos clássicos (Cohon 1978; Steuer 1986) encontram tais soluções escalarizando os objetivos, formando apenas um. O resultado é a otimização de um escalar e não de um vetor. Os problemas decorrentes são o estabelecimento sistemático de parâmetros e algumas restrições adicionais.

A seguir são apresentados dois métodos que fazem uso dessa técnica.

3.3.1 Método da soma ponderada

Justamente por ser tão simples e intuitivo, pode ser dito que este método é o mais empregado quando da utilização de otimização multiobjetivo. A técnica consiste em transformar os vários objetivos em um só, com o seguinte problema escalar resultante:

$$\begin{aligned} \text{Minimizar } f(x) &= \sum_{i=1}^r w_i \cdot f_i(x) \\ \text{Sujeito a :} & \\ &x \in X^* \end{aligned} \tag{3.12}$$

No problema 3.12, $w_i \geq 0$ é o peso que representa a importância relativa do objetivo f_i comparado com os outros. Há, em geral, uma normalização desses pesos da seguinte forma:

$$\sum_{i=1}^r w_i = 1$$

O teorema a seguir fornece as condições suficientes para que uma solução obtida para o problema 3.12 seja Pareto-ótima (Chankong e Haimes 1983).

Teorema

Dado um vetor de pesos $w = (w_1, w_2, \dots, w_r)$, uma solução x^* do problema 3.12 é solução Pareto-ótima se ao menos uma das duas sentenças é válida:

- a. x^* é a solução única do problema 3.12;
- b. $w_i > 0, \forall i = 1, 2, \dots, r$.

A grande desvantagem desse método é que o decisor deve encarregar-se de atribuir pesos adequados aos objetivos. Para que os pesos w_i reflitam a importância relativa de cada objetivo, é preciso que as funções objetivo sejam normalizadas. Dessa forma, resolve-se vários problemas 3.12, cada um com um vetor w distinto (positivo).

Outra desvantagem reside na limitação do método de gerar todas as soluções Pareto-ótimas quando o espaço objetivo for não convexo, conforme ilustra a Figura 3.1. Considerando os pesos w_1 e w_2 para minimizar a seguinte função:

$$y = w_1 \cdot f_1(x) + w_2 \cdot f_2(x), \quad x \in X^*.$$

Reescrevendo-a:

$$f_2(x) = -\frac{w_1}{w_2} \cdot f_1(x) + \frac{y}{w_2}.$$

Esta última equação define uma reta L cuja inclinação é $-\frac{w_1}{w_2}$ e atravessa o eixo f_2 em $\frac{y}{w_2}$. Esta reta é tangente ao espaço objetivo factível Z^* em um ponto Pareto-ótimo. O método consiste em gerar diversas retas suporte através de valores apropriados para w_1 e w_2 . A limitação é que nem todos os pontos Pareto-ótimos admitem retas suporte. Pela Figura 3.1 verifica-se que os pontos C e D não possuem retas suporte, o que significa que eles não serão encontrados utilizando o problema 3.12.

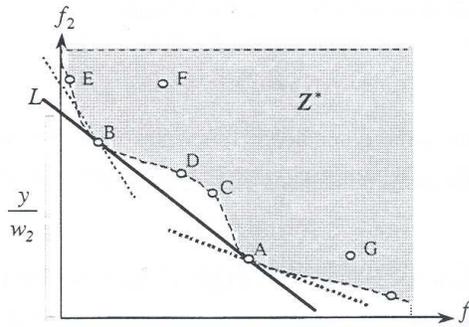


Figura 3.1: Interpretação gráfica do método da soma ponderada (Fonte: Arroyo (2002)).

3.3.2 Método do ε -restrito

Nesse método a idéia fundamental é transformar os objetivos em restrições, menos um que corresponde ao de maior prioridade. Este será o único considerado na função objetivo, conforme o seguinte problema:

$$\begin{aligned} & \text{Minimizar } f_1(x) \\ & \text{Sujeito a :} \\ & f_i(x) \leq \varepsilon_i \quad i = 2, \dots, r \\ & x \in X^* \end{aligned} \tag{3.13}$$

onde ε_i são limitantes superiores dos objetivos f_i , $i = 2, \dots, r$.

Com essa técnica é possível gerar o conjunto Pareto-ótimo mesmo quando o espaço objetivo não é convexo, variando-se convenientemente os limitantes ε_i . Quando o problema 3.13 apresenta funções objetivo e restrições lineares, tal problema pode ser considerado como de programação linear.

Um exemplo de aplicação deste método é mostrado na Figura 3.2, considerando-se um problema com dois objetivos. A reta $\varepsilon_2 = k$ limita o espaço de soluções factíveis, o qual contém os pontos A, B, C, D e G. Também é possível perceber uma das limitações do método ε -restrito: caso o limitante superior não seja adequadamente escolhido ($\varepsilon_2 = k'$), o espaço de busca resultante pode ser vazio, o que não permite nenhuma solução factível para o problema 3.13. Para evitar tal situação, deve-se, inicialmente, gerar um conjunto de valores apropriados para ε_i , opcionalmente com o método de Cohon (1978).

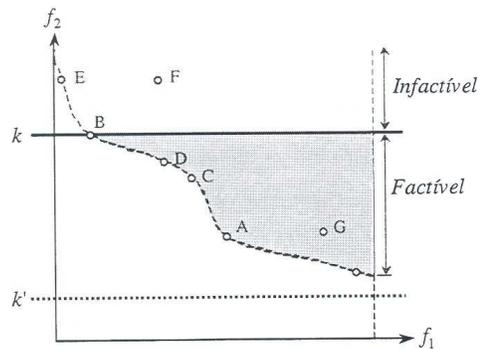


Figura 3.2: Interpretação gráfica do método ε -restrito (Fonte: Arroyo (2002)).

3.4 Metaheurísticas para POCM

O emprego de metaheurísticas para os POCM é justificado pela dificuldade de solução destes problemas. Dado que a tarefa fundamental dos POCM é fornecer um ou mais pontos não-dominados, o uso de metaheurísticas pode fornecer uma aproximação satisfatória ou até mesmo o próprio conjunto de soluções não-dominadas. O que se espera desse conjunto é que os pontos nele contidos sejam bem distribuídos pelo espaço objetivo, de forma a torná-los representativos, bem como que esses pontos estejam próximos ou mesmo que sejam os próprios pontos contidos na fronteira de Pareto. Dessa forma, todo método que obtenha uma aproximação da fronteira de Pareto com essas características será considerado um método de otimização multiobjetivo eficiente.

Com o avanço no desempenho de processamento dos computadores observado nos últimos anos foi possível considerar aplicações que incluem POCMs de grande complexidade. Mesmo considerando apenas as versões mono-objetivo destes problemas, a complexidade associada ainda é muito elevada, o que justifica o crescente emprego de metaheurísticas como alternativa para resolução dos mesmos.

Aproveitando o sucesso no emprego de metaheurísticas para resolução de problemas mono-objetivo (Osman e Laporte 1996), vários trabalhos vêm propondo o emprego destas

técnicas, com diversas abordagens, para resolução de POCMs. Caso o POCM esteja incluído em um sistema que envolva uma TDM, não basta ter uma aproximação ou o conjunto de soluções não-dominadas. É preciso obter apenas uma única solução que deve ser selecionada dentre todas as outras não-dominadas. Para tal finalidade é comum o emprego de uma metodologia como um *sistema de suporte à decisão* (Vidal 1987; Turban, Aronson e Liang 2004).

Nas seções a seguir serão apresentadas algumas metaheurísticas multiobjetivo capazes de gerar um conjunto aproximado de soluções Pareto-ótimas em uma única execução, objeto de estudo nesta tese.

3.4.1 *Simulated Annealing*

A versão multiobjetivo da metaheurística *Simulated Annealing* (SA), descrita na Seção 2.1.2 trata da probabilidade de aceitação de soluções vizinhas. Na versão mono-objetivo, toda solução que seja melhor que a atual é sempre aceita, enquanto que soluções piores são aceitas segundo uma probabilidade que varia de acordo com temperatura do sistema e com o grau de deterioração que essa solução promove. No contexto multiobjetivo, quando uma solução x^1 é comparada com uma solução x^2 com relação aos n objetivos $z^j(x)$, $j = 1, \dots, n$, onde Δz^j é a diferença entre a solução x^1 e x^2 no objetivo j , três situações podem ocorrer:

- Caso 1: $\forall j \Delta z^j \leq 0$
- Caso 2: $\exists j, j' \Delta z^j < 0$ e $\Delta z^{j'} > 0$
- Caso 3: $\forall j \Delta z^j \geq 0$

A idéia fundamental quando da aplicação de SA em otimização multiobjetivo consiste em utilizar uma norma ponderada para determinar a aceitação de soluções de baixa qualidade (casos 2 e 3).

Serafini (1994) sugere transições de probabilidade para utilizar SA no contexto multiobjetivo. Ele considera duas famílias de regras de aceitação: uma regra forte, na qual apenas soluções dominantes são aceitas com probabilidade igual a 1 (caso 1), e uma regra fraca, que aceita soluções dominadas com probabilidade menor que 1 (casos 2 e 3). Experimentos computacionais mostraram que a melhor escolha é a combinação das duas regras.

De forma independente, mas também utilizando transições de probabilidade, Ulungu, Teghem e Ost (1998) propõem o *Multiple Objective Simulated Annealing* (MOSA), que

usa uma função de aproximação ponderada $s(x, \lambda) = \sum_{j=1, \dots, n} \lambda_j z^j(x)$ com peso λ . Essa função é utilizada para calcular a diferença $\Delta s = s(x^2, \lambda) - s(x^1, \lambda)$ entre duas soluções e tem por objetivo aproximar todo o espaço de soluções não-dominadas. Procedese com o cálculo da diferença Δs , através da função de aproximação, para uma solução atual x^t e uma solução vizinha $y \in \mathcal{N}(x^t)$ gerada aleatoriamente na vizinhança $\mathcal{N}(x^t)$ de x^t , decidindo-se:

- a) Se $\Delta s < 0$ então $x^{t+1} \leftarrow y$.
- b) Se $\Delta s \geq 0$ então $x^{t+1} \leftarrow y$ com probabilidade p e $x^{t+1} \leftarrow x^t$ com probabilidade $1 - p$.

Czyzak e Jaszkiwicz (1998) apresentaram uma abordagem diferenciada das anteriores, o método *Pareto Simulated Annealing*. A manutenção dos pesos e a consideração de um conjunto de soluções atuais corresponde à inovação introduzida pelos autores. Cada solução é utilizada seguindo os mesmos critérios de aceitação de uma solução vizinha definidos por Ulungu et al. (1998), no entanto, os pesos são ajustados dinamicamente de tal forma que a solução aceita seja distinta das já encontradas. O mecanismo de dispersão incluído nessa determinação dos pesos permite uma aproximação uniforme e bem distribuída da fronteira de soluções não-dominadas.

3.4.2 Busca Tabu

Exatamente como na técnica anterior, os primeiros trabalhos relacionando Busca Tabu (BT) para otimização multiobjetivo faziam uso de estratégias mono-objetivo. A descrição da Busca Tabu mono-objetivo é apresentada na Seção 2.1.3.

É o caso da abordagem proposta por Gandibleux, Mezdaoui e Fréville (1997), que inclui princípios da BT para determinar uma boa aproximação do conjunto de soluções eficientes. A adaptação prevê o uso de um ponto utópico z^U como parâmetro de ponderação da função $s(x, \lambda)$. Considerando uma iteração t , uma solução atual x^t e sua vizinhança $\mathcal{N}(x^t)$, z^U é atualizado de acordo com os valores $z(x)$ para todo $x \in \mathcal{N}(x^t)$. A nova solução atual x^{t+1} será a melhor solução não-tabu de acordo com a direção atual imposta pelo parâmetro λ da função $s(x, \lambda)$. Uma memória tabu relacionada aos objetivos é considerada, armazenando a melhoria ocasionada em cada objetivo, servindo para atualizar a direção de busca de modo a cobrir toda a fronteira eficiente. Conceitos como intensificação, diversificação e critério de aspiração também são incluídos nesta técnica.

O método proposto por Hansen (1997) alguns meses depois de Gandibleux et al. (1997) inova por considerar não apenas uma, mas um conjunto de soluções incumbentes. Cada

uma dessas soluções possui sua própria lista tabu, seu próprio vetor de pesos e realiza movimentos de maneira independente das demais. A determinação dos pesos de cada solução é influenciada pelos outros vetores de pesos das outras soluções. Definida a direção de busca pelos pesos associados a cada objetivo, uma função de aproximação ponderada semelhante à abordagem anterior é utilizada para escolha da solução vizinha. Um critério *drift* substitui uma solução do conjunto atual pela cópia de outra, enfatizando a busca em certas regiões e, portanto, influenciando na definição dos pesos para essas soluções. Uma importante observação é que a cardinalidade do conjunto de soluções atuais também pode ser ajustada dinamicamente, conferindo, segundo os autores, uma grande versatilidade justamente pela associação com o critério *drift*.

Outra importante contribuição em BT multiobjetivo foi dada por Baykasoglu, Owen e Gindy (1999). A idéia de otimizar somente uma solução a cada iteração do método é resgatada nesse trabalho, que mantém três conjuntos de soluções: um conjunto de soluções proibidas, um conjunto de soluções dominantes (que armazena todas as soluções dominantes encontradas ao longo da busca) e um conjunto de soluções candidatas. A cada iteração é gerada a vizinhança da solução atual, da qual excluem-se as soluções proibidas e as soluções dominadas por alguma outra do conjunto de soluções dominantes e do conjunto de soluções candidatas. Desse conjunto de soluções restantes é escolhida a solução para a próxima iteração, procedendo-se, a seguir, com a atualização dos três conjuntos soluções armazenados. A diversificação é obtida com a escolha adequada de uma solução do conjunto de soluções candidatas para ser otimizada, direcionando a busca para regiões pouco exploradas da fronteira eficiente.

3.4.3 Algoritmos evolutivos

Algoritmos Evolutivos Multiobjetivo (AEM) têm sido amplamente utilizados desde o trabalho pioneiro de Rosenberg (1967). Um dos motivos que tornam estas técnicas muito adequadas para a aproximação do conjunto Pareto-ótimo é o emprego de uma população de soluções candidatas, permitindo que uma aproximação de todo o conjunto Pareto-ótimo seja obtida em uma única execução. Os trabalhos de Coello Coello e Christiansen (2000) e Zitzler, Laumanns e Bleuler (2004) apresentam uma ótima revisão do estado da arte em AEM.

Há várias formas de contemplar o aspecto multiobjetivo em algoritmos evolutivos: através de funções de agregação, que combinam as várias funções objetivo em uma única; técnicas desvinculadas da dominância de Pareto; técnicas baseadas no conceito de dominância de Pareto, onde as soluções são classificadas de acordo com a relação de do-

minância que umas exercem sobre as outras; e técnicas híbridas.

A seguir são apresentadas cada uma destas classes com as abordagens mais representativas de cada uma delas.

3.4.3.1 Algoritmos evolutivos baseados em funções de agregação

Nesta classe de algoritmos evolutivos multiobjetivo, a noção mais intuitiva para abordar um problema multiobjetivo, através da combinação dos objetivos em um único escalar, deriva da observação de que um algoritmo evolutivo precisa de um escalar para denotar a adaptação de um indivíduo.

Um dos maiores problemas associados refere-se ao conhecimento de faixa de cada um dos objetivos, de forma a não permitir que um domine os outros. Portanto, é necessário um conhecimento prévio, muitas vezes indisponível, acerca do comportamento de cada uma das funções objetivo do problema.

Um método muito amplamente conhecido é o da *Soma Ponderada* descrito na Seção 3.3.1.

A resolução do problema 3.12 fornecerá resultados que estarão associados aos pesos escolhidos. Como, de forma geral, não se dispõe de um conhecimento prévio do decisor sobre a importância de cada objetivo, é necessário resolver vários vezes o problema 3.12 com diferentes pesos de modo a gerar uma aproximação do conjunto ótimo de Pareto.

Hajela e Lin (1992) sugeriram uma abordagem que faz uso de uma função de agregação para resolver um problema de projeto ótimo multiobjetivo, conhecida como *Hajela's Lin's Genetic Algorithm* (HLGA). A grande inovação foi a incorporação dos pesos na representação da solução, com a adoção de mecanismos especiais para preservação da diversidade de combinações destes pesos. A consequência é que o algoritmo evolui soluções e combinações de pesos simultaneamente. Por fim os autores ressaltam a importância do emprego de restrições para recombinação dos indivíduos de forma a controlar a velocidade de convergência e a estabilidade da busca.

Outra abordagem para algoritmos evolutivos multiobjetivo é conhecido como método ϵ -restrito, descrito na Seção 3.3.2.

Para obter valores adequados para ϵ_i , otimizações mono-objetivo independentes para cada função objetivo são realizadas através do uso de técnicas de programação matemática ou algoritmos genéticos independentes. Para cada função objetivo f_i ($i = 1, 2, \dots, k$) há um valor ótimo x_i^* para o qual $f_i(x_i^*)$ é um mínimo. Seja $f_i(x_i^*)$ o limite inferior de ϵ_i , ou seja,

$$\epsilon_i \geq f_i(x_i^*) \quad i = 1, 2, \dots, r-1, r+1, \dots, k \quad (3.14)$$

e $f_i(x_r^*)$ o limite superior de ϵ_i , ou seja,

$$\epsilon_i \leq f_i(x_i^*) \quad i = 1, 2, \dots, r-1, r+1, \dots, k \quad (3.15)$$

Quando os limites de ϵ_i são muito “apertados” não há solução possível e pelo menos um deles deve ser relaxado.

Esta abordagem foi proposta por Ritzel, Eheart e Ranjithan (1994) como uma forma simples de resolver problemas de otimização multiobjetivo com algoritmos genéticos. Através da codificação de um algoritmo genético, onde todos os objetivos são mantidos constantes exceto um que corresponderá à função de adaptação, uma aproximação do conjunto Pareto-ótimo será obtida quando este algoritmo for executado diversas vezes, cada uma com valores distintos para as funções objetivo mantidas constantes.

3.4.3.2 Algoritmos evolutivos desvinculados da dominância de Pareto

Esta classe de algoritmos evolutivos foi desenvolvida na tentativa de superar algumas restrições decorrentes do uso da classe anterior, especialmente pela dificuldade de tratar funções objetivo não convexas.

Schaffer (1985) propôs um algoritmo genético, que ele chamou de *Vector Evaluated Genetic Algorithm* (VEGA), para otimização multiobjetivo. A principal diferença em relação ao algoritmo genético convencional é o emprego de um operador especial de seleção: ele foi modificado de forma que a cada geração são criadas diversas subpopulações, uma para cada objetivo. Dessa forma, um problema com k objetivos, k subpopulações de tamanho N/k , sendo N o tamanho máximo da população, são geradas da seguinte forma: considerando o objetivo i , monta-se uma roleta ponderada segundo a avaliação de todos os indivíduos da população para este objetivo i e geram-se aleatoriamente N/k indivíduos, repetindo-se o processo para os demais objetivos. A seguir, estes indivíduos, incluídos nas subpopulações, são misturados para dar origem à população sob a qual serão aplicados os operadores genéticos de recombinação e mutação.

Um problema observado por Schaffer foi a criação de novas espécies durante o processo evolutivo, processo também conhecido como *speciation*. A consequência imediata é a geração de indivíduos que se destacam em relação aos demais indivíduos da população em determinados aspectos, ou seja, são mais aptos somente quando determinada função objetivo é considerada como função de adaptação. Pela forma “míope” com que a seleção observa a adaptação de cada indivíduo, de forma independente para cada função objetivo,

o processo evolutivo poderá conduzir a uma população com indivíduos que não apresentem bom desempenho quando todos objetivos do problema são considerados. Uma das heurísticas sugeridas por Schaffer para evitar tal comportamento é o emprego de um mecanismo de seleção que privilegie indivíduos não-dominados.

Outra abordagem que não é baseada na dominância de Pareto é o *Ordenamento Lexicográfico*, onde os objetivos são classificados em ordem de importância segundo a preferência do decisor. A solução ótima x^* é obtida através da minimização das funções objetivo, uma de cada vez, começando pela mais importante.

Tomando k objetivos definidos pelas funções $f_1(x), \dots, f_k(x)$, sendo a função $f_1(x)$ a mais importante e $f_k(x)$ a menos importante, segundo as preferências do decisor, o primeiro problema de otimização pode ser definido pelas equações 3.16 e 3.17.

$$\text{Minimizar } f_1(x) \quad (3.16)$$

Sujeito a:

$$g_j(x) \leq 0; \quad j = 1, 2, \dots, m \quad (3.17)$$

Deste problema é obtida uma solução ótima x_1^* . Passa-se então para o próximo problema (3.18 - 3.20):

$$\text{Minimizar } f_2(x) \quad (3.18)$$

Sujeito a:

$$g_j(x) \leq 0; \quad j = 1, 2, \dots, m \quad (3.19)$$

$$f_1(x) = f_1^* \quad (3.20)$$

A solução x_2^* obtida deste problema é utilizada para calcular $f_2^* = f_2(x_2^*)$. Este procedimento é repetido até que todos os k objetivos sejam considerados.

A solução obtida depois da resolução dos k problemas, x_k^* , é tomada como a solução para o problema multiobjetivo original.

O maior problema desta abordagem é a determinação da ordem dos objetivos. Fourman (1985) sugere que a ordem de preferência pode ser determinada de forma aleatória, sendo que a consequência negativa disso é a obtenção de apenas uma parte da fronteira de Pareto devido ao favorecimento de alguns objetivos decorrente da aleatoriedade empregada. A grande vantagem desta técnica é a simplicidade, tornando-a competitiva com as técnicas baseadas em função de agregação e com o próprio VEGA.

3.4.3.3 Algoritmos evolutivos baseados na dominância de Pareto

A motivação para o uso de função de adaptação associada à relação de dominância de Pareto foi resolver os problemas de Schaffer (1985), descritos por Goldberg (1989): inabilidade em gerar soluções Pareto-ótimas na presença de um espaços de busca não convexos. O conceito fundamental da proposta de Goldberg (1989) é o uso de classificação e seleção baseada na relação de dominância, de forma a mover a população em direção à fronteira ideal de Pareto. Primeiramente obtém-se os indivíduos não-dominados da população inteira. A eles é atribuído a maior classificação e os mesmos são retirados da população. Dos indivíduos remanescentes é obtido o próximo conjunto não-dominado e lhe é atribuído o próximo valor na classificação, necessariamente inferior ao valor atribuído ao conjunto não-dominado anterior. Repete-se este processo até que toda a população esteja devidamente classificada. Goldberg sugere ainda que seja adotado algum mecanismo de nicho, de modo a distribuir os indivíduos por toda a fronteira ideal de Pareto.

A abordagem proposta por Fonseca e Fleming (1993), conhecida como *Multiple Objective Genetic Algorithm* (MOGA), inclui um esquema no qual a classificação de cada indivíduo é dada pelo número de indivíduos que o dominam. Por exemplo, a classificação de um indivíduo x_i na geração t , que é dominado por $p_i^{(t)}$ indivíduos nesta geração, é dada pela equação 3.21.

$$rank(x_i, t) = 1 + p_i^{(t)} \quad (3.21)$$

Com isso, todos os indivíduos não-dominados são classificados com valor igual a 1, enquanto aqueles dominados têm as suas classificações penalizadas pelos indivíduos que os dominam. O procedimento proposto para atribuição da adaptação de cada indivíduo é o seguinte:

1. Classificar a população como descrito anteriormente;
2. Atribuir um valor de função de adaptação aos indivíduos de acordo com a interpolação a partir do melhor até o pior valor de classificação, tal como proposto por Goldberg (1989), de acordo com alguma função pré-estabelecida, usualmente uma função linear;
3. Tomar a média dos valores de função de adaptação para indivíduos com a mesma classificação e atribuir-lhes este valor médio como novo valor de função de adaptação. Com isso mantém-se uma pressão seletiva apropriada segundo a função utilizada na interpolação.

Para contornar o problema de convergência prematura, no mesmo trabalho, Fonseca e Fleming propõem o uso de métodos para formação de nichos através do compartilhamento dos valores de função objetivo. Embora esta tentativa tenha obtido relativo sucesso, ela também representa um dos grandes problemas desta abordagem: a escolha do fator de compartilhamento, que é um dos parâmetros críticos. Pela sua simplicidade e eficiência, o MOGA é uma técnica muito empregada para resolução de problemas de otimização multiobjetivo.

Outra técnica que também utiliza as idéias de Goldberg é o *Non-Dominated Sorting Genetic Algorithm* (NSGA), proposto por Srinivas e Deb (1994). Ela é baseada na classificação em várias camadas dos indivíduos da população: todos aqueles indivíduos não-dominados são classificados em uma única categoria, com avaliação de cada um proporcional ao tamanho da população, com o intuito de promover a igualdade dos potenciais reprodutivos destes indivíduos. A seguir, este grupo é ignorado e uma nova camada de indivíduos não-dominados, oriundos da população remanescente, é considerado. A seleção para reprodução é realizada aleatoriamente, de forma proporcional ao valor de função de adaptação de cada indivíduo.

Como a chance daqueles indivíduos com maior valor de função de adaptação, pertencentes à primeira camada, de participar do processo de reprodução é maior, proporciona-se uma busca por soluções não-dominadas, o que pode ocasionar uma convergência rápida. A avaliação proporcional ajuda a distribuir os indivíduos pelo espaço objetivo.

A principal vantagem do NSGA é a possibilidade de resolver problemas com qualquer número de objetivos, além da divisão proporcional da função de adaptação, permitindo uma melhor distribuição dos indivíduos sobre o espaço objetivo. Nos estudos comparativos realizados por Coello Coello (1996) foi evidenciado que o desempenho do NSGA seria inferior ao MOGA, e também bastante sensível à divisão da função de adaptação.

Alguns anos mais tarde, Deb, Agrawal, Pratap e Meyarivan (2000) desenvolveram uma nova versão do *Non-Dominated Sorting Genetic Algorithm* chamada NSGA-II. A grande inovação foi a incorporação de elitismo no processo evolutivo, além de uma redução substancial na complexidade associada ao processo de seleção, considerado um dos grandes pontos fracos do NSGA. Adicionalmente, a densidade populacional também foi considerada de modo a distribuir melhor os indivíduos sobre a fronteira aproximada. Nos estudos computacionais, os autores comparam o NSGA-II com o PAES (descrito a seguir) na minimização de cinco funções não-lineares, evidenciando a superioridade da abordagem proposta sobre esta outra.

Horn, Nafpliotis e Goldberg (1994) propuseram um novo tipo de algoritmo genético multiobjetivo chamado *Niched Pareto Genetic Algorithm* (NPGA). Seu funcionamento

é baseado num mecanismo de seleção baseada em torneio, utilizando a relação de dominância como critério de escolha. A diferença no torneio empregado é que ele não se restringe a dois indivíduos. Um número determinado de indivíduos (usualmente 10) da população é utilizado para determinar a dominância. Quando ambos os competidores são dominados ou não-dominados, o resultado do torneio é decidido pela divisão da função de adaptação: o indivíduo que estiver no menor nicho (com o menor número de indivíduos) é tomado como vencedor do torneio.

A grande vantagem deste tipo de algoritmo genético é a rapidez: a seleção restrita a uma pequena parte da população ajuda a reduzir o tempo gasto. Um dos problemas associados ao NPGA refere-se à escolha do tamanho do torneio: o desempenho do algoritmo tem muita relação com esta escolha.

Também com o propósito de desenvolver uma abordagem simples e eficiente, Knowles e Corne (1999) desenvolveram o *Pareto Archived Evolution Strategy* (PAES). A idéia fundamental é manter um conjunto com as soluções não-dominadas e extrair, a cada iteração, uma solução deste conjunto para realizar pequenas perturbações. A solução vizinha gerada será candidata a participar do conjunto de soluções não-dominadas se ela dominar a solução incumbente. No entanto, esta inclusão somente será realizada caso a solução gerada não for dominada por nenhuma solução do conjunto. Outros dois critérios também são considerados durante o processo de aceitação: a diversidade da população e a distribuição da população sobre a fronteira aproximada.

Na abordagem proposta por Zitzler e Thiele (1999), o *Strength Pareto Evolutionary Algorithm* (SPEA), várias características muito desejáveis foram incorporadas, tais como:

- Armazenamento independente das soluções não-dominadas encontradas pela busca;
- Uso da relação de dominância de Pareto para designar a adaptação de cada indivíduo da população;
- Emprego de um método de *clustering* para reduzir o número de soluções não-dominadas armazenadas no conjunto externo, sem perder as características da fronteira aproximada.

Nos experimentos computacionais para resolver o problema da mochila 0/1, Zitzler e Thiele (1999) demonstraram que o SPEA é muito competitivo frente às outras quatro abordagens: o HLGA, o NPGA, o VEGA e o NSGA. No entanto, os mesmos autores perceberam pontos fracos desta abordagem e propuseram alterações substanciais em Zitzler, Laumanns e Thiele (2002), dando origem ao SPEA2. As principais diferenças são:

- *Função de adaptação*: a avaliação de um indivíduo é dada pelo número de indivíduos que são dominados por ele e também pelo número de indivíduos que o dominam, evitando valores idênticos de funções de adaptação;
- *Densidade populacional*: além da dominância, uma informação sobre a concentração dos indivíduos da população inteira tem um papel fundamental na condução da busca por regiões pouco exploradas;
- *Redução do conjunto de soluções não-dominadas*: embora o método de *clustering* seja relativamente eficiente em relação à manutenção das características, soluções localizadas nos extremos de cada grupo podem ser perdidas, comprometendo a distribuição das soluções não-dominadas sobre a fronteira aproximada. Dessa forma, um método alternativo muito similar ao anterior é proposto para evitar a perda destas soluções fronteiriças.

Novamente, os experimentos computacionais, com a minimização de diversas funções não-lineares, demonstram o bom desempenho da abordagem SPEA2 quando comparada a outras técnicas como PESA (descrita a seguir) e NSGA-II. Um ponto negativo do PESA observado por Zitzler et al. foi a rápida convergência, prejudicando uma boa distribuição das soluções sobre a fronteira aproximada. SPEA2 e NSGA-II conseguem manter uma diversidade maior que os favorece na melhor distribuição dos indivíduos.

Visando combinar seleção e preservação de diversidade, Corne, Knowles e Oates (2000) desenvolveram o *Pareto Envelope-based Selection Algorithm* (PESA). Este algoritmo mantém duas populações: uma interna, que mantém todos os indivíduos da população, e outra externa, dedicada ao armazenamento daqueles indivíduos não-dominados. Por meio de uma estrutura em grade usada para avaliar a concentração da população interna, uma seleção por torneio simples é realizada para obter dois indivíduos que darão origem a um terceiro. Os vencedores destes dois torneios serão aqueles indivíduos que estiverem na região com menor densidade populacional. Um terceiro indivíduo será obtido destes dois pais a partir de recombinação, sofrendo a seguir um processo de mutação. Na fase de aceitação deste novo indivíduo, a concentração da população interna também é observada, e a participação na população externa estará sujeita somente à relação de dominância de Pareto. Nos experimentos computacionais, Corne et al. (2000) empregaram a minimização de seis funções não-lineares e compararam o PESA com o PAES e o SPEA. Ainda que com uma pequena vantagem, o desempenho do PESA foi superior aos outros dois, e o SPEA, por sua vez, foi superior ao PAES.

3.4.3.4 Algoritmos evolutivos híbridos

Como não existe um único algoritmo capaz de resolver todos os problemas da melhor forma (Michalewicz e Fogel 2000), torna-se necessário incorporar conhecimento acerca do problema no algoritmo a ser desenvolvido de tal forma que ele tenha um desempenho melhor que uma simples busca aleatória. Tal agregação pode ser realizada por meio da inclusão de métodos de busca local ou outras técnicas convencionais em algoritmos evolutivos, caracterizando uma nova classe chamada de *algoritmos evolutivos híbridos*.

Um dos primeiros trabalhos que combinam busca local com algoritmos evolutivos para resolver problemas de otimização multiobjetivo foi apresentado por Ishibuchi e Murata (1998). O cálculo da função de adaptação para cada indivíduo da população é realizado através do uso de uma função de agregação. Quando dois pais são selecionados da população para dar origem a um filho, um conjunto de pesos é gerado aleatoriamente e utilizado em ambos os pais e posteriormente no filho, durante a etapa de busca local. Para reduzir o tempo gasto nesta fase, Ishibuchi e Murata limitam o número de soluções vizinhas avaliadas. Os experimentos computacionais reportam a comparação do algoritmo proposto com o VEGA, para resolução de uma instância do problema de *flowshop* multiobjetivo com 10 tarefas e 5 máquinas e outra com 20 tarefas e 10 máquinas. Para a primeira são considerados dois objetivos e na segunda três. A superioridade da abordagem proposta é observada pelos autores frente ao algoritmo proposto por Schaffer (1985).

Outra abordagem que combina algoritmos genéticos com busca local foi apresentada por Jaszkiwicz (2002). Nela, a otimização não é baseada na relação de dominância de Pareto e sim na minimização da função utilidade de Tchebycheff, descrita na seção 3.1.1, equação 3.10. O autor mostra que, como a solução ótima de uma função utilidade de Tchebycheff corresponde a uma solução pertencente ao conjunto Pareto-ótimo, para encontrar este conjunto basta encontrar as soluções ótimas de todas as funções utilidade de Tchebycheff. Com isso, a cada iteração uma função utilidade é gerada para servir como função a ser otimizada pelo algoritmo genético e pela busca local. Na fase inicial, onde é gerado um número fixo de soluções iniciais, a cada iteração é gerada uma função utilidade e construída uma nova solução aleatoriamente, a qual é otimizada localmente segundo esta função. Todas estas soluções iniciais são armazenadas em um conjunto *CS* que armazena um número limitado de todas as soluções geradas pela busca. Na fase seguinte também é gerada uma função utilidade a cada iteração, no entanto são extraídas as *K* melhores soluções do conjunto *CS* para compor o conjunto *TP*. Deste, são obtidas aleatoriamente as duas soluções que darão origem a uma terceira através de um processo de recombinação. Esta solução gerada somente será incluída em *CS* caso ela seja melhor

que a pior solução do conjunto TP , considerando a função de utilidade gerada. Por fim, é atualizado um terceiro conjunto PE que inclui todas as soluções não-dominadas geradas pela busca.

Jaszkiewicz observou que, mesmo depois de um grande número de iterações, a recombinação e a busca local ainda promovem soluções não-dominadas. Segundo ele, isto se deve à diversidade proporcionada pela seleção aleatória da função utilidade.

Recentemente, Ishibuchi e Narukawa (2004) realizaram um estudo comparativo de dois algoritmos evolutivos híbridos que eles desenvolveram: o primeiro baseado em função de agregação e o segundo baseado na relação de dominância de Pareto. Dois problemas teste foram utilizados: o problema da mochila e o problema de *flowshop*, ambos com dois objetivos e apenas uma instância de cada um. Ishibuchi e Narukawa concluíram que a abordagem que emprega funções de agregação teve um desempenho superior à outra que utiliza a relação de dominância de Pareto, especialmente quanto à diversidade das soluções não-dominadas geradas. Este algoritmo com melhor desempenho também foi comparado com o NSGA-II, superando este último no problema da mochila com respeito à diversidade das soluções obtidas. Para o problema de *flowshop* o resultado foi inverso: o algoritmo NSGA-II proporcionou melhores soluções inclusive quanto à diversidade obtida. Os autores concluem deste último estudo que o mecanismo de seleção do NSGA-II apresentou-se muito adequado para esta instância do problema de *flowshop*.

3.5 Métodos de avaliação de heurísticas multiobjetivo

As heurísticas ou metaheurísticas mono-objetivo são geralmente avaliadas por dois critérios: a qualidade da solução obtida e o esforço computacional gasto para tanto. Para o caso multiobjetivo, avaliar o esforço computacional é similar ao anterior, por isso utilizar-se-á apenas um critério de parada correspondente ao número de soluções avaliadas. O outro critério, entretanto, a qualidade da solução, é a grande diferença: como no caso mono-objetivo a saída é uma única solução, a comparação é facilmente realizada através do valor correspondente à função objetivo; como a versão multiobjetivo trata não apenas de uma, mas de um conjunto de soluções num espaço objetivo multidimensional, torna-se mais complexa a avaliação da qualidade desse conjunto.

Infelizmente, não há um consenso quanto à medida de comparação mais apropriada. No entanto, além da avaliação gráfica, restrita apenas a casos bi ou tri-objetivo, é possível observar o uso de medidas quantitativas e qualitativas na literatura especializada recente

(Jaszkiewicz 2004). Dois critérios embasam fortemente a grande maioria delas, quando observa-se o conjunto de soluções gerado: (1) a distância do conjunto aproximação gerado em relação ao conjunto de soluções Pareto-ótimas; (2) a distribuição das soluções do conjunto sobre o espaço objetivo.

Nas seções subseqüentes serão apresentadas primeiramente algumas definições, para a posterior apresentação de algumas medidas comumente empregadas. Por fim é descrita a medida escolhida para uso nesta tese, juntamente com um exemplo ilustrativo.

3.5.1 Métodos que consideram o conjunto Pareto-ótimo

A noção mais intuitiva para avaliação de metaheurísticas multiobjetivo refere-se à qualidade com que o conjunto A de pontos gerados aproxima-se do conjunto não-dominado ND .

Considerando que a melhor aproximação é o próprio conjunto ND , uma das medidas mais imediatas, decorrente desta idéia, é a razão entre pontos encontrados pertencentes à ND . A medida pode ser definida da seguinte forma, utilizando-se o próprio ND ou uma boa aproximação, onde $|A \cap ND|$ e $|ND|$ indicam as cardinalidades dos conjuntos $|A \cap ND|$ e $|ND|$, respectivamente:

$$Q_1(A) = \frac{|A \cap ND|}{|ND|} \quad (3.22)$$

Um aspecto bastante desencorajador quando do uso desta medida é o fato de que uma boa aproximação pode conter pontos não-dominados que não pertençam a ND . Além disso, muitas metaheurísticas multiobjetivo podem ser incapazes de fornecer um único ponto não-dominado pertencente a ND , o que não permite a diferenciação entre a qualidade de duas aproximações que recaiam nesta condição, ainda que uma delas seja superior à outra. Na Figura 3.3 ilustra-se esta limitação da medida Q_1 .

Há também medidas que consideram a “distância” entre o conjunto aproximado obtido pelo algoritmo e o conjunto não-dominado, ou, na ausência deste, sua aproximação (chamado de conjunto de referência RS). A medida proposta por Czyak e Jaszkiewicz (1998) é deste tipo, e pode ser definida da seguinte forma:

$$Q_2(A) = \frac{1}{|RS|} \sum_{r \in RS} \min_{z \in A} \{d(z, r)\}, \quad (3.23)$$

onde $d(.,.)$ corresponde à distância no espaço objetivo (podem ser usadas a distância euclideana ou a métrica ponderada de Tchebycheff, por exemplo). Ela corresponde à distância média de cada ponto de referência até o seu vizinho mais próximo em A . O

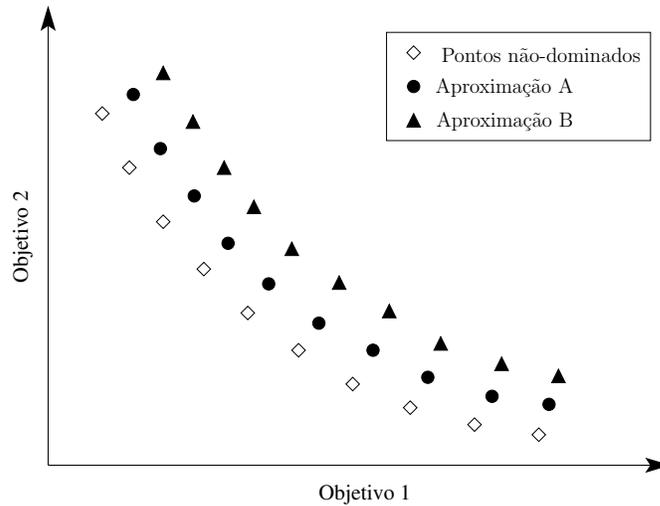


Figura 3.3: A aproximação A é visivelmente melhor que B, embora a medida Q_1 indique que ambas são equivalentes.

grande problema desta medida é o “favorecimento” de regiões do conjunto de referência com muitos pontos agrupados, as quais influenciarão a medida mais do que outras com um número menor de pontos. Dessa forma, uma aproximação com muitos pontos concentrados pode obter valores maiores de avaliação quando comparada a outra com os pontos mais distribuídos no conjunto de referência. Cabe salientar que a concentração de pontos em determinada região do conjunto de referência não significa maior importância para o decisor.

3.5.2 Medidas baseadas na preferência do decisor

Supondo que as preferências do decisor são conhecidas, é possível modelá-las, por exemplo, com uma função utilidade u . Considerando a comparação de duas aproximações A e B do conjunto de soluções não-dominadas, do ponto de vista do decisor seria intuitivo escolher a aproximação que fornecesse o menor valor da função u . Assim, a escolha torna-se imediata a partir do conhecimento das preferências do decisor.

Caso estas preferências não sejam conhecidas, torna-se necessário fazer algumas suposições para basear a escolha. Uma delas apóia-se no fato de que o decisor preferirá soluções não-dominadas, o que permite classificar todas as medidas que usam somente a relação de dominância como *baseadas na preferência* do decisor.

Duas novas relações entre aproximações do conjunto de soluções não-dominadas foram

propostas por Hansen e Jaszkievicz (1998), ambas baseadas nas relações de dominância.

Definição 1

Uma aproximação A supera fracamente outra aproximação B , $A O_W B$, se $A \neq B$ e se $ND(A \cup B) = A$, onde $ND(A \cup B)$ representa o conjunto de pontos não-dominados em $A \cup B$. De outra forma, $A O_W B$ se existir um ponto $z^1 \in A$ que seja igual ou domine qualquer ponto $z^2 \in B$ e se ao menos um ponto $z^1 \in A$ não esteja contido em B .

Definição 2

Uma aproximação A supera completamente B , $A O_C B$, se $ND(A \cup B) = A$ e $B \cap ND(A \cup B) = \emptyset$. De outra forma, $A O_C B$ se todo ponto de B for dominado por um ponto $z \in A$.

Caso $A O_W B$, toda solução de A não é “pior” que qualquer outra solução de B . Caso $A O_C B$, toda solução de A é “melhor” que qualquer outra solução de B . Considerando estes critérios, nas três aproximações apresentadas na Figura 3.4, percebe-se que $B O_W C$ e $A O_C C$.

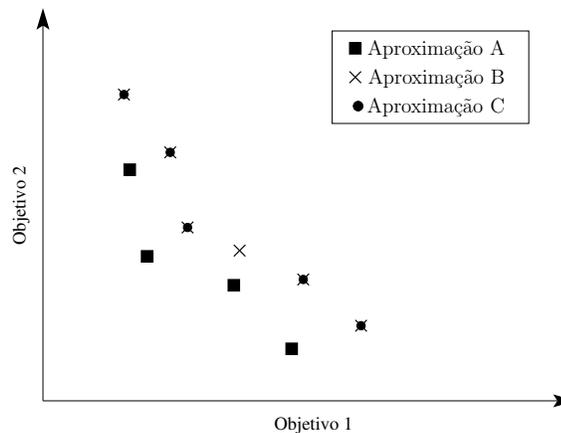


Figura 3.4: Exemplos das relações descritas: $B O_W C$ e $A O_C C$.

Zitzler e Thiele (1999) propuseram uma medida chamada de *cobertura de duas aproximações* do conjunto de soluções não-dominadas. Com a definição a seguir, esta medida permite uma avaliação relativa de duas aproximações:

$$Q_3(A, B) = \frac{|\{z^2 \in B\} | \exists z^1 \in A : z^1 \succeq z^2 |}{|B|} \quad (3.24)$$

Quando a medida tem valor $Q_3(A, B) = 1$ significa que todos os pontos da aproximação

B são dominados ou iguais aos pontos contidos na aproximação A . O valor $Q_3(A, B) = 0$ significa que não existe nenhum ponto em A que cubra algum ponto em B . Dessa forma, $A O_W B \Leftrightarrow Q_3(A, B) = 1$ e $Q_3(B, A) < 1$. Além disso, $A O_C B \Leftrightarrow Q_3(A, B) = 1$ e $Q_3(B, A) = 0$. Dessa forma, a medida Q_3 pode ser entendida como a “intensidade” da relação da definição 1, o que significa dizer que, quanto mais próximo de zero for $Q_3(B, A)$, mais próximo da relação da definição 2 estarão as duas aproximações.

Uma desvantagem desta medida refere-se ao favorecimento de conjuntos que apresentem um pequeno número de pontos muito próximos no espaço objetivo. Na Figura 3.5 a aproximação A é superior à aproximação B segundo a medida Q_3 , no entanto, caso o decisor procure regiões não representadas na aproximação A , ele preferirá a aproximação B .

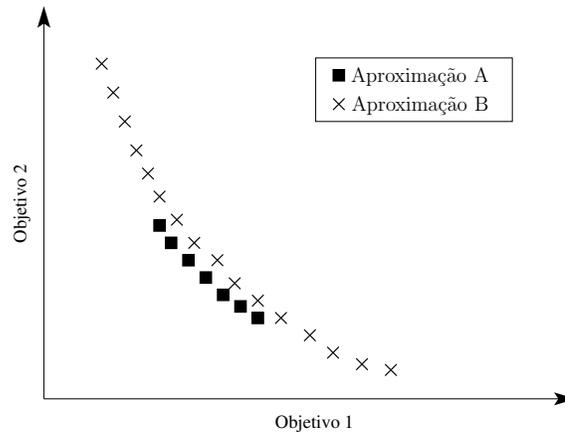


Figura 3.5: A medida Q_3 favorece a aproximação A em relação à B .

Dado que a avaliação baseada somente na relação de dominância pode apresentar certas anomalias indesejadas, Hansen e Jaszkiwicz (1998) propuseram algumas medidas que associam as preferências do decisor a um conjunto de funções utilidade U . Seja $U(A < B) \subseteq U$ o subconjunto de funções utilidade para o qual a aproximação A é melhor que a aproximação B , ou seja, $U(A < B) = \{u \in U | u^*(A) < u^*(B)\}$, onde $u^*(A)$ corresponde ao melhor valor da função utilidade u considerando a aproximação A : $u^*(A) = \min_{z \in A} \{u(z)\}$.

Definição 3

Uma aproximação A supera outra aproximação B sujeito a um conjunto de funções utilidade, ou seja, $A O_{/U} B$, se $U(A < B) \neq \emptyset$ e $U(B < A) = \emptyset$, se existir alguma função utilidade no conjunto U que apresente melhores valores para A em relação a B , e

o contrário não deve ser verdadeiro.

Em alguns casos pode ser impraticável aplicar a relação acima para determinadas aproximações. Na Figura 3.6, a aproximação A é melhor que a aproximação B para um determinado conjunto de funções utilidade e vice-versa. Diante disso, torna-se necessário avaliar a probabilidade com que uma aproximação é melhor que outra. Isto é feito assumindo-se $p(u)$ como a função densidade de probabilidade associada à função utilidade do decisor.

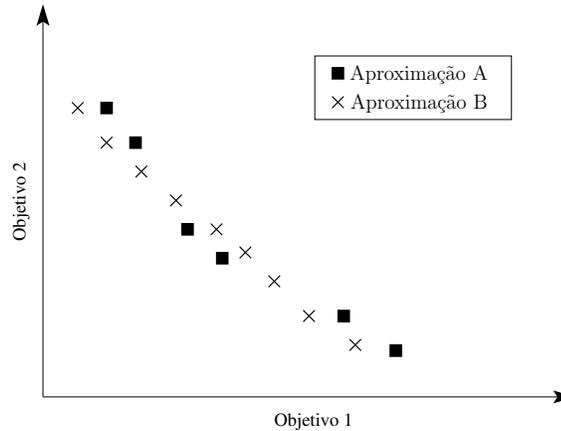


Figura 3.6: Aproximações incomparáveis com a utilização da **Definição 3**.

Na equação 3.25 é apresentada a medida Q_4 , que fornece um escalar como resultado do valor esperado da função utilidade $u^*(A)$ no conjunto U . Uma aproximação A é melhor que uma aproximação B , segundo a medida, se $Q_4(A, U, p) < Q_4(B, U, p)$

$$Q_4(A, U, p) = \int_{u \in U} u^*(A) p(u) du = E(u^*(A)) \quad (3.25)$$

Hansen e Jaszkievicz (1998) propuseram a consideração de funções utilidade ponderadas de Tchebycheff uniformemente distribuídas, com vetores de peso normalizados e um ponto de referência equivalente a uma aproximação do ponto ideal. Como a definição de função utilidade ponderada de Tchebycheff é equivalente a função Tchebycheff ponderada, a medida Q_5 (equação 3.26) pode ser interpretada como o valor médio destas funções.

$$Q_5(A, S_\infty, p) = \int_{s_\infty \in S_\infty} s_\infty^*(A) p(s_\infty) ds_\infty = E(s_\infty^*(A)), \quad (3.26)$$

onde S_∞ é o conjunto de funções Tchebycheff ponderadas com vetores de pesos normalizados e $s_\infty^*(A) = \min_{z \in A} \{s_\infty(z)\}$.

Como não há método geral para o cálculo da medida Q_5 , conforme observam Hansen e Jaszkiewicz (1998), eles sugerem a adoção de um valor médio das funções de utilidade como valor aproximado de Q_5 . As funções são definidas por vetores de pesos normalizados e uniformemente distribuídos em que cada peso individual assume um valor do conjunto $\{l/k, l = 0, \dots, k\}$, onde k é o tamanho da amostra. O conjunto destes vetores é denotado por Λ_R e definido na equação 3.27.

$$\Lambda_R = \left\{ \lambda = [\lambda_1, \dots, \lambda_J] \in \Lambda \mid \lambda_j \in \left\{ 0, \frac{1}{k}, \frac{2}{k}, \dots, \frac{k-1}{k}, 1 \right\} \right\}, \quad (3.27)$$

onde Λ é o conjunto de todos os vetores de pesos normalizados. Com isso, a medida Q_6 pode ser definida pela equação 3.28 e corresponde a uma aproximação da medida Q_5 .

$$Q_6 = \frac{\sum_{\lambda \in \Lambda_R} s_{\infty}^*(A)}{|\Lambda_R|} \quad (3.28)$$

3.5.3 Escolha do método para avaliação das heurísticas multi-objetivo

Como já foi salientado neste capítulo, dois aspectos tornam-se fundamentais quando da comparação de heurísticas multiobjetivo: a distribuição dos pontos sobre o espaço objetivo e a proximidade destes pontos da fronteira de Pareto. Diante disso optou-se pela medida Q_6 pela vantagem de combinar a avaliação da distribuição e da proximidade da fronteira ideal num único escalar.

Como exemplo do uso desta medida e também como justificativa pela escolha, apresenta-se o cálculo da eficiência de duas aproximações ilustradas na Figura 3.7. Para o cálculo da medida Q_6 para as aproximações A e B é necessário dispor de um ponto ideal, $(0, 0)$ neste caso, e de um parâmetro K associado ao tamanho da amostra dos vetores de peso.

Com $K = 2$ e um número de objetivos $J = 2$ tem-se:

$$\begin{aligned} \Lambda &= \{[\lambda_1, \lambda_2] \mid \lambda_j \in \{0, 1/2, 1\} \wedge \sum_{k=1}^J \lambda_k = 1\} \\ \Lambda &= \{\lambda_1, \lambda_2, \lambda_3\}, \quad \lambda_1 = [0, 1], \quad \lambda_2 = [1, 0], \quad \lambda_3 = [1/2, 1/2] \end{aligned} \quad (3.29)$$

Dessa forma, $Q_6(A)$ é dado pela equação 3.30.

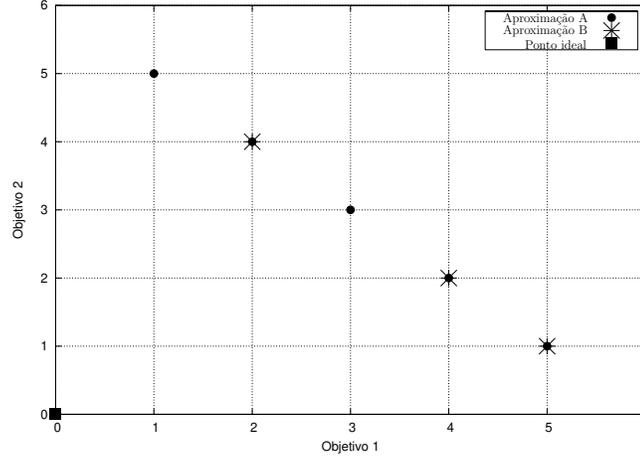


Figura 3.7: Aproximações comparadas pela medida Q_6 .

$$\begin{aligned}
 A &= \{(5, 1), (4, 2), (3, 3), (2, 4), (1, 5)\} \\
 Q_6(A) &= \frac{\sum_{\lambda \in \Lambda} s_{\infty}^*(A)}{|\Lambda|} = \frac{\sum_{\lambda \in \Lambda} \min_{z \in A} \{s_{\infty}(z, z^0, \lambda)\}}{3} \quad (3.30) \\
 Q_6(A) &= \frac{\min\{s_{\infty}((5, 1), z^0, \lambda_1), s_{\infty}((4, 2), z^0, \lambda_1), s_{\infty}((3, 3), z^0, \lambda_1), \\
 &\quad s_{\infty}((2, 4), z^0, \lambda_1), s_{\infty}((1, 5), z^0, \lambda_1)\}}{3} + \\
 &\quad + \frac{\min\{s_{\infty}((5, 1), z^0, \lambda_2), s_{\infty}((4, 2), z^0, \lambda_2), s_{\infty}((3, 3), z^0, \lambda_2), \\
 &\quad s_{\infty}((2, 4), z^0, \lambda_2), s_{\infty}((1, 5), z^0, \lambda_2)\}}{3} + \\
 &\quad + \frac{\min\{s_{\infty}((5, 1), z^0, \lambda_3), s_{\infty}((4, 2), z^0, \lambda_3), s_{\infty}((3, 3), z^0, \lambda_3), \\
 &\quad s_{\infty}((2, 4), z^0, \lambda_3), s_{\infty}((1, 5), z^0, \lambda_3)\}}{3} \\
 Q_6(A) &= \frac{\min\{1, 2, 3, 4, 5\} + \min\{5, 4, 3, 2, 1\} + \min\{5/2, 2, 3/2, 2, 5/2\}}{3} = \\
 Q_6(A) &= \frac{1 + 1 + 3/2}{3} \\
 Q_6(A) &= \frac{7}{6}
 \end{aligned}$$

O valor de $Q_6(B)$ é dado pela equação 3.31.

$$\begin{aligned}
B &= \{(4, 2), (2, 4), (1, 5)\} \\
Q_6(B) &= \frac{\sum_{\lambda \in \Lambda} s_\infty^*(B)}{|\Lambda|} = \frac{\sum_{\lambda \in \Lambda} \min_{z \in B} \{s_\infty(z, z^0, \lambda)\}}{3} \tag{3.31} \\
Q_6(B) &= \frac{\min\{s_\infty((4, 2), z^0, \lambda_1), s_\infty((2, 4), z^0, \lambda_1), s_\infty((1, 5), z^0, \lambda_1)\}}{3} + \\
&\quad + \frac{\min\{s_\infty((4, 2), z^0, \lambda_2), s_\infty((2, 4), z^0, \lambda_2), s_\infty((1, 5), z^0, \lambda_2)\}}{3} + \\
&\quad + \frac{s_\infty((4, 2), z^0, \lambda_3), s_\infty((2, 4), z^0, \lambda_3), s_\infty((1, 5), z^0, \lambda_3)\}}{3} + \\
Q_6(B) &= \frac{\min\{2, 4, 5\} + \min\{4, 2, 1\} + \min\{2, 2, 5/2\}}{3} = \\
Q_6(B) &= \frac{2 + 1 + 2}{3} \\
Q_6(B) &= \frac{5}{3}
\end{aligned}$$

Analisando-se visualmente a Figura 3.7 é possível perceber que a aproximação A apresenta melhor qualidade que a aproximação B. Como $Q_6(A) < Q_6(B)$, a medida Q_6 confirma esta hipótese.

Capítulo 4

Definição do problema de restauração do serviço

O problema de restauração do serviço refere-se ao sistema de distribuição de energia elétrica. Como as companhias necessitam manter os padrões de qualidade do serviço e a satisfação do consumidor, o tempo e a frequência de interrupção têm grande correspondência com estes aspectos. Dessa forma, quando da ocorrência de determinada falha, provocada por intempéries climáticas ou contato de objetos ou animais nas linhas de distribuição, certa parte do sistema perde o fornecimento de energia devido a atuação de disjuntores. O problema de restauração do serviço consiste em restabelecer o fornecimento de energia elétrica aos consumidores afetados no menor tempo possível, respeitando restrições operacionais e de carga, conforme mencionado na seção 1.1.2.

Analisando-se um pouco mais a natureza do problema prático, é possível perceber que a intervenção na rede, de forma a restabelecer ao máximo o fornecimento de energia aos consumidores afetados, deve ser mínima. Isto porque esse estado é transitório e somente deverá ser mantido enquanto o defeito não é eliminado. Daí surge a natureza multicritério do problema de restauração do serviço, a minimização da carga sem fornecimento e do número de chaves manipuladas, sugerindo que a solução para esse problema seja um compromisso entre estes dois critérios. Há outras particularidades que também podem ser consideradas, tais como minimização de perdas elétricas e balanceamento do carregamento entre alimentadores. No entanto este trabalho irá se ater somente aos primeiros dois, devido ao fato de que os últimos são, em geral, considerados apenas para condições normais de operação (Miu et al. 2000).

Convém salientar que os algoritmos propostos para resolver este problema com dois objetivos, descritos nos capítulos subseqüentes, por serem baseados no conceito de oti-

mização de Pareto, podem ser facilmente adaptados para considerar outros critérios além destes dois.

Quanto às restrições, é imprescindível considerar limites de capacidade dos transformadores das subestações, limites de fluxo nas linhas e limites de quedas de tensão nas barras de carga, além, é claro, da radialidade da rede.

A Tabela 4.1 identifica os conjuntos, índices e variáveis do modelo que será apresentado a seguir. Como é preciso considerar o fluxo de potência nas arestas da rede em ambos os sentidos, duplicou-se cada uma delas, com a segunda sendo uma aresta fictícia, a fim de contemplar esta situação. Dessa forma, o fluxo que atravessa a aresta fictícia é sempre no sentido inverso ao convencional.

Outra importante observação diz respeito às unidades das grandezas envolvidas: tensão, corrente elétrica, impedância e potência são descritas em *p.u.*, ou por unidade. Esta representação consiste na definição de valores de base para todas essas grandezas, seguida da substituição dos valores das variáveis e constantes (expressos no Sistema Internacional de unidades) pelas suas relações com os valores base pré-definidos.

O modelo matemático apresentado a seguir baseia-se no trabalho de Ciric e Popovic (2000).

Grafo da rede	
N	Conjunto de nós da rede
M	Conjunto de arestas da rede
Variáveis	
Z_p	Indica se o nó p está atendido ($Z_p = 1$) ou não ($Z_p = 0$)
X_k	Indica se a chave normalmente fechada da aresta k está fechada ($X_k = 1$) ou aberta ($X_k = 0$) após a restauração do serviço
X'_k	Indica se a chave normalmente fechada da aresta fictícia k está fechada ($X'_k = 1$) ou aberta ($X'_k = 0$) após a restauração do serviço
Y_k	Indica se a chave normalmente aberta da aresta k está fechada ($Y_k = 1$) ou aberta ($Y_k = 0$) após a restauração do serviço
Y'_k	Indica se a chave normalmente aberta da aresta fictícia k está fechada ($Y'_k = 1$) ou aberta ($Y'_k = 0$) após a restauração do serviço
Referente aos nós	
N_S	Conjunto de nós fonte da rede
CP	Conjunto de nós consumidores da rede
L_p	Demanda, ou potência, do nó p
R_p	Impedância do nó p
G_p	Potência disponível no nó p
Referente às arestas	
P_k	Fluxo de potência na aresta k
P'_k	Fluxo de potência na aresta fictícia k
F_i	Conjunto de todas as arestas cujos nós iniciais são o nó i
T_i	Conjunto de todas as arestas cujos nós terminais são o nó i
IL_k	Limite de fluxo de potência na aresta k
IL'_k	Limite de fluxo de potência na aresta fictícia k
Referente às chaves	
CF	Conjunto de todas as chaves normalmente fechadas
CA	Conjunto de todas as chaves normalmente abertas
Outros	
V_{max}	Tensão máxima aceitável
V_{min}	Tensão mínima aceitável

Tabela 4.1: Nomenclatura.

$$\text{Min} \sum_{p \in CP} (1 - Z_p) L_p \quad (4.1)$$

$$\text{Min} \sum_{k \in CF} (1 - X_k) + \sum_{k \in CF} (1 - X'_k) + \sum_{k \in CA} Y_k + \sum_{k \in CA} Y'_k \quad (4.2)$$

Sujeito a:

Restrições de capacidade dos alimentadores:

$$\sum_{k \in F_k} P_k \leq G_q, \quad (q = 1, 2, \dots, N_s) \quad (4.3)$$

Restrições de balanço entre oferta e consumo:

$$\sum_{k \in T_i} (P_k + P'_k) - \sum_{k \in F_i} (P_k + P'_k) = L_i Z_i, \quad (i = 1, 2, \dots, N) \quad (4.4)$$

Restrições de limites de fluxo nas linhas (arestas):

$$P_k - IL_k X_k \leq 0, \quad (k = 1, 2, \dots, M) \quad (4.5)$$

$$P'_k - IL_k X'_k \leq 0, \quad (k = 1, 2, \dots, M) \quad (4.6)$$

$$P_k - IL_k Y_k \leq 0, \quad (k = 1, 2, \dots, M) \quad (4.7)$$

$$P'_k - IL_k Y'_k \leq 0, \quad (k = 1, 2, \dots, M) \quad (4.8)$$

Restrições de limites de quedas de tensão nos nós:

$$V_{min} \leq \sqrt{L_k \cdot Z_k \cdot R_k} \leq V_{max}, \quad (k = 1, 2, \dots, N) \quad (4.9)$$

Restrições para que a rede proposta seja radial:

$$\sum_{k \in F_i} (X_k + X'_k + Y_k + Y'_k) \leq 1, \quad (i = 1, 2, \dots, N) \quad (4.10)$$

Restrições para que seja utilizada apenas uma aresta entre dois nós:

$$(X_k + X'_k) \leq 1, \quad (k = 1, 2, \dots, M) \quad (4.11)$$

$$(Y_k + Y'_k) \leq 1, \quad (k = 1, 2, \dots, M) \quad (4.12)$$

Este modelo apresentado é multiobjetivo: minimizar a carga sem fornecimento (expresso pela equação 4.1) e minimizar o número de chaves manipuladas (expresso pela equação 4.2).

As variáveis de decisão do problema incluem o fornecimento de energia às cargas (Z_p) e a condição de cada chave disponível (X_k , X'_k , Y_k e Y'_k). Com a definição das condições de todas as chaves do sistema e das cargas a serem atendidas (pelas variáveis de decisão), o fluxo de carga determina o fluxo de potência nas arestas da rede (P_k e P'_k), condição necessária para avaliar a violação ou não das restrições 4.3-4.12.

O primeiro conjunto de restrições, equação 4.3, refere-se à capacidade dos alimentadores. De fato, esta restrição é muito importante para que o plano de restauração possa ser efetivamente aplicado, sob pena de provocar mais áreas escuras em decorrência da atuação dos dispositivos de proteção pela sobre-capacidade imposta.

O conjunto de restrições especificado pela equação 4.4 corresponde à conservação do fluxo de potência nos nós da rede.

As equações 4.5, 4.6, 4.7 e 4.8 não permitem que seja excedida a capacidade de fluxo nos condutores da rede. Isto torna-se imprescindível, porque pode haver mais desligamentos durante o processo de restauração.

O respeito aos limites de tensão em todos os nós é garantido pelas restrições 4.9. Nesta equação, a queda de tensão em cada nó é calculada através da raiz quadrada do produto entre a potência do nó, a impedância do nó e a variável de decisão que indica se o nó está ou não atendido. O propósito é observar os limites de tensão apenas para os nós atendidos.

As restrições 4.10 garantem que a rede formada pelo processo de restauração seja radial, ou seja, não apresente ciclos. A restrição parte do pressuposto de que não deve haver mais de uma chave fechada incidente em um nó da rede, garantindo, assim, a inexistência de malhas.

Por fim, as equações 4.11 e 4.12 garantem que haverá somente um sentido de fluxo para cada arco da rede. Isto se deve à criação de um arco fictício para descrever sentido de fluxo contrário ao convencional, mas o fluxo existente deve ser ou no sentido convencional (X_k e Y_k) ou no sentido contrário (X'_k e Y'_k).

O modelo matemático apresentado nesta seção servirá como base teórica para o desenvolvimento dos métodos propostos. Tanto os objetivos envolvidos no problema como as restrições consideradas serão baseadas nos aspectos definidos nesse modelo.

4.1 Metodologia para resolução

A partir da seção 1.2 pode-se verificar as várias abordagens de resolução para o problema de restauração do serviço. Percebe-se que predominam aquelas baseadas em heurísticas, das mais simples às mais sofisticadas.

A abordagem proposta neste trabalho também faz uso de heurísticas, pela capacidade de aproveitar ao máximo o conhecimento do problema, além, é claro, de sua desejável simplicidade e rapidez. Serão adotadas algumas sofisticações de forma a torná-la mais robusta e eficiente.

Para resolver o problema de restauração do serviço, algumas premissas são assumidas como válidas:

- O sistema de distribuição corresponde a uma rede radial, tanto pré ou pós-falha;
- O estado da rede antes da falha é conhecido;
- O defeito já foi devidamente isolado, resultando uma rede na qual todas as arestas e todos os nós podem ser utilizados pelo plano de restauração.

Como já foi dito, um requisito desse problema é que a rede de distribuição apresente algum defeito, fazendo com que seja “retirada” da rede parte dela. O subgrafo resultante corresponderá, portanto, à *instância* para o problema de restauração do serviço, com as informações a respeito do estado das chaves antes da falha.

Convenciona-se chamar de *área escura* ao conjunto de consumidores que ficou sem energia após o isolamento do defeito. A *área clara*, que é o conjunto de consumidores que não teve interrupção no fornecimento de energia, apresenta, possivelmente, fronteiras com a *área escura*, nas quais identificam-se os nós-fonte que são utilizados no processo de restauração (Figura 4.1). Convém salientar que poderá haver, após o isolamento do defeito, várias *áreas escuras*, o que pode resultar num subgrafo disjunto.

O problema de restauração do serviço que será resolvido, definido na seção anterior deste capítulo, pressupõe um grafo¹ não orientado como sendo a instância do problema.

Os nós que fazem parte desse grafo correspondem a todos os consumidores sem energia e aos nós-fonte², e mais do que isso: apenas os nós-fonte que forem alcançáveis a partir da(s) *área(s) escura(s)* é que serão incluídos no subgrafo resultante. A Figura 4.1 ilustra um pequeno exemplo do grafo original de uma rede hipotética, após o isolamento do defeito. O subgrafo resultante obtido a partir do grafo da Figura 4.1 é apresentado pela Figura 4.2.

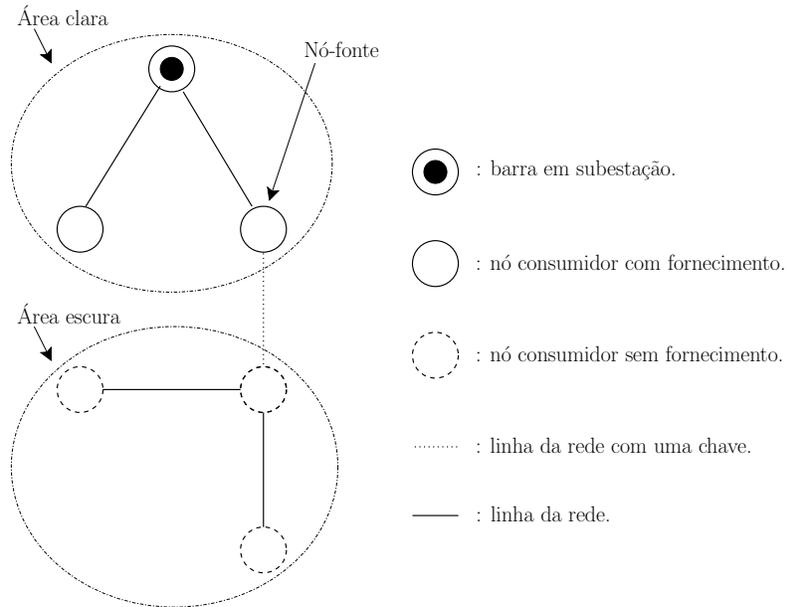


Figura 4.1: Grafo resultante após isolamento do defeito para uma rede hipotética.

Quanto às arestas, um subconjunto delas corresponde às chaves manipuláveis, sendo esse subconjunto dividido em outros dois: o primeiro corresponde às chaves normalmente abertas, enquanto o segundo contém as chaves normalmente fechadas. Com as informações dos estados das chaves antes da falha é possível contabilizar o número de chaveamentos para restabelecer o maior número de consumidores, bastando calcular a diferença de estados das chaves, pré e pós-falha (Equações 4.1 e 4.2).

Com a definição das características e da entrada do problema é possível partir para a determinação da metodologia de resolução. Primeiramente é realizado um pré-processamento da entrada, que compreende uma busca em largura no grafo da instância, para que seja identificada a(s) *área(s) escura(s)*, além de um fluxo de potência para identificar os fluxos e as capacidades de reserva dos alimentadores adjacentes à(s) *área(s) escura(s)*. Depois da execução do algoritmo multicritério proposto, é entregue como saída um conjunto de soluções eficientes, conforme definição da seção 3.1.1. Cada uma destas soluções deverá atender aos requisitos mínimos para uma rede em operação: quedas de tensão e capacidade dos elementos da rede devem estar dentro dos limites estabelecidos (restrições 4.3, 4.9, 4.5, 4.6, 4.7 e 4.8).

Nas seções a seguir apresentam-se uma simplificação adotada e o método de fluxo de

¹Na verdade esse grafo será o subgrafo resultante do processo de isolamento do defeito.

²Esses nós serão os únicos nós da *área clara* que farão parte do subgrafo resultante.

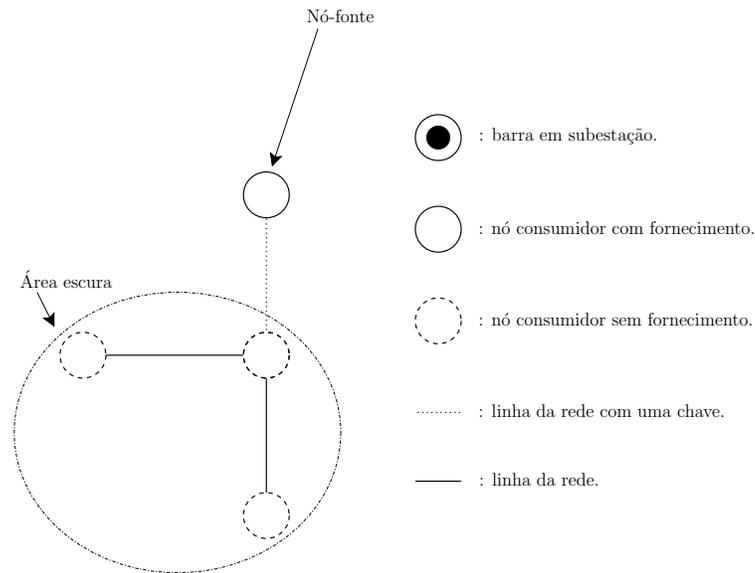


Figura 4.2: Subgrafo obtido a partir da simplificação do grafo da Figura 4.1.

carga utilizado para verificar a factibilidade das soluções encontradas.

4.1.1 Simplificação adotada

Pode-se definir *bloco* (ou *bloco de carga*, ou setor) como sendo parte da rede que reúne vários nós consumidores sem possibilidade de seccionamento entre si. Ou seja, a estrutura de ligação desses blocos permanece inalterada diante de qualquer manobra das chaves da rede. Isto sugere que se aproveite essa estrutura para reduzir o número de nós e arestas da entrada do problema, facilitando, assim, a sua respectiva resolução.

Logo, são considerados nós do grafo apenas os blocos de carga ou nós consumidores que são afetados pela alteração no estado das chaves. As arestas internas aos blocos também não são consideradas no problema. Com isso, procura-se reduzir as alternativas e a complexidade, do ponto de vista da análise combinatória, dos algoritmos envolvidos na busca de caminhos alternativos para interligação de um nó da área escura até um outro nó da área clara. No entanto, são consideradas todas as arestas, inclusive aquelas que se encontram dentro desses blocos, para fins de cálculo do fluxo de potência. De outra forma, para fluxo de potência considera-se a rede completa, para os algoritmos que buscam os caminhos de interligação apenas uma rede resultante com os blocos previamente definidos.

A Figura 4.3(a) ilustra uma rede com três blocos de carga circulados. A rede resultante da simplificação é apresentada pela Figura 4.3(b). As linhas descontínuas representam as

chaves, enquanto o nó com dois círculos concêntricos representa o nó-fonte.

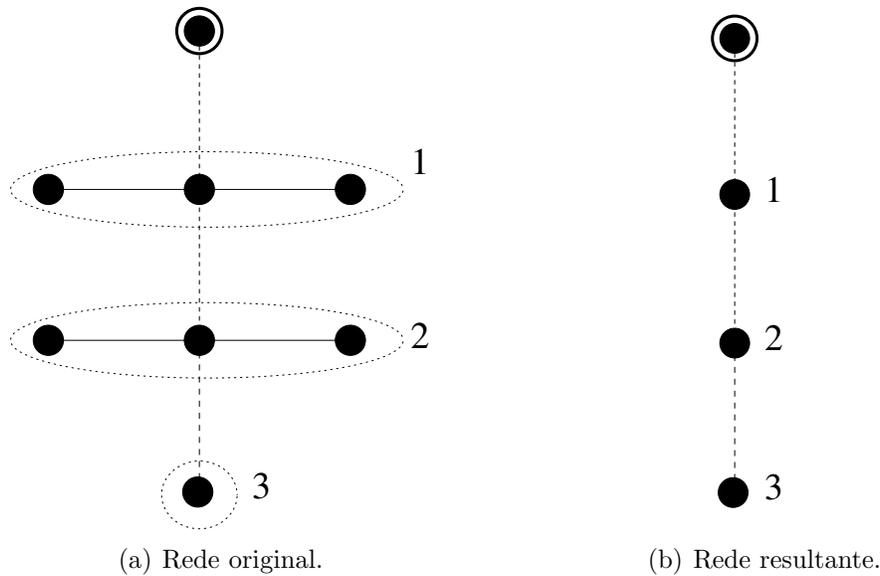


Figura 4.3: Redução da rede através da criação de *blocos de carga*.

4.1.2 Fluxo de carga utilizado

Para verificação das restrições do PRS é necessário calcular o fluxo de carga da configuração correspondente à solução construída. Visto que antes mesmo de avaliar qualquer solução proposta em relação aos dois objetivos escolhidos é preciso verificar se a configuração correspondente de cálculo atende às restrições do problema, torna-se imprescindível a utilização de um método rápido.

Por isso, o método escolhido foi o *backward-forward-sweep*, ou *varredura vai e vem*, proposto por Baran e Wu (1989). Embora seja um método numérico aproximado (em função da tolerância escolhida), sua eficiência em fornecer uma boa aproximação do fluxo real CA em um tempo computacional muito razoável, inclusive para redes que não apresentem compensação do fator de potência, tornou-o muito adequado para a proposta desta tese.

Antes de apresentar as equações utilizadas, ilustra-se na Figura 4.4 um diagrama simplificado de uma rede de distribuição radial. Em cada barra de carga i é considerada uma injeção de potência constante dada pelo número complexo $S_i = P_i + jQ_i$ (P_i refere-se à potência ativa e Q_i à potência reativa), com a tensão calculada nesta barra dada por $V_i = V_{i_{RE}} + jV_{i_{IM}}$ ($V_{i_{RE}}$ é a parte real e $V_{i_{IM}}$ é a parte imaginária do fasor V_i). Um conjunto $A_i = \{a_{i1}, a_{i2}, \dots, a_{im}\}$ inclui todos os arcos m que saem da barra i . O k -ésimo

arco apresenta uma impedância dada por $z_k = r_k + jx_k$ (r é a resistência e x_k é a reatância da aresta k) e uma corrente dada por $I_k = I_{kRE} + jI_{kIM}$, onde I_{kRE} é a parte real e I_{kIM} é a parte imaginária do fasor I_k .

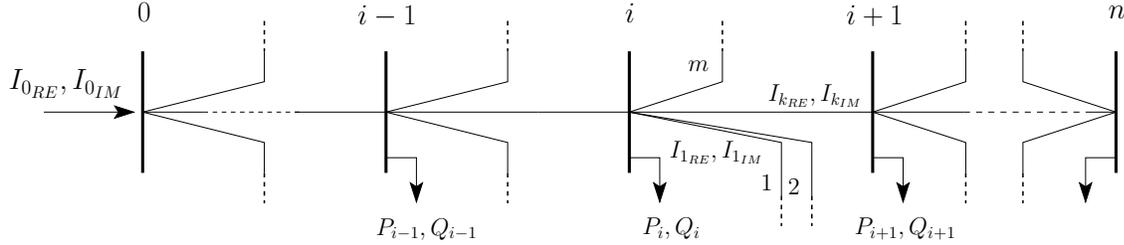


Figura 4.4: Diagrama simplificado de uma rede de distribuição radial.

O método descrito por Baran e Wu (1989) compreende duas etapas básicas, cada uma delas incluindo um conjunto de equações recursivas descritas a seguir: na primeira etapa, chamada de *backward*, fixam-se as tensões nas barras e procede-se com o cálculo dos fluxos nos arcos; a segunda etapa, chamada de *forward*, encarrega-se do ajuste da tensão em função dos fluxos calculados.

A etapa *backward* compreende o cálculo do fluxo nas linhas da rede. Determina-se a corrente de cada arco do conjunto A_i , $i = 1, \dots, n$, através das equações 4.13 e 4.14. Na etapa *forward* são utilizadas as equações 4.15 e 4.16 para os cálculos das tensões nas barras, a partir da corrente obtida da etapa anterior. Nas equações 4.15 e 4.16, o arco k é aquele que interliga as barras de carga $(i+1)$ e (i) . A convergência do método é avaliada através das equações 4.17 e 4.18, que fornecem o desvio entre a potência calculada, obtida através da corrente da etapa *backward* e da tensão da etapa *forward*, e a potência efetiva da barra i . Na primeira delas considera-se o desvio da potência ativa e na segunda o desvio da potência reativa. Caso um dos dois seja maior que uma certa tolerância, em pelo menos uma barra da rede, considera-se que o algoritmo não convergiu.

$$I_{kRE} = \sum_{p \in A_i} (I_{pRE}) + \frac{(P_i V_{iRE} + Q_i V_{iIM})}{V_i^2}, \quad \forall k \in A_{i-1} \quad (4.13)$$

$$I_{kIM} = \sum_{p \in A_i} (I_{pIM}) + \frac{(P_i V_{iIM} - Q_i V_{iRE})}{V_i^2}, \quad \forall k \in A_{i-1} \quad (4.14)$$

$$V_{(i+1)RE} = V_{iRE} - (r_k I_{kRE} - x_k I_{kIM}) \quad (4.15)$$

$$V_{(i+1)IM} = V_{iIM} - (x_k I_{kRE} + r_k I_{kIM}) \quad (4.16)$$

$$\Delta P_i = |(V_{i_{RE}} I_{i_{RE}} + V_{i_{IM}} I_{i_{IM}}) - P_i| \quad (4.17)$$

$$\Delta Q_i = |V_{i_{IM}} I_{i_{RE}} - V_{i_{RE}} I_{i_{IM}} - Q_i| \quad (4.18)$$

O algoritmo simplificado é descrito pela Figura 4.5. Os parâmetros são o conjunto B das barras de carga, o conjunto A dos arcos da rede e a tolerância admitida TOL . Primeiramente inicializam-se as tensões em $p.u.$ em todas as barras de carga da rede (passos 1-3). A seguir inicia-se o laço principal, passos 4-11, que compreende o cálculo dos fluxos (passos 5-7) em todos os arcos da rede, com as equações 4.13 e 4.14, e a posterior atualização das tensões (passo 9) em todas as barras da rede (passos 8 e 9), através das equações 4.15 e 4.16. Ao final, no passo 10, calcula-se o erro entre a potência calculada e a potência efetiva, onde Δ representa o maior valor de todas as barras. Este laço repete-se até que este erro seja menor que a tolerância estabelecida.

```

VarreduraVaiEVem( $B, A, TOL$ )
1  para cada  $i \in B$ 
2  faça  $V_{i_{RE}} \leftarrow 1$ ;
3      $V_{i_{IM}} \leftarrow 0$ ;
4  repita
5     para cada  $i \in B$ 
6     faça para cada  $k \in A$ 
7         faça CALCULAFLUXOS( $i, k$ )
8     para cada  $i \in B$ 
9         faça ATUALIZATENSÕES( $i$ )
10     $\Delta \leftarrow$  CALCULADESVIO()
11  ate  $\Delta < TOL$ 

```

Figura 4.5: Fluxo de carga *VarreduraVaiEVem*.

Maiores detalhes do algoritmo e sobre as equações descritas podem ser encontrados em Baran e Wu (1989) e Shirmohammadi e Hong (1989).

Capítulo 5

Modelagem computacional

Neste capítulo é descrita em detalhes a modelagem computacional desenvolvida. A idéia fundamental é apresentar como os métodos foram transformados em códigos de linguagens computacionais e mais: mostra-se como toda a estrutura necessária para a execução do método transformou-se num sistema capaz de suportar inclusive a agregação de outros métodos, assim como de outros problemas relacionados com os sistemas distribuição de energia elétrica.

Já na fase de concepção do método destinado à resolução do problema de restauração do serviço, com toda riqueza de detalhes práticos associados a uma rede real, constatou-se a alta complexidade associada devido à quantidade de informação a ser manipulada. Isto inclui muitas entidades relacionadas à rede elétrica (cabos, transformadores, chaves, etc.), além daquelas relativas ao método matemático (solução, instância, algoritmo construtivo, etc.). Desse modo, o desenvolvimento do sistema computacional teve de fazer uso de ferramentas que promovessem a modularização como forma de favorecer a compreensão, manutenção e organização do mesmo.

O paradigma de orientação a objetos (Stroustrup 1988) apresenta-se como a metodologia muito adequada, justamente por proporcionar a organização do sistema a partir de conceitos do mundo real (Rumbaugh, Blaha, Lorenzen, Eddy e Premerlani 1991). A linguagem computacional escolhida foi C++ (Stroustrup 2000), por ser orientada a objetos e por possuir recursos que dão ao desenvolvedor grande flexibilidade inclusive para controlar o compromisso legibilidade/eficiência.

As seções a seguir apresentam a estrutura geral do sistema (seção 5.1) e as estruturas de dados empregadas (seção 5.2).

5.1 Diagrama de classes

Para um bom entendimento da arquitetura geral do sistema computacional desenvolvido, apresenta-se na Figura 5.1 o diagrama de classes segundo a *UML (Unified Modeling Language)* (Larman 1998), que ilustra a estrutura lógica estática mostrando as classes e os relacionamentos entre elas.

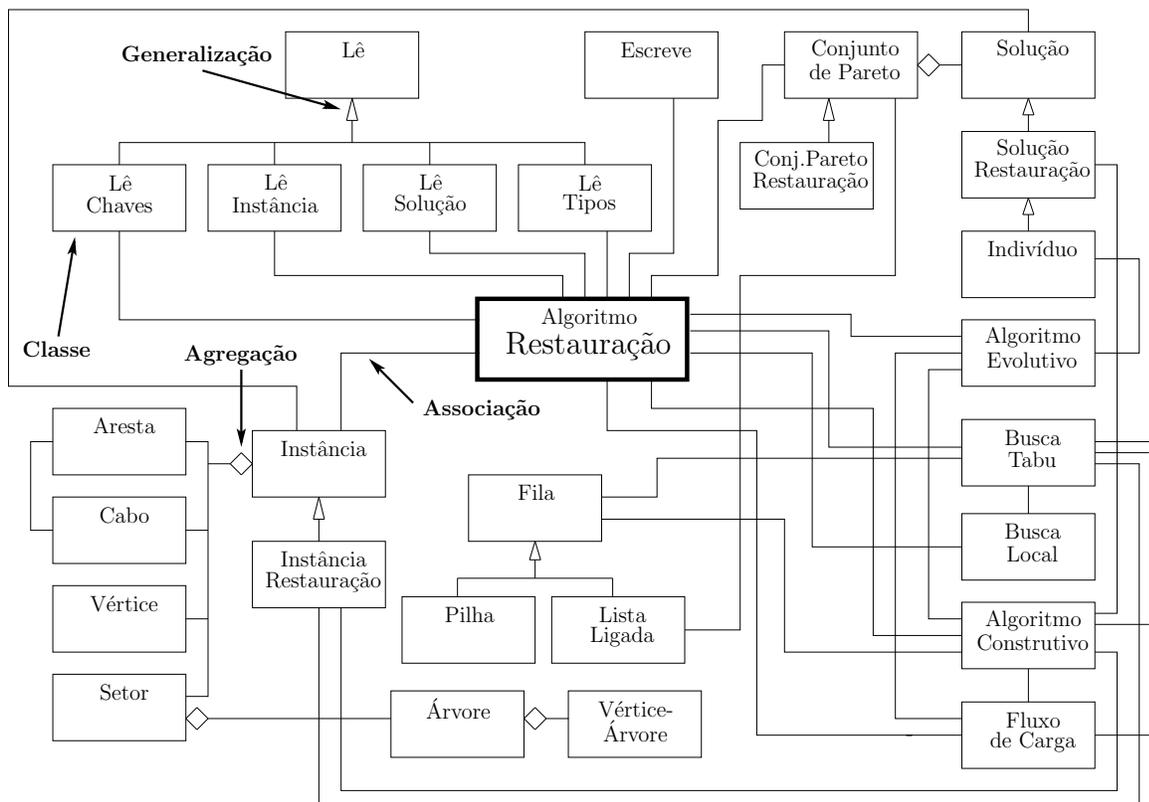


Figura 5.1: Diagrama de classes do sistema computacional.

Na figura anterior estão discriminados os dois elementos principais do diagrama: *classe* e *relacionamento*. O elemento *classe* designa um conjunto de coisas reais ou abstratas que são reconhecidas como sendo do mesmo tipo por compartilhar as mesmas características de atributos, operações, relações e semântica (Furlan 1997). Um *relacionamento* denota qualquer relação entre as classes, que podem ser de três tipos básicos (Furlan 1997):

- *Generalização/Especialização*: indica o relacionamento entre um elemento mais geral e um elemento mais específico (superclasse e subclasse, respectivamente),

também conhecido como herança ou classificação. O elemento mais específico pode conter somente a informação adicional acerca do elemento mais geral.

- *Agregação*: usada para denotar relacionamentos todo/parte. Pode-se ilustrar como exemplo o conjunto de vértices, arestas e setores, que são parte de uma instância.
- *Associação*: utilizada para denotar relacionamentos entre classes não correlatas. Na *UML*, uma associação é definida como um relacionamento que descreve um conjunto de vínculos, onde vínculo é definido como uma conexão semântica entre tuplas de objetos.

O conjunto de classes apresentado na Figura 5.1 está subdividido em cinco módulos: algoritmos, entrada e saída, estruturas de dados auxiliares, instância, e solução. Cada um desses módulos, também chamados de pacotes, relaciona uma coleção de classes com características semelhantes, permitindo mais um nível de modularização do sistema.

A seguir descreve-se cada um desses módulos e os relacionamentos entre as classes incluídas nos mesmos.

5.1.1 Módulo Entrada e Saída

Nesse módulo estão relacionadas classes que servem para fazer a leitura dos dados de entrada para o problema e fornecer o resultado da execução do algoritmo.

As classes relacionadas são:

- *Lê*: essa classe reúne todos os elementos próprios da linguagem para realizar a leitura de dados de um arquivo. Estão incluídos um conjunto de métodos capazes de ler vários tipos de dados, como números (inteiros e reais), seqüências de caracteres e linhas inteiras.
- *Lê chaves*: corresponde a uma especialização da classe *Lê*, destinada à leitura de um arquivo utilizado para descrever quais tipos de chaves poderão ser manipuladas pelo algoritmo de restauração. Isso equivale a dizer quais arestas do grafo, segundo seus tipos, serão considerados como manipuláveis.
- *Lê instância*: também é uma especialização da classe *Lê*. O arquivo que contém todas as informações da instância do problema é lido por ela, compreendendo, entre outras coisas, a relação de vértices e arestas que formarão o grafo sobre o qual o algoritmo trabalhará.

- *Lê solução*: esta classe também aproveita todo código incluído na classe *Lê* através de seu relacionamento de especialização com a mesma. Sua função é ler o arquivo que informa quais arestas estão energizadas, o que significa dizer qual é a rede (árvore) em operação após o isolamento do defeito, portanto a rede pós-falha.
- *Lê tipos*: o arquivo lido por esta classe, outra subclasse da classe *Lê*, informa todos os tipos possíveis para as arestas da rede. Isto inclui não somente os tipos de cabos mas também as possíveis chaves.
- *Escreve*: esta classe agrega métodos para manipulação de arquivos de saída, incluindo mecanismos para escrita de números e seqüências de caracteres.

Todas as classes deste módulo estão relacionadas com a classe *Algoritmo Restauração*, justamente por representarem os procedimentos básicos que precedem e sucedem a execução do método matemático propriamente dito.

5.1.2 Módulo Estruturas de Dados Auxiliares

Neste módulo estão incluídas classes que servem como suporte para os algoritmos e para as estruturas de dados próprias destes. Segue uma breve descrição de todas elas¹.

- *Fila*: esta classe é a implementação de uma estrutura de dados muito conhecida em computação, a chamada fila (*first-in-first-out FIFO*). As funções básicas incluídas são a inclusão (no final) e remoção de elementos (no início).
- *Pilha*: correspondendo a uma especialização da classe *Fila*, a Pilha é um tipo particular de fila na qual tanto a inclusão quanto a remoção de elementos ocorre no final (*last-in-first-out LIFO*).
- *Lista Ligada*: nesta classe é implementada uma versão de uma lista ligada. A justificativa para sua criação é a necessidade de manipular elementos em um conjunto com a inclusão e/ou remoção de elementos em qualquer posição, sugerindo uma estrutura de armazenamento não-sequencial com tamanho variável.
- *Árvore*: a estrutura de dados implementada nesta classe corresponde a uma lista de objetos da classe *Vértice-Árvore* (agregação de objetos desta classe). Seu objetivo é representar um grafo árvore com uma propriedade especial: qualquer vértice pode ser raiz.

¹Para maiores detalhes ver Tenenbaum, Langsam e Augenstein (1995).

- *Vértice-Árvore*: esta classe destina-se unicamente a armazenar o vértice do grafo árvore implementado pela classe *Árvore*. Ela prevê uma lista de vértices sucessores e um campo para armazenar o vértice predecessor.

5.1.3 Módulo Instância

Neste módulo estão reunidas todas as classes relativas à instância do problema de restauração do serviço. Elas correspondem a um relacionamento de agregação, ou seja, as classes *Aresta*, *Vértice*, *Cabo*, e *Setor* são parte da classe *Instância*, que, por sua vez, possui uma classe especializada *Instância Restauração*. Segue uma breve descrição de cada uma delas.

- *Aresta*: esta classe reúne informações a respeito da ligação entre dois objetos da classe *Vértice*. Os dados relacionam o número de cada um desses vértices, o tipo (se é chave, um ramo de ligação, um disjuntor, etc.), além de guardar um campo que relaciona o tipo de cabo (elemento da classe *Cabo*) que forma essa aresta.
- *Cabo*: todos os dados elétricos como resistência, reatância, máxima corrente admissível e corrente econômica são reunidos nessa classe. Ela está associada à classe *Aresta* para relacionar o tipo de cabo e a referência física na rede.
- *Vértice*: esta classe guarda detalhes de cada ponto da rede elétrica. Esses pontos podem ser de vários tipos (vértices de consumo, transformadores, subestações, etc.) e devem ter a referência para a localização física na rede (coordenadas no plano cartesiano, por exemplo).
- *Setor* (bloco): esta classe é uma abstração utilizada neste trabalho e com uma semântica particular. Sua função é armazenar blocos da rede (coleções de arestas e vértices) que não terão sua configuração (ligação) alterada, independente da manobra de chaves escolhida. A estrutura desses blocos é sempre uma árvore, dado o modelo das redes elétricas brasileiras. O relacionamento com a classe *Árvore* denota o uso desta estrutura de dados para realizar a armazenagem dos blocos de forma eficiente.
- *Instância*: o problema propriamente dito, com todos os dados e estruturas auxiliares, é armazenado nesta classe. Ela compreende um conjunto de arestas, um de vértices, um de cabos e um de setores, os quais formam a representação matemática do problema a ser resolvido. Vale lembrar que esta representação é adequada também para problemas de reconfiguração de redes de energia elétrica.

- *Instância Restauração*: esta classe é uma especialização da classe *Instância*, criada com o intuito de particularizar os dados e a representação para o problema de restauração do serviço.

5.1.4 Módulo Solução

Para resolver o problema de restauração, a partir da representação adequada do problema incluída no módulo Instância, é preciso representar adequadamente uma solução. Este módulo tem justamente esta função, qual seja, a de prover classes relacionadas com a representação da solução e classes que incluem conjuntos delas. Segue uma breve descrição de todas elas.

- *Solução*: uma estrutura de dados e os demais itens necessários para representar uma solução para um problema de reconfiguração de redes são incluídos nesta classe. Particularmente, incluem-se nesta classe os dados referentes à estrutura da rede e aos fatores elétricos decorrentes da configuração que a solução representa.
- *Solução Restauração*: esta classe particulariza, através de uma especialização da classe *Solução*, os dados necessários para representar um problema de restauração do serviço.
- *Indivíduo*: esta classe especializa a classe *Solução Restauração* para incluir as estruturas para representar o cromossomo do indivíduo a ser utilizado pelo classe *Algoritmo Evolutivo*.
- *Conjunto de Pareto*: como neste trabalho propõe-se um método multiobjetivo, o resultado do processo de otimização não é uma mas um conjunto de soluções eficientes. Esta classe corresponde à estrutura para armazenar este conjunto de soluções.
- *Conjunto de Pareto Restauração*: através do relacionamento de especialização com a classe *Conjunto de Pareto*, esta classe particulariza a estrutura de conjunto para armazenar as soluções eficientes do problema de restauração do serviço.

Quanto aos relacionamentos entre as classes deste módulo e as demais, salienta-se o relacionamento entre uma solução e a instância do problema a que ela se refere. Isso se faz necessário para que os dados referentes ao problema estejam corretamente associados à solução.

5.1.5 Módulo Algoritmo

Este módulo destina-se às classes que compõem o método matemático propriamente dito. Uma outra classe em particular auxilia nos cálculos elétricos das soluções potenciais: a classe *Fluxo de Carga*.

A seguir são descritas sucintamente essas quatro classes.

- *Fluxo de Carga*: inclui o método de fluxo de carga CA chamado *Varredura VaiEVem* (*backward-forward-sweep method*), descrito na Seção 4.1.2.
- *Algoritmo Construtivo*: esta classe inclui o algoritmo construtivo responsável por criar as soluções iniciais, eventualmente soluções eficientes. Seu relacionamento com a classe *Instância Restauração* se faz necessário pela dependência dos dados relativos ao problema.
- *Busca Local*: nesta classe estão incluídos os algoritmos de busca local do método proposto. O relacionamento com a classe *Busca Tabu* se justifica pela relação de dependência entre elas.
- *Busca Tabu*: esta é a classe que reúne uma das metaheurísticas desenvolvidas para o PRS. Compreende as funções necessárias para melhorar o conjunto de soluções eficientes recebido do algoritmo construtivo, com o uso adequado da busca local.
- *Algoritmo Evolutivo*: esta classe inclui a outra metaheurística desenvolvida para o PRS. Compreende todas as funções necessárias para realizar um processo evolutivo, por meio de um operador de recombinação e outro de mutação, com os indivíduos fornecidos pelo algoritmo construtivo.

5.2 Estruturas de dados

A preocupação em adotar uma representação adequada para o problema se justifica pela grande influência que ela tem no desempenho do algoritmo. Não só a armazenagem como também a manutenção e atualização da estrutura têm um impacto direto no tempo gasto para a execução do algoritmo.

A seguir são descritas as estruturas de dados utilizadas para representar a instância do problema (seção 5.2.1), para o setor (seção 5.2.1.1), para a solução (seção 5.2.2) e para o conjunto de Pareto (seção 5.2.3).

5.2.1 Estrutura de dados para a Instância

Uma rede primária de distribuição define uma instância para o problema de restauração do serviço. Para representá-la, optou-se por um modelo de grafo (Ahuja, Magnanti e Orlin 1993) pela abstração intuitiva e pela existência de algoritmos eficientes para manipulá-lo.

Um modelo de grafo pressupõe uma topologia (estrutura de vértices e arestas) e os dados associados aos custos e capacidades dos vértices e arestas. Como será visto, a forma adotada para representar a topologia sugere imediatamente uma abordagem para incluir os dados associados aos vértices e arestas.

O grafo que se quer representar possui uma característica muito importante: é não-direcionado. Isto porque, dado uma conexão física entre dois postes através de um cabo, o fluxo que o atravessa pode ser em ambos os sentidos. Aproveitando essa informação, e, dado que se necessita com muita freqüência saber se há ligação entre dois vértices, utilizou-se a matriz de adjacência vértice-vértice (Ahuja et al. 1993), somente com a diagonal superior. Embora ela não seja muito eficiente quanto ao espaço gasto, a associação imediata entre dois vértices favorece as operações que são executadas com mais freqüência. Utilizou-se também uma lista para armazenar vértices, pertencentes à classe *Vértice*, e outra para as arestas, da classe *Aresta*, guardando, assim, as propriedades dos elementos referenciados na topologia da rede.

A Figura 5.2 ilustra uma rede hipotética representada através de um grafo não-orientado $G = (N, A)$, sendo N o conjunto de vértices (os círculos da figura, com o respectivo número do vértice) representando os blocos de carga ou subestações e A o conjunto de arestas (identificadas através de um número) representando os cabos físicos que interligam dois vértices. As linhas contínuas correspondem aos trechos de rede e as linhas pontilhadas correspondem às chaves. Eventualmente, algumas arestas podem corresponder a dispositivos seccionadores (chaves), representados na Figura 5.2 pelas linhas descontínuas. Tanto o conjunto de vértices quanto o conjunto de arestas é representado através de uma lista de objetos das classes *Vértice* e *Aresta*, respectivamente.

Na Figura 5.3 é apresentada a matriz que corresponde à representação matemática do grafo da Figura 5.2. Na Tabela 5.1 são apresentadas as listas de vértices e arestas. Vale lembrar que cada elemento da matriz de adjacência vértice-vértice ($m[i][j]$) indica se há uma aresta interligando dois vértices i e j ($m[i][j] > 0$) ou não ($m[i][j] = 0$). No primeiro caso, o (i, j) -ésimo elemento indica o índice da aresta que interliga os vértices i e j na lista de arestas. Com isso é possível acessar em tempo constante ($\mathcal{O}(1)$) a aresta que interliga dois vértices.

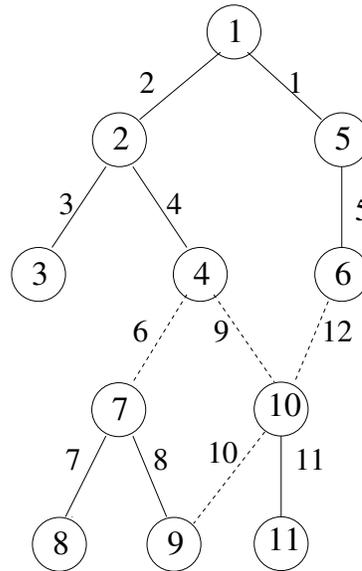


Figura 5.2: Modelo de grafo para uma rede de distribuição.

Lista de vértices										
1	2	3	4	5	6	7	8	9	10	11
1	2	3	4	5	6	7	8	9	10	11

Lista de arestas											
1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	5	6	7	8	9	10	11	12

Tabela 5.1: Listas de vértices e arestas do grafo da Figura 5.2.

5.2.1.1 Estrutura de dados para o Setor

A entidade abstrata *Setor* foi criada com o intuito de facilitar a manipulação do grafo que representa a rede. Sua função é reunir um bloco de vértices e arestas que não têm a sua topologia alterada, qualquer que seja a configuração escolhida para a rede. Em outras palavras, dois vértices só farão parte de um setor se a aresta que os interliga não for uma chave.

Embora um setor reúna um conjunto de vértices, o grafo original permanece inalterado, ou seja, ele é uma estrutura adicional e não substitutiva. A vantagem de tal abordagem surge quando se percorre o grafo e necessita-se construir uma árvore a partir de um vértice. Utilizando a Figura 5.2 como exemplo, identificam-se três blocos: o primeiro

	1	2	3	4	5	6	7	8	9	10	11
1		2	0	0	1	0	0	0	0	0	0
2			3	4	0	0	0	0	0	0	0
3				0	0	0	0	0	0	0	0
4					0	0	6	0	0	9	0
5						5	0	0	0	0	0
6							0	0	0	12	0
7								7	8	0	0
8									0	0	0
9										10	0
10											11
11											

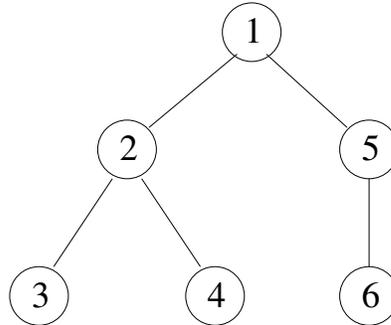
Figura 5.3: Representação para o grafo da Figura 5.2.

formado pelos vértices 1, 2, 3, 4, 5 e 6 (interligados por arestas fixas - não são chaves); o segundo, formado pelos vértices 7, 8 e 9; e o terceiro, que inclui os vértices 10 e 11. Supondo-se a construção de uma árvore que tem como raiz o vértice 1, quando se for incluir as arestas a partir de 1, só é possível incluir 1, 2, 3, 4 e 5 de uma única vez, ou seja, a inclusão do vértice 1 na árvore pressupõe a inclusão conjunta dos vértices 2, 3, 4, 5 e 6. O mesmo critério vale para os outros dois blocos: incluir o vértice 7 a partir do vértice 4 requer a inclusão simultânea dos vértices 8 e 9, enquanto que a inclusão do vértice 10 a partir do vértice 6 pressupõe a inclusão concomitante do vértice 11.

Com isso, dado um vértice a partir do qual pretende-se conectar outro através da escolha de uma aresta adjacente, faz-se necessário saber se este vértice pode ser incluído na árvore isoladamente ou se sua inclusão pressupõe a inclusão conjunta de outros vértices. A estrutura de dados criada para a classe *Setor* permite descobrir se um vértice está ou não incluído em algum setor, em tempo constante ($\mathcal{O}(1)$). Além disso, o subgrafo do setor (equivalente a uma árvore) é obtido através da exploração do mesmo com a escolha adequada de um dos vértices desse setor como raiz. Exemplificando, caso se queira determinar o subgrafo correspondente ao setor dos vértices 7, 8, e 9, basta que se escolha o vértice 7 como raiz e efetuar a exploração a partir dele, o que requer um tempo $\mathcal{O}(a)$, onde a equivale à cardinalidade do conjunto de arestas do setor.

Para exemplificar a representação através da estrutura de dados desenvolvida, incluída na classe *Árvore* (seção 5.1.2), apresentam-se nas Figuras 5.4, 5.5 e 5.6 o subgrafo e a estrutura de dados para os setores 1, 2 e 3, respectivamente, considerando o grafo da

Figura 5.2.



(a) Subgrafo do setor 1.

Setor 1											
	1	2	3	4	5	6	7	8	9	10	11
Árvore	VA ₁	VA ₂	VA ₃	VA ₄	VA ₅	VA ₆	0	0	0	0	0
Vértice-Árvore 1 (VA ₁)			Vértice-Árvore 2 (VA ₂)			Vértice-Árvore 3 (VA ₃)					
Num.vértice: 1			Num.vértice: 2			Num.vértice: 3					
Vértices adj.: VA ₂ VA ₅			Vértices adj.: VA ₁ VA ₃ VA ₄			Vértices adj.: VA ₂					
Vértice-Árvore 4 (VA ₄)			Vértice-Árvore 5 (VA ₅)			Vértice-Árvore 6 (VA ₆)					
Num.vértice: 4			Num.vértice: 5			Num.vértice: 6					
Vértices adj.: VA ₂			Vértices adj.: VA ₁ VA ₆			Vértices adj.: VA ₅					

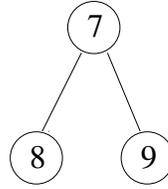
(b) Estrutura de dados do setor 1.

Figura 5.4: Subgrafo e estrutura de dados para o setor 1.

Pelas figuras apresentadas verifica-se que as arestas 6, 9, 10 e 12 do grafo apresentado na Figura 5.2 não constam em nenhum dos três setores, justamente porque são chaves e servem para determinar a fronteira entre eles. O maior objetivo da estrutura de dados do setor é armazenar uma árvore que possa ser enraizada a partir de vários vértices, sem que haja necessidade de mudança na representação. Utilizando-se o setor 2 como exemplo, verifica-se que somente os vértices 7 e 9 podem ser raiz, pois somente estes dois têm arestas adjacentes que são chaves: o nó 7 tem a aresta 6 e o nó 9 tem a aresta 10, conforme ilustra a Figura 5.7. Portanto, a árvore do setor é determinada pelo vértice raiz escolhido e a posterior exploração do grafo, necessitando-se um tempo $\mathcal{O}(a)$ para qualquer caso.

5.2.2 Estrutura de dados para a Solução

Uma das formas mais comuns para representar uma configuração de uma rede de distribuição é através dos estados das chaves. Um vetor, contendo os estados de todas as chaves, pode ser usado para armazenar tal configuração: $[x_1, x_2, \dots, x_n]$, onde $x_i = 1$ se



(a) Subgrafo do setor 2.

Setor 2											
Árvore	1	2	3	4	5	6	7	8	9	10	11
	0	0	0	0	0	0	VA ₇	VA ₈	VA ₉	0	0
Vértice-Árvore 7 (VA ₇)				Vértice-Árvore 8 (VA ₈)				Vértice-Árvore 9 (VA ₉)			
Num.vértice: 7				Num.vértice: 8				Num.vértice: 9			
Vértices adj.: VA ₈ VA ₉				Vértices adj.: VA ₇				Vértices adj.: VA ₇			

(b) Estrutura de dados do setor 2.

Figura 5.5: Subgrafo e estrutura de dados para o setor 2.



(a) Subgrafo do setor 3.

Setor 3											
Árvore	1	2	3	4	5	6	7	8	9	10	11
	0	0	0	0	0	0	0	0	0	VA ₁₀	VA ₁₁
Vértice-Árvore 10 (VA ₁₀)						Vértice-Árvore 11 (VA ₁₁)					
Num.vértice: 10						Num.vértice: 11					
Vértices adj.: VA ₁₁						Vértices adj.: VA ₁₀					

(b) Estrutura de dados do setor 3.

Figura 5.6: Subgrafo e estrutura de dados para o setor 3.

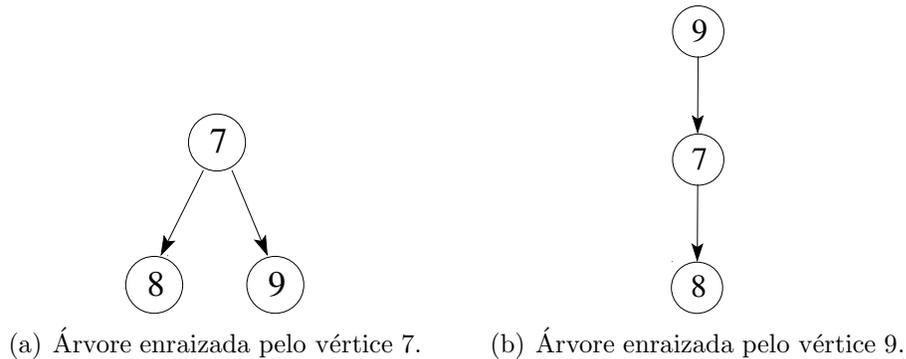


Figura 5.7: Duas possíveis árvores para o setor 2, escolhendo-se os vértices 7 e 9 como raiz.

a chave i está fechada e $x_i = 0$ caso contrário. Esta abordagem foi adotada por Morelato e Monticelli (1989) e Hsu e Kuo (1994). Uma desvantagem está associada à dificuldade na preservação de configurações factíveis quando o espaço de busca de 2^n combinações é considerado. Mesmo com técnicas como as propostas por Aoki et al. (1989), as quais tentam reduzir o número de combinações através da utilização de técnicas heurísticas, tais estruturas podem incorrer em insucesso quando do tratamento de grandes áreas escuras.

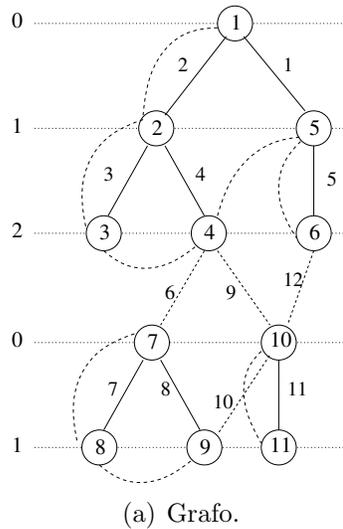
Uma solução para o problema de restauração do serviço corresponde, em geral, a uma floresta, eventualmente a uma árvore. Isto quer dizer que, no melhor caso, todos os nós de consumo terão seu fornecimento restabelecido (haverá apenas uma árvore) ou um subconjunto deles ainda permanecerá sem energia (uma floresta).

Dessa forma, a estrutura de dados para uma solução deve ser suficientemente eficiente para contemplar a manutenção e atualização da floresta. Isso se refere à recuperação de uma árvore, sua exploração, além da identificação do número de árvores e das fronteiras entre elas (arestas que ligam duas árvores).

A Figura 5.8 apresenta um exemplo de solução com a sua respectiva representação na estrutura de dados desenvolvida. A descrição das listas referenciadas na estrutura é a seguinte:

- $P(i)$: apontador para o próximo nó quando a árvore é percorrida em pré-ordem (Ahuja et al. 1993);
- $V(i)$: apontador para o nó pai;
- $N(i)$: indica o nível de profundidade do nó i ;
- $A(i)$: indica o número da árvore na qual o nó está incluído;

- $R(a)$: indica o nó raiz da árvore a .



(a) Grafo.

	Nós										
	1	2	3	4	5	6	7	8	9	10	11
$P(i)$	2	3	4	5	6	0	8	9	0	11	0
$V(i)$	0	1	2	2	1	5	0	7	7	0	10
$N(i)$	0	1	2	2	1	2	0	1	1	0	1
$A(i)$	1	1	1	1	1	1	2	2	2	3	3
	Árvores										
	1	2	3								
$R(a)$	1	7	10								

(b) Estrutura de dados.

Figura 5.8: Grafo e estrutura de dados para uma solução.

Nas quatro listas para os nós, apresentadas na Figura 5.8, são mantidas as árvores da solução. Outra lista, $R(a)$, fornece quais nós são as raízes das árvores existentes. Percebe-se que, dessa forma, é possível saber em tempo constante ($\mathcal{O}(1)$) se dois nós pertencem ou não à árvores distintas, o que é de grande valia para a eficiência computacional dos métodos desenvolvidos.

5.2.3 Estrutura de dados para o Conjunto de Pareto

Conforme dito no Capítulo 3, o resultado do processo de otimização multiobjetivo é um conjunto de soluções eficientes (não-dominadas). Sendo assim, é preciso ter uma estrutura de dados que suporte as operações básicas durante o processo de otimização, a saber, (1) verificar se uma solução é dominada ou não-dominada, (2) incluir uma solução no conjunto e (3) excluir soluções dominadas por uma solução candidata.

Com essas três operações básicas definidas, é possível determinar uma estrutura de dados adequada para que o tempo gasto para efetuá-las seja o menor possível. Pelo estudo de Mostaghim, Teich e Tyagi (2002), verifica-se que a sofisticação das estruturas de dados para armazenagem de conjuntos de Pareto depende muito do tamanho do conjunto e do número de objetivos a serem otimizados. No caso particular deste trabalho, o uso de listas lineares estabelece uma relação custo-benefício muito interessante.

A Figura 5.9 ilustra a lista linear empregada para o problema de restauração do serviço. Ela é composta de um descritor que armazena três informações: o início e o final da lista, bem como o número de elementos nela contidos. Para cada elemento (solução) incluído na lista, emprega-se uma estrutura capaz não só de armazená-lo como também de obter o elemento seguinte. A operação mais custosa é a verificação de dominância de uma solução candidata, que deve ser comparada com todas as soluções do conjunto de Pareto, no pior caso, pois assim que houver uma solução do conjunto que domine a solução candidata o processo é interrompido. Quanto à eliminação de soluções já armazenadas que são dominadas pela solução atual, isso é feito no mesmo momento em que é verificada a dominância, bastando, para isso, apenas uma passagem pelo conjunto.

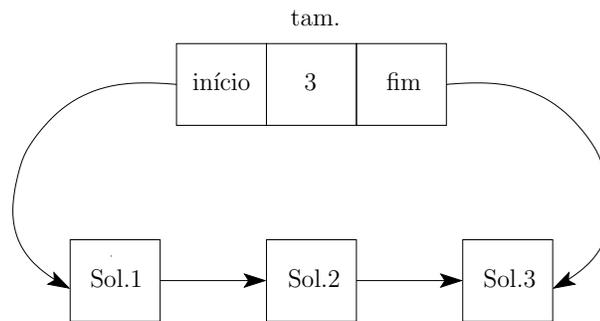


Figura 5.9: Lista linear empregada como estrutura de dados para manter o conjunto de Pareto.

Capítulo 6

Algoritmos heurísticos multiobjetivo para o PRS

Neste capítulo são descritos dois algoritmos heurísticos multiobjetivo desenvolvidos para resolver o PRS definido no Capítulo 4. O primeiro é um algoritmo construtivo que gera o conjunto inicial de soluções não-dominadas e incorpora aleatoriedade num procedimento baseado em algoritmos de construção de árvores geradoras. O segundo é um algoritmo de busca local que encarrega-se do melhoramento deste conjunto inicial através do uso de uma vizinhança multiobjetivo. Em ambos, o critério de otimização utilizado é a relação de dominância de Pareto.

Dois aspectos ocupam lugar de destaque quando se constroem algoritmos para resolução do PRS: a verificação da factibilidade e a avaliação das funções objetivo de uma solução potencial. Considerando o primeiro, as duas fases (construtiva e de melhoramento) já incorporam a verificação da factibilidade para gerar novas soluções. Isso significa que apenas são construídas soluções que não violam as restrições impostas¹. Para a avaliação de uma solução, consideram-se os objetivos definidos no Capítulo 4, a saber o número de chaves manipuladas e a carga dos consumidores que não tiveram o fornecimento restabelecido.

Para a verificação das restrições é necessário calcular o fluxo de carga na rede. Esse procedimento, incluído na classe “Fluxo de carga” (Seção 5.1.5), requer um esforço computacional bastante significativo, merecendo uma atenção especial pois é executado sempre que uma nova solução candidata é gerada. Justamente pela eficiência quanto ao tempo computacional e pela qualidade da aproximação fornecida, optou-se pelo método descrito

¹As restrições do PRS são os limites de corrente nos condutores, da capacidade dos transformadores e de quedas de tensão.

na Seção 4.1.2. Convém salientar que a aplicação do método de fluxo de carga restringe-se somente ao alimentador que teve alterações, ou seja, tanto no algoritmo construtivo quanto no algoritmo de busca local são calculados os fluxos de uma rede somente nos alimentadores que tiveram alteração de carregamento. Obtém-se, com isso, uma redução sensível quanto ao tempo gasto para validar uma solução.

As seções a seguir detalham os algoritmos construtivo (Seção 6.1) e de busca local (Seção 6.2).

6.1 Algoritmo construtivo

O algoritmo construtivo tem por objetivo criar soluções não-dominadas que servirão como soluções iniciais para a busca local. Não é possível garantir que existam soluções não-dominadas, contudo somente elas serão armazenadas no conjunto inicial. Pressupõe-se que houve o isolamento do defeito que ocasionou a contingência, o que proporciona a completa remoção dos elementos indisponíveis. Desta forma, supõe-se que todos os vértices e arestas que foram informados na instância do problema não apresentam nenhum tipo de defeito e podem ser usados nas soluções propostas sem restrições.

As Figuras 6.1(a) e 6.1(b) ilustram uma rede hipotética antes e depois do isolamento do defeito, respectivamente. A parte circulada na Figura 6.1(a) representa o local do defeito propriamente dito, incluindo o vértice 3 e as arestas 3 e 6. Percebe-se que a rede obtida com o isolamento do defeito (Figura 6.1(b)) não inclui nenhum dos elementos circulados na Figura 6.1(a). É esta rede que corresponde à instância do PRS.

Também na Figura 6.1 é apresentada a rede em operação (Figura 6.1(c)), demonstrando quais arestas da instância são utilizadas e em que sentido (arcos), sempre mantendo uma estrutura radial. Essa rede em operação é, portanto, um grafo acíclico direcionado (árvore direcionada) que determinará quais consumidores continuam com o fornecimento de energia e quais tiveram o fornecimento interrompido, ou seja, quais consumidores pertencem à *área clara* e quais pertencem à *área escura*³. Na Figura 6.1(c) verifica-se que, da configuração original, a manutenção dos arcos 1, 2, 4 e 5 no estado original resultou na permanência dos nós 1, 2, 4, 5 e 6 na área clara (considerando o nó 1 como raiz). Os nós 7-11 ficaram na área escura ainda que os arcos 10, 11, 12 e 14 estejam mantidos no mesmo estado, porque as arestas 7-9 são chaves e estão abertas. Estes dois subgrafos do

²As linhas contínuas correspondem às chaves fechadas e as linhas pontilhadas correspondem às chaves abertas; a direção do arco indica o caminho até a barra da subestação; os nós correspondem às barras de carga.

³Parte da rede com o fornecimento de energia elétrica interrompido.

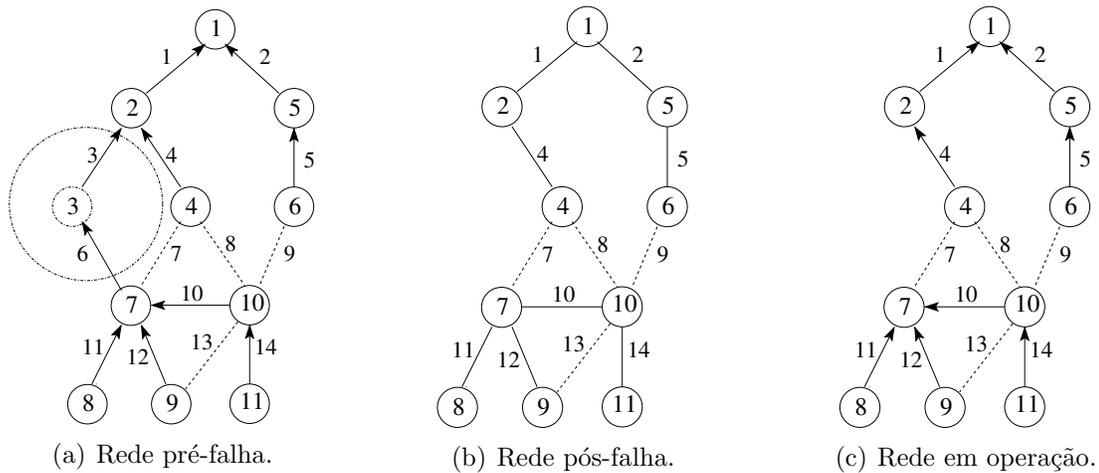


Figura 6.1: Redes antes e depois do isolamento do defeito e a rede em operação².

grafo da instância, uma árvore para a área clara e outra para a área escura, representam uma possível solução para o PRS com a característica especial de não apresentar carga restaurada.

Com a determinação dessas duas áreas, vislumbrou-se uma metodologia para construção de soluções para o PRS que fizesse uso do conceito de *nó-fonte*: são nós que estão localizados na fronteira entre a área clara e a área escura. Na Figura 6.1(c) identificam-se os nós 4 e 6 como sendo os nós-fonte da rede em operação. A partir deles serão construídas as soluções para o PRS. Cabe salientar que a ordem de escolha dos nós adjacentes aos nós-fonte influenciará a solução final obtida, dado que a capacidade de fluxo dos arcos predecessores é limitada.

A construção das novas soluções é realizada por meio de dois algoritmos distintos, um dando ênfase às características do primeiro objetivo e o outro dando ênfase às características do segundo objetivo. Na verdade são criados dois problemas de otimização mono-critério, o primeiro com a função objetivo incluindo a carga restaurada e o segundo tendo como função objetivo o número de chaveamentos. Com isso espera-se incluir soluções potencialmente eficientes e com boa diversidade.

Além disso, a exploração da área escura é feita de duas formas: com uma busca em largura para o primeiro objetivo e por meio de uma busca em profundidade para o segundo. Segundo avaliações experimentais, o emprego de ambas as buscas torna-se imprescindível para a obtenção de soluções topologicamente distintas. Entretanto, a qualidade delas com respeito aos critérios estabelecidos não se mostrou muito alta. Dessa forma, os

resultados obtidos não apresentaram diferenças significativas quando da utilização da busca em largura (mais abrangente) no primeiro algoritmo em comparação à utilização desta busca no segundo algoritmo.

As seções a seguir detalham cada um desses algoritmos.

6.1.1 Algoritmo construtivo segundo o critério de máxima carga restaurada

O primeiro algoritmo realiza um procedimento semelhante ao algoritmo de *Prim* (Ahuja et al. 1993) para o problema de árvore geradora de custo mínimo. A partir de cada nó-fonte, adicionam-se **um a um** os nós da área escura (ou um bloco - seção 5.2.1.1) de forma a manter o grafo acíclico, verificando-se a cada elemento adicionado se as restrições são respeitadas. Como a função objetivo busca minimizar a carga não restaurada, ou maximizar a carga restaurada, a escolha do próximo nó a ser adicionado deve levar em conta a respectiva carga. Mais especificamente, da lista de nós adjacentes ao nó-fonte escolhido, realiza-se uma escolha aleatória ponderada pela carga de cada nó adjacente⁴, ou seja, o nó com maior carga tem maior chance de ser escolhido.

Neste algoritmo, a forma com que os nós adjacentes são escolhidos corresponde a uma busca em largura, ou seja, dado um nó-fonte, somente são considerados os nós filhos de seus sucessores depois que todos estes já tenham sido avaliados. Este procedimento torna-se particularmente importante porque não é possível saber *a-priori* quantos nós da área escura serão adicionados à árvore da área clara, devido às restrições do problema. Isso faz com que a ordem de inclusão seja um fator determinante no número final de nós da área escura que serão interligados à área clara.

Com a escolha do nó adjacente definida, parte-se para a sua interligação ao grafo que corresponde à área clara, com a posterior avaliação da factibilidade pelo cálculo dos fluxos. Caso a inclusão do nó adjacente não viole as restrições do problema, tenta-se incluir a nova solução (com um novo nó no grafo) no conjunto de soluções eficientes (conjunto aproximado de Pareto). O procedimento se repete até que não seja possível incluir nenhum nó adjacente (pertencente à área escura) devido à violação de alguma restrição.

Dado a aleatoriedade incluída no procedimento, espera-se obter soluções diferentes a cada execução. A Figura 6.2 ilustra um conjunto de soluções geradas com o algoritmo construtivo, segundo o critério de máxima carga restaurada, a partir da rede em operação

⁴Esse procedimento de escolha corresponde a uma roleta ponderada (*roulette wheel*), detalhada em Bäck et al. (2000).

ilustrada na Figura 6.1(c).

Observa-se que a construção de uma solução visa incluir os nós da área escura a partir dos nós-fonte, mantendo, sempre que possível, a mesma posição das chaves da área escura quando comparadas à rede pré-falha (Figura 6.1(a)). Como exemplo, seja a Figura 6.2(a): a chave correspondente ao arco 14 é mantida fechada porque ela não interfere na inclusão do nó 7 na área clara. No entanto, as chaves correspondentes aos arcos 10, 12 e 11 tiveram de ser abertas para que somente o nó 7 fosse adicionado à área clara. Caso isso não fosse feito, os nós 8, 9 e 10 também seriam adicionados à área quando o nó 7 o fosse, o que contraria a idéia original de incluir apenas um nó por vez.

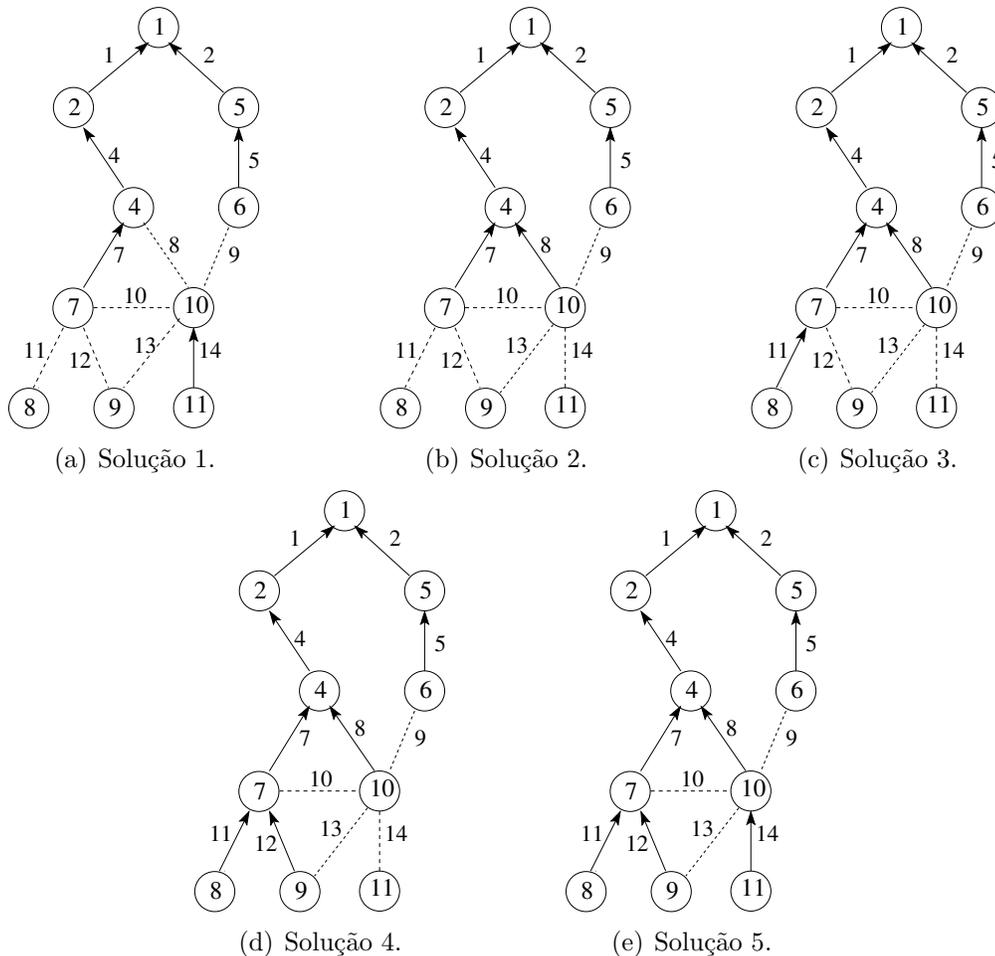


Figura 6.2: Soluções obtidas² com o algoritmo construtivo segundo o critério de máxima carga restaurada, a partir do nó-fonte 4.

As cinco soluções ilustradas nas Figuras 6.2(a)-6.2(e) são resultado da aplicação do

algoritmo construtivo à rede apresentada na Figura 6.1(c), a partir do nó-fonte 4. As duas primeiras, Figura 6.2(a) e 6.2(b), são obtidas a partir da ligação dos dois nós adjacentes ao nó-fonte 4, quando se ligam os nós 7 e 10 e eles se tornam novos nós-fonte. A escolha do próximo nó-fonte, a partir do qual serão verificados os nós adjacentes, será de tal forma que eles sejam “filhos” do nó fonte original (nó-fonte 4). Neste caso, considerando-se o critério de máxima carga, somente será escolhido um novo nó-fonte filho quando todos os nós adjacentes forem avaliados, conforme a seqüência da Figura 6.2: somente passa-se a analisar o nó-fonte 7 (Figura 6.2(c)) após a inclusão de todos os nós adjacentes ao seu nó-fonte pai, nó 4 (Figuras 6.2(a) e 6.2(b)). Outra observação importante diz respeito à ordem de escolha dos nós adjacentes: será sempre realizada de forma aleatória, com probabilidade de escolha proporcional à carga.

As próximas cinco figuras (6.3(a)-6.3(e)) ilustram as soluções obtidas pela aplicação do algoritmo construtivo segundo o critério de máxima carga restaurada, a partir do próximo nó-fonte original, o nó 6. Percebe-se que o nó-fonte filho 7 somente é considerado (Figura 6.3(e)) após a inclusão de todos os nós adjacentes ao seu nó-fonte pai (10).

O procedimento completo referente ao algoritmo construtivo segundo o critério de máxima carga restaurada é apresentado em pseudo-código na Figura 6.4. Ele resume-se à determinação de um nó-fonte e à escolha subsequente de um nó adjacente pertencente à área escura que não exceda a capacidade de reserva deste nó (carga que pode ser transferida sem que os limites do alimentador sejam violados).

O algoritmo começa com a inicialização da solução candidata, que é feita com a solução em operação, passo 1. No passo 4 é escolhido o primeiro elemento da lista de nós-fonte (*LNF*) como sendo o nó-fonte corrente, calculando-se a seguir a sua capacidade de reserva (passo 5). A escolha do nó sucessor, passo 6, é realizada observando-se a carga daqueles nós que pertencem à área escura, escolhendo-se um deles de forma aleatória com probabilidade proporcional às respectivas cargas, sempre observando se a carga é menor ou igual à capacidade de reserva. Caso exista um nó candidato sorteado que respeite a capacidade de reserva, ele é adicionado à árvore no passo 9, realizando-se, antes, uma cópia da solução candidata construída até então (passo 8). Também no passo 9 é incluído no final da lista de nós-fonte (*LNF*) o nó sorteado⁵ adjacente ao nó-fonte atual. No passo 10 é verificado se a solução construída, obtida através da interligação do nó sorteado à árvore correspondente à área clara, respeita as restrições do problema. Caso isso não ocorra, a solução é marcada como inactível, implicando seu descarte. No entanto, caso a solução seja factível, o passo 11 possibilitará primeiramente que ela se torne a nova solução candidata (passo 12), a qual será incluída ou não no conjunto de soluções eficientes (*CP*), de acordo com a relação

⁵Caso o nó sorteado pertença a um bloco, são incluídos todos os nós-fonte encontrados no bloco.

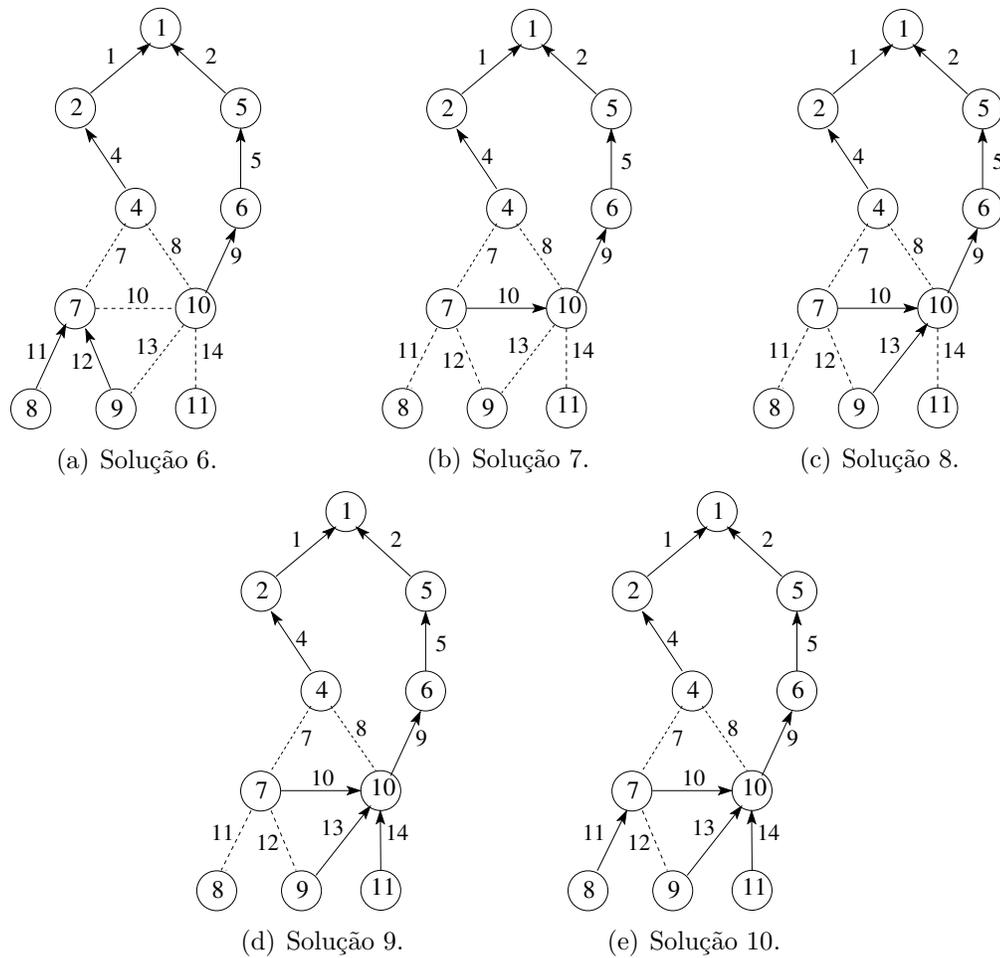


Figura 6.3: Soluções obtidas² com o algoritmo construtivo, segundo o critério de máxima carga restaurada, a partir do nó-fonte 6.

de dominância previamente definida (passo 13).

Esta seqüência (passos 2-14) repete-se enquanto houver nós-fonte que possuam nós adjacentes pertencentes à área escura com carga menor ou igual à capacidade de reserva destes nós-fonte. Quando for escolhido um nó-fonte que não atenda a estas restrições, o mesmo é excluído da lista LNF no passo 14.

```

ALGCONSTMAXCARGA( $LNF, redeOp, CP$ )
1   $solCand \leftarrow redeOp$ ;
2  enquanto  $|LNF| > 0$ 
3  faça
4       $noFonte \leftarrow LNF[0]$ ;
5       $capRes \leftarrow CALCULACAPRESERVA(noFonte, solCand)$ ;
6       $noCand \leftarrow SORTEIASUCESSOR(noFonte, capRes, solCand)$ ;
7      se  $noCand \neq 0$ 
8          então  $sol \leftarrow solCand$ ;
9              ADICIONANÓ( $noCand, sol, LNF$ );
10             VERIFICARESTRICÇÕES( $sol$ );
11             se FACTÍVEL( $sol$ )
12                 então  $solCand \leftarrow sol$ ;
13                 ADICIONASOL( $solCand, CP$ );
14     senão  $LNF \leftarrow LNF - \{noFonte\}$ ;

```

Figura 6.4: Pseudo-código do algoritmo construtivo, segundo o critério de máxima carga restaurada.

6.1.2 Algoritmo construtivo segundo o critério de mínimo chaveamento

Este segundo algoritmo construtivo inclui nós adjacentes aos nós-fonte escolhidos segundo o critério de mínimo chaveamento, o que significa escolher preferencialmente aqueles arcos (chaves) cujos estados não sejam alterados em relação à rede antes da contingência.

Tal como no algoritmo construtivo anterior, este procedimento assemelha-se muito ao algoritmo de *Prim* (Ahuja et al. 1993). Também se baseia na inclusão, um a um, de nós adjacentes aos nós-fonte com a escolha aleatória, apenas observando o critério de mínimo chaveamento ao invés da carga restaurada. A verificação das restrições e a manutenção de um grafo acíclico permanece exatamente igual ao algoritmo anterior. Além do critério para escolha dos nós, existe outra diferença fundamental: a manutenção da lista de nós-fonte. Aqui esta lista é administrada como uma pilha, com o objetivo

de realizar uma exploração segundo uma busca em profundidade. Dessa forma obtém-se soluções topologicamente diferentes das obtidas com o algoritmo anterior, sempre visando máxima diversidade no conjunto de soluções eficientes obtido ao final da execução da etapa construtiva.

A Figura 6.5 ilustra um conjunto de soluções geradas com o algoritmo construtivo segundo o critério de mínimo chaveamento e com o nó 4 como sendo o nó-fonte inicial, a partir da rede em operação ilustrada na Figura 6.1(c). Percebe-se pelas Figuras 6.5(a)-6.5(e) a diferença topológica entre essas soluções e aquelas apresentadas nas Figuras 6.2(a)-6.2(e). Isso se deve à escolha dos nós-fonte, administrados neste algoritmo como uma pilha. Além disso, ainda há a diferença quanto à escolha dos nós adjacentes, os quais serão escolhidos segundo o mínimo número de chaveamentos em relação ao estado das chaves encontrado na rede pré-falha e necessário para agregar um nó adjacente de cada vez. A escolha também será ponderada tal como no algoritmo anterior, sendo que aqueles nós adjacentes interligados por arcos representam uma chave fechada na rede pré-falha terão mais chance de serem escolhidos para interligação à área clara.

A Figura 6.6 apresenta as soluções obtidas com o algoritmo construtivo segundo o critério de mínimo chaveamento, considerando como nó-fonte inicial o nó 6. Para exemplificar a utilização do critério de mínimo chaveamento, toma-se o nó 10 como nó fonte corrente, Figura 6.6(b). Neste caso há três nós adjacente: 7, 9 e 11. A escolha pelo nó 7 se deve ao estado do arco que interliga o nó 10 ao nó 7: tanto nesta solução como na rede antes da falha, Figura 6.1(a), este arco é utilizado, o que equivale a fechar a chave que interliga os dois nós. Dessa forma, com probabilidade bem maior que os outros dois (nós 9 e 11), interliga-se o nó 7 à área clara.

Também nas soluções das Figuras 6.6(a)-6.6(e) é possível perceber uma grande diferença topológica em relação àquelas obtidas com o critério de máxima carga (Figuras 6.3(a)-6.3(e)), privilegiando a diversidade no conjunto de soluções eficientes e favorecendo o algoritmo de melhoramento que fará uso destas soluções.

Por fim ilustra-se em pseudo-código o algoritmo construtivo que utiliza o critério de mínimo chaveamento na Figura 6.7. Esse algoritmo é muito semelhante àquele apresentado na Figura 6.4, os únicos passos que apresentam diferença são o 4 e o 6. No passo 4 toma-se como nó-fonte atual o último elemento na lista de nós-fonte (LNF). No passo 6, o sorteio do nó adjacente escolhido considera o critério de mínimo chaveamento, o que privilegia aqueles nós cujos estados dos arcos que os interligam ao nó-fonte atual permaneçam inalterados em relação à rede antes da falha. Dizendo de outra forma, como utilizar um arco significa fechar uma chave, aqueles arcos que já eram utilizados na rede antes da falha (Figura 6.1(a)) terão maior chance na escolha dos nós adjacentes do passo 6.

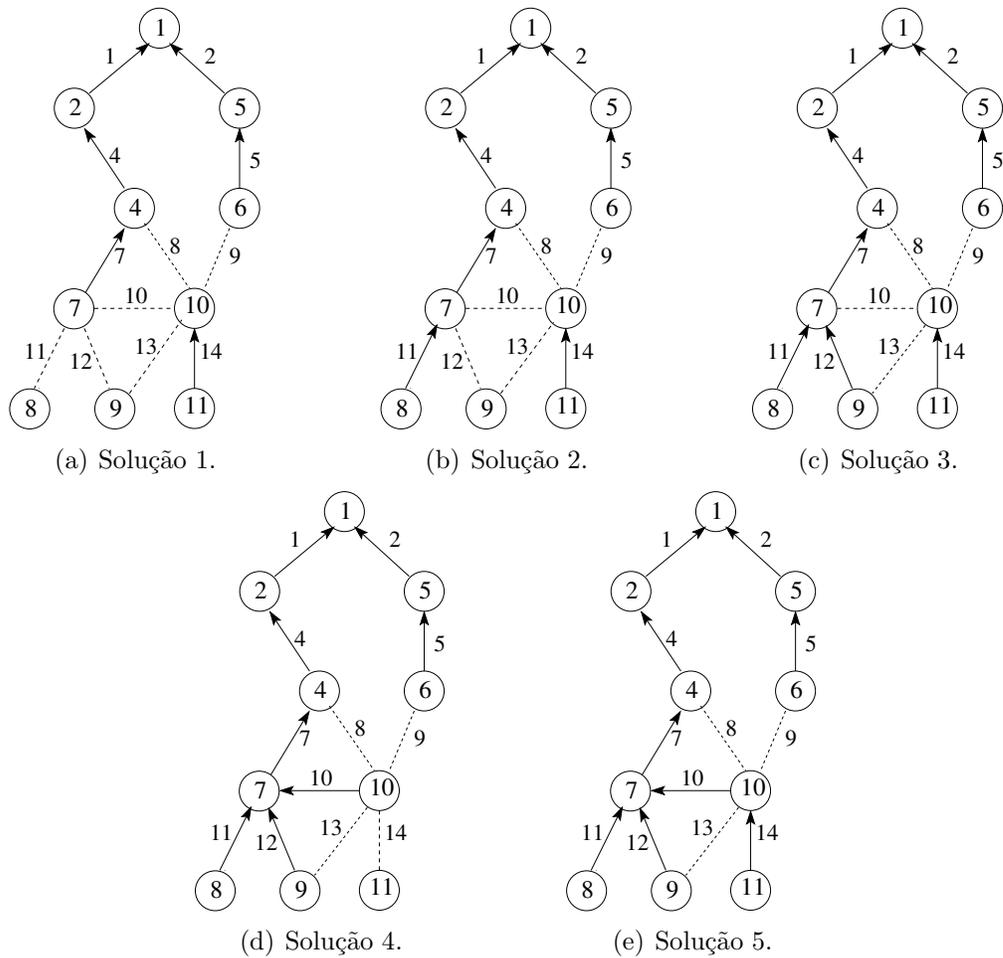


Figura 6.5: Soluções obtidas² com o algoritmo construtivo, segundo o critério de mínimo chaveamento, a partir do nó-fonte 4.

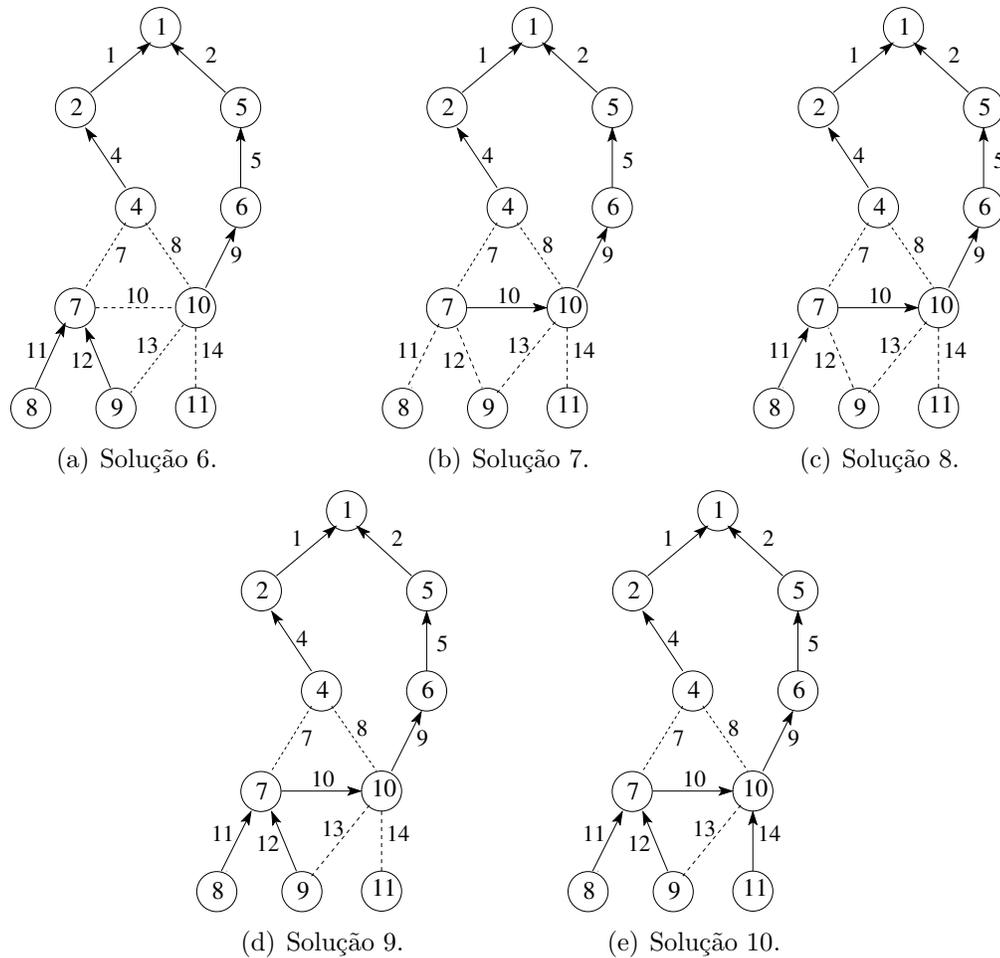


Figura 6.6: Soluções obtidas² com o algoritmo construtivo, segundo o critério de mínimo chaveamento, a partir do nó-fonte 6.

```

ALGCONSTMINCHAV( $LNf$ ,  $redeOp$ ,  $CP$ )
1   $solCand \leftarrow redeOp$ ;
2  enquanto  $|LNf| > 0$ 
3  faça
4       $noFonte \leftarrow LNf[|LNf| - 1]$ ;
5       $capRes \leftarrow \text{CALCULACAPRESERVA}(noFonte, solCand)$ ;
6       $noCand \leftarrow \text{SORTEIASUCESSOR}(noFonte, capRes, solCand)$ ;
7      se  $noCand \neq 0$ 
8          então  $sol \leftarrow solCand$ ;
9               $\text{ADICIONANÓ}(noCand, sol, LNf)$ ;
10              $\text{VERIFICARESTRICÇÕES}(sol)$ ;
11             se  $\text{FACTÍVEL}(sol)$ 
12                 então  $solCand \leftarrow sol$ ;
13                  $\text{ADICIONASOL}(solCand, CP)$ ;
14         senão  $LNf \leftarrow LNf - \{noFonte\}$ ;

```

Figura 6.7: Pseudo-código do algoritmo construtivo, segundo o critério de mínimo chaveamento.

6.2 Busca local

O método de busca local desenvolvido para o PRS é baseado no trabalho de Arroyo (2002) e tem por objetivo melhorar as soluções fornecidas pelo algoritmo construtivo, tentando aproximá-las da fronteira Pareto-ótima através de um procedimento de otimização baseado na relação de dominância de Pareto. Com isso, não é otimizado um, mas sim todos os critérios definidos, buscando-se soluções eficientes preferencialmente bem distribuídas na fronteira obtida. Quer-se, portanto, obter uma aproximação do conjunto de Pareto onde existam soluções não-dominadas que representem toda a fronteira aproximada de Pareto.

A busca consiste em primeiramente definir uma vizinhança de uma dada solução construtiva da qual é gerada um conjunto de soluções vizinhas. Ao contrário da busca local mono-objetivo, onde se escolhe apenas uma solução vizinha que melhore a solução incumbente, na busca local multiobjetivo escolhem-se todas as soluções não-dominadas. Tenta-se, ao final de cada iteração, incluir as soluções vizinhas não-dominadas no conjunto de soluções melhoradas (CP_{mel}), ou seja, num conjunto que reúne todas as soluções não-dominadas encontradas desde a solução construtiva. O conceito de mínimo local associado à busca local mono-objetivo tem um novo significado para o caso multiobjetivo: aqui um mínimo local se caracteriza pela impossibilidade de gerar soluções vizinhas não-dominadas. Esta situação é normalmente utilizada como critério de parada. Um outro

pode ser a fixação de um determinado número máximo de iterações.

A Figura 6.8 apresenta o pseudo-código da busca local multiobjetivo desenvolvida. Os parâmetros do algoritmo são o número máximo de iterações (MAX_{iter}), o número de soluções nas quais será realizada a busca local simultaneamente (N_{sol}) e o conjunto de soluções eficientes obtido com o algoritmo construtivo (CP). A saída do algoritmo será um conjunto de soluções eficientes que compreende todas as soluções de CP mais aquelas que foram obtidas com a busca (CP_{mel}).

```

BUSCALOCALMULTIOBJETIVO( $MAX_{iter}, N_{sol}, CP$ )
1   $CP_{mel} \leftarrow CP$ ;
2   $it \leftarrow 0$ ;
3  enquanto  $it < MAX_{iter}$  e  $CP_{mel} \neq \emptyset$ 
4  faça  $CP' \leftarrow REDUZ(N_{sol}, CP_{mel})$ ;
5      $CP_{viz} \leftarrow \emptyset$ ;
6     para cada  $s_k \in CP'$ 
7     faça  $N(s_k) \leftarrow GERAVIZINHANÇA(s_k)$ ;
8         para cada  $s \in N(s_k)$ 
9         faça se  $NÃODOMINADA(s, CP_{viz})$ 
10            então  $CP_{viz} \leftarrow CP_{viz} \cup \{s\}$ ;
11    para cada  $s \in CP_{viz}$ 
12    faça se  $NÃODOMINADA(s, CP_{mel})$ 
13        então  $CP_{mel} \leftarrow CP_{mel} \cup \{s\}$ ;
14     $it \leftarrow it + 1$ ;
15 retorna  $CP_{mel}$ ;

```

Figura 6.8: Pseudo-código da busca local multiobjetivo.

Nos passos 1 e 2 do algoritmo da Figura 6.8 são realizadas as inicializações do conjunto de soluções eficientes corrente (CP_{mel}) e do contador de iterações (it). O laço entre os passos 3 e 14 gera as vizinhanças para as N_{sol} soluções e verifica quais delas são não-dominadas, armazenando-as no conjunto da iteração corrente (CP_{viz}) e, a seguir, no conjunto de soluções eficientes da busca (CP_{mel}). No passo 4 são obtidas as N_{sol} soluções mais representativas do conjunto CP_{mel} através de um procedimento de *clustering* descrito com detalhes na Seção 6.2.1. A seguir, nos passos 6-10, são geradas as vizinhanças para as N_{sol} soluções (passo 7), armazenando as soluções não-dominadas no conjunto de soluções eficientes da iteração corrente CP_{viz} (passo 10). Após a geração das vizinhanças, verificam-se quais soluções pertencentes ao conjunto CP_{viz} são não-dominadas em relação ao conjunto CP_{mel} (passos 11-13). Por fim, armazenam-se todas as soluções eficientes vizinhas (CP_{viz}) no conjunto CP_{mel} (passo 13), sendo este conjunto a nova aproximação

do conjunto de Pareto que será utilizada pela busca na próxima iteração. Com isso, são mantidas todas as soluções que permanecerem não-dominadas desde o início até o final do algoritmo, favorecendo a diversidade e possibilitando a escolha de novas soluções eficientes.

As seções 6.2.1 e 6.2.2 a seguir apresentam a forma de redução de um conjunto (função “Reduz”, passo 4 da Figura 6.8) e a vizinhança desenvolvida (função “GeraVizinhança”, passo 7 da Figura 6.8), respectivamente.

6.2.1 Redução por agrupamento (*clustering*)

Como é muito custoso realizar a busca local para todos os elementos do conjunto aproximado de Pareto (CAP), necessita-se obter um subconjunto desses elementos que sejam representativos, ou seja, elementos bem distribuídos no CAP.

No caso específico do algoritmo de busca local, quando se quer obter um número N_{sol} de elementos de um conjunto (CP) é preciso utilizar algum procedimento que permita extraí-los. Podem-se usar vários critérios, tais como extrair os primeiros N_{sol} elementos ou até mesmo sortear N_{sol} elementos do conjunto.

O problema torna-se mais complicado quando se quer obter N_{sol} elementos representativos com relação a alguma medida de qualidade. No caso em questão, essa medida será definida através das funções objetivo de cada solução do conjunto CP . Esse problema pode ser entendido como um problema de agrupamento, que corresponde a particionar um conjunto com m elementos em n grupos, sendo $n < m$. Bern e Eppstein (1996) demonstram que problemas de agrupamento são NP -difíceis quando o número de agrupamentos é variável.

O algoritmo escolhido para resolver o problema de particionar o conjunto CP em N_{sol} conjuntos foi desenvolvido por Morse (1980), que demonstrou a eficiência deste algoritmo justamente para redução de conjuntos aproximados de Pareto em otimização multiobjetivo. O pseudo-código da Figura 6.9 apresenta este algoritmo com detalhes, que realiza o particionamento do conjunto CP em N_{sol} grupos, escolhendo-se o centróide, ou o elemento mais representativo, de cada um deles. A medida de distância utilizada neste algoritmo, também empregada por Zitzler e Thiele (1999), é simbolizada por $||.||$ e denota a distância euclidiana entre quaisquer dois elementos do conjunto CP . Este conjunto, por sua vez, é formado por soluções nas quais a medida de qualidade é composta pelos valores que as mesmas apresentam para as funções objetivo do PRS.

No passo 1 do algoritmo da Figura 6.9 é feita uma cópia do conjunto CP para CP' . A partir deste último é criado um conjunto C contendo $|CP'|$ subconjuntos (onde $|CP'|$

```

REDUZ( $N_{sol}, CP$ )
1   $CP' \leftarrow CP$ ;
2   $C \leftarrow \bigcup_i c_i, c_i = \{i\} \forall i \in CP'$ ;
3  enquanto  $|C| > N_{sol}$ 
4  faça  $d_{min} \leftarrow min_i \leftarrow min_j \leftarrow 0$ ;
5      para cada  $c_i \subset C$ 
6          faça para cada  $c_j \subset C \mid j \neq i$ 
7              faça  $d_{c_i, c_j} \leftarrow \frac{1}{|c_i| \cdot |c_j|} \cdot \sum_{i_1 \in c_i, i_2 \in c_j} \|i_1 - i_2\|$ ;
8                  se  $min_i == 0$  ou  $d_{c_i, c_j} < d_{min}$ 
9                      então  $d_{min} \leftarrow d_{c_i, c_j}$ ;
10                          $min_i \leftarrow i; min_j \leftarrow j$ ;
11          $C \leftarrow C - \{c_{min_i}, c_{min_j}\} \cup \{c_{min_i} \cup c_{min_j}\}$ ;
12   $CP \leftarrow \emptyset$ ;
13  para cada  $c_i \subset C$ 
14  faça  $d_{min} \leftarrow min_k \leftarrow 0$ ;
15      para cada  $k \in c_i$ 
16          faça  $d_k \leftarrow \sum_{l \in c_i, l \neq k} \|l - k\|$ ;
17              se  $min_k == 0$  ou  $d_k < d_{min}$ 
18                  então  $d_{min} \leftarrow d_k$ ;
19                       $min_k \leftarrow k$ ;
20   $CP \leftarrow CP \cup \{min_k\}$ ;
21  retorna  $CP' - CP$ ;

```

Figura 6.9: Pseudo-código do método de redução por agrupamento.

denota a cardinalidade do conjunto CP'), cada um incluindo um elemento de CP' (passo 2). O laço 3-11 se encarrega de reduzir o número dessas partições através do agrupamento de algumas delas, até que a cardinalidade de C seja igual a N_{sol} (que corresponde ao número desejado de partições). Para tal agrupamento determina-se a distância mínima entre quaisquer dois pares de elementos de C (passos 4-7), sendo c_{min_i} e c_{min_j} os pares escolhidos. No passo 7 calcula-se a distância d_{c_i, c_j} entre os agrupamentos c_i e c_j , a qual é composta pela distância euclidiana entre quaisquer dois elementos, um pertencente a c_i e o outro a c_j , ponderada pelo produto da cardinalidade dos dois agrupamentos. A seguir, excluem-se do conjunto C os dois agrupamentos que ocasionaram a menor distância (c_{min_i} e c_{min_j}), adiciona-se outro subconjunto resultante da união destes dois, passo 11.

Após criado o número desejado de partições, parte-se para a determinação do centróide de cada uma delas (passos 12-20). No passo 12, o conjunto CP é inicializado vazio, o laço que segue, passos 13-20, determina o centróide de cada partição. Para tanto toma-se um determinado elemento do conjunto c_i (passo 15) e verifica-se a distância entre ele e todos os demais (passo 16), escolhendo-se aquele que apresenta a menor (passos 17-19). Depois, inclui-se esse centróide (min_k) no conjunto CP , passo 20. Por fim, o conjunto CP armazena os centróides escolhidos e o conjunto dos elementos restantes é retornado.

A Figura 6.10 apresenta o resultado da redução do conjunto de soluções eficientes de 11 para 4 elementos representativos, sendo estes últimos os pontos cheios na figura. Percebe-se que estes 4 elementos estão bem distribuídos, sendo verdadeiramente representativos da fronteira original.

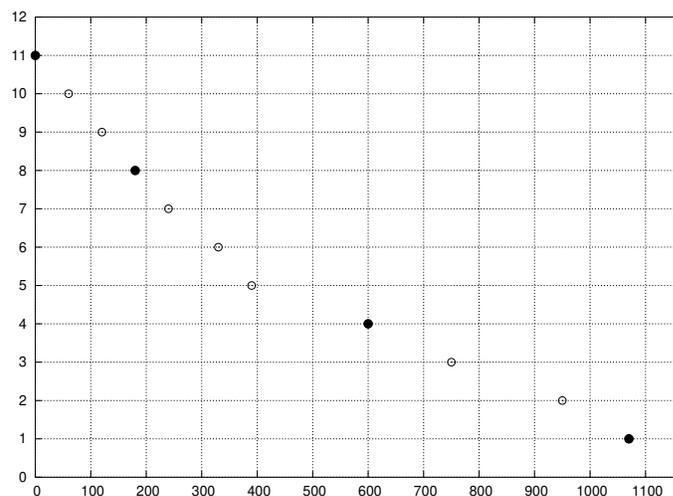


Figura 6.10: Resultado da redução de um conjunto pelo método de Morse (1980).

6.2.2 Vizinhança desenvolvida

O método de busca local, apresentado na Figura 6.8, faz uso de uma vizinhança desenvolvida especificamente para o PRS multiobjetivo.

Através do conceito de nó-fonte, introduzido na Seção 4.1 e definido na seção 6.1, define-se uma operação que perturba uma dada solução, gerando várias soluções vizinhas. Essa operação, também conhecida como mecanismo de geração de vizinhança, faz uso de todos os nós pertencentes à área escura e, para cada um deles, é verificado quais são todos os caminhos distintos possíveis que conduzem até os nós-fonte originais, os quais deverão obrigatoriamente pertencer à área clara original. A seguir, busca-se conectar este nó da área escura através dos caminhos determinados. Logo, é possível obter soluções topologicamente diferentes que tenham grande possibilidade de influenciar os critérios de otimização estabelecidos (mínima carga não restaurada e mínimo chaveamento).

Para um melhor entendimento, apresenta-se na Figura 6.11 as redes pré (Figura 6.11(a)) e pós-falha (Figura 6.11(b)), bem como os grafos com todos os caminhos possíveis para cada nó da área escura (Figuras 6.11(c)-6.11(g)). A partir dos grafos das Figuras 6.11(c)-6.11(g) monta-se a Tabela 6.1, que relaciona todos os caminhos possíveis de todos os nós da área escura. Observa-se, por exemplo, que somente para o nó 9 é possível interligá-lo à área clara através de 6 caminhos distintos, 4 deles terminando no nó-fonte 4 e os outros dois terminando no nó-fonte 6. Com isso, a vizinhança da solução representada na Figura 6.11(b) contém 24 soluções, cada uma correspondente a um caminho apresentado na Tabela 6.1. Embora represente um custo computacional razoável para a determinação de todos estes caminhos, eles são calculados apenas uma vez, na inicialização do método de busca local.

A Figura 6.12(a) apresenta uma solução x , (obtida, por exemplo, por um dos algoritmos construtivos) com os nós 7, 8 e 10 interligados à área clara. As soluções vizinhas desta solução correspondem àquelas obtidas através da escolha de cada um dos caminhos apresentados na Tabela 6.1. A título de exemplo e para simplificar, considere apenas os caminhos oriundos do nó 7 e que a solução da Figura 6.12(a) já esteja utilizando o caminho 7-4. As soluções obtidas são apresentadas nas Figuras 6.12(b)-6.12(e). Observa-se nessas soluções que o nó 8 acompanhou a alteração de caminho com o nó 7, justamente porque está vinculado a este. Além disso, nas soluções 6.12(d) e 6.12(e) o nó 10 teve o seu nó-fonte alterado, justamente porque está incluído nos caminhos 7-10-6 e 7-9-10-6.

Com estas alterações sistemáticas de caminhos entre os nós da área escura e os nós-fonte, busca-se gerar todas as soluções possíveis, na esperança de encontrar soluções não-dominadas. Devido à grande diversidade topológica proporcionada pelas soluções

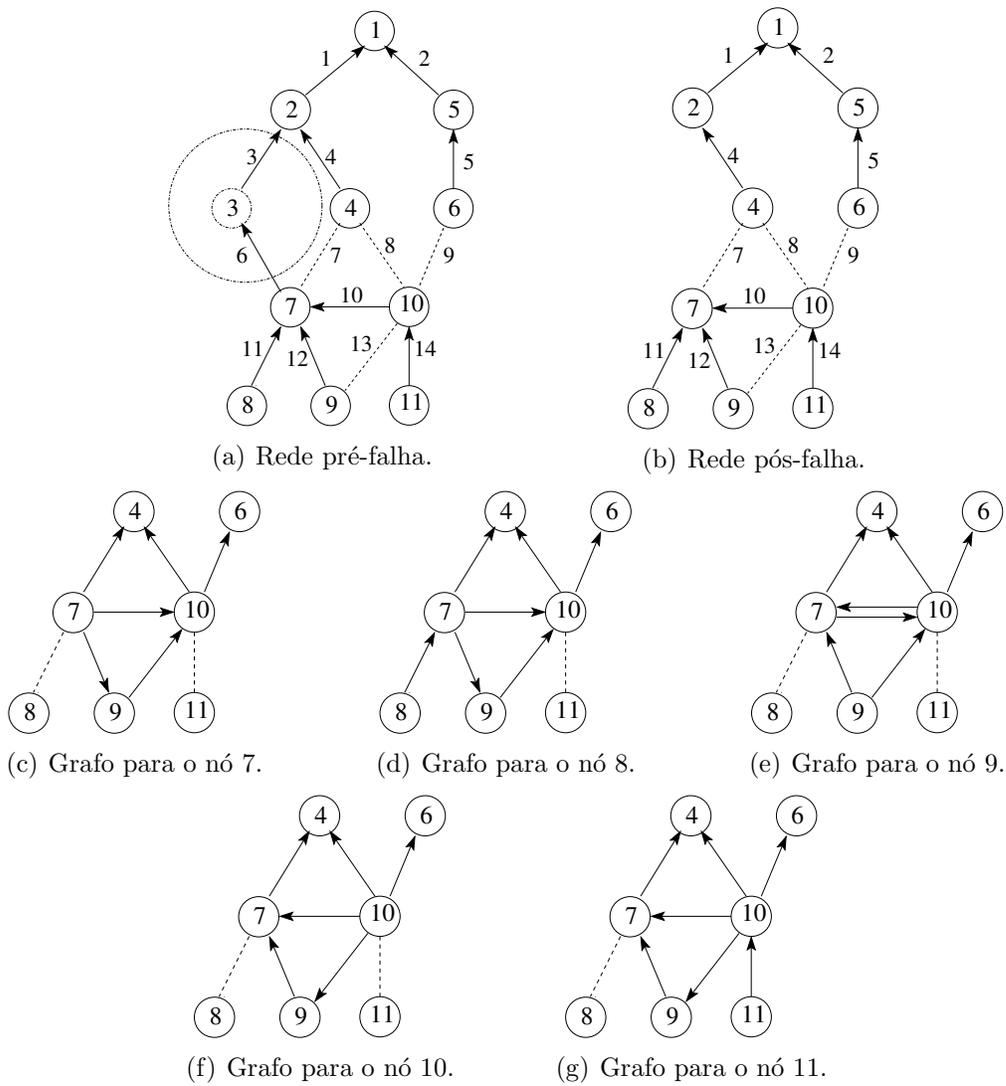
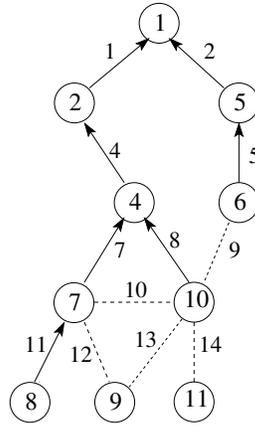


Figura 6.11: Vizinhaça para o PRS².

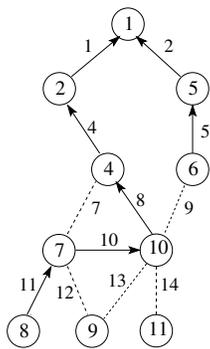
Nº caminho	Nó da área escura	Nó-fonte	Caminho
1	7	4	7-4
2	7	4	7-10-4
3	7	4	7-9-10-4
4	7	6	7-10-6
5	7	6	7-9-10-6
6	8	4	8-7-4
7	8	4	8-7-10-4
8	8	4	8-7-9-10-4
9	8	6	8-7-10-6
10	8	6	8-7-9-10-6
11	9	4	9-7-4
12	9	4	9-7-10-4
13	9	4	9-10-4
14	9	4	9-10-7-4
15	9	6	9-7-10-6
16	9	6	9-10-6
17	10	4	10-4
18	10	4	10-7-4
19	10	4	10-9-7-4
20	10	6	10-6
21	11	4	11-10-4
22	11	4	11-10-7-4
23	11	4	11-10-9-7-4
24	11	6	11-10-6

Tabela 6.1: Caminhos de interligação entre todos os nós da área escura e todos os nós-fonte, obtidos dos grafos das Figuras 6.11(c)-6.11(g).

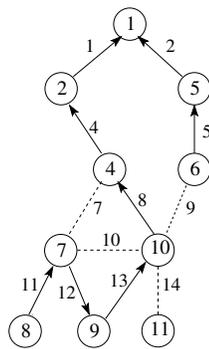
vizinhas, espera-se ainda cobrir uma boa parte do espaço objetivo.



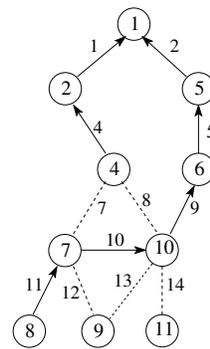
(a) Solução x .



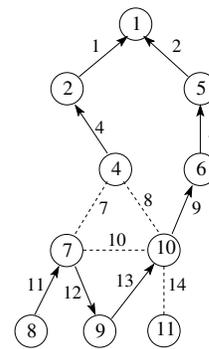
(b) Solução com o caminho 2.



(c) Solução com o caminho 3.



(d) Solução com o caminho 4.



(e) Solução com o caminho 5.

Figura 6.12: Soluções vizinhas da solução x com a utilização dos caminhos oriundos do nó 7 até os nós-fonte alcançáveis², definidos na Tabela 6.1.

Capítulo 7

Algoritmo evolutivo multiobjetivo para o PRS

Depois da análise de todas as abordagens discutidas na Seção 3.4.3, é possível observar um consenso quanto às características desejáveis de um AEM:

1. Eficiência computacional quanto ao tempo gasto;
2. Boa diversidade e garantia de convergência;
3. Baixa sensibilidade aos parâmetros.

Quanto ao primeiro aspecto, deseja-se ter um algoritmo que possua etapas de avaliação e seleção com baixa complexidade computacional. Em geral, o tempo computacional de uma geração de um algoritmo evolutivo multiobjetivo é dominado pelo tempo gasto nestas duas etapas, ou seja, elas têm uma contribuição majoritária no tempo final do algoritmo.

O segundo aspecto desejável é a manutenção da diversidade e a garantia de convergência: o algoritmo deve fornecer o conjunto Pareto-ótimo quando o número de gerações e os recursos de memória são infinitos (Zitzler et al. 2004). Na prática, como recursos de memória são limitados bem como o tempo computacional, o conjunto de soluções não-dominadas obtido ao final da execução do algoritmo será apenas uma aproximação do conjunto Pareto-ótimo. Entretanto, não é desejável que a limitação devido ao projeto do algoritmo acabe por comprometer a convergência quando o tempo e a memória tendam ao infinito.

Quanto à diversidade, é desejável terminar o algoritmo com um conjunto de soluções não-dominadas que recubra o mais uniformemente possível o espaço multiobjetivo, de

modo que o decisor possa dispor de um elenco de soluções alternativas que não limite significativamente a sua decisão final.

A terceira e última característica desejável de um AEM refere-se à sensibilidade aos parâmetros. Ainda que os parâmetros mantenham alguma relação com o desempenho final do algoritmo, não é oportuno que esta relação seja muito estrita a ponto de inviabilizar o ajuste dos mesmos.

Diante destes requisitos, o paradigma defendido pelo SPEA2 (Zitzler et al. 2002) é muito consistente justamente pela combinação de várias estratégias interessantes:

- **Elitismo:** todas as soluções não-dominadas são mantidas durante o processo evolutivo;
- **Avaliação:** o critério utilizado para atribuir um valor de adaptação aos indivíduos baseia-se na relação de dominância de Pareto combinada com uma função para estimar a densidade populacional. Procura-se, com isso, melhorar a discriminação entre indivíduos que apresentam o mesmo número de soluções dominadas;
- **Diversidade:** caso o número de soluções não-dominadas seja inferior ao tamanho máximo da população, soluções dominadas são incluídas de forma a proporcionar maior diversidade.

A versão do SPEA2 desenvolvida para o PRS parte de uma aproximação conjunto de Pareto, fornecida pelo algoritmo construtivo descrito na Seção 6.1, e recombina estas configurações na busca por soluções que se aproximem da fronteira ideal de Pareto. Outro detalhe importante refere-se à representação adotada: ela inclui uma codificação baseada nos caminhos de interligação entre cada nó da área escura e os respectivos nós-fonte alcançáveis. Com isso, torna-se necessário o uso de uma função de decodificação para converter estes caminhos em uma solução factível para o PRS.

O algoritmo da Figura 7.1 apresenta o processo evolutivo baseado no SPEA2 desenvolvido para o PRS. Os quatro parâmetros do algoritmo correspondem ao tamanho da população interna ($T_{P_{ger}}$), que neste caso será igual ao número de indivíduos gerados a cada geração (P_{ger}); ao tamanho da população externa (T_{P_e}); à taxa de mutação (Tx_{alt}), relativa ao percentual de indivíduos da população que sofrerá uma perturbação aleatória; e ao número de gerações do algoritmo (N_{ger}).

No passo 1 é gerada a população inicial usando o algoritmo construtivo da Seção 6.1. A seguir, no passo 2, são realizadas as inicializações do contador de gerações (ger) e da população externa (P_e). O laço entre os passos 3 e 11 corresponde às gerações do algoritmo e é repetido até que o número máximo de gerações seja alcançado. A avaliação das soluções

é realizada no passo 4, sendo que é avaliado o conjunto de indivíduos obtidos pela união da população interna (P_{ger}) e da população externa (P_e). O objetivo é proporcionar maior discriminação entre os indivíduos avaliados, visto que a dominância é um dos critérios utilizados. No passo 5 determina-se a nova população externa, na qual são inseridos os indivíduos não-dominados de P_{ger} e P_e . Caso o conjunto resultante P_e seja maior que o tamanho máximo estabelecido (T_{P_e}), aplica-se um procedimento seletivo para reduzir este conjunto ao tamanho T_{P_e} (passos 6 e 7).

Depois da avaliação dos indivíduos, procede-se com a fase de seleção, recombinação e mutação. Na primeira, passo 8 do algoritmo da Figura 7.1, seleciona-se um conjunto de $2.T_{P_{ger}}$ pais para dar origem a $T_{P_{ger}}$ filhos. A seguir, no passo 9, aplica-se o operador de recombinação para gerar a nova população, a qual sofre mutação no passo 10. Como última operação do laço de gerações, o contador ger é incrementado no passo 11. Finalmente no passo 12, a população externa é devolvida como a melhor aproximação obtida pelo processo evolutivo.

```

ALGORITMOEVOLUTIVO_PRS( $T_{P_{ger}}, T_{P_e}, Tx_{alt}, N_{ger}$ )
1   $P_{ger} \leftarrow \text{GERAPOPINICIAL}();$ 
2   $ger \leftarrow 0; P_e \leftarrow \emptyset$ 
3  enquanto  $ger \leq N_{ger}$ 
4  faça  $\text{AVALIA}(P_{ger}, P_e);$ 
5      $P_e \leftarrow \text{SELECIONA}(P_{ger}, P_e, T_{P_e});$ 
6     se  $|P_e| \geq T_{P_e}$ 
7         então  $P_e \leftarrow \text{REDUZ}(P_e, T_{P_e});$ 
8      $PAIS \leftarrow \text{SELECIONAPAIS}(P_e, 2.T_{P_{ger}});$ 
9      $P_{ger} \leftarrow \text{RECOMBINA}(PAIS);$ 
10     $P_{ger} \leftarrow \text{ALTERA}(P_{ger}, Tx_{alt});$ 
11     $ger \leftarrow ger + 1;$ 
12  retorna  $P_e$  ;

```

Figura 7.1: Algoritmo evolutivo desenvolvido para o PRS.

Nas seções a seguir são detalhadas a representação dos indivíduos, a função de adaptação, o mecanismo de seleção, os operadores de recombinação e mutação, e a redução da população.

7.1 Representação dos indivíduos

Em algoritmos evolutivos, a solução para um mesmo problema pode ser representada de diferentes formas, no entanto a eficiência do processo evolutivo pode ser maior com

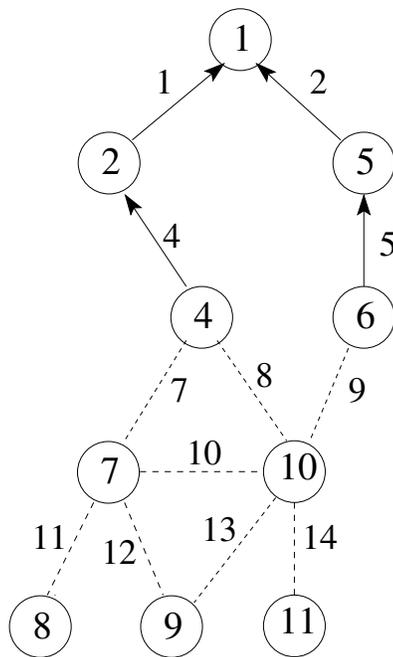
uma representação específica. Isso se deve justamente pela relação da representação com os operadores evolutivos, como os de recombinação e de mutação. Por isso, torna-se imprescindível a escolha de uma representação que seja adequadamente combinada com estes operadores no sentido de produzir preferencialmente soluções factíveis. Além disso, a representação escolhida deve ser capaz de codificar qualquer elemento do espaço de solução do problema.

Para o PRS, considerando a abordagem desenvolvida, descrita no Capítulo 4, vislumbrou-se uma codificação baseada no caminho de interligação entre cada nó da área escura e o respectivo nó-fonte correspondente. O indivíduo é representado, portanto, por um cromossomo de tamanho n , onde n é o número de nós pertencentes à área escura. Cada alelo do cromossomo representa o número do caminho que interliga um determinado nó da área escura até um nó-fonte incluído no respectivo caminho. No entanto, torna-se necessária uma etapa de decodificação para a interpretação do cromossomo e a construção da rede baseada nos caminhos fornecidos. Ao final é obtida uma solução representada da mesma forma que a solução descrita na Seção 5.2.2. Assim, tanto a representação por caminhos de interligação como a função de decodificação são capazes de gerar qualquer configuração de rede para o PRS.

Para melhor entendimento, ilustra-se na Figura 7.2 a configuração de rede pós-falha (Figura 7.2(a)) e todos os caminhos possíveis de interligação entre os nós da área escura e os respectivos nós-fonte alcançáveis (Figura 7.2(b)). A representação da Figura 7.3(a) relaciona os caminhos entre os nós da área escura e os nós-fonte alcançáveis, com a posterior decodificação para montar a configuração de rede resultante, apresentada na Figura 7.3(f), que corresponde a uma possível solução para o PRS. Em cada posição do cromossomo apresentado na Figura 7.3(a) é armazenado o número de um caminho que interliga um nó da área escura a um nó-fonte. Neste exemplo, são utilizados os caminhos 14, 2, 8, 26 e 24 da Figura 7.2(b) para interligar o nó 9 ao nó-fonte 4 (Figura 7.3(b)), o nó 7 ao nó-fonte 4 (Figura 7.3(c)), o nó 8 ao nó-fonte 4 (Figura 7.3(d)), o nó 11 ao nó-fonte 4 (Figura 7.3(e)) e o nó 10 ao nó-fonte 6 (Figura 7.3(f)), respectivamente.

Os caminhos 1, 7, 13, 20 e 25 representam a simples desconexão dos respectivos nós da área escura de seus nós-fonte atuais. Como exemplo, considerando o cromossomo apresentado na Figura 7.3(a) e a alteração do caminho 14 para o caminho 13, na mesma posição 1 (cromossomo da Figura 7.4(a)), a solução final ocasionará a permanência do nó 9 na área escura, como ilustra a Figura 7.4(b).

¹As linhas contínuas correspondem às chaves fechadas e as linhas pontilhadas correspondem às chaves abertas; a direção do arco indica o caminho até a barra da subestação; os nós correspondem às barras de carga.



(a) Rede pós-falha.

Nº caminho	Nó da área escura	Nó-fonte	Caminho
1	7	0	7
2	7	4	7-4
3	7	4	7-10-4
4	7	4	7-9-10-4
5	7	6	7-10-6
6	7	6	7-9-10-6
7	8	0	8
8	8	4	8-7-4
9	8	4	8-7-10-4
10	8	4	8-7-9-10-4
11	8	6	8-7-10-6
12	8	6	8-7-9-10-6
13	9	0	9
14	9	4	9-7-4
15	9	4	9-7-10-4
16	9	4	9-10-4
17	9	4	9-10-7-4
18	9	6	9-7-10-6
19	9	6	9-10-6
20	10	0	10
21	10	4	10-4
22	10	4	10-7-4
23	10	4	10-9-7-4
24	10	6	10-6
25	11	0	11
26	11	4	11-10-4
27	11	4	11-10-7-4
28	11	4	11-10-9-7-4
29	11	6	11-10-6

(b) Caminhos.

Figura 7.2: Rede pós-falha¹ com os caminhos de interligação entre os nós da área escura e os respectivos nós-fonte alcançáveis.

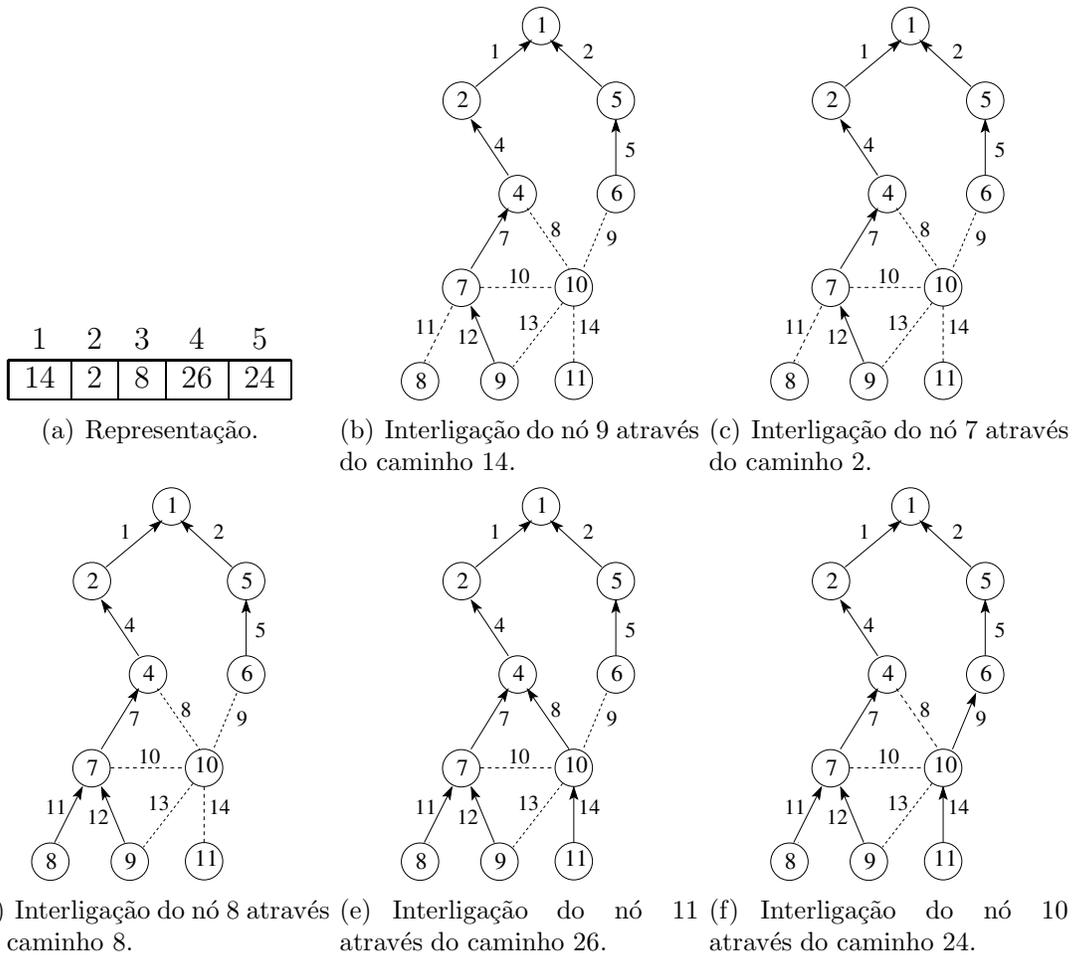


Figura 7.3: Representação de um indivíduo no AE desenvolvido para o PRS e as etapas de decodificação¹.

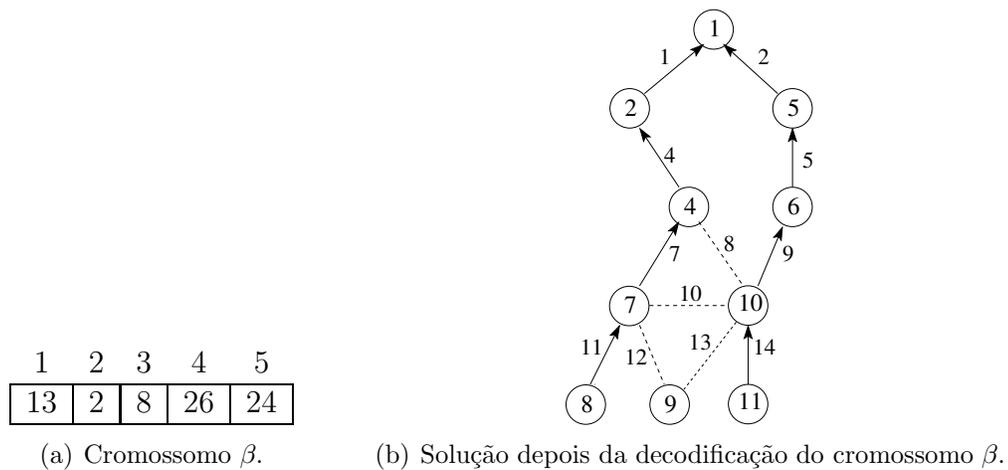


Figura 7.4: Representação da permanência do nó 9 na área escura¹.

	Nós									
	1	2	4	5	6	7	8	9	10	11
$P(i)$	2	4	7	6	10	8	9	5	11	0
$V(i)$	0	1	2	1	5	4	7	7	6	10
$N(i)$	0	1	2	1	2	3	4	4	3	4
$A(i)$	1	1	1	1	1	1	1	1	1	1
	Árvores									
	1									
$R(a)$	1									

Figura 7.5: Estrutura de dados para a solução da Figura 7.3(f), obtida depois da decodificação do indivíduo representado pelo cromossomo da Figura 7.3(a).

Com a solução final obtida ao final do processo de decodificação, representada na Figura 7.3(f), monta-se a estrutura correspondente conforme apresenta a Figura 7.5. Percebe-se que esta decodificação mantém radialidade da área clara, ou seja, a interligação dos nós da área escura, através dos caminhos incluídos no cromossomo do indivíduo, não ocasiona ciclos. Entretanto, as demais restrições referentes às quedas de tensão e aos limites dos equipamentos (condutores e transformadores) devem ser verificadas com a execução do fluxo de carga escolhido. Caso não haja violações, o indivíduo é então avaliado com a função de adaptação definida no algoritmo evolutivo e descrita na próxima seção.

Vale observar que o nó 10 estava originalmente interligado ao nó-fonte 4 porque ele estava no caminho de interligação do nó 11 ao nó-fonte 4. Entretanto, como o último caminho do cromossomo refere-se à interligação do nó 10 ao nó-fonte 6, houve uma troca de nó-fonte para o nó 10 juntamente com os seus sucessores, neste caso somente o nó 11. Na verdade estas operações de interligação de nós são exatamente iguais àquelas realizadas pela busca local descrita na Seção 6.2.

Outra importante observação refere-se à ordem na qual os caminhos são construídos para interligar os nós da área escura à área clara. Ao longo do processo evolutivo, com as alterações ocasionadas pelos operadores de recombinação e mutação, esta ordem é alterada e proporcionará configurações topologicamente distintas. Um bom exemplo disso são os cromossomos apresentados nas Figuras 7.6(a) e 7.6(b). A única diferença entre eles é que no primeiro, os caminhos 2 e 18 encontram-se nas posições 2 e 3, respectivamente, e no segundo, o caminho 2 passa para a posição 3 e o caminho 18 passa para a posição 2. Pelas Figuras 7.6(c) e 7.6(d), que ilustram as soluções obtidas através da decodificação dos cromossomos apresentados nas Figuras 7.6(a) e 7.6(b), respectivamente, percebe-se que apenas a alteração de ordem na construção dos caminhos 2 e 18 proporcionou uma solução topologicamente distinta da anterior. Dessa forma, mantém-se o compromisso de gerar todas as soluções possíveis do PRS definido neste trabalho.

7.2 Função de adaptação

A função de adaptação desenvolvida para o AE da Figura 7.1 considera, primeiramente, a relação de dominância dos indivíduos da população. A seguir emprega-se uma medida de distância para avaliar a densidade dos indivíduos sobre o espaço objetivo. Assim, promove-se uma melhor discriminação, inclusive entre os indivíduos que apresentam a mesma classificação obtida através da relação de dominância. Todo o procedimento descrito a seguir está incluído no passo 4, função $AVALIA(P_{ger}, P_e)$, do algoritmo da

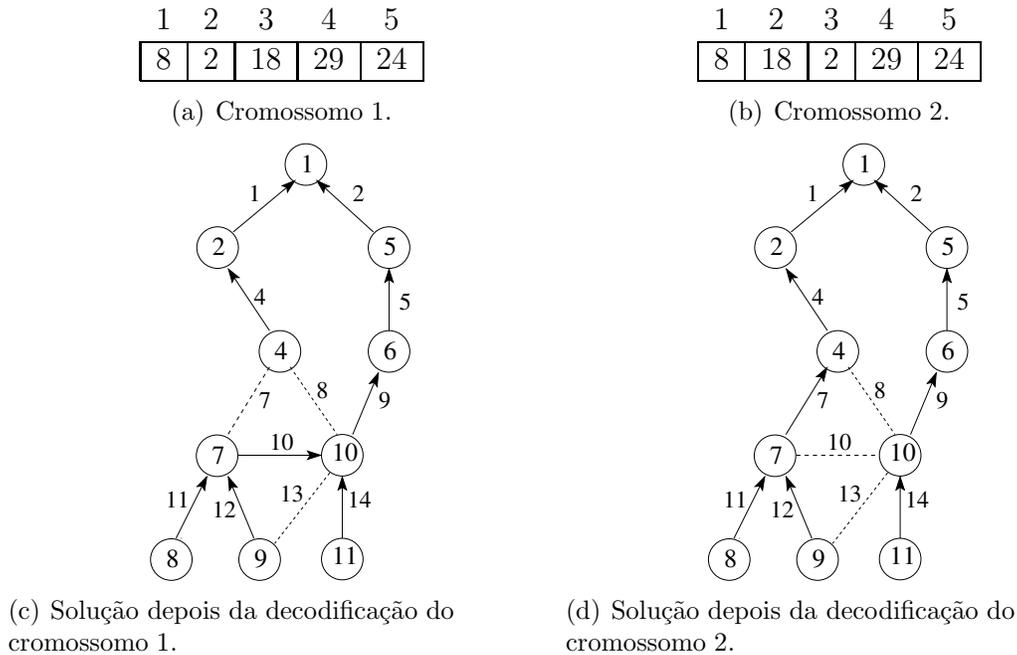


Figura 7.6: Comparação entre soluções quando a seqüência de caminhos nos respectivos cromossomos é alterada¹.

Figura 7.1, e foi baseado nas definições do SPEA2 incluídas em Zitzler et al. (2004).

Primeiramente calcula-se a relevância de cada indivíduo quanto ao número de outros indivíduos que o mesmo domina (Equação 7.1). O valor de F_{a_1} para um dado indivíduo i corresponde ao número de elementos dominados por i no conjunto formado pela união entre a população interna P_{ger} e a população externa P_e . A expressão $i \succ j$ denota a relação de dominância do elemento i sobre o elemento j , conforme definido na Seção 3.1.1.

$$F_{a_1}(i) = |\{j | (j \in P_{ger} \cup P_e) \wedge (i \succ j)\}| \quad (7.1)$$

A seguir calcula-se outro índice que corresponde à relevância, quanto à dominância, dos elementos que dominam um certo indivíduo i . Na Equação 7.2 é introduzida a função F_{a_2} que denota este índice, fazendo uso da função anterior F_{a_1} para todos os elementos j incluídos no conjunto formado pela união entre a população interna P_{ger} e a população externa P_e e que dominam um certo indivíduo i .

$$F_{a_2}(i) = \sum_{j \in P_{ger} \cup P_e, j \succ i} F_{a_1}(j) \quad (7.2)$$

Convém salientar que se busca a minimização do valor de F_{a_2} , o que significa que $F_{a_2}(i) = 0$ para todos os indivíduos i não-dominados. Um valor grande de $F_{a_2}(i)$ denota

a dominância de i por vários outros indivíduos do conjunto $P_{ger} \cup P_e$.

Especialmente quando o conjunto $P_{ger} \cup P_e$ é formado por indivíduos não-dominados, a função F_{a_2} pode gerar valores idênticos para todos eles. Para evitar tal anomalia e facilitar a discriminação destes indivíduos, adota-se, como sugere Zitzler et al. (2004), uma função para estimar a densidade populacional. Esta função é calculada a partir de cada indivíduo do conjunto $P_{ger} \cup P_e$ e corresponde à distância entre este indivíduo i até o seu k -ésimo vizinho mais próximo. Em geral, adotam-se valores pequenos para k , tipicamente 1 ou 2. A Equação 7.3 define esta terceira função, onde σ_i^k é a distância euclidiana entre o indivíduo i e o seu k -ésimo vizinho mais próximo. Com o inverso desta distância, privilegia-se aqueles indivíduos mais distantes do k -ésimo vizinho, facilitando a melhor distribuição dos indivíduos sobre o espaço objetivo.

$$D(i) = \frac{1}{\sigma_i^k + 2} \quad (7.3)$$

Finalmente é possível obter a função que define a adaptação propriamente dita de um indivíduo i ($F_a(i)$), descrita pela Equação 7.4.

$$F_a(i) = F_{a_2}(i) + D(i) \quad (7.4)$$

Com esta equação busca-se não uma maior adaptação do indivíduo, mas sim uma menor, ao contrário do que afirma Darwin.

7.3 Mecanismo de seleção

A seleção dos indivíduos ocorre em dois momentos no algoritmo da Figura 7.1. No primeiro, passo 5, busca-se atualizar a população externa com os indivíduos da população interna. No segundo, passo 8, é obtido o conjunto de indivíduos pais que darão origem à próxima população.

A atualização da população externa P_e , no passo 5 do algoritmo da Figura 7.1, compreende a seleção de todos os indivíduos i tais que $F_a(i) < 1$, ou seja, de todos os indivíduos não-dominados, considerando uma população formada pela união de P_{ger} e P_e . Todos estes indivíduos selecionados formarão a nova população externa P_{e+1} , como define a Equação 7.5. Com isso promove-se o elitismo, característica desejável que corresponde à salvaguarda dos melhores indivíduos da população (aqueles não-dominados) durante o processo evolutivo.

$$P_{e+1} = \{ i \mid (i \in P_{ger} \cup P_e) \wedge (F_a(i) < 1) \} \quad (7.5)$$

Caso a cardinalidade de P_{e+1} ($|P_{e+1}|$) seja menor que o tamanho estabelecido (T_{P_e}), os melhores $T_{P_e} - |P_{e+1}|$ indivíduos com $F_a(i) \geq 1$ completam o conjunto P_{e+1} . Com esta estratégia, proporciona-se diversidade na população externa P_e , justamente porque será deste conjunto que serão obtidos os indivíduos pais que darão origem à população interna da próxima geração.

No passo 8 é gerado um conjunto com cardinalidade $2.T_{P_{ger}}$ incluindo elementos da população externa P_e . Por meio de um torneio com reposição são escolhidos os $2.T_{P_{ger}}$ indivíduos não necessariamente distintos. Os elementos participantes do torneio são selecionados de forma aleatória da população P_e .

7.4 Operador de recombinação

A etapa de recombinação é responsável pela busca de soluções no espaço objetivo. O operador de recombinação desenvolvido para o PRS, mencionado no passo 9 da Figura 7.1, combina duas soluções para a geração de uma terceira, que será sempre factível quanto à restrição de radialidade. Restará, entretanto, a verificação das restrições elétricas, a cargo do algoritmo de fluxo de carga.

Cada par de elementos do conjunto *PAIS* dará origem a um indivíduo filho, através da combinação entre os cromossomos de cada um dos pais. O operador escolhido é uma modificação do *crossover* uniforme, de tal forma que um e somente um caminho para cada nó da área escura é incluído no cromossomo filho. Esta garantia é de particular importância para preservar a informação genética do indivíduo, do contrário, caso mais de um caminho fosse escolhido, poderiam ser gerados cromossomos com vários caminhos de interligação para somente um único nó da área escura, deixando os demais sem representação no cromossomo do indivíduo.

Um exemplo ilustrativo deste processo de recombinação é apresentado na Figura 7.7, considerando como rede pós-falha² a solução apresentada na Figura 7.2(a) e os respectivos caminhos da Figura 7.2(b). Os dois indivíduos pais são apresentados com seus respectivos cromossomos pelas Figuras 7.7(a)-7.7(d). O cromossomo do indivíduo filho e sua respectiva solução são apresentados pelas Figuras 7.7(e) e 7.7(f), respectivamente.

O processo de recombinação dos dois pais para gerar um filho tem início com a criação de um cromossomo vazio com número de genes igual ao número de nós da área escura. A seguir, do primeiro gene até o último, sorteia-se com igual probabilidade o pai que contribuirá com o seu respectivo alelo para o indivíduo filho. Dessa forma, como ilustra a

²Conforme dito na Seção 7.1, a decodificação do cromossomo sempre faz uso da rede pós-falha para construir a solução equivalente a partir da informação genética do indivíduo.

Figura 7.7(e), o Pai 1 insere o seu alelo (4) localizado no gene 1 para o gene do Filho. A seguir, continuando o sorteio, o Pai 2 passa o alelo (23) para o gene 2 do indivíduo Filho. Procede-se desta forma até que todos os genes estejam preenchidos.

Ao se tentar preencher a quarta posição do indivíduo Filho, verifica-se que um erro certamente ocorrerá: como cada nó da área escura deve ter um e somente um caminho de interligação à área clara no cromossomo, tanto o Pai 1 quanto o Pai 2 só podem contribuir com caminhos (22 e 5) que partem de nós de área escura já contemplados, a saber: o caminho 22 tem o mesmo nó da área escura que o caminho 23 (nó 10) e o caminho 5 tem o mesmo nó da área escura que o caminho 4 (nó 7). Com isso, o quarto gene permanece vazio até que o último gene seja preenchido; só aí são verificados quais nós da área escura não estão representados no cromossomo do Filho. Neste exemplo, o único nó da área escura que não é nó inicial de nenhum caminho é o nó 8, relacionado pelo caminho 12 do Pai 1 e pelo caminho 9 do Pai 2. Depois de realizado o sorteio, da mesma forma que para os demais genes, para decidir qual dos pais contribuirá para preencher o quarto gene do cromossomo do Filho, o Pai 1 foi o escolhido, gerando o cromossomo final 4-23-13-12-26.

Uma outra possibilidade de recombinação que evita o possível erro de repetição de nós da área escura no filho é conceber um cromossomo em que há uma associação rígida entre cada nó da área escura e o locus que este ocupa no cromossomo. Com isso, uma possibilidade de montar o cromossomo usando o exemplo da Figura 7.2 seria associar o gene 1 com o nó 7, o gene 2 com o nó 8, o gene 3 com o nó 9, o gene 4 com o nó 10 e o gene 5 com o nó 11, sendo que os respectivos alelos estariam restritos somente aos caminhos iniciando com cada um dos seus respectivos nós escuros. Ao se aplicar o operador uniforme de recombinação aos dois pais, não ocorre o erro de duplicação do nó escuro. Uma desvantagem aparente desse operador é a rigidez na montagem da solução a partir dos cromossomos, que seguiria sempre a mesma rígida ordenação.

Como visto, a ordem (ou a “desordem”) influencia positivamente na diversidade das soluções obtidas. Uma maneira de preservar essa diversidade é montar a solução dos filhos de modo aleatório, sorteando a ordem dos genes na seqüência de montagem. Com isso, garante-se ao operador de recombinação um amplo acesso a todo o espaço de soluções do problema.

7.5 Operador de mutação

Seguindo o paradigma darwiniano, para simular o erro estatístico no processo de cópia do material genético durante o processo evolutivo, aplica-se uma variação aleatória com

1	2	3	4	5
4	12	13	22	26

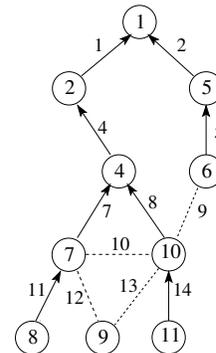
(a) Cromossomo do Pai 1.

1	2	3	4	5
9	23	13	5	29

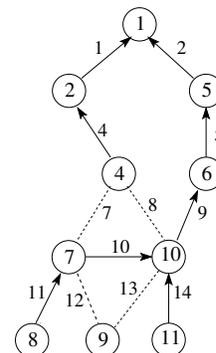
(c) Cromossomo do Pai 2.

P1				
4				
P1	P2			
4	23			
P1	P2	P2		
4	23	13		
P1	P2	P2	X	
4	23	13		
P1	P2	P2	X	P1
4	23	13		26
P1	P2	P2	P1	P1
4	23	13	12	26

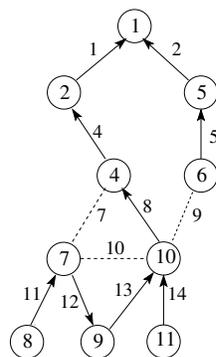
(e) Cromossomo do Filho.



(b) Solução do Pai 1.



(d) Solução do Pai 2.



(f) Solução do Filho.

Figura 7.7: Exemplo de aplicação do operador de recombinação¹.

uma certa probabilidade para alterar os genes dos cromossomos dos indivíduos filho selecionados.

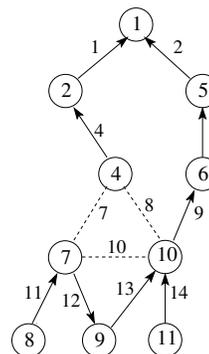
No AE desenvolvido para o PRS, o operador de mutação (passo 10 do algoritmo da Figura 7.1) escolhe uma parcela dos indivíduos da população, segundo a taxa de mutação estabelecida, para realizar a perturbação restrita a cada gene dos respectivos cromossomos. Depois de selecionados estes indivíduos, para cada um deles é percorrido todo o cromossomo, realizando-se a perturbação do alelo de cada gene com probabilidade de 50%. Ela corresponde à escolha aleatória de outro caminho distinto do atual para o nó inicial da área escura.

Como exemplo, ilustra-se na Figura 7.8 a alteração do cromossomo do indivíduo Filho da Figura 7.7(e). Novamente será utilizada a Figura 7.2(b), que relaciona todos os caminhos possíveis a partir da rede pós-falha da Figura 7.2(a) (com a área escura original).

Durante o processo, apenas os genes 1 e 5 foram alterados, como ilustra a Figura 7.8(a): o primeiro continha o caminho 4 para interligar o nó 7 à área clara e passou para o caminho 2; no segundo, gene 5, o caminho 26, que interligava o nó 11 à área clara, foi alterado para o caminho 29. A solução obtida depois da decodificação do indivíduo alterado é apresentada na Figura 7.8. Percebe-se que quando é escolhido um gene para sofrer mutação, a escolha dos caminhos é restrita ao nó inicial do caminho atual: quando o primeiro gene foi escolhido, os caminhos possíveis eram 1, 2, 3, 5 e 6, todos eles com o mesmo nó inicial do caminho 4.

1	2	3	4	5
2	23	13	12	29

(a) Cromossomo depois da mutação do indivíduo da Figura 7.7(e).



(b) Solução depois da mutação.

Figura 7.8: Exemplo de aplicação do operador de mutação¹.

7.6 Redução da população

Em se tratando de algoritmos para otimização multiobjetivo, o tamanho do conjunto para armazenar as soluções não-dominadas tem uma relação direta com a eficiência dos métodos nos quais a relação de dominância de Pareto é utilizada como critério de otimização. Uma solução somente será incluída no conjunto aproximado caso não seja dominada por nenhuma outra solução do conjunto. Para tanto, é preciso verificar a dominância para todas as soluções do conjunto, o que relaciona o tempo gasto pelo algoritmo ao tamanho deste conjunto.

Diante disso, manter este conjunto aproximado com as soluções não-dominadas obtidas com uma dimensão máxima restrita se torna imperativo. O mecanismo de redução da população externa P_e do AE desenvolvido, incluído no passo 7 do algoritmo da Figura 7.1, se baseia na distância do k -ésimo vizinho mais próximo (σ_i^k , apresentada na Seção 7.2) de cada indivíduo do conjunto P_e . Exclui-se iterativamente aquele indivíduo com a menor distância até este k -ésimo vizinho, enquanto o tamanho de P_e for maior que o tamanho máximo estabelecido T_{P_e} . Dito de outra forma e utilizando a definição de Zitzler et al. (2004), a cada iteração exclui-se o indivíduo i tal que $i \leq_d j$, para todos j indivíduos pertencentes à população atual P_e .

Na Equação 7.6 é apresentada formalmente a relação $i \leq_d j$ (Zitzler et al. 2002). Como os pontos na fronteira do espaço objetivo são mais distantes quando se considera a distância destes até os k vizinhos mais próximos, evita-se que estes pontos sejam eliminados na redução, desde que sejam escolhidos aqueles com a menor distância.

$$i \leq_d j \Leftrightarrow (\forall 0 < k < |P_e| : \sigma_i^k = \sigma_j^k) \vee (\exists 0 < k < |P_e| : [(\forall 0 < l < k : \sigma_i^l = \sigma_j^l) \wedge (\sigma_i^k = \sigma_j^k)]) \quad (7.6)$$

Capítulo 8

Busca Tabu multiobjetivo para o PRS

Como salientado nas Seções 2.1.3 e 3.4.2, que descrevem respectivamente as versões mono e multiobjetivo da metaheurística Busca Tabu, umas das potencialidades desta técnica reside na exploração sistemática do espaço de busca por meio da coordenação de um operador de busca local. A partir de uma solução atual x , move-se para uma solução vizinha pertencente à vizinhança de x , $N(x)$, e assim sucessivamente até que um critério de parada seja satisfeito. Quando é considerado o caso mono-objetivo, busca-se uma solução pertence a $N(x)$ que apresente um valor de função objetivo melhor que o da solução x . Entretanto, visando escapar de ótimos locais, aceitam-se, em situações restritas e bem específicas, soluções vizinhas que apresentem uma degradação no valor da função objetivo, na esperança de mais tarde se alcançar regiões mais promissoras.

O uso de uma memória adaptativa se faz necessário para armazenar um histórico das trajetórias descritas no espaço objetivo, as quais incluirão as soluções escolhidas de cada vizinhança avaliada. Combinando adequadamente estas informações com estratégias específicas, é possível descrever uma busca ampla e eficiente no espaço objetivo, caracterizando um bom compromisso entre exploração/exploração.

A memória adaptativa, neste caso, pode ser de dois tipos: *de curto* ou *de longo prazo*. A principal função da memória de curto prazo é evitar que regiões já exploradas tornem a ser visitadas, caracterizando uma ciclagem indesejada na trajetória da busca, o que pode comprometer a eficiência do processo de otimização. Ao mesmo tempo, verifica-se que esta proibição favorece o alcance de regiões pouco exploradas. Tal mecanismo é construído com o armazenamento de atributos de movimentos já realizados em uma *lista*, consultando-a toda vez que um novo *movimento* está para ser realizado. Este movimento somente será

escolhido depois que a sua permanência nesta lista tabu estiver expirada, permanência esta definida por um número especificado de iterações, também conhecido como *duração tabu*. Outra forma de aceitar um movimento que seja proibido é pelo critério de aspiração: um movimento proibido será realizado sempre que ele proporcionar uma solução com um valor de função objetivo melhor que a solução incumbente.

Embora esta memória de curto prazo já promova um bom aumento no poder de exploração do método baseado em busca local, há problemas nos quais certas regiões do espaço de busca podem não ser visitadas em função da ênfase dada ao aspecto agressivo de busca por melhoramento da função objetivo. Diante disso, torna-se oportuna a combinação de uma memória de *longo prazo* para buscar explicitamente regiões ainda não exploradas pela busca. Através do armazenamento de atributos referentes à *freqüência* de certos movimentos, serão incentivados aqueles cujas freqüências sejam baixas. Define-se, assim, um mecanismo que busca maior *diversidade* com a esperança de alcançar soluções de boa qualidade em regiões ainda não avaliadas. Outra utilização possível deste mecanismo de freqüência é a retomada da busca em regiões promissoras, identificadas a partir de atributos de *soluções de elite* previamente armazenadas.

Embora o uso de BT para problemas mono-objetivo seja bem difundido (Seção 2.1.3), o emprego desta técnica para problemas multiobjetivo ainda é pouco comum (Jones, Mirrazavi e Tamiz 2002). Os poucos métodos da literatura que empregam BT em problemas multiobjetivo o fazem com modificações da versão mono-objetivo, justamente pela facilidade em se aplicar os conceitos discutidos anteriormente. No entanto, quando se considera um método baseado em otimização de Pareto, o único critério para determinar a qualidade de uma solução vizinha é a relação de dominância, dificultando o uso dos conceitos originalmente aplicáveis para o caso mono-objetivo.

As seções a seguir descrevem o algoritmo da BT multiobjetivo desenvolvido para o PRS.

8.1 Algoritmo proposto

O uso de um método como a BT para o resolver o PRS se torna muito adequado porque o emprego de toda a informação possível acerca do problema é de fundamental importância para uma resposta rápida e eficiente. Especialmente pela necessidade de resposta instantânea, pode ser comprometedor o desenvolvimento de um método de busca que requeira muito tempo devido à necessidade de avaliação de muitas soluções para fornecer uma resposta viável. Logo, a possibilidade de uso das várias formas de memória adaptativa confere ao método uma grande potencialidade quanto à inclusão de estratégias

que utilizem características bem específicas do PRS.

Na versão da BT desenvolvida para resolver o PRS definido no Capítulo 4 é utilizado o critério de otimização de Pareto. Ainda que com certas limitações, extrapolam-se os conceitos da técnica para o caso multiobjetivo no sentido de obter versatilidade e eficiência a partir da exploração de características particulares do problema. A primeira e mais importante dessas características se refere à geração da vizinhança e à escolha da solução vizinha: segundo a otimização de Pareto, uma solução será melhor que outra caso a primeira domine a segunda. Dessa forma, apenas na vizinhança de uma solução x podem existir várias soluções não-dominadas, o que exige que todas elas sejam incluídas na aproximação do conjunto de Pareto. Exatamente a manutenção deste conjunto é outra característica importante da versão da BT desenvolvida em relação à versão mono-objetivo: ao invés de manter apenas uma única solução sob a qual escolhe-se iterativamente uma solução vizinha, no caso multiobjetivo é necessário manter um conjunto de soluções a partir do qual serão escolhidas soluções para geração das respectivas vizinhanças a serem exploradas em paralelo. Portanto, a cada iteração é preciso selecionar do conjunto de soluções não-dominadas aquelas soluções para as quais serão obtidas as respectivas soluções vizinhas.

A correspondência com o conceito de aceitação de soluções que piorem o valor de função objetivo, oriunda da versão mono-objetivo, é obtida através da aceitação de soluções dominadas, com o objetivo de que estas soluções funcionem como soluções-ponte que futuramente permitam alcançar soluções vizinhas não-dominadas. Especialmente no PRS definido neste trabalho, as funções objetivo descrevem um espaço objetivo não convexo e descontínuo, onde a busca pode ter muitos problemas para alcançar certas regiões.

Em relação à memória adaptativa, optou-se pelo armazenamento de atributos que relacionam a proibição temporária e a frequência de um movimento. A partir da proibição evita-se a ciclagem, e com a frequência é possível conduzir a busca por regiões ainda não exploradas.

O algoritmo da Figura 8.1 apresenta a BT multiobjetivo desenvolvida para o PRS, composta de duas etapas básicas: uma *construtiva*, com a utilização do algoritmo descrito na Seção 6.1, e outra de *melhoramento*, correspondente ao algoritmo da Seção 6.2 com algumas modificações. As duas memórias utilizadas são as de curto e longo prazo: a primeira corresponde à lista tabu, que contém os movimentos temporariamente proibidos; a segunda armazena a frequência com que um certo movimento foi realizado, frequência esta computada ao longo da história pregressa da busca. A Seção 8.1.1 descreve com mais detalhes estas duas memórias empregadas na BT multiobjetivo proposta.

Três parâmetros fazem parte do procedimento BT da Figura 8.1: o número máximo

de iterações da BT ($Niter$), o número de vizinhanças a serem exploradas em paralelo (N_{sol}) e a rede em operação¹ antes do defeito que ocasionou a contingência ($redeOp$). No primeiro passo do algoritmo são inicializadas as duas listas que compõem a memória adaptativa da BT desenvolvida, a *lista tabu* (LT) e a *lista de frequência* (LF), além do conjunto para armazenar as soluções não-dominadas CP , do contador de iterações it do algoritmo e do contador de iterações da etapa de diversificação it_{diver} .

Os passos 2-6 do algoritmo da Figura 8.1 correspondem à fase construtiva, incluindo dois algoritmos, um para minimizar a carga não restaurada (passo 4) e outro para minimizar o número de chaves manipuladas (passo 5). A execução dos dois algoritmos é repetida por $Niter$ vezes e, ao final, todas as soluções encontradas são armazenadas na aproximação do conjunto de Pareto CP .

A fase de melhoramento, ou de busca local, compreende os passos 8-35, onde os mecanismos da BT irão atuar. Quatro etapas distintas são bem evidentes neste laço:

Etapa 1. (passos 9-11): refere-se à escolha das N_{sol} soluções para as quais serão geradas as respectivas vizinhanças, caso o conjunto CP_r tenha uma cardinalidade maior que N_{sol} (passos 10 e 11). Conforme dito na Seção 6.2, com o intuito de promover uma exploração mais ampla no espaço objetivo, obtém-se N_{sol} soluções representativas do conjunto CP_r , que é uma cópia do conjunto de soluções não-dominadas PS encontrado pelo algoritmo (passo 9), através do algoritmo de *clustering* descrito na Seção 6.2.1 (passo 11).

Etapa 2. (passos 12-17): corresponde ao procedimento de geração de vizinhança para cada uma das soluções do conjunto CP_r . Ele é idêntico ao procedimento descrito na Seção 6.2, no entanto, pequenas modificações tornam-se necessárias em função das duas listas (LT e LF) referentes à memória adaptativa da BT. Por isso, a Seção 8.1.1 a seguir discutirá com detalhes todo o procedimento envolvido no passo 14. Nessa fase, entre os passos 12-17, primeiramente o conjunto criado para armazenar as soluções vizinhas não-dominadas CP_{viz} é inicializado vazio (passo 12). Depois, no laço entre os passos 13 e 17, constrói-se um conjunto $N(s_k)$ com todas as soluções vizinhas da solução s_k (passo 14), do qual serão extraídas as soluções não-dominadas para serem armazenadas no conjunto CP_{viz} (passos 15-17).

Etapa 3. (passos 18-31): nesta etapa serão consideradas as soluções vizinhas não-dominadas do conjunto CP_{viz} . Primeiro, no laço entre os passos 19-22 serão armazenadas

¹A partir do estado das chaves desta rede será possível avaliar o número de chaves que tiveram seus estados alterados, e assim avaliar a segunda função objetivo do PRS definida no Capítulo 4.

em CP (conjunto de soluções não-dominadas da BT) todas as soluções s , contidas em CP_{viz} , que forem não-dominadas em relação ao conjunto CP . A seguir, nos passos 23-31, são selecionadas N_{sol} soluções dominadas do conjunto CP_{viz} (passo 27, descrito com detalhes na Seção 8.1.1) sempre que nenhuma solução não-dominada for incluída em CP no passo 22. Caso o conjunto CP_{viz} tenha participado com ao menos uma solução não-dominada para CP , exclui-se deste conjunto aquelas soluções dominadas que eventualmente foram incluídas em iterações anteriores. Além disso, o contador it_{diver} , utilizado para disparar a fase de diversificação da BT, é incrementado toda vez que as soluções dominadas são incluídas no conjunto CP e inicializado com zero assim que ao menos uma solução não-dominada de CP_{viz} seja incluída em CP .

Etapa 4. (passos 32-34): nestes passos ocorre a execução da fase de diversificação da BT. Ela será disparada (passo 34) sempre que o contador it_{diver} for maior que a décima parte do número total de iterações da BT ($Niter$).

No último passo do laço 8-35 é incrementado o contador de iterações da BT. Finalmente no passo 37 é retornado como resultado do algoritmo o conjunto com as soluções não-dominadas encontradas pela busca.

Na Seção 8.1.1 a seguir será discutida com detalhes a geração da vizinhança, passo 14 da Figura 8.1, e a memória adaptativa utilizada (listas LT e LF). Nas Seções 8.1.2 e 8.1.3 são discutidas a seleção de soluções dominadas do passo 27 e a fase de diversificação do passo 34, respectivamente.

8.1.1 Geração da vizinhança e memória adaptativa

O mecanismo para gerar soluções vizinhas utilizado na BT da Figura 8.1 é aquele descrito na Seção 6.2 com algumas modificações para incluir as restrições de movimentos proporcionadas pela memória adaptativa da BT.

A memória adaptativa utilizada é composta por duas listas que guardam os atributos dos movimentos realizados: a lista tabu (LT), que armazena os movimentos temporariamente proibidos, e a lista de frequência (LF), que armazena a frequência com que os movimentos foram selecionados no decorrer da busca.

Para a vizinhança escolhida, definida na Seção 6.2.2, os movimentos possíveis a partir de uma solução são todos os caminhos de interligação até um nó-fonte para cada nó pertencente à área escura. Considerando a solução pós-falha da Figura 6.11(b), todos os caminhos possíveis são dados pela Tabela 6.1, os quais foram obtidos a partir dos grafos

```

BUSCATABUMULTIOBJETIVO( $Niter, N_{sol}, redeOp$ )
1   $LT \leftarrow LF \leftarrow CP \leftarrow \emptyset; it \leftarrow it_{diver} \leftarrow 0;$ 
2   $LNF \leftarrow DETERMINANÓSFONTE(redeOp);$ 
3  enquanto  $it < Niter$ 
4  faça  $CP \leftarrow ALGCONSTMAXCARGA(LNF, redeOp, CP);$ 
5      $CP \leftarrow ALGCONSTMINCHAV(LNF, redeOp, CP);$ 
6      $it \leftarrow it + 1;$ 
7   $it \leftarrow 0;$ 
8  enquanto  $it < Niter$ 
9  faça  $CP_r \leftarrow CP;$ 
10     se  $|CP_r| \geq N_{sol}$ 
11         então  $CP_r \leftarrow REDUZ(CP_r, N_{sol});$ 
12      $CP_{viz} \leftarrow \emptyset;$ 
13     para cada  $s_k \in CP_r$ 
14         faça  $N(s_k) \leftarrow GERA VIZINHANÇA(k, s_k, LT, LF, CP);$ 
15             para cada  $s \in N(s_k)$ 
16                 faça se  $NÃO DOMINADA(s, CP_{viz})$ 
17                     então  $CP_{viz} \leftarrow CP_{viz} \cup \{s\};$ 
18      $I \leftarrow 0;$ 
19     para cada  $s \in CP_{viz}$ 
20         faça se  $NÃO DOMINADA(s, CP)$ 
21             então  $I \leftarrow 1;$ 
22              $CP \leftarrow CP \cup \{s\};$ 
23     se  $I = 0$ 
24         então  $it_{diver} \leftarrow it_{diver} + 1;$ 
25          $k \leftarrow 0;$ 
26         enquanto  $k < N_{sol}$ 
27             faça  $s \leftarrow SELECIONADOMINADA(CP_{viz});$ 
28              $CP \leftarrow CP \cup \{s\};$ 
29              $k \leftarrow k + 1;$ 
30         senão  $it_{diver} \leftarrow 0;$ 
31              $EXCLUIDOMINADAS(CP);$ 
32     se  $it_{diver} > (Niter/10)$ 
33         então  $it_{diver} \leftarrow 0;$ 
34          $CP \leftarrow DIVERSIFICA((Niter/20), N_{sol}, LT, LF, CP);$ 
35      $it \leftarrow it + 1;$ 
36  $EXCLUIDOMINADAS(CP);$ 
37 retorna  $CP;$ 

```

Figura 8.1: Pseudo-código da Busca Tabu multiobjetivo desenvolvida.

das Figuras 6.11(c)-6.11(g). Assim que for gerada a vizinhança de uma solução x , todos os caminhos utilizados serão proibidos nas próximas $N_{iter}/10$ iterações (duração tabu), parâmetro que pode ser alterado sempre que for conveniente. Esta proibição será incluída na LT de cada uma das N_{sol} trajetórias, o que significa que um movimento pode ser permitido para uma trajetória k e proibido para a trajetória $k + 1$. Busca-se, com isso, explorar de forma mais ampla e eficiente o espaço objetivo.

Como exemplo, considerando a rede pós-falha da Figura 6.11(b) com os nós 7-11 na área escura, será mantida uma lista de tamanho igual ao número de caminhos definidos na Tabela 6.1, ou seja, 24. Além disso, cada uma das N_{sol} trajetórias terá a sua própria lista de movimentos proibidos LT . Além da proibição dos movimentos, também são mantidas outras N_{sol} listas LF que guardam a frequência com que os movimentos são escolhidos. Portanto, cada trajetória possui uma lista tabu e outra lista de frequência. A Figura 8.2 ilustra estas n listas tabu e as n listas de frequência, com tamanho igual a 24, para as n trajetórias de busca, onde $n = N_{sol}$.

Cabe salientar que é utilizado um critério de aspiração na BT desenvolvida: ainda que um movimento seja proibido, caso a sua realização ocasione uma solução não-dominada com relação à CP , este movimento é aceito e a respectiva solução incluída no conjunto CP_{viz} .

Na Figura 8.3 é apresentado o procedimento utilizado para gerar a vizinhança proposta para o PRS, incluído no passo 14 da Figura 8.1. Os parâmetros incluem o índice k da trajetória de busca, a solução s_k para a qual será gerada a vizinhança, as listas tabu LT e as listas de frequência LF , e o conjunto de soluções não-dominadas CP encontradas pela BT.

No primeiro passo do algoritmo da Figura 8.3 é realizada a inicialização do conjunto que armazena as soluções vizinhas de s_k . O laço a seguir, entre os passos 2 e 15, gera as soluções vizinhas alterando o caminho de interligação à área clara de cada nó i pertencente à área escura original AE^2 . Neste processo, primeiramente a solução s_k é copiada para a solução s (passo 3) para, a seguir, o nó i ser desconectado do nó-fonte atual (passo 4). Isto significa abrir a primeira chave no caminho do nó i até o nó-fonte atual. A solução s obtida deste processo é avaliada quanto à dominância (passos 5 e 6), sendo incluída na vizinhança de s_k caso ela seja não-dominada em relação ao conjunto CP (passo 6). No laço entre os passos 7 e 15 são geradas todas as soluções obtidas através da utilização de todos os caminhos C_i que interligam o nó i à área clara. A conexão propriamente dita é realizada no passo 9, com a posterior verificação se a solução s_i , gerada pela utilização do caminho c , é não-dominada em relação ao conjunto CP . Em caso afirmativo, a solução

²Na Figura 6.11(b), $AE = \{7, 8, 9, 10, 11\}$.

	<i>LT 1</i>	<i>LT 2</i>	...	<i>LT n</i>		<i>LF 1</i>	<i>LF 2</i>	...	<i>LF n</i>
1	0	1	...	3	1	0	1	...	3
2	2	1	...	0	2	2	1	...	0
3	0	0	...	0	3	0	0	...	0
4	0	0	...	0	4	0	0	...	0
5	0	0	...	0	5	0	0	...	0
6	4	5	...	4	6	4	5	...	4
7	6	2	...	1	7	6	2	...	1
8	2	3	...	2	8	2	3	...	2
9	3	0	...	1	9	3	0	...	1
10	0	0	...	0	10	0	0	...	0
11	1	2	...	0	11	1	2	...	0
12	4	5	...	2	12	4	5	...	2
13	1	2	...	3	13	1	2	...	3
14	3	2	...	2	14	3	2	...	2
15	1	0	...	6	15	1	0	...	6
16	3	6	...	2	16	3	6	...	2
17	4	4	...	0	17	4	4	...	0
18	0	0	...	0	18	0	0	...	0
19	0	0	...	0	19	0	0	...	0
20	1	2	...	3	20	1	2	...	3
21	7	8	...	6	21	7	8	...	6
22	0	0	...	0	22	0	0	...	0
23	0	0	...	0	23	0	0	...	0
24	0	0	...	0	24	0	0	...	0

Figura 8.2: Memória adaptativa utilizada na BT desenvolvida: uma lista tabu (*LT*) e uma lista de frequência (*LF*) para cada uma das N_{sol} trajetórias de busca.

s_i é incluída no conjunto $N(s_k)$ e o caminho c é declarado proibido pela inclusão na lista LT_k e incrementado o seu valor na lista LF_k (passo 12). Em caso negativo, se a solução s_i gerada no passo 9 for dominada em relação ao conjunto CP , esta solução somente é incluída no conjunto $N(s_k)$ caso ela não seja proibida (passos 13-15). Finalmente no passo 16 é retornado o conjunto $N(s_k)$, com as soluções vizinhas de s_k .

```

GERAVIZINHANÇA( $k, s_k, LT, LF, CP$ )
1   $N(s_k) \leftarrow \emptyset$ ;
2  para cada  $i \in AE$ 
3  faça  $s \leftarrow s_k$ ;
4       $s \leftarrow \text{DESCONECTANO FONTE ATUAL}(i, s)$ ;
5      se NÃODOMINADA( $s, CP$ )
6          então  $N(s_k) \leftarrow N(s_k) \cup \{s\}$ ;
7      para cada  $c \in C_i$ 
8          faça  $s_i \leftarrow s$ ;
9               $s_i \leftarrow \text{CONECTANOVO CAMINHO}(i, c, s_i)$ ;
10             se NÃODOMINADA( $s_i, CP$ )
11                 então  $N(s_k) \leftarrow N(s_k) \cup \{s_i\}$ ;
12                     INCLUIMOV( $c, LT_k, LF_k$ );
13             senão se !PROIBIDO( $c, LT$ )
14                 então  $N(s_k) \leftarrow N(s_k) \cup \{s_i\}$ ;
15                     INCLUIMOV( $c, LT_k, LF_k$ );
16 retorna  $N(s_k)$ ;

```

Figura 8.3: Pseudo-código da procedimento “GeraVizinhança” do passo 14 da Figura 8.1.

Com o objetivo de reduzir o número de movimentos e, conseqüentemente, o número de avaliações, foi desenvolvido um procedimento para reduzir o número de caminhos do conjunto C_i do passo 7. Ao invés de avaliar todos os caminhos possíveis de interligação do nó i à área clara, é considerado somente aquele com a maior frequência segundo a lista LF_k . Tal escolha privilegia aqueles caminhos que mais contribuíram com soluções não-dominadas, tanto para o conjunto CP quanto para o conjunto CP_{viz} .

8.1.2 Seleção de soluções dominadas

No passo 27 da Figura 8.1 é realizada a seleção de soluções dominadas do conjunto CP_{viz} para posterior inclusão no conjunto CP (passo 28). Essas soluções dominadas poderão contribuir para aumentar a diversidade entre as soluções do conjunto CP , já que tal procedimento só é executado quando nenhuma solução não-dominada é obtida em todas as k trajetórias da busca da iteração atual.

A questão que permanece é como escolher as soluções dominadas. Obviamente não é desejável que sejam escolhidas soluções muito distantes da aproximação atual da fronteira de Pareto (conjunto CP). Diante disso, a mesma abordagem empregada para atribuir um valor de função de adaptação aos indivíduos no AE proposto para o PRS (Seção 7.2), e descrito no Capítulo 7, foi escolhida para tal propósito. Isso se deve à relação direta desta abordagem com o conceito de otimização de Pareto, ao mesmo tempo em que privilegia aquelas soluções que apresentarem o menor grau de dominância.

Dessa forma, é escolhida do conjunto CP_{viz} aquela solução que tiver o menor valor da função F_{a_2} , definida na Eq. 7.2, aqui repetida por já ter demonstrado sua utilidade.

8.1.3 Mecanismo de diversificação

Na BT da Figura 8.1, passo 34, emprega-se um procedimento de diversificação com o objetivo de explorar regiões não visitadas pela busca. Este procedimento faz uso das listas LF de cada trajetória para influenciar as escolhas dos caminhos durante a geração das vizinhanças das soluções representativas.

A Figura 8.4 apresenta o algoritmo utilizado para a fase de diversificação da BT proposta. Ele é muito semelhante à fase de melhoramento do algoritmo da Figura 8.1. Os cinco parâmetros são o número máximo de iterações $Niter$ da diversificação, o número de vizinhanças a serem exploradas em paralelo (N_{sol}), a lista tabu (LT), a lista de frequência (LF) e o conjunto de soluções não-dominadas encontrado pela busca até o momento (CP). Primeiramente é inicializado o conjunto CP_{viz} , o contador de iterações it e a lista tabu de todas as trajetórias de busca. Dessa forma, depois da fase de diversificação não haverá nenhum caminho proibido.

No laço entre os passos 3 e 12 da Figura 8.4 são geradas as soluções vizinhas para cada uma das N_{sol} trajetórias de busca. Cabe salientar que as escolhas dos caminhos no passo 7 é orientada pelos atributos da lista LF : isto significa que aqueles movimentos com menor frequência têm prioridade na escolha, de forma a conduzir a busca por regiões ainda não visitadas. Outra diferença em relação ao algoritmo da Figura 8.1 refere-se à verificação da dominância das soluções vizinhas geradas: toda solução que for não-dominada em relação ao conjunto CP_{viz} é incluída no mesmo. Ao final das $Niter$ iterações, o conjunto CP_{viz} contém todas as soluções não-dominadas geradas pelo processo de diversificação.

Por fim, entre os passos 13 e 15, são armazenadas em CP aquelas soluções pertencentes ao conjunto CP_{viz} que forem não-dominadas em relação a CP . Este conjunto é retornado como resultado final do processo de diversificação e inclui, além das soluções não-dominadas encontradas antes desta etapa, as soluções não-dominadas eventualmente

encontradas pelo algoritmo da Figura 8.4.

```

DIVERSIFICA( $Niter, N_{sol}, LT, LF, CP$ )
1   $LT \leftarrow CP_{viz} \leftarrow \emptyset; it \leftarrow 0;$ 
2   $CP_r \leftarrow CP;$ 
3  enquanto  $it < Niter$ 
4  faça se  $|CP_r| \geq N_{sol}$ 
5      então  $CP_r \leftarrow \text{REDUZ}(CP_r, N_{sol});$ 
6      para cada  $s_k \in CP_r$ 
7      faça  $N(s_k) \leftarrow \text{GERAVIZINHANÇA}(k, s_k, LT, LF, CP);$ 
8           $LT \leftarrow \emptyset;$ 
9          para cada  $s \in N(s_k)$ 
10         faça se  $\text{NÃODOMINADA}(s, CP_{viz})$ 
11             então  $CP_{viz} \leftarrow CP_{viz} \cup \{s\};$ 
12          $CP_r \leftarrow CP_{viz};$ 
13 para cada  $s \in CP_{viz}$ 
14 faça se  $\text{NÃODOMINADA}(s, CP)$ 
15     então  $CP \leftarrow CP \cup \{s\};$ 
16 retorna  $CP;$ 

```

Figura 8.4: Pseudo-código da fase de diversificação empregada na BT da Figura 8.1.

Capítulo 9

Resultados computacionais

Neste capítulo serão apresentados os resultados da aplicação dos dois métodos propostos para o PRS do Capítulo 4: o algoritmo evolutivo do Capítulo 7 e a busca tabu do Capítulo 8. Cabe salientar que uma das contribuições deste trabalho é a resolução deste problema multiobjetivo com um tempo computacional razoável, considerando redes de distribuição de grande porte.

Primeiramente é conveniente lembrar as principais características do PRS e da abordagem adota, discutidas nos capítulos anteriores. Quando da ocorrência de uma contingência no sistema de distribuição de energia elétrica, algumas regiões podem perder o fornecimento. Considerando os requisitos de qualidade das companhias de distribuição, procedimentos para restabelecer o fornecimento tornam-se necessários e incluem planos de manobra que alteram a configuração da rede. Nestes planos, é preciso definir quais chaves serão abertas e quais chaves serão fechadas de modo a restabelecer o fornecimento em parte ou em todas as áreas afetadas. A abordagem desenvolvida propõe vários planos de restabelecimento considerando todas estas áreas, as quais podem se localizar eventualmente em alimentadores distintos. Estes vários planos são proporcionados pela aproximação do conjunto de Pareto obtida ao final da execução dos métodos propostos, visto que ambos são baseados no conceito de otimização de Pareto.

Para avaliar experimentalmente a adequação dos métodos propostos quanto aos requisitos estabelecidos (qualidade da aproximação do conjunto de Pareto e o tempo computacional necessário para execução), foram utilizadas cinco redes de distribuição, com dimensões variando de pequenas, com algumas dezenas de chaves e de barras de carga, até grandes, com algumas centenas de chaves e alguns milhares de barras de carga. A Tabela 9.1 apresenta estas cinco redes: as redes 1 e 2 foram obtidas dos trabalhos de Baran e Wu (1989) e Augugliaro et al. (2000), respectivamente; as três últimas correspondem a

sistemas reais e foram obtidas de uma concessionária brasileira. Cada uma delas é caracterizada com informações relativas ao número de subestações, nós, arcos, setores, chaves, alimentadores e barras de carga. Também são apresentadas as informações relativas à potência ativa total da rede (em kW), ao número de áreas escuras, ao percentual das barras de carga que perdeu o fornecimento de energia elétrica, ao percentual de chaves nas áreas escuras e, por fim, ao percentual da potência ativa total da rede que está sem fornecimento. As Figuras 9.1-9.5 ilustram as topologias das redes da Tabela 9.1.

	Redes de distribuição				
	1	2	3	4	5
Num. subestações	1	5	1	1	2
Num. nós	34	90	1.057	2.645	5.736
Num. arcos	40	113	1.078	2.676	5.846
Num. setores	1	1	42	224	467
Num. chaves	40	113	63	256	587
Num. alimentadores	1	6	3	11	27
Num. barras	32	75	417	716	1.575
Potência ativa total (kW)	3.715	28.621	11.542	42.664	103.750
Num. áreas escuras	11	10	3	4	5
Barras sem fornecimento (%)	34,37	13,33	39,08	38,68	21,39
Chaves nas áreas escuras (%)	37,5	14,15	38,09	31,25	20,61
Potência ativa sem fornecimento (%)	40,1	13,32	28,96	29,38	16,3

Tabela 9.1: Redes de distribuição empregadas nos testes computacionais.

Conforme já discutido, torna-se muito conveniente em otimização multiobjetivo a disponibilidade, senão da fronteira ideal de Pareto, de uma boa aproximação desta para avaliar a qualidade de um procedimento heurístico, como as propostos nesta tese. Por isso foi desenvolvido um método de enumeração pseudo-exaustiva para gerar combinações que nada mais são do que os indivíduos do algoritmo evolutivo. O processo é baseado na aleatoriedade com que são geradas as combinações de caminhos de interligação entre os nós da área escura e os nós-fonte da área clara. Além disso, também é escolhido aleatoriamente a cardinalidade da combinação, o que significa que uma delas pode conter caminhos de interligação para apenas uma parte dos nós da área escura, com a ordem dos caminhos também aleatória. Para as redes 1 e 2 foi possível obter uma boa aproximação



Figura 9.3: Topologia da rede de distribuição 3.

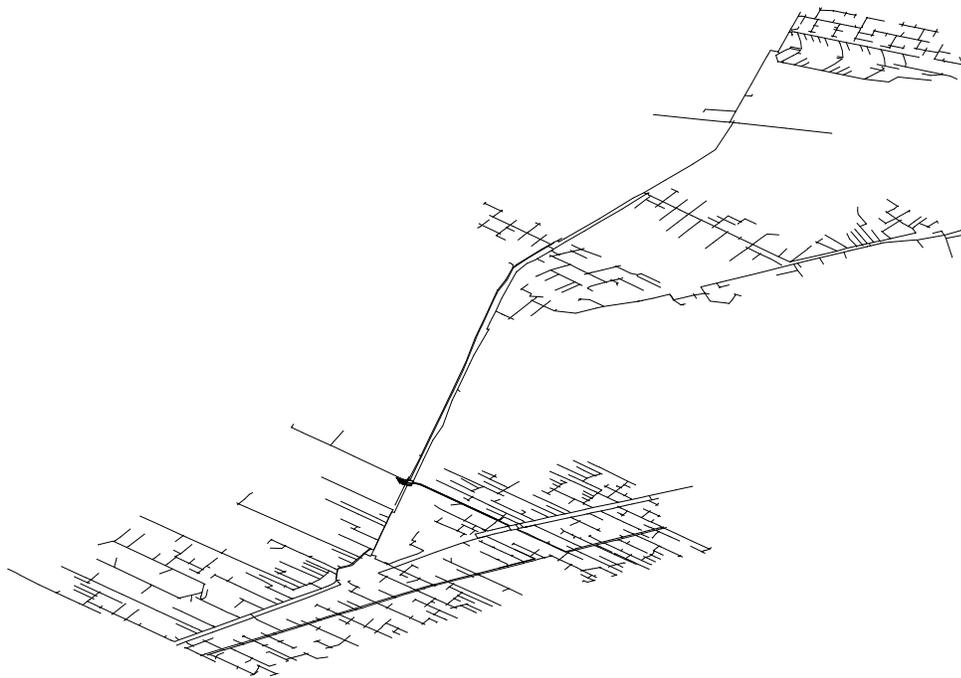


Figura 9.4: Topologia da rede de distribuição 4.



Figura 9.5: Topologia da rede de distribuição 5.

da fronteira ideal de Pareto com o emprego deste procedimento, entretanto, para as redes 3-5, devido ao elevado número de combinações, os resultados deste procedimento não foram satisfatórios considerando a qualidade a aproximação obtida e o tempo gasto, ainda que seja executado por vários dias. A aproximação para estas três redes maiores foi obtida através da execução de ambos algoritmos por várias horas, com reinícios sistemáticos de modo a cobrir todo o espaço de busca. Neste capítulo, esta aproximação será tratada somente como fronteira “ideal”, embora se tenha consciência de que não é possível afirmar que ela seja a própria fronteira ideal de Pareto.

De posse da melhor aproximação da fronteira ideal de Pareto para as cinco redes da Tabela 9.1, calcula-se o índice com a função descrita na Seção 3.5.3. Para todos os resultados apresentados a seguir também são calculados os respectivos índices segundo esta mesma função e, por questão de simplicidade, são mostrados segundo o desvio percentual em relação a essa melhor fronteira.

Quanto aos limites de quedas de tensão, segundo a resolução 505 da ANEEL é aceitável um desvio de $+/- 5\%$ na tensão nominal. Entretanto, como as redes 3, 4 e 5 são redes reais que apresentam equipamentos para regulagem de tensão e como estes não são contemplados no fluxo de carga desenvolvido neste trabalho, há um desvio maior do que o regulamentar quando são desprezados os ajustes promovidos pela correção da tensão. Ainda que seja uma aproximação, utilizou-se para as redes 3-5 o desvio máximo de tensão encontrado na rede depois da contingência. Dessa forma, não é promovida uma deterioração nos níveis de tensão nas barras de carga da rede maior que aquela já encontrada depois da ocorrência do defeito e antes do plano de restauração.

Com o intuito de analisar de forma mais particular e efetiva a influência dos parâmetros de cada um dos algoritmos, são apresentados os resultados individualizados de ambos para as cinco redes empregadas: a Seção 9.1 contém os resultados para o algoritmo evolutivo e a Seção 9.2 apresenta os resultados da busca tabu. Finalmente, a Seção 9.3 faz um estudo comparativo com os dois métodos desenvolvidos, salientando as potencialidades e eventuais desvantagens de cada um. Em todos os testes, a plataforma computacional utilizada foi um computador PC Pentium 4 2.8 GHz, com sistema operacional Linux 2.6.10, distribuição FEDORA.

9.1 Resultados para o algoritmo evolutivo

Esta seção inclui os resultados da aplicação do Algoritmo Evolutivo do Capítulo 7 para as cinco redes de distribuição da Tabela 9.1.

Através de várias combinações dos parâmetros, buscou-se evidenciar a contribuição de

cada um deles na eficiência no algoritmo. A Tabela 9.2 apresenta estes parâmetros e o conjunto de valores empregado para cada um.

Parâmetro	Valores
Num. gerações	{ 100 200 300 400 500 }
Tam. população	{ 20 40 60 80 100 }
Taxa de mutação	{ 0,2 0,5 0,75 }

Tabela 9.2: Parâmetros utilizados para o AE e os respectivos valores associados.

A partir da execução do AE para todas as configurações de parâmetros em todas as redes, com 10 repetições para cada uma, escolheu-se uma destas configurações que apresentou o melhor desempenho quanto à qualidade da aproximação obtida. A Tabela 9.3 mostra os resultados da aplicação do AE com esta mesma configuração de parâmetros para todas as redes da Tabela 9.1 e as Figuras 9.6-9.10 apresentam graficamente estas fronteiras obtidas, tanto a inicial, alcançada pelo algoritmo construtivo (“front.inicial”), como a final, devolvida como resultado do AE (“front.final”). Também são ilustradas nas figuras a melhor fronteira (“front.ideal”) obtida através da execução por várias horas do AE e da BT propostos. As duas últimas colunas, qualidade inicial e qualidade final, referem-se, respectivamente, ao desvio percentual da fronteira inicial e da fronteira final em relação à fronteira “ideal”. Estes cálculos foram realizados com a medida descrita na Seção 3.5.3.

Rede	Num.gerações	Tam.pop.	Taxa de mutação	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
1	500	100	0,75	8,96	2,34	0
2	500	100	0,75	6,04	0	0
3	500	100	0,75	44,87	47,2	0
4	500	100	0,75	119,62	19,47	19,09
5	500	100	0,75	319,89	23,6	14,19

Tabela 9.3: Resultados da aplicação do AE com uma única configuração de parâmetros para todas as redes da Tabela 9.1.

Pelos resultados da Tabela 9.3 é possível concluir que as duas primeiras redes são facilmente resolvidas pelo AE, na primeira ele reduz o desvio de 2,34% para 0% e na segunda somente o algoritmo construtivo já foi capaz de construir uma aproximação idêntica à

melhor obtida. Na terceira percebe-se uma boa redução pelo AE da fronteira inicial, de 47,20% para 0%. Entretanto, nas duas últimas redes, justamente por serem as maiores, o AE teve dificuldade em obter uma boa fronteira em relação à fronteira “ideal”: na rede 4 o desvio final foi de 19,09% e na rede 5 de 14,19%. Quanto aos tempos computacionais, nas três primeiras instâncias o AE gastou menos de 1 minuto. Nas duas últimas, entretanto, foram necessários quase 2 minutos para a rede 4 e pouco mais de 5 minutos para a rede 5. Mesmo assim, esses resultados estão dentro do esperado, inclusive por se tratar de redes de grande porte.

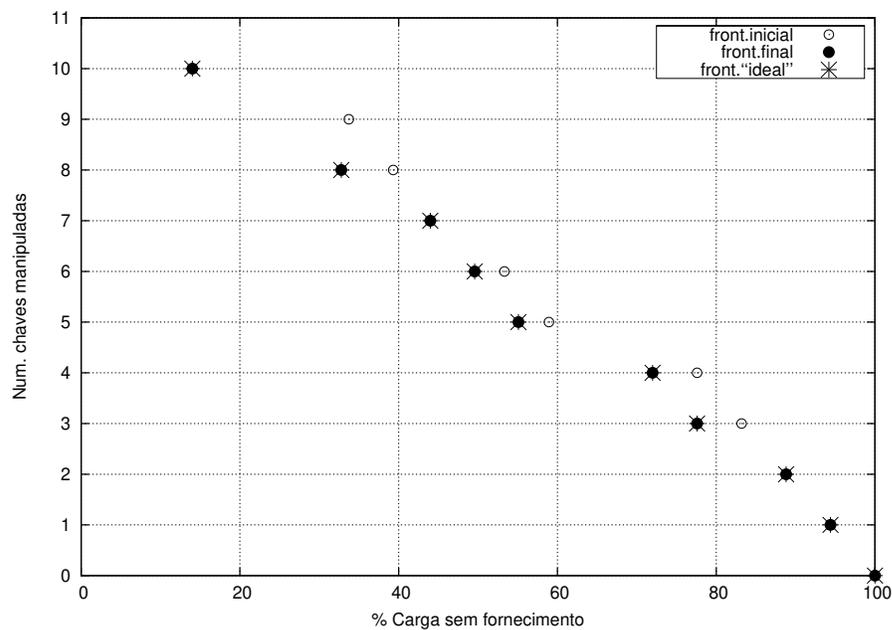


Figura 9.6: Fronteira inicial e final obtidas pela execução do AE para a rede 1 com os parâmetros da Tabela 9.3.

9.2 Resultados para a busca tabu

Nesta seção são apresentados os resultados da aplicação da Busca Tabu multiobjetivo do Capítulo 8 para as redes da Tabela 9.1.

Os parâmetros considerados são o número de iterações da BT (*Niter*), a duração tabu (*DT*), e o número de vizinhanças a serem exploradas em paralelo (*NVP*). A Tabela 9.4 apresenta os valores possíveis de cada um deles. Tal como na seção anterior, foi escolhida

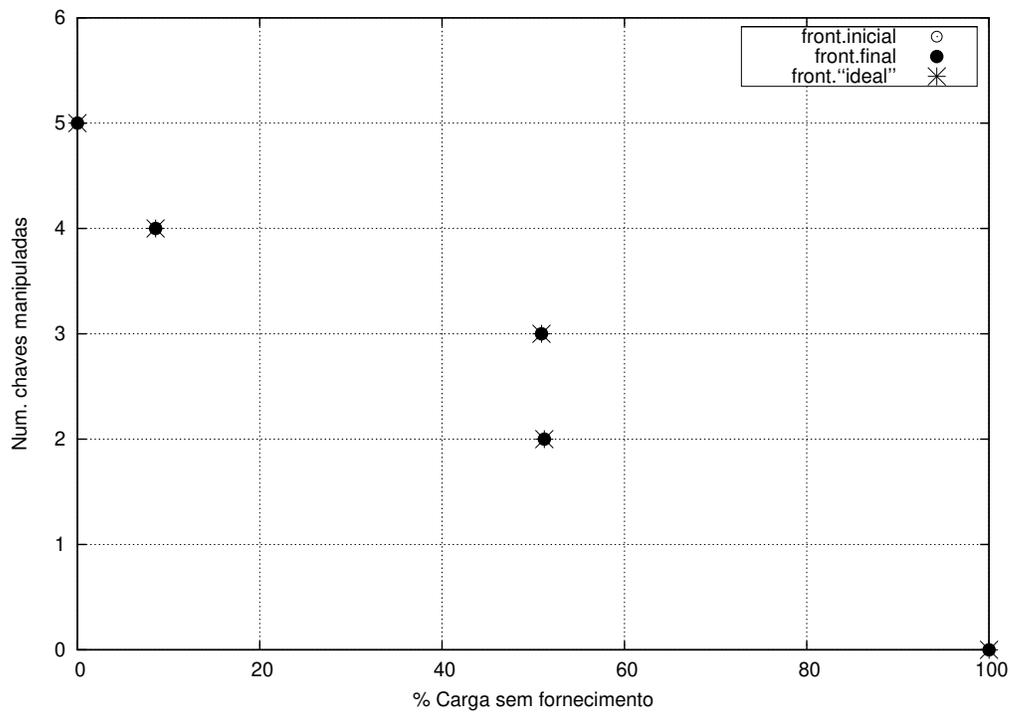


Figura 9.7: Fronteira inicial e final obtidas pela execução do AE para a rede 2 com os parâmetros da Tabela 9.3.

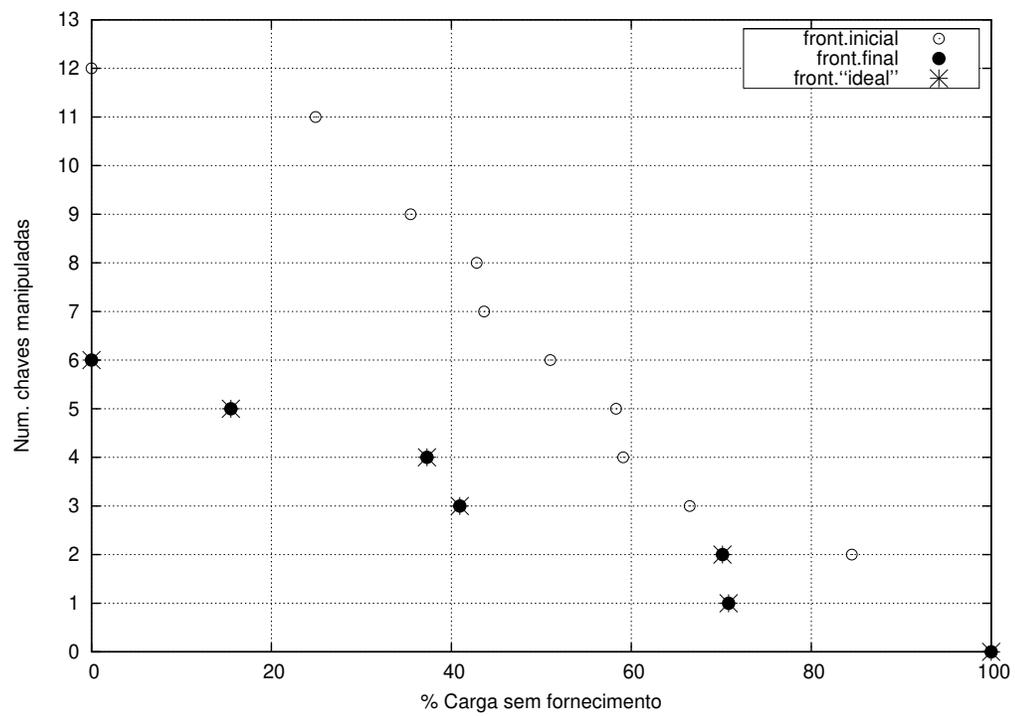


Figura 9.8: Fronteira inicial e final obtidas pela execução do AE para a rede 3 com os parâmetros da Tabela 9.3.

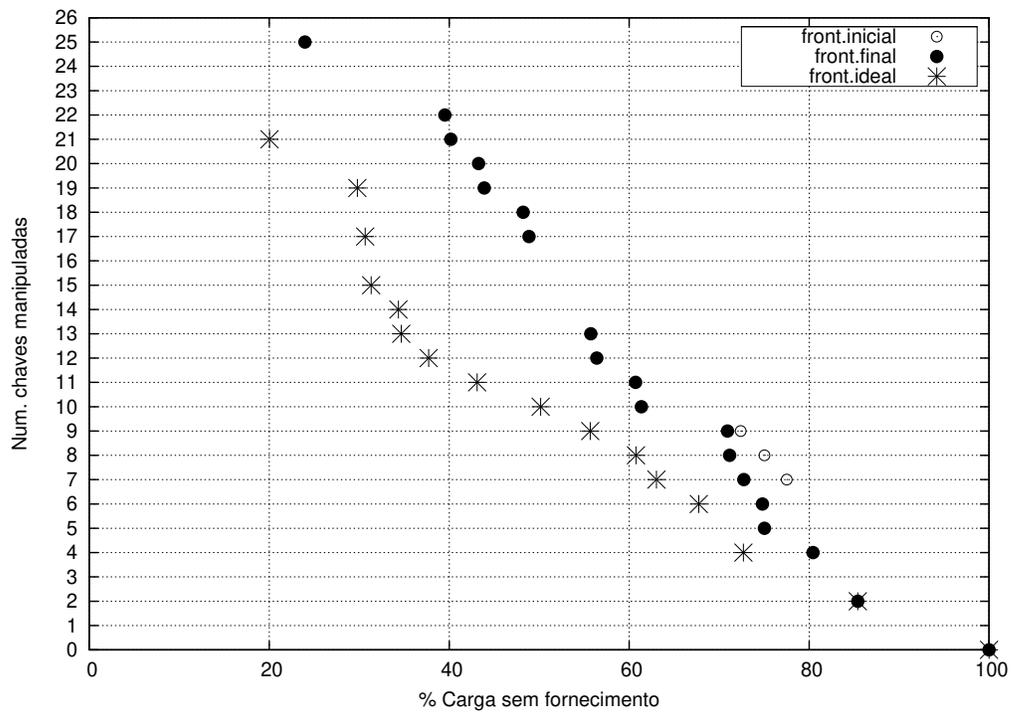


Figura 9.9: Fronteira inicial e final obtidas pela execução do AE para a rede 4 com os parâmetros da Tabela 9.3.

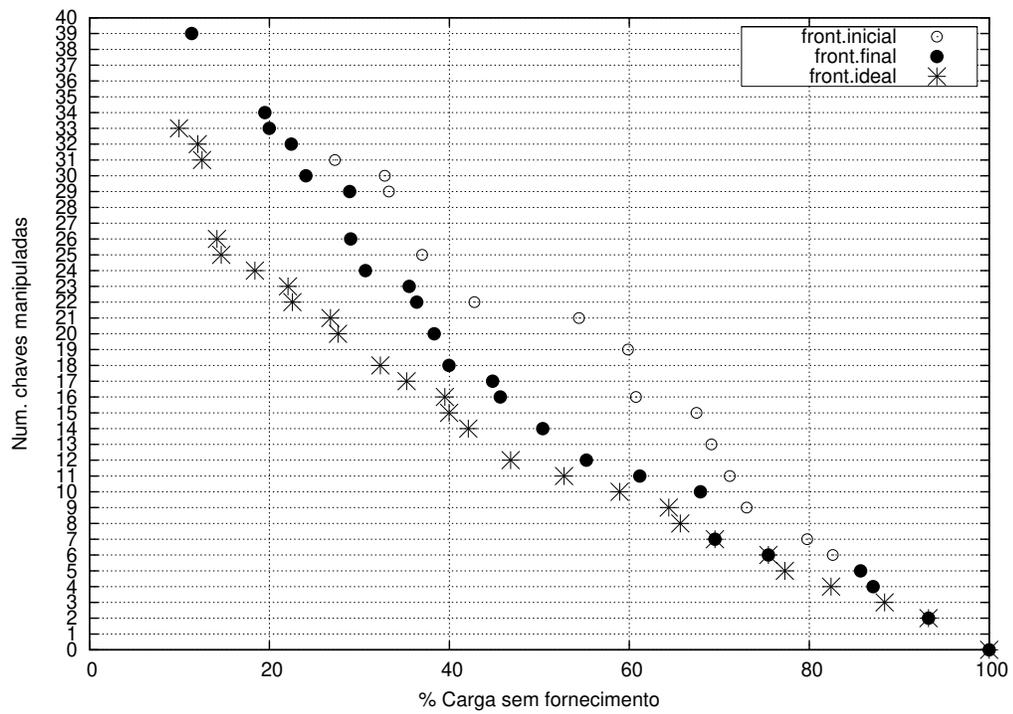


Figura 9.10: Fronteira inicial e final obtidas pela execução do AE para a rede 5 com os parâmetros da Tabela 9.3.

uma configuração de parâmetros que proporcionou os melhores resultados para aplicar o algoritmo nas 5 redes utilizadas.

Parâmetro	Valores
Num. iterações	{ 100 200 300 400 500 }
Duração tabu	{ 10 20 30 40 50 }
Num.viz.paral.	{ 2 4 6 }

Tabela 9.4: Parâmetros utilizados para a BT com os respectivos valores associados.

A Tabela 9.5 mostra os resultados da aplicação da BT com esta mesma configuração de parâmetros para todas as redes da Tabela 9.1, sendo que cada execução, para cada rede, foi repetida 10 vezes e o valor que é apresentado é o menor, considerando a qualidade da aproximação¹. As Figuras 9.11-9.15 ilustram as fronteiras iniciais, finais e “ideais” obtidas para cada uma das cinco redes com os parâmetros apresentados na Tabela 9.5. Tal como nos resultados para o algoritmo anterior, apresentam-se nas duas últimas colunas a qualidade inicial e a qualidade final, referentes ao desvio percentual da fronteira inicial e da fronteira final em relação à fronteira “ideal”.

Rede	Num.iter	DT	NVP	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
1	300	10	6	2,01	2,34	0
2	300	10	6	1,26	0	0
3	300	10	6	33,99	47,2	2,68
4	300	10	6	129,7	19,47	0
5	300	10	6	320,6	26,23	3,14

Tabela 9.5: Resultados da aplicação do BT com uma única configuração de parâmetros para todas as redes da Tabela 9.1.

Nota-se, a partir dos resultados da Tabela 9.5, que a BT teve melhores resultados na aproximação da fronteira “ideal” em comparação com o AE. Para as duas primeiras redes os resultados foram idênticos aos obtidos com o AE. Na rede 3 a BT não conseguiu um desvio igual a zero, tal como o AE. No entanto, para as duas redes maiores, redes 4 e 5, os desvios em relação à fronteira ideal foram bem inferiores aos obtidos pelo AE: para a rede 4, o AE obteve 19,09 enquanto a BT obteve um desvio igual a zero; na rede 5, o AE obteve

¹O tempo computacional, em segundos, é o tempo médio das 10 execuções.

um desvio de 14,19% e a BT 3,14%. Ainda que os tempos computacionais de ambos os métodos sejam semelhantes, a BT teve um número muito maior de soluções avaliadas na maioria das instâncias, o que não permite afirmar a superioridade deste método em relação ao AE baseado apenas nestas comparações.

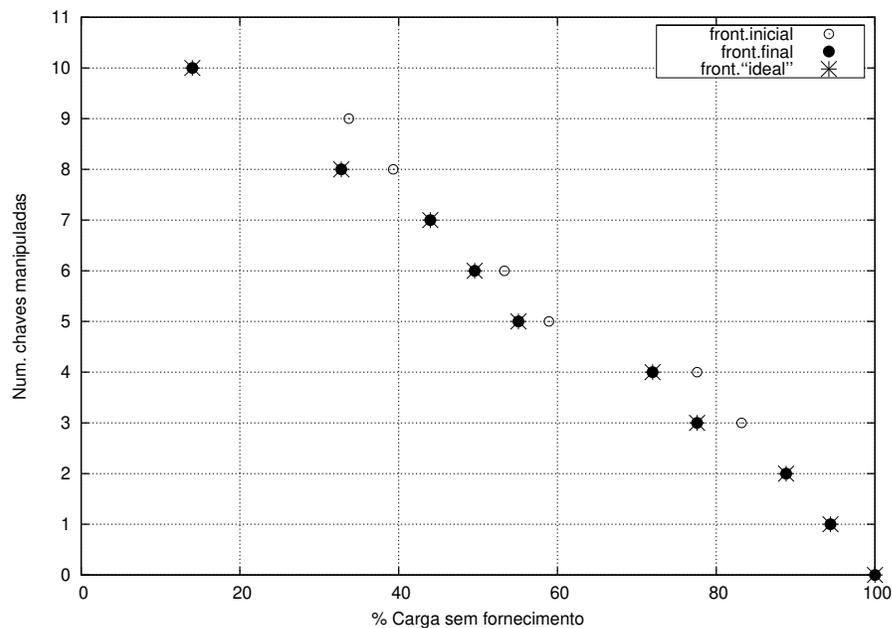


Figura 9.11: Fronteira inicial e final obtidas pela execução da BT para a rede 1 com os parâmetros da Tabela 9.5.

9.3 Comparação dos métodos propostos

Depois da análise dos resultados individuais do AE e da BT propostos, torna-se intuitivo a comparação entre ambos. Quer-se descobrir qual dos dois algoritmos é o mais eficiente para resolver o PRS definido neste trabalho.

A primeira preocupação é proporcionar igualdade de condições para ambos, com o objetivo de fazer uma comparação isenta e realista. Jaszkiwicz (2004) apresenta uma excelente discussão sobre o assunto, destacando que dois critérios devem nortear qualquer comparação de metaheurísticas, sejam elas mono ou multiobjetivo: a qualidade dos resultados e o tempo gasto. Como os dois algoritmos foram escritos na mesma linguagem e usam o mesmo algoritmo construtivo para criar a aproximação inicial do conjunto de Pareto, é bem adequado considerar um critério de parada para ambos relacionado ao

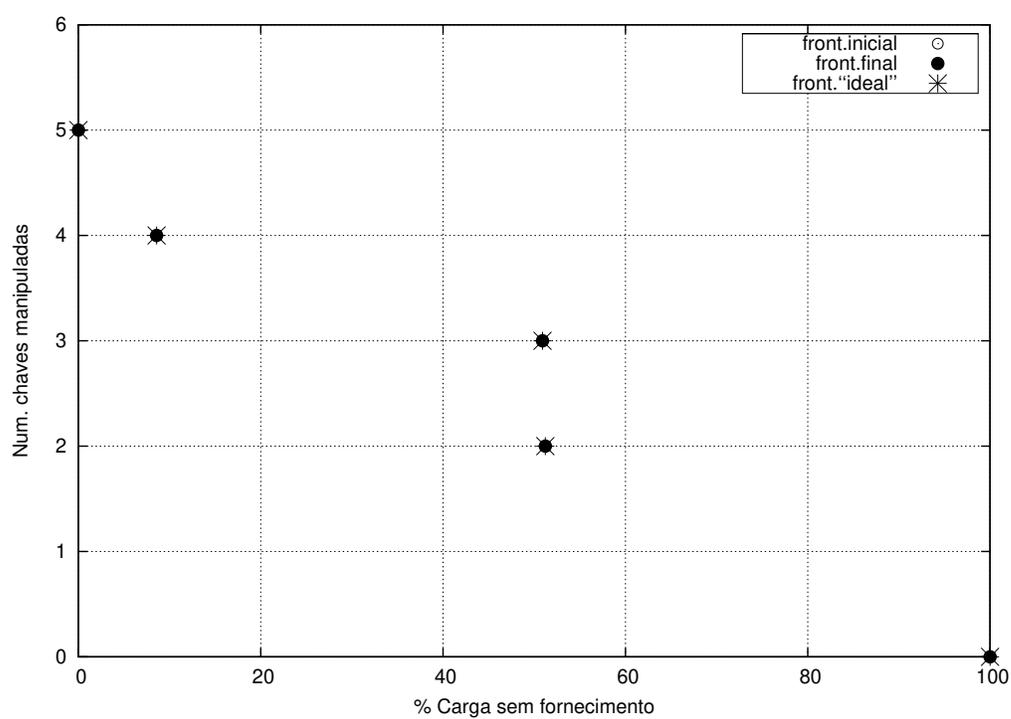


Figura 9.12: Fronteira inicial e final obtidas pela execução da BT para a rede 2 com os parâmetros da Tabela 9.5.

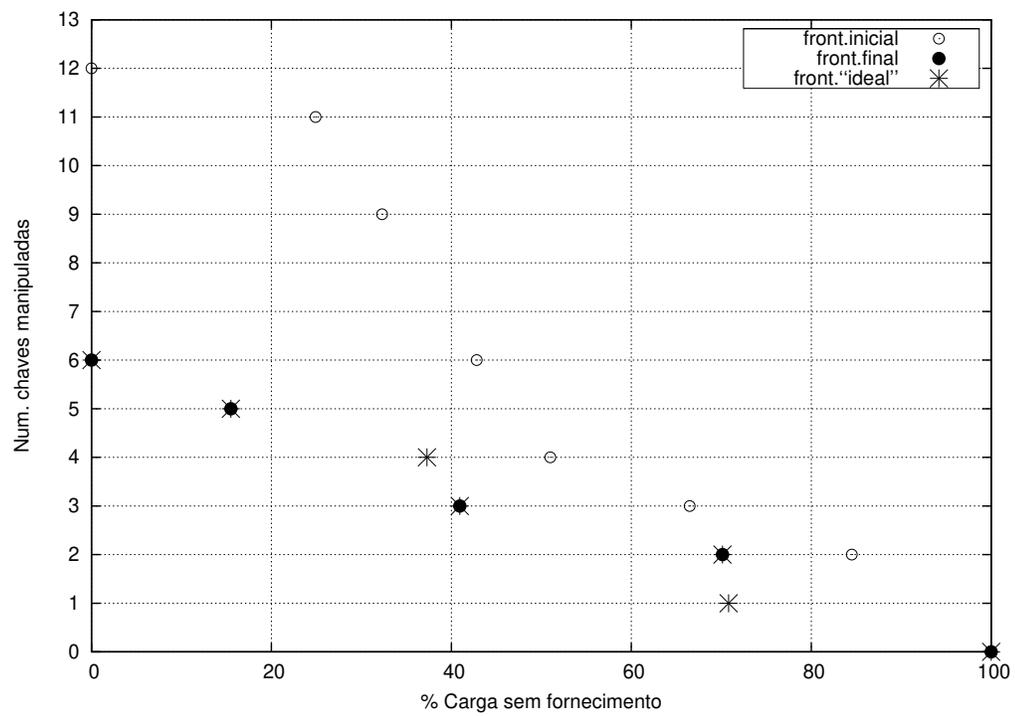


Figura 9.13: Fronteira inicial e final obtidas pela execução da BT para a rede 3 com os parâmetros da Tabela 9.5.

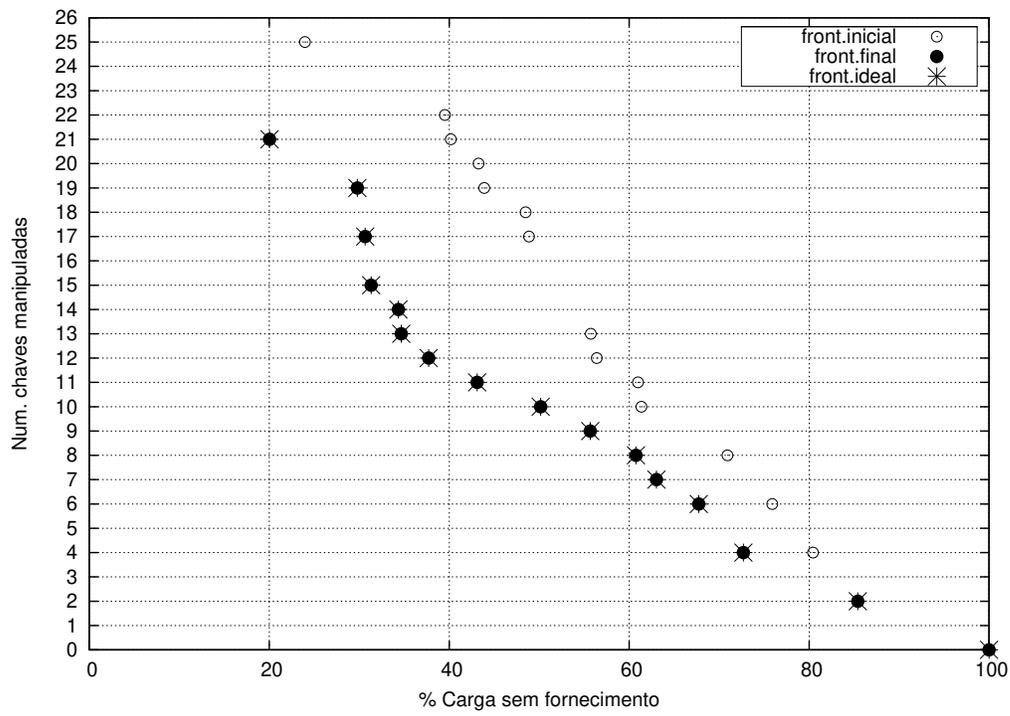


Figura 9.14: Fronteira inicial e final obtidas pela execução da BT para a rede 4 com os parâmetros da Tabela 9.5.

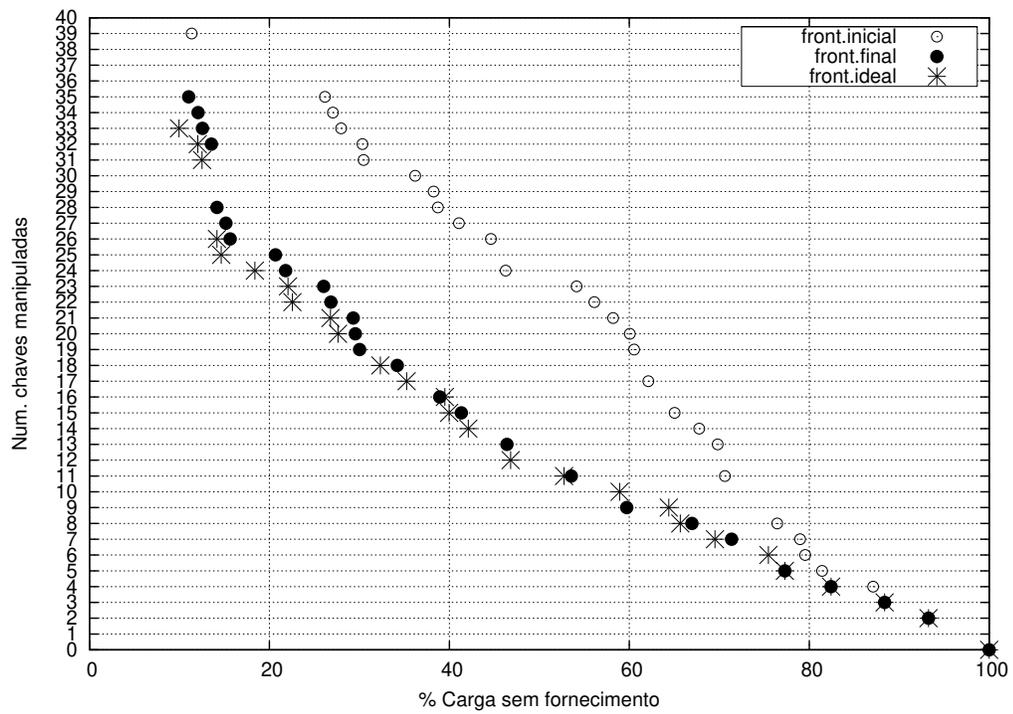


Figura 9.15: Fronteira inicial e final obtidas pela execução da BT para a rede 5 com os parâmetros da Tabela 9.5.

número de soluções avaliadas. No entanto, como em ambos os algoritmos a avaliação envolve a verificação da factibilidade, é preciso separar o processo de avaliação em duas etapas: a primeira refere-se à verificação da factibilidade, com a execução do método de fluxo de carga empregado, e a segunda refere-se à avaliação da solução, quanto às funções objetivo definidas. Tanto no AE quanto na BT, a segunda fase só ocorre se a solução for factível. Por isso, um critério razoável é considerar o número de execuções do método de fluxo de carga, visto que este procedimento é necessário sempre que a rede for alterada de alguma forma.

Definido o critério de parada para a comparação, resta definir os parâmetros para o AE e para a BT. Visto que nos testes apresentados nas duas seções anteriores foram realizadas boas avaliações em relação à melhor combinação para cada um dos algoritmos, optou-se por manter os mesmos parâmetros apresentados nas Tabelas 9.3 e 9.5. Vale ressaltar que o número de gerações e o número de iterações não atuarão como critérios de parada, sendo substituídos pelo número de execuções do método de fluxo de carga: a partir dos testes nas duas seções anteriores, para cada rede escolhe-se o maior valor de soluções avaliadas realizadas na execução do AE e da BT.

Na Tabela 9.6 são apresentados os resultados da execução do AE e da BT para as cinco redes da Tabela 9.1, com o número de soluções avaliadas dado pela segunda coluna. Para cada um dos métodos é especificado o tempo computacional (em segundos) necessário para obter uma fronteira com a qualidade mostrada, medida através do desvio em relação à fronteira “ideal”. A BT mostrou certa superioridade em relação ao AE: nas duas primeiras redes, ambos os métodos tiveram desempenho idêntico quanto ao desvio obtido; na terceira rede, a BT obteve um desvio maior que o do AE; nas duas últimas, justamente as maiores, a BT apresentou um desvio muito menor que o do AE.

Um aspecto importante a ser observado é o tempo gasto pelo AE. Embora o número de soluções avaliadas seja o mesmo para o AE e para a BT, cada um realiza operações particulares durante o processo de busca. Uma provável causa deste tempo elevado são as operações necessárias para avaliar os indivíduos no AE e os mecanismos de redução da população.

Para uma melhor visualização dos resultados da Tabela 9.6, são apresentadas nas Figuras 9.16-9.18 as fronteiras obtidas pelo AE e pela BT para as três últimas redes, justamente aquelas onde houve diferença na qualidade das fronteiras alcançadas.

Rede	Num.sol.avaliadas	AE		BT	
		Tempo	Qualidade (%)	Tempo (s)	Qualidade (%)
1	80.225	13,43	0	2,01	0
2	42.180	5,07	0	1,26	0
3	78.125	70	0	33,99	2,68
4	156.565	355,76	19,08	129,7	0
5	370.039	2.227,72	8,37	320,6	3,14

Tabela 9.6: Comparação de desempenho entre o AE e a BT com as redes da Tabela 9.1.

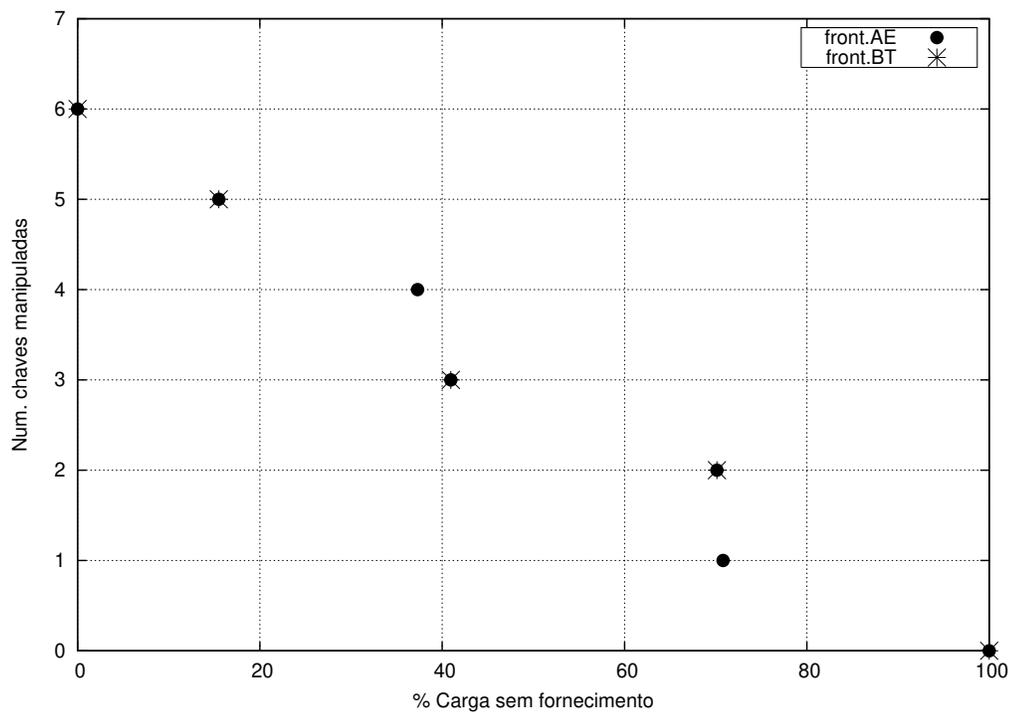


Figura 9.16: Fronteiras obtidas pelo AE e pela BT para a rede 3 da Tabela 9.5.

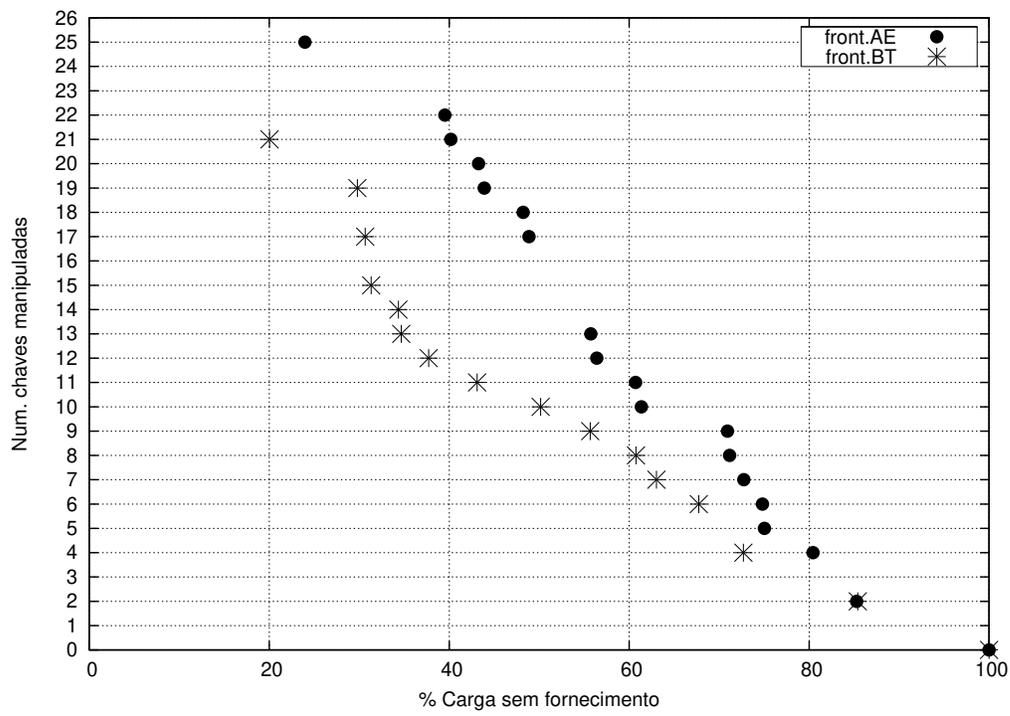


Figura 9.17: Fronteiras obtidas pelo AE e pela BT para a rede 4 da Tabela 9.5.

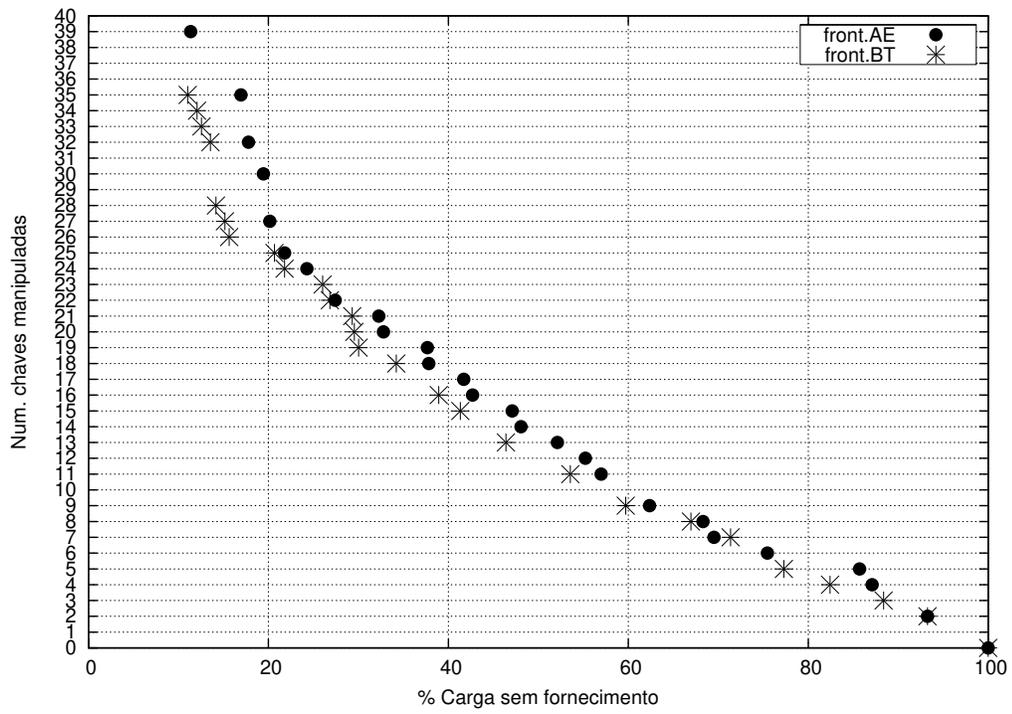


Figura 9.18: Fronteiras obtidas pelo AE e pela BT para a rede 5 da Tabela 9.5.

Capítulo 10

Conclusões

Neste capítulo será apresentado um resumo deste trabalho e suas respectivas contribuições, além de algumas perspectivas de futuras extensões.

Primeiramente foram apresentados os conceitos fundamentais dos sistemas de energia elétrica, juntamente com a revisão bibliográfica do problema. A seguir, no Capítulo 2, foi definido um problema de otimização combinatória e descritas algumas metaheurísticas comumente empregadas para resolução dos mesmos. No Capítulo 3 definem-se os problemas de otimização combinatória multiobjetivo, com os métodos clássicos e algumas metaheurísticas frequentemente utilizadas na literatura para abordar problemas desta natureza. Também são apresentados neste capítulo alguns métodos para avaliação de heurísticas multiobjetivo, com a descrição da medida escolhida. O Capítulo 4 inclui a definição do PRS abordado neste trabalho, com o respectivo modelo matemático, as simplificações adotadas e o método de fluxo de carga utilizado. No Capítulo 5 é descrita a modelagem computacional desenvolvida para o PRS com todos os algoritmos desenvolvidos. Nos Capítulos 6, 7 e 8 são descritos os algoritmos multiobjetivo desenvolvidos para o PRS. Por fim, no Capítulo 9 são apresentados os resultados computacionais da aplicação dos algoritmos desenvolvidos para um conjunto com cinco redes de distribuição.

O maior propósito deste trabalho foi desenvolver um algoritmo que resolvesse o problema multiobjetivo de restauração do serviço (PRS) em redes de distribuição de energia elétrica. Como se trata de um problema emergencial onde o plano de restauração deve ser definido com a maior brevidade, torna-se imperativo que o método de resolução forneça uma resposta satisfatória em um tempo computacional razoável. Esta resposta deve ser segura o suficiente quanto às restrições operativas da rede, na medida em que não permite a violação dos limites de tensão e mantém a radialidade da rede. Além disso, este plano de restauração proposto deve ser seguro o suficiente para evitar o desligamento de outras

partes da rede em função da atuação dos dispositivos de proteção, devido à violação dos limites de fluxo de potência nas linhas.

Com todos estes requisitos, foi desenvolvida uma abordagem multiobjetivo que contempla todos eles e ainda propõe não apenas um, mas um conjunto de planos de restauração, cada um com valor correspondente para cada função objetivo considerada no PRS abordado: a minimização da carga não restaurada e a minimização do número de chaves manipuladas. O emprego da primeira é bem intuitivo, pois se quer restabelecer o fornecimento para a maior área possível. A segunda é particularmente importante para facilitar o retorno da rede para o estado normal de operação. Devido à complexidade deste problema e com o estudo da bibliografia especializada, partiu-se para o desenvolvimento de algoritmos heurísticos, sendo uma heurística construtiva, uma de melhoramento e duas metaheurísticas que fazem uso dos dois primeiros: um algoritmo evolutivo e uma busca tabu.

Para avaliar experimentalmente os algoritmos desenvolvidos e verificar se os requisitos descritos anteriormente foram atendidos, foi utilizado um conjunto com cinco redes de distribuição nos testes computacionais, três delas correspondendo a sistemas atuais de cidades de médio porte do estado de São Paulo. Os resultados apresentados evidenciam o êxito da abordagem desenvolvida, tanto em relação ao tempo gasto para resolução quanto em relação à qualidade da aproximação obtida, verificada através da comparação com a fronteira obtida a partir da execução por várias horas, e com reinícios sistemáticos, das duas metaheurísticas propostas. Quando considerada a comparação entre as duas abordagens e o compromisso entre tempo gasto e qualidade da aproximação, a busca tabu teve uma boa vantagem sobre o algoritmo evolutivo.

A seguir destacam-se as principais contribuições deste trabalho.

- Desenvolvimento de um modelo para o PRS que contempla várias áreas escuras disjuntas na rede de distribuição;
- Abordagem multiobjetivo para o PRS e utilização do critério de otimização de Pareto, com a sugestão de vários planos de manobra, correspondentes à cada solução incluída na aproximação do conjunto Pareto-ótimo;
- Desenvolvimento de um pequeno *framework*, em linguagem C++, para problemas de distribuição de energia elétrica, capaz de suportar a fácil inclusão de outros aspectos do PRS bem como a sua reutilização em problemas correlatos;
- O desenvolvimento de dois algoritmos heurísticos multiobjetivo para o PRS, um construtivo e outro de melhoramento, baseados no conceito de otimização de Pareto;

- O desenvolvimento de duas metaheurísticas multiobjetivo para o PRS: um algoritmo evolutivo e uma busca tabu, ambos baseados em otimização de Pareto.

Por fim, listam-se a seguir algumas perspectivas de trabalhos futuros.

- Análise de outros mecanismos para diversificação e intensificação e a consideração de vizinhanças variáveis;
- Desenvolvimento de outras representações de indivíduos para o algoritmo evolutivo proposto;
- Desenvolvimento de uma versão híbrida para o algoritmo evolutivo, que agregue um operador de busca local tal como o descrito na Seção 6.2;
- Consideração de outros aspectos do PRS, como consumidores prioritários e classificação das chaves da rede quanto aos tipos e à distância geográfica entre elas (para facilitar eventuais deslocamentos das equipes de manutenção que farão as manobras);
- Estudo de uma abordagem em duas fases que inclua, além da reconfiguração da área escura para restabelecimento do fornecimento, a transferência de carga na área clara com o objetivo de aumentar a capacidade de reserva dos alimentadores adjacentes às áreas escuras.

Referências Bibliográficas

- Aarts, E. e Korst, J. (1989). *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons.
- Ahuja, R., Magnanti, T. e Orlin, J. (1993). *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs.
- Aoki, K., Kuwabara, H., Satoh, T. e Kanezashi, M. (1987). Outage state optimal load allocation by automatic sectionalizing switches operation in distribution systems, *IEEE Transactions on Power Delivery* **2**(4): 1177–1185.
- Aoki, K., Nara, K., Itoh, M., Satoh, T. e Kuwabara, H. (1989). A new algorithm for service restoration in distribution systems, *IEEE Transactions on Power Delivery* **4**(3): 1832–1839.
- Aoki, K., Satoh, T., Itoh, M., Kuwabara, H. e Kanezashi, M. (1988). Voltage drop constrained restoration of supply by switch operation in distribution systems, *IEEE Transactions on Power Delivery* **3**(3): 1267–1274.
- Arroyo, J. E. C. (2002). *Heurísticas e metaheurísticas para otimização combinatória multiobjetivo*, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação. Universidade Estadual de Campinas.
- Augugliaro, A., Dusonchet, L. e Sanseverino, E. R. (1998). Service restoration in compensated distribution networks using a hybrid genetic algorithm, *Electric Power Systems Research* **46**: 59–66.
- Augugliaro, A., Dusonchet, L. e Sanseverino, E. R. (2000). Multiobjective service restoration in distribution networks using an evolutionary approach and fuzzy sets, *Electrical Power and Energy Systems* **22**: 103–110.

- Augugliaro, A., Dusonchet, L. e Sanseverino, E. R. (2001). Evolving non-dominated solutions in multiobjective service restoration for automated distribution networks, *Electric Power Systems Research* **59**: 185–195.
- Bäck, T., Fogel, D. B. e Michalewicz, T. (2000). *Evolutionary Computation 1*, Institute of Physics Publishing.
- Baran, M. E. e Wu, F. F. (1989). Network reconfiguration in distribution systems for loss reduction and load balancing, *IEEE Transactions on Power Delivery* **4**(2): 1401–1407.
- Baykasoğlu, A., Owen, S. e Gindy, N. (1999). A taboo search based approach to find the pareto optimal set in multiple objective optimization, *Engineering Optimization* **31**: 731–748.
- Bern, M. e Eppstein, D. (1996). *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, chapter Approximation algorithms for geometric problems, pp. 325–329.
- Cantú-Paz, E. (1997). A survey of parallel genetic algorithms, *Relatório Técnico 97003*, University of Illinois at Urbana-Champaign.
- Cavellucci, C. (1989). *Sistema especialista em manobras de redes de distribuição de energia elétrica*, Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação. Universidade Estadual de Campinas.
- Chankong, V. e Haimes, Y. (1983). *Multiobjective Decision Making – Theory and Methodology*, New York: North-Holland.
- Ciric, R. M. e Popovic, D. S. (2000). Multi-objective distribution network restoration using heuristic approach and mixed integer programming method, *Electrical Power and Energy Systems* **22**: 497–505.
- Coello Coello, C. A. (1996). *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design*, Tese de Doutorado, Department of Computer Science, Tulane University, New Orleans, LA.
- Coello Coello, C. A. e Christiansen, A. D. (2000). Multiobjective optimization of trusses using genetic algorithms, *Computers and Structures* **75**(6): 647–660.

- Cohon, J. L. (1978). *Multiobjective Programming and Planning*, New York: Academic Press.
- Corne, D., Knowles, J. D. e Oates, M. J. (2000). The pareto envelope-based selection algorithm for multi-objective optimisation, *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, Springer-Verlag, London, UK, pp. 839–848.
- Curcic, S., Ozveren, C., Crowe, L. e Lo, P. (1996). Electric power distribution network restoration: a survey of papers and a review of the restoration problem, *Electric Power Systems Research* **35**: 73–86.
- Czyzak, P. e Jaskiewicz, A. (1998). Pareto simulated annealing – a metaheuristic technique for multiple-objective combinatorial optimization, *Journal of Multi-Criteria Decision Analysis* **7**: 34–47.
- Czyzak, P. e Jaskiewicz, A. (1998). Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization, *Journal of Multi-Criteria Decision Analysis* **7**: 34–47.
- Deb, K., Agrawal, S., Pratap, A. e Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II, *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, Springer-Verlag, London, UK, pp. 849–858.
- Fogel, D. B. (1995). *Evolutionary Computation: Toward a new Philosophy of Machine Intelligence*, IEEE Press.
- Fonseca, C. M. e Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, *Genetic Algorithms: Proceedings of the Fifth International Conference*, Morgan Kaufmann, pp. 416–423.
- Fortunato, L., Neto, T. A., Albuquerque, J. e Pereira, M. (1990). *Introdução ao Planejamento da Expansão e Operação de Sistemas de Produção de Energia Elétrica*, Editora da Universidade Federal Fluminense.
- Fourman, M. P. (1985). Compaction of symbolic layout using genetic algorithms, *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum, pp. 141–153.

- Fukuyama, Y., Chiang, H. D. e Miu, K. N. (1996). Parallel genetic algorithm for service restoration in electric power distribution systems, *International Journal of Electrical Power & Energy Systems* **18**(2): 111–119.
- Furlan, J. D. (1997). *Modelagem de Objetos através da UML*, Makron Books.
- Gandibleux, X., Mezdaoui, N. e Fréville, A. (1997). A Tabu Search Procedure to Solve Combinatorial Optimisation Problems, in R. Caballero, F. Ruiz e R. E. Steuer (eds), *Advances in Multiple Objective and Goal Programming*, Vol. 455 of *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, pp. 291–300.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the theory of NP-Completeness*, Freeman, San Francisco.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence, *Computers and Operations Research* **13**: 533–549.
- Glover, F. (1989). Tabu search - part i, *ORSA Journal on Computing* **1**(3): 190–206.
- Glover, F. (1990). Tabu search - part i, *ORSA Journal on Computing* **2**(1): 4–32.
- Glover, F. e Laguna, M. (1997). *Tabu Search*, Kluwer Academic Publishers.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.
- Greenberg, H. (1999). Mathematical programming glossary, <http://www.cudenver.edu/~hgreenbe/glossary>.
- Hajela, P. e Lin, C. Y. (1992). Genetic search strategies in multicriterion optimal design, *Structural Optimization* **4**: 99–107.
- Hansen, M. P. (1997). Tabu Search in Multiobjective Optimisation : MOTS, *Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM'97)*, Cape Town, South Africa.
- Hansen, M. P. e Jaszkiwicz, A. (1998). Evaluating the quality of approximations to the non-dominated set, *Relatório Técnico IMM-REP-1998-7*, University of Denmark.
- Horn, J., Nafpliotis, N. e Goldberg, D. E. (1994). A Niche Pareto Genetic Algorithm for Multiobjective Optimization, *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Vol. 1, IEEE Service Center, Piscataway, New Jersey, pp. 82–87.

- Hsiao, Y.-T. e Chien, C.-Y. (2000). Enhancement of restoration service in distribution systems using a combination fuzzy-ga method, *IEEE Transactions on Power Systems* **15**(4): 1394–1400.
- Hsu, Y. Y. e Kuo, H. C. (1994). A heuristic based fuzzy reasoning approach for distribution system service restoration, *IEEE Transactions on Power Delivery* **9**(2): 948–953.
- Huang, C.-M. (2003). Multiobjective service restoration of distribution systems using fuzzy cause-effect networks, *IEEE Transactions on Power Systems* **18**(2): 867–874.
- Ishibuchi, H. e Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling, *IEEE Transactions on Systems, Man, and Cybernetics* **28**(3): 392–403.
- Ishibuchi, H. e Narukawa, K. (2004). Comparison of local search implementation schemes in hybrid evolutionary multiobjective optimization algorithms, *Fourth International Conference on Hybrid Intelligent Systems*, IEEE Service Center, pp. 404–409.
- Jaszkiewicz, A. (2002). Genetic local search for multiple objective combinatorial optimization, *European Journal of Operational Research* **137**(1): 50–71.
- Jaszkiewicz, A. (2004). Evaluation of multiple objective metaheuristics, in X. Gandibleux, M. Sevaux, K. Sörensen e V. T'kindt (eds), *Metaheuristics for Multiobjective Optimisation*, Vol. 535 of *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, pp. 65–89.
- Jones, D. F., Mirrazavi, S. K. e Tamiz, M. (2002). Multi-objective metaheuristics: An overview of the current state-of-the-art, *European Journal of Operational Research* **137**(1): 1–9.
- Kim, H., Ko, Y. e Jung, K. H. (1992). Algorithm of transferring the load of the faulted substation transformer using the best-first method, *IEEE Transactions on Power Delivery* **7**(3): 1434–1442.
- Knowles, J. D. e Corne, D. W. (1999). The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation, *1999 Congress on Evolutionary Computation*, IEEE Service Center, Washington, D.C., pp. 98–105.
- Larman, C. (1998). *Applying UML and patterns : an introduction to object-oriented analysis and design*, Prentice Hall.

- Lee, S.-J., Lim, S.-I. e Ahn, B.-S. (1998). Service restoration of primary distribution systems based on fuzzy evaluation of multi-criteria, *IEEE Transactions on Power Systems* **13**(3): 1156–1163.
- Lewis, H. R. e Papadimitriou, C. H. (1981). *Elements of the Theory of Computation*, Prentice-Hall International.
- Liu, C. C., Lee, S. e Venkata, S. S. (1988). An expert system operational aid for restoration and loss reduction of distribution systems, *IEEE Transactions on Power Delivery* **3**(2): 619–626.
- Matos, M. A. e Melo, P. (1999). Multiobjective reconfiguration for loss reduction and service restoration using simulated annealing, *Proceedings of IEEE Budapest Power Tech'99*, IEEE Service Center.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, third edn, Springer.
- Michalewicz, Z. e Fogel, D. B. (2000). *How to Solve it*, Spring-Verlag, New York.
- Miu, K. N., Chiand, H.-D., Yuan, B. e Darling, G. (1998). Fast service restoration for large-scale distribution systems with priority customers and constraints, *IEEE Transactions on Power Delivery* **13**(3): 789–795.
- Miu, K. N., Chiang, H.-D. e McNulty, R. J. (2000). Multi-tier service restoration through network reconfiguration and capacitor control for large-scale radial distribution networks, *IEEE Transactions on Power Systems* **15**(3): 1001–1007.
- Morelato, A. L. e Monticelli, A. (1989). Heuristic search approach to distribution system restoration, *IEEE Transactions on Power Delivery* **4**(4): 2235–2241.
- Mori, H. e Takeda, K. (1994). Parallel simulated annealing for power system decomposition, *IEEE Transactions on Power Systems* **9**(2): 789–795.
- Morse, J. N. (1980). Reducing the size of the nondominated set: Pruning by clustering, *Computers and Operations Research* **7**(1–2): 55–66.
- Mostaghim, S., Teich, J. e Tyagi, A. (2002). Comparison of data structures for storing pareto-sets in MOEAs, *2002 World Congress on Computational Intelligence (CEC02)*, pp. 843–849.

- Murphy, L. e Wu, F. F. (1990). A comprehensive analysis of distribution automation systems, *Relatório Técnico M90/72*, University of California, Berkeley.
- Nahman, J. e Strbac, G. (1994). A new algorithm for service restoration in large-scale urban distribution systems, *Electric Power Systems Research* **29**: 181–192.
- Nara, K., Shiose, A., Kitagawa, M. e Ishihara, T. (1992). Implementation of genetic algorithm for distribution systems loss minimum reconfiguration, *IEEE Transactions on Power Systems* **7**(3): 1044–1051.
- Osman, I. H. e Laporte, G. (1996). Metaheuristics: A bibliography, *Annals of Operations Research* **63**: 513–623.
- Pansini, A. T. (1983). *Electrical distribution engineering*, McGraw-Hill.
- Pardalos, P. M. e Resende, M. G. C. (2001). *Handbook of Applied Optimization*, Oxford University Press, New York.
- Ritzel, B. J., Eheart, J. W. e Ranjithan, S. (1994). Using genetic algorithms to solve a multiple objective groundwater pollution containment problem, *Water Resources Research* **30**(5): 1589–1603.
- Rosenberg, R. S. (1967). *Simulation of genetic populations with biochemical properties*, Tese de Doutorado, University of Michigan, Ann Harbor, Michigan, USA.
- Rumbaugh, J. R., Blaha, M. R., Lorensen, W., Eddy, F. e Premerlani, W. (1991). *Object-Oriented Modeling and Design*, Prentice-Hall.
- Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum, pp. 93–100.
- Serafini, P. (1987). Some considerations about computational complexity for multiobjective combinatorial problems, *LNEMS*, Vol. 294, Springer Verlag, pp. 222–232.
- Serafini, P. (1994). Simulated Annealing for Multiple Objective Optimization Problems, in G. H. Tzeng, H. F. Wang, U. P. Wen e P. L. Yu (eds), *Proceedings of the Tenth International Conference on Multiple Criteria Decision Making: Expand and Enrich the Domains of Thinking and Application*, Vol. 1, Springer-Verlag, Berlin, pp. 283–294.

- Shirmohammadi, D. (1992). Service restoration in distribution networks via network reconfiguration, *IEEE Transactions on Power Delivery* **7**(2): 952–958.
- Shirmohammadi, D. e Hong, H. W. (1989). Reconfiguration of electric distribution networks for resistive line losses reduction, *IEEE Transactions on Power Delivery* **4**(2): 1492–1498.
- Silver, E. A., Vidal, R. V. V. e Werra, D. (1980). A tutorial on heuristic methods, *European Journal of Operational Research* **5**: 153–162.
- Srinivas, N. e Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation* **2**(3): 221–248.
- Steuer, R. E. (1986). *Multiple Criteria Optimization: Theory, Computation and Application*, New York: Wiley.
- Stroustrup, B. (1988). “What is Object-Oriented Programming?”, *IEEE Software* **5**(3): 10–20.
- Stroustrup, B. (2000). *The C++ Programming Language*, special 3rd edn, Addison-Wesley.
- Tenenbaum, A. M., Langsam, Y. e Augenstein, M. J. (1995). *Estruturas de dados usando C*, Makron Books.
- Toune, S., Fudo, H., Genji, T. e Fukuyama, Y. (2002). Comparative study of modern heuristic algorithms to service restoration in distribution systems, *IEEE Transactions on Power Delivery* **17**(1): 173–181.
- Toune, S., Fudo, H., Genji, T., Fukuyama, Y. e Nakanishi, Y. (1998). A reactive tabu search for resistive service restoration in electric power distribution systems, *IEEE International Conference on Evolutionary Computation*, pp. 1–7. Anchorage.
- Turban, E., Aronson, J. E. e Liang, T.-P. (2004). *Decision Support Systems and Intelligent Systems*, 7th edn, Prentice Hall.
- Ulungu, E. L. e Teghem, J. (1994). Multi-objective Combinatorial Optimization Problems: A Survey, *Journal of Multi-Criteria Decision Analysis* **3**: 83–101.
- Ulungu, E. L., Teghem, J. e Ost, C. (1998). Efficiency of interactive multi-objective simulated annealing through a case study, *Journal of the Operational Research Society* **49**: 1044–1050.

- Vidal, R. V. V. (1987). What is a decision support system?, *Relatório Técnico IMSOR Research Report 13/1987*, Technical University of Denmark.
- Zitzler, E., Laumanns, M. e Bleuler, S. (2004). A tutorial on evolutionary multiobjective optimization, in X. Gandibleux, M. Sevaux, K. Sörensen e V. T'kindt (eds), *Metaheuristics for Multiobjective Optimisation*, Vol. 535 of *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, pp. 3–38.
- Zitzler, E., Laumanns, M. e Thiele, L. (2002). SPEA2: Improving the Strength Pareto Evolutionary Algorithm, in K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou e T. Fogarty (eds), *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Athens, Greece, pp. 95–100.
- Zitzler, E. e Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the Strength Pareto approach, *IEEE Transactions on Evolutionary Computation* **3**(4): 257–271.

Apêndice A

Conceitos básicos da teoria dos grafos

Um grafo $G = (N, A)$ compreende um conjunto finito e não vazio de pontos denominados nós (N) e um conjunto finito e não vazio de pares de pontos denominados arcos (A). Se i e j pertencerem ao conjunto N e o par (i, j) pertencer ao conjunto A , diz-se que o arco (i, j) conecta i e j e (i, j) é dito incidente a i e j . Na Figura A.1(a), o arco $(1, 2)$ conecta os nós 1 e 2 e este arco é incidente a 1 e também a 2.

O *grau* de um nó corresponde ao número de arcos incidentes a ele. Na Figura A.1(a), o grau dos nós 1 e 4 é 2 e dos nós 2 e 3 é 3.

Um *subgrafo* H de $G = (N, A)$ é definido como $H = (N_H, A_H)$, onde $N_H \subseteq N$ e $A_H \subseteq A$.

Existem duas classes distintas de grafos, a dos *grafos orientados* e a dos *grafos não-orientados*. Um grafo orientado $G_o = (N, A_o)$ é formado por um conjunto de nós N e um conjunto de arcos A_o , cujos elementos são pares ordenados de nós distintos. Um grafo não-orientado $G_{no} = (N, A_{no})$ também apresenta um conjunto de nós tal como o tipo anterior, no entanto o conjunto de arcos inclui pares de nós não-ordenados e não necessariamente distintos. Logo, num grafo não-orientado um arco pode ser denotado por (i, j) ou (j, i) .

As Figuras A.1(a) e A.1(b) ilustram um grafo não-orientado e um grafo orientado, respectivamente. Em ambos, o conjunto de nós é o mesmo, $N = \{1, 2, 3, 4\}$. O conjunto de arcos, no entanto, é $A = \{(2, 1), (1, 3), (3, 2), (3, 4), (4, 2)\}$ para o grafo orientado da Figura A.1(b) e $A = \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 4)\}$ para o grafo não-orientado da Figura A.1(a). É usual para os grafos não-orientados chamar o conjunto A de arestas e não arcos, conceito este reservado aos grafos orientados.

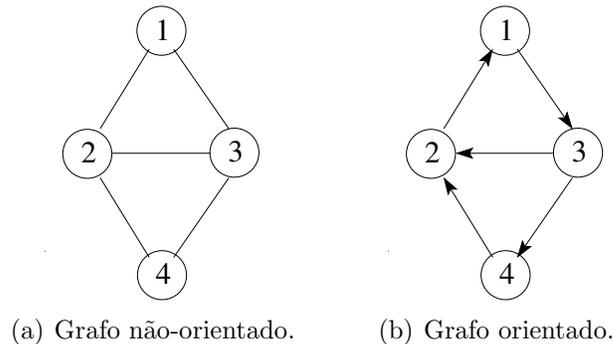


Figura A.1: Exemplo de grafo não-orientado e orientado.

Um *caminho* entre os nós i_1 e i_k em um grafo $G = (N, A)$ é uma seqüência de arcos $(i_1, i_2), (i_2, i_3), \dots, (i_{k-2}, i_{k-1}), (i_{k-1}, i_k) \in A$. Um caminho com k nós é formado por $k - 1$ arcos. Na Figura A.1(a), um caminho possível entre os nós 1 e 4 é dado por $(1, 3), (3, 4)$.

Um *ciclo* em um grafo $G = (N, A)$ é um caminho com o nó final igual ao nó inicial, ou seja, $i_1, i_2, \dots, i_{k-1}, i_k, i_1 \in N$, para $k \geq 3$. Caso o grafo G não possua ciclos, ele é chamado de grafo acíclico. Um exemplo de ciclo, a partir do grafo da Figura A.1(a), é a seqüência de arestas $(1, 3), (3, 2), (2, 1)$.

Um grafo $G = (N, A)$ é dito *conexo* se existir ao menos um caminho entre cada par de nós do conjunto N . Uma *árvore* é um grafo acíclico e conexo, se ela tiver n nós no conjunto N terá obrigatoriamente $n - 1$ arcos. Um grafo G (não conexo) formado por um conjunto de árvores é chamado de *floresta*.

É comum chamar um dos nós da árvore de *nó raiz*. Ele corresponde a uma referência e geralmente marca o início da árvore. O nó raiz pode ter grau maior ou igual a um. Os nós *terminais*, ou nós *folhas*, são aqueles nós com grau igual a um, com exceção do nó raiz.

Apêndice B

Resultados para o algoritmo evolutivo proposto

As Tabelas B.1-B.10 a seguir apresentam os resultados completos obtidos com a execução de um conjunto de combinações de valores para os parâmetros *número de gerações* (“Num.gerações”), *tamanho da população* (“Tam.pop.”) e *taxa de mutação* (“Taxa de mutação”) para cada uma das cinco redes da Tabela 9.1. Na quarta coluna de tabela é mostrado o número de soluções avaliadas pelo algoritmo e na quinta o tempo computacional em segundos. Na sexta e sétima colunas são apresentados os desvios percentuais do algoritmo construtivo (qualidade inicial) e do algoritmo evolutivo (qualidade final), respectivamente, em relação à melhor fronteira encontrada.

Num.gerações	Tam.pop.	Taxa de mutação	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
100	20	0,2	2.243	0,09	2,34	0
100	20	0,5	2.243	0,09	2,34	0
100	20	0,75	2.244	0,1	2,34	0
100	40	0,2	4.243	0,2	2,34	0
100	40	0,5	4.242	0,22	2,34	0
100	40	0,75	4.244	0,26	2,34	0
100	60	0,2	6.244	0,36	2,34	0
100	60	0,5	6.245	0,41	2,34	0
100	60	0,75	6.243	0,55	2,34	0
100	80	0,2	8.241	0,55	2,34	0
100	80	0,5	8.242	0,68	2,34	0
100	80	0,75	8.238	0,96	2,34	0
100	100	0,2	10.237	0,79	2,34	0
100	100	0,5	10.239	1,06	2,34	0
100	100	0,75	10.239	1,56	2,34	0
200	20	0,2	4.240	0,18	2,34	0
200	20	0,5	4.238	0,18	2,34	0
200	20	0,75	4.239	0,2	2,34	0
200	40	0,2	8.238	0,39	2,34	0
200	40	0,5	8.238	0,43	2,34	0
200	40	0,75	8.237	0,52	2,34	0
200	60	0,2	12.240	0,67	2,34	0
200	60	0,5	12.234	0,82	2,34	0
200	60	0,75	12.240	1,13	2,34	0
200	80	0,2	16.243	1,08	2,34	0
200	80	0,5	16.240	1,43	2,34	0
200	80	0,75	16.242	2	2,34	0
200	100	0,2	20.237	1,59	2,34	0
200	100	0,5	20.239	2,2	2,34	0
200	100	0,75	20.238	3,35	2,34	0
300	20	0,2	6.237	0,28	2,34	0
300	20	0,5	6.237	0,29	2,34	0
300	20	0,75	6.234	0,3	2,34	0
300	40	0,2	12.239	0,61	2,34	0
300	40	0,5	12.244	0,68	2,34	0
300	40	0,75	12.242	0,79	2,34	0
300	60	0,2	18.242	1,03	2,34	0
300	60	0,5	18.238	1,26	2,34	0
300	60	0,75	18.236	1,75	2,34	0
300	80	0,2	24.237	1,77	2,34	0
300	80	0,5	24.236	2,3	2,34	0
300	80	0,75	24.238	3,16	2,34	0
300	100	0,2	30.239	2,46	2,34	0
300	100	0,5	30.242	3,43	2,34	0
300	100	0,75	30.237	5,12	2,34	0

Tabela B.1: Primeira parte dos resultados da aplicação do AE para a rede 1 da Tabela 9.1.

Num.gerações	Tam.pop.	Taxa de mutação	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
400	20	0,2	8.246	0,35	2,34	0
400	20	0,5	8.243	0,36	2,34	0
400	20	0,75	8.243	0,4	2,34	0
400	40	0,2	16.243	0,79	2,34	0
400	40	0,5	16.236	0,91	2,34	0
400	40	0,75	16.243	1,07	2,34	0
400	60	0,2	24.245	1,41	2,34	0
400	60	0,5	24.239	1,7	2,34	0
400	60	0,75	24.238	2,27	2,34	0
400	80	0,2	32.241	2,18	2,34	0
400	80	0,5	32.238	2,94	2,34	0
400	80	0,75	32.240	4,11	2,34	0
400	100	0,2	40.239	3,2	2,34	0
400	100	0,5	40.238	4,67	2,34	0
400	100	0,75	40.240	6,93	2,34	0
500	20	0,2	10.240	0,44	2,34	0
500	20	0,5	10.240	0,45	2,34	0
500	20	0,75	10.235	0,49	2,34	0
500	40	0,2	20.235	0,98	2,34	0
500	40	0,5	20.241	1,16	2,34	0
500	40	0,75	20.235	1,34	2,34	0
500	60	0,2	30.244	1,76	2,34	0
500	60	0,5	30.241	2,21	2,34	0
500	60	0,75	30.235	2,87	2,34	0
500	80	0,2	40.243	2,74	2,34	0
500	80	0,5	40.234	3,74	2,34	0
500	80	0,75	40.241	5,43	2,34	0
500	100	0,2	50.240	4,04	2,34	0
500	100	0,5	50.235	5,77	2,34	0
500	100	0,75	50.237	8,96	2,34	0

Tabela B.2: Segunda parte dos resultados da aplicação do AE para a rede 1 da Tabela 9.1.

Num.gerações	Tam.pop.	Taxa de mutação	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
100	20	0,2	2.286	0,15	0	0
100	20	0,5	2.286	0,15	0	0
100	20	0,75	2.287	0,15	0	0
100	40	0,2	4.286	0,31	0	0
100	40	0,5	4.287	0,32	0	0
100	40	0,75	4.287	0,33	0	0
100	60	0,2	6.289	0,51	0	0
100	60	0,5	6.287	0,54	0	0
100	60	0,75	6.288	0,57	0	0
100	80	0,2	8.289	0,76	0	0
100	80	0,5	8.288	0,81	0	0
100	80	0,75	8.287	0,85	0	0
100	100	0,2	10.288	1,04	0	0
100	100	0,5	10.286	1,11	0	0
100	100	0,75	10.288	1,18	0	0
200	20	0,2	4.289	0,28	0	0
200	20	0,5	4.290	0,28	0	0
200	20	0,75	4.289	0,28	0	0
200	40	0,2	8.289	0,61	0	0
200	40	0,5	8.289	0,63	0	0
200	40	0,75	8.289	0,65	0	0
200	60	0,2	12.288	1,06	0	0
200	60	0,5	12.286	1,09	0	0
200	60	0,75	12.289	1,12	0	0
200	80	0,2	16.287	1,49	0	0
200	80	0,5	16.284	1,64	0	0
200	80	0,75	16.284	1,71	0	0
200	100	0,2	20.286	2,08	0	0
200	100	0,5	20.288	2,26	0	0
200	100	0,75	20.289	2,4	0	0
300	20	0,2	6.288	0,41	0	0
300	20	0,5	6.287	0,41	0	0
300	20	0,75	6.287	0,41	0	0
300	40	0,2	12.289	0,92	0	0
300	40	0,5	12.285	0,95	0	0
300	40	0,75	12.288	0,97	0	0
300	60	0,2	18.288	1,54	0	0
300	60	0,5	18.288	1,61	0	0
300	60	0,75	18.288	1,68	0	0
300	80	0,2	24.287	2,28	0	0
300	80	0,5	24.287	2,44	0	0
300	80	0,75	24.287	3,34	0	0
300	100	0,2	30.289	3,13	0	0
300	100	0,5	30.287	3,39	0	0
300	100	0,75	30.287	3,76	0	0

Tabela B.3: Primeira parte dos resultados da aplicação do AE para a rede 2 da Tabela 9.1.

Num.gerações	Tam.pop.	Taxa de mutação	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
400	20	0,2	8.285	0,55	0	0
400	20	0,5	8.288	0,55	0	0
400	20	0,75	8.286	0,55	0	0
400	40	0,2	16.286	1,21	0	0
400	40	0,5	16.286	1,25	0	0
400	40	0,75	16.288	1,3	0	0
400	60	0,2	24.287	2,06	0	0
400	60	0,5	24.290	2,16	0	0
400	60	0,75	24.290	2,26	0	0
400	80	0,2	32.286	3,04	0	0
400	80	0,5	32.287	3,23	0	0
400	80	0,75	32.285	3,42	0	0
400	100	0,2	40.285	4,19	0	0
400	100	0,5	40.289	4,54	0	0
400	100	0,75	40.289	4,83	0	0
500	20	0,2	10.287	0,68	0	0
500	20	0,5	10.290	0,68	0	0
500	20	0,75	10.289	0,68	0	0
500	40	0,2	20.285	1,52	0	0
500	40	0,5	20.286	1,56	0	0
500	40	0,75	20.289	1,61	0	0
500	60	0,2	30.287	2,6	0	0
500	60	0,5	30.289	2,68	0	0
500	60	0,75	30.287	2,81	0	0
500	80	0,2	40.287	3,8	0	0
500	80	0,5	40.286	4,06	0	0
500	80	0,75	40.286	4,3	0	0
500	100	0,2	50.286	5,31	0	0
500	100	0,5	50.285	5,73	0	0
500	100	0,75	50.291	6,04	0	0

Tabela B.4: Segunda parte dos resultados da aplicação do AE para a rede 2 da Tabela 9.1.

Num.gerações	Tam.pop.	Taxa de mutação	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
100	20	0,2	2.703	2,02	47,2	14,44
100	20	0,5	2.689	2,01	47,2	12,33
100	20	0,75	2.693	1,95	47,2	8,46
100	40	0,2	4.696	3,6	47,2	19,79
100	40	0,5	4.699	3,58	47,2	8,13
100	40	0,75	4.688	3,53	47,2	6,83
100	60	0,2	6.698	5,18	47,2	17,36
100	60	0,5	6.694	5,11	47,2	10,65
100	60	0,75	6.695	5,13	47,2	6,83
100	80	0,2	8.692	6,78	47,2	16,99
100	80	0,5	8.691	6,74	47,2	7,99
100	80	0,75	8.694	6,98	47,2	6,86
100	100	0,2	10.696	8,49	47,2	22,24
100	100	0,5	10.696	8,42	47,2	7,78
100	100	0,75	10.686	8,69	47,2	3,35
200	20	0,2	4.685	3,56	47,2	16,27
200	20	0,5	4.695	3,55	47,2	8,13
200	20	0,75	4.692	3,52	47,2	0,07
200	40	0,2	8.689	6,65	47,2	17,03
200	40	0,5	8.694	6,7	47,2	0,07
200	40	0,75	8.689	6,83	47,2	0,81
200	60	0,2	12.690	10	47,2	10,55
200	60	0,5	12.687	9,94	47,2	1,37
200	60	0,75	12.692	10,22	47,2	0,81
200	80	0,2	16.690	13,11	47,2	8,14
200	80	0,5	16.687	13,2	47,2	0,81
200	80	0,75	16.700	13,54	47,2	0,07
200	100	0,2	20.699	16,69	47,2	7,95
200	100	0,5	20.691	16,68	47,2	0,07
200	100	0,75	20.692	17,59	47,2	0,07
300	20	0,2	6.692	5,13	47,2	13,3
300	20	0,5	6.691	5,14	47,2	6,83
300	20	0,75	6.691	5,05	47,2	0,07
300	40	0,2	12.697	10,05	47,2	4,48
300	40	0,5	12.693	9,9	47,2	0,81
300	40	0,75	12.694	9,76	47,2	0
300	60	0,2	18.695	15,2	47,2	5,05
300	60	0,5	18.688	14,82	47,2	0,81
300	60	0,75	18.693	14,95	47,2	0,07
300	80	0,2	24.691	19,38	47,2	6,17
300	80	0,5	24.694	19,89	47,2	0
300	80	0,75	24.693	20,24	47,2	0
300	100	0,2	30.695	24,66	47,2	5
300	100	0,5	30.688	25,37	47,2	0,07
300	100	0,75	30.688	26,66	47,2	0

Tabela B.5: Primeira parte dos resultados da aplicação do AE para a rede 3 da Tabela 9.1.

Num.gerações	Tam.pop.	Taxa de mutação	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
400	20	0,2	8.688	6,82	47,2	13,37
400	20	0,5	8.693	6,67	47,2	1,76
400	20	0,75	8.690	6,64	47,2	1,37
400	40	0,2	16.695	12,88	47,2	2,89
400	40	0,5	16.692	13,06	47,2	1,37
400	40	0,75	16.696	12,95	47,2	0
400	60	0,2	24.692	19,42	47,2	0,81
400	60	0,5	24.689	19,53	48,08	0
400	60	0,75	24.680	20,16	47,2	0
400	80	0,2	32.696	27,58	47,2	0,07
400	80	0,5	32.694	26,44	47,2	0,07
400	80	0,75	32.695	27,51	47,2	0,07
400	100	0,2	40.693	32,68	47,2	2,48
400	100	0,5	40.697	33,69	47,2	0
400	100	0,75	40.698	35,52	47,2	0
500	20	0,2	10.702	8,21	47,2	6,62
500	20	0,5	10.691	8,15	47,2	1,29
500	20	0,75	10.694	8,09	47,2	1,76
500	40	0,2	20.701	16,1	47,2	12,52
500	40	0,5	20.688	16,29	47,2	0
500	40	0,75	20.691	16,36	47,2	0
500	60	0,2	30.690	24,06	47,2	0
500	60	0,5	30.700	24,92	47,2	0,07
500	60	0,75	30.695	25,13	47,2	0
500	80	0,2	40.697	32,24	47,2	1,29
500	80	0,5	40.693	32,88	47,2	0
500	80	0,75	40.692	34,02	47,2	0
500	100	0,2	50.694	40,63	47,2	0
500	100	0,5	50.695	42,61	47,2	0
500	100	0,75	50.693	44,87	47,2	0

Tabela B.6: Segunda parte dos resultados da aplicação do AE para a rede 3 da Tabela 9.1.

Num.gerações	Tam.pop.	Taxa de mutação	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
100	20	0,2	10.611	8,93	19,47	19,42
100	20	0,5	10.596	8,83	19,47	19,39
100	20	0,75	10.600	8,68	19,47	19,39
100	40	0,2	12.596	13,79	19,47	19,39
100	40	0,5	12.598	13,5	19,47	19,47
100	40	0,75	12.599	13,16	19,47	19,21
100	60	0,2	14.606	18,87	19,47	19,39
100	60	0,5	14.594	18,3	19,47	19,24
100	60	0,75	14.601	17,5	19,47	19,21
100	80	0,2	16.598	23,94	19,47	19,38
100	80	0,5	16.594	23,05	19,47	19,39
100	80	0,75	16.600	22,08	19,47	19,09
100	100	0,2	18.614	29,12	19,47	19,39
100	100	0,5	18.582	28,04	19,47	19,15
100	100	0,75	18.616	26,87	19,47	19,19
200	20	0,2	12.591	13,63	19,47	19,39
200	20	0,5	12.611	13,33	19,5	19,42
200	20	0,75	12.598	13,12	19,47	19,28
200	40	0,2	16.603	23,54	19,47	19,29
200	40	0,5	16.597	22,75	19,47	19,19
200	40	0,75	16.613	22,04	19,5	19,16
200	60	0,2	20.604	33,71	19,47	19,19
200	60	0,5	20.608	32,47	19,47	19,24
200	60	0,75	20.599	30,99	19,47	19,22
200	80	0,2	24.606	44,16	19,47	19,39
200	80	0,5	24.603	42,39	19,47	19,22
200	80	0,75	24.605	40,42	19,47	19,18
200	100	0,2	28.602	54,24	19,47	19,39
200	100	0,5	28.587	52,51	19,47	19,17
200	100	0,75	28.595	49,88	19,47	19,39
300	20	0,2	14.600	18,14	19,72	19,43
300	20	0,5	14.598	17,89	19,47	19,39
300	20	0,75	14.596	17,53	19,47	19,18
300	40	0,2	20.607	33,22	19,47	19,19
300	40	0,5	20.592	32,14	19,72	19,19
300	40	0,75	20.593	31,72	19,47	19,18
300	60	0,2	26.601	48,77	19,47	19,39
300	60	0,5	26.605	46,6	19,62	19,19
300	60	0,75	26.616	44,67	19,47	19,11
300	80	0,2	32.601	63,78	19,47	19,45
300	80	0,5	32.601	61,08	19,47	19,11
300	80	0,75	32.602	58,66	19,62	19,26
300	100	0,2	38.593	79,26	19,47	19,39
300	100	0,5	38.601	76,81	19,47	19,22
300	100	0,75	38.600	73,41	19,47	19,13

Tabela B.7: Primeira parte dos resultados da aplicação do AE para a rede 4 da Tabela 9.1.

Num.gerações	Tam.pop.	Taxa de mutação	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
400	20	0,2	16.621	22,94	19,47	19,39
400	20	0,5	16.596	22,45	19,5	19,22
400	20	0,75	16.617	22,2	19,48	19,2
400	40	0,2	24.602	42,86	19,47	19,39
400	40	0,5	24.591	41,46	19,47	19,21
400	40	0,75	24.601	40,15	19,47	19,09
400	60	0,2	32.603	63,68	19,47	19,15
400	60	0,5	32.588	60,66	19,47	19,09
400	60	0,75	32.586	58,83	19,48	19,11
400	80	0,2	40.604	83,7	19,63	19,38
400	80	0,5	40.601	80,6	19,47	19,17
400	80	0,75	40.598	77,18	19,47	19,08
400	100	0,2	48.611	105,18	19,6	19,37
400	100	0,5	48.591	101,24	19,47	19,18
400	100	0,75	48.603	96,57	19,47	19,11
500	20	0,2	18.597	27,83	19,47	19,22
500	20	0,5	18.596	27,14	19,47	19,39
500	20	0,75	18.589	26,45	19,47	19,22
500	40	0,2	28.612	53,75	19,53	19,4
500	40	0,5	28.590	51,13	19,47	19,19
500	40	0,75	28.597	49,48	19,47	19,18
500	60	0,2	38.598	79,31	19,47	19,19
500	60	0,5	38.611	74,97	19,47	19,12
500	60	0,75	38.609	71,78	19,48	19,12
500	80	0,2	48.580	104,93	19,5	19,21
500	80	0,5	48.596	101,41	19,47	19,13
500	80	0,75	48.590	95,24	19,47	19,09
500	100	0,2	58.596	131,1	19,48	19,25
500	100	0,5	58.599	125,68	19,47	19,13
500	100	0,75	58.608	119,62	19,47	19,09

Tabela B.8: Segunda parte dos resultados da aplicação do AE para a rede 4 da Tabela 9.1.

Num.gerações	Tam.pop.	Taxa de mutação	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
100	20	0,2	20.478	31,11	22,94	21,46
100	20	0,5	20.446	31,65	24,52	23,05
100	20	0,75	20.387	31,24	20,88	19,74
100	40	0,2	22.455	43,93	23,35	22,49
100	40	0,5	22.331	43,27	23,08	21,74
100	40	0,75	22.443	42,46	26,61	25,63
100	60	0,2	24.333	55,97	26,65	24,63
100	60	0,5	24.365	55,18	25,19	24,17
100	60	0,75	24.346	53,86	23,47	21,39
100	80	0,2	26.428	68,15	24,61	23,05
100	80	0,5	26.354	67,1	23,71	20,95
100	80	0,75	26.351	65,63	25,31	23,82
100	100	0,2	28.397	81,14	25,27	24,49
100	100	0,5	28.381	79,1	21,94	20,68
100	100	0,75	28.460	77,42	24	21,78
200	20	0,2	22.286	43,61	25,13	23,67
200	20	0,5	22.464	42,95	21,79	19,23
200	20	0,75	22.387	42,45	21,9	20,21
200	40	0,2	26.395	68,34	21,95	19,74
200	40	0,5	26.469	66,83	22,67	21,59
200	40	0,75	26.352	65,94	26,49	24,49
200	60	0,2	30.382	92,99	26,06	25,2
200	60	0,5	30.371	90,97	25,42	24,44
200	60	0,75	30.372	89,66	26,32	20,91
200	80	0,2	34.398	118,47	21,64	20,34
200	80	0,5	34.351	120,93	25,06	20,39
200	80	0,75	34.394	115,27	22,79	19,6
200	100	0,2	38.425	142,56	23,31	21,69
200	100	0,5	38.373	139,54	24,36	22,76
200	100	0,75	38.272	137,26	26,34	17,71
300	20	0,2	24.363	55,95	26,57	24,87
300	20	0,5	24.292	54,93	25,01	23,81
300	20	0,75	24.357	54,39	25,3	21,4
300	40	0,2	30.356	93,22	26,13	25,43
300	40	0,5	30.351	91,67	24,24	22,28
300	40	0,75	30.282	89,69	23,27	17,1
300	60	0,2	36.432	130,12	22,59	21,57
300	60	0,5	36.459	127,42	25,26	21,78
300	60	0,75	36.414	126,45	23,32	16,93
300	80	0,2	42.440	167,57	24,09	22
300	80	0,5	42.384	163,93	23,73	22,52
300	80	0,75	42.446	160,94	24,38	16,99
300	100	0,2	48.414	205,12	21,79	20,82
300	100	0,5	48.343	200,92	26,92	20,21
300	100	0,75	48.460	199,3	21,24	17,11

Tabela B.9: Primeira parte dos resultados da aplicação do AE para a rede 5 da Tabela 9.1.

Num.gerações	Tam.pop.	Taxa de mutação	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
400	20	0,2	26.404	68,63	23,59	23,25
400	20	0,5	26.448	68,31	22,51	20,27
400	20	0,75	26.416	66,03	23,83	21,87
400	40	0,2	34.499	117,99	22,82	21,96
400	40	0,5	34.482	116,56	24,89	22,07
400	40	0,75	34.357	113,07	26,48	20,92
400	60	0,2	42.351	166,65	25,77	25,24
400	60	0,5	42.438	164,08	25,03	19,72
400	60	0,75	42.385	161,79	22,86	15,44
400	80	0,2	50.377	217,15	24,51	22,8
400	80	0,5	50.463	213,25	23,19	18,73
400	80	0,75	50.449	210,6	20,36	14,62
400	100	0,2	58.456	267,12	23,72	22,23
400	100	0,5	58.373	261,91	21,77	18,51
400	100	0,75	58.453	257,18	23,11	16,78
500	20	0,2	28.323	80,96	23,49	23,14
500	20	0,5	28.409	80,87	25,33	21,51
500	20	0,75	28.334	78,42	23,08	20,16
500	40	0,2	38.471	142,25	23,37	21,46
500	40	0,5	38.409	139,64	23,59	20,72
500	40	0,75	38.393	136,93	26,52	18,82
500	60	0,2	48.349	205,47	25,06	23,26
500	60	0,5	48.400	200,67	22,53	20,7
500	60	0,75	48.319	196,39	25,76	20,31
500	80	0,2	58.339	268,25	26,35	24,31
500	80	0,5	58.375	261,28	24,5	20,19
500	80	0,75	58.427	258,84	24,63	19,94
500	100	0,2	68.446	331,69	26,43	23,15
500	100	0,5	68.388	325,28	23,01	19,2
500	100	0,75	68.415	319,89	23,6	14,19

Tabela B.10: Segunda parte dos resultados da aplicação do AE para a rede 5 da Tabela 9.1.

Apêndice C

Resultados para a busca tabu proposta

As Tabelas C.1-C.10 a seguir apresentam os resultados completos obtidos com a execução de um conjunto de combinações de valores para os parâmetros *número de iterações* (“Num.iterações”), *duração tabu* e *número de vizinhanças em paralelo* (“Num.viz.paral.”) para cada uma das cinco redes da Tabela 9.1. Na quarta coluna da tabela é mostrado o número de soluções avaliadas pelo algoritmo e na quinta o tempo computacional em segundos. Na sexta e sétima colunas são apresentados os desvios percentuais do algoritmo construtivo (qualidade inicial) e da busca tabu (qualidade final), respectivamente, em relação à melhor fronteira encontrada.

Num.iterações	Duração Tabu	Num.viz. paral.	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
100	10	2	9.264	0,25	2,34	0
100	10	4	18.216	0,48	2,34	0
100	10	6	27.084	0,68	2,34	0
100	20	2	9.305	0,25	2,34	0
100	20	4	18.219	0,46	2,34	0
100	20	6	27.035	0,67	2,34	0
100	30	2	9.283	0,25	2,34	0
100	30	4	18.235	0,46	2,34	0
100	30	6	27.044	0,68	2,34	0
100	40	2	9.304	0,26	2,34	0
100	40	4	18.236	0,46	2,34	0
100	40	6	27.042	0,67	2,34	0
100	50	2	9.296	0,25	2,34	0
100	50	4	18.229	0,46	2,34	0
100	50	6	27.069	0,69	2,34	0
200	10	2	18.315	0,49	2,34	0
200	10	4	36.148	0,92	2,34	0
200	10	6	53.616	1,31	2,34	0
200	20	2	18.295	0,49	2,34	0
200	20	4	36.052	0,92	2,34	0
200	20	6	53.621	1,35	2,34	0
200	30	2	18.320	0,5	2,34	0
200	30	4	36.038	0,91	2,34	0
200	30	6	53.689	1,35	2,34	0
200	40	2	18.298	0,49	2,34	0
200	40	4	36.021	0,92	2,34	0
200	40	6	53.696	1,34	2,34	0
200	50	2	18.267	0,49	2,34	0
200	50	4	36.103	0,93	2,34	0
200	50	6	53.748	1,34	2,34	0
300	10	2	27.341	0,73	2,34	0
300	10	4	53.840	1,38	2,34	0
300	10	6	80.225	2,01	2,34	0
300	20	2	27.248	0,75	2,34	0
300	20	4	53.833	1,38	2,34	0
300	20	6	80.302	2,01	2,34	0
300	30	2	27.310	0,73	2,34	0
300	30	4	53.915	1,37	2,34	0
300	30	6	80.234	1,99	2,34	0
300	40	2	27.328	0,74	2,34	0
300	40	4	53.829	1,36	2,34	0
300	40	6	80.221	2	2,34	0
300	50	2	27.345	0,73	2,34	0
300	50	4	53.818	1,38	2,34	0
300	50	6	80.208	2	2,34	0

Tabela C.1: Primeira parte dos resultados da aplicação da BT para a rede 1 da Tabela 9.1.

Num.iterações	Duração Tabu	Num.viz. paral.	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
400	10	2	36.363	0,98	2,34	0
400	10	4	71.972	1,84	2,34	0
400	10	6	107.174	2,67	2,34	0
400	20	2	36.292	0,97	2,34	0
400	20	4	71.718	1,81	2,34	0
400	20	6	107.017	2,63	2,34	0
400	30	2	36.383	0,97	2,34	0
400	30	4	71.939	1,82	2,34	0
400	30	6	107.148	2,63	2,34	0
400	40	2	36.358	0,98	2,34	0
400	40	4	71.706	1,81	2,34	0
400	40	6	106.946	2,62	2,34	0
400	50	2	36.266	0,97	2,34	0
400	50	4	71.856	1,8	2,34	0
400	50	6	106.970	2,64	2,34	0
500	10	2	45.324	1,2	2,34	0
500	10	4	89.946	2,23	2,34	0
500	10	6	133.833	3,25	2,34	0
500	20	2	45.274	1,17	2,34	0
500	20	4	89.629	2,23	2,34	0
500	20	6	133.613	3,27	2,34	0
500	30	2	45.385	1,19	2,34	0
500	30	4	89.452	2,25	2,34	0
500	30	6	133.481	3,29	2,34	0
500	40	2	45.341	1,2	2,34	0
500	40	4	89.690	2,23	2,34	0
500	40	6	133.589	3,31	2,34	0
500	50	2	45.341	1,21	2,34	0
500	50	4	89.659	2,24	2,34	0
500	50	6	133.624	3,29	2,34	0

Tabela C.2: Segunda parte dos resultados da aplicação da BT para a rede 1 da Tabela 9.1.

Num.iterações	Duração Tabu	Num.viz. paral.	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
100	10	2	4.929	0,18	0	0
100	10	4	9.762	0,32	0	0
100	10	6	14.380	0,44	0	0
100	20	2	4.947	0,19	0	0
100	20	4	9.811	0,33	0	0
100	20	6	14.309	0,46	0	0
100	30	2	4.903	0,19	0	0
100	30	4	9.828	0,33	0	0
100	30	6	14.142	0,44	0	0
100	40	2	4.964	0,18	0	0
100	40	4	9.828	0,32	0	0
100	40	6	14.277	0,45	0	0
100	50	2	4.948	0,18	0	0
100	50	4	9.804	0,33	0	0
100	50	6	14.599	0,45	0	0
200	10	2	9.522	0,35	0	0
200	10	4	19.290	0,62	0	0
200	10	6	28.682	0,89	0	0
200	20	2	9.598	0,33	0	0
200	20	4	19.146	0,61	0	0
200	20	6	28.015	0,85	0	0
200	30	2	9.552	0,33	0	0
200	30	4	19.333	0,6	0	0
200	30	6	28.690	0,85	0	0
200	40	2	9.597	0,33	0	0
200	40	4	19.348	0,59	0	0
200	40	6	28.584	0,88	0	0
200	50	2	9.475	0,33	0	0
200	50	4	19.301	0,61	0	0
200	50	6	28.344	0,85	0	0
300	10	2	14.165	0,48	0	0
300	10	4	28.719	0,88	0	0
300	10	6	42.180	1,26	0	0
300	20	2	14.300	0,49	0	0
300	20	4	28.947	0,9	0	0
300	20	6	42.226	1,28	0	0
300	30	2	14.083	0,49	0	0
300	30	4	29.013	0,92	0	0
300	30	6	42.342	1,29	0	0
300	40	2	14.298	0,48	0	0
300	40	4	28.694	0,88	0	0
300	40	6	42.212	1,26	0	0
300	50	2	14.161	0,48	0	0
300	50	4	28.889	0,89	0	0
300	50	6	42.308	1,26	0	0

Tabela C.3: Primeira parte dos resultados da aplicação da BT para a rede 2 da Tabela 9.1.

Num.iterações	Duração Tabu	Num.viz. paral.	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
400	10	2	18.836	0,63	0	0
400	10	4	38.349	1,21	0	0
400	10	6	56.423	1,69	0	0
400	20	2	18.871	0,65	0	0
400	20	4	38.288	1,2	0	0
400	20	6	56.765	1,72	0	0
400	30	2	18.846	0,65	0	0
400	30	4	38.280	1,2	0	0
400	30	6	56.216	1,73	0	0
400	40	2	18.960	0,65	0	0
400	40	4	38.770	1,22	0	0
400	40	6	56.582	1,72	0	0
400	50	2	18.896	0,65	0	0
400	50	4	37.972	1,17	0	0
400	50	6	55.845	1,72	0	0
500	10	2	23.536	0,8	0	0
500	10	4	48.335	1,51	0	0
500	10	6	70.127	2,13	0	0
500	20	2	23.419	0,8	0	0
500	20	4	47.954	1,5	0	0
500	20	6	70.691	2,18	0	0
500	30	2	23.436	0,8	0	0
500	30	4	47.665	1,49	0	0
500	30	6	70.157	2,12	0	0
500	40	2	23.410	0,8	0	0
500	40	4	48.252	1,52	0	0
500	40	6	70.565	2,18	0	0
500	50	2	23.549	0,81	0	0
500	50	4	47.692	1,51	0	0
500	50	6	70.066	2,13	0	0

Tabela C.4: Segunda parte dos resultados da aplicação da BT para a rede 2 da Tabela 9.1.

Num.iterações	Duração Tabu	Num.viz. paral.	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
100	10	2	9.432	5	47,2	14,03
100	10	4	17.981	8,74	47,2	2,68
100	10	6	26.659	12,46	47,2	2,68
100	20	2	9.457	5,06	47,2	13,38
100	20	4	17.891	8,95	47,2	2,68
100	20	6	26.693	12,5	47,2	2,68
100	30	2	9.394	5,24	47,2	12,3
100	30	4	18.043	9,04	47,2	2,68
100	30	6	26.699	12,59	47,2	2,68
100	40	2	9.354	5,17	47,2	22,4
100	40	4	17.979	8,55	47,2	2,68
100	40	6	26.647	12,84	47,2	2,68
100	50	2	9.377	5,13	47,2	20,32
100	50	4	17.877	8,96	47,2	2,68
100	50	6	26.698	12,87	47,2	2,68
200	10	2	18.087	9,56	47,2	15,1
200	10	4	35.107	16,16	47,2	2,68
200	10	6	52.429	23,45	47,2	2,68
200	20	2	18.059	8,98	47,2	2,68
200	20	4	35.136	16,75	47,2	2,68
200	20	6	52.382	23,99	47,2	2,68
200	30	2	18.129	9,46	47,2	2,68
200	30	4	35.116	16,62	47,2	2,68
200	30	6	52.322	24,56	47,2	2,68
200	40	2	18.036	9,42	47,2	11,92
200	40	4	35.029	16,63	47,2	2,68
200	40	6	52.326	23,45	47,2	2,68
200	50	2	18.048	9,09	47,2	18,2
200	50	4	35.090	15,91	47,2	2,68
200	50	6	52.427	23,39	47,2	2,68
300	10	2	26.676	12,89	47,2	2,68
300	10	4	52.195	24,41	47,2	2,68
300	10	6	78.125	33,99	47,2	2,68
300	20	2	26.771	12,87	47,7	2,68
300	20	4	52.195	23,73	47,2	2,68
300	20	6	78.193	35,34	47,2	2,68
300	30	2	26.859	13,55	47,2	2,68
300	30	4	52.202	24,01	47,2	2,68
300	30	6	77.997	36,2	47,2	2,68
300	40	2	26.823	13,5	47,2	10,64
300	40	4	52.230	22,28	47,2	2,68
300	40	6	77.864	33,61	47,2	2,68
300	50	2	26.860	12,7	47,2	2,68
300	50	4	52.158	24,82	47,2	2,68
300	50	6	78.077	34,86	47,2	2,68

Tabela C.5: Primeira parte dos resultados da aplicação da BT para a rede 3 da Tabela 9.1.

Num.iterações	Duração Tabu	Num.viz. paral.	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
400	10	2	35.630	16,98	47,2	2,68
400	10	4	69.316	30,92	47,2	2,68
400	10	6	103.839	45,46	47,2	2,68
400	20	2	35.490	17,85	47,2	2,68
400	20	4	69.432	32,13	47,2	2,68
400	20	6	103.746	46,54	47,2	2,68
400	30	2	35.362	17,12	47,2	2,68
400	30	4	69.457	31,84	47,2	2,68
400	30	6	103.780	44,79	47,2	2,68
400	40	2	35.370	17,01	47,2	2,68
400	40	4	69.200	31,69	47,2	2,68
400	40	6	103.781	45,72	47,2	2,68
400	50	2	35.591	16,68	47,2	2,68
400	50	4	69.382	31,56	47,2	2,68
400	50	6	103.717	46,58	47,2	2,68
500	10	2	44.336	21,62	47,2	2,68
500	10	4	86.484	39,07	47,2	2,68
500	10	6	129.493	56,81	47,2	2,68
500	20	2	44.229	21,52	47,2	2,68
500	20	4	86.498	38,44	47,2	2,68
500	20	6	129.578	56,68	47,2	2,68
500	30	2	44.264	21,42	47,2	2,68
500	30	4	86.260	39,57	47,2	2,68
500	30	6	129.522	57,52	47,2	2,68
500	40	2	44.220	21,37	47,2	2,68
500	40	4	86.409	39,91	47,2	2,68
500	40	6	129.731	56,83	47,2	2,68
500	50	2	44.068	21,62	47,2	16,97
500	50	4	86.431	38,7	47,2	2,68
500	50	6	129.649	57,63	47,2	2,68

Tabela C.6: Segunda parte dos resultados da aplicação da BT para a rede 3 da Tabela 9.1.

Num.iterações	Duração Tabu	Num.viz. paral.	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
100	10	2	24.884	21	19,47	13,95
100	10	4	41.652	34,99	19,49	3,29
100	10	6	58.182	48,02	19,47	3,64
100	20	2	24.811	21,46	19,47	13,01
100	20	4	41.601	35,73	19,47	4,85
100	20	6	58.175	48,19	19,47	1,02
100	30	2	24.821	22,18	19,62	5,18
100	30	4	41.811	35,87	19,66	5,13
100	30	6	58.258	49,24	19,47	3,42
100	40	2	24.862	22,22	19,72	12,37
100	40	4	41.731	36,48	19,47	4,45
100	40	6	58.382	47,85	19,47	2,85
100	50	2	24.885	22,15	19,63	14,64
100	50	4	41.479	35,55	19,72	3,45
100	50	6	57.756	48,89	19,5	4,49
200	10	2	41.046	35,67	19,47	13,12
200	10	4	74.518	63,61	19,62	1,4
200	10	6	107.345	90,07	19,47	1,24
200	20	2	41.109	36,82	19,72	7,66
200	20	4	73.555	60,91	19,6	6,04
200	20	6	107.977	90,24	19,49	1,86
200	30	2	40.951	36,01	19,61	9,31
200	30	4	74.110	61,8	19,47	1,15
200	30	6	106.560	90,3	19,75	0,62
200	40	2	41.253	36,38	19,47	12,78
200	40	4	74.462	62,1	19,47	2,54
200	40	6	106.921	89,19	19,47	1,26
200	50	2	41.008	35,99	19,47	11,25
200	50	4	74.779	64,05	19,47	0,77
200	50	6	107.613	89,66	19,48	2,12
300	10	2	57.653	50,7	19,48	3,07
300	10	4	107.055	87,53	19,72	3,07
300	10	6	156.565	129,7	19,47	0
300	20	2	57.498	50,22	19,47	6,61
300	20	4	108.234	90,43	19,63	3,12
300	20	6	155.421	128,92	19,51	2,27
300	30	2	57.419	49,84	19,6	6,36
300	30	4	107.018	93,41	19,47	2,97
300	30	6	156.477	135,28	19,47	0,56
300	40	2	57.205	51,95	19,47	12,77
300	40	4	106.930	95,97	19,72	2,65
300	40	6	155.665	132,34	19,62	1,24
300	50	2	57.249	52,07	19,51	13,27
300	50	4	107.313	96,76	19,47	2,05
300	50	6	155.757	138,1	19,49	3,35

Tabela C.7: Primeira parte dos resultados da aplicação da BT para a rede 4 da Tabela 9.1.

Num.iterações	Duração Tabu	Num.viz. paral.	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
400	10	2	73.957	65,39	19,48	10,49
400	10	4	140.469	122,17	19,53	3,17
400	10	6	205.623	175,13	19,5	0,87
400	20	2	73.318	67,24	19,47	10,5
400	20	4	140.542	124,42	19,6	0,53
400	20	6	205.514	179,44	19,5	0,69
400	30	2	73.255	66,76	19,5	9,41
400	30	4	140.670	124,12	19,63	0,3
400	30	6	205.178	177,49	19,47	1,34
400	40	2	74.030	66,71	19,5	4,85
400	40	4	140.218	121,57	19,47	1,24
400	40	6	205.381	180,52	19,48	1,22
400	50	2	73.716	66,63	19,6	8,38
400	50	4	139.851	123,12	19,77	1,55
400	50	6	205.792	180,6	19,53	0,75
500	10	2	90.166	82,84	19,63	7,2
500	10	4	172.123	152,82	19,53	1,86
500	10	6	255.011	220,89	19,47	1,54
500	20	2	89.629	83,58	19,47	8,22
500	20	4	173.751	153,95	19,72	1,1
500	20	6	253.553	218,21	19,66	2,02
500	30	2	89.282	79,78	19,5	5,62
500	30	4	173.405	156,03	19,47	0,92
500	30	6	255.120	221,47	19,62	0,35
500	40	2	90.109	79,19	19,76	8,15
500	40	4	173.021	153,56	19,47	1,7
500	40	6	254.448	223,92	19,49	0,47
500	50	2	90.060	82,48	19,49	7,95
500	50	4	174.510	151,58	19,62	1,41
500	50	6	254.542	223,7	19,72	1,65

Tabela C.8: Segunda parte dos resultados da aplicação da BT para a rede 4 da Tabela 9.1.

Num.iterações	Duração Tabu	Num.viz. paral.	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
100	10	2	56.411	54,21	26,35	24,75
100	10	4	94.417	118,09	26,94	23,25
100	10	6	137.021	169,27	25,42	12,02
100	20	2	56.482	77,53	23,98	19,69
100	20	4	95.638	121,34	19,13	14,7
100	20	6	135.839	170,87	25,96	8,15
100	30	2	56.338	76,83	22,71	20,54
100	30	4	95.205	122,97	24,11	17,3
100	30	6	137.388	173,37	24,24	9,41
100	40	2	56.249	75,36	26,52	22,27
100	40	4	95.563	123,9	26	14,91
100	40	6	135.602	169,73	25,13	13,65
100	50	2	56.757	77,8	27,03	22,36
100	50	4	96.976	125,69	26,06	16,74
100	50	6	137.338	171,47	27,01	9,78
200	10	2	94.363	123,03	25,11	21,33
200	10	4	170.994	212,6	23,7	15,4
200	10	6	255.849	314,15	23,2	5,56
200	20	2	94.289	96,06	25,31	19,74
200	20	4	171.255	167,56	24,93	15,32
200	20	6	253.040	229,16	25,08	6,9
200	30	2	95.038	86,22	22,9	18,06
200	30	4	173.828	154,21	26,73	9,51
200	30	6	255.061	222,5	23,22	6,92
200	40	2	94.861	85,96	23,67	20,21
200	40	4	174.966	153,79	23,8	8,68
200	40	6	257.112	225,61	22,84	6,71
200	50	2	94.423	85,65	24,14	19,64
200	50	4	173.564	154,19	26,51	9,47
200	50	6	254.781	222,81	25,24	5,82
300	10	2	133.303	119,87	25,63	22,06
300	10	4	252.267	222,81	26,05	10,05
300	10	6	370.039	320,6	26,23	3,14
300	20	2	131.798	118,01	27,17	22,18
300	20	4	253.126	222,51	24,54	8,05
300	20	6	372.936	322,47	25,86	5,12
300	30	2	130.490	121,48	25,45	20,32
300	30	4	249.321	216,89	26,17	4,84
300	30	6	372.413	321,79	22,2	3,17
300	40	2	131.325	116,95	24,51	20,84
300	40	4	249.027	216,15	28,1	8,4
300	40	6	376.809	325,42	26,53	3,01
300	50	2	132.726	119,03	28,25	24,35
300	50	4	254.300	235,83	22,31	4,69
300	50	6	365.965	283,79	26,49	5,88

Tabela C.9: Primeira parte dos resultados da aplicação da BT para a rede 5 da Tabela 9.1.

Num.iterações	Duração Tabu	Num.viz. paral.	Num.sol. avaliadas	Tempo (s)	Qualidade inicial (%)	Qualidade final (%)
400	10	2	170.390	150,66	25,02	18,16
400	10	4	334.680	290,5	25,16	2,62
400	10	6	499.160	429,82	28,37	1,14
400	20	2	172.354	151,73	25,03	19,14
400	20	4	326.721	280,03	24,07	4,54
400	20	6	494.183	454,82	26,36	3,19
400	30	2	168.950	160,82	24,64	20,12
400	30	4	333.119	309,39	25,31	5,93
400	30	6	496.751	454,61	23,95	1,95
400	40	2	171.577	152,3	20,81	18,7
400	40	4	323.722	278,63	22,75	8,06
400	40	6	498.254	486,06	27,19	2,57
400	50	2	171.469	156,1	27,19	17,09
400	50	4	324.445	286,42	23,48	6,15
400	50	6	491.150	377	24,64	2,43
500	10	2	211.177	190,13	23,92	15,35
500	10	4	408.104	360,16	23,36	3,11
500	10	6	602.088	523,05	22,25	0
500	20	2	210.134	188,84	25,71	16,76
500	20	4	401.831	350,66	27,84	6,19
500	20	6	611.235	553,13	24,65	0,54
500	30	2	205.627	189,47	24,02	18,39
500	30	4	402.297	347,92	25,93	6,3
500	30	6	612.005	523,17	26,94	1,36
500	40	2	208.783	187,38	26,22	21,44
500	40	4	409.749	351,2	26,43	3,03
500	40	6	608.967	528,52	24,28	1,67
500	50	2	208.350	182,7	25,31	18,3
500	50	4	405.663	346,82	25,19	3,82
500	50	6	614.680	527,99	23,06	2,85

Tabela C.10: Segunda parte dos resultados da aplicação da BT para a rede 5 da Tabela 9.1.