

Algoritmo para obtenção de planos de restabelecimento para sistemas de distribuição de grande porte

Moussa Reda Mansour

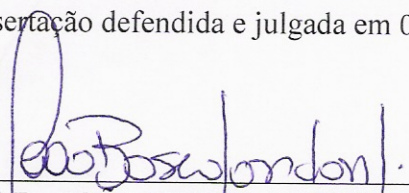
Dissertação apresentada à Escola de Engenharia de São Carlos, da Universidade de São Paulo, como parte dos requisitos para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. João Bosco Augusto London Junior

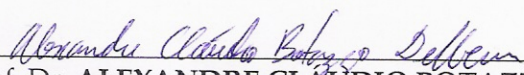
FOLHA DE JULGAMENTO

Candidato: Bacharel **MOUSSA REDA MANSOUR**

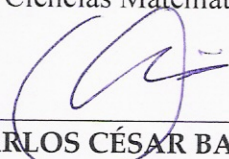
Dissertação defendida e julgada em 03/04/2009 perante a Comissão Julgadora:



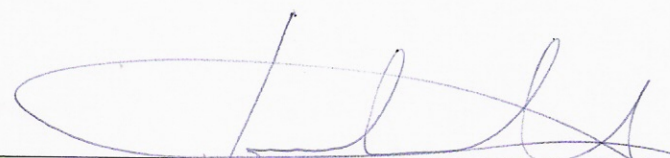
Prof. Dr. **JOÃO BOSCO AUGUSTO LONDON JÚNIOR (Orientador)**
(Escola de Engenharia de São Carlos/USP) APROVADO



Prof. Dr. **ALEXANDRE CLÁUDIO BOTAZZO DELBEM**
(Instituto de Ciências Matemáticas e de Computação/USP) APROVADO



Prof. Dr. **CARLOS CÉSAR BARIONI DE OLIVEIRA**
(Escola Politécnica/USP) APROVADO



Prof. Associado **GERALDO ROBERTO MARTINS DA COSTA**
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica e
Presidente da Comissão de Pós-Graduação

RESCURS
Serviço de Pós-Graduação
Protocolado em 03/04/09
12/04/09

“Dedico este trabalho ao meu avô Moussa (in memoriam).”

*“Lutar e vencer todas as batalhas não é a glória suprema.
A glória suprema consiste em quebrar a resistência do inimigo sem lutar.”
(Sun Tzu)*

Agradecimentos

Aos meus pais, Reda e Fatme, pelo amor
e pela confiança que me deram.

As minhas irmãs, Noura, Sara e Eva por todo o
apoio moral e carinho que me passaram.

Ao meu cunhado Shadi pela valiosa amizade que criamos.

Ao professor Dr. João Bosco A. London Jr. pela orientação,
dedicação e confiança para o sucesso do projeto de pesquisa.

Ao professor Dr. Alexandre Cláudio B. Delbem,
pela co-orientação e apoio neste projeto de pesquisa.

Ao meu grande amigo Narnia (Marcelo Nanni) por me apresentar
o maravilhoso mundo dos Sistemas Elétricos de Potência.

Aos professores do LACo-SEP (Laboratório de Análise
Computacional em Sistemas Elétricos de Potência), Newton Geraldo Bretas,
Luís Fernando Costa alberto e Rodrigo Andrade Ramos.

Aos meus amigos e colegas Escama (Marcelo Castoldi), Doido (Robson Dutra),
Perninha (Raphael Bedito), Valdick Soriano (Leandro Brolin),

A lenda (Elmer), Pedrão Fenômeno (Pedro), Perdigão (Rodrigo Salim),
Banqueiro (Rafael Borges), Zero 2 (Fabiolo), Aderbaalll (Carlisson), Krow (Carol),

Augustus (Augusto), Jaja (Saulo), Madlein (Madaleine), Marceleza (Marcelo),
Cabelera (Marcel), Prodígio (Edson), Japoneis Doido (Marcelo Suetake),
Gordin (Gabriel), Mulher Maravilha (Karen), Maranhão (Antonio) e Fabin (Fabinho).

A todos meus amigos e colegas cujo nome não foi citado acima.

E à FAPESP, pelo apoio financeiro.

Resumo

MANSOUR, M. R., Algoritmo para obtenção de planos de restabelecimento para sistemas de distribuição de grande porte. São Carlos, 2009, 110p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.

A elaboração de planos de restabelecimento de energia (PRE) de forma rápida, para re-energização de sistemas de distribuição radiais (SDR), faz-se necessária para lidar com situações que deixam regiões dos SDR sem energia. Tais situações podem ser causadas por faltas permanentes ou pela necessidade de isolar zonas dos SDR para serviços de manutenção.

Dentre os objetivos de um PRE, destacam-se: (i) Reduzir o número de consumidores interrompidos (ou nenhum), e (ii) minimizar o número de manobras; que devem ser atendidos sem desprezar os limites operacionais dos equipamentos. Conseqüentemente, a obtenção de PRE em SDR é um problema com múltiplos objetivos, alguns conflitantes.

As principais técnicas desenvolvidas para obtenção de PRE em SDR baseiam-se em Algoritmos Evolutivos (AE). A limitação da maioria dessas técnicas é a necessidade de simplificações na rede, para lidar com SDR de grande porte, que limitam consideravelmente a possibilidade de obtenção de um PRE adequado.

Propõe-se, neste trabalho, o desenvolvimento e implantação computacional de um algoritmo para obtenção de PRE em SDR, que consiga lidar com sistemas de grande porte sem a necessidade de simplificações, isto é, considerando uma grande parte (ou a totalidade) de linhas, barras, cargas e chaves do sistema.

O algoritmo proposto baseia-se em um AE multi-objetivo e na estrutura de dados, para armazenamento de grafos, denominada Representação Nó-Profundidade (RNP),

bem como em dois operadores genéticos que foram desenvolvidos para manipular de forma eficiente os dados armazenados na RNP.

Em razão de se basear em um AE multi-objetivo, o algoritmo proposto possibilita uma investigação mais ampla do espaço de busca. Por outro lado, fazendo uso da RNP, para representar computacionalmente os SDR, e de seus operadores genéticos, o algoritmo proposto aumenta significativamente a eficiência da busca por adequados PRE. Isto porque aqueles operadores geram apenas configurações radiais, nas quais todos os consumidores são atendidos.

Para comprovar a eficiência do algoritmo proposto, várias simulações computacionais foram realizadas, utilizando o sistema de distribuição real, de uma companhia Brasileira, que possui 3.860 barras, 635 chaves, 3 subestações e 23 alimentadores.

Palavras-chave: Restauração de Energia, Sistemas de Distribuição de Grande porte, Estrutura de Dados, Representação Nó-Profundidade, Algoritmo Evolutivo Multi-objetivo, NSGA-II.

Abstract

MANSOUR, M. R., Algorithm for elaboration of plans for service restoration to large-scale distribution systems. Sao Carlos, 2009. 110p. Dissertation (Master study), Engineering School of Sao Carlos, University of Sao Paulo.

An elaborated and fast energy restoration plan (ERP) is required to deal with steady faults in radial distribution systems (RDS). That is, after a faulted zone has been identified and isolated by the relays, it is desired to elaborate a proper ERP to restore energy on that zone. Moreover, during the normal system operation, it is frequently necessary to elaborate ERP to isolate zones to execute routine tasks of network maintenance. Some of the objectives of an ERP are: (i) very few interrupted customers (or none), and (ii) operating a minimal number of switches, while at the same time respecting security constraints. As a consequence, the service restoration is a multiple objective problem, with some degree of conflict.

The main methods developed for elaboration of ERP are based on Evolutionary Algorithms (EA). The limitation of the majority of these methods is the necessity of network simplifications to work with large-scale RDS. In general, these simplifications restrict the achievement of an adequate ERP.

This work proposes the development and implementation of an algorithm for elaboration of ERP, which can deal with large-scale RDS without requiring network simplifications, that is, considering a large number (or all) of lines, buses, loads and switches of the system.

The proposed algorithm is based on a multi-objective EA, on a new graph tree encoding called Node-depth Encoding (NDE), as well as on two genetic operators developed to efficiently manipulate a graph trees stored in NDEs.

Using a multi-objective EA, the proposed algorithm enables a better exploration of the search space. On the other hand, using NDE and its operators, the efficiency of the search is increased when the proposed algorithm is used generating proper ERP, because those operators generate only radial configurations where all consumers are attended.

The efficiency of the proposed algorithm is shown using a Brazilian distribution system with 3,860 buses, 635 switches, 3 substations and 23 feeders.

Key-words: Energy Restoration, Large-Scale Distribution System, Data Structure, Node-Depth Encoding, Multi-Objective Evolutionary Algorithms, NSGA-II.

Sumário

| | |
|---|------------|
| Resumo | vii |
| Abstract | ix |
| Lista de Figuras | xv |
| Lista de Tabelas | xix |
| Lista de Abreviaturas e Siglas | xxi |
| 1 Introdução | 23 |
| 1.1 Sistema de Distribuição | 24 |
| 1.2 Restabelecimento de Energia em Sistemas de Distribuição Radiais . . . | 25 |
| 1.3 Objetivo | 28 |
| 1.4 Organização da Dissertação | 28 |
| 2 Revisão Bibliográfica | 31 |
| 2.1 Considerações Finais | 35 |
| 3 Fundamentos de Algoritmos Evolutivos | 37 |
| 3.1 Base Biológica | 37 |
| 3.1.1 O Processo Evolutivo | 38 |
| 3.1.2 Terminologia Básica | 39 |
| 3.2 Algoritmos Evolutivos | 41 |
| 3.2.1 AEs de Última Geração | 43 |
| 3.3 Operadores Genéticos | 44 |
| 3.3.1 Seleção | 44 |
| 3.3.2 Cruzamento | 45 |
| 3.3.3 Mutação | 45 |
| 3.3.4 Elitismo | 46 |

| | | |
|----------|---|------------|
| 4 | Algoritmos Evolutivos para Otimização Multi-Objetivo | 47 |
| 4.1 | Otimização Multi-Objetivo | 47 |
| 4.1.1 | Problemas de Otimização Multi-Objetivo | 47 |
| 4.1.2 | Metas em Otimização Multi-Objetivo | 52 |
| 4.1.3 | Diferenças entre Otimização Multi-Objetivo e a Otimização Mono-Objetivo | 53 |
| 4.1.4 | Técnicas Tradicionais para MOOP | 54 |
| 4.2 | Algoritmos Evolutivos para Otimização Multi-Objetivo | 59 |
| 4.2.1 | <i>NSGA-II: Elitist Non-Dominanted Sorting Genetic Algorithm</i> | 61 |
| 5 | Estruturas de Dados para AEs Aplicados a Problemas de Projeto de Redes | 67 |
| 5.1 | Principais Conceitos da Teoria de Grafos | 67 |
| 5.2 | Representações de PPRs para AEs | 69 |
| 5.3 | Representação Nó-Profundidade | 70 |
| 5.3.1 | Operador 1 | 72 |
| 5.3.2 | Operador 2 | 73 |
| 6 | Fluxo de Carga | 77 |
| 6.1 | Considerações Iniciais | 77 |
| 6.2 | Método <i>Backward/Forward</i> de Soma de Correntes | 79 |
| 6.3 | Método <i>Backward/Forward</i> da Soma de Potências | 81 |
| 7 | Algoritmo Evolutivo para Reconfiguração de SDR | 83 |
| 7.1 | Restabelecimento: um problema especial de reconfiguração de SDR | 83 |
| 7.2 | Formulação Matemática | 86 |
| 7.3 | Avaliação das soluções | 90 |
| 7.3.1 | Extensão da RNP para fluxo de carga | 90 |
| 7.3.2 | Método <i>Backward/Forward</i> com RNP | 93 |
| 7.3.3 | Cálculo do número de manobras | 95 |
| 7.4 | Algoritmo Evolutivo para Restabelecimento de SDR | 98 |
| 7.5 | Comentários Adicionais | 99 |
| 8 | Algoritmo Proposto para Restabelecimento de Energia para SDR | 101 |
| 8.1 | Formulação Matemática | 101 |
| 8.2 | Algoritmo Proposto | 103 |
| 8.3 | Exemplo Ilustrativo da Aplicação do PRN | 104 |

| | | |
|-----------|---|------------|
| 9 | Testes e Resultados | 111 |
| 9.1 | Testes Realizados | 111 |
| 9.2 | Resultados das Simulações no SDR Real de São Carlos | 114 |
| 9.2.1 | Falta Única | 114 |
| 9.2.2 | Múltiplas Faltas | 118 |
| 9.3 | Resultados das Simulações no SDR Real de São Carlos Duplicado . . . | 123 |
| 9.4 | Comentários Adicionais | 128 |
| 10 | Conclusões | 129 |
| 10.1 | Publicações | 131 |
| | Referências Bibliográficas | 133 |

Lista de Figuras

| | | |
|------------|--|----|
| FIGURA 1.1 | Representação de um sistema de distribuição. | 24 |
| FIGURA 3.1 | Exemplo da aplicação do operador de cruzamento em um ponto. | 45 |
| FIGURA 3.2 | Exemplo da aplicação do operador de mutação. | 45 |
| FIGURA 4.1 | Exemplo que ilustra o preço e o desempenho para 5 alternativas de compra de computadores. | 50 |
| FIGURA 4.2 | Soluções de Pareto-ótimas locais e global. | 52 |
| FIGURA 4.3 | Diferentes distribuições de soluções na fronteira de Pareto. | 52 |
| FIGURA 4.4 | O método do somatório de pesos. | 55 |
| FIGURA 4.5 | Método de restrições ϵ (Deb, 2001). | 56 |
| FIGURA 4.6 | Método de programação de metas lexicográficas (Deb, 2001). | 58 |
| FIGURA 4.7 | Ordenação por não dominância (Deb, 2001). | 62 |
| FIGURA 4.8 | Esquema do modelo NSGA-II (Deb, 2001). | 65 |
| FIGURA 5.1 | Exemplo de um grafo. | 68 |
| FIGURA 5.2 | Exemplo de um grafo e sua RNP | 71 |
| FIGURA 5.3 | Ilustração dos passos do operador 1 | 73 |
| FIGURA 5.4 | Ilustração dos passos do operador 2. | 75 |
| FIGURA 6.1 | Exemplo de um SDR. | 78 |
| FIGURA 6.2 | Sistema de distribuição radial. | 80 |

| | | |
|------------|--|-----|
| FIGURA 7.1 | SDR com 3 alimentadores. | 84 |
| FIGURA 7.2 | SDR em falta no setor 15. | 85 |
| FIGURA 7.3 | Setores a jusante do setor em falta desconectados do SDR. . . | 85 |
| FIGURA 7.4 | Nova configuração. | 86 |
| FIGURA 7.5 | SDR com dois alimentadores. | 91 |
| FIGURA 7.6 | Agrupamento das linhas e barras em setores. | 91 |
| FIGURA 7.7 | Grafo representando setores do SDR da Figura 7.6. | 91 |
| FIGURA 7.8 | Árvore do setor C, com o nó adicional. | 92 |
| FIGURA 7.9 | Operações de manobras necessárias para isolar o setor em falta. | 96 |
| | | |
| FIGURA 8.1 | SDR com 8 alimentadores. | 105 |
| FIGURA 8.2 | SDR com falta no setor 4. | 106 |
| FIGURA 8.3 | SDR restabelecido. | 106 |
| FIGURA 8.4 | Melhor configuração alterada. | 107 |
| FIGURA 8.5 | Melhor configuração alterada gerada no meio do processo iter- ativo. | 108 |
| FIGURA 8.6 | Melhor configuração gerada no final do processo iterativo. . . . | 109 |
| | | |
| FIGURA 9.1 | Primeira Fronteira de Pareto após a falta única no setor 3668. | 116 |
| FIGURA 9.2 | Redução das perdas por geração para falta única no setor 3668. | 117 |
| FIGURA 9.3 | Aumento das manobras por geração para falta única no setor 3668. | 117 |
| FIGURA 9.4 | Manobras e perdas resistivas linearizadas para falta única no setor 3668. | 118 |
| FIGURA 9.5 | Primeira Fronteira de Pareto (múltiplas faltas). | 119 |
| FIGURA 9.6 | Redução das perdas por geração (múltiplas faltas). | 121 |
| FIGURA 9.7 | Aumento das manobras por geração (múltiplas faltas). | 121 |
| FIGURA 9.8 | Manobras e perdas resistivas linearizadas (múltiplas faltas). . . | 122 |

| | |
|--|-----|
| FIGURA 9.9 Primeira Fronteira de Pareto após as múltiplas faltas no SDR duplicado. | 124 |
| FIGURA 9.10 Redução das perdas por geração para múltiplas faltas no SDR duplicado. | 126 |
| FIGURA 9.11 Redução das manobras por geração para múltiplas faltas no SDR duplicado. | 126 |
| FIGURA 9.12 Manobras e perdas resistivas linearizadas para múltiplas faltas no SDR duplicado. | 127 |

Lista de Tabelas

| | | |
|------------|---|-----|
| TABELA 2.1 | Classificação das publicações segundo a técnica utilizada. . . . | 32 |
| TABELA 4.1 | Diferentes modelos de MOEAS. | 61 |
| TABELA 5.1 | Grau de cada um dos nós do grafo da figura 5.1. | 68 |
| TABELA 5.2 | Principais representações de PPRs para AEs. | 70 |
| TABELA 7.1 | Manobras de chaves - caso 1. | 97 |
| TABELA 7.2 | Manobras de chaves - caso 2. | 97 |
| TABELA 7.3 | Manobras de chaves - caso 3. | 97 |
| TABELA 9.1 | Configurações da primeira Fronteira de Pareto após a falta única. | 114 |
| TABELA 9.1 | Configurações da primeira Fronteira de Pareto após a falta única. (continuação) | 115 |
| TABELA 9.2 | Comparativo entre os métodos AERT e PRN para falta única no setor 3668. | 116 |
| TABELA 9.3 | Configurações da primeira Fronteira de Pareto (múltiplas faltas). | 119 |
| TABELA 9.3 | Configurações da primeira Fronteira de Pareto após múltiplas faltas (continuação). | 120 |
| TABELA 9.4 | Comparativo entre os métodos AERT e PRN em múltiplas faltas. | 122 |
| TABELA 9.5 | Configurações da primeira Fronteira de Pareto após múltiplas faltas no SDR duplicado | 124 |

| | | |
|------------|--|-----|
| TABELA 9.5 | Configurações da primeira Fronteira de Pareto após múltiplas faltas no SDR duplicado(continuação). | 125 |
| TABELA 9.6 | Comparativo entre os métodos AERT e PRN em múltiplas faltas. | 127 |

Lista de Abreviaturas e Siglas

| | |
|------------------|---|
| ϵ -MOEA | <i>ϵ-dominance Multi-Objective Evolutionary Algorithm</i> |
| AE | <i>Algoritmo Evolutivo</i> |
| AG | <i>Algoritmo Genético</i> |
| BH | <i>Busca Heurística</i> |
| BT | <i>Busca Tabu</i> |
| BTR | <i>Busca Tabu Reativa</i> |
| DEC | <i>Duração Equivalente por Consumidor</i> |
| DRO | <i>Depths Recombination Operator</i> |
| EHO | <i>Evolutionary History Operator</i> |
| EE | <i>Estratégias Evolutivas</i> |
| FEC | <i>Frequência Equivalente de Interrupção por Consumidor</i> |
| PE | <i>Programação Evolutiva</i> |
| PG | <i>Programação Genética</i> |
| SA | <i>Simulated Annealing</i> |
| MOO | <i>Multi-Objective Optimization</i> |
| MOEA | <i>Multi-Objective Evolutionary Algorithm</i> |
| MOGA | <i>Multiple Objective Genetic Algorithm</i> |
| MOOP | <i>Multi-Objective Optimization Problem</i> |
| MPF | <i>Modelo Pai-Filho</i> |
| Micro-GA | <i>Multi-Objective Micro-Genetic Algorithm</i> |
| MOMGA-I | <i>Multi-Objective Messy Genetic Algorithm I</i> |
| MOMGA-II | <i>Multi-Objective Messy Genetic Algorithm II</i> |
| NA | <i>Normalmente Aberta</i> |
| NDDE | <i>Node-Depth-Degree Encoding</i> |

| | |
|---------------|---|
| NDRO | <i>Node Depth Recombination Operator</i> |
| NF | <i>Normalmente Fechada</i> |
| NPGA | <i>Niched-Pareto Genetic Algorithm</i> |
| NS2R | <i>NSGA-II com RNP</i> |
| NSGA | <i>Non-Dominanted Sorting Genetic Algorithm</i> |
| NSGA-II | <i>Elitist Non-Dominanted Sorting Genetic Algorithm</i> |
| PESA-I | <i>Pareto Envelope-Base Selection Algorithm I</i> |
| PESA-II | <i>Pareto Envelope-Base Selection Algorithm II</i> |
| PAES | <i>Pareto-Archived Evolutionary Strategy</i> |
| PPR | <i>Problema de Projeto de Redes</i> |
| PPES | <i>Predator-Prey Evolutionary Strategy</i> |
| PRN | <i>Programa para Restabelecimento de SDRs utilizando o NS2R</i> |
| REMOEA | <i>Rudolph's Elitist Multi-Objective Evolutionary Algorithm</i> |
| RNP | <i>Representação Nó-Profundidade</i> |
| SI | <i>Sistemas Inteligentes</i> |
| SDR | <i>Sistema de Distribuição Radial</i> |
| SEP | <i>Sistema Elétrico de Potência</i> |
| SPEA | <i>Strenght Pareto Evolutionary Algorithm</i> |
| SPEA2 | <i>Strenght Pareto Evolutionary Algorithm 2</i> |
| TGA | <i>Thermodynamical Genetic Algorithm</i> |
| VEGA | <i>Vector Evaluated Genetic Algorithm</i> |
| WBGA | <i>Weight Based Genetic Algorithm</i> |

Capítulo 1

Introdução

Em razão do aumento da demanda de energia elétrica e da expansão dos sistemas elétricos de potência (SEP), para manter o suprimento de energia elétrica, obedecendo à trilogia de continuidade, qualidade e economia de serviço (Miller, 1987), torna-se necessário, cada vez mais, o desenvolvimento de ferramentas computacionais de auxílio à operação dos SEP.

Os SEP podem ser divididos nos três grandes blocos (Kagan et al., 2005):

- Sistema de **geração**: composto pelas usinas de geração de energia elétrica, que geram a energia elétrica a partir da conversão eletromecânica de energia. A fonte primária de energia pode ser a água, o carvão, o óleo, a fissão nuclear, etc;
- Sistema de **transmissão**: composto basicamente por linhas de transmissão e transformadores reguladores, que conectam os pontos de geração aos pontos de consumo (até as subestações de distribuição);
- Sistema de **distribuição**: composto por subestações abaixadoras e circuitos elétricos (chamados alimentadores). É responsável pelo fornecimento de energia às áreas urbanas, rurais ou grandes empresas consumidoras.

Este trabalho trata do problema de Restabelecimento de energia em Sistemas de Distribuição Radiais de média tensão.

1.1 Sistema de Distribuição

O sistema de Distribuição pode ser dividido em:

- **Sistema (ou rede) de Distribuição Primária (ou Distribuição de Média Tensão):** opera geralmente em redes radiais aéreas na tensão de 13,8kV. É projetado com possibilidade de transferência de blocos de cargas entre circuitos para o atendimento da operação em condições de contingências ou para manutenção preventiva e/ou corretiva. Esse sistema atende aos consumidores primários (industriais de médio porte, conjuntos comerciais, grandes hospitais, *shopping centers*, instalações de iluminação pública, etc) e aos transformadores de distribuição que, por sua vez, suprem os sistemas de distribuição secundária ou de baixa tensão;
- **Sistema (ou rede) de Distribuição Secundária (ou Distribuição de Baixa Tensão):** opera geralmente em redes radiais ou em malha com tensões de 220/127V ou 380/220V. Atende aos consumidores de baixa tensão, pequenos comércios e indústrias e, principalmente, os consumidores domésticos. Essa parte do sistema de distribuição usualmente não conta com recurso para o atendimento de contingências.

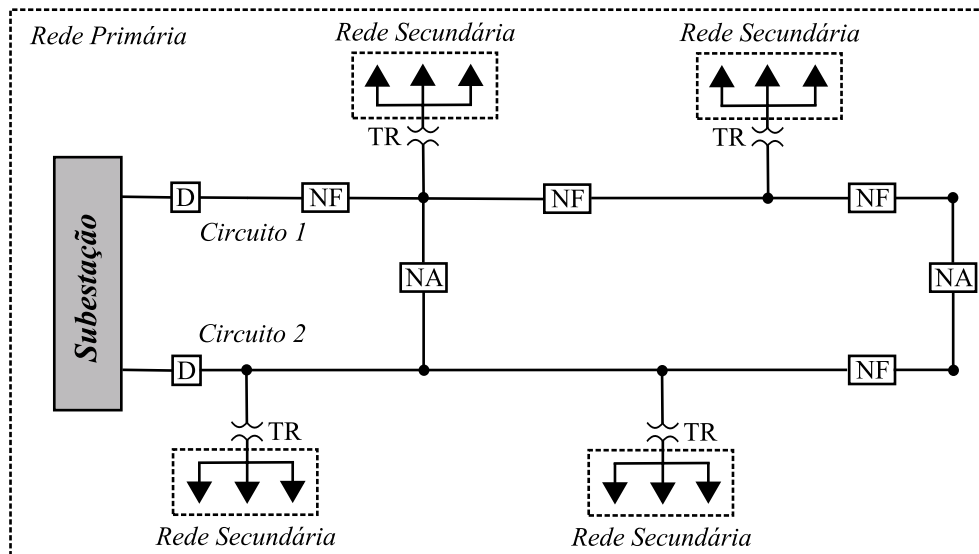


Figura 1.1: Representação de um sistema de distribuição.

A Figura 1.1 ilustra o diagrama unifilar de um sistema de distribuição dividido em Rede Primária e Secundária, onde TR é um transformador, D é um disjuntor, NF é uma chave Normalmente Fechada e NA é uma chave Normalmente Aberta.

1.2 Restabelecimento de Energia em Sistemas de Distribuição Radiais

No crescente mercado competitivo as distribuidoras de energia elétrica têm enfrentado muitos desafios relacionados com os objetivos de melhorar a qualidade e confiabilidade do serviço de fornecimento de energia elétrica. Ou seja, a energia deve ser entregue ao consumidor de forma contínua, com a tensão mais constante possível, sem conteúdo harmônico, com mínimas perdas de potência ativa e máxima margem de segurança em termos de estabilidade de tensão.

A característica radial da maioria dos sistemas de distribuição simplifica a operação e proteção dos mesmos, porém diminui a confiabilidade desses sistemas em relação à continuidade do fornecimento de energia elétrica, que é geralmente avaliada pelas empresas de distribuição, a partir das ocorrências no sistema de distribuição.

A contabilização da continuidade de fornecimento de energia aos consumidores é avaliada após um determinado período, por exemplo, o ano, através de índices operativos, como por exemplo (Kagan et al., 2005):

- **Duração Equivalente por Consumidor (DEC):** representa o espaço de tempo em que, em média, cada consumidor, na área em estudo, teve seu fornecimento de energia elétrica interrompido no período considerado;
- **Frequência Equivalente de Interrupção por Consumidor (FEC):** representa o número de interrupções que, em média, cada consumidor, na área de estudo, sofreu, no período considerado;

Face ao exposto, a reconfiguração de redes é uma ferramenta importante na automação dos Sistemas de Distribuição Radiais (SDR), pois é um dos principais recursos para manutenção da qualidade e confiabilidade do fornecimento de energia elétrica (Kagan e Oliveira, 1996).

As interrupções no fornecimento de energia nos SDR são inevitáveis, isto em virtude da execução de obras de expansão, intervenções de manutenção preventiva em componentes da rede ou pela atuação de um dispositivo de proteção em decorrência de faltas permanentes. Desta forma, o agrupamento de vários pontos de carga em blocos separados por chaves, que operam no estado Normalmente Aberto (NA) e Normalmente

Fechado (NF), foi uma solução encontrada para melhorar a confiabilidade dos SDR sem incorrer em gastos excessivos. Assim, a partir da reconfiguração da rede, isto é, da operação de chaves, é possível a troca de carga entre os alimentadores em caso de interrupção em algum ponto da rede. Nessas situações, torna-se necessário um plano de restabelecimento de energia, que consiste, basicamente, em determinar um conjunto de manobras de chaves para restringir as interrupções à menor parte possível do sistema.

Vale destacar ainda que em condições normais de operação, pode-se utilizar a reconfiguração de redes, através da manobra de chaves NA e NF, para reduzir as perdas totais por efeito joule (Gomes et al., 2005; Sarfi et al., 1995) e/ou para balanceamento de carga entre os alimentadores (Chen et al., 2000), aliviando os alimentadores que estão com carregamento crítico. Nesse contexto, a reconfiguração permite a redução de queda de tensão (Mantovani et al., 2000) e alívio de trechos da rede com sobrecarga (Strogatz, 2001) (corrente elétrica em níveis acima do suportado pelos cabos).

Como mencionado anteriormente, reconfiguração de redes também pode ser aplicada numa condição mais extrema, como, por exemplo, na ocorrência de contingências como faltas permanentes (Inagaki et al., 2006; Kumar et al., 2008). Tal aplicação é o foco deste projeto de pesquisa. Nesse caso, torna-se necessário a obtenção de um plano de restabelecimento de energia elaborado e rápido. Ou seja, depois de o setor¹ em falta ter sido identificado e isolado, pela atuação do sistema de proteção, é de interesse dos operadores encontrar um plano apropriado para o restabelecimento da energia na área que havia ficado sem energia.

Um plano de restabelecimento de energia adequado tem como principais necessidades práticas:

- Encontrar um plano em intervalo de tempo o mais curto possível (*tempo-real*);
- O número de manobras deve ser mínimo²;
- Reduzido número de consumidores interrompidos;
- Nenhum componente sobrecarregado;

¹Um setor corresponde a um conjunto de barras e linhas sem a presença de chaves seccionadoras.

²Busca-se um reduzido número de chaveamentos basicamente por dois motivos: a operação frequente das chaves reduz a expectativa de vida destas; quanto mais manobras, maior o tempo para implementar o plano (as chaves são em geral operadas manualmente).

- A estrutura radial (sem formar anéis) do sistema deve ser mantida;
- Reduzir o total de perdas resistivas;
- Reduzir quedas de tensão.

Face ao exposto, o restabelecimento de energia é um problema com múltiplos objetivos, alguns conflitantes. Naturalmente, outros objetivos, além dos supracitados, podem ser considerados na formulação do problema.

Devido ao problema de explosão combinatória, as técnicas de programação matemática não são utilizadas nos problemas de restabelecimento de energia em SDR de grande porte. A maioria dos Algoritmos Evolutivos (AEs) são uma técnica alternativa que têm se mostrado capaz de lidar com essa dificuldade, porém produzem muitas configurações não factíveis³ quando aplicados em SDR de tamanho real (dos Santos, 2004).

O desempenho de um AE convencional⁴, para restabelecimento de energia em SDR, é afetado principalmente pelos seguintes fatores:

1. A estrutura de dados adotada: a elaboração de planos de restabelecimento via AEs requer um algoritmo de busca em grafo. Assim o desempenho dos AEs torna-se fortemente afetado pela forma com que as árvores de grafo são computacionalmente representadas;
2. Os operadores genéticos adotados, que podem produzir muitas configurações não factíveis; e
3. A conversão de um problema multi-objetivo em um mono-objetivo através da utilização de fatores de ponderação.

Buscando melhorar o desempenho dos AEs para o tratamento do PRE, em (Santos, Delbem e Bretas, 2008b) o SDR foi representado computacionalmente através de uma nova estrutura de dados, denominada Representação Nó-Profundidade (RNP). Associados a essa estrutura há dois operadores que permitem a realização de poda ou enxerto

³No contexto em pauta, configurações factíveis são configurações radiais, que atendem a todos os consumidores respeitando os limites de operação do sistema.

⁴AEs convencionais são aqueles que convertem um problema de otimização multi-objetivo em um problema mono-objetivo através da utilização de uma função agregação e de fatores de ponderação.

nas árvores da floresta de grafo armazenada na RNP. Os alimentadores podem ser vistos como árvores da floresta que, por sua vez, representa um SDR. Em outras palavras, os operadores da RNP modificam a floresta de grafo gerando uma somente configurações factíveis. A garantia de geração de somente configurações factíveis, aumenta significativamente a eficiência da busca por melhores configurações do AE.

Vale destacar que a utilização da RNP possibilita ainda outra vantagem para o tratamento do problema de restabelecimento de energia. Cada configuração gerada através da RNP e de seus operadores possui sempre todos os seus nós ordenados de acordo com uma relação conhecida como pai-filho. Essa ordenação possibilita a execução de um fluxo de carga extremamente rápido. Trabalhando com outras estruturas de dados e operadores, para possibilitar a utilização de um fluxo de carga rápido, torna-se necessário executar um algoritmo de ordenação toda vez que uma nova configuração for gerada, para organizar os nós de acordo com o Modelo Pai-Filho (MPF).

Recentemente, as técnicas de AEs Multi-Objetivo (MOEA, do inglês *Multi-Objective Evolutionary Algorithm*) têm sido aplicadas para o problema de restabelecimento de energia em SDR, com resultados que se mostram bastante promissores. Por exemplo, Kumar et al. (2008) aplicou a técnica NSGA-II (*Elitist Non-Dominated Sorting Genetic Algorithm*).

1.3 Objetivo

O principal objetivo deste trabalho é a elaboração de um programa computacional que possibilite a obtenção de planos de restabelecimento de energia, em SDRs de grande porte, considerando todas as suas linhas, barramentos, cargas e chaves. Para isso, utilizar-se-á a estrutura de dados RNP e seus operadores, bem como uma versão modificada do NSGA-II.

1.4 Organização da Dissertação

Os próximos Capítulos desta dissertação estão organizados da seguinte forma:

- O Capítulo 2 revisa os principais trabalhos desenvolvidos para tratar o problema de restabelecimento de energia em SDR;

- O Capítulo 3 descreve fundamentos dos AEs;
- O Capítulo 4 introduz conceitos de otimização multi-objetivo, descrevendo as técnicas tradicionais para resolver problemas de otimização multi-objetivo; e a técnica NSGA-2;
- O Capítulo 5 apresenta os conceitos básicos da teoria de grafos e o Problema de Projeto de Redes (PPRs) para AE, bem como a estrutura de dados denominada RNP;
- O Capítulo 6 revisa métodos para cálculo de fluxo de carga para SDR.
- O Capítulo 7 apresenta os estudos realizados sobre o método desenvolvido em (dos Santos, 2008).
- O Capítulo 8 apresenta o NS2R, isto é, o AE proposto neste trabalho para a obtenção de planos de restabelecimento de energia em SDR;
- O Capítulo 9 apresenta testes comprobatórios da eficiência do NS2R, juntamente com a análise dos resultados; e
- O Capítulo 10 apresenta as conclusões do trabalho, bem como as publicações originadas desta pesquisa.

Capítulo 2

Revisão Bibliográfica

Destacam-se, neste Capítulo, algumas das principais técnicas para geração de planos de restabelecimento de energia em SDR encontradas na literatura.

Em (Curcic et al., 1996) é apresentada uma análise de diversos artigos publicados entre os anos 1987 e 1994, relacionados a restabelecimento de energia em SDRs. Ressalta-se, nesse artigo, a importância do rápido processamento computacional, bem como as vantagens e desvantagens em se utilizar uma topologia de rede radial. No total foram revisados 19 artigos, os quais estão classificados segundo a técnica utilizada, conforme indicado na Tabela 2. Nessa análise, foram explorados os tipos de faltas possíveis para SDR: nas linhas, barras e transformadores.

Diversos trabalhos publicados em 2000 utilizaram Algoritmos Evolutivos (AEs) e lógica *fuzzy* para resolver o problema de restabelecimento de energia em SDR.

Em (Augugliaro et al., 2000) foram utilizadas Estratégias Evolutivas (EE), com uma definição *fuzzy* de múltiplos objetivos que compõem um problema de restabelecimento de energia em SDR, tais objetivos sendo conflitantes. Foi considerado que o estado de operação normal possibilita o controle remoto das chaves, de bancos de capacitores e conexões de cargas. Desta forma, após a ocorrência de uma falta permanente, torna-se possível executar remotamente ações para restabelecer energia nas áreas afetadas. Na formulação do problema duas funções foram consideradas como principais: minimização de perdas resistivas e a maximização da quantidade de cargas a ser restabelecida. As configurações geradas são avaliadas através de conjuntos *fuzzy*. Como restrições foram consideradas: a permanência da estrutura radial do SDR, carregamento nos transfor-

Tabela 2.1: Classificação das publicações segundo a técnica utilizada.

| Técnica | Trabalhos publicados |
|--|--|
| Sistemas Inteligentes (SI) | (Teo, 1992) (Kim et al., 1992) (Liu et al., 1988) (Fujii et al., 1992) (Kendrew e Graham, 1989) (Okuda et al., 1988) (Srinivasan et al., 1994) |
| Busca Heurística (BH) | (Shirmohammadi, 1992) (Wu et al., 1991) (Hsu et al., 1992) (Devi et al., 1990) (Devi et al., 1991) (Morelato e Monticelli, 1989) (Nahman e Strbac, 1994) |
| Método do gradiente efetivo dual | (Aoki, Nara, Itoh, Satoh e Kuwabara, 1989) (Aoki, Nara e Satoh, 1989) |
| Busca Tabu (BT) e caminho mínimo | (Dialynas e Michos, 1989) (Sarma et al., 1990) |
| Programação inteira binária e <i>branch and bound</i> | (Chen et al., 1989) |

madores, carregamento das linhas e queda de tensão nas barras. Testes foram realizados em um SDR inicialmente malhado contendo 98 setores, 81 barras de carga e 24 bancos de capacitores. Considerou-se apenas a ocorrência de uma única falta.

A formulação híbrida apresentada em (Hsiao e Chen, 2000) faz uso de conjunto *fuzzy* e de AE. Conjunto *fuzzy* é utilizado para modelar as funções objetivo e avaliar a natureza imprecisa que estas apresentam,; já AE é utilizado para resolver o problema de otimização. Na formulação do problema foram consideradas cinco funções objetivo: área fora de serviço, número de operações de chaveamento, queda de tensão nas barras, carregamento nas linhas e carregamento nos transformadores. Como restrições foram consideradas, manter a estrutura radial do SDR e a sequência de operações de chaveamento. Testes foram realizados em um SDR contendo 2 transformadores, 10 alimentadores, 102 ramos, 102 barras, 217 chaves NF e 13 chaves NA. Foram considerados três casos distintos: uma única falta, múltiplas faltas e múltiplas faltas deixando uma grande área fora de serviço.

Toune et al. (2002) realizaram um estudo comparativo de 4 algoritmos heurísticos utilizados para restabelecimento de energia em SDR. Os algoritmos estudados foram: Busca Tabu (BT), Busca Tabu Reativa (BTR), *Simulated Annealing* (SA) e AEs. O

estudo foi realizado considerando o objetivo de encontrar, após a ocorrência de uma falta, planos de restabelecimento de energia que sejam capazes de minimizar o número de consumidores sem energia. Como restrições foram consideradas: manter a estrutura radial do SDR, queda de tensão, carregamento nos transformadores e carregamento nas linhas. Apresentou-se a formulação matemática de cada um dos algoritmos e foram realizadas comparações qualitativas e quantitativas entre os mesmos. Realizaram-se testes em um SDR contendo 3 alimentadores e 60 barras.

Em (Shin et al., 2004) foi utilizada uma abordagem híbrida, combinando AEs e BTs, para resolver o problema de restabelecimento de energia e reconfiguração ótima de redes em SDR. No referido trabalho uma configuração é dita ótima se o plano de restabelecimento minimiza as perdas e atende as restrições operacionais do sistema, mantendo a rede radial. O algoritmo proposto procura utilizar as propriedades que os AEs e BTs têm de melhor, dando origem ao método denominado AG-Tabu. Na formulação do problema foram avaliados o custo das perdas resistivas e o custo total após a interrupção e reconexão do sistema devido a ocorrência de uma falta. Como restrições foram consideradas: carregamento nos transformadores, carregamento nas linhas e manutenção da estrutura radial do SDR. Testes foram realizados em um SDR com 7 alimentadores e 38 barras, com a ocorrência de uma única falta.

Delbem et al. (2005) propuseram uma nova codificação para SDRs baseada em Cadeia de Grafos, de modo a melhorar o desempenho dos AEs. A partir dessa codificação foram desenvolvidos operadores de reprodução não convencionais, que possibilita a geração de configurações factíveis, a partir de uma configuração já existente. Utilizando conceitos de grafos e, partindo do princípio que uma árvore de grafo pode ser representada por cadeias conectando a raiz às folhas, o conjunto de todas essas cadeias armazenadas adequadamente representa um alimentador de um SDR. Portanto, o conjunto de todos os alimentadores representa um sistema completo. A técnica proposta pode lidar com problemas multi-objetivo, utilizando sub-populações, que é semelhante à técnica empregada em (R. Benayoun e Laritchev, 1971). Testes foram realizados em um SDR de grande porte composto de 1471 barras, 249 chaves, 3 subestações e 23 alimentadores. Como restrições foram consideradas: queda de tensão, carregamento nas linhas e carregamento nos transformadores. A estrutura radial do SDR é sempre uma condição satisfeita no problema, pois os operadores de reprodução propostos geram

apenas configurações factíveis. O artigo trata de uma falta por vez. Foram consideradas faltas em setores críticos da rede, por exemplo, que isolem todo um alimentador. Vale destacar que a técnica foi aplicada ao problema de reconfiguração de SDR, sendo testada em restabelecimento de energia, redução de perdas resistivas e planejamento de SDR.

Em (Inagaki et al., 2006) utilizou-se uma abordagem multi-objetivo baseada na obtenção de soluções pertencentes ao conjunto de Pareto (ver Capítulo 4). Para encontrar essas soluções são utilizados AEs. Desta forma, um número maior de configurações é disponibilizado para o operador decidir qual se adapta melhor ao problema. Uma combinação de AEs e SA é realizada com o objetivo de melhorar a precisão das soluções. Na formulação do problema foram considerados dois objetivos: reduzir a área fora de serviço após uma falta e o número de operações (ou manobras) de chaveamento. Como restrições foram consideradas: manter a estrutura radial do SDR, a energia deve ser restabelecida às áreas a jusante do setor em falta, carregamento nas linhas, carregamento nos transformadores e queda de tensão. Testes foram realizados em um SDR com 4 transformadores, 6 alimentadores e 78 barras de carga. Apenas a ocorrência de uma única falta foi abordada nos testes. Vale destacar que os objetivos priorizam consumidores como hospitais, *shopping centers*, etc.

Buscando melhorar o desempenho dos AEs para o tratamento do problema de restabelecimento de energia, em (Santos, Delbem e Bretas, 2008b) o SDR foi representado computacionalmente através de uma nova estrutura de dados, denominada Representação Nó-Profundidade (RNP). Associados a essa estrutura há dois operadores que permitem a realização de poda ou enxerto nas árvores da floresta de grafo armazenada na RNP. Os alimentadores podem ser vistos como árvores da floresta que, por sua vez, representa um SDR. Em outras palavras, os operadores da RNP modificam a floresta de grafo gerando somente configurações factíveis. A garantia de geração somente de configurações factíveis, aumenta significativamente a eficiência da busca por melhores configurações do AE, este trabalho será apresentado no Capítulo 7.

Recentemente Kumar et al. (2008) desenvolveram um algoritmo para restabelecimento de energia em SDR, baseado no algoritmo de otimização multi-objetivo proposto por (Deb et al., 2000), denominado NSGA-II. Modificações no NSGA-II foram realizadas para melhorar o processamento computacional do mesmo. Os resultados obtidos pelo

método proposto, denominado NSGA-II avançado, foram comparados com os resultados obtidos pelo NSGA-II básico proposto por (Deb et al., 2000) e por AE mono-objetivo. O método proposto conseguiu obter os mesmos resultados encontrados pelas outras técnicas, porém com um melhor tempo computacional. Isso deve-se à implementação do NSGA-II utilizando a estrutura de dados apresentada em (Jensen, 2003). Na formulação do problema foram considerados quatro objetivos: reduzir a área fora de serviço, minimização do número de manobras (tanto para chaves remotamente controladas, quanto para chaves manualmente controladas) e minimizar as perdas resistivas. Como restrições foram consideradas: manter a estrutura radial do SDR, queda de tensão, carregamento da rede, priorizar o restabelecimento para cargas “especiais”, como hospitais e grandes centros industriais. Testes foram realizados em quatro SDR, todos de pequeno porte. A dimensão varia de 13 barras e 10 chaves até 173 barras e 75 chaves.

2.1 Considerações Finais

Conforme apresentado neste capítulo, a maioria das técnicas para obtenção de planos de restabelecimento de energia, em SDR, baseia-se em AEs convencionais, isto é, aqueles que convertem um problema de otimização multi-objetivo em um problema mono-objetivo através da utilização de fatores de ponderação.

Vale destacar, entretanto, que as técnicas baseadas em AEs convencionais possuem ainda algumas limitações, restringindo a aplicação das mesmas para SDR de pequeno porte ou para modelos simplificados de SDR de grande porte. Isso em razão de o desempenho de um AE convencional ser fortemente afetado pelos seguintes fatores:

1. A estrutura de dados adotada: como a busca por planos de restabelecimento via AEs exige normalmente busca em grafo, o desempenho dos AEs torna-se fortemente afetado pela forma com que as árvores de grafo são computacionalmente representadas;
2. Os operadores genéticos adotados, que podem produzir muitas configurações não factíveis; e
3. A conversão de um problema multi-objetivo em um mono-objetivo através da utilização de fatores de ponderação.

Na tentativa de obter um algoritmo para obtenção de planos de restabelecimento de energia mais eficiente, aplicável em SDR de grande porte sem simplificações, propõe-se, neste projeto, o desenvolvimento de um algoritmo baseado no NSGA-II e na estrutura de dados RNP e seus operadores.

Em razão de o algoritmo proposto basear-se no NSGA-II, o mesmo vai possibilitar o tratamento do problema multi-objetivo de obtenção de planos de restabelecimento de energia de forma direta, sem a necessidade de converter o problema original em um problema mono-objetivo. Importa lembrar que para possibilitar o tratamento de problemas multi-objetivos, o NSGA-II faz uso da ordenação elitista por dominância chamada de Pareto *Ranking* (será apresentada no Capítulo 4).

As vantagens de utilizar a RNP e seus operadores, para representar e manipular computacionalmente os SDR, são as seguintes:

1. Abordagens baseadas na RNP, para problemas que requerem a manipulação de grafos, têm apresentado melhor desempenho computacional em relação àquelas que utilizam outras estruturas de dados (Delbem et al., 2004);
2. A utilização dos operadores da RNP, ao invés dos operadores genéticos convencionais, aumenta significativamente a eficiência da busca por melhores soluções (configurações), pois a RNP e seus operadores produzem somente configurações factíveis;
3. A RNP de um SDR possui, naturalmente, as barras de cada árvore (alimentador) ordenadas segundo o MPF. Com isso, evita-se o uso de um algoritmo de busca para obter tal modelo. Assim, o fluxo de carga pelo MPF com RNP é mais eficiente que fluxos de carga convencionais para SDR.

Capítulo 3

Fundamentos de Algoritmos Evolutivos

Os Algoritmos Evolutivos (AEs) são métodos de otimização e busca inspirados nos princípios da Teoria de Darwin, isto é, são baseados em princípios que são encontrados na evolução dos sistemas biológicos. Este capítulo introduz os principais conceitos sobre AEs os quais receberam maior atenção dos pesquisadores após a proposta dos Algoritmos Genéticos (AGs) por John Holland (Hayes-Roth, 1975) e a popularização dos mesmos por meio dos trabalhos de David Goldberg (Goldberg, 1989). Na seção 3.1.1 é apresentada a base biológica dos AEs. Na Seção 3.2 são descritos os AEs, bem como as subáreas que vêm se destacando da computação evolutiva. Na Seção 3.3 são descritos os operadores genéticos.

Para escrever este Capítulo utilizou-se (Gabriel e Delbem, 2008) como referência principal.

3.1 Base Biológica

OS AEs podem ser vistos como técnicas de *Computação Bioinspirada* (Teuscher et al., 2003) ou *Computação Natural* (Ballard, 1999). Tais áreas de pesquisa abordam uma série de técnicas computacionais fundamentadas em conceitos Biológicos. As técnicas evolutivas apresentam conceitos cuja origem está em diversos campos da Biologia, em especial em idéias evolucionistas e na Genética. Esta Seção foca nesses conceitos e

resume a terminologia empregada na definição de AEs.

3.1.1 O Processo Evolutivo

Conforme dito anteriormente, os AEs baseiam-se nos processos evolutivos que ocorrem na natureza. Como principais componentes dos sistemas evolutivos têm-se (Arciszewski e Jong, 2001):

- Populações de indivíduos: uma ou mais populações concorrem por recursos limitados;
- *Fitness*: reflete a habilidade de um indivíduo para sobreviver ou reproduzir-se;
- A noção de mudanças dinâmicas nas populações devido ao nascimento e morte dos indivíduos;
- Os conceitos de variabilidade e hereditariedade, ou seja, os novos indivíduos possuem muitas das características de seus pais, embora não sejam idênticos.

Tais conceitos foram inspirados no *neodarwinismo* (Ridley, 1996), que admite que os principais fatores evolutivos são a **mutação**, a **recombinação** e a **seleção natural**, os quais são resumidos a seguir:

- **Mutação Gênica**

A origem da variabilidade é a mutação, processo pelo qual o gene ¹ sofre alterações em sua estrutura. Tais alterações são modificações na sequência de bases do DNA. Essa molécula, quando duplicada, produz cópias idênticas de si, ou seja, diferentes da original (sem mutação), transmitindo hereditariamente a mudança. Isso pode acarretar a alteração da sequência de aminoácidos da proteína, modificando o metabolismo celular, podendo favorecer o organismo ou mesmo ser letal.

- **Recombinação Gênica**

O processo evolutivo seria relativamente lento se não fosse possível colocar juntas, em um mesmo indivíduo, mutações ocorridas em indivíduos da geração anterior.

¹**Gene** é um segmento de DNA que contém uma informação codificada para determinada característica ou processo que a célula tem ou executa (Amabis e Martho, 1985).

O fenômeno que possibilita esse evento é a reprodução sexuada. É importante considerar que a seleção natural não atua aceitando ou rejeitando mudanças individuais, mas sim escolhendo as melhores combinações gênicas entre todas as variações presentes na população.

- **Seleção Natural**

A seleção natural é consequência de dois fatores:

1. Os membros de uma espécie diferem entre si;
2. A espécie produz descendência em maior número de indivíduos que de fato podem sobreviver.

Os indivíduos mais aptos a sobreviver são aqueles que, graças à variabilidade genética, herdaram a combinação gênica mais adaptada para determinadas condições naturais.

3.1.2 Terminologia Básica

Apresenta-se, a seguir, a terminologia necessária para o estudo de AEs (Sait e Youssef, 1999).

Cromossomo, Genes e Alelos

A estrutura que codifica como os organismos são construídos é chamada **cromossomo**. Os cromossomos associam-se de modo a formar um organismo e seu número varia de uma espécie para outra (Amabis e Martho, 1985). O conjunto completo de cromossomos de um ser vivo é chamado **genótipo** e as características do organismo gerado com base no genótipo constituem o **fenótipo**. De forma similar, a representação de soluções de um problema podem ser codificadas em uma estrutura de dados chamada cromossomo.

Os cromossomos são codificados em um conjunto de símbolos chamados **genes**. Os diferentes valores de um gene são chamados **alelos**. A posição do gene em um cromossomo é denominada **locus** (Lamont e Veldhuizen, 2002).

A representação das soluções candidatas (ou seja, os indivíduos) é o primeiro estágio

da elaboração de um AE e é crucial para o desempenho do algoritmo. Essa etapa consiste em definir o genótipo e a forma como este é mapeado no fenótipo.

A codificação mais simples é a **representação binária**: o genótipo é definido como um arranjo de 0s e 1s. É necessário definir o tamanho do arranjo, bem como o mapeamento genótipo-fenótipo. Entretanto, em muitas aplicações do mundo real, a representação binária pode apresentar fraco poder de expressão (Deb, 2001), não sendo eficiente na representação das possíveis soluções. Uma alternativa empregada é a **representação em ponto-flutuante** ou **representação oral**, segundo a qual as soluções são arranjos de números reais. Essa representação é usualmente empregada quando os genes são distribuídos em um intervalo contínuo, em vez de um conjunto de valores discretos (Andrew, 2004).

Fitness

O valor de *fitness* de um indivíduo (seja um genótipo ou um cromossomo) é um número **positivo** que mede o quanto adequado é o indivíduo, que representa uma solução. Em problemas de otimização, o *fitness* pode ser o custo da solução. Se o problema for de minimização, as soluções de maior *fitness* são as de menor custo.

Pais, Operadores de Reprodução e Descendentes

Os AEs trabalham sobre um ou mais cromossomos a fim de gerar novas soluções, chamadas **descendentes**. Os operadores que trabalham sobre cromossomos, chamados operadores de reprodução, são a **recombinação** (também conhecido como **crossover**) e a **mutação**. Esses operadores fazem analogia aos principais mecanismos da evolução natural, ou seja, a recombinação e a mutação gênica. A recombinação é aplicada, em geral, a um par de cromossomos. Os indivíduos selecionados para o processo de recombinação são chamados **pais**. A mutação é aplicada a um simples cromossomo, modificando-o aleatoriamente.

Geração e Seleção

A **geração** é uma iteração do AE, na qual os indivíduos da população atual são selecionados e recombinados e/ou mutados, gerando descendentes. Devido à criação

de novos descendentes, o tamanho da população cresce; deste modo um mecanismo de seleção controla esse tamanho.

A ideia básica da seleção é a seguinte: seja uma população de tamanho M e seja N_d o número de descendentes, então, para a próxima geração, são selecionados M novos indivíduos (N_d pode ser maior que M). Cada AE desenvolve, com base nesse princípio, uma estratégia de seleção.

3.2 Algoritmos Evolutivos

Os AEs funcionam basicamente da seguinte forma:

1. Primeiramente é criada uma população inicial com soluções aleatórias;
2. A partir da população atual, é gerada uma nova população. Os novos indivíduos desta nova população, são criados através do uso dos operadores genéticos. Esta tarefa é realizada aplicando-se o operador de cruzamento nos indivíduos com o melhor *fitness*, que são escolhidos através de um processo chamado de seleção;
3. Retornar para o item 2 até atender à condição de parada;

O Algoritmo 1 mostra o pseudocódigo de um AE.

Os AEs são utilizados para problemas de otimização em decorrência de ser o método preferencialmente utilizado pela natureza, que é considerada por muitos como o sistema mais perfeito. Além disso, resolvem problemas com modelos matemáticos complexos de modo simples, sendo de fácil acoplamento com outras técnicas (hibridação)(dos Santos, 2004).

Existem várias subáreas na Computação Evolutiva, das quais destacam-se:

Algoritmos Genéticos (AG)

Tais algoritmos foram propostos por Holland na década de 1970 e trabalham com populações de indivíduos (cromossomos), que durante o processo de evolução são submetidos aos procedimentos de seleção e reprodução. Deste modo o algoritmo consegue aproveitar das melhores soluções e ao mesmo tempo explorar o espaço de busca.

Algoritmo 1 Algoritmo Evolutivo.

```
1: //Inicia o contador de tempo
2:  $g \leftarrow 1$ 
3: IniciaPopulação( $P_g$ )
4: //Avalia o fitness dos indivíduos  $P_g$ 
5: Avalia( $P_g$ )

6: //Verifica o critério de parada
7: while critério de parada não é atingido do
8:     //Incrementa a geração
9:      $g \leftarrow g + 1$ 
10:    //Seleciona os indivíduos para a geração dos descendentes
11:     $P_g \leftarrow$  Seleciona( $P_{g-1}$ )
12:    //Realiza o cruzamento dos pais selecionados
13:    Cruzamento( $P_g$ )
14:    //Realiza a mutação sobre a nova população
15:    Muta( $P_g$ )
16:    //Avalia o fitness dos indivíduos  $P_g$ 
17:    Avalia( $P_g$ )
18: end while
```

Programação Evolutiva (PE)

Foi proposta por Lawrence J. Fogel na década de 1960, originalmente como uma estratégia de otimização estocástica similar aos AGs. No entanto, enfatiza o relacionamento entre os progenitores e seus descendentes aos invés de tentar emular operadores genéticos específicos observados na natureza (Castro, 2001).

A PE também opera com populações, mas apenas diferentes níveis de mutação são efetuados sobre os progenitores na criação de novas soluções. O tamanho da população não necessita ser mantido constante, como também não é necessário um número fixo de descendentes por progenitor. A PE trabalha com representações mais flexíveis que as empregadas pelos AGs por não efetuarem recombinações.

Programação Genética (PG)

A Programação Genética (PG) foi proposta em (Koza, 1989) e pode ser vista como uma extensão dos AGs. A PG difere dos AEs devido a sua representação, seus operadores de reprodução e seus métodos de avaliação do *fitness*. Introduzida a para solucionar os problemas de aprendizado de máquina, a PG busca a construção automática de programas de computadores. Os indivíduos são codificados na forma de

árvores, onde cada nó folha contém constantes, variáveis ou parâmetros para a execução de procedimentos e funções. Os nós internos contém operações primárias.

Os operadores de reprodução utilizados são operadores de recombinação e mutação específicos para representações por árvores. Na recombinação, partes das árvores são trocadas, o ponto de corte na árvore é escolhido de forma a evitar a criação de operações inválidas. Na mutação, o valor de um nó ou subárvore é alterado. Se o nó escolhido para a mutação for um nó interno, este será alterado para ter uma nova operação ou função. No caso de mutação de subárvore, a subárvore selecionada é substituída por uma nova subárvore gerada aleatoriamente.

O processo de avaliação ocorre por meio da execução do programa representado pela árvore do indivíduo. Se este resolver o problema proposto ou se aproximar da resposta correta, terá um valor de *fitness* elevado; caso contrário, seu *fitness* será baixo. Geralmente, os algoritmos de PG utilizam somente o operador de recombinação no processo de busca pelas melhores soluções.

Estratégias Evolutivas (EE)

Propostas originalmente para tratarem problemas técnicos de otimização como alternativa aos métodos convencionais. Operam com cromossomos na forma de vetores de números reais e originalmente na proporção $(1 + 1)$, isto é, cada progenitor gera um herdeiro por geração, normalmente por mutações distribuídas. Caso esse descendente seja melhor que seu progenitor, ele lhe toma o lugar. Essas estratégias foram estendidas para as proporções $(m + 1)$, isto é, m progenitores geram um herdeiro por geração, e $(m+n)$, isto é, m progenitores geram n herdeiros por geração. As EE tiveram estratégias de recombinações introduzidas no seu processo evolutivo (Castro, 2001).

3.2.1 AEs de Última Geração

Nos últimos dez anos vários estudos utilizando modelos probabilísticos para as populações em AEs foram desenvolvidos buscando aumentar o desempenho de algoritmos de busca populacionais (ou baseados em populações). O sucesso dessas novas técnicas para uma diversidade de problemas complexos e de larga-escala, que ainda não eram resolvidos satisfatoriamente pelos AEs da primeira geração, fez esses novos algoritmos

merecerem uma nova classificação para distingui-los dos AEs convencionais da primeira geração (Goldberg, 1989). Esse novos AEs na literatura também têm sido chamados Algoritmos de Estimação de Distribuição. Dentre esses destacam-se: ECGA (Harik et al., 2006), BOA (Pelikan et al., 2000) e *h*-BOA (Pelikan, 2005).

3.3 Operadores Genéticos

Nesta Seção são abordados os principais aspectos dos operadores genéticos utilizados nos AEs.

3.3.1 Seleção

O objetivo deste operador é escolher um ou mais indivíduos para gerar um ou mais descendentes para a próxima população do processo evolutivo. Os indivíduos com o melhor grau de *fitness* têm uma maior probabilidade de serem escolhidos nesta etapa.

Existe, na literatura, uma grande variedade de estratégias de seleção. Porém, as mais utilizadas são a **seleção por torneio**, **roda da roleta** e ***ranking***.

Na **seleção por torneio**, são realizadas várias competições entre duas ou mais soluções, e a melhor solução é a escolhida. Na **roda da roleta**, geralmente, os pais são selecionados com probabilidade proporcional aos seus *fitness*. Para tal seleção usa-se a expressão 3.1 (Michalewicz, 1992).

$$P_i = \frac{F_i}{\sum_{i=1}^N (F_i)} \quad (3.1)$$

onde, F_i é o *fitness* da solução i e N é o tamanho da população. Logo, é gerado um valor aleatório k , no intervalo de 0 a $PTOTAL$ (Soma de todos os valores de *fitness*). Finalmente, o indivíduo selecionado é o primeiro que possui uma probabilidade de seleção maior que k . Na seleção por ***ranking***, são ordenadas as soluções de acordo com o seu valor de *fitness* (sendo o *ranking 1* pertencente a pior solução e o *ranking N* pertencente a melhor solução, N sendo o número de soluções). Com isso, determina-se a probabilidade de seleção para cada solução. Logo, a escolha das soluções progenitoras é referente ao valor do *ranking*.

3.3.2 Cruzamento

O operador de cruzamento gera as soluções descendentes das soluções progenitoras. Basicamente, para cada duas das soluções progenitoras selecionadas corta-se o seu vetor de símbolos em uma posição aleatória, produzindo duas cabeças e duas caudas. Em seguida as caudas são trocadas, gerando dois novos indivíduos (Figura 3.1).

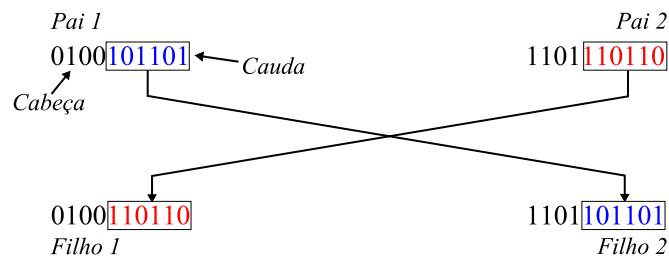


Figura 3.1: Exemplo da aplicação do operador de cruzamento em um ponto.

Existem diversas variações desse operador, vários deles são específicos para determinado problema (Goldberg, 1989).

3.3.3 Mutação

Este operador gera uma determinada taxa de “perturbação” em um determinado número de soluções, isto é, gera pequenas alterações em um determinado número de soluções, com o objetivo de explorar o espaço de busca (Figura 3.2) e manter a diversidade das soluções. Deste forma, o AE tende a não ter uma convergência rápida, evitando a sua estabilização em regiões chamadas de mínimos locais, nos quais os AEs sempre estão sujeitos a cair.

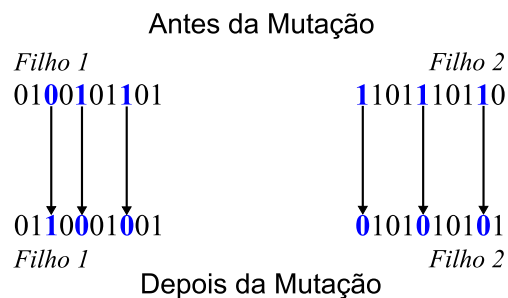


Figura 3.2: Exemplo da aplicação do operador de mutação.

3.3.4 Elitismo

Existe um grande risco de perder os melhores indivíduos na transição de uma geração para outra, isto devido à aplicação dos operadores de mutação e cruzamento. Desse modo, o objetivo do operador de elitismo é preservar os melhores indivíduos para as próximas gerações que possam surgir, sem que esses sofram alguma alteração. Assim, as melhores soluções não se deterioram.

Capítulo 4

Algoritmos Evolutivos para Otimização Multi-Objetivo

Este Capítulo introduz os principais aspectos da otimização multi-objetivo e algumas das principais técnicas de Algoritmos Evolutivos Multi-Objetivo (MOEA, do inglês *Multi-Objective Evolutionary Algorithm*).

Para escrever este Capítulo utilizou-se (Ticona e Delbem, 2008) como referência principal.

4.1 Otimização Multi-Objetivo

Esta seção introduz as noções básicas de Otimização Multi-objetivo (MOO, do inglês *Multi-Objective Optimization*). Na Seção 4.1.1 são apresentados os principais conceitos da área. Na Seção 4.1.2 são definidas as metas em MOO. Na Seção 4.1.3 são explicadas as principais diferenças entre MOO e otimização mono-objetivo (objetivo simples). Na Seção 4.1.4 são apresentadas as principais técnicas convencionais para resolver MOO.

4.1.1 Problemas de Otimização Multi-Objetivo

Um Problema de Otimização Multi-Objetivo (MOOP, do inglês *Multi-Objective Optimization Problem*) possui um conjunto de funções objetivo a serem otimizadas (maximizar ou minimizar). Além disso, possui restrições que devem ser satisfeitas para que uma solução seja factível ao problema. O enunciado geral de um MOOP é o seguinte

(Deb, 2001):

$$\left. \begin{array}{ll} \text{Maximizar/Minimizar} & f_m(x), & m = 1, 2, \dots, N_{obj}; \\ \text{sujeito a:} & g_j(x) \leq 0, & j = 1, 2, \dots, NR_{des}; \\ & h_k(x) = 0, & k = 1, 2, \dots, NR_{igu}; \\ & x_i^{(inf)} \leq x_i \leq x_i^{(sup)}, & i = 1, 2, \dots, N_{var} \end{array} \right\} \quad (4.1)$$

onde x é um vetor de N_{var} variáveis de decisão, $x = (x_1, x_2, \dots, x_{N_{var}})^T$, também denominado de **solução**. Os valores $x_i^{(inf)}$ e $x_i^{(sup)}$ representam os limites inferior e superior, respectivamente, para a variável x_i . Esses limites definem o **espaço de variáveis de decisão** ou **espaço de decisão** S_{dec} . As NR_{des} desigualdades (g_j) e as NR_{igu} igualdades (h_k) são chamadas de funções de restrição. Uma solução x factível satisfaz as $NR_{igu} + NR_{des}$ funções de restrição e os $2N_{var}$ limites. Caso contrário, a solução não será factível. O conjunto de todas as soluções factíveis formam a **região factível** ou **espaço de busca** S_{fact} .

Cada função $f_m(x)$ pode ser maximizada ou minimizada. Porém, para trabalhar com os algoritmos de otimização, é necessário converter todas as funções para serem apenas de maximização ou minimização. O vetor de funções objetivo $f(x) = [f_1(x), f_2(x), \dots, f_{N_{obj}}(x)]$ compõe um espaço multidimensional chamado **espaço de objetivos** S_{obj} . Para cada solução x no espaço de decisão, existe um $f(x)$ em S_{obj} . Esta é uma diferença fundamental em relação à otimização de objetivos simples, cujo espaço de objetivos é unidimensional. O mapeamento ocorre então entre um vetor x (de dimensão N_{var}) e um vetor $f(x)$ (de dimensão N_{obj}). Por exemplo, se cada elemento de x e $f(x)$ são números reais, então $f(x)$ estaria mapeada como $f(x) : \mathfrak{R}^{N_{var}} \rightarrow \mathfrak{R}^{N_{obj}}$.

Solução Pareto-Ótimas

As funções objetivo empregadas nos MOOPs são em geral **conflitantes** entre si. Uma função objetivo f_1 é conflitante com uma outra função f_2 quando não é possível melhorar o valor de f_1 sem piorar o valor da função f_2 . Um exemplo prático de objetivos conflitantes são preço e desempenho na compra de equipamentos, como por exemplo, computadores. Os equipamentos de maior custo apresentam usualmente um melhor

desempenho e vice-versa. Assim, em uma compra devem ser considerados vários modelos de computadores com diversos valores nos objetivos de preço e desempenho. Se ambos os objetivos possuem a mesma importância (ou prioridade), não há como afirmar, por exemplo, que certa redução do preço compensa determinada perda do desempenho.

Em um MOOP, emprega-se o conceito de **dominância de Pareto** para comparar duas soluções factíveis de um problema. Dadas duas soluções x e y , diz-se que x domina y (denotado como $x \preceq y$) se as seguintes condição forem satisfeitas:

1. A solução x é pelo menos igual a y em todas as funções objetivo;
2. A solução x é superior a y em pelo menos uma função objetivo.

Desta forma, existe um conjunto de soluções que possuem vantagens em desempenho mas que não são melhores em custo e vice-versa. Ou seja, existe um conjunto de alternativas ótimas que são **não dominadas** entre si nos objetivos de custo e desempenho. Em um MOOP, o conjunto de soluções não dominadas é chamado de **conjunto Pareto-ótimo**, que representa as soluções ótimas do problema. A **fronteira de Pareto** é o conjunto dos valores das funções objetivo das soluções do conjunto Pareto-ótimo. A figura 4.1 ilustra: os valores de preço e desempenho (de 0 a 100) para cinco alternativas para o exemplo de compra de computadores; as relações de dominância entre as soluções; o conjunto Pareto-Ótimo; e a fronteira de Pareto.

De acordo com as 5 alternativas de compra, indicadas na Figura 4.1, temos:

- Relação de dominância: $3 \preceq 2$, $5 \preceq 1$, $5 \preceq 2$;
- Conjunto de Pareto-ótimo: $\{3, 4, 5\}$;
- Fronteira de Pareto: indicada na figura 4.1.

Dominância de Pareto: Definição e Propriedades

Nesta seção serão apresentados, de forma mais formal, os conceitos descritos anteriormente.

Definição 4.1.1 *Uma solução x domina uma solução y ($x \preceq y$) se as seguintes condições são satisfeitas:*

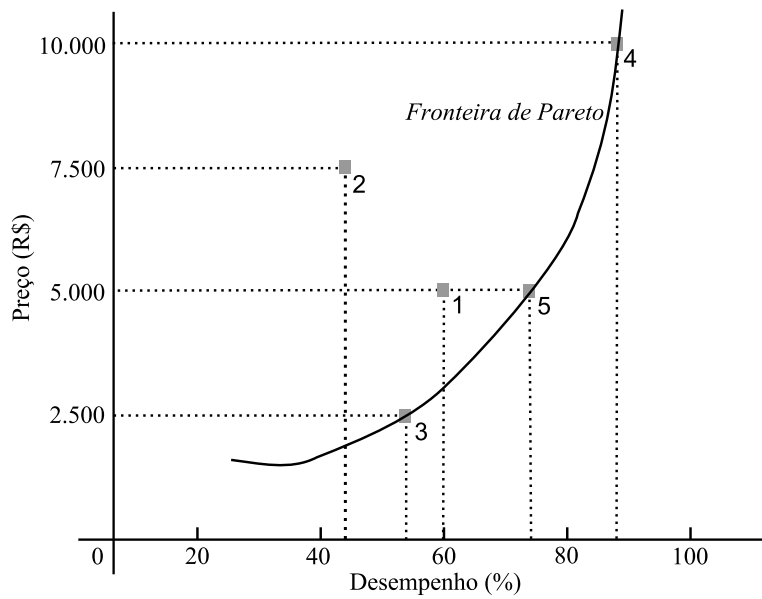


Figura 4.1: Exemplo que ilustra o preço e o desempenho para 5 alternativas de compra de computadores.

1. A solução x não é pior que y em nenhum dos N_{obj} , ou seja, $f_m(x) \leq f_m(y)$ para todo $m = 1, 2, \dots, N_{obj}$;
2. A solução x é estritamente melhor que y pelo menos em um objetivo, ou seja, $f_m(x) < f_m(y)$ pelo menos para um valor de m .

Vale ressaltar que a definição 4.1.1 é aplicada em um MOOP onde as funções objetivo devem ser minimizadas. Se ambas as condições desta definição são satisfeitas, pode-se dizer que:

1. y é dominada por x ;
2. x não é dominada por y ;
3. x não é inferior que y .

Na figura 4.1 temos que a solução 5 domina a solução 1 ($5 \preceq 1$), e a solução 3 domina a solução 2 ($3 \preceq 2$).

A relação de dominância satisfaz as seguintes propriedades:

1. Não é reflexiva. Conforme a definição 4.1.1, uma solução não pode ser dominada por si mesma;

2. Não é simétrica, ou seja, $x \preceq y$ não implica que $y \preceq x$;
3. É transitiva, isto é, se $x \preceq y$ e $y \preceq z$ então $x \preceq z$.

Essas propriedades caracterizam a relação de dominância como uma relação de ordem parcial estrita (Deb, 2001). Um conjunto de soluções para um MOOP, pode ser dividido em um conjunto de soluções dominadas e não-dominadas empregando o operador de dominância.

Definição 4.1.2 *Dado um conjunto de soluções \mathcal{P} , o conjunto de soluções não-dominadas \mathcal{P}' é formado por:*

$$\mathcal{P}' = \{x \in \mathcal{P} \mid \nexists y \in \mathcal{P} : y \preceq x\}. \quad (4.2)$$

Quando um conjunto de soluções \mathcal{P} corresponde ao conjunto de soluções factíveis de um MOOP ($\mathcal{P} = S_{fact}$), o conjunto não-dominado \mathcal{P}' é chamado de **conjunto Pareto-ótimo**. Utiliza-se também em MOOP o conceito de otimalidade local. Um conjunto Pareto-ótimo local é definido conforme segue:

Definição 4.1.3 *Dado um conjunto de soluções \mathcal{P} e ϵ , um número positivo arbitrariamente pequeno, o conjunto Pareto-ótimo local \mathcal{P}'' é formado por:*

$$\mathcal{P}'' = \{x \in \mathcal{P} \mid \nexists y \in \mathcal{P} : y \prec x \wedge \|y - x\|_{\infty} \leq \epsilon\} \quad (4.3)$$

A Figura 4.2 ilustra dois conjuntos Pareto-ótimos que são não-dominados localmente, mostrando a sua vizinhança no seu espaço de objetivos e no espaço de variáveis.

Finalmente, a fronteira de Pareto de um MOOP pode ser definida:

Definição 4.1.4 *Dado um MOOP com f_m , $m = 1, 2, \dots, N_{obj}$ funções objetivo, e cujo conjunto Pareto-ótimo é \mathcal{P}' , a fronteira de Pareto \mathcal{PF} é formada por:*

$$\mathcal{PF} = \{f(x) \mid x \in \mathcal{P}'\}, \quad (4.4)$$

onde $f(x) = [f_1(x), f_2(x), \dots, f_{N_{obj}}(x)]$ é o vetor de funções objetivo para a solução x

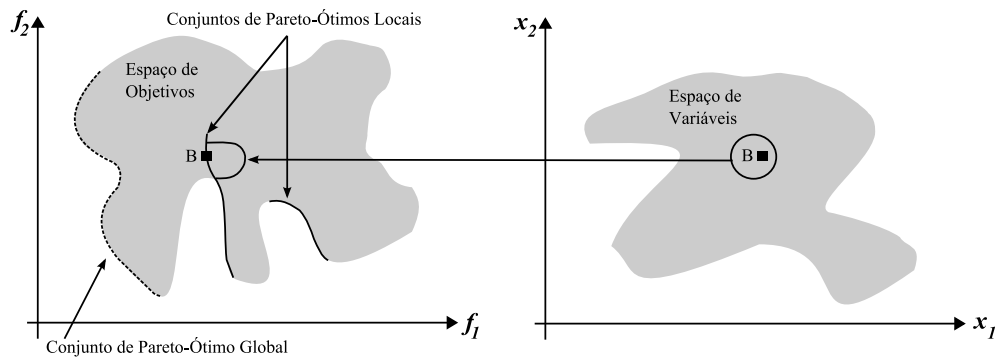


Figura 4.2: Soluções de Pareto-ótimas locais e global.

4.1.2 Metas em Otimização Multi-Objetivo

Em (Deb, 2001) são destacadas três importantes metas em otimização multi-objetivo:

1. Encontrar um conjunto de soluções que esteja o mais próximo possível da fronteira de Pareto;
2. Encontrar um conjunto de soluções com a maior diversidade possível;
3. Realizar as duas metas anteriores com a maior eficiência computacional possível.

A primeira meta é comum a qualquer processo de otimização. Soluções muito distantes da fronteira de Pareto não são desejáveis. Por outro lado, encontrar a maior diversidade dentro das soluções é a meta específica para otimização multi-objetivo. A figura 4.3a ilustra uma distribuição quase uniforme das soluções na fronteira de Pareto. A figura 4.3b ilustra a fronteira com soluções apenas em algumas regiões, isto é, com baixa diversidade. É necessário assegurar a maior cobertura possível da fronteira.

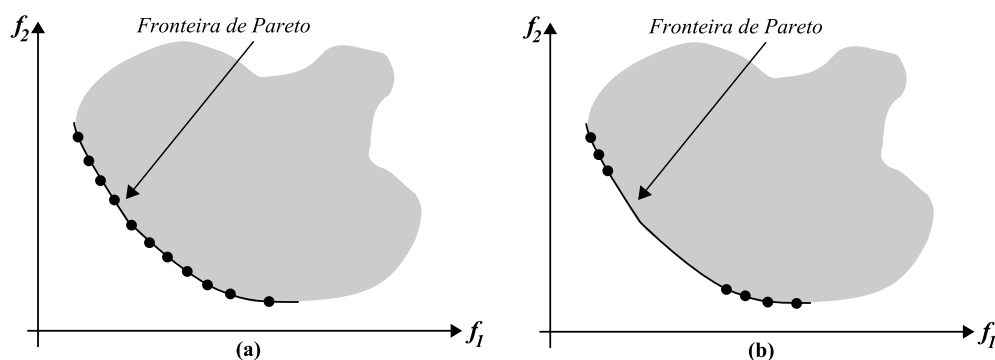


Figura 4.3: Diferentes distribuições de soluções na fronteira de Pareto.

Como em MOOP trabalha-se com o espaço de decisões e o espaço de objetivos, é também desejável que as soluções estejam adequadamente distribuídas em ambos os espaços. Em geral, a diversidade em um desses espaços garante também a diversidade no outro. Entretanto, para alguns problemas isso não acontece. Tendo em vista que encontrar um conjunto de soluções uniformemente distribuído é uma tarefa que pode consumir consideráveis recursos computacionais (Deb, 2001), é necessário que tais soluções sejam obtidas eficientemente.

4.1.3 Diferenças entre Otimização Multi-Objetivo e a Otimização Mono-Objetivo

Em Deb (2001) identificam-se três importantes aspectos que diferenciam a otimização multi-objetivo e a otimização mono-objetivo, estes sendo:

1. Em problemas de otimização com um único objetivo (otimização mono-objetivo), a meta é encontrar uma solução ótima global. Se a função objetivo desses problemas for multimodal, poderia existir mais de um ótimo global. Neste caso, todos os ótimos são equivalentes. Por outro lado, em MOOP, determinar um conjunto de soluções da fronteira de Pareto é tão importante quanto preservar a diversidade neste conjunto. Um algoritmo eficiente para otimização multi-objetivo deve considerar ambos os aspectos;
2. Um MOOP trabalha com dois espaços, das variáveis e dos objetivos. Por outro lado, problemas de objetivo simples trabalham unicamente no espaço de variáveis, pois procuram apenas uma solução no espaço de objetivos. Manter a diversidade em ambos espaços complica mais o problema, dado que a proximidade de duas soluções no espaço de variáveis não implica proximidade no espaço de objetivos;
3. Os métodos tradicionais de otimização multi-objetivo reduzem o conjunto de funções objetivo a uma função simples que pondera cada objetivo. Estes métodos podem também tratar cada objetivo separadamente, utilizando os demais objetivos como restrições. Portanto, um MOOP pode ser convertido, por meio de algumas técnicas, em um problema de otimização simples.

4.1.4 Técnicas Tradicionais para MOOP

Nesta seção serão descritas as principais técnicas clássicas usadas em MOOP: somatório de pesos, métodos de restrições ϵ e programação por metas.

Somatório de pesos

O método de somatório de pesos consiste em criar uma função objetivo somando cada objetivo multiplicado por um peso (Deb, 2001). Os pesos são fornecidos como parâmetros. A escolha dos pesos é um problema importante que depende da relevância de cada objetivo. É necessário realizar a normalização de cada função objetivo dado que os diferentes objetivos podem ter diferentes magnitudes. Por exemplo, o preço de um carro pode variar de R\$4.000 a R\$30.000; enquanto o conforto pode estar entre 0% e 100%.

Uma vez que os objetivos estejam normalizados, pode-se formular uma função $F(x)$ que soma os objetivos normalizados e multiplicados por seus respectivos pesos. Assim, um MOOP pode ser formulado como segue:

$$\left. \begin{array}{l} \text{Minimizar } F(x) = \sum_{m=1}^{N_{obj}} w_m f_m(x), \\ \text{sujeito a: } g_j(x) \leq 0, \quad j = 1, 2, \dots, NR_{des}; \\ \quad \quad \quad h_k(x) = 0, \quad k = 1, 2, \dots, NR_{igu}; \\ \quad \quad \quad x_i^{(inf)} \leq x_i \leq x_i^{(sup)}, \quad i = 1, 2, \dots, N_{var} \end{array} \right\} \quad (4.5)$$

onde $w_m \in [0, 1]$ é o peso para cada função objetivo f_m . Pode-se mostrar que a solução do problema na Equação 4.5 pertence ao conjunto Pareto-ótimo se os pesos são positivos para todos os objetivos. Além disso, pode-se garantir que se um MOOP é convexo (Deb, 2001), qualquer solução Pareto-ótima pode ser encontrada usando o método de somatório dos pesos, empregando diferentes combinações de valores de w_m .

Seja um MOOP com dois objetivos. O espaço de objetivos e a Fronteira de Pareto são mostrados na Figura 4.4. Tem-se um vetor de pesos $w = (w_1, w_2)$ para cada objetivo.

Dado um vetor de pesos w é possível plotar o contorno de F no espaço de objetivos. Dado que F é uma combinação linear dos objetivos, obtém-se uma linha reta. Encontrar o mínimo valor da equação 4.5 é equivalente a achar uma linha de contorno com um

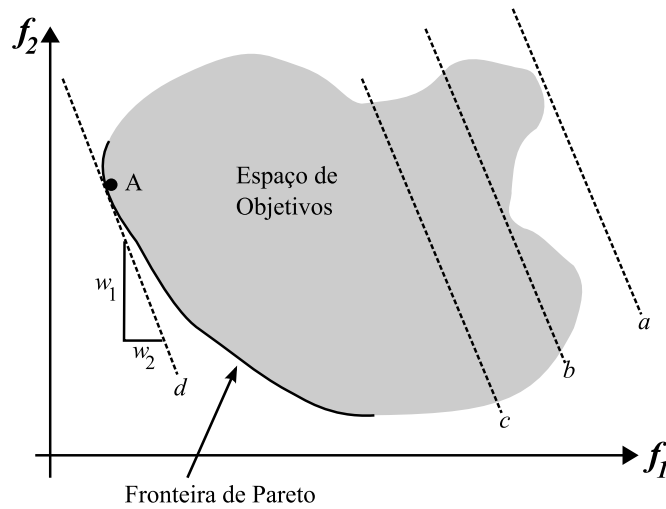


Figura 4.4: O método do somatório de pesos.

valor mínimo para F .

A Figura 4.4 mostra várias linhas de contorno para F , sendo que a linha d é tangencial a um ponto do espaço de objetivos (A). Esse ponto encontra-se na Fronteira de Pareto e, conseqüentemente, é uma solução Pareto-ótima. Modificando os valores para w_1 e w_2 encontra-se uma outra solução Pareto-ótima.

Embora esse método seja simples, precisa de várias iterações para atingir toda a fronteira de Pareto. No caso de um MOOP não convexo, este método não é capaz de determinar todas as soluções. Além disso, a aplicação de vetor de pesos uniformemente distribuídos não garante que seja obtido um conjunto de soluções uniformemente distribuídas.

Método de restrições ϵ

Haimes et al. (1971) sugeriram uma reformulação de MOOPs considerando qualquer objetivo, mantendo restritos os demais objetivos com valores definidos pelo usuário. A formulação adotada é descrita a seguir:

$$\left. \begin{array}{ll}
 \text{Minimizar } f_u(x), & m = 1, 2, \dots, N_{obj}; \\
 \text{sujeito a: } f_m(x) \leq \epsilon_m, & m = 1, 2, \dots, N_{obj} \text{ e } m \neq u; \\
 g_j(x) \leq 0, & j = 1, 2, \dots, NR_{des}; \\
 h_k(x) = 0, & k = 1, 2, \dots, NR_{igu}; \\
 x_i^{(inf)} \leq x_i \leq x_i^{(sup)}, & i = 1, 2, \dots, N_{var}
 \end{array} \right\} \quad (4.6)$$

onde cada ϵ_m representa um limite máximo para o valor de f_m . Por exemplo, para um MOOP não convexo de dois objetivos f_1 e f_2 , escolhe-se f_2 para ser minimizado e mantém-se f_1 com a restrição $f_1 < \epsilon_1$.

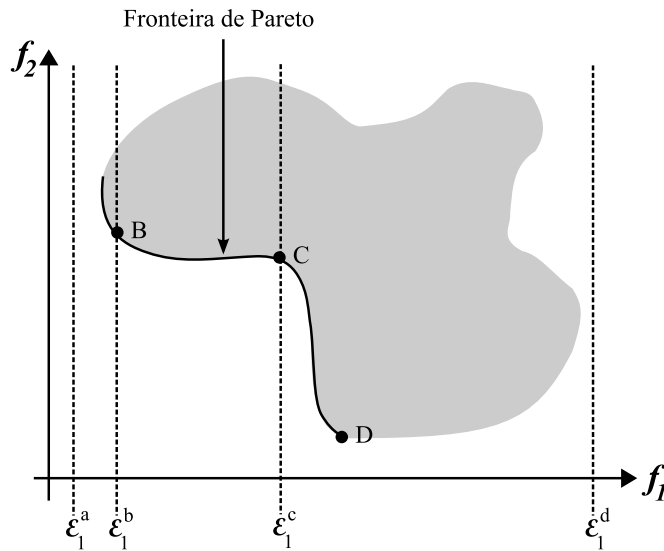


Figura 4.5: Método de restrições ϵ (Deb, 2001).

A Figura 4.5 apresenta o espaço de objetivos e vários valores para ϵ_1 . O mínimo para f_2 depende da escolha do ϵ . Por exemplo, usando ϵ_1^c , o valor mínimo para f_2 é o ponto C . Então, empregando valores diferentes de ϵ , encontram-se diferentes soluções Pareto-ótimas.

Desta forma, o método de restrições ϵ pode ser usado para gerar soluções Pareto-ótimas independentemente de o espaço de objetivos ser convexo, não convexo ou discreto (Deb, 2001). Esse método precisa que a escolha do vetor ϵ esteja em uma região factível para cada objetivo. Por exemplo, na Figura 4.5, se for escolhido ϵ_1^a , nenhuma solução será obtida. Assim, como no somatório de pesos, são necessárias várias iterações para determinar a fronteira de Pareto e o uso de uma distribuição uniforme de ϵ não garante um conjunto de soluções com a mesma distribuição.

Programação por metas

Esta técnica tenta encontrar soluções que possam atingir uma meta pré-determinada para uma ou mais funções objetivo. Caso não exista uma solução factível que alcance as metas para todos os objetivos, esta minimiza os *desvios* em relação às metas.

Considere uma função $f(x)$ para ser minimizada dentro do espaço de busca S_{fact} . Para cada objetivo é escolhido, pelo usuário, um valor meta z . Então, o problema é formulado para encontrar uma solução cujo valor em f seja igual a z .

Para resolver um problema de programação por metas, cada meta é convertida em uma restrição de igualdade. Busca-se então minimizar todos os desvios em relação às metas. Existem várias formas de trabalhar com esses problemas, estas descritas a seguir:

- **Programação de metas com pesos:** para um problema com N_{obj} objetivos, formula-se uma função somando os desvios para cada um dos N_{obj} objetivos. A formulação geral desse problema pode ser descrita da seguinte forma:

$$\left. \begin{array}{l} \text{Minimizar } F(x) = \sum_{m=1}^{N_{obj}} (\alpha_m \phi_m + \beta_m \eta_m), \\ \text{sujeito a: } f(x) - \phi_m + \eta_m = z_m, \quad m = 1, 2, \dots, N_{obj}; \\ x \in S_{fact}, \\ \phi_m, \eta_m \geq 0, \end{array} \right\} \quad (4.7)$$

onde α_m e β_m são os pesos dos desvios positivo e negativo (α_m e η_m , respectivamente) para o j -ésimo objetivo, z_m é a meta para a função f_m e S_{fact} é o espaço de decisão factível. As soluções obtidas por este métodos dependem consideravelmente da escolha dos valores para α_m e β_m . Além disso, segundo (Deb, 2001), este método possui dificuldades similares ao método do somatório de pesos;

- **Programação de metas lexicográficas:** aqui metas são organizadas em vários níveis de prioridade. Resolvem-se sequencialmente vários problemas de programação de metas. Inicialmente, as metas de primeira ordem de prioridade são consideradas na formulação do problema. Caso existam múltiplas soluções, as metas de segunda ordem de prioridade são consideradas formulando outro prob-

lema para minimizar apenas os desvios para as metas de segunda ordem. As metas de primeira ordem de prioridade são usadas como restrições. O processo continua com os demais níveis de prioridade até que seja encontrada uma única solução. Utilizando esse método, é encontrada frequentemente uma solução Pareto-ótima. A Figura 4.6 mostra um espaço de objetivos para as funções f_1 e f_2 . Se f_1 é mais importante, minimiza-se f_1 primeiro e obtém-se as soluções das regiões AB e CD nas quais f_1 é mínima. Dado que existem múltiplas soluções, minimiza-se f_2 somente nas regiões AB e CD , encontradas na iteração anterior. A solução é o ponto D , que corresponde ao mínimo para f_2 . Então, D é a solução para todo o problema de programação de metas lexicográficas.

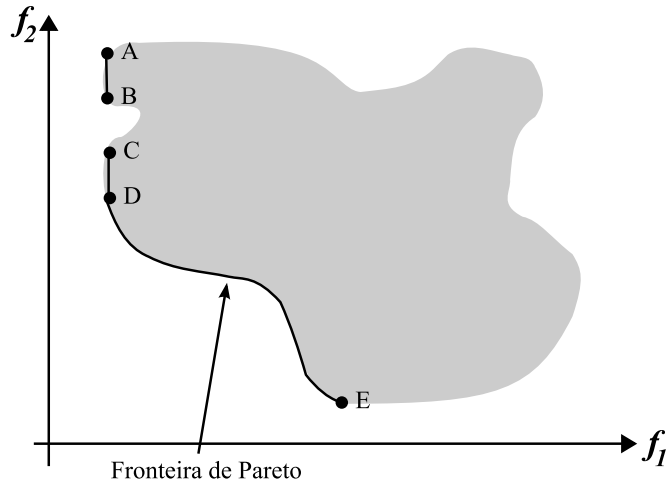


Figura 4.6: Método de programação de metas lexicográficas (Deb, 2001).

- **Programação de metas min-max:** neste método é minimizado o máximo desvio em relação às metas. A formulação adotada é a seguinte:

$$\left. \begin{array}{l}
 \text{Minimizar } \delta, \\
 \text{sujeito a: } \alpha_m \phi_m + \beta_m \eta_m \leq \delta, \quad m = 1, 2, \dots, N_{obj}; \\
 f_j(x) - \phi_m + \eta_m = z_m, \\
 x \in S_{fact}, \\
 \phi_m, \eta_m \leq 0,
 \end{array} \right\} \quad (4.8)$$

onde δ é o desvio máximo para qualquer meta, ϕ_m e η_m são os desvios positivos e negativos para cada objetivos, α_m e β_m representam os pesos para cada desvio. Este método requer também a escolha dos pesos α_m e β_m .

Vantagens e desvantagens das técnicas tradicionais

A principal vantagem das técnicas tradicionais é que possuem provas de convergência que garantem encontrar pelo menos uma solução Pareto-ótima (Coello et al., 2002). Todas as técnicas descritas neste Capítulo reduzem o MOOP para um problema de objetivo simples. Cada técnica utiliza uma forma diferente de redução e introduz parâmetros adicionais. A escolha desses parâmetros afeta diretamente os resultados obtidos. Cada vez que os parâmetros são modificados, é necessário resolver um novo problema de otimização simples. Portanto, para encontrar cada solução Pareto-ótima, precisa-se solucionar um problema de objetivo simples.

Alguns métodos não garantem soluções ao longo de toda a fronteira de Pareto. Se esta não é convexa, o método do somatório de pesos não encontra certas soluções, independentemente dos pesos escolhidos.

Finalmente, todas as técnicas descritas precisam de parâmetros adicionais, tais como pesos, metas e vetores de restrição. Além disso, a distribuição uniforme desses parâmetros não garante a diversidade das soluções Pareto-ótimas. Porém, existem técnicas alternativas para tratar MOOPs. Dentre dessas técnicas, destacam-se os AEs que apresentam vários aspectos positivos que motivam a aplicação dos mesmos.

4.2 Algoritmos Evolutivos para Otimização Multi-Objetivo

Os AEs são promissores para serem empregados em MOOP, em razão de apresentarem as seguintes características: trabalham com mais de uma função simultaneamente; não precisam de informações adicionais e são capazes de escapar de ótimos locais. Essa seção apresenta os conceitos envolvidos no desenvolvimento de AEs para MOOPs.

A primeira implementação de AEs para MOOPs foi proposta por Schaffer (1985). O modelo sugerido foi denominado VEGA (do inglês *Vector Evaluated Genetic Algorithm*). Schaffer fez uma modificação no AG convencional para avaliar cada objetivo separadamente. Contudo, o método proposto não obtém uma diversidade adequada nas soluções ao longo da fronteira de Pareto.

Goldberg (1989) propôs diversas abordagens para estender a aplicação de AEs para

MOOPs. Uma das propostas utiliza um procedimento para ordenação de soluções baseado no conceito de dominância. Nesse método, o valor de *fitness* para uma solução i é proporcional ao número de soluções que i domina. Desta forma, as soluções não dominadas são enfatizadas obtendo maior quantidade de cópias na lista de reprodução.

Para manter a diversidade das soluções, Goldberg sugeriu o emprego de um método de compartilhamento (Goldberg, 1989), que permite levar em conta a densidade de soluções em uma vizinhança no espaço de busca. Assim, soluções que estejam melhor espalhadas na fronteira de Pareto têm um melhor valor de compartilhamento.

Uma diversidade de modelos de MOEAs foram propostos baseados nessas idéias iniciais. A principal diferença dos MOEAs, em relação aos AEs tradicionais, é o operador de seleção, dado que a comparação entre duas soluções é efetuada com base no conceito de dominância de Pareto. Em alguns métodos, o valor de *fitness* é proporcional à dominância da solução, outros métodos utilizam apenas a dominância de Pareto e não calculam o valor de *fitness* com base no nível de dominância. Em (Coello et al., 2002) apresentam-se três grandes vantagens da aplicação dos MOEAs para MOOP com relação às técnicas tradicionais:

1. Não introduzem parâmetros adicionais no problema;
2. Trabalham diretamente com várias funções usando o conceito de dominância de Pareto;
3. Um conjunto diversificado de soluções pode ser encontrado apenas em uma execução do MOEA.

De acordo com Deb (2001), modelos de MOEA são classificados em dois tipos:

1. **Não elitistas:** modelos que não utilizam alguma forma de elitismo nas suas iterações;
2. **Elitistas:** modelos que empregam alguma forma de elitismo.

A Tabela 4.1 enumera os principais modelos de MOEAs. A seguir será apresentada o NSGA-II, que é a técnica utilizada neste projeto, por ser uma das técnicas mais utilizadas na literatura e por apresentar um bom desempenho quando aplicada ao PRE, conforme pode ser visto em (Kumar et al., 2008).

Tabela 4.1: Diferentes modelos de MOEAS.

| Sigla | Autoria | Elitista |
|---|--|----------|
| VEGA(<i>Vector Evaluated Genetic Algorithm</i>) | (Schaffer, 1985) | Não |
| WBGA(<i>Weight Based Genetic Algorithm</i>) | (Hajela e Lin, 1992) | Não |
| MOGA(<i>Multiple Objective Genetic Algorithm</i>) | (Fonseca e Fleming, 1993) | Não |
| NSGA(<i>Non-Dominated Sorting Genetic Algorithm</i>) | (Srinivas e Deb, 1994) | Não |
| NPGA(<i>Niched-Pareto Genetic Algorithm</i>) | (Horn et al., 1994) | Não |
| PPES(<i>Predator-Prey Evolutionary Strategy</i>) | (Laumanns et al., 1998) | Não |
| REMOEA(<i>Rudolph's Elitist Multi-Objective Evolutionary Algorithm</i>) | (Rudolph, 2001) | Sim |
| NSGA-II(<i>Elitist Non-Dominated Sorting Genetic Algorithm</i>) | (Deb et al., 2000; Deb e SUNDAR, 2006a) | Sim |
| SPEA, SPEA2(<i>Strength Pareto Evolutionary Algorithm</i>) 1 e 2 | (Zitzler e Thiele, 1999; Zitzler et al., 2001) | Sim |
| TGA(<i>Thermodynamical Genetic Algorithm</i>) | (Kita et al., 1996) | Sim |
| PAES(<i>Pareto-Archived Evolutionary Strategy</i>) | (Knowles e Corne, 1999) | Sim |
| MOMGA-I, MOMGA-II(<i>Multi-Objective Messy Genetic Algorithm</i>) I e II | (Van Veldhuizen, 1999) | Sim |
| Micro-GA(<i>Multi-Objective Micro-Genetic Algorithm</i>) | (Coello et al., 2002) | Sim |
| PESA-I, PESA-II(<i>Pareto Envelope-Base Selection Algorithm</i>) | (Corne et al., 2000; Corne et al., 2001) | Sim |
| ϵ -MOEA(<i>ϵ-dominance Multi-Objective Evolutionary Algorithm</i>) | (Deb et al., 2005) | Sim |

4.2.1 NSGA-II: Elitist Non-Dominated Sorting Genetic Algorithm

Proposto por Deb et al. (2000), o algoritmo *Elitist Non-Dominated Sorting Genetic Algorithm* (NSGA-II) baseia-se na ordenação elitista por dominância chamada de *Pareto ranking*. Esse procedimento consiste em classificar as soluções de um conjunto M em diversas fronteiras ($\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_k$, onde k é o número de fronteiras) conforme o grau de dominância de cada solução. Deste modo, a fronteira \mathcal{F}_1 contém as soluções não dominadas de todo o conjunto de soluções M , \mathcal{F}_2 contém as soluções não dominadas de $M - \mathcal{F}_1$, \mathcal{F}_3 contém as soluções não dominadas de $M - (\mathcal{F}_1 \cup \mathcal{F}_2)$ e assim sucessivamente.

O procedimento de ordenação por não dominância proposto por Deb et al. (2000) é descrito no algoritmo 2. Para cada solução i , contida em P , são calculados dois valores:

- nd_i , o número de soluções que dominam a solução i ;
- U_i , o conjunto de soluções que são dominadas pela solução i .

As linhas 1-15 do Algoritmo 2 calculam tais valores para as soluções em M . Além disso, as soluções com $nd_i = 0$ estão contidas na fronteira \mathcal{F}_1 . Em seguida, as linhas

17-29 percorrem o conjunto de soluções dominadas U_i para cada solução i em \mathcal{F}_1 . O contador nd_j , de cada solução j em U_i , é decrementado em 1. Se $nd_j = 0$, então a solução j pertence à próxima fronteira, neste caso, \mathcal{F}_2 . A iteração das linhas 17-29 é repetida até que todas as soluções estejam classificadas em uma fronteira. A Figura 4.7 ilustra este procedimento aplicado às soluções que minimizam f_1 e f_2 .

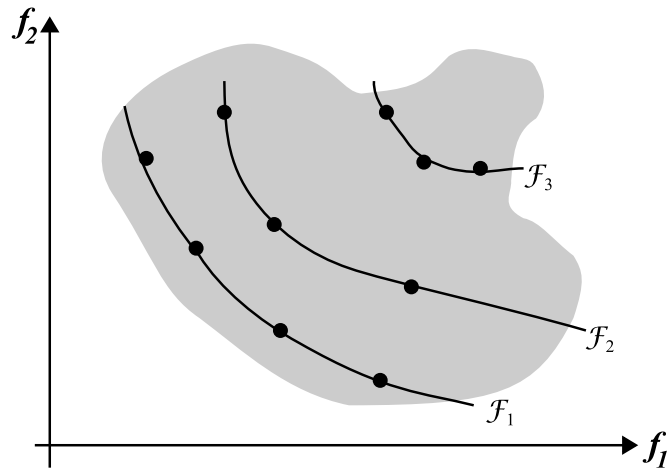


Figura 4.7: Ordenação por não dominância (Deb, 2001).

O algoritmo NSGA-II trabalha com duas populações, denotadas por P e Q , ambas de tamanho N_{ind} . As populações P e Q em cada iteração $t = 1, 2, \dots, N_{iter}$ são denotadas por P_t e Q_t , respectivamente. Na primeira geração, os indivíduos iniciais da população P_1 geram as soluções em Q_1 , através da aplicação dos operadores genéticos. Em seguida, estabelece-se um processo competitivo para preencher N_{ind} vagas para a solução P_{t+1} , entre $2 * N_{ind}$ indivíduos contido em $R_t = P_t \cup Q_t$. Esta operação é realizada utilizando a ordenação por não dominância em R_t , encaminhando as soluções não dominadas contidas nas fronteiras diretamente para a próxima geração (elitismo).

Para garantir a diversidade na fronteira, o NSGA-II emprega uma estimativa de densidade das soluções que rodeiam cada indivíduo da população. Assim, calcula-se a média da distância das duas soluções adjacentes a cada indivíduo para todos os objetivos. Esse valor é denominado distância de multidão. O Algoritmo 3 descreve os passos para calcular tal valor, onde $crowdist_n$ é o valor da distância de multidão do n -ésimo indivíduo do conjunto M (denotado como M_n) e $f_m(M_n)$ é o valor da m -ésima função objetivo para o n -ésimo indivíduo.

O *fitness* de cada solução i é determinado pelos seguintes valores:

Algoritmo 2 Ordenação por não-dominância

```

1: for solução  $i \in M$  do
2:    $nd_i \leftarrow 0$ 
3:    $U_i \leftarrow \phi$ 
4:   for solução  $j \neq i$  e  $j \in M$  do
5:     if  $i \prec j$  then
6:        $U_i \leftarrow U_i \cup \{j\}$ 
7:     end if
8:     if  $j \prec i$  then
9:        $nd_i \leftarrow nd_i + 1$ 
10:    end if
11:  end for
12:  if  $nd_i = 0$  then
13:     $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{i\}$ 
14:  end if
15: end for
16:  $k \leftarrow 1$ 
17: while  $\mathcal{F}_k \neq \phi$  do
18:    $Temp \leftarrow \phi$ 
19:   for solução  $i \in \mathcal{F}_k$  do
20:     for solução  $j \in U_i$  do
21:        $n_j \leftarrow n_j - 1$ 
22:       if  $n_j = 0$  then
23:          $Temp \leftarrow Temp \cup \{j\}$ 
24:       end if
25:     end for
26:   end for
27:    $k \leftarrow k + 1$ 
28:    $\mathcal{F}_k \leftarrow Temp$ 
29: end while

```

Algoritmo 3 Cálculo da distância de multidão.

```

1: for solução  $n \leftarrow 1, 2, \dots, |M|$  do
2:    $dist_n \leftarrow 0$ 
3: end for
4: for  $m \leftarrow 1, 2, \dots, N_{obj}$  do
5:   Classificar  $M$  por  $f_m$ , em ordem decrescente
6:    $crowdist_1 \leftarrow crowdist_{|M|} \leftarrow \infty$ 
7:   for  $n \leftarrow 2, \dots, |M| - 1$  do
8:     Classificar  $M$  por  $f_m$ , em ordem decrescente
9:      $crowdist_n \leftarrow crowdist_n + \frac{f_m(M_{n+1}) - f_m(M_{n-1})}{f_m^{max} - f_m^{min}}$ 
10:  end for
11: end for

```

1. $rank_i = k$, o valor de *ranking* i é igual ao número da fronteira \mathcal{F}_k à qual i pertence;
2. $crowdist_i$, o valor de distância de multidão de i .

O NSGA-II emprega um processo de seleção por torneio, que é guiado por um novo operador denominado *crowded-comparison operator* (\prec_c). Em tal abordagem, duas soluções são comparadas para escolher qual delas vai gerar descendentes na nova população. Uma solução i é escolhida sobre uma solução j se:

1. i possui um *ranking* menor que j , ou seja, $rank_i < rank_j$;
2. Se ambas as soluções possuem o mesmo *ranking* e i possui um maior valor de distância de multidão, ou seja, $rank_i = rank_j$ e $crowdist_i > crowdist_j$.

O cálculo da distância de multidão permite que as soluções melhores espalhadas passem a ocupar as últimas vagas disponíveis de P_{t+1} , garantindo a diversidade das soluções. A população Q_{t+1} é gerada utilizando os operadores de seleção por torneio, recombinação e mutação em P_{t+1} . O NSGA-II continua por N_{iter} iterações e as soluções finais encontram-se em $P_{N_{iter}} \cup Q_{N_{iter}}$. A sequência de passos do pelo NSGA-II é descrita no Algoritmo 4. A Figura 4.8 ilustra o esquema para uma iteração do NSGA-II.

Algoritmo 4 NSGA-II.

- 1: Criar uma população de soluções aleatórias P_1 de N_{ind} indivíduos
 - 2: Ordenar P_1 utilizando o algoritmo 2
 - 3: Gerar a população Q_1 de tamanho N_{ind} , aplicando os operadores genéticos em P_1
 - 4: **for** geração $t \leftarrow 1, 2, \dots, N_{iter}$ **do**
 - 5: Aplicar algoritmo 2 em $R_t \leftarrow P_t \cup Q_t$
 - 6: $k \leftarrow 1$
 - 7: **while** $|P_{t+1} + \mathcal{F}_k| \leq N_{ind}$ **do**
 - 8: Aplicar algoritmo 3 em \mathcal{F}_k
 - 9: $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_k$
 - 10: $k \leftarrow k + 1$
 - 11: **end while**
 - 12: Aplicar o algoritmo 3 em \mathcal{F}_k
 - 13: Classificar a fronteira \mathcal{F}_k pelo *ranking* e a distância usando o operador (\prec_c)
 - 14: Copiar as primeiras $N_{ind} - |P_{t+1}|$ soluções de \mathcal{F}_k para P_{t+1}
 - 15: Gerar a nova população Q_{t+1} aplicando os operadores genéticos em P_{t+1}
 - 16: **end for**
 - 17: $P_{final} \leftarrow P_t$
 - 18: $Q_{final} \leftarrow Q_t$
-

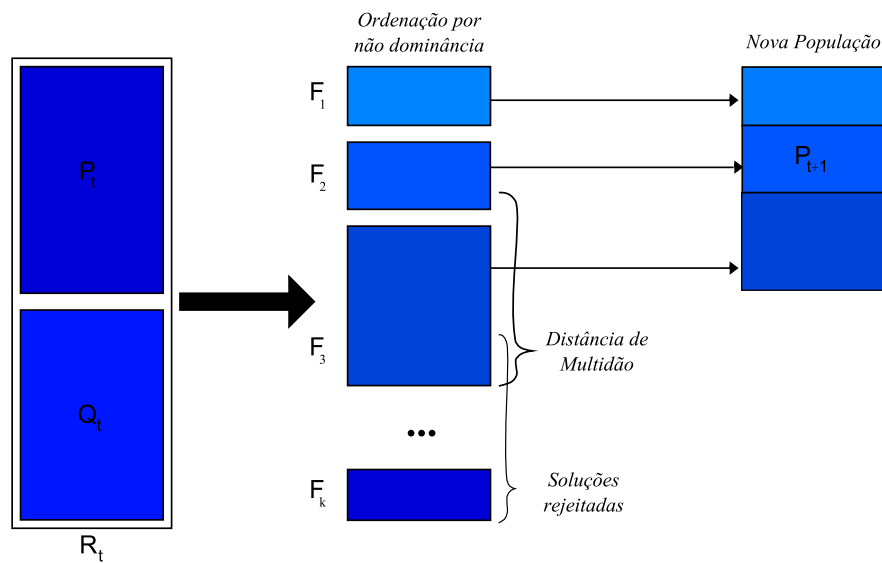


Figura 4.8: Esquema do modelo NSGA-II (Deb, 2001).

Complexidade Computacional

A complexidade do NSGA-II, no pior caso, é (Ticona e Delbem, 2008):

- Na etapa de ordenação por não dominância da população R é necessário comparar cada uma das $2*N$ soluções com $2*N - 1$ soluções, para cada um dos M objetivos. Isto resulta no total de comparações de ordem $O(M * N^2)$;
- Na etapa de cálculo da distância de multidão, o pior caso ocorre quando todas as soluções da população R estão na fronteira \mathcal{F}_1 , logo, é necessário ordenar \mathcal{F}_1 para cada objetivo M , isto resultando na ordem de $O(M * N * \log N)$;
- Finalmente, para selecionar as N melhores soluções da fronteira \mathcal{F}_1 , para a nova população P_{t+1} , é utilizado o operador \prec_c , o que resulta em uma ordem de $O(M * N^2)$.

Logo, a complexidade de tempo total do algoritmo NSGA-II é $O(M * N^2)$ (Deb, 2001).

Capítulo 5

Estruturas de Dados para AEs Aplicados a Problemas de Projeto de Redes

Este capítulo está organizado da seguinte forma: primeiramente, na Seção 5.1, são apresentados os conceitos básicos da teoria de grafos necessários para o entendimento da estrutura de dados RNP, que é utilizada neste trabalho para representar computacionalmente SDRs. Na Seção 5.2 são apresentadas representações de Problemas de Projeto de Redes (PPRs) para AEs e, na Seção 5.3, é descrita a RNP.

5.1 Principais Conceitos da Teoria de Grafos

Uma diversidade de problemas pode ser representada por meio de diagramas que consistem em um conjunto de pontos e linhas que conectam alguns desses pontos. Um exemplo conveniente são as redes elétricas, as quais, basicamente, consistem de um conjunto de barras e de ramos que as interconectam, onde cada barra pode ser representada por um ponto, e as conexões entre as mesmas podem ser representadas pelas linhas. A abstração matemática de problemas desse tipo dá lugar ao conceito de grafo.

Um **grafo** G consiste de um conjunto finito $N(G)$ de elementos, chamados **nós**, e um conjunto finito $E(G)$ de pares de nós não-ordenados, chamados **arestas**. Um grafo é simbolicamente representado por $G = (N, E)$. Se u e v são dois nós de um grafo, e se

o par u, v é uma aresta denotada por e , diz-se que e conecta u e v , como pode ser visto na Figura 5.1. Neste caso, a aresta u, v é dita incidente ao nó u e ao nó v .

A **ordem** de um grafo G é dada pelo número de elementos do conjunto finito $N(G)$, ou seja, pelo número de nós de G . O tamanho de um grafo G é dado pelo seu número de arestas, isto é, o tamanho $N(E)$. Assim, a ordem do grafo da Figura 5.1 é 4 e seu tamanho também é 4. O **grau** de um nó é dado pelo número de arestas que lhe são incidentes. A Tabela 5.1 apresenta o grau de cada nó do grafo da Figura 5.1.

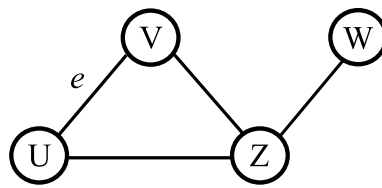


Figura 5.1: Exemplo de um grafo.

Tabela 5.1: Grau de cada um dos nós do grafo da figura 5.1.

| Nó | Grau |
|----|------|
| W | 1 |
| U | 2 |
| V | 2 |
| Z | 3 |

Dado um grafo G , uma seqüência de arestas $\{s_0, s_1\}, \{s_1, s_2\}, \dots, \{s_{m-2}, s_{m-1}\}, \{s_{m-1}, s_m\}$, em que todas as arestas são distintas, é chamada de **caminho**, isto é, um caminho é uma seqüência de nós, tal que de cada um dos nós exista uma única aresta distinta para o nó seguinte. Além disso, se os nós $s_0, s_1, s_2, \dots, s_{m-1}$ e s_m são distintos, nenhum dos nós se repete, então o caminho é chamado de **cadeia** ou **caminho simples**. O comprimento do caminho é o número de arestas que o caminho usa. Se somente os nós s_0 e s_m são iguais, o caminho é chamado de **ciclo**.

Da Figura 5.1, tem-se:

- Exemplo de um caminho: $\{w, z\}, \{z, u\}$ e $\{u, v\}$;
- Exemplo de uma cadeia: $\{u, v\}, \{v, z\}$ e $\{z, w\}$;
- Exemplo de um ciclo: $\{u, v\}, \{v, z\}$ e $\{z, u\}$;

Um par de nós de um grafo é um **par conexo**, se existir um caminho entre esses nós. Um grafo G é um grafo **conexo**, se todo par de nós em G é um par conexo.

Diz-se que H é um **subgrafo conexo máximo** de um grafo G , se um único subgrafo conexo contendo H é o próprio H . Um subgrafo conexo H máximo também é chamado de **componente**. Um grafo G é conexo se o número de seus componentes for igual a um.

Um grafo **acíclico** é um grafo sem ciclos. Uma **árvore** é um grafo acíclico conexo. Uma **floresta** é um grafo formado por um conjunto de árvores. Logo cada componente de uma floresta é uma árvore. Quando uma floresta tem apenas uma árvore, ela é uma floresta conexa. Assim, uma árvore é uma floresta conexa.

Geralmente, chama-se um dos nós de uma árvore de **nó raiz**. Este é tomado como uma referência e pode ter grau maior ou igual a um. Nós que possuem grau um são chamados de **nós terminais**, exceto se for o nó raiz. Uma **árvore geradora** (*spanning tree*) de um grafo G é qualquer subárvore de G que contenha todos os nós de G .

5.2 Representações de PPRs para AEs

Para escrever esta seção utilizou-se (de Lima e Delbem, 2007) como referência.

Nos capítulos anteriores foram apresentados os AEs como ferramenta poderosa de otimização inspiradas na teoria da evolução natural. No entanto, AEs com codificações convencionais tem sido ineficientes para Problemas de Projeto de Redes (PPRs)(de Lima e Delbem, 2007), em especial para grandes sistemas. Tais abordagens evolutivas geram muitos componentes desconexos ou grafos acíclicos quando aplicados a grandes sistemas, enquanto que em PPRs geralmente busca-se a produção de árvores (ou florestas) geradoras. Dependendo da codificação adotada, a produção de tais grafos pode consumir uma grande parte do tempo de execução, o qual reduz a eficiência dos AEs. Além disso, as redes produzidas podem ser muito diferentes de seus pais, reduzindo demasiadamente a convergência dos AEs.

Os PPRs envolvem problemas do mundo real das diversas áreas da engenharia e ciências, tais como circuitos elétricos, roteamento de veículos, redes de computadores e sistemas de distribuição de energia elétrica. Com o objetivo de melhorar a eficiência dos AEs para PPRs, novas codificações têm sido propostas as quais têm produzido aumento

significativo da eficiência dos AEs. Uma dessas configurações é a Representação Nó-Profundidade (RNP) (Delbem et al., 2004) que tem como vantagem a capacidade de trabalhar com redes correspondendo a florestas.

As estruturas de dados para AEs aplicadas a PPRs devem lidar com representações de árvores geradoras de grafos (Gross e Yellen, 2004), pois as soluções de PPRs, em geral, envolvem árvores geradoras de um grafo que representa o problema.

Ao longo dos anos diversas representações de PPRs para AEs. Na Tabela 5.2 são apresentadas as principais delas, dentre quais destaca-se a RNP, que têm sido aplicada para o problema de reconfiguração de SDR e apresentado ótimos resultados com uma redução considerável do tempo de processamento e na quantidade de memória RAM utilizada (dos Santos, 2008).

Destaca-se que no presente projeto foi utilizada a RNP para modelar um SDR (ver Capítulo 7).

Tabela 5.2: Principais representações de PPRs para AEs.

| Representações de PPRs | Referência |
|---|---|
| Vetor Características | (Sinclair, 1995) |
| Predecessores ou Codificação Determinante | (Abuali et al., 1995) |
| Número de Prüfer | (Prüfer, 1918) |
| <i>Blob Code</i> | (Picciotto, 1999) |
| Tendência de Ligação e Nó | (Palmer, 1994) |
| Chaves Aleatórias para Redes | (Rothlauf et al., 2002) |
| Conjunto de Arestas | (Raidl, 2000) |
| Representação Nó-Profundidade | (Delbem et al., 2004) |
| Precedentes Diretos | (Carvalho et al., 2001) |
| Permutação Baseada em Árvore | (Zhou e Gen, 2003) |
| <i>D-Based</i> | (Zhou et al., 2007) |
| Ajuste Adaptativo das Ligações | (SOAK et al., 2005) |
| Sub-Conjunto de Comprimento Fixo | (Julstrom, 1994) |
| <i>Dandelion Code</i> | (Raidl e Julstrom, 2003; Picciotto, 1999) |
| Representação Nó-Profundidade e Grau | (de Lima e Delbem, 2007) |

5.3 Representação Nó-Profundidade

A Representação Nó-Profundidade (RNP), proposta por (Delbem et al., 2004), baseia-se nos conceitos de nó e profundidade de nó em um grafo acíclico e conexo (árvore). Basicamente, a RNP é composta por uma lista linear contendo os nós da

árvore, e suas respectivas profundidades. Essa lista é formada por pares (n_x, p_x) , onde n_x representa o nó da árvore e p_x a profundidade do nó. A ordem em que os pares são dispostos na lista é importante.

Computacionalmente, esta lista é formada por uma matriz de dimensão $2 \times n$, onde n é o número de nós de uma determinada árvore. De tal forma que, cada par (n_x, p_x) é armazenado em uma determinada coluna da matriz, onde p_x e n_x são armazenados na primeira e na segunda linha respectivamente (Figura 5.2b). Para armazenar um nó e a sua respectiva profundidade na RNP, é utilizada um algoritmo de busca em profundidade (Cormen, 2002). Desta maneira, começando a busca a partir do nó raiz da árvore, é produzida uma lista de pares (n_x, p_x) em uma sequência apropriada enquanto um nó n_x é visitado.

Para o entendimento de como uma árvore é armazenada na RNP, será realizada uma análise da Figura 5.2b, que apresenta a RNP para a árvore geradora representada por linhas espessas no grafo da Figura 5.2a. Inicialmente é armazenado o nó raiz da árvore, no caso o nó 1, com profundidade igual a 0. Logo, realiza-se uma busca em profundidade na árvore, através dos ramos conectados ao nó raiz, para armazenar os demais nós juntamente com suas respectivas profundidades, as quais são sempre calculadas em relação ao nó raiz.

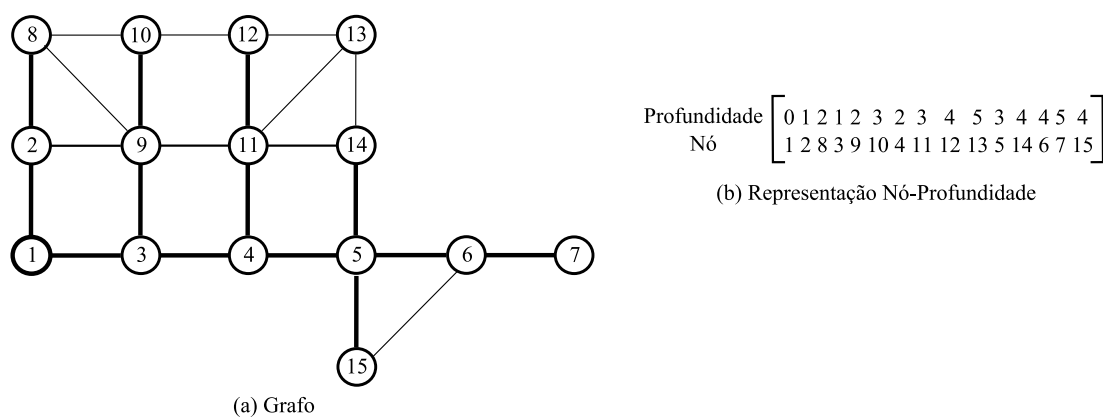


Figura 5.2: Exemplo de um grafo e sua RNP

A codificação para uma floresta é composta pela união das codificações de todas as árvores da mesma. Assim, a estrutura de dados da floresta pode ser facilmente implementada utilizando uma lista de ponteiros, onde cada ponteiro indica a RNP de uma árvore da floresta.

Para facilitar a manipulação da floresta armazenada em RNPs, criaram-se dois op-

eradores bastante similares, chamados de **operador 1** e **operador 2** (Delbem et al., 2004). Ambos os operadores transferem uma sub-árvore (parte podada) de uma árvore A_{de} (árvore origem) para uma árvore A_{para} (árvore destino). Entretanto, no operador 1 a raiz da sub-árvore podada será a raiz dessa subárvore na nova árvore A_{para} ; já no operador 2, um novo nó (diferente da raiz) é escolhido para ser a nova raiz da sub-árvore em A_{para} (Delbem et al., 2004).

O operador 1 requer a definição prévia de dois nós: o nó de poda p , que indica a raiz da sub-árvore que será podada; e o nó adjacente a , que é o nó da árvore A_{para} , onde a sub-árvore será inserida. Além desses dois nós, o operador 2 requer ainda o nó r , que será a nova raiz da sub-árvore que será transferida.

5.3.1 Operador 1

Para descrição do operador 1, considera-se que os nós p e a sejam previamente escolhidos. A RNP é implementada utilizando-se matrizes, sendo conhecidos os índices de $p(i_p)$ e $a(i_a)$ (Figura 5.3a) nas matrizes A_{de} e A_{para} , respectivamente.

O operador 1 pode ser descrito através dos seguintes passos (Figura 5.3):

1. Determinam-se as posições (i_p, i_l) dos índices na árvore A_{de} , correspondente à sub-árvore enraizada no nó p . Conhecido i_p , é necessário encontrar apenas i_l , que corresponde ao índice do último nó na sub-árvore que tem o nó p como raiz. O conjunto (i_p, i_l) corresponde ao nó p , em i_p , e consecutivos nós x na segunda linha da matriz A_{de} , tal que $i_x > i_p$ e $p_x > p_p$ (entre as linhas tracejadas na Figura 5.3a), p_x é a profundidade do nó x ;
2. Copiam-se os dados do conjunto (i_p, i_l) , da árvore A_{de} , em uma matriz temporária A_{tmp} (contendo os dados da sub-árvore que está sendo transferida); ver Figura 5.3b. A profundidade de cada nó x , do conjunto (i_p, i_l) , é atualizada utilizando a seguinte equação: $p_x = p_x - p_p + p_a + 1$, onde: p_x , p_p e p_a são as profundidades dos nós x , p e a , respectivamente;
3. Cria-se a matriz $A_{para'}$, contendo os nós A_{para} e inserindo depois a matriz A_{tmp} na posição $i_a + 1$ de A_{para} , isto é, gera-se uma nova árvore que conecta a sub-árvore na árvore $A_{para'}$ (Figura 5.3c);

4. Constrói-se uma matriz $A_{de'}$, que possua os nós A_{de} , sem os nós de A_{tmp} ;
5. Atualiza-se a floresta, fazendo com que a estrutura de dados que antes apontava para A_{de} e A_{para} , aponte agora para $A_{de'}$ e $A_{para'}$.

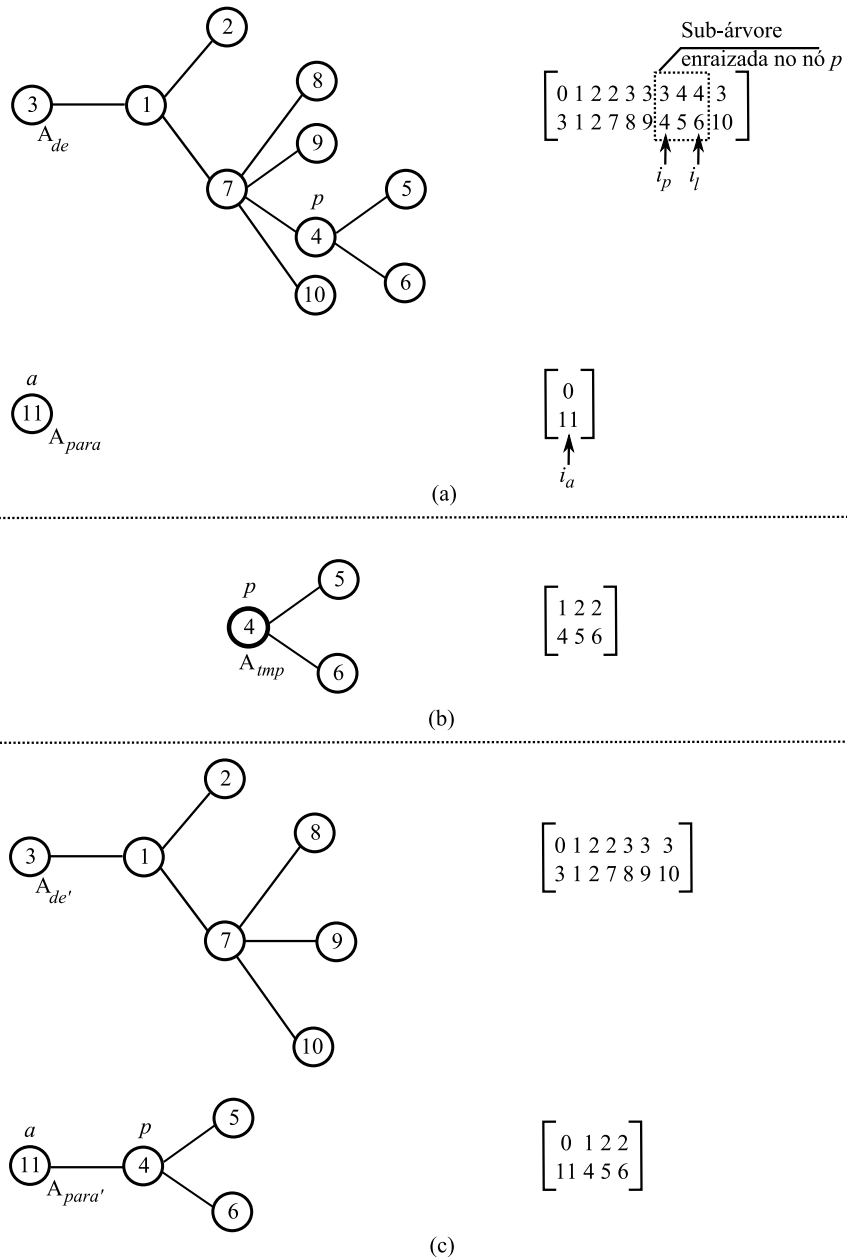


Figura 5.3: Ilustração dos passos do operador 1

5.3.2 Operador 2

Para a descrição do operador 2, considera-se que um conjunto de nós seja previamente determinado: o nó de poda p , o novo nó raiz r e o nó adjacente a . Os nós p e r

pertencem à árvore A_{de} , e o nó a à A_{para} ;

As diferenças do operador 1 para o 2 estão nos passos 2 e 3 do procedimento do operador 1 (ver Operador 1), isto é, a formação da sub-árvore cortada e como a mesma é armazenada em um vetor temporário tmp são diferentes. Os passos 2 e 3, para o operador 2, são descritos na sequência. As Figuras 5.4a, 5.4b, 5.4c e 5.4d ilustram um exemplo destes passos para as mesmas árvores A_{de} e A_{para} utilizadas na aplicação do operador 1, ilustrado na Figura 5.3.

O procedimento da cópia da sub-árvore, para o operador 2, pode ser dividida em dois passos: o primeiro é similar ao passo 2 do operador 1, com a diferença de que, no operador 2, troca-se o índice i_p por i_r

No segundo passo consideram-se os nós de r até p de A_{de} , isto é $r_0, r_1, r_2, \dots, r_n$, onde $r_0 = r$ e $r_n = p$, como raízes de sub-árvores (ver os nós destacados na Figura 5.4a). O algoritmo para o segundo passo deve copiar a sub-árvore enraizada em r_i ($i = 1, 2, \dots, n$), sem a sub-árvore enraizada em r_{i-1} (veja Figura 5.4b) e armazena o resultado das sub-árvores, na matriz temporária A_{tmp2} (veja Figura 5.4c). Em seguida o operador 2 cria a matriz $A_{para'}$, contendo os nós de A_{para} e inserindo depois a matriz A_{tmp2} na posição $i_a + 1$ de A_{para} . Ou seja, cria-se uma nova árvore que conecta a sub-árvore na árvore $A_{para'}$ (ver Figura 5.4d).

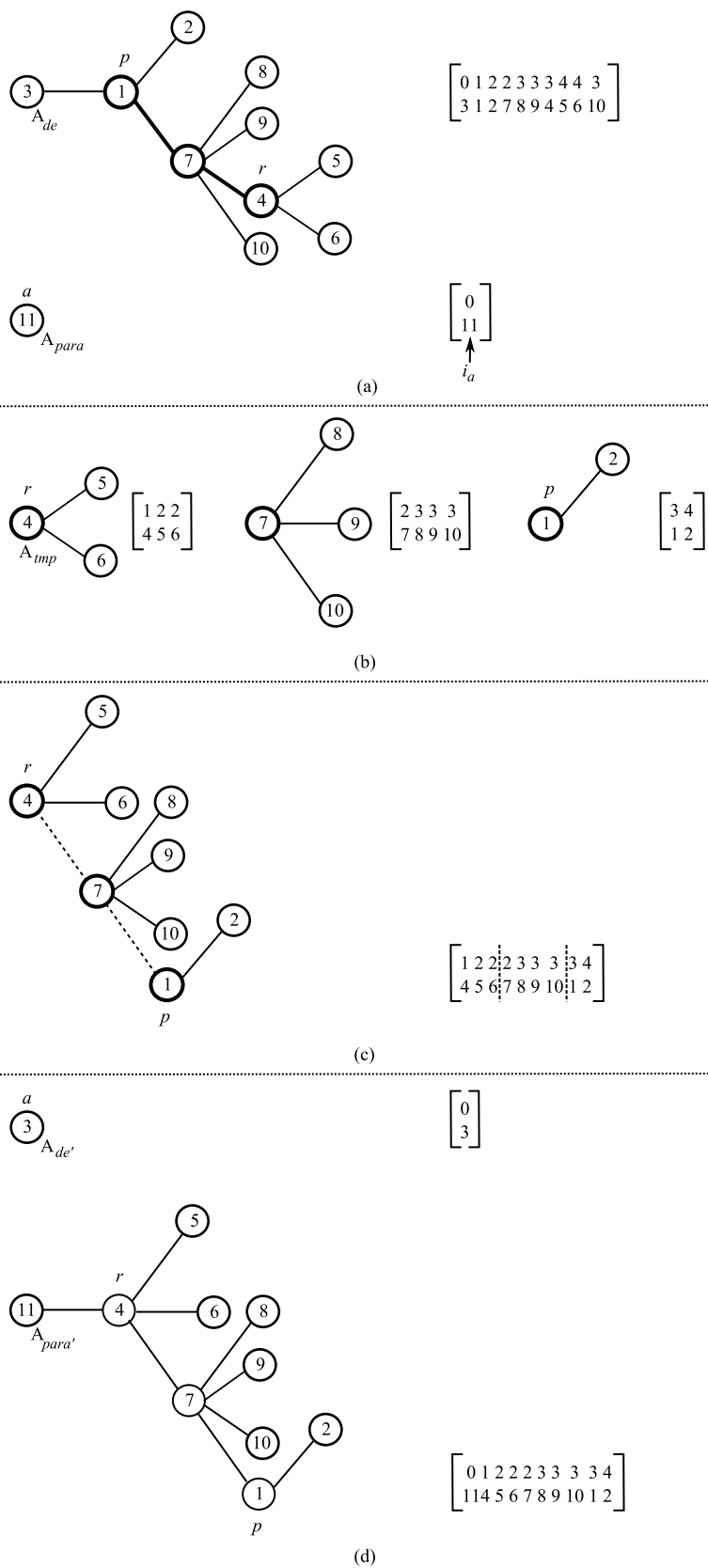


Figura 5.4: Ilustração dos passos do operador 2.

Capítulo 6

Fluxo de Carga

Este Capítulo introduz, de maneira sucinta, alguns dos principais métodos para cálculo de fluxo de carga usados em SDR. Na Seção 6.1 são apresentadas as considerações iniciais sobre o estudo de fluxo de carga. Na Seção 6.3 é descrito o método *Backward/Forward* de soma de correntes. Na Seção 6.3 é descrito o método *Backward/Forward* de soma de potências.

6.1 Considerações Iniciais

O estudo de fluxo de carga (ou fluxo de potência) em uma rede de energia elétrica consiste na obtenção das condições de operação da mesma (tensões complexas nas barras, fluxos de potência nas linhas e transformadores), em função da topologia da rede e dos seus níveis de demanda e geração de potência.

Métodos convencionais para o cálculo de fluxo de carga em redes de transmissão de energia elétrica, tais como o método de Newton-Raphson, Desacoplado Rápido e versões modificadas dos mesmos (Monticelli, 1983; Monticelli et al., 1990), podem apresentar um mal desempenho quando aplicados em redes de distribuição. Principalmente para redes radiais com grande número de barras. Isso ocorre devido às características particulares dos SDRs, dentre as quais podemos citar: baixa relação X/R (reatância/resistência) dos parâmetros dos alimentadores, trechos com impedâncias relativamente baixas¹ associa-

¹Representação de chaves seccionadoras, reguladores de tensão e trechos pequenos de linhas entre cargas muito próximas.

dos a trechos com impedâncias altas e grande número de barras de carga distribuídas. Em razão dessas características, as matrizes associadas aos SDRs são mal-condicionadas, dificultando o cálculo de fluxo de carga através dos métodos tradicionais supracitados (Das et al., 1994), que exigem a fatoração de matrizes, pois, afetam a convergência daqueles métodos exigindo um grande número de iterações, podendo causar, até mesmo, divergência do processo iterativo.

Face ao exposto, diversos métodos para o cálculo de fluxo de carga para SDRs foram propostos, os quais são divididos em duas categorias (Srinivas, 2000):

- Métodos *backward/forward*;
- Métodos baseados na matriz de impedância nodal implícita.

Os métodos *backward/forward* são bastante empregados em SDRs(ou fracamente malhados). A rede, nestes métodos, é representada como um grafo acíclico (árvore), cujo nó raiz corresponde à subestação (ver Figura 6.1).

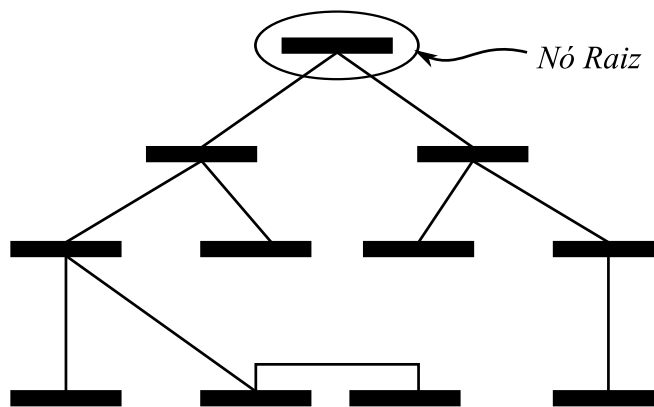


Figura 6.1: Exemplo de um SDR.

Esses métodos são também conhecidos como métodos de varredura direta/inversa, devido ao fato de apresentarem um processo iterativo que faz um percurso das barras extremas em direção à subestação e vice-versa. Nestes métodos, primeiramente realiza-se a etapa *Backward*, que partindo das barras extremas (nós folhas) e usando um *valor* inicial das tensões nodais, consiste em calcular as correntes ou fluxos de potência nas linhas até a subestação (nó raiz). Após esta etapa, realiza-se a etapa *Forward*, que a partir do resultado da injeção de corrente ou potência da subestação, e do valor conhecido da tensão nessa barra, são calculados novamente os valores de tensão das

barras da rede, até as barras extremas. Tal procedimento é repetido até que os valores de tensão de duas iterações consecutivas não variem mais que uma determinada tolerância, bem próxima a zero (*mistake*). Este método, a princípio, tem duas versões, sendo estas:

- Soma de Correntes (Shirmohammadi et al., 1988);
- Soma de Potências (Baran e Wu, 1989; Cespedes, 1990).

Os métodos baseados na matriz de impedância nodal implícita baseiam-se na formação e fatoração da matriz de admitância nodal e injeções de corrente equivalentes. Nesses métodos, o efeito da fonte e das cargas é representado separadamente por superposição (Srinivas, 2000; Chen et al., 1991).

6.2 Método *Backward/Forward* de Soma de Correntes

O método de soma de correntes (Shirmohammadi et al., 1988) foi desenvolvido inicialmente para SDRs, podendo ser aplicado a sistemas de distribuição fracamente malhados. Tal método é conceitualmente simples e apresenta desempenho eficiente. O procedimento para o cálculo de fluxo de carga desse método é composto pelos seguintes passos:

1. Inicialmente deve-se especificar a tensão do nó raiz e assumir, como estimativa inicial, tensão igual a 1 (um) p.u. com ângulo de 0 (zero) graus para todas as demais barras do SDR;
2. **Cálculo da corrente nodal:** na iteração k , a injeção de corrente nodal $\dot{I}_i^{(k)}$ é calculada segundo a expressão;

$$\dot{I}_i^{(k)} = (\dot{S}_i / \dot{V}_i^{(k-1)})^* - \dot{Y}_i^{sh} \dot{V}_i^{k-1}, i = 1, 2, \dots, n, \quad (6.1)$$

onde $\dot{V}_i^{(k-1)}$ é a tensão na barra i , calculada durante $(k - 1)$ -ésima iteração; \dot{S}_i é a injeção de potência complexa especificada na barra i ; \dot{Y}_i^{sh} é a soma de todos os elementos *shunt* da barra i ; e n é o número total de barras da representação radial do sistema. O símbolo $(.)^*$ indica o conjugado do valor complexo entre parênteses;

3. **Backward:** Na iteração k , a partir das linhas conectadas às barras extremas do grafo (barras com maiores profundidades) e movendo-se até as linhas conectadas à barra raiz (com profundidade zero), calcula-se a corrente (\dot{F}_L) na linha L , que liga uma barra $L2$ à sua barra antecessora $L1$, conforme ilustrado na Figura 6.2, da seguinte forma:

$$\dot{F}_L^{(k)} = -\dot{I}_{L2}^{(k)} + \sum (\text{Corrente nas linhas que saem do nó } L2), \quad (6.2)$$

onde $L = p, p - 1, \dots, 1$, $\dot{I}_{L2}^{(k)}$ é a injeção de corrente no nó $L2$ e p é o número de linhas que o sistema possui;

4. **Forward:** as tensões complexas das barras são atualizadas, iniciando pelas barras que estão conectadas à barra raiz (subestação) e seguindo até as barras extremas. Seja L a linha que liga uma barra $L2$ à sua barra antecessora $L1$, a tensão de $L2$ é calculada usando a atualização da tensão na iteração k de $L1$ e o fluxo de corrente na linha calculado no passo 3:

$$\dot{V}_{L2}^{(k)} = \dot{V}_{L1}^{(k)} - \dot{Z}_L \dot{F}_L^{(k)}, L = 1, 2, \dots, p, \quad (6.3)$$

onde \dot{Z}_L é a impedância série da linha L ;

5. Os passos 2, 3 e 4 são repetidos até que seja alcançado o critério de convergência.

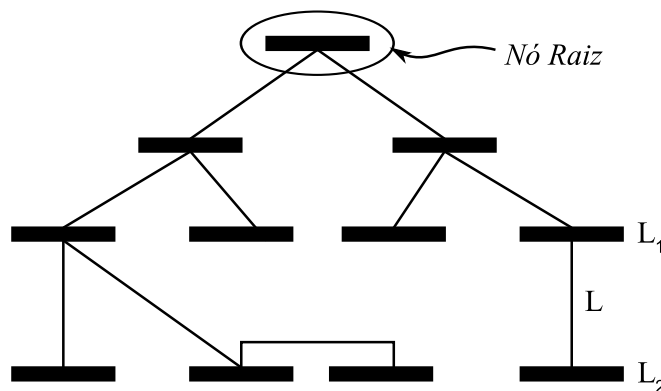


Figura 6.2: Sistema de distribuição radial.

Adota-se como **critério de convergência** o maior erro de potência ativa e reativa nas barras do sistema, tal que este erro seja menor que um ϵ . Conforme descrito nos passos acima, em cada iteração é calculada a injeção de corrente e, posteriormente, as

tensões das barras do sistema. Assim, a potência complexa injetada na barra i (ou a potência complexa líquida na barra i) na iteração k , $\dot{S}_i^{(k)}$, é calculada utilizando a expressão 6.4.

$$\dot{S}_i^{(k)} = \dot{V}_i^{(k)} \dot{I}_i^{(k)*} - \dot{Y}_i |\dot{V}_i^{(k)}|^2, i = 1, 2, \dots, n. \quad (6.4)$$

O erro de potência ativa e reativa na barra i é calculado da seguinte forma:

$$\begin{aligned} \Delta P_i^{(k)} &= \text{Re}[\dot{S}_i^{(k)} - \dot{S}_i] \\ \Delta Q_i^{(k)} &= \text{Im}[\dot{S}_i^{(k)} - \dot{S}_i]. \end{aligned} \quad i = 1, 2, \dots, n \quad (6.5)$$

6.3 Método *Backward/Forward* da Soma de Potências

O método da soma de potências (Baran e Wu, 1989; Cespedes, 1990) é o método mais difundido na literatura. Tal método é relativamente simples do ponto de vista conceitual e apresenta um desempenho eficiente na resolução de problemas de fluxo de carga radial (Brandini, 2000).

Na etapa *Backward* são utilizadas as equações de fluxo de carga 6.6 e 6.7:

$$P_{i-1} = P_i + r_i \frac{P_i'^2 + Q_i'^2}{V_i^2} + P_{Li} \quad (6.6)$$

$$Q_{i-1} = Q_i + x_i \frac{P_i'^2 + Q_i'^2}{V_i^2} + Q_{Li}, \quad (6.7)$$

onde:

- P_i é o fluxo de potência ativa no ramo i ;
- Q_i é o fluxo de potência reativa no ramo i ;
- P_{Li} é a injeção de potência ativa líquida na barra i ;
- Q_{Li} é a injeção de potência reativa líquida na barra i ;

- $P'_i = P_i + P_{Li}$;
- $Q'_i = Q_i + Q_{Li}$.

Na etapa *Forward*, as Equações 6.8 e 6.9 são utilizadas para atualizar as tensões nas barras.

$$V_{i+1}^2 = V_i^2 - (r_i P_i + x_i Q_i) + (r_i^2 + x_i^2) \frac{P_i^2 + Q_i^2}{V_i^2}, \quad (6.8)$$

$$\delta_{i+1} = \delta_i - \tan^{-1}\left(\frac{k_1}{k_2}\right), \quad (6.9)$$

onde,

- V_i é a tensão na barra i ;
- δ_i é o ângulo na barra i ;
- r_i é a resistência em série na linha que conecta a barra i ;
- x_i é a reatância em série na linha que conecta i ;
- $k_1 = \frac{P_i x_i - Q_i r_i}{V_i}$;
- $k_2 = V_i - \frac{P_i x_i - Q_i r_i}{V_i}$.

O cálculo do fluxo de carga através deste método é composto pelos seguintes passos:

1. Assumir que as tensões iniciais em todas as barras são iguais à tensão da subestação (nó raiz);
2. **Backward:** calcular os fluxos de potência ativa e reativa para cada linha usando as equações 6.6 e 6.7;
3. **Forward:** calcular a tensão e o ângulo de cada barra utilizando as equações 6.8 e 6.9;
4. Como **critério de convergência**, verificar a variação da tensão e do ângulo na iteração atual com a iteração anterior. Se a diferença for maior ou igual a uma tolerância próxima de zero, repita o processo a partir do item 2; caso contrário, encerram-se os cálculos.

Capítulo 7

Algoritmo Evolutivo para Reconfiguração de SDR

Neste Capítulo é apresentado, de forma sucinta, o trabalho desenvolvido em (Santos, Delbem e Bretas, 2008b), o qual é utilizado como base para o desenvolvimento do presente trabalho. Para escrever este capítulo utilizou-se como referência o texto de qualificação de doutorado de Augusto Cesar dos Santos (dos Santos, 2008). Na Seção 7.1 é descrito o problema de restabelecimento de energia de SDR. Na Seção 7.2 é apresentada a formulação do problema de restabelecimento de energia de SDR. Na Seção 7.3 é descrita a forma como são avaliadas as soluções geradas pelo AE para reconfiguração de SDR. Na Seção 7.4 é apresentado o AE para reconfiguração de SDR.

7.1 Restabelecimento: um problema especial de reconfiguração de SDR

Um SDR pode ter a sua topologia representada por grafos (ver Capítulo 5). A Figura 7.1 mostra um SDR com 3 alimentadores. Cada nó do sistema representa um setor e as arestas interligando as barras são chaves seccionadoras. As arestas em linha cheia representam chaves NF (Normalmente Fechadas), enquanto as arestas em linha pontilhada representam as chaves NA (Normalmente Abertas). As barras 1, 2 e 3 encontram-se em uma subestação.

Na ocorrência de uma falta no sistema, o setor em falta deve ser isolado do SDR

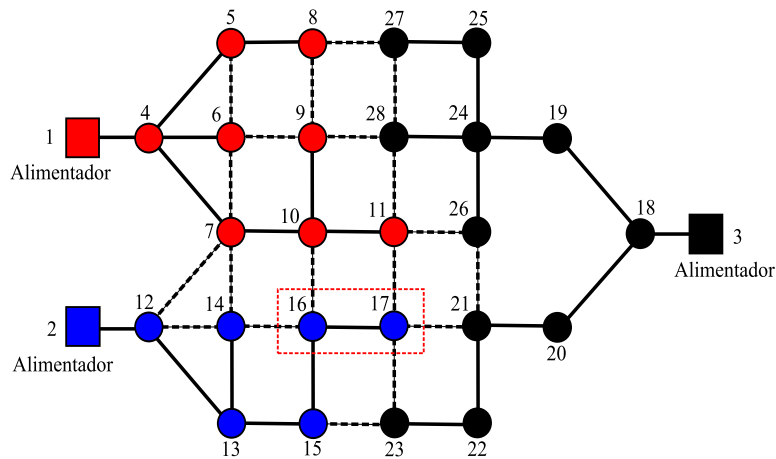


Figura 7.1: SDR com 3 alimentadores.

abrindo-se todas as chaves que o conecta ao restante do sistema. Conseqüentemente, as áreas a justante do setor em falta também ficarão sem energia. Torna-se, então, necessário, reconectar as áreas desenergizadas aos setores supridos de eletricidade, por meio do fechamento de chaves (Santos, Delbem e Bretas, 2008b).

A Figura 7.2 mostra um exemplo de reconfiguração de SDR para restabelecimento de energia. Considerando que o setor 15 em falta., o mesmo deve ser isolado do restante do SDR. Esse procedimento pode ser realizado abrindo-se as chaves A e B. Após a abertura dessas chaves, os setores 16 e 17 estarão em uma área fora de serviço, representados na Figura 7.3. Neste caso, o objetivo da reconfiguração é encontrar outro caminho que forneça energia a esses setores sem violar as restrições operacionais do sistema. Para isso, neste exemplo, existem os seguintes caminhos:

1. Fecha-se a chave C, conectando a área fora de serviço ao alimentador 2;
2. Fecha-se a chave D, conectando a área fora de serviço ao alimentador 1;
3. Fecha-se a chave E, conectando a área fora de serviço ao alimentador 3;
4. Fecha-se a chave F, conectando a área fora de serviço ao alimentador 3;
5. Fecha-se a chave G, conectando a área fora de serviço ao alimentador 1.

A Figura 7.4 mostra a nova configuração fechando a chave F. Após encontrar uma nova configuração para o SDR, restabelecendo a energia neste, é necessário que as seguintes restrições tenham sido satisfeitas (Inagaki et al., 2006):

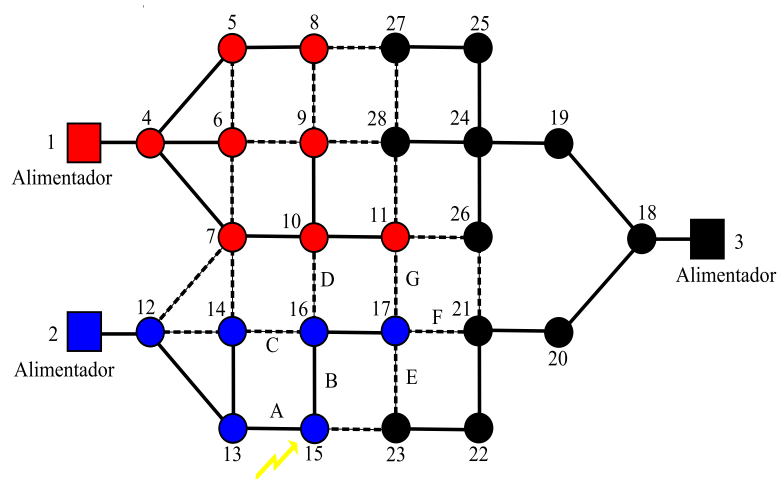


Figura 7.2: SDR em falta no setor 15.

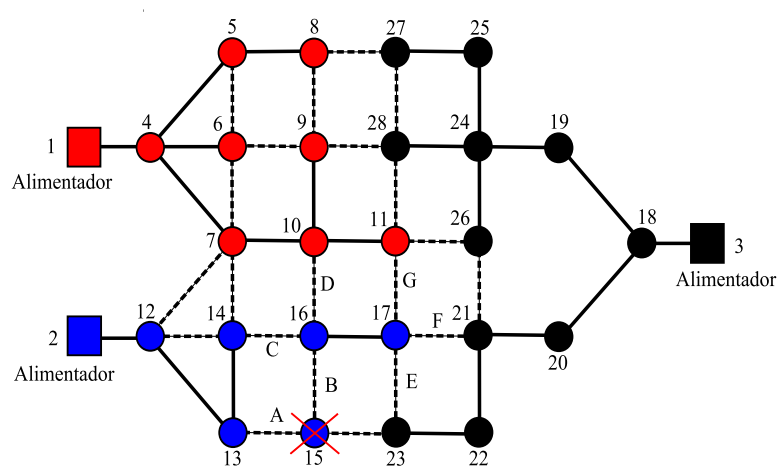


Figura 7.3: Setores a jusante do setor em falta desconectados do SDR.

1. Manter a estrutura radial após o serviço de restabelecimento;
2. Atender, quando possível, todas as áreas a jusante do setor em falta, isto é, os setores que ficaram fora de serviço;
3. A capacidade limite do transformador não deve ser excedida pelo montante de carga de cada alimentador do sistema;
4. A capacidade das linhas e chaves não deve ser ultrapassada pela corrente elétrica em cada ramo;
5. A queda de tensão em qualquer barra do SDR não deve exceder o limite permitido.

Além disso, o problema de reconfiguração em SDR, neste trabalho, envolve a minimização de dois objetivos:

1. Minimizar o número de manobras de operação de chaveamento;
2. Minimizar o total de perdas resistivas.

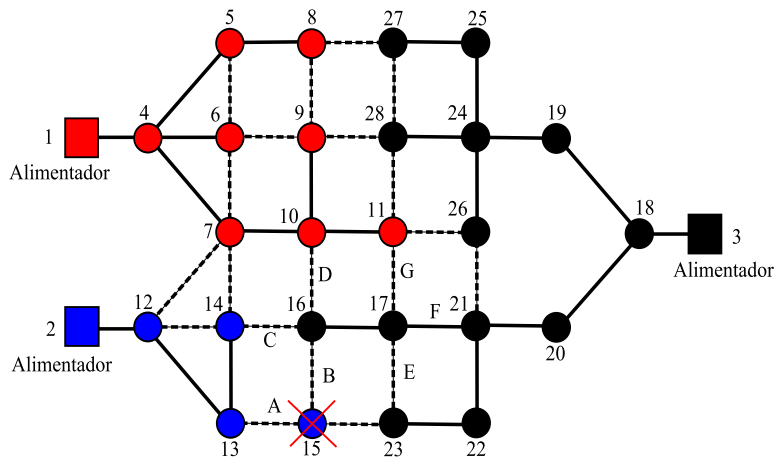


Figura 7.4: Nova configuração.

7.2 Formulação Matemática

Para obter a formulação matemática para problemas de reconfiguração de SDR, consideram-se os seguintes objetivos: minimização das áreas fora de serviço, do número

de operações de chaveamentos e do total de perdas resistivas, sem violar as restrições de carga e tensão. Também devem ser levados em consideração as 5 restrições operacionais apresentadas na seção anterior.

Assim, o problema geral de reconfiguração de SDR pode ser formulado como segue (Delbem et al., 2005):

$$\begin{aligned}
 \text{Min.} \quad & E(F) \\
 \text{s.a. :} \quad & H(F) = 0 \\
 & I(F) \leq 0 \\
 & F \text{ é uma floresta,}
 \end{aligned} \tag{7.1}$$

onde:

- F - grafo correspondente a uma configuração do sistema, onde cada árvore corresponde a um alimentador ligado a uma subestação;
- $E(F)$ - função objetivo;
- $H(F)$ - restrições de igualdade representando as equações de fluxo de carga;
- $I(F)$ - restrições de desigualdade representando as equações operacionais do sistema.

A função $E(F)$ contém, em geral, os seguintes componentes:

- $\phi(F)$ - quantidade de cargas fora de serviço para uma topologia radial da rede (uma floresta F);
- $\varphi(F)$ - perdas resistivas no sistema para F ;
- $\psi(F, F^0)$ - número de operações de chaveamento para obter uma dada configuração F , a partir da configuração original F^0 .

As restrições de igualdade correspondem às equações de fluxo de carga. Um sistema linear do tipo $Ax = b$ pode representá-las, onde:

- A - matriz de incidência de F ;

- x - vetor de corrente de linha;
- b - vetor com as injeções de correntes (de carga) nas barras ($b_i \leq 0$), ou injeções de correntes nas subestações ($b_i > 0$).

As restrições operacionais de $I(F)$ para problemas de reconfiguração de SDR geralmente incluem:

- Um limitante superior de corrente \bar{x}_j para cada corrente de linha x_j . A maior taxa x_j/\bar{x}_j é denominada carregamento da rede;
- A máxima injeção de corrente \bar{b}_i possível para cada subestação i , onde a maior taxa b_i/\bar{b}_i é denominada carregamento da subestação;
- Um limitante inferior para a tensão no nó \underline{v} . Seja v_i a tensão na barra i e v_b a tensão base no sistema; a maior taxa v_i/v_b é denominada maior taxa de tensão. O vetor de tensão v é dado por $Yv = b$, onde Y é a matriz de admitância nodal ($Y = AY_xA^T$, com Y_x sendo a matriz de admitância diagonal).

É comum a utilização do modelo de corrente constante e ordenação das barras segundo o modelo pai-filho (MPF) (Delbem et al., 2005), em problemas de reconfiguração de SDR. Assim, através do fluxo de carga, calculam-se os fluxos de corrente das barras partindo-se dos nós terminais (nós folhas) em direção à subestação (nó raiz); enquanto as tensões podem ser obtidas de forma encadeada partindo da subestação até as barras terminais.

A função objetivo para problemas envolvendo reconfiguração de SDR geralmente é não linear, descontínua e com vários ótimos locais, dificultando a utilização de Programação Matemática. Quando os AE são empregados para a resolução desse tipo de problema, algumas modificações são realizadas na formulação apresentada na expressão 7.1. São inseridos fatores de penalidades (Goldberg, 1989) a fim de penalizar as configurações da rede que violarem as restrições operacionais $I(F)$. Assim, o problema pode ser reformulado como segue:

$$\begin{aligned}
 \text{Min.} \quad & E(F) + |\Omega I(F)| \\
 \text{s.a. :} \quad & H(F) = 0 \\
 & F \text{ é uma floresta,}
 \end{aligned} \tag{7.2}$$

onde Ω é uma matriz diagonal com os seguintes elementos:

$$w_{11} = \begin{cases} w_x, & \text{se, pelo menos para um } j, x_j > \bar{x}_j \\ 0, & \text{caso contrário;} \end{cases}$$

$$w_{22} = \begin{cases} w_s, & \text{se, pelo menos para um } i, b_i > \bar{b}_i \\ 0, & \text{caso contrário;} \end{cases}$$

$$w_{33} = \begin{cases} w_v, & \text{se, pelo menos para um } i, v_i < \underline{v} \\ 0, & \text{caso contrário.} \end{cases}$$

Os pesos w_x , w_s e w_v são valores positivos e, $|\cdot|$ é a norma infinita usual (Gradshteyn e Ryzhik, 2000), isto é, a norma L_1 de um vetor z de tamanho n é dada por $\sum_{r=1}^n |Z_r|$.

A formulação do problema anterior pode ser simplificada através da utilização a RNP e de seus operadores. Estes operadores realizam modificações em uma floresta para a produção de novas florestas (configurações de SDR), que correspondam a configurações radiais factíveis. Dessa forma, utilizando a RNP, o problema descrito na Equação 7.2 pode ser reescrito como segue:

$$\begin{aligned} \text{Min.} \quad & E(F) + |\Omega I(F)| \\ \text{s.a. :} \quad & H(F) = 0 \end{aligned} \tag{7.3}$$

F é dado pelos operadores da RNP.

A RNP de um SDR possui, naturalmente, as barras de cada árvore (alimentador) ordenadas segundo o MPF. Com isso, evita-se a utilização de um algoritmo de busca (Cormen, 2002) para obter tal modelo. Assim, o fluxo de carga pelo MPF com RNP é mais eficiente que fluxos de carga convencionais para SDR (Santos, Nanni, Mansour, Delbem, London e Bretas, 2008). Além disso, o uso de MPF garante que as restrições de igualdade ($H(F)$) na Equação 7.3 sejam satisfeitas. Assim, fazendo uso da propriedade da RNP de permitir o armazenamento dos nós de acordo com o MPF, pode-se escrever

o problema de restabelecimento de SDR da seguinte forma:

$$\begin{aligned}
 & \text{Min.} \quad E(F) + |\Omega I(F)| \\
 & \text{s.a. :} \quad \text{Utilizar MPF com RNP} \\
 & \quad \quad \quad F \text{ é dado pelos operadores da RNP.}
 \end{aligned} \tag{7.4}$$

A utilização da RNP e seus operadores juntamente com o fluxo de carga pelo MPF tornam a modelagem matemática do problema mais simples, isso é facilmente percebido comparando a Equação 7.4 com a Equação 7.1. Além disso, com esta formulação são geradas exclusivamente configurações factíveis. Assim, somente as restrições de queda de tensão, carregamento na rede e nos transformadores são consideradas na formulação matemática do problema.

7.3 Avaliação das soluções

Esta seção apresenta a forma como se realiza a avaliação das soluções no trabalho proposto em (dos Santos, 2008). Basicamente, esta avaliação é realizada através do cálculo do fluxo de carga (ver Capítulo 6), com exceção do número de operações de chaveamento. O fluxo de carga utilizado é similar aos apresentados anteriormente, porém sofre algumas alterações devido ao uso da RNP.

7.3.1 Extensão da RNP para fluxo de carga

Geralmente, em SDR reais nem todos os trechos entre as barras são separados por chaves seccionadoras, logo, denomina-se setor o conjunto de barras e linhas não separadas por chaves seccionadoras. Vale ressaltar que em muitos trabalhos, as barras de carga de um setor são modeladas como se estivessem concentradas em um único ponto. Esse procedimento reduz o grau de confiabilidade do sistema.

Procurando reproduzir um SDR real, com a maior fidelidade possível, em (dos Santos, 2008) utilizou-se a RNP em dois níveis diferentes: a RNP do alimentador e a RNP do setor. Considere os 2 alimentadores da Figura 7.5. As barras em azul pertencem ao alimentador 1 e as barras em preto ao alimentador 2. Nas Figuras 7.5 e 7.6, os retângulos são barras em subestações, círculos são barras do SDR (barras de carga,

extremidade de chaves, ponto de conexão de duas ou mais linhas), linhas pontilhadas são chaves seccionadoras NF, linhas cheias são linhas do SDR e linhas interrompidas são chaves seccionadoras NA.

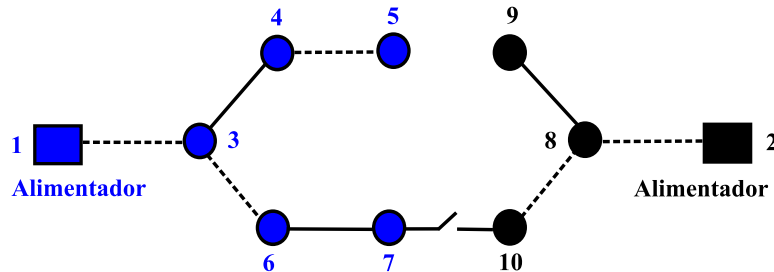


Figura 7.5: SDR com dois alimentadores.

A Figura 7.6 mostra o agrupamento das barras e linhas não separadas por chaves na Figura 7.5, desta forma, tem-se um grafo em que todas as arestas são chaves seccionadoras, conforme a Figura 7.7.

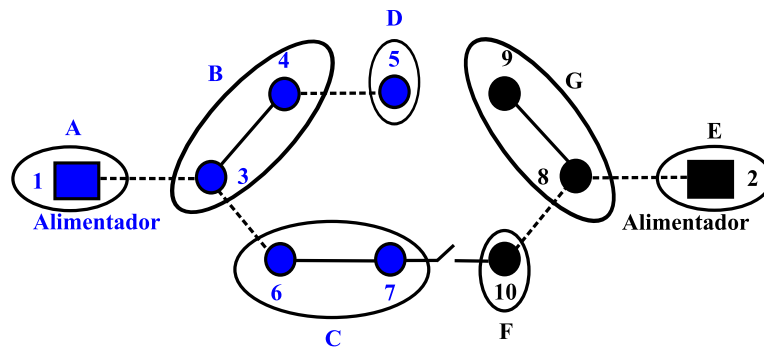


Figura 7.6: Agrupamento das linhas e barras em setores.

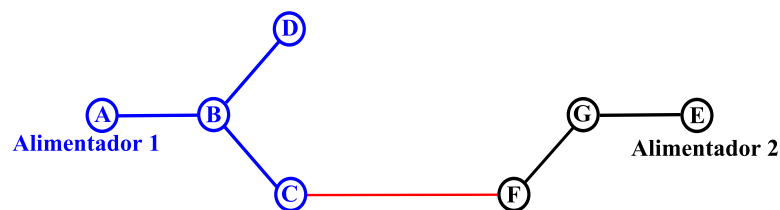


Figura 7.7: Grafo representando setores do SDR da Figura 7.6.

A Figura 7.7 possui 2 RNPs, uma para o alimentador 1 (cor azul) e outra para o alimentador 2 (cor preta), denominadas RNP do alimentador. As chaves seccionadoras NF são as arestas em azul e preto e a chave aberta é a aresta em vermelho. Assim, temos uma estrutura NP_1 que armazena o endereço de memória da RNP do alimentador 1, e a estrutura NP_2 armazena o endereço de memória da RNP do alimentador 2:

$$NP_1 = \begin{bmatrix} 0 & 1 & 2 & 2 \\ A & B & D & C \end{bmatrix},$$

$$NP_2 = \begin{bmatrix} 0 & 1 & 2 \\ E & G & F \end{bmatrix}.$$

A partir da Figura 7.6 cada trecho de linhas e barras não separadas por chaves pode ser analisado como uma árvore de grafo, isto é, é possível também associar RNP's aos setores. Cada setor pode ter mais de um nó raiz, dependendo do sentido em que está sendo alimentado. Assim, pode haver mais de uma árvore representando cada setor. Para finalidade de fluxo de carga, acrescenta-se a cada uma dessas árvores o nó adjacente ao seu nó raiz. A figura 7.8 mostra o setor C da figura 7.6 com os seus respectivos nós adjacentes (cor azul).

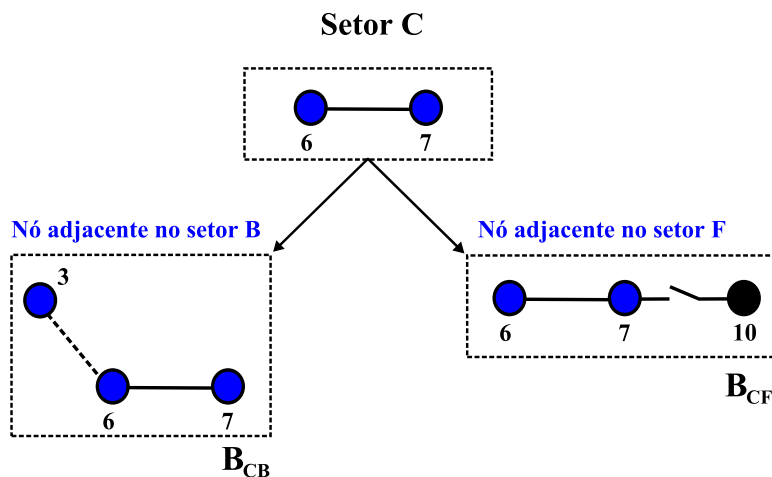


Figura 7.8: Árvore do setor C, com o nó adicional.

A RNP do setor pode ser representada de forma semelhante à RNP do alimentador, onde as árvores foram armazenadas em estruturas denotadas por NP_i . Para a RNP do setor, denota-se B_{sr} , onde s representa o setor em análise e r o setor pelo qual a energia chega ao setor s . Conforme pode ser visto na Figura 7.8, para o mesmo setor s pode existir mais de um setor r .

Para a Figura 7.8, o fluxo de corrente pode chegar ao setor C por dois caminhos diferentes, através do setor B ou do setor F . Assim, têm-se $B_{sr} = B_{CB}$ para o setor B e $B_{sr} = B_{CF}$ para o setor F . As RNPs possíveis para o setor C são:

$$B_{CB} = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 6 & 7 \end{bmatrix},$$

$$B_{CF} = \begin{bmatrix} 0 & 1 & 2 \\ 10 & 7 & 6 \end{bmatrix}.$$

Para descobrir qual das configurações acima deve ser utilizada é necessário realizar uma análise da RNP do alimentador. Deste modo, analisando o alimentador 1, descobre-se que o setor C está conectado ao setor B , conforme abaixo:

$$NP_1 = \begin{bmatrix} 0 & \mathbf{1} & 2 & \mathbf{2} \\ \mathbf{A} & \mathbf{B} & D & \mathbf{C} \end{bmatrix}.$$

Portanto, no exemplo da Figura 7.8, o setor B é o r correto.

Importa destacar que a determinação de todas as RNPs de cada setor pode ser executada por um procedimento *off-line*, deixando todas as estruturas prontas para serem utilizadas pelo fluxo de carga *on-line*.

7.3.2 Método *Backward/Forward* com RNP

O método de restabelecimento de SDR proposto em (dos Santos, 2008) utiliza o equivalente monofásico e o modelo de corrente constante para o cálculo de fluxo de carga. Isto em razão das características específicas dos SDR e da necessidade de encontrar configurações em um curto tempo de processamento.

Método Computacional

O algoritmo de fluxo de carga proposto em (dos Santos, 2008) é composto por duas subrotinas:

- Subrotina **CORRENTES**: obtém as correntes a jusante por meio do processo *backward* para todas as barras de um alimentador;
- Subrotina **TENSÕES**: utiliza as correntes a jusante para obter as tensões nas barras do mesmo alimentador, por meio do processo *forward*.

Em razão de estar sendo utilizado o modelo de corrente constante, a convergência é atingida em um único ciclo.

No Algoritmo 5 é descrita a subrotina **CORRENTES**, observe que a ordem dos nós visitados é no sentido dos nós terminais para a raiz, ordem que está pre-determinada nas RNPs do alimentador e do setor. A carga de uma barra m é denominada $I(m)$ e a sua corrente jusante é denominada $J(m)$.

Algoritmo 5 *Subrotina* **CORRENTES**.

```

1: for  $k \leftarrow |NP|, |NP| - 1, \dots, 1$  do
2:    $u \leftarrow NP.prof(k) - 1$ 
3:    $s \leftarrow NP.no(k)$ 
4:    $r \leftarrow NP.no(u)$ 
5:   //A corrente jusante em todas as barras em NP recebem zero
6:    $J(NP) \leftarrow 0$ 
7:   while  $q \leftarrow |B_{sr}|, |B_{sr}| - 1, \dots, 1$  do
8:      $p \leftarrow B_{sr}.prof(q) - 1$ 
9:      $m \leftarrow B_{sr}.no(p)$ 
10:     $n \leftarrow B_{sr}.no(q)$ 
11:     $J(m) \leftarrow J(m) + J(n) + I(n)$ 
12:  end while
13: end for

```

Algoritmo 6 *Subrotina* **TENSÕES**.

```

1: for  $k \leftarrow 1, 2, \dots, |NP|$  do
2:    $u \leftarrow NP.prof(k) - 1$ 
3:    $s \leftarrow NP.no(k)$ 
4:    $r \leftarrow NP.no(u)$ 
5:   //A tensão nas barras da subestação geralmente é 13.8 kV
6:    $B_{sr} \leftarrow V_{sub}$ 
7:   while  $q \leftarrow 1, 2, \dots, |B_{sr}|$  do
8:      $p \leftarrow B_{sr}.prof(q) - 1$ 
9:      $m \leftarrow B_{sr}.no(p)$ 
10:     $n \leftarrow B_{sr}.no(q)$ 
11:    Calcular a queda de tensão na rede entre as barras m e n
12:     $\Delta V \leftarrow Z_{mn} * (J(n) + I(n))$ 
13:     $V(n) \leftarrow V(m) - \Delta V$ 
14:  end while
15: end for

```

No Algoritmo 6 é descrita a subrotina **TENSÕES**, observe agora que a ordem dos nós visitados é no sentido do nó raiz para os nós terminais. Essa ordem já está

determinada nas RNPs do alimentador e do setor. Os dados necessários para obter as tensões em cada barra são a corrente na barra $I(n)$, a tensão a montante da barra n , $V(m)$, e a impedância Z_{mn} entre as barras m e n .

Os dados retornados pelas subrotinas apresentadas acima são utilizados pelo algoritmo principal do fluxo de carga utilizando a RNP, este descrito no Algoritmo 7.

Algoritmo 7 Fluxo de carga.

```

1: //Para cada estrutura  $NP_i$  de  $F$ 
2: for  $i \leftarrow 1, 2, \dots, |F|$  do
3:   CORRENTES( $F \rightarrow NP_i$ )
4:   TENSOES( $F \rightarrow NP_i$ )
5: end for

```

7.3.3 Cálculo do número de manobras

Cada configuração gerada pela aplicação dos operadores da RNP é avaliada por um fluxo de carga e pelo número de manobras necessárias para implementar essa nova configuração, determinando assim a sua adequação para o problema de reconfiguração de SDR. Em geral, determina-se o número de manobras a partir de comparações entre vetores binários que armazenam o estado das chaves (1 - aberta; 0 - fechada) de cada configuração. O custo computacional deste procedimento é relativamente alto, pois é criado um vetor de estado atual das chaves para cada nova configuração gerada e realiza-se uma comparação deste com o vetor da configuração inicial. O tamanho de cada vetor é equivalente ao número de chaves (m) no SDR. Assim, o tempo para percorrer cada vetor é bem maior que o tempo de realizar uma modificação no SDR pelos operadores genéticos da RNP, que requerem tempo computacional da ordem do tamanho do alimentador do SDR, em geral, um número bem menor que m .

Em (dos Santos, 2008) foi proposto um algoritmo que determina o número de manobras de forma mais eficiente, melhorando, deste modo, o desempenho computacional. Para esta formulação são utilizados dois vetores: um com o estado das chaves na configuração inicial e outro, de tamanho G_{max} (número máximo de gerações), que armazena a quantidade de chaves alteradas em relação à configuração inicial.

Na ocorrência de uma falta, o procedimento de isolar o setor em falta e conectar os setores a jusante do mesmo ao SDR exige manobras de chaves que nem sempre ocorrerão

aos pares. Para ilustrar esse procedimento, o SDR da Figura 7.9 têm o setor 15 em falta. As chaves A e B são abertas para isolar tal setor, com isso são realizadas 2 manobras de chaves. Com este procedimento, os setores 16 e 17 ficaram desconectados do SDR. Para reconectá-los novamente, é necessário fechar uma chave (dentre as chaves C, D, E, F ou G) que os conectam a algum alimentador, isto custará 1 manobra de chave. Deste modo, na primeira alteração de topologia foram necessárias 3 manobras de chaves.

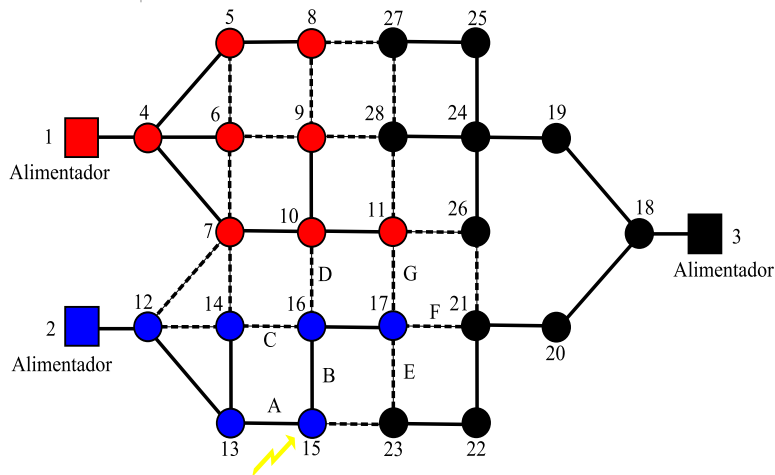


Figura 7.9: Operações de manobras necessárias para isolar o setor em falta.

Após o procedimento descrito acima, as chaves alteradas ocorrerão em pares, isto é, quando uma chave é aberta, outra deve ser fechada. O estado das chaves é dividido em 3 configurações, assim é possível determinar o número de manobras para originar uma determinada configuração:

1. Configuração inicial (o)¹;
2. Configuração alterada (x);
3. Configuração final (y).

Considerando que uma configuração y originou-se de alterações em uma configuração x , têm-se 3 possibilidade para calcular o número de chaves alteradas da configuração y ;

1. Os estados das duas chaves alteradas em y , em relação a x , são diferentes dos estados dessas chaves em o . Deste modo, o número de chaves alteradas de y será o número de chaves alteradas de x **mais 2**;

¹Na configuração inicial o setor em falta se encontra isolado e a energia restabelecida ao setores a jusante.

Tabela 7.1: Manobras de chaves - caso 1.

| Chaves | Configurações | | |
|--------|---------------|-----|----------|
| | o | x | y |
| 1 | 1 | 1 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 0 | 0 | 1 |
| 5 | 1 | 1 | 1 |

2. Os estados das duas chaves alteradas em y , em relação a x , são iguais aos estados dessas chaves em o . Deste modo, o número de chaves alteradas de y será o número de chaves alteradas de x **menos 2**;

Tabela 7.2: Manobras de chaves - caso 2.

| Chaves | Configurações | | |
|--------|---------------|-----|----------|
| | o | x | y |
| 1 | 1 | 1 | 1 |
| 2 | 1 | 0 | 1 |
| 3 | 0 | 1 | 0 |
| 4 | 0 | 0 | 0 |
| 5 | 1 | 1 | 1 |

3. O estado de uma das chaves alteradas em y , em relação a x , é igual ao estado dessa chave em o e, o estado da outra chave alterada é diferente. Deste modo, o número de chaves alteradas de y será igual ao número de chaves alteradas de x .

Tabela 7.3: Manobras de chaves - caso 3.

| Chaves | Configurações | | |
|--------|---------------|-----|----------|
| | o | x | y |
| 1 | 1 | 1 | 1 |
| 2 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 |
| 4 | 0 | 0 | 0 |
| 5 | 1 | 1 | 0 |

Em todos os casos acima, são necessárias 2 mudanças de chaves em x para originar y , porém não garante que essas mudanças sejam efetivas, ou seja, em relação à configuração inicial o . Por esse motivo, não se pode simplesmente dizer que o número de manobras necessárias para implantar y é o número de manobras para implantar x mais 2.

7.4 Algoritmo Evolutivo para Restabelecimento de SDR

Em (dos Santos, 2008), foi desenvolvido um algoritmo evolutivo que busca estratégias para restabelecer energia, em SDR de grande porte, após a ocorrência de uma ou múltiplas faltas. Devido à radialidade do sistema, uma área fora de serviço pode ser considerada como uma subárvore de grafo, tornando possível a aplicação dos operadores da RNP (ver Capítulo 5) para conectá-la a uma outra árvore (alimentador), quando possível. A aplicação desses operadores gera um novo indivíduo na população, que representa uma configuração factível. Porém, este indivíduo pode violar as restrições operacionais do sistema, inviabilizando o uso de tal configuração na prática. Desta forma, é necessário encontrar uma nova configuração que atenda áquelas restrições. Para suprir este problema, o AE proposto em (Santos, Delbem e Bretas, 2008b) inicia a sua busca por melhores configurações partindo de uma configuração factível e, a partir de tal, é capaz de produzir soluções melhores e adequadas para utilização na prática.

Basicamente, o AE proposto em (dos Santos, 2008) trabalha em paralelo com várias subpopulações armazenadas em tabelas (R. Benayoun e Laritchev, 1973). Os melhores indivíduos para cada característica (perdas resistivas, queda de tensão, carregamento da rede, carregamento da subestação/transformadores) do problema são armazenados em sua respectiva subpopulação. Uma subpopulação foi criada para armazenar os indivíduos avaliados por uma função agregação.

Alguns parâmetros são estabelecidos para o desempenho do AE:

1. O tamanho de uma subpopulação S_{pi} , indica o número máximo de indivíduos que podem permanecer em uma subpopulação P_i de uma geração para outra;
2. O número máximo de geração (G_{max}).

As soluções geradas pelo AE, dependendo do grau de adaptação do indivíduo a cada objetivo (característica do problema em uma subpopulação P_i) do problema, podem ser armazenadas ou descartadas. Inicialmente, é gerado apenas um indivíduo em cada população i (corresponde à configuração original F_0). Na etapa de seleção de sobreviventes acrescenta-se um novo indivíduo a subpopulação, apenas se o número de indivíduos for menor que S_{pi} ou, se a sua adequação ao objetivo daquela subpopulação for melhor que pelo menos um indivíduo da mesma. Dependendo do critério de seleção, um indivíduo

pode ser incluído em mais de um conjunto. Como a população é estacionária, os novos indivíduos substituem os piores, desta forma, mantendo os melhores indivíduos.

Para aumentar a diversidade da população geral, os indivíduos selecionado para a reprodução podem ser escolhidos de qualquer tabela. Com isso, o AE consegue escapar de possíveis ótimos locais, aumentando assim a possibilidade de encontrar uma solução global, ou, soluções bem próximas a um ótimo global.

A escolha do operador de reprodução é feita segundo uma taxa de adaptação variável. O algoritmo inicia usando a mesma taxa de probabilidade para os dois operadores ($OP1 = OP2 = 0,50$). Suponha que o operador 1 foi escolhido para gerar um novo indivíduo. Se o indivíduo gerado tiver sucesso, ou seja, for acrescentado a uma ou mais subpopulações, aumenta-se $OP1$ para 0,51 e, como consequência, $OP2$ é reduzido para 0,49, e assim por diante. Os resultados apresentados em (Santos, Delbem e Bretas, 2008b; Santos, Delbem e Bretas, 2008a) mostram que esse ajuste dinâmico, do processo de escolha dos operadores, melhora consideravelmente o desempenho do algoritmo.

A cada novo indivíduo gerado, a rotina de fluxo de carga com RNP é executada para avaliar a adaptação do mesmo.

7.5 Comentários Adicionais

A simplificação do problema utilizando a RNP e seus operadores reduz o tempo de processamento, devido a produção exclusiva de configuração factíveis e a avaliação relativamente rápida de cada configuração por um fluxo de carga mais eficiente. Foi implementado um AE que utiliza o método de tabelas para tratar o problema de re-configuração. As configurações encontradas pelo AE atendem as restrições operacionais e minimizam tanto manobras quanto o total de perdas resistivas. Porém este não apresenta um bom mapeamento do problema, pois não gera uma Fronteira de Pareto, diminuindo assim o número de configurações possíveis de serem implementadas em um SDR.

Deste modo, propõe-se, neste trabalho, substituir o método de tabelas por alguma técnica de MOEA, gerando assim uma Fronteira de Pareto, de forma a conseguir encontrar configurações melhores das obtidas em (Santos, Delbem e Bretas, 2008b).

Capítulo 8

Algoritmo Proposto para Restabelecimento de Energia para SDR

Neste Capítulo apresenta-se o algoritmo proposto para obtenção de planos de restabelecimento para SDR de grande porte, que se baseia no trabalho apresentado em (Santos, Delbem e Bretas, 2008b), descrito no capítulo 7, e na técnica NSGA-II.

O NSGA-II é a técnica de MOEA mais utilizada na literatura e já foi aplicada para resolver o problema de restabelecimento em SDR (Kumar et al., 2008). Porém, o NSGA-II não apresenta bons resultados quando aplicado em problemas com um grande número de funções objetivo (geralmente acima de três objetivos), conforme abordado em (Deb e Sundar, 2006b). Deste modo, para lidar com o problema de restabelecimento de energia em SDR, foram necessárias algumas modificações no NSGA-II. Tais modificações serão descritas a seguir. Na Seção 8.1 é apresentada a formulação matemática utilizada no AE proposto. Na Seção 8.2 é descrito o AE proposto e na Seção 8.3, é apresentado um exemplo ilustrativo da sua aplicação.

8.1 Formulação Matemática

Os AEs convencionais, ou mono-objetivos, quando aplicados para o problema multi-objetivo de restabelecimento de energia em SDR, fazem uso de uma função agregação

e perdem eficiência quando aplicados em SDR de grande porte (Kumar et al., 2008).

Para contornar esse problema, em (Santos, Delbem e Bretas, 2008b) desenvolveu-se um AE para reconfiguração de SDR que faz uso da RNP e de seus operadores. Destaca-se o fato de tais operadores gerarem apenas configurações factíveis. Além disso, em razão de a RNP permitir o armazenamento dos nós de acordo com o MPF, em (Santos, Delbem e Bretas, 2008b) o problema de restabelecimento foi formulado de uma forma mais simples (veja Equação 7.4).

O AE proposto neste trabalho também faz uso da RNP e dos seus operadores, entretanto baseia-se no NSGA-II. Assim, são necessárias algumas modificações na formulação do problema de restabelecimento de energia proposta em (Santos, Delbem e Bretas, 2008b).

Na formulação que está sendo proposta neste trabalho consideram-se duas funções objetivo: a primeira é o número de operações de chaveamento, que deve ser o menor possível para garantir um rápido restabelecimento de energia; e a segunda é uma função agregação que contempla os demais objetivos do problema de restabelecimento (perdas resistivas, restrições operacionais dentre outros). Dessa forma, o problema descrito na Equação 7.4 é reescrito como segue:

$$\begin{aligned} \text{Min.} \quad & E = [e_1(F) \mid |\Omega I'(F)|] \\ \text{s.a.} \quad & \\ & \textit{Utilizar MPF com RNP} \\ & \textit{F é dado pelos operadores da RNP,} \end{aligned} \tag{8.1}$$

onde E é um vetor de duas funções objetivo: $e_1(F)$ é o número de operações de chaveamento; e $|\Omega I'(F)|$ é a função agregação composta pelas restrições operacionais (queda de tensão, carregamento da rede e carregamento do sistema) e pelas perdas resistivas. A penalidade Ω é uma matriz diagonal composta pelos seguintes elementos:

$$w_{11} = \begin{cases} w_x, & \text{se, pelo menos para um } j, x_j > \bar{x}_j \\ 0, & \text{caso contrário;} \end{cases}$$

$$w_{22} = \begin{cases} w_s, & \text{se, pelo menos para um } i, b_i > \bar{b}_i \\ 0, & \text{caso contrário;} \end{cases}$$

$$w_{33} = \begin{cases} w_v, & \text{se, pelo menos para um } i, v_i < \underline{v} \\ 0, & \text{caso contrário.} \end{cases}$$

onde,

- Um limitante superior de corrente \bar{x}_j para cada corrente de linha x_j . A maior taxa x_j/\bar{x}_j é denominada carregamento da rede;
- A máxima injeção de corrente \bar{b}_i possível para cada subestação i , onde a maior taxa b_i/\bar{b}_i é denominada carregamento da subestação;
- Um limitante inferior para a tensão no nó \underline{v} . Seja v_i a tensão na barra i e v_b a tensão base no sistema; a maior taxa v_i/v_b é denominada maior taxa de tensão.

Conforme mencionado anteriormente, o NSGA-II não apresenta um bom desempenho para problemas com mais de três funções objetivo, como descrito em Deb e Sundar (2006b). No AE proposto, o NSGA-II não apresenta bom desempenho para mais de duas funções objetivo. Eis a razão de a formulação proposta (8.1) utilizar apenas duas funções objetivo.

8.2 Algoritmo Proposto

Como mencionado anteriormente, o algoritmo proposto (NS2R) baseia-se na RNP e no NSGA-II. Porém, devido à utilização da RNP, são necessárias algumas modificações no NSGA-II tradicional, apresentado no Capítulo 4, pois no NS2R as populações P e Q são geradas pelos operadores da RNP. Lembrando que no NSGA-II tradicional, aquelas populações são geradas por operadores genéticos tradicionais.

Inicialmente, o operador 1 é utilizado para conectar a área fora de serviço a algum alimentador, gerando assim o primeiro indivíduo da população P_1 . Este indivíduo representa uma configuração factível e através dele são gerados os demais $N_{ind} - 1$

indivíduos restantes da população P_1 . Os demais passos são similares ao algoritmo 4 (ver Seção 4.2.1), com exceção da geração das populações, conforme discutido anteriormente.

Algoritmo 8 NS2R

```

1: Criar o primeiro indivíduo da população  $P_1$  usando o operador 1
2: Criar os demais  $N_{ind} - 1$  indivíduos da população  $P_1$  usando o operador 1
3: Aplicar Algoritmo 2, para classificar  $P_1$  por não dominância
4:  $Pais \leftarrow$  selecionar os indivíduos progenitores em  $P_1$  utilizando o operador  $\prec_c$ 
5: Gerar a população  $Q_1$  de tamanho  $N_{ind}$ , aplicando o operador 1 em  $Pais$ 
6: for geração  $t \leftarrow 1, 2, \dots, G_{max}$  do
7:   Aplicar Algoritmo 2, para classificar  $R_t \leftarrow P_t \cup Q_t$  por não dominância
8:    $k \leftarrow 1$ 
9:    $P_{t+1} \leftarrow \{\}$ 
10:  while  $|P_{t+1} + \mathcal{F}_k| \leq N_{ind}$  do
11:    Aplicar algoritmo 3 em  $\mathcal{F}_k$ , para calcular as distâncias de multidão
12:     $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_k$ 
13:     $k \leftarrow k + 1$ 
14:  end while
15:  Aplicar o Algoritmo 3 em  $\mathcal{F}_k$ , para calcular as distâncias de multidão
16:  Classificar a fronteira  $\mathcal{F}_k$  pelo ranking e a distância usando o operador  $\prec_c$ 
17:  Copiar as primeiras  $N_{ind} - |P_{t+1}|$  soluções de  $\mathcal{F}_k$  para  $P_{t+1}$ 
18:   $Pais \leftarrow$  selecionar indivíduos progenitores em  $P_{t+1}$  pelo o operador  $\prec_c$ 
19:   $Op \leftarrow$ decidir-operador(operador1, operador2,  $t$ )
20:  Gerar a nova população  $Q_{t+1}$  aplicando o operador  $Op$  em  $Pais$ 
21:   $t \leftarrow t + 1$ 
22: end for
23:  $P_{final} \leftarrow P_t$ 
24:  $Q_{final} \leftarrow Q_t$ 

```

O algoritmo NS2R foi implementado computacionalmente dando origem ao Programa para obtenção de planos de Restabelecimento de energia em SDRs utilizando o NS2R (PRN).

8.3 Exemplo Ilustrativo da Aplicação do PRN

Para ilustrar o funcionamento do PRN, considere o SDR composto por 93 barras, 25 setores, 8 alimentares e 30 chaves, que é uma versão modificada do SDR utilizado em (Kagan, 1999). Após aplicar o procedimento apresentado no Capítulo 7, Seção 7.3.1, este SDR é representado da forma ilustrada na Figura 8.1, onde os círculos são os setores, os retângulos os alimentadores, as linhas cheias são as chaves NF e as linhas trasejadas

são chaves NA.

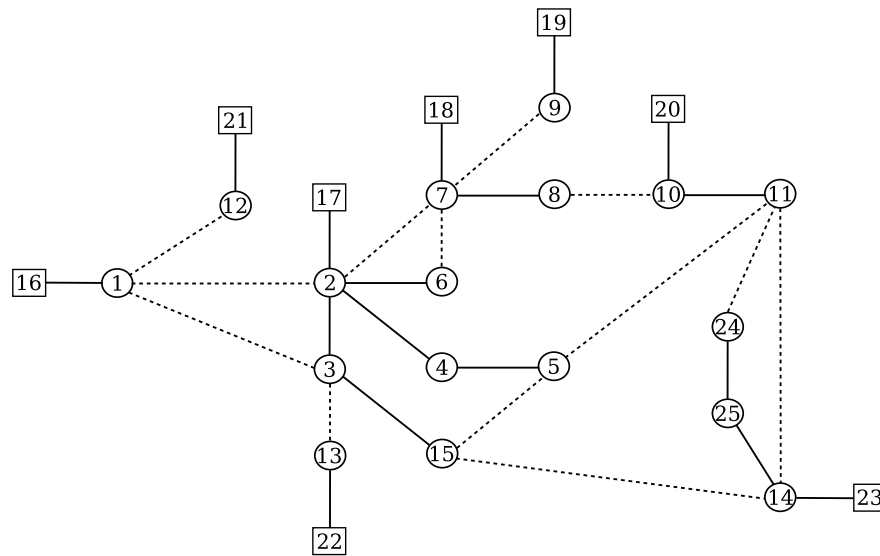


Figura 8.1: SDR com 8 alimentadores.

Vamos considerar que o setor 4, em vermelho na Figura 8.2, está em falta. Em seguida esse setor é isolado do restante do SDR. Esse procedimento pode ser realizado abrindo-se as chaves $c13$ e $c18$ (em azul na Figura 8.3). Após a abertura dessas chaves, o setor 5 ficou sem fornecimento de energia elétrica. Nessa situação o PRN deve encontrar outro caminho que forneça energia a esse setor, sem violar as restrições operacionais do sistema. Para isso, nesse exemplo, a chave $c21$ (em azul na Figura 8.3) foi fechada. Assim, nessa primeira etapa foram necessárias 3 manobras para isolar o setor faltoso e restabelecer energia no SDR.

Após a obtenção dessa configuração inicial, ilustrada na Figura 8.3, avalia-se a mesma através do fluxo de carga baseado na RNP e do número de manobras necessárias para implementá-la, determinando assim a sua adequação para o problema de restabelecimento de energia. Obtêm-se os seguintes resultados:

- Perdas resistivas = 1403,33kW;
- Carregamento da rede = 103,00%;
- Carregamento do Sistema = 70,25%;
- Queda de tensão = 7,32%.

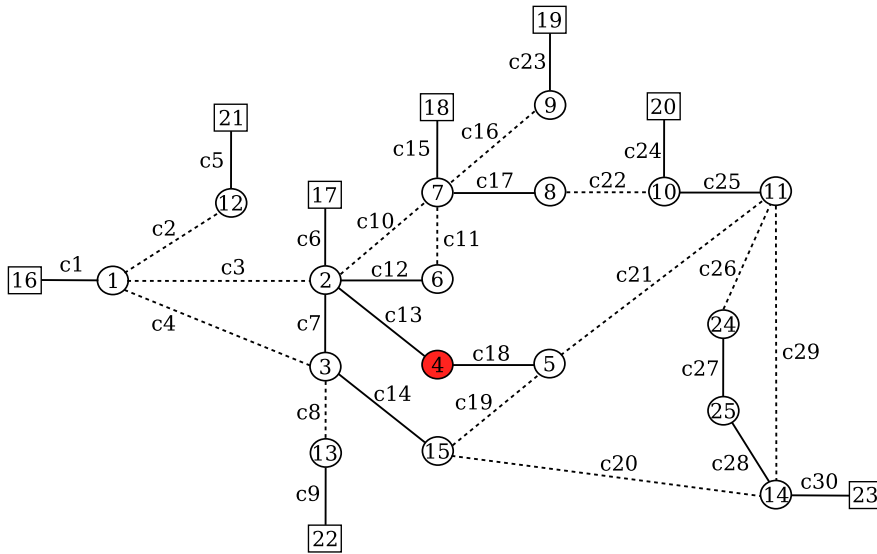


Figura 8.2: SDR com falta no setor 4.

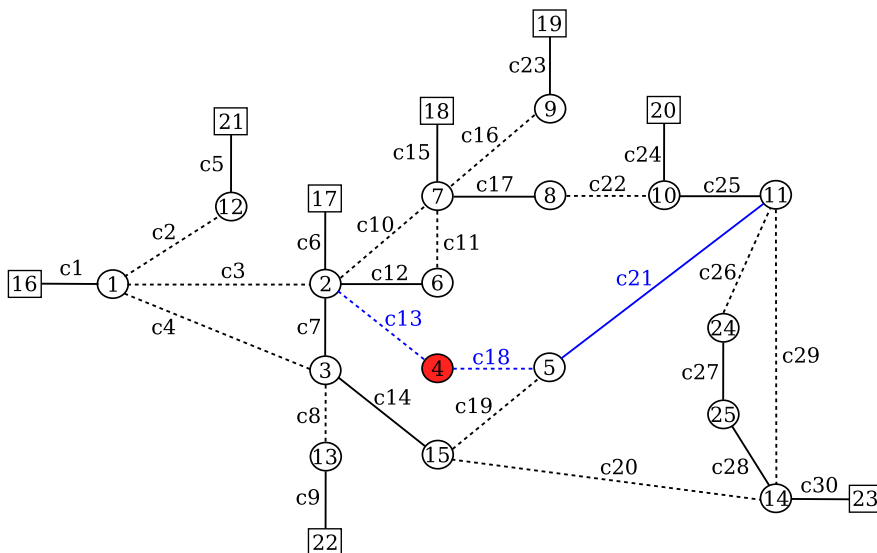


Figura 8.3: SDR restabelecido.

Verifica-se então que os limites operacionais do sistema não foram obedecidos, que no caso são o carregamento da rede que têm que estar abaixo dos 100%, e a queda de tensão que têm que ser menor que 7%, conforme a especificação da ANEEL (Agência Nacional de Energia Elétrica) (ANEEL, 2008). Desta forma, de acordo com o Algoritmo 8, tomando por base essa primeira configuração, são criadas as populações P_1 e Q_1 .

Na Figura 8.4 apresenta-se a melhor configuração obtida, em termos de função agregação, a partir da união das populações P_1 e Q_1 . Para obter essa configuração, a partir da configuração inicial, as chaves $c17$ e $c14$ foram abertas e as chaves $c22$ e $c20$ foram fechadas, resultando um total de 4 manobras. Assim, para implantar essa configuração são necessárias 7 manobras, pois para obter a configuração inicial foram realizadas 3 manobras (ver Capítulo 7, Seção 7.3.3).

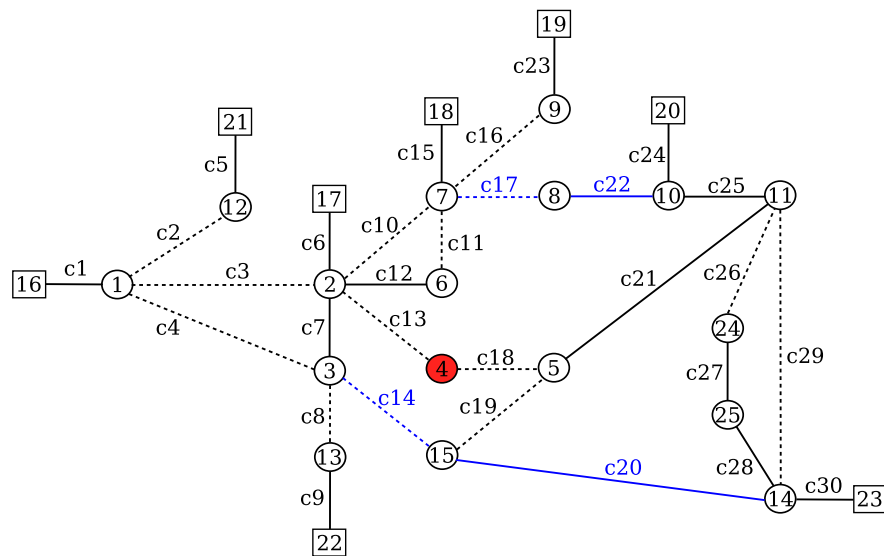


Figura 8.4: Melhor configuração alterada.

O resultado do fluxo de carga, baseado na RNP, para essa configuração fornece os seguintes resultados:

- Perdas resistivas = 1331,53kW;
- Carregamento da rede = 97,80%;
- Carregamento do Sistema = 78,15%;
- Queda de tensão = 7,13%.

Após a obtenção das populações P_1 e Q_1 , de acordo com o Algoritmo 8, o PRN entra

em um processo iterativo para encontrar configurações melhores das que foram geradas até o presente momento. Desta forma, até atingir o critério de parada (o número máximo de gerações neste problema é igual a 30), o PRN gera várias configurações factíveis, avaliando-as e mantendo as melhores.

Na Figura 8.5 é ilustrada a melhor configuração gerada no meio do processo iterativo, isto é, na iteração 15. Observa-se que, em relação à configuração inicial, a chave $c17$ foi aberta e a chave $c22$ foi fechada, resultando um total de 2 manobras. Assim, para implantar essa configuração são necessárias 5 manobras, pois para obter a configuração inicial foram realizadas 3 manobras.

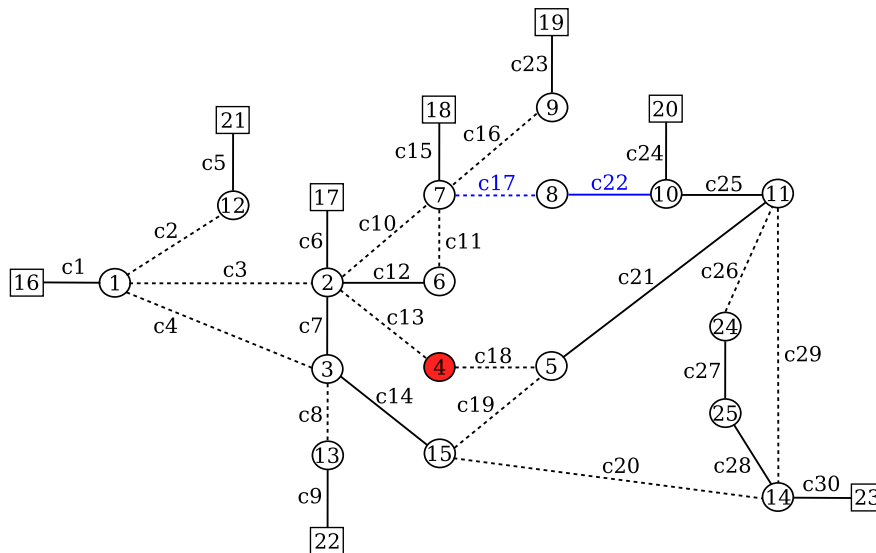


Figura 8.5: Melhor configuração alterada gerada no meio do processo iterativo.

O resultado do fluxo de carga, baseado na RNP, para essa configuração fornece os seguintes resultados:

- Perdas resistivas = 1329,36kW;
- Carregamento da rede = 85,58%;
- Carregamento do Sistema = 78,15%;
- Queda de tensão = 7,06%.

Attingido o número máximo de iterações, são obtidas as populações P_{final} e Q_{final} , que contêm as melhores configurações geradas pelo PRN. Na Figura 8.6 é ilustrada a melhor configuração, em termos de função agregação. Observa-se que, em relação à

configuração inicial, as chaves $c7$, $c17$ e $c25$ foram abertas e as chaves $c4$, $c22$ e $c26$ foram fechadas, resultando um total de 6 manobras. Assim, para implantar essa configuração são necessárias 9 manobras, pois para obter a configuração inicial foram realizadas 3 manobras são necessárias 9 manobras.

O resultado do fluxo de carga, baseado na RNP, para essa configuração fornece os seguintes resultados:

- Perdas resistivas = 1324,40kW;
- Carregamento da rede = 73,77%;
- Carregamento do Sistema = 78,15%;
- Queda de tensão = 6,92%V.

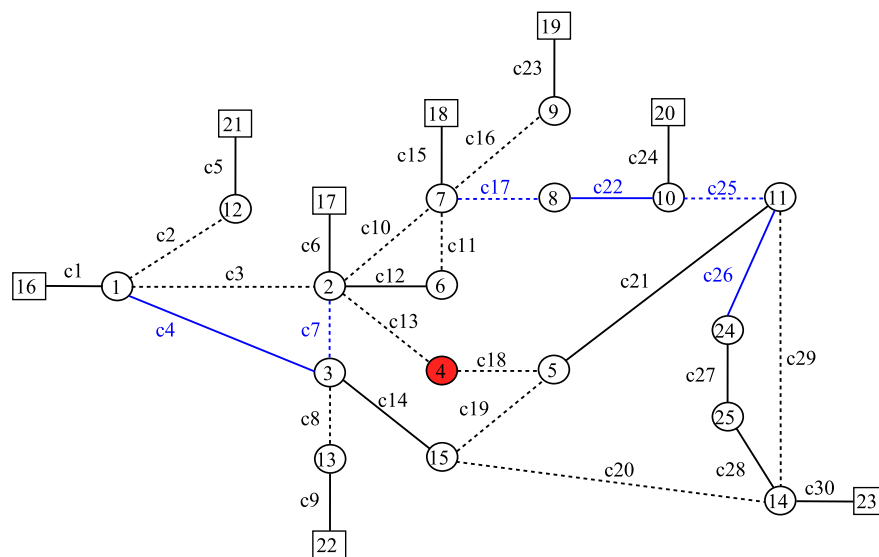


Figura 8.6: Melhor configuração gerada no final do processo iterativo.

Capítulo 9

Testes e Resultados

Os testes realizados, descritos neste Capítulo, visam comprovar a eficiência do PRN. Foi utilizado um SDR real de grande porte sem simplificações (foram consideradas todas as linhas, chaves e barras do sistema). O sistema é composto de: 3860 barras, 635 chaves, 3 subestações com 2 transformadores com potência de 50MVA e 1 transformador de 25MVA e 23 alimentadores. O sistema utilizado corresponde ao SDR da cidade de São Carlos/SP.

O PRN foi implementado num computador convencional com processador Turion64 X2; 2 Gbytes de memória RAM, Sistema operacional Linux, distribuição Ubuntu 8.04; e o GCC (*GNU Compiler Collection*) com compilador de linguagem C. Os resultados obtidos pelo PRN são comparados com os obtidos pelo programa desenvolvido em (dos Santos, 2008), que será aqui denominado AERT (Algoritmo Evolutivo utilizando a RNP e o Método de Tabelas).

Na Seção 9.1 são apresentados os parâmetros utilizados nos programas PRN e AERT. Na Seção 9.2 são apresentados os resultados das simulações com falta únicas e multifaltas no SDR real da cidade de São Carlos/SP. Na Seção 9.3 o SDR real de São Carlos/SP tem o seu tamanho duplicado para avaliar a eficiência do PRN com um número mais elevado de barras, chaves, linhas e cargas.

9.1 Testes Realizados

Os parâmetros utilizados para análise do PRN, são:

1. Número máximo de gerações: $G_{max} = 200$;
2. Número de indivíduos nas populações P e Q : $N_{ind} = 110$.

Na formulação matemática 8.1 a função agregação é $f(x) = \delta_1 x_1 + \delta_2 x_2 + \delta_3 x_3 + \delta_4 x_4$, onde, x_1, x_2, x_3 e x_4 correspondem respectivamente a: perdas resistivas em kW, máximo carregamento da rede em p.u., máximo carregamento do sistema em p.u., máxima queda de tensão em p.u.; δ_i é o peso de cada objetivo x_i , sendo:

$$\delta_i = \begin{cases} 100, & \text{se, uma restrição foi violada para pelo menos uma barra} \\ 0, & \text{caso contrário,} \end{cases}$$

para $i = 2, 3, 4$ e $\delta_1 = 1$.

Os objetivos a serem minimizados em cada execução do PRN foram:

1. Redução do número de manobras;
2. Redução da função agregação.

Para o AERT, tem-se apenas uma função objetivo, que consiste de uma função agregação composta por 5 objetivos: $f(x) = \delta_1 x_1 + \delta_2 x_2 + \delta_3 x_3 + \delta_4 x_4 + \delta_5 x_5$, onde: x_1, x_2, x_3, x_4 e x_5 correspondem respectivamente a: perdas resistivas em kW, operações de chaveamento, máximo carregamento da rede em p.u., máximo carregamento do sistema em p.u., máxima queda de tensão em p.u.; δ_i é o peso de cada objetivo x_i , sendo:

$$\delta_i = \begin{cases} 100, & \text{se, uma restrição foi violada para pelo menos uma barra} \\ 0, & \text{caso contrário,} \end{cases}$$

para $i = 3, 4, 5$, $\delta_1 = 1$ e $\delta_2 = 2$.

Para este programa foram utilizados os seguintes parâmetros: 10000 indivíduos ($G_{max} = 10000$), e os melhores indivíduos de uma população (de tamanho 20) são mantidos.

As configurações encontradas por ambos os programas (PRN e AERT) são avaliadas de acordo com as seguintes características:

- Número de manobras de chaveamento necessário para a implementação do plano de restabelecimento;
- Total de perdas resistivas em kW;
- A maior queda de tensão (%), que é dada pela expressão:

$$\max \left\{ \frac{V_b - V_i}{V_b} \right\}, \quad (9.1)$$

onde, $V_b = 13800V$ e V_i é a tensão na barra i .

- O maior carregamento da rede (%), que é dado pela expressão:

$$\max \left\{ \frac{I_c}{I_{ij}} \right\}, \quad (9.2)$$

onde, I_c é a corrente calculada na linha ij e I_{ij} é a corrente máxima suportada na linha ij .

- Carregamento máximo do SDR (%), que corresponde ao carregamento do transformador mais sobrecarregado do SDR, dado pela expressão:

$$\begin{aligned} CarT_i &= \frac{\sum S_j^* + Perdas_j}{T_i}, \\ \max \{CarT_i\} \end{aligned} \quad (9.3)$$

onde $CarT_i$ é o carregamento no transformador i , S_j^* é a potência complexa na barra j , $Perdas_j$ são as perdas na barra j e T_i é o máximo carregamento suportado no transformador i .

Em todas as simulações realizadas ambos os programas foram executados 20 vezes. Para o AERT, considerou-se como melhor configuração aquela que apresentou a menor função agregação na população analisada; para o PRN, considerou-se como melhor configuração aquela que apresentou o menor número de manobras, menor queda de tensão e menor perdas resistivas, simultaneamente, isto é, a configuração selecionada é aquela que garante a continuidade do fornecimento de energia, com o menor número de chaveamento e perdas resistivas.

9.2 Resultados das Simulações no SDR Real de São Carlos

A seguir serão considerados testes com falta única no setor 3668, que pertence ao alimentador mais sobrecarregado do SDR, bem como múltiplas faltas, estas localizadas no setores 3668, 3542 e 1031 do SDR.

9.2.1 Falta Única

Após a ocorrência da falta no setor 3668, o serviço do maior alimentador do SDR é interrompido e a primeira topologia da rede, após isolar os setores defeituosos, possui as seguintes características:

- Perdas resistivas = 415,02kW;
- Carregamento da rede = 139,60%;
- Carregamento do Sistema = 52,72%;
- Queda de tensão = 5,02%.

Após a execução do PRN várias Fronteiras de Pareto, de diferentes níveis, foram geradas. Isto ocorre em razão da complexidade do problema de restabelecimento de energia em SDR. Para facilitar a análise dos resultados, na Figura 9.1 e na Tabela 9.1, foram apresentadas apenas as configurações da primeira Fronteira de Pareto.

Tabela 9.1: Configurações da primeira Fronteira de Pareto após a falta única.

| Função Agregação | Perdas Resistivas [kW] | Queda de Tensão [%] | Carregamento da Rede [%] | Carregamento do Sistema [%] | Manobras |
|------------------|------------------------|---------------------|--------------------------|-----------------------------|----------|
| 522,28 | 382,68 | 4,39 | 139,60 | 52,78 | 5 |
| 480,25 | 357,50 | 4,14 | 122,74 | 52,81 | 9 |
| 512,64 | 373,04 | 4,39 | 139,60 | 52,78 | 7 |
| 480,25 | 357,50 | 4,14 | 122,74 | 52,71 | 9 |
| 480,25 | 357,50 | 4,14 | 122,74 | 53,08 | 9 |
| 480,25 | 357,50 | 4,14 | 122,74 | 52,81 | 9 |
| 480,25 | 357,50 | 4,14 | 122,74 | 50,82 | 9 |
| 269,85 | 269,85 | 3,14 | 82,54 | 51,05 | 39 |
| 267,76 | 267,76 | 3,14 | 82,54 | 76,07 | 41 |
| 400,66 | 400,66 | 5,63 | 103,00 | 56,12 | 29 |
| 464,16 | 346,66 | 4,08 | 117,50 | 54,90 | 15 |
| 272,79 | 272,79 | 3,14 | 82,54 | 52,89 | 37 |
| 471,80 | 349,05 | 4,14 | 122,74 | 52,78 | 11 |

Tabela 9.1: Configurações da primeira Fronteira de Pareto após a falta única. (continuação)

| Função Agregação | Perdas Resistivas [kW] | Queda de Tensão [%] | Carregamento da Rede [%] | Carregamento do Sistema [%] | Manobras |
|------------------|------------------------|---------------------|--------------------------|-----------------------------|----------|
| 289,87 | 289,87 | 3,14 | 82,54 | 52,98 | 35 |
| 416,61 | 416,61 | 5,50 | 101,03 | 47,12 | 17 |
| 412,07 | 412,07 | 5,50 | 101,03 | 52,68 | 21 |
| 261,76 | 261,76 | 3,13 | 71,61 | 50,89 | 57 |
| 266,09 | 266,09 | 3,14 | 82,54 | 74,80 | 43 |
| 264,79 | 264,79 | 3,13 | 82,54 | 75,40 | 49 |
| 413,71 | 413,71 | 5,50 | 101,03 | 55,22 | 19 |
| 264,79 | 264,79 | 3,13 | 82,54 | 74,94 | 49 |
| 409,68 | 409,68 | 5,50 | 101,03 | 53,80 | 23 |
| 409,68 | 409,68 | 5,50 | 101,03 | 53,80 | 23 |
| 264,55 | 264,55 | 3,13 | 82,54 | 52,77 | 51 |
| 409,68 | 409,68 | 5,50 | 101,03 | 53,80 | 23 |
| 261,76 | 261,76 | 3,13 | 71,61 | 51,16 | 57 |
| 259,93 | 259,93 | 3,13 | 71,61 | 54,86 | 65 |
| 408,70 | 408,70 | 5,50 | 101,03 | 50,89 | 25 |
| 265,26 | 265,26 | 3,13 | 82,54 | 53,37 | 47 |
| 261,46 | 261,46 | 3,13 | 71,61 | 52,72 | 59 |
| 261,11 | 261,11 | 3,13 | 71,61 | 75,47 | 63 |
| 265,68 | 265,68 | 3,13 | 82,54 | 78,46 | 45 |
| 408,70 | 408,70 | 5,50 | 101,03 | 52,78 | 25 |
| 408,52 | 408,52 | 5,50 | 101,03 | 47,14 | 27 |
| 408,52 | 408,52 | 5,50 | 101,03 | 47,14 | 27 |
| 262,12 | 262,12 | 3,13 | 74,23 | 50,24 | 55 |
| 259,76 | 259,76 | 3,13 | 71,61 | 70,02 | 67 |
| 464,16 | 346,66 | 4,08 | 117,50 | 52,77 | 15 |
| 261,76 | 261,76 | 3,13 | 71,61 | 48,12 | 57 |
| 264,45 | 264,45 | 3,13 | 82,54 | 52,78 | 53 |
| 467,47 | 347,34 | 4,10 | 120,12 | 55,16 | 13 |
| 290,90 | 290,90 | 3,14 | 82,54 | 76,52 | 33 |
| 259,19 | 259,19 | 3,13 | 71,61 | 69,27 | 69 |
| 292,28 | 292,28 | 3,14 | 82,54 | 76,26 | 31 |
| 264,45 | 264,45 | 3,13 | 82,54 | 48,92 | 53 |
| 261,22 | 261,22 | 3,13 | 71,61 | 54,71 | 61 |

Analisando a Tabela 9.1, percebe-se que em determinadas configurações o valor da função agregação não é igual ao de perdas resistivas, isso ocorre pelo fato de algumas restrições estarem sendo penalizadas na função de agregação por violarem os limites operacionais dos equipamentos. A Figura 9.2 mostra a redução total de perdas resistivas para a melhor configuração em cada geração (considera-se melhor configuração em cada geração aquela que apresentar o menor valor de função agregação). Para essas mesmas configurações, a Figura 9.3 mostra as manobras. Como os objetivos são conflitantes, percebe-se que quando ocorre uma redução nas perdas resistivas aumenta a necessidade de manobras no sistema, isto pode ser facilmente observado na Figura 9.4, nesta figura

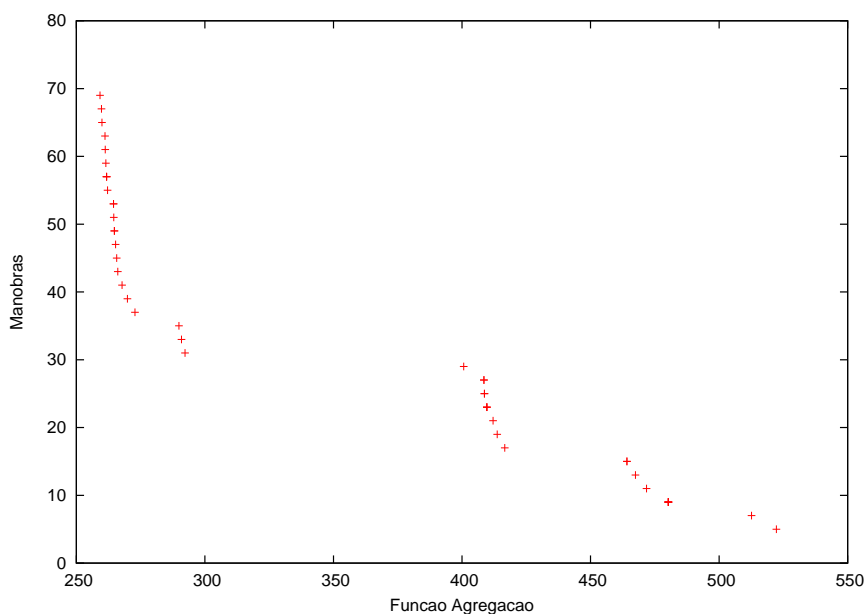


Figura 9.1: Primeira Fronteira de Pareto após a falta única no setor 3668.

são linearizados o número de manobras e o de perdas de cada configuração. Estes resultados foram obtidos utilizando semente 6, para o gerador de números aleatórios, e o tempo de processamento foi de 3250ms.

Análise Comparativa

Testes foram realizados utilizando o programa AERT, aplicando a falta no mesmo setor 3668. A Tabela 9.2 mostra um comparativo entre os dois programas (AERT e PRN).

Tabela 9.2: Comparativo entre os métodos AERT e PRN para falta única no setor 3668.

| | AERT | | PRN | |
|-----------------------------|--------------|---------------|-------------|---------------|
| | Média | Desvio Padrão | Média | Desvio Padrão |
| Perdas Resistivas [kW] | 325,49 | 33,78 | 355,50 | 37,47 |
| Queda de Tensão [%] | 4,95 | 1,12 | 4,17 | 0,70 |
| Carregamento da Rede [%] | 107,62 | 10,55 | 98,85 | 18,49 |
| Carregamento do Sistema [%] | 55,97 | 5,66 | 52,74 | 3,22 |
| Manobras | 21,90 | 4,23 | 14,8 | 6,19 |
| Tempo [ms] | 2564,50 | 163,63 | 3110,00 | 165,40 |

Analisando a Tabela 9.2, nota-se que o PRN conseguiu uma maior redução no número de manobras, consequentemente, o AERT obteve um valor de perdas resistivas menor. Para ambos os programas, com exceção do carregamento da rede, as restrições

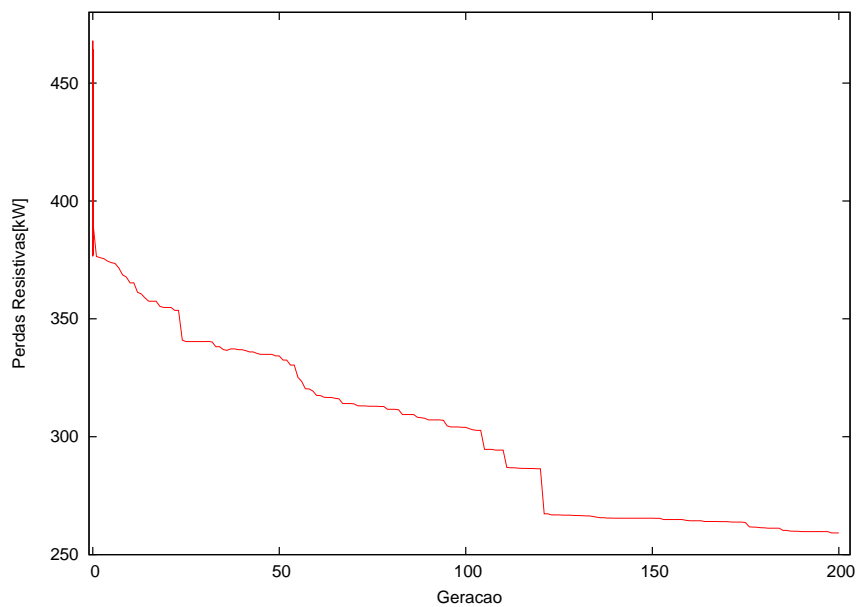


Figura 9.2: Redução das perdas por geração para falta única no setor 3668.

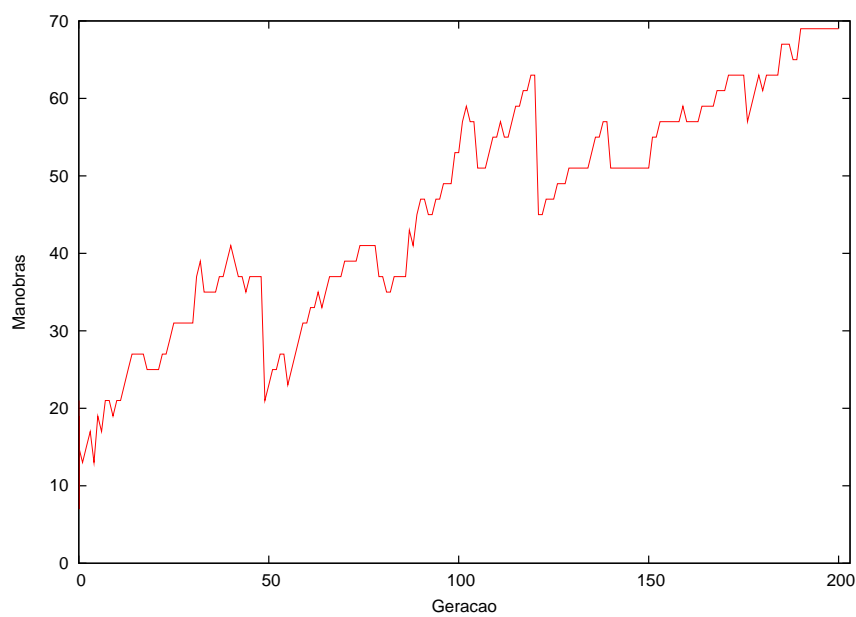


Figura 9.3: Aumento das manobras por geração para falta única no setor 3668.

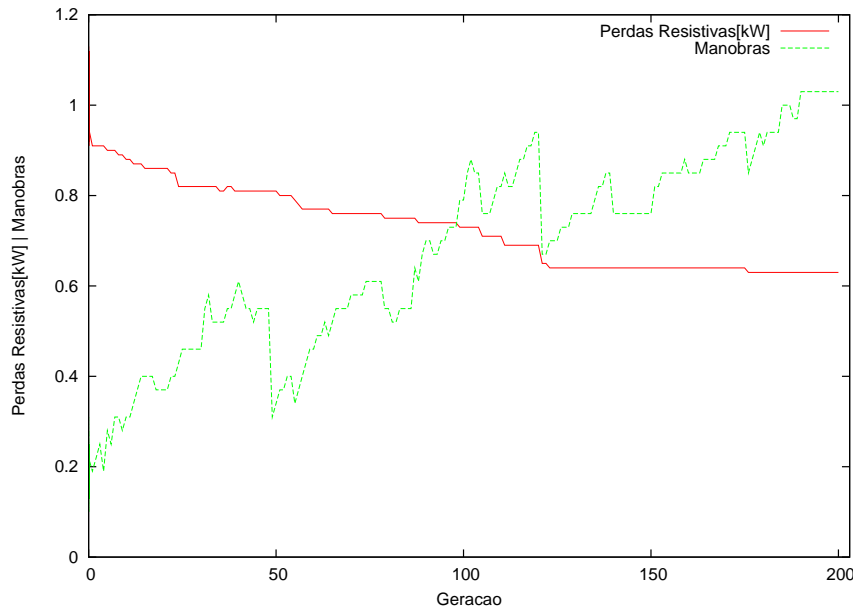


Figura 9.4: Manobras e perdas resistivas linearizadas para falta única no setor 3668.

operacionais não são violadas. O carregamento da rede viola os limites no AERT devido ao setor em falta ser do alimentador mais sobrecarregado do SDR, mesmo gerando apenas configurações factíveis. A vantagem do PRN, neste aspecto, é que é gerada uma Fronteira de Pareto contendo diversas configurações factíveis, aumentando assim a possibilidade de ser encontrada uma solução que não viole os limites operacionais, ao contrário do AERT que gera apenas uma solução por geração.

9.2.2 Múltiplas Faltas

O serviço de fornecimento de energia elétrica é interrompido após a ocorrência das faltas nos setores 3668, 3542 e 1031, assim, a primeira topologia da rede, após isolar os setores defeituosos, possui as seguintes características:

- Perdas resistivas = 447,63kW;
- Carregamento da rede = 83,15%;
- Carregamento do Sistema = 55,07%;
- Queda de tensão = 4,72%.

A Figura 9.5 e a Tabela 9.3 apresentada apenas as configurações da primeira Fronteira de Pareto.

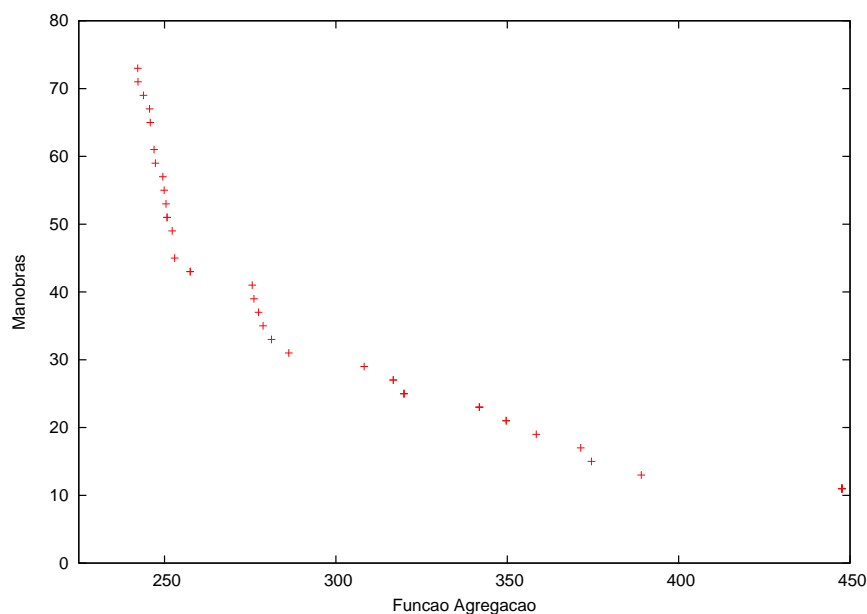


Figura 9.5: Primeira Fronteira de Pareto (múltiplas faltas).

Tabela 9.3: Configurações da primeira Fronteira de Pareto (múltiplas faltas).

| Função Agregação | Perdas Resistivas [kW] | Queda de Tensão [%] | Carregamento da Rede [%] | Carregamento do Sistema [%] | Manobras |
|------------------|------------------------|---------------------|--------------------------|-----------------------------|----------|
| 440,92 | 440,92 | 4,93 | 80,71 | 62,14 | 11 |
| 429,94 | 429,94 | 4,69 | 77,66 | 62,14 | 13 |
| 440,92 | 440,92 | 4,93 | 80,71 | 61,69 | 11 |
| 429,94 | 429,94 | 4,69 | 77,66 | 61,65 | 13 |
| 440,92 | 440,92 | 4,93 | 80,71 | 62,14 | 11 |
| 440,92 | 440,92 | 4,93 | 80,71 | 62,14 | 11 |
| 429,94 | 429,94 | 4,69 | 77,66 | 61,65 | 13 |
| 440,92 | 440,92 | 4,93 | 80,71 | 62,14 | 11 |
| 440,92 | 440,92 | 4,93 | 80,71 | 62,14 | 11 |
| 421,49 | 421,49 | 4,69 | 77,66 | 61,65 | 15 |
| 440,92 | 440,92 | 4,93 | 80,71 | 62,09 | 11 |
| 421,49 | 421,49 | 4,69 | 77,66 | 62,09 | 15 |
| 421,49 | 421,49 | 4,69 | 77,66 | 62,09 | 15 |
| 415,55 | 415,55 | 4,69 | 77,66 | 61,15 | 17 |
| 421,49 | 421,49 | 4,69 | 77,66 | 62,14 | 15 |
| 421,49 | 421,49 | 4,69 | 77,66 | 62,13 | 15 |
| 415,55 | 415,55 | 4,69 | 77,66 | 62,14 | 17 |
| 372,34 | 372,34 | 3,39 | 86,96 | 62,09 | 19 |
| 415,55 | 415,55 | 4,69 | 77,66 | 63,28 | 17 |
| 421,49 | 421,49 | 4,69 | 77,66 | 62,14 | 15 |
| 337,42 | 337,42 | 3,25 | 88,00 | 62,08 | 25 |
| 358,04 | 358,04 | 3,39 | 89,15 | 64,11 | 21 |
| 415,55 | 415,55 | 4,69 | 77,66 | 62,14 | 17 |
| 303,93 | 303,93 | 3,25 | 77,37 | 62,09 | 47 |
| 293,93 | 293,93 | 3,19 | 81,17 | 58,24 | 53 |
| 295,60 | 295,60 | 3,22 | 81,17 | 62,55 | 51 |
| 295,60 | 295,60 | 3,22 | 81,17 | 63,28 | 51 |
| 298,69 | 298,69 | 3,22 | 81,17 | 62,09 | 49 |
| 285,96 | 285,96 | 3,18 | 87,29 | 62,14 | 77 |

Tabela 9.3: Configurações da primeira Fronteira de Pareto após múltiplas faltas (continuação).

| Função Agregação | Perdas Resistivas [kW] | Queda de Tensão [%] | Carregamento da Rede [%] | Carregamento do Sistema [%] | Manobras |
|------------------|------------------------|---------------------|--------------------------|-----------------------------|----------|
| 289,74 | 289,74 | 3,18 | 81,17 | 60,95 | 59 |
| 320,95 | 320,95 | 3,25 | 89,15 | 64,81 | 29 |
| 318,21 | 318,21 | 3,25 | 89,15 | 58,24 | 31 |
| 288,46 | 288,46 | 3,26 | 69,67 | 62,09 | 61 |
| 285,96 | 285,96 | 3,18 | 87,29 | 61,48 | 77 |
| 286,21 | 286,21 | 3,26 | 69,67 | 62,09 | 69 |
| 337,88 | 337,88 | 3,25 | 89,15 | 62,14 | 23 |
| 292,49 | 292,49 | 3,18 | 81,17 | 58,24 | 55 |
| 285,05 | 285,05 | 3,18 | 87,29 | 62,14 | 85 |
| 286,27 | 286,27 | 3,26 | 69,67 | 60,90 | 67 |
| 284,88 | 284,88 | 3,18 | 87,29 | 60,59 | 89 |
| 291,70 | 291,70 | 3,18 | 81,17 | 62,09 | 57 |
| 285,22 | 285,22 | 3,18 | 87,29 | 62,08 | 79 |
| 285,98 | 285,98 | 3,18 | 87,29 | 62,09 | 75 |
| 285,16 | 285,16 | 3,18 | 85,78 | 62,14 | 83 |
| 318,21 | 318,21 | 3,25 | 89,15 | 62,13 | 31 |
| 285,18 | 285,18 | 3,18 | 85,78 | 62,14 | 81 |
| 309,85 | 309,85 | 3,25 | 89,15 | 62,14 | 33 |
| 288,31 | 288,31 | 3,26 | 69,67 | 63,28 | 63 |
| 304,33 | 304,33 | 3,25 | 95,98 | 54,10 | 43 |
| 286,15 | 286,15 | 3,18 | 87,29 | 62,14 | 73 |
| 328,94 | 328,94 | 3,25 | 89,15 | 54,98 | 27 |
| 304,20 | 304,20 | 3,25 | 95,98 | 62,14 | 45 |
| 286,40 | 286,40 | 3,26 | 69,67 | 62,60 | 65 |
| 292,49 | 292,49 | 3,18 | 81,17 | 62,09 | 55 |

Analisando a Tabela 9.3 percebe-se que, ao contrário do ocorrido na Tabela 9.1, o valor da função agregação é igual ao de perdas resistivas em todas as configurações. Isso ocorre pelo fato de as configurações encontradas não violarem os limites operacionais dos equipamentos. Considerando como melhor configuração aquela que apresentar a menor função agregação, a Figura 9.6 mostra a redução total das perdas resistivas para a melhor configuração em cada geração. Para essas mesmas configurações, a Figura 9.7 mostra as manobras e na Figura 9.8 são mostrados os valores linearizados do número de manobras e o de perdas de cada configuração. Estes resultados foram obtidos utilizando semente 14, para o gerador de número aleatórios, e o tempo de processamento foi de 3080ms.

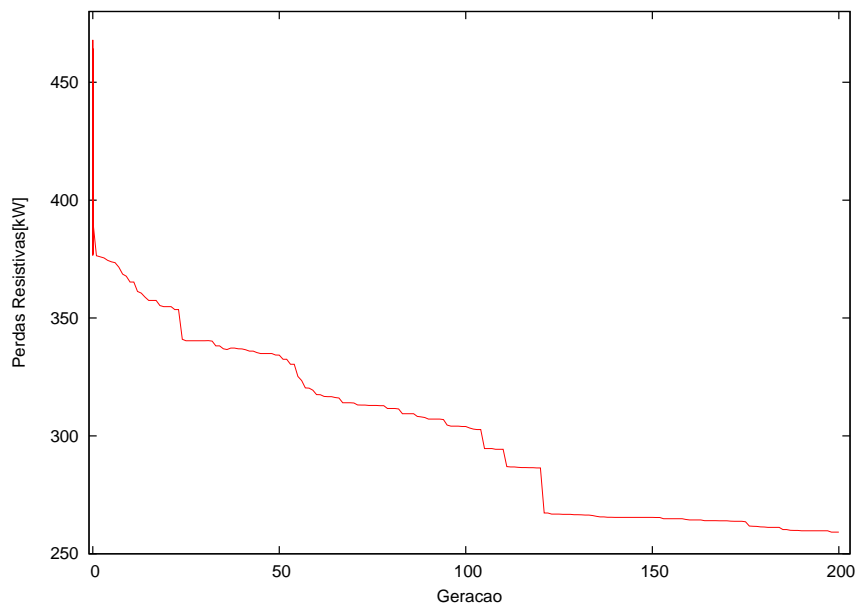


Figura 9.6: Redução das perdas por geração (múltiplas faltas).

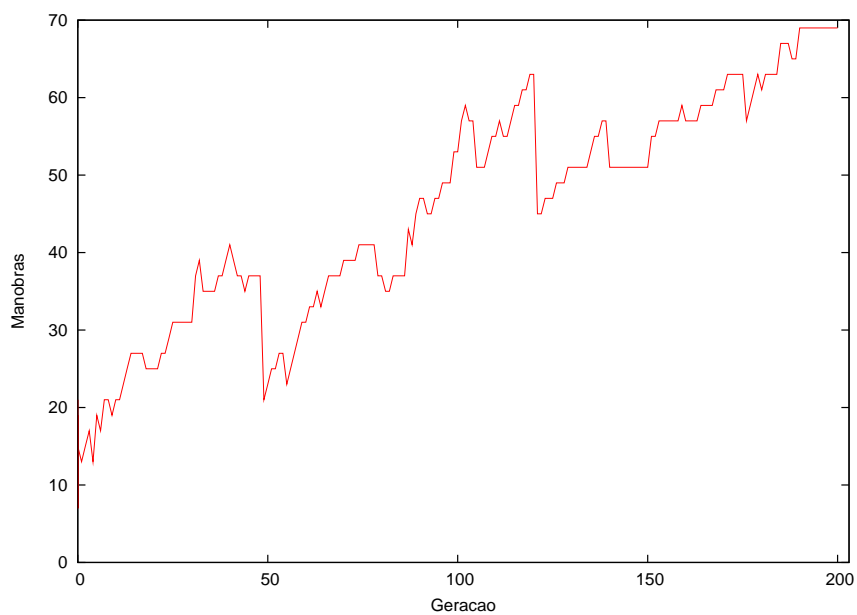


Figura 9.7: Aumento das manobras por geração (múltiplas faltas).

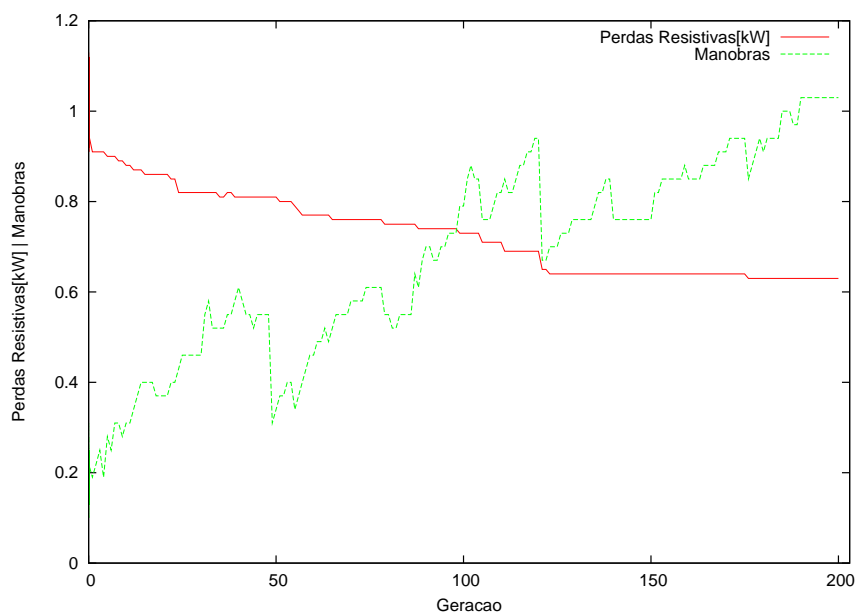


Figura 9.8: Manobras e perdas resistivas linearizadas (múltiplas faltas).

Análise Comparativa

Testes foram realizados utilizando o programa AERT, aplicando as faltas nos setores 3668, 3542 e 1031. A Tabela 9.4 mostra um comparativo entre os dois programas (AERT e PRN).

Tabela 9.4: Comparativo entre os métodos AERT e PRN em múltiplas faltas.

| | AERT | | PRN | |
|-----------------------------|--------------|---------------|-------------|---------------|
| | Média | Desvio Padrão | Média | Desvio Padrão |
| Perdas Resistivas [kW] | 291,27 | 15,29 | 347,59 | 27,86 |
| Queda de Tensão [%] | 4,33 | 1,69 | 3,31 | 0,16 |
| Carregamento da Rede [%] | 90,84 | 12,54 | 92,63 | 8,26 |
| Carregamento do Sistema [%] | 59,76 | 6,12 | 54,52 | 3,38 |
| Manobras | 30,90 | 4,33 | 18,7 | 4,32 |
| Tempo [ms] | 2815,50 | 260,35 | 3230,50 | 159,92 |

Analisando a Tabela 9.4 nota-se que o PRN conseguiu uma maior redução no número de manobras, consequentemente, o AERT obteve um valor de perdas resistivas. Os limites operacionais foram obedecidos por ambos os programas.

9.3 Resultados das Simulações no SDR Real de São Carlos Duplicado

Objetivando verificar o quesito tempo de processamento computacional e analisar o comportamento do PRN em SDRs maiores daquele utilizado até agora, o SDR da cidade de São Carlos/SP foi duplicado. O SDR duplicado de São Carlos/SP é composto de: 7720 barras, 1270 chaves, 6 subestações com 4 transformadores com potência de 50MVA e 2 transformadores de 25MVA e, 46 alimentadores.

Os objetivos e as penalidades na função agregação são os mesmos considerados no PRN nas simulações anteriores. Os parâmetros utilizados para análise do PRN, são:

1. Número máximo de gerações: $G_{max} = 250$;
2. Número de indivíduos nas populações P e Q : $N_{ind} = 110$.

Para o AERT também foram considerados os mesmos objetivos e as penalidades na função agregação utilizados nas simulações anteriores, porém foram definidos os seguintes parâmetros: 15000 indivíduos ($G_{max} = 15000$), e os melhores indivíduos de uma população (de tamanho 20) são mantidos.

A seguir serão considerados testes com múltiplas faltas localizadas nos setores 182, 486, 504 do SDR, interrompendo assim o serviço de fornecimento de energia elétrica. Logo, a primeira topologia da rede, após isolar os setores defeituosos, possui as seguintes características:

- Perdas resistivas = 728,97kW;
- Carregamento da rede = 83,15%;
- Carregamento do Sistema = 44,64%;
- Queda de tensão = 4,72%.

A Figura 9.9 apresentada apenas as configurações da primeira Fronteira de Pareto. As configurações mostradas na Figura 9.9 são apresentadas na Tabela 9.5.

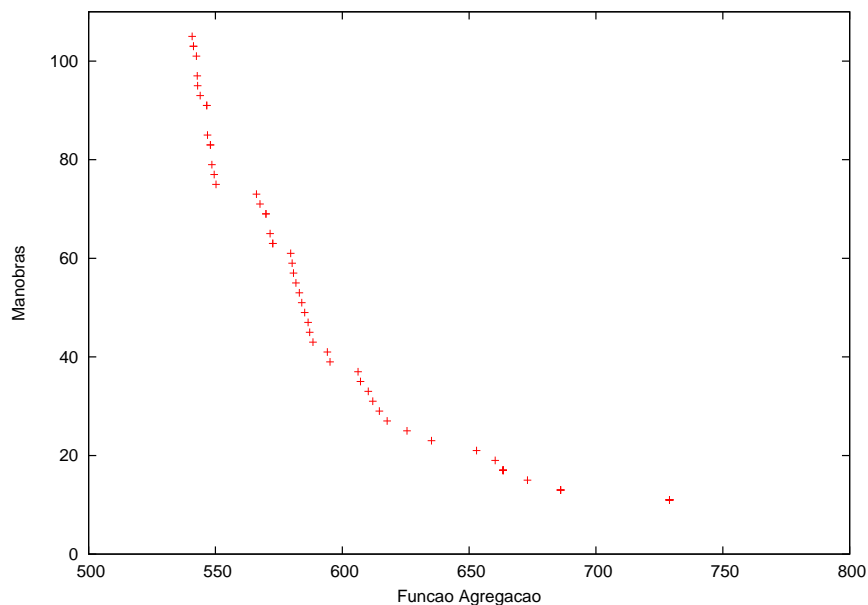


Figura 9.9: Primeira Fronteira de Pareto após as múltiplas faltas no SDR duplicado.

Tabela 9.5: Configurações da primeira Fronteira de Pareto após múltiplas faltas no SDR duplicado

| Função Agregação | Perdas Resistivas [kW] | Queda de Tensão [%] | Carregamento da Rede [%] | Carregamento do Sistema [%] | Manobras |
|------------------|------------------------|---------------------|--------------------------|-----------------------------|----------|
| 728,97 | 728,97 | 4,72 | 83,15 | 44,64 | 11 |
| 728,97 | 728,97 | 4,72 | 83,15 | 55,07 | 11 |
| 728,97 | 728,97 | 4,72 | 83,15 | 55,07 | 11 |
| 728,97 | 728,97 | 4,72 | 83,15 | 55,07 | 11 |
| 686,05 | 686,05 | 3,33 | 83,15 | 55,07 | 13 |
| 728,97 | 728,97 | 4,72 | 83,15 | 55,39 | 11 |
| 686,05 | 686,05 | 3,33 | 83,15 | 54,68 | 13 |
| 673,02 | 673,02 | 3,33 | 77,63 | 55,07 | 15 |
| 686,05 | 686,05 | 3,33 | 83,15 | 55,07 | 13 |
| 728,97 | 728,97 | 4,72 | 83,15 | 55,07 | 11 |
| 728,97 | 728,97 | 4,72 | 83,15 | 55,07 | 11 |
| 686,05 | 686,05 | 3,33 | 83,15 | 55,07 | 13 |
| 663,38 | 663,38 | 3,33 | 77,63 | 54,02 | 17 |
| 617,69 | 617,69 | 3,33 | 88,00 | 58,43 | 27 |
| 663,38 | 663,38 | 3,33 | 77,63 | 55,07 | 17 |
| 663,38 | 663,38 | 3,33 | 77,63 | 55,07 | 17 |
| 660,29 | 660,29 | 3,33 | 77,63 | 56,30 | 19 |
| 583,05 | 583,05 | 3,25 | 86,52 | 60,81 | 53 |
| 610,27 | 610,27 | 3,33 | 85,14 | 69,49 | 33 |
| 586,52 | 586,52 | 3,25 | 89,15 | 55,07 | 47 |
| 652,91 | 652,91 | 3,33 | 77,63 | 54,02 | 21 |
| 607,18 | 607,18 | 3,33 | 85,14 | 55,07 | 35 |
| 567,57 | 567,57 | 3,25 | 78,21 | 55,17 | 71 |
| 548,03 | 548,03 | 3,25 | 66,51 | 55,07 | 83 |
| 548,03 | 548,03 | 3,25 | 66,51 | 54,02 | 83 |
| 635,15 | 635,15 | 3,33 | 77,63 | 59,99 | 23 |
| 566,17 | 566,17 | 3,25 | 74,24 | 55,07 | 73 |
| 542,51 | 542,51 | 3,25 | 77,18 | 55,07 | 101 |
| 625,50 | 625,50 | 3,33 | 77,63 | 55,34 | 25 |

Tabela 9.5: Configurações da primeira Fronteira de Pareto após múltiplas faltas no SDR duplicado(continuação).

| Função Agregação | Perdas Resistivas [kW] | Queda de Tensão [%] | Carregamento da Rede [%] | Carregamento do Sistema [%] | Manobras |
|------------------|------------------------|---------------------|--------------------------|-----------------------------|----------|
| 584,01 | 584,01 | 3,25 | 89,15 | 55,39 | 51 |
| 580,77 | 580,77 | 3,25 | 84,34 | 55,70 | 57 |
| 595,18 | 595,18 | 3,25 | 89,15 | 55,07 | 39 |
| 572,61 | 572,61 | 3,25 | 79,11 | 55,07 | 63 |
| 542,99 | 542,99 | 3,25 | 66,95 | 53,33 | 95 |
| 546,60 | 546,60 | 3,25 | 66,95 | 53,33 | 91 |
| 548,60 | 548,60 | 3,25 | 78,21 | 55,07 | 79 |
| 663,38 | 663,38 | 3,33 | 77,63 | 55,07 | 17 |
| 614,59 | 614,59 | 3,33 | 77,63 | 59,14 | 29 |
| 549,50 | 549,50 | 3,25 | 78,21 | 55,07 | 77 |
| 588,44 | 588,44 | 3,25 | 89,15 | 54,02 | 43 |
| 594,11 | 594,11 | 3,25 | 89,15 | 55,07 | 41 |
| 612,02 | 612,02 | 3,33 | 88,00 | 55,62 | 31 |
| 580,22 | 580,22 | 3,25 | 84,34 | 55,07 | 59 |
| 585,18 | 585,18 | 3,25 | 89,15 | 55,39 | 49 |
| 587,16 | 587,16 | 3,25 | 89,15 | 54,68 | 45 |
| 569,92 | 569,92 | 3,25 | 78,21 | 54,02 | 69 |
| 606,20 | 606,20 | 3,33 | 85,14 | 55,07 | 37 |
| 581,72 | 581,73 | 3,25 | 86,09 | 55,47 | 55 |
| 541,32 | 541,32 | 3,25 | 77,18 | 71,00 | 103 |
| 663,38 | 663,38 | 3,33 | 77,63 | 55,07 | 17 |
| 579,64 | 579,64 | 3,25 | 84,34 | 54,68 | 61 |
| 569,92 | 569,92 | 3,25 | 78,21 | 54,02 | 69 |
| 572,61 | 572,61 | 3,25 | 79,11 | 59,19 | 63 |
| 540,82 | 540,82 | 3,25 | 77,18 | 55,07 | 105 |
| 546,88 | 546,88 | 3,25 | 66,51 | 55,07 | 85 |
| 543,99 | 543,99 | 3,25 | 66,95 | 55,07 | 93 |
| 541,32 | 541,32 | 3,25 | 77,18 | 71,30 | 103 |
| 542,85 | 542,85 | 3,25 | 66,95 | 55,07 | 97 |
| 550,22 | 550,22 | 3,25 | 78,21 | 59,21 | 75 |
| 571,55 | 571,55 | 3,25 | 79,11 | 59,19 | 65 |
| 546,60 | 546,60 | 3,25 | 66,95 | 55,07 | 91 |
| 663,38 | 663,38 | 3,33 | 77,63 | 55,07 | 17 |

Analisando a Tabela 9.5 percebe-se que o valor da função agregação é igual ao de perdas resistivas em todas as configurações, isso ocorre pelo fato de as configurações encontradas não violarem os limites operacionais dos equipamentos. Considerando como melhor configuração aquela que apresentar a menor função agregação, a Figura 9.10 mostra a redução total de perdas resistivas para a melhor configuração em cada geração. Para essas mesmas configurações, a Figura 9.11 mostra as manobras e na Figura 9.12 são mostrados os valores linearizados do número de manobras e o de perdas de cada configuração. Estes resultados foram obtidos utilizando semente 0, para o gerador de número aleatórios, e o tempo de processamento foi de 4530ms.

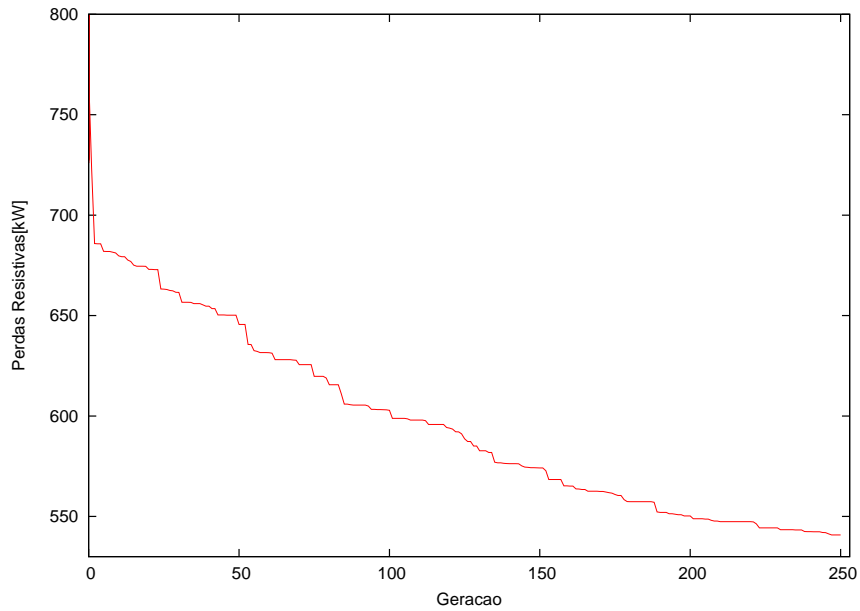


Figura 9.10: Redução das perdas por geração para múltiplas faltas no SDR duplicado.

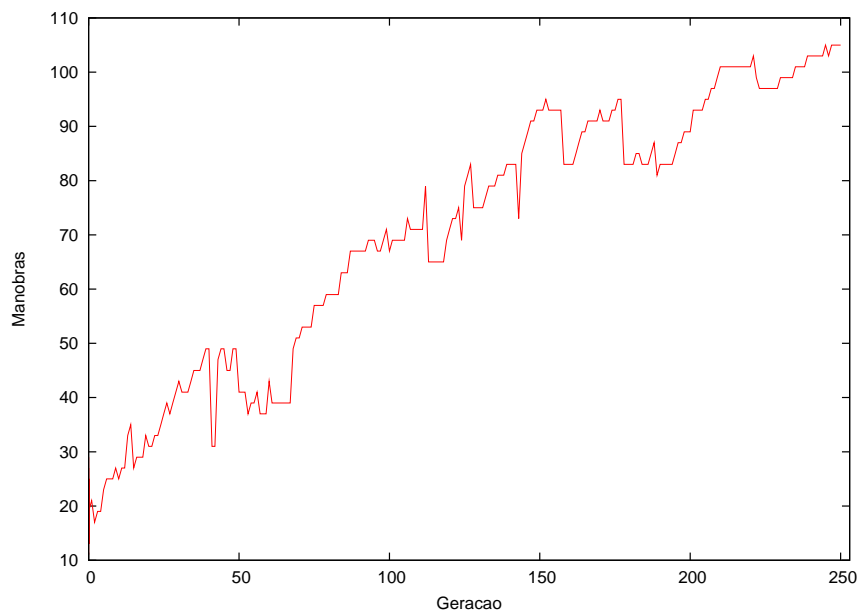


Figura 9.11: Redução das manobras por geração para múltiplas faltas no SDR duplicado.

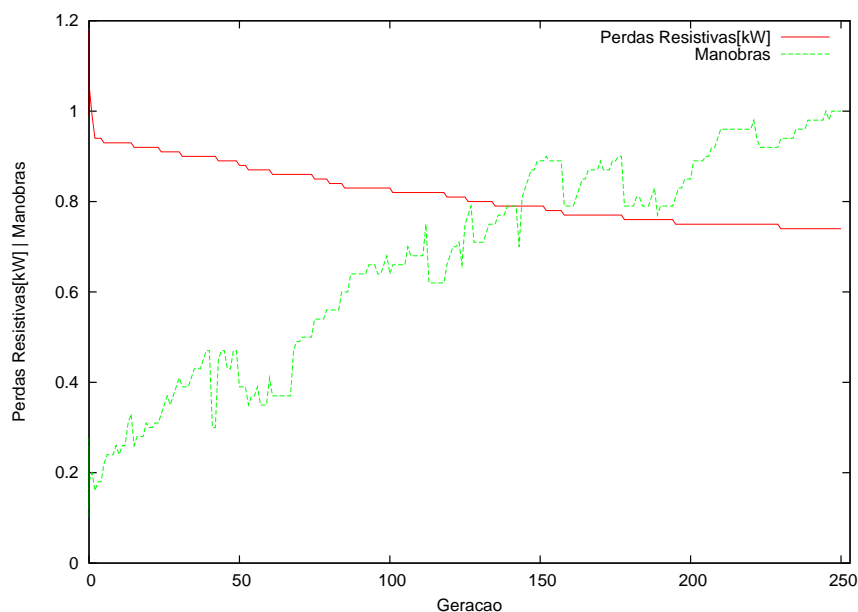


Figura 9.12: Manobras e perdas resistivas linearizadas para múltiplas faltas no SDR duplicado.

Análise Comparativa

Testes foram realizados utilizando o programa AERT, aplicando as faltas nos setores 182, 486 e 504. A Tabela 9.6 mostra um comparativo entre os dois programas (AERT e PRN).

Tabela 9.6: Comparativo entre os métodos AERT e PRN em múltiplas faltas.

| | AERT | | PRN | |
|-----------------------------|--------------|---------------|--------------|---------------|
| | Média | Desvio Padrão | Média | Desvio Padrão |
| Perdas Resistivas [kW] | 552,58 | 17,23 | 634,36 | 27,55 |
| Queda de Tensão [%] | 4,10 | 1,57 | 3,52 | 0,53 |
| Carregamento da Rede [%] | 83,95 | 6,81 | 88,21 | 9,15 |
| Carregamento do Sistema [%] | 62,98 | 7,72 | 55,22 | 1,97 |
| Manobras | 36,20 | 5,71 | 24,10 | 7,91 |
| Tempo [ms] | 3307,00 | 334,81 | 4229,50 | 217,29 |

Analisando a Tabela 9.6, nota-se novamente que o PRN conseguiu uma maior redução no número de manobras, conseqüentemente, o AERT obteve um valor de perdas resistivas menor. Os limites operacionais em ambos os programas não foram violados. No quesito tempo de processamento computacional, nota-se que é necessário em média, aproximadamente, 4230ms para o PRN encontrar uma Fronteira de Pareto que contenha configurações factíveis que não violem os limites operacionais, no AERT este tempo é

inferior. Porém vale destacar que no AERT não é fornecida uma Fronteira de Pareto que contenha diversas configurações factíveis.

9.4 Comentários Adicionais

O PRN mostrou-se uma ferramenta capaz de lidar com redes de grande porte, com tempo de processamento reduzido. A média de manobras encontradas nas 20 execuções é baixa. Observa-se também que mesmo em problemas com número de barras elevado, o algoritmo não incorreu em problema de explosão combinatória, como acontece com as ferramentas envolvendo PM.

Cada uma das populações P e Q contém 110 indivíduos. Ao final do processo, os melhores indivíduos de PUQ são apresentados como configurações, estas se encontram na primeira Fronteira de Pareto, cabendo assim ao operador do sistema escolher a configuração mais adequada a se utilizar. Esta escolha pode envolver o número de manobras, a queda de tensão, a redução de perdas resistivas, além de outras possibilidades.

Capítulo 10

Conclusões

O foco deste trabalho foi o problema de restabelecimento de energia em SDR, que se trata de um problema com múltiplos objetivos, normalmente conflitantes¹, com funções não lineares e descontínuas.

Dentre os objetivos de um PRE têm-se: (i) Reduzir o número de consumidores interrompidos (ou nenhum), e (ii) minimizar o número de manobras; que devem ser atendidos sem desrespeitar os limites operacionais dos equipamentos. Conseqüentemente, a obtenção de PRE em SDR é um problema com múltiplos objetivos, alguns conflitantes.

A natureza das funções utilizadas no problema de restabelecimento de energia, em SDR, inviabiliza a utilização de técnicas de programação matemática, pois as mesmas ficam sujeitas a explosão combinatória. Assim, conforme mostrado no Capítulo 2, nas últimas décadas diversas técnicas baseadas em AEs foram propostas para o tratamento do problema em pauta. Embora essas técnicas tenham apresentado resultados satisfatórios, as mesmas produzem muitas soluções não factíveis, quando aplicadas em SDR considerados de grande porte, restringindo assim as suas aplicações.

No Capítulo 7 apresentou-se um AE que pode ser aplicado mesmo em SDR de grande porte, proposto em (Santos, Delbem e Bretas, 2008b). O grande diferencial desse AE é a utilização da RNP e de seus operadores, que possibilitam a geração apenas de configurações factíveis. Dessa forma, aumenta significativamente a eficiência da busca

¹Por exemplo, para reduzir perdas resistivas é necessário um aumento do número de operações de manobra.

por melhores soluções (configurações) do AE, pois diminui a possibilidade de geração de soluções não factíveis. Esse AE faz uso do método de tabelas para lidar com problemas multi-objetivo.

Neste trabalho desenvolveu-se um novo AE para obtenção de planos de restabelecimento de energia em SDR, que foi chamado de NS2R (NSGA-II com RNP), que se baseia numa versão modificada do NSGA-II juntamente com a RNP e seus operadores.

Optou-se por utilizar como base o NSGA-II, em razão de o mesmo ser a técnica de MOEA que tem apresentado melhor desempenho para o tratamento do problema de restabelecimento de energia em SDR. Vale destacar ainda que foram propostas algumas alterações no NSGA-II. Primeiro em razão da utilização da RNP e de seus operadores e segundo porque o NSGA-II não é eficiente para solução de problemas com mais de três objetivos.

O NS2R modela o problema considerando duas funções objetivo. A primeira é o número de operações de chaveamento, que deve ser minimizado, e a segunda é uma função agregação que inclui as demais restrições do problema. Ao final de sua execução, o NS2R gera várias fronteiras de pareto, porém, apresenta apenas a primeira fronteira, em razão de a mesma conter as melhores soluções. Assim, cabe ao especialista do sistema escolher a solução (configuração) que melhor atenda a sua necessidade.

O NS2R foi implementado computacionalmente dando origem ao PRN (Programa para obtenção de planos de Restabelecimento de energia em SDRs utilizando o NS2R).

Para validar o PRN, o mesmo foi aplicado a dois SDR: o primeiro corresponde à cidade de São Carlos-SP, que é composto por 3860 barras e 635 chaves (que pode ser considerado como um sistema de grande porte); e o segundo corresponde ao SDR da cidade de São Carlos duplicado. Importa destacar que nos testes realizados foram consideradas todas as linhas, barras e chaves desse sistema.

Realizaram-se testes envolvendo faltas únicas e múltiplas e o tempo de processamento médio necessário para encontrar configurações factíveis via o PRN foi de aproximadamente 3 segundos. Por se tratar de uma técnica que analisa vários objetivos, pode-se considerar como um tempo bastante pequeno.

Apresentou-se, também, um comparativo entre o programa proposto (PRN) e o AERT, proposto em (Santos, Delbem e Bretas, 2008b), que faz uso de um AE mono-

objetivo juntamente com o método das tabelas. Os resultados de alguns testes indicam que o PRN possibilita uma redução do número de manobras, porém com um aumento de perdas. De uma forma mais geral, pode-se dizer que, em razão de basear-se no NSGA-II, o PRN apresenta um melhor mapeamento do problema, em razão de possibilitar a obtenção das Fronteiras de Pareto.

O tempo de processamento de ambos os programas é bem próximo, isto em razão de o PRN exigir um número menor de gerações para obtenção de boas soluções.

10.1 Publicações

1. A. C. Santos, M. Nanni, **M. R. Mansour**, A. C. B. Delbem, J. B. A. London Jr., N. G. Bretas. “A Power Flow Method Computationally Efficient For Large-Scale Distribution Systems”, 2008 *IEEE PES Latin America Transmission and Distribution Conference and Exposition*, Bogota-Colombia, Aug. 2008.
2. **M. R. Mansour**; A. C. Santos.; J. B. A. London Jr.; A. C. B. Delbem; N. G. Bretas. “Aplicação de Conjuntos Fuzzy para Definição do Critério de Parada de um Algoritmo Evolutivo Desenvolvido para Reconfiguração de Redes de Distribuição”, Congresso da Academia Trinacional das Ciências - C3N, Out., 2008.
3. **M. R. Mansour**, A. C. Santos.; J. B. A. London Jr.; A. C. B. Delbem; N. G. Bretas. “Aplicação de Conjuntos Fuzzy para Avaliação das Funções Multi-Objetivo de um Algoritmo Evolutivo para a Reconfiguração de Sistemas de Distribuição de Energia Elétrica”, Simpósio Brasileiro de Pesquisa Operacional - XL SBPO, João Pessoa-PB, Set., 2008.
4. **M. R. Mansour**; A. C. Santos.; J. B. A. London Jr.; A. C. B. Delbem; N. G. Bretas. “Energy Restoration in Distribution Systems using Multi-Objective Evolutionary Algorithm and an Efficient Data Structure”, PowerTech, Bucharest, Romania, 2009.

Referências Bibliográficas

- Abuali, F. N., Wainwright, R. L. and Schoenefeld, D. A. (1995). Determinant factorization: A new encoding scheme for spanning trees applied to the probabilistic minimum spanning tree problem, In Eschelmann, L. (Ed.), Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann, pp. 470–477.
- Amabis, J. M. and Martho, G. R. (1985). Curso Básico de Biologia, Editora Moderna Ltda., São Paulo.
- Andrew, A. M. (2004). Introduction to evolutionary computing, by a. e. eiben and j. e. smith (natural computing series), springer, berlin, 2003, hardback, xv + 299 pp., isbn 3-540-40184-9, Robotica **22**(3): 349–349.
- ANEEL (2008). Procedimentos de distribuição de energia elétrica no sistema elétrico nacional - prodist, Qualidade de energia elétrica: Agência nacional de energia elétrica-aneel.
- Aoki, K., Nara, K., Itoh, M., Satoh, T. and Kuwabara, H. (1989). A new algorithm for service restoration in distribution systems, Power Delivery, IEEE Transactions on **4**(3): 1832–1839.
- Aoki, K., Nara, K. and Satoh, T. (1989). New configuration algorithm for distribution system - priority constrained emergency service restoration, Proceedings of IFAC Conference on Power Systems and Power Plant Control, pp. 443–448.
- Arciszewski, T. and Jong, K. A. D. (2001). Evolutionary computation in civil engineering: research frontiers, Civil and structural engineering computing: 2001 pp. 161–184.

- Augugliaro, A., Dusonchet, L. and Sanseverino, E. R. (2000). Multiobjective service restoration in distribution networks using an evolutionary approach and fuzzy sets, International Journal of Electrical Power & Energy Systems **22**: 103–110.
- Ballard, D. H. (1999). An Introduction to Natural Computation, MIT Press, Cambridge, MA, USA.
- Baran, M. and Wu, F. (1989). Optimal sizing of capacitors placed on a radial distribution system, Power Delivery, IEEE Transactions on **4**(1): 735–743.
- Brandini, A. C. (2000). Análise crítica de algoritmos de fluxo de carga usados em sistemas de distribuição radial, Dissertação de Mestrado, FEIS-UNESP.
- Carvalho, P., Ferreira, L. and Barruncho, L. (2001). On spanning-tree recombination in evolutionary large-scale network problems - application to electrical distribution planning, Evolutionary Computation, IEEE Transactions on **5**(6): 623–630.
- Castro, R. E. (2001). Otimização de Estruturas com Multi-objetivos via Algoritmos Genéticos de Pareto, Tese de Doutorado, COPPE/UFRJ.
- Cespedes, R. (1990). New method for the analysis of distribution networks, Power Delivery, IEEE Transactions on **5**(1): 391–396.
- Chen, C., Lin, C., Wu, C. and Kang, M. (2000). Feeder reconfiguration for distribution system contingencies by object oriented programming, Power Engineering Society Summer Meeting, 2000. IEEE **1**: 431–436 vol. 1.
- Chen, C. S., Wu, J. S. and Moo, C. S. (1989). Fault restoration by optimizing switch configuration in distribution systems, J. Chin. Inst. Eng. **12**: 781–789.
- Chen, T.-H., Chen, M.-S., Hwang, K.-J., Kotas, P. and Chebli, E. (1991). Distribution system power flow analysis-a rigid approach, Power Delivery, IEEE Transactions on **6**(3): 1146–1152.
- Coello, C., Veldhuizen, D. and Lamont, G. (2002). Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, New York.
- Cormen, T. H. (2002). Algoritmos: Teoria e Prática, Campus.

- Corne, D., Knowles, J. D. and 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.
- Corne, D. W., Jerram, N. R., Knowles, J. D. and J, M. (2001). Pesa-ii: Region-based selection in evolutionary multiobjective optimization, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001), Morgan Kaufmann Publishers, pp. 283–290.
- Curcic, S., Özveren, C. S., Crowe, L. and Lo, P. K. L. (1996). Electric power distribution network restoration: a survey of papers and a review of the restoration problem, Electric power systems research **35**: 73–86.
- Das, D., Nagi, H. S. and Kothari, D. P. (1994). Novel method for solving radial distribution networks, Generation, Transmission and Distribution, IEE Proceedings **141**(4): 291–298.
- de Lima, T. W. and Delbem, A. C. B. (2007). Estruturas de dados eficientes para algoritmos evolutivos aplicados ao projeto de redes, Relatório Técnico 301. Notas Didáticas do ICMC-USP.
- Deb, K. (2001). Multi-objective optimization using evolutionary algorithms, Wiley, New York.
- Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii, Springer, pp. 849–858.
- Deb, K., Mohan, M. and Mishra, S. (2005). Evaluating the e-domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions, Evol. Comput. **13**(4): 501–525.
- Deb, K. and Sundar, J. (2006a). Reference point based multi-objective optimization using evolutionary algorithms, GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM, New York, NY, USA, pp. 635–642.

- Deb, K. and Sundar, J. (2006b). Reference point based multi-objective optimization using evolutionary algorithms, GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM, New York, NY, USA, pp. 635–642.
- Delbem, A. C. B., de Carvalho, A. C. P. L. F., Policastro, C. A., Pinto, A. K. O., Honda, K. and Garcia, A. C. (2004). Node-depth encoding for evolutionary algorithms applied to network design, GECCO (1), pp. 678–687.
- Delbem, A., de Carvalho, A. and Bretas, N. (2005). Main chain representation for evolutionary algorithms applied to distribution system reconfiguration, Power Systems, IEEE Transactions on **20**(1): 425–436.
- Devi, S., Gupta, D. P. S. and Sargunraj, S. (1990). A search technique for restoring power supply in complex distribution systems, In power systems for the year 2000 and beyond. Proc. 6th Nat. Power Sys. Conf., McGraw-Hill, New Delhi, pp. 122–125.
- Devi, S., Sen Gupta, D. and Sargunraj, S. (1991). Optimal restoration of supply following a fault on large distribution systems, Advances in Power System Control, Operation and Management, 1991. APSCOM-91., 1991 International Conference on pp. 508–513 vol.2.
- Dialynas, E. N. and Michos, D. G. (1989). Interactive modeling of supply restoration procedures in distribution system operation, Power Engineering Review, IEEE **9**(7): 71–71.
- dos Santos, A. C. (2004). Restabelecimento de energia considerando todas as barras e chaves de um sistemas de distribuição real, Dissertação de Mestrado, EESC/USP.
- dos Santos, A. C. (2008). Algoritmo evolutivo computacionalmente eficiente para reconfiguração de sistemas de distribuição. Texto do exame de qualificação de doutorado em Engenharia Elétrica/USP/EESC.
- Fonseca, C. and Fleming, P. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kauffman Publishers, San Mateo - California, pp. 416–423.

- Fujii, Y., Miura, A., Miura, A., Tsukamoto, J., Youssef, M. G. and Noguchi, Y. (1992). On-line expert system for power distribution system control, International Journal of Electrical Power & Energy Systems **14**: 45–53.
- Gabriel, P. H. R. and Delbem, A. C. B. (2008). Fundamentos de algoritmos evolutivos, Relatório técnico. Notas Didáticas do ICMC-USP, 75.
- Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Gomes, F., Carneiro, S., J., Pereira, J., Vinagre, M., Garcia, P., Oliveira, E. and Araujo, L. (2005). A new distribution system reconfiguration approach using optimal power flow technique and sensitivity analysis for loss reduction, Power Engineering Society General Meeting, 2005. IEEE pp. 897–901 Vol. 1.
- Gradshteyn, I. S. and Ryzhik, I. M. (2000). Tables of Integrals, Series, and Products, Academic Press, San Diego.
- Gross, J. L. and Yellen, J. (2004). Handbook of Graph Theory, CRC Press.
- Haimes, Y., Lasdon, L. and Wismer, D. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization, Systems, Man and Cybernetics, IEEE Transactions on **1**(3): 296–297.
- Hajela, P. and Lin, C. Y. (1992). Genetic search strategies in multicriterion optimal design, Structural Optimisation **4**: 99–107.
- Harik, G. R., Lobo, F. G. and Sastry, K. (2006). Linkage Learning via Probabilistic Modeling in the Extended Compact Genetic Algorithm (ECGA).
- Hayes-Roth, F. (1975). Review of "adaptation in natural and artificial systems by john h. holland", the u. of michigan press, 1975, SIGART Bull. (53): 15–15.
- Horn, J., Horn, J., Nafpliotis, N., Nafpliotis, N., Goldberg, D. E. and Goldberg, D. E. (1994). A niched pareto genetic algorithm for multiobjective optimization, In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, pp. 82–87.

- Hsiao, Y. T. and Chen, C. Y. (2000). Enhancement of restoration service in distribution systems using a combination fuzzy-ga method, IEEE Transactions on Power Systems **15**: 1394–1400.
- Hsu, Y.-Y., Huang, H.-M., Kuo, H.-C., Peng, S., Chang, C., Chang, K., Yu, H., Chow, C. and Kuo, R. (1992). Distribution system service restoration using a heuristic search approach, Power Delivery, IEEE Transactions on **7**(2): 734–740.
- Inagaki, J., Nakajima, J. and Haseyama, M. (2006). A multiobjective service restoration method for power distribution systems, Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on pp. 4 pp.–.
- Jensen, M. (2003). Reducing the run-time complexity of multiobjective eas: The nsga-ii and other algorithms, Evolutionary Computation, IEEE Transactions on **7**(5): 503–515.
- Julstrom, B. A. (1994). Codings and operators in two genetic algorithms for the leaf-constrained minimum spanning tree problem, International Journal of Applied Mathematics and Computer Science **3**(14): 385–396.
- Kagan, N. (1999). Configuração de Redes de Distribuição através de Algoritmos Genéticos e Tomada de Decisão Fuzzy, Tese de Livre Docência, EP-USP.
- Kagan, N., de Oliveira, C. C. B. and Robba, E. J. (2005). Introdução aos Sistemas de Distribuição de Energia Elétrica, Edgard Blücher, São Paulo.
- Kagan, N. and Oliveira, C. C. B. (1996). Reconfiguração de sistema de distribuição de energia elétrica através de ferramenta para solução de problemas de decisão com múltiplos objetivos e incertezas., CONGRESSO BRASILEIRO DE AUTOMÁTICA (11): 268–276.
- Kendrew, T. J. and Graham, J. (1989). Applying expert system technology to automated distribution feeder deployment and sectionalizing, American Power Conj., USA, pp. 563–568.
- Kim, H., Ko, Y. and Jung, K.-H. (1992). Algorithm of transferring the load of the faulted substation transformer using the best-first search method, Power Delivery, IEEE Transactions on **7**(3): 1434–1442.

- Kita, H., Yabumoto, Y., Mori, N. and Nishikawa, Y. (1996). Multi-objective optimization by means of the thermodynamical genetic algorithm, PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, London, UK, pp. 504–512.
- Knowles, J. and Corne, D. (1999). The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimization, Congress on Evolutionary Computation, IEEE Service Center, Washington, pp. 98–105.
- Koza, J. R. (1989). Hierarchical genetic algorithms operating on populations of computer programs, in N. S. Sridharan (ed.), Proceedings of the Eleventh International Joint Conference on Artificial Intelligence IJCAI-89, Vol. 1, Morgan Kaufmann, pp. 768–774.
- Kumar, Y., Das, B. and Sharma, J. (2008). Multiobjective, multiconstraint service restoration of electric power distribution system with priority customers, Power Delivery, IEEE Transactions on **23**(1): 261–270.
- Lamont, G. B. and Veldhuizen, D. A. V. (2002). Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, Norwell, MA, USA.
- Laumanns, M., Rudolph, G. and paul Schwefel, H. (1998). A spatial predator-prey approach to multi-objective optimization: A preliminary study, Proceedings of the Parallel Problem Solving from Nature, Springer, pp. 241–249.
- Liu, C.-C., Lee, S. and Venkata, S. (1988). An expert system operational aid for restoration and loss reduction of distribution systems, Power Systems, IEEE Transactions on **3**(2): 619–626.
- Mantovani, J., Casari, F. and Romero, R. (2000). Reconfiguração de sistemas de distribuição radiais utilizando o critério de queda de tensão, SBA Controle e Automação **11**: 150–159.
- Michalewicz, Z. (1992). Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, New York.
- Miller, R. H. (1987). Operação de Sistemas de Potência, McGrawHill.

- Monticelli, A., Garcia, A. and Saavedra, O. (1990). Fast decoupled load flow: hypothesis, derivations, and testing, Power Systems, IEEE Transactions on **5**(4): 1425–1431.
- Monticelli, A. J. (1983). Fluxo de Carga em Redes de Energia Elétrica, Edgard Blücher, São Paulo.
- Morelato, A. L. and Monticelli, A. (1989). Heuristic search approach to distribution system restoration, Power Engineering Review, IEEE **9**(10): 65–66.
- Nahman, J. and Strbac, G. (1994). A new algorithm for service restoration in large-scale urban distribution systems, Electric power systems research **29**: 181–192.
- Okuda, K., Watanabe, H., Wang, F., Yamazaki, K. and Baba, T. (1988). An application of knowledge engineering for fault restoration operation in secondary power systems, Electr. Eng. Jpn. **108**: 51–59.
- Palmer, C. C. (1994). An approach to a problem in network design using genetic algorithms, Tese de Doutorado, Brooklyn, NY, USA.
- Pelikan, M. (2005). Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms, Studies in Fuzziness and Soft Computing, 1 edn, Springer.
- Pelikan, M., Goldberg, D. E. and Cantú-paz, E. E. (2000). Linkage problem, distribution estimation, and bayesian networks, Evol. Comput. **8**(3): 311–340.
- Picciotto, S. (1999). How to Encode a Tree, Tese de Doutorado.
- Prüfer, H. (1918). Neuer beweis eines satzes ueber permutationen, Archiv für Mathematik und Physik, (27): 742–744.
- R. Benayoun, J. de Montgolfier, J. T. and Laritchev, O. (1971). Linear programming with multiple objective functions: Step method (stem), Mathematical Programming **1**: 366–375.
- R. Benayoun, J. de Montgolfier, J. T. and Laritchev, O. (1973). Linear programming with multiple objective functions: Step method (stem), Journal Mathematical Programming **1**(1): 366–375.

- Raidl, G. and Julstrom, B. (2003). Edge sets: an effective evolutionary coding of spanning trees, Evolutionary Computation, IEEE Transactions on **7**(3): 225–239.
- Raidl, G. R. (2000). An efficient evolutionary algorithm for the degree-constrained minimum spanning tree problem, Evolutionary Computation, 2000. Proceedings of the 2000 Congress on, Vol. 1, pp. 104–111 vol.1.
- Ridley, M. (1996). Evolution, Backwell Science, Cambridge, MA, USA.
- Rothlauf, F., Goldberg, D. E. and Heinzl, A. (2002). Network random keys: a tree representation scheme for genetic and evolutionary algorithms, Evolutionary Computation **10**.
- Rudolph, G. (2001). Evolutionary search under partially ordered fitness sets, In Proceedings of the International Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent Systems (ISI 2001, ICSC Academic Press, pp. 818–822.
- Sait, S. M. and Youssef, H. (1999). Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems, IEEE Computer Society Press, Los Alamitos, CA, USA.
- Santos, A. C. d., Delbem, A. C. B. and Bretas, N. G. (2008a). A multiobjective evolutionary algorithm with node-depth encoding for energy restoration, Natural Computation, 2008. ICNC '08. Fourth International Conference on **6**: 417–422.
- Santos, A. C., Nanni, M., Mansour, M. R., Delbem, A. C. B., London, J. B. A. and Bretas, N. G. (2008). A power flow method computationally efficient for large-scale distribution systems, Transmission and Distribution Conference and Exposition: Latin America, 2008 IEEE/PES pp. 1–6.
- Santos, A., Delbem, A. and Bretas, N. (2008b). Energy restoration for large-scale distribution system using ea and a new data structure, Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE pp. 1–8.
- Sarfi, R., Salama, M. and Chikhani, A. (1995). Distribution system reconfiguration for loss reduction: an algorithm based on network portioning theory, IEEE Power Industry Computer Application Conference pp. 503–509.

- Sarma, N. D. R., Prasad, V. C. and Pao, K. S. P. (1990). Network reconfiguration in distribution network for service restoration, In power systems for the year 2000 and beyond. Proc. 6th Nat. Power Sys. Conf., McGraw-Hill, pp. 131–135.
- Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms, Proceedings of the 1st International Conference on Genetic Algorithms, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, pp. 93–100.
- Shin, D. J., Kim, J. O., Kim, T. K., Choo, J. B. and Singh, C. (2004). Optimal service restoration and reconfiguration of network using genetic-tabu algorithm, Electric power systems research **20**: 422–436.
- Shirmohammadi, D. (1992). Service restoration in distribution networks via network reconfiguration, Power Delivery, IEEE Transactions on **7**(2): 952–958.
- Shirmohammadi, D., Hong, H., Semlyen, A. and Luo, G. (1988). A compensation-based power flow method for weakly meshed distribution and transmission networks, Power Systems, IEEE Transactions on **3**(2): 753–762.
- Sinclair, M. C. (1995). Minimum cost topology optimisation of the cost 239 european optical network, In, Springer-Verlag, pp. 26–29.
- SOAK, S.-M., CORNE, D. and AHN, B.-H. (2005). A new evolutionary algorithm for spanning-tree based communication network design, IEICE Trans Commun **E88-B**(10): 4090–4093.
- Srinivas, M. (2000). Distribution load flows: a brief review, Power Engineering Society Winter Meeting, 2000. IEEE **2**: 942–945 vol.2.
- Srinivas, N. and Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary Computation **2**(3): 221–248.
- Srinivasan, D., Liew, A., Chang, C. and Chen, J. (1994). Intelligent operation of distribution network, Generation, Transmission and Distribution, IEE Proceedings- **141**(2): 106–116.
- Strogatz, S. H. (2001). Exploring complex networks, Nature **410**: 268–276.
- Teo, C. Y. (1992). A computer aided system to automate the restoration of electrical power supply, Electric power systems research **24**: 119–125.

- Teuscher, C., Mange, D. and Tempesti, G. (2003). Bio-inspired computing tissues: Towards machines that evolve, Grow, and Learn, BioSystems **68**: 235–244.
- Ticona, W. G. C. and Delbem, A. C. B. (2008). Algoritmos evolutivos para otimização multi-objetivo, Relatório técnico. Notas Didáticas do ICMC-USP, 76.
- Toune, S., Fudo, H., Genji, T., Fukuyama, Y. and Nakanishi, Y. (2002). Comparative study of modern heuristic algorithms to service restoration in distribution systems, Power Delivery, IEEE Transactions on **17**(1): 173–181.
- Van Veldhuizen, D. A. (1999). Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations, Tese de Doutorado, Wright-Patterson AFB, OH.
- Wu, J., Tomsovic, K. and Chen, C. (1991). A heuristic search approach to feeder switching operations for overload, faults, unbalanced flow and maintenance, Power Delivery, IEEE Transactions on **6**(4): 1579–1586.
- Zhou, G. and Gen, M. (2003). A genetic algorithm approach on tree-like telecommunication network design problem, Operational Research Society **10**(53): 248–254.
- Zhou, G., Meng, Z., Cao, Z. and Cao, J. (2007). A new tree encoding for the degree-constrained spanning tree problem, CIS '07: Proceedings of the 2007 International Conference on Computational Intelligence and Security, IEEE Computer Society, Washington, DC, USA, pp. 85–90.
- Zitzler, E., Laumanns, M. and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm, Technical report.
- Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, Evolutionary Computation, IEEE Transactions on **3**(4): 257–271.