DISSERTAÇÃO DE MESTRADO Nº 834

RESTAURAÇÃO DE REDES DE ENERGIA UTILIZANDO ALGORITMOS GENÉTICOS MULTIOBJETIVO

Gisele Pinheiro da Silva

DATA DA DEFESA: 05/08/2014

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

RESTAURAÇÃO DE REDES DE ENERGIA UTILIZANDO ALGORITMOS GENÉTICOS MULTIOBJETIVO

Gisele Pinheiro da Silva

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Eduardo Gontijo Carrano

Belo Horizonte - MG

Agosto de 2014

"Restauração de Redes de Energia Utilizando Algoritmos Genéticos Multiobjetivo"

Gisele Pinheiro da Silva

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 05 de agosto de 2014.

Por:

Prof. Dr. Eduardo Gontijo Carrano
DEE (UFMG) - Orientador

Prof. Dr. Mauro de Oliveira Prates
DEE (UFV)

Prof. Dr. João Antônio de Vasconcelos
DEE (UFMG)

Prof. Dr. Lucas de Souza Batista
DEE (UFMG)

Gisele Pinheiro da Silva

Restauração de Redes de Energia Utilizando Algoritmos Genéticos Multiobjetivo

Belo Horizonte
05 de Agosto de 2014

Gisele Pinheiro da Silva

Restauração de Redes de Energia Utilizando Algoritmos Genéticos Multiobjetivo

Dissertação submetida à banca examinadora designada pelo Colegiado do Programa de Pós Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do grau Mestre em Engenharia Elétrica.

Orientadores:
Eduardo Gontijo Carrano
Oriane Magela Neto *(In Memoriam)*

Co-orientador:
Mauro de Oliveira Prates

Esta dissertação foi desenvolvida como parte do projeto P&D 369, financiado pela Cemig D/ANEEL.

Belo Horizonte
05 de Agosto de 2014

AGRADECIMENTOS

À Deus, por ter me concedido a oportunidade e guiado meus passos nessa difícil etapa de minha vida dando-me serenidade, fé e força para revigorar-me em todos os momentos.

À minha família, especialmente aos meus pais, Alberto e Denise, e irmã, Natália, pelo carinho e, principalmente, pelo apoio em momentos difíceis sempre me motivando a seguir em frente. Ao meu marido, Felipe, por todo amor e compreensão às minhas ausências com resignação e incentivo.

Aos meus orientadores, Oriane Magela Neto e Eduardo Gontijo Carrano. Ao mestre Oriane, por todo aprendizado deixado. Ao Carrano, por toda disponibilidade, esforço e paciência.

À todos os meus amigos de biblioteca e de gole que foram essenciais nessa conquista.

A todos os responsáveis e membros do Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais (PPGEE/UFMG) e do Departamento de Engenharia Elétrica da Universidade Federal de Viçosa (DEL/UFV).

À Cemig D, pelo apoio incialmente financeiro, essencial durante o trabalho e posteriormente pelo incentivo e flexibilidade oferecida por parte dos membros da gerência de Planejamento Integrado do Sistema Elétrico da Distribuição (PE/PS). E a ANEEL pelo apoio financeiro.

A todos que, direta ou indiretamente, contribuíram para a conclusão deste estudo.

RESUMO

Neste trabalho é proposta uma nova estratégia para restauração de sistemas de distribuição de energia elétrica. Devido à natureza de suas funções objetivo e restrições, a restauração deve ser modelada como um problema de otimização não-linear multiobjetivo. Isso torna difícil a tarefa de encontrar soluções adequadas. Nesta proposta, um algoritmo genético multiobjetivo, Strenght Pareto Evolutionary Algorithm (SPEA2), foi implementado com o objetivo de realizar varreduras e gerar soluções eficientes e não repetidas. Para comprovar a eficiência da estratégia proposta, um sistema teste de 16 barras foi utilizado. O algoritmo foi, então, aplicado a dois sistemas reais de grande porte disponibilizados pela Cemig Distribuição, um com 703 barras e outro com 484 barras. O algoritmo gera como resultado indivíduos decodificados em várias soluções sequenciais. Uma novidade apresentada neste trabalho é que o algoritmo não só minimiza a carga desconectada ao fim do conjunto de manobras, mas também minimiza a carga desconectada ao fim de cada iteração, dessa forma o decisor terá certeza que a sequência de manobras apresentada é a que recuperará a maior quantidade de carga.

Palavras-chave— restauração de energia, algoritmos genéticos multiobjetivo, SPEA2.

ABSTRACT

A novel strategy is proposed in this work for power system restoration. Due to the nature of the objective functions and constraints, the restoration problem should be modelled as a non-linear multi-objective optimization problem. This makes it hard to find suitable solutions for the problem. In this proposal, a multi-objective genetic algorithm, Strenght Pareto Evolutionary Algorithm (SPEA2), was implemented with the goal of performing scans in order to generate efficient unique solutions. To prove the efficiency of the proposed strategy, one test system, with 16 buses, was considered. The algorithm creates as result a decoded individual with sequential solutions, always recovering a big amount of system load. The algorithm was then applied to two real large systems offered by Cemig Distribution, one with 703 buses and another with 484 buses. The algorithm generates as a result individuals decoded over sequential solutions. A novelty presented in this paper is that the algorithm not only minimizes the load disconnected at the end of the set of maneuvers, but also minimizes disconnected after each iteration load, thus the decision-maker will make sure that the sequence of maneuvers is presented that retrieve the largest amount of charge.

Keywords— power restoration, multiobjective genetic algorithmos, SPEA 2.

Lista de Figuras

2.1 – Exemplo de Grafo 8
2.2 – Exemplo de Árvore..... 9
2.3 – Estrutura de um Algoritmo Genético..... 17
2.4 – Estrutura Geral do SPEA2 24
3.1 – Arquitetura do Algoritmo em Camadas..... 27
3.2 – Sistema na Camada C..... 28
3.3 – Sistema na Camada B..... 29
3.4 – Fluxograma do Algoritmo Implementado..... 31
3.5 – Pseudocódigo da Decodificação..... 32
3.6 – Esquema Ilustrativo para o Sistema de 16 Blocos..... 33
3.7 – Exemplo de solução para o SD de 16 Blocos..... 35
3.8 – Esquema Ilustrativo de Fronteira Pareto de Dois Indivíduos 37
3.9 – Decodificação de Dois Indivíduos..... 38
3.10 – Exemplo de Decodificação das Cidades..... 40
3.11 – Exemplo de Permutação Infactível..... 40
3.12 – Exemplo de Procedimento de *Cycle Crossover*..... 41
3.13 – Exemplo de Mutação *Swap*..... 42
3. 14 – Exemplo de Mutação *Reverse*..... 43
4.1 – Esquema Ilustrativo para o Sistema de 16 Blocos..... 46
4.2 – Aproximação da Fronteira Pareto para o Sistema de 16 Blocos..... 46
4.3 – Exemplo de Solução para SD de 16 Blocos..... 47
4.4 – Esquema Ilustrativo para o Sistema Cemig A..... 48
4.5 – Aproximação da Fronteira Pareto para o Sistema Cemig A, exemplo 1..... 49
4.6 – Exemplo de Solução para o SD da C A, exemplo 1 – Primeiro Estágio..... 50
4.7 – Exemplo de Solução para o SD da Cemig A, exemplo 1 – Segundo Estágio..... 50
4.8 – Aproximação da Fronteira Pareto para o Sistema Cemig A, exemplo 2..... 51
4.9 – Exemplo de Solução para o SD da Cemig A, exemplo 2..... 52
4.10 – Aproximação da Fronteira Pareto para o Sistema Cemig A, exemplo 3 – Topologia 1 54
4.11 – Exemplo de Solução para o SD da Cemig A, exemplo 3 – Topologia 1..... 55
4.12 – Aproximação da Fronteira Pareto para o Sistema Cemig A, exemplo 3 – Topologia 2 56
4.13 – Exemplo de Solução para o SD da Cemig A, exemplo 3 – Topologia 2..... 57
4.14 – Esquema Ilustrativo para o Sistema Cemig B..... 58
4.15 – Aproximação da Fronteira Pareto para o Sistema Cemig B, exemplo 1..... 59
4.16 – Exemplo de Solução para o SD da Cemig B, exemplo 1..... 59
4.17 – Aproximação da Fronteira Pareto para o Sistema Cemig B, exemplo 2..... 61
4.18 – Exemplo de Solução para o SD da Cemig B, exemplo 2..... 62

4.19 – Aproximação da Fronteira Pareto para o Sistema Cemig B, exemplo 3 – Topologia 1 64
4.20 – Exemplo de Solução para o SD da Cemig B, exemplo 3 – Topologia 1..... 65
4.21 – Aproximação da Fronteira Pareto para o Sistema Cemig B, exemplo 3 – Topologia 2 67
4.22 – Exemplo de Solução para o SD da Cemig B, exemplo 3 – Topologia 2..... 68

Lista de Tabelas

Tabela 1 – Tipos das Chaves para o Sistema Teste de 16 Blocos 34

Tabela 2 – Tipos das Chaves para o Sistema Teste de 16 Blocos após a Primeira Iteração 34

Tabela 3 – Tipos das Chaves para o Sistema Teste de 16 Blocos após a Segunda Iteração 35

Tabela 4 – Exemplo de Pais Subdivididos em Grupos Conforme os Tipos de Chaves para o Sistema Teste de 16 Blocos 42

Tabela 5 – Filhos Subdivididos em Conforme os Tipos de Chaves para o Sistema Teste de 16 Blocos 42

Tabela 6 – Resultado para o exemplo 3 do sistema Cemig A, topologia 1 53

Tabela 7 – Chaves abertas e fechas por estágio para o exemplo 3 do sistema Cemig A, topologia 1 54

Tabela 8 - Resultado para o exemplo 3 do sistema Cemig A, topologia 2 55

Tabela 9 – Chaves abertas e fechas por estágio para o exemplo 3 do sistema Cemig A, topologia 2 57

Tabela 10 - Resultado para o exemplo 2 do sistema Cemig B 60

Tabela 11 – Chaves abertas e fechas por estágio para o exemplo 2 do sistema Cemig B 61

Tabela 12 - Resultado para o exemplo 3 do sistema Cemig B, topologia 1 62

Tabela 13 – Chaves abertas e fechas por estágio para o exemplo 3 do sistema Cemig B, topologia 1 64

Tabela 14 - Resultado para o exemplo 3 do sistema Cemig B, topologia 2 66

Tabela 15 – Chaves abertas e fechas por estágio para o exemplo 3 do sistema Cemig B, topologia 2 67

Lista de Abreviaturas e Siglas

AEs: Algoritmos Evolutivos

ARG: Ações de Restauração Genérico

Cemig: Companhia Energética de Minas Gerais

DEC: Duração Equivalente de Continuidade

DIC: Duração de Interrupção Individual por Unidade Consumidora.

DICRI: Duração de Interrupção Individual Ocorrida em dia Crítico por Unidade Consumidora

DEMIC: Duração Máxima de Interrupção Contínua por Unidade Consumidora

FEC: Frequência Equivalente de Continuidade

FIC: Frequência de Interrupção Individual por Unidade Consumidora

GA: Algoritmo Genético

MOGA: Multi-objective Genetic Algorithm

MOOP:*Multi-Objective Optimization Problem*

NA: Normalmente Aberto

NF: Normalmente Fechado

NPGA: Niched Pareto Genetic Algorithm

NSGA: Non-dominated Sorting Genetic Algorithm

NC: Número de Chaves por Manobras

PRE: Plano de Reestabelecimento de Energia

RNP: Representação Nó-Profundidade

SDEE: Sistema de Distribuição elétrica SPEA

SDR: Sistema de Distribuição Radial

SEP: Sistema Elétrico de Potência

TSP: Problema do Caixeiro Viajante

VEGA: Vector Enabled Genetic algorithm

Sumário

1 Introdução **02**

1.1 Sistemas Elétricos de Potência..... 02
1.2 Objetivos..... 05
1.3 Relevância do Estudo 05
1.4 Visão Geral da Dissertação..... 05

2 Revisão Bibliográfica **07**

2.1 Grafos e Árvores..... 07
2.1.1 Árvores..... 08
2.2 Restauração e Reconfiguração de Redes..... 09
2.2.1 Otimização de SDEE..... 14
2.3 Otimização Mono e MultiObjetivo..... 15
2.4 Algoritmos Genéticos MonoObjetivo..... 16
2.5 Problema de Otimização Multiobjetivo..... 18
2.5.1 Soluções Pareto Ótimas..... 19
2.6 Algoritmos Genéticos MultiObjetivo..... 20
2.6.1 SPEA2 – Strenght Pareto Evolutionary Algorithm..... 23
2.7 Síntese do Capítulo..... 25

3 Abordagem Proposta **26**

3.1 Modelagem do Problema..... 26
3.2 Arquitetura do Algortimo..... 27
3.3 Classificação Blocos e Chaves..... 29
3.5 Codificação e Decodificação..... 30
3.6 Seleção, Cruzamento e Mutação 39
3.7 Síntese do Capítulo..... 44

4 Resultados e Discussão 45

4.1 Sistema de 16 Blocos..... 45
4.2 Sistema Cemig A..... 47
4.2.1 Exemplo 1 – Cemig A..... 48
4.2.2 Exemplo 2 – Cemig A..... 51
4.2.3 Exemplo 3 – Topologia 1 - Cemig A..... 52
4.2.4 Exemplo 3 – Topologia 2 - Cemig A..... 55
4.3 Sistema Cemig B..... 57
4.3.1 Exemplo 1 – Cemig B..... 58
4.3.2 Exemplo 2 – Cemig B..... 60
4.3.3 Exemplo 3 – Topologia 1 - Cemig B..... 62
4.3.4 Exemplo 3 – Topologia 2 - Cemig B..... 65

4.4 Síntese do Capítulo..... 68

5 Conclusões 70

6 Referências Bibliográficas 72

Capítulo 1

Introdução

Neste capítulo é feita uma breve introdução sobre os sistemas elétricos de potência (SEP), enfatizando a necessidade da restauração rápida do mesmo. São apresentados também, os objetivos do trabalho e a motivação que levou ao seu desenvolvimento.

1.1. Sistemas Elétricos de Potência

A energia elétrica já se tornou um recurso indispensável para a manutenção das estruturas socioeconômicas das sociedades contemporâneas. Desde grandes consumidores, como indústrias, centros comerciais, irrigantes, grandes produtores rurais, até pequenos e médios consumidores, como condomínios, edifícios e residências, todos dependem da energia elétrica para exercer suas atividades.

Dessa forma, tem-se buscado sempre a maior qualidade e confiabilidade possível no fornecimento de energia elétrica, fazendo com que este seja cada vez menos susceptível a falhas. Para manter esse fornecimento, existe um complexo sistema de redes constituído por grandes e pequenas centrais geradoras, extensas linhas de transmissão e uma rede altamente ramificada de distribuição. A esse sistema como um todo, dá-se o nome de Sistema Elétrico de Potência (SEP).

O propósito de um sistema de energia elétrica é atender às demandas das cargas, promovendo um suprimento adequado dessa energia. Segundo Thakur and Jaswanti, 2006, cerca de 40% do investimento total no setor de energia elétrica se concentra no sistema de distribuição. Ainda segundo Thakur and Jaswanti, 2006, o sistema de distribuição de energia elétrica (SDEE) consiste em dois circuitos radiais (primário e secundário) interconectados e possui um número finito de restrições, como: operação coordenada estabelecida dos dispositivos de sobrecorrente, configuração radial, perfil de tensão dentro dos limites estabelecidos, atendimento de todas as cargas, dentre outras.

É no estágio de distribuição que se concentra a maior perda de potência mensurável em todo o sistema elétrico. Pelo menos 2% da energia elétrica são perdidos nas etapas de média tensão. Além disso, mais de 80% de todas as interrupções no fornecimento de energia que afetam diretamente os consumidores devem-se a falhas no sistema de distribuição (Thakur and Jaswanti, 2006).

Nas últimas décadas, os SEPs não passaram apenas por um amplo processo de expansão em escala, mas também receberam um aumento expressivo no número de cargas que utilizam eletrônica, cargas estas potencialmente perturbadoras ao sistema, além de interligação de centrais de cogeração, tornando o sistema cada vez mais complexo de se operar (Santos, 2009).

Devido à essa complexidade, os sistemas elétricos de potência frequentemente são submetidos a condições anormais de operação em função de eventos tais como curtos-circuitos, subtensões, sobretensões, descargas elétricas, dentre outros. Nestas situações torna-se necessário identificar a condição anormal e isolar o trecho afetado, de forma a permitir a continuidade do fornecimento de energia elétrica nas áreas sãs do sistema. Esta função é executada pelo sistema de proteção, que consiste de um conjunto de relés. Estes recebem informações dos transformadores de corrente e de tensão e, caso detectem alguma anormalidade, enviam comandos de abertura para um ou mais disjuntores de forma a isolar o defeito. Outra possível função do sistema de proteção é informar às equipes de operação qual o tipo e a estimativa da localização do defeito ou regime anormal de operação, com o objetivo de permitir sua rápida mitigação.

Além disso, devido à necessidade das concessionárias de atingirem as metas dos indicadores de continuidade coletivos, DEC (Duração Equivalente de Continuidade), FEC (Frequência Equivalente de Continuidade) e individuais DIC (Duração de Interrupção Individual por unidade consumidora), FIC (Frequência de Interrupção Individual por unidade consumidora) e DMIC (Duração máxima de Interrupção Contínua por Unidade Consumidora) estabelecidas pela ANEEL, tornou-se crucial a automatização dos sistemas elétricos. Diversos métodos e propostas têm surgido para tentar resolver o problema do reestabelecimento de energia em tempo real através de procedimentos de reconfiguração.

A restauração da rede tem o principal objetivo de viabilizar uma configuração factível, após a ocorrência de contingências no sistema elétrico. Essa nova

configuração deve ser capaz de atender de forma aceitável o maior número de consumidores e/ou as cargas mais importantes, reestabelecendo o funcionamento do sistema da melhor forma possível. Assim, a restauração pode ser entendida como um caso particular e emergencial da reconfiguração, que é necessária imediatamente após a ocorrência de uma ou mais faltas. Os requisitos e restrições do sistema devem ser atendidos de forma que a nova configuração possa ser implementada com segurança.

Muitos fatores contribuem para a justificativa da importância de se estudar o tema da restauração em sistema de distribuição primária. Alguns deles são citados abaixo (Barbosa, 2012):

- crescimento contínuo da demanda por energia no mundo, principalmente nos países com cenário econômico favorável (ex. Brasil, Índia, China...);
- crescente complexidade dos sistemas elétricos de distribuição fomentada pela automação e informatização dos mecanismos de monitoramento e controle;
- necessidade de atendimento aos índices de qualidade do serviço;
- liberalização do mercado de energia, tratada como *commodity*, favorecida pela Medida Provisória nº 579, de 11 de setembro de 2012;
- oportunidade clara de conhecimento e recursos aptos a promover uma operação mais eficiente do sistema elétrico com o principal objetivo de reduzir os custos de operação;
- mudanças nas características das cargas (ex. cargas eletrônicas) que se tornaram mais sensíveis e complexas de se modelar;
- possibilidade de postergação de novos investimentos e a capacidade de tornar o sistema menos vulnerável aos agentes causadores de incidentes ou que perturbam seu funcionamento normal (tempestades, falhas em equipamentos, indisponibilidade de potência etc.).

A modificação da configuração em sistemas de distribuição primária não apenas pode otimizar suas condições de operação como também possibilita o uso mais apropriado da infraestrutura disponível (PEPONIS e PAPADOPOULOS, 1995).

Os algoritmos evolutivos (AEs) são técnicas alternativas que têm se demonstrado capazes de lidar com problemas desse tipo mesmo em redes de grande dimensão, como pode ser visto em (Barbosa, 2012).

Esta dissertação propõe uma nova estratégia para restauração de sistemas de distribuição de energia elétrica baseada em otimização não linear multiobjetivo. Os dois objetivos a serem minimizados são: tempo estimado para a realização das manobras, tendo em vista que há diferença entre as chaves telecomandadas e as manuais, e o número total de cargas que foram desligadas devido à contingência. O algoritmo genético multiobjetivo base usado na implementação é o SPEA2 (do inglês, *Strength Pareto Evolutionary Algorithm 2*).

1.2. Objetivo

Este trabalho tem como objetivo desenvolver uma ferramenta que seja capaz de auxiliar no processo de restauração da rede de distribuição primária, quando esta sofre faltas em um ou mais pontos do sistema.

Essa ferramenta, dado o sistema de distribuição de interesse e um conjunto de faltas, deve encontrar soluções que sejam capazes de restaurar a maior quantidade de carga no menor tempo possível. Além disso, o percurso percorrido até a realização de todas as manobras deve demandar o mínimo tempo possível. Com isso espera-se atingir melhorias significativas nos índices de DEC, DIC e DMIC da concessionária.

1.3. Relevância do Estudo

Este trabalho tem como principal contribuição o fato do algoritmo implementado, não só minimizar a carga desconectada ao fim do conjunto de manobras, mas também minimizar a carga desconectada ao fim de cada iteração, de forma a dar certeza ao decisor que a sequência de manobras apresentada é a que recuperará a maior quantidade de carga ao longo de todo o percurso.

1.4. Visão Geral da Dissertação

O trabalho está organizado da seguinte forma: O Capítulo 2 é uma revisão bibliográfica onde são discutidos alguns trabalhos relacionados ao uso de técnicas de otimização na reconfiguração de redes. O Capítulo 3 apresenta a modelagem do

problema em questão e o algoritmo implementado. O Capítulo 4 apresenta e discute os resultados computacionais. O Capítulo 5 trata das conclusões do trabalho.

Capítulo 2

Revisão Bibliográfica

Neste capítulo é feita uma revisão da teoria básica de grafos e árvores, importante ferramenta na representação computacional dos sistemas de distribuição radiais (SDRs). É apresentado também o estado da arte de restauração e reconfiguração dos SEPs. Ainda neste capítulo é feita uma breve revisão sobre os conceitos de otimização mono e multiobjetivo e sobre o algoritmo genético utilizado para desenvolver a ferramenta.

2.1 Grafos e Árvores

A teoria de grafos é um ramo da matemática que estuda a relação entre objetos de um conjunto. Tem aplicação em diversos tipos de problemas como o clássico problema do caixeiro viajante, problemas de caminho mínimo e da inspeção de rotas e fluxos de redes.

Os problemas dos SDEE são geralmente modelados por grafos e daí parte a necessidade do entendimento das definições e as associações entre a teoria de grafos e a topologia das redes de distribuição.

Um grafo G é constituído de um par (V, A) e uma matriz de adjacência M, onde V é um conjunto finito de pontos denominados vértices (ou nós), A é um conjunto finito de arestas (ou conexões ou ramos) interligando esses vértices e M associa a cada aresta de G um par não ordenado de vértices (não necessariamente distintos) de G, chamados extremos de A (Bondy e Murty, 1976; Wilson, 1996).

Existem diversos tipos de grafos, entretanto na representação das redes de distribuição de energia são utilizados grafos planares, estruturados como árvores, com arestas não direcionadas e sem realimentação dos nós. A figura 2.1 apresenta um grafo não direcional com seis vértices, $V = [1, 2, 3, 4, 5\ e\ 6]$ e dez arestas, $A = [a, b, c, d, e, f, g, h, i, j]$.

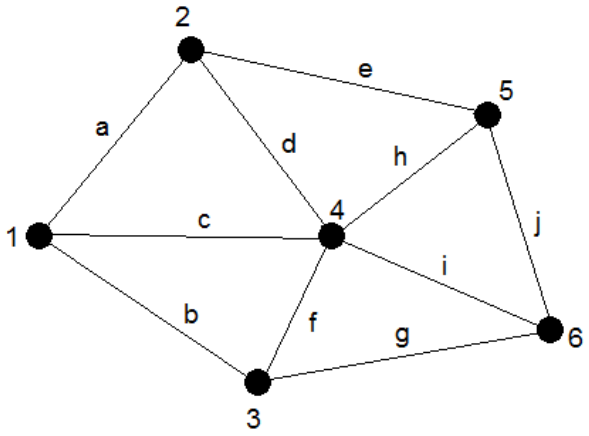

Figura 2.1 – Exemplo de grafo

A matriz de adjacência M é quadrada e possui seus elementos M(i,j) iguais a 0 ou 1, indicando se existe ou não uma aresta ligando o vértice V_i ao vértice V_j. A matriz M abaixo representa a matriz adjacente referente ao grafo da Figura 2.1.

$$M = \begin{bmatrix} & V1 & V2 & V3 & V4 & V5 & V6 \\ V1 & 0 & 1 & 1 & 1 & 0 & 0 \\ V2 & 1 & 0 & 0 & 1 & 1 & 0 \\ V3 & 1 & 0 & 0 & 1 & 0 & 1 \\ V4 & 1 & 1 & 1 & 0 & 1 & 1 \\ V5 & 0 & 1 & 0 & 1 & 0 & 1 \\ V6 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

2.1.1. Árvores

Um grafo acíclico e conexo, ou seja, no qual existe um caminho entre qualquer par de vértices distintos, é chamado de árvore. Toda árvore é um grafo bipartido e planar. Um grafo G é uma árvore se satisfaz as seguintes condições:

- G é conexo e existe um e apenas um caminho entre dois vértices quaisquer;
- G é conexo, acíclico e tem n-1 arestas.
- G é acíclico, e um simples ciclo é formado se qualquer aresta for adicionada a G;
- G é conexo, mas deixará de ser caso qualquer aresta seja removida de G;

A primeira e a segunda condições citadas acima são dois teoremas propostos por Narsingh, 1984.

Cayley, 1989 propôs:

Teorema de Cayley: em um grafo completo G_c, com *n* vértices e $m = \frac{n*(n-1)}{2}$ arestas, existem n^{n-2} árvores que são sub-grafos de G_c.

Pelo teorema de Cayley é possível ter uma noção sobre a dimensão do espaço de busca do problema, no qual o aumento do número de nós faz com que essa dimensão aumente exponencialmente.

A Figura 2.2 abaixo apresenta um exemplo de árvore.

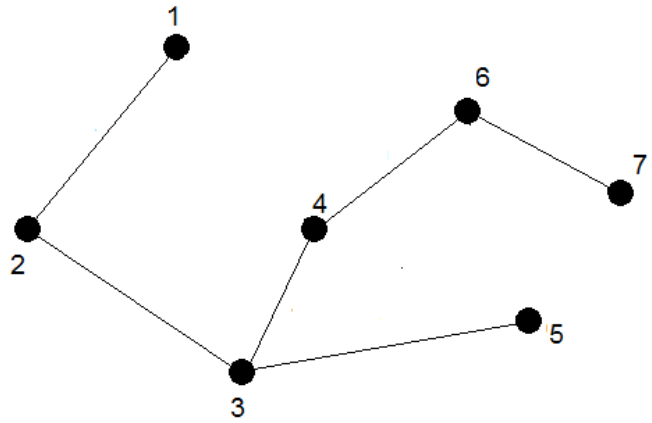

Figura 2.2 – Exemplo de árvore.

2.2. Restauração e Reconfiguração de Redes

Os sistemas de distribuição de energia elétrica possuem característica radial. Essa radialidade simplifica a operação e a proteção, porém diminui a confiabilidade desses sistemas em relação a determinados fatores para a qualidade no funcionamento da energia (Mansour, 2009). No caso de uma falha no tronco do alimentador que cause atuação da proteção e desligamento do trecho, por exemplo, todos os ramais subsequentes a este ponto são afetados (desconectados).

Dessa forma, surge a necessidade de se ter um plano de restabelecimento de energia (PRE) que seja eficiente, com o intuito de não existirem desligamentos de grandes áreas por longos períodos de tempo. De acordo com (Santos, 2009) um PRE adequado envolve os seguintes objetivos:

- Encontrar um plano em um curto intervalo de tempo;

- Minimizar o número de manobras;
- Reduzir o número de consumidores interrompidos;
- Evitar sobrecarga dos componentes;
- Manter a estrutura radial do sistema;
- Reduzir o total de perdas de potência;
- Reduzir quedas de tensão.

Diversos métodos e propostas têm surgido para tentar resolver o problema do reestabelecimento de energia em tempo real através de procedimentos de reconfiguração. Entretanto, devido à característica combinatória desse problema, as técnicas de programação matemática exatas não são adequadas para tratamento desse problema em sistemas de grande porte. Os algoritmos evolutivos (AEs) são uma técnica alternativa que tem se demonstrado capaz de lidar com essas dificuldades.

Conforme observado por (Mansour, 2009) um dos fatores que influenciam o desempenho de um AE convencional é a estrutura de dados adotada. Para se elaborar um PRE utilizando AEs, é necessário um algoritmo de busca em grafos, sendo assim o desempenho do AE afetado pela forma que as árvores de grafos são representadas computacionalmente, principalmente devido à forma com que o algoritmo lida com a factibilidade de soluções.

Para melhorar o desempenho dos AEs aplicados à formação de PREs, em Santos et al., 2010, foi empregada uma nova estrutura de dados, chamada Representação Nó-Profundidade (RNP). Esta estrutura possui dois operadores, que permitem a realização de poda ou enxerto nas árvores da floresta de grafo armazenada na RNP, as quais representam os alimentadores de um Sistema de Distribuição Radial (SDR), de forma a gerarem novas árvores, que correspondem a configurações factíveis, isto é, redes radiais capazes de fornecer energia para todo a sistema (Simone, 2012). Esta última característica concede à ferramenta maior eficiência, pois diminui a quantidade de soluções não factíveis.

Moussa et al, 2010 e Sanches et al, 2011 também utilizaram a estrutura de dados RNP para PRE. Entretanto, a metodologia apresentada em Mansour et. al., 2009, denominada NSDE, baseia-se na combinação da RNP com uma versão modificada do NSGAII. Já a metodologia proposta em Sanches et. al., 2011, denominada MEAN_NDS, procura combinar os melhores aspectos da metodologia

MEAN e NSDE, para tratamento do problema de redução de perdas em SDs. Assim como a metodologia MEAN, a MEAN_NDS trabalha em paralelo com varias subpopulações armazenadas em tabelas, nos quais as melhores soluções, para cada característica do problema, são armazenadas em suas respectivas tabelas. Entretanto, a MEAN-NDS faz uso de tabelas adicionais de subpopulações não dominadas, que armazenam as soluções não dominadas obtidas durante as gerações (Simone, 2013).

(El-werfelli, 2009) descreve um novo algoritmo de restauração de sistemas de potência utilizando a associação de Algoritmos Genéticos (AG) e Sistemas Especialistas (SE). O AG é utilizado para obter o esqueleto otimizado da rede, enquanto o SE age como um operador de sistemas capaz de restringir eficientemente o espaço de soluções para o AG. A estratégia é testada para o caso do IEEE de 39 barras. As funções objetivos utilizadas são: Minimização do tempo de restauração e minimização dos limites de segurança, integrando o sistema de energia, tais como os limites de estabilidade e perfil de tensão. Os resultados comprovam que a aplicação do SE pode ser significativamente melhorada através da combinação proposta.

O trabalho desenvolvido por (Nouri-Zadeh, 2008) apresenta um modelo estocástico para o planejamento da restauração de unidades geradoras. Para resolver o problema, os autores desenvolvem um sistema de planejamento de recuperação avançada multiobjetivo baseado em programação não linear inteira mista. As funções objetivos utilizadas foram: minimização do risco total do sistema, da energia total do sistema no período de restauração e o tempo total da restauração. Os resultados apresentaram que mais restrições levam a planos mais reais de restauração.

(Hou, 2009) estabeleceu o novo conceito de "marcos de restauração genéricos" (MRG). Este conceito é captado a partir das estratégias atuais de restauração do sistema de energia. Com base nas condições do sistema, uma estratégia de restauração específica pode ser estabelecida por uma combinação de MRGs. De acordo com as características e políticas de um sistema, cada GRM pode ser implementado por uma sequência de "ações de restauração genérico" (ARG). Todas as restrições são satisfeitas durante a implementação destas ações, e a sequência de ARG é otimizada com a finalidade de MRGs. Simulações com base

num sistema de teste padrão, com 39 barras e 10 unidades geradoras com um total geração de 7347 MW e 2470 Mvar, ilustram a generalidade dos métodos propostos. A metodologia abrangente para a construção de estratégias de restauração do sistema pode ser obtida a partir dos conceitos propostos.

(Mil, 1998) projetou um método de restauração de serviço rápido, eficaz para sistemas de distribuição de larga escala com clientes prioritários. O problema de restauração de serviço é formulado como um problema de otimização multi-objetivo restrito. O autor projetou um método de pesquisa baseado em classificação empregando informações analíticas obtidas a partir de simulações de fluxo de potência trifásico para restaurar o serviço ao maior número de clientes prioritários quanto possível. As funções objetivos foram: maximizar a quantidade de carga total restaurada e minimizar o número de operações de comutação necessários. O método proposto foi concebido considerando redes que têm interruptores predominantemente manuais. Um sistema de distribuição de 416 barras foi utilizado para testar a eficiência da proposta.

Em 2006, Mendoza et al. apresentaram aplicações de duas outras técnicas de otimização multiobjetivo, NSGA (Srinivas and Deb, 1994) e SPEA (Zitzler and Thiele, 1999) no planejamento de sistemas de planejamento de distribuição de energia elétrica. Foram utilizadas duas funções objetivo: a redução do custo total envolvido e a minimização da área sem energia. A lei de *Kirchhoff* para todos os nós, o carregamento nos alimentadores e subestações e a queda de tensão foram as restrições consideradas pelos autores. O NSGA trabalha identificando soluções não-dominadas na população em cada geração para formar fronteiras não-dominadas baseado no conceito de não-dominância de Pareto e em seguida utilizando operadores de cruzamento e mutação. O SPEA utiliza o elitismo através da manutenção de um conjunto externo de melhores soluções encontradas durante cada iteração. As soluções não-dominadas no conjunto externo são utilizadas para determinar a adequação da população atual e para seleção no processo de reprodução. Um algoritmo baseado em conjuntos fuzzy (Bezdek, 1981,1999) é utilizado para agrupar os indivíduos que permaneceram na população externa caso a quantidade de indivíduos ultrapasse o limite determinado a priori. Um sistema de distribuição radial com 41 barras e 2 subestações de 20MVA foi utilizado nos testes.

Ambas as técnicas apresentaram resultados computacionais similares em relação ao tempo de processamento e às fronteiras Pareto encontradas.

Em (Ramaswamy et. al., 2012) os autores utilizam um Algoritmo Genético (AG) básico e o NSGA-II [DEB 2002] (do inglês*, Non-Dominated Sorting Genetic Algorithm*) para reconfiguração de um SD. Neste caso, os objetivos são: minimizar a perda de potência e minimizar os desvios de tensão. Os autores mostram que a utilização do NSGA-II é mais eficiente que utilizar somente o AG básico quando múltiplos objetivos são considerados. O método proposto foi testado em um sistema de 16 barras. Os resultados mostraram que o AG simples é capaz de encontrar eficientemente o ótimo global para redes simples. Entretanto, o mesmo falha em problemas multiobjetivos ao tentar encontrar a fronteira Pareto ótima. Esta desvantagem pode ser superada com a utilização do NSGA-II o qual lida com soluções Pareto ótimas.

A reconfiguração e a restauração de redes de distribuição de energia elétrica são procedimentos similares, mas com objetivos distintos.

A restauração da rede tem o principal objetivo de viabilizar uma configuração factível que atenda de forma satisfatória no menor tempo possível o maior número de consumidores e/ou as cargas mais importantes após a ocorrência de uma contingência no sistema elétrico. Assim, a restauração pode ser entendida como um caso particular e emergencial da reconfiguração, ou seja, a restauração é a mudança a ser realizada na topologia da rede imediatamente após a ocorrência da falta. Os requisitos e restrições do sistema devem ser atendidos de forma que a nova configuração possa ser implementada com segurança.

A implementação de um plano de restauração eficiente é crucial para que as concessionárias consigam atender os clientes de forma adequada dentro dos índices de qualidade do serviço. Destacam-se no aspecto da qualidade do serviço os indicadores coletivos, o DEC (Duração Equivalente de Continuidade) que registra quantas horas em média por ano o consumidor fica sem energia elétrica, o FEC (Frequência Equivalente de Continuidade) que indica quantas vezes em média a energia faltou para os consumidores e o DICRI (Duração da interrupção individual ocorrida em dia critico por unidade consumidora ou ponto de conexão) que corresponde à duração de cada interrupção ocorrida em dia crítico, para cada unidade consumidora ou ponto de conexão (ANEEL, 2013).

Além dos indicadores coletivos, os serviços prestados pelas empresas distribuidoras são avaliados por indicadores individuais DIC (Duração de Interrupção Individual por unidade Consumidora) e FIC (Frequência de Interrupção Individual por Unidade Consumidora), que medem respectivamente a duração e a frequência das interrupções do fornecimento de energia em cada unidade consumidora. Outro indicador individual que as distribuidoras devem observar é o DMIC (Duração Máxima de Interrupção Contínua por Unidade Consumidora), que registra o tempo máximo que uma unidade consumidora permaneceu sem energia no intervalo de tempo de apuração (ANEEL, 2013).

2.2.1. Otimização de SDEE

Os problemas de SDEE devido a sua estrutura em árvore (radial) têm natureza combinatória e assim as possíveis soluções crescem exponencialmente com a dimensionalidade do problema.

Algoritmos evolutivos têm sido bastante utilizados nos últimos anos em problemas envolvendo redes de distribuição de energia elétrica, podendo citar: planejamento, posicionamento de subestações, alocação de linhas, chaves e equipamentos e problemas de reconfiguração e restauração em geral, envolvendo muitos objetivos.

O primeiro trabalho encontrado na literatura sobre reconfiguração com objetivo de minimizar perdas (MERLIN e BACK, 1975) utilizou dois métodos para resolver o problema, um heurístico e um exato, para uma rede de pequeno porte. O método heurístico consistia no fechamento de todas as chaves abertas e posterior cálculo do fluxo de potência, o arco com menor fluxo é então aberto. Esse processo continua até que uma rede radial é obtida e foi posteriormente denominado "método da abertura sequencial de chaves".

Posteriormente, Civanlar et. al (1988) propôs o método conhecido como *branch-exchange* para resolver um problema de reconfiguração. Seu método consistia na troca de ramos, com o fechamento de uma chave e abertura de outra, mantendo a radialidade. Foi testado para dois sistemas, de 13 e 16 barras. Este método podia avaliar a parcela de perda por manobra executada, evitando assim o cálculo de fluxo de potência na nova configuração.

Posteriormente vários autores, como Mantovani et. al. (2000) e Coelho M. et. al. (2011), propuseram mudanças nos métodos de Merlin e Back (1975) e Civanlar et. al (1988) com intuito de melhorar o desempenho computacional.

Mantovani et. al. (2000) apresenta um algoritmo heurístico (tipo *branch-and-bound*) que realiza a reconfiguração de redes de distribuição com vistas a obter menores perdas ativas nos alimentadores. O algoritmo desenvolvido é baseado no limite da queda de tensão com sistemas hipotéticos (14, 32 e 69 barras) e um sistema real 135 barras, com utilização do método aproximado de Fluxo de Potencia (Forward Update Distflow). Os resultados mostraram a solução ótima ou muito próxima da ótima em tempo computacional relativamente baixo.

Coelho M. et. al. (2011) aborda o problema da reconfiguração de redes de distribuição utilizando a meta-heurística GRASP. Oliveira M. X. et. al. (2011) considerou que todos os ramos do sistema possuem uma chave de interconexão que pode ser aberta ou fechada a qualquer momento, a fim de aumentar a qualidade da função objetivo. O sistema foi testado para 84, 119 e 136 barras. Os resultados não foram diferentes de trabalhos passados.

2.3. Otimização Mono e MultiObjetivo

A otimização consiste em descobrir, utilizando métodos computacionais, a maneira mais eficiente de executar ou projetar um processo ou uma tarefa. Assim, para se obter um resultado melhor (ou um conjunto de melhores resultados), é necessário minimizar ou maximizar a função (ou funções) que representa o objetivo (ou objetivos) a ser alcançado. O processo de otimização sempre resulta, ou busca justificação, em um impacto econômico, como a qualidade do produto, o custo da produção, a competitividade, dentre vários outros.

A otimização pode ser classificada quanto ao número de objetivos, mono-objetivo, quando se tem um objetivo ou multiobjetivo, quando se tem dois ou mais objetivos. Quanto ao domínio das variáveis, podendo ser discretas ou combinatórias, quando não se conhece nenhum algoritmo exato rápido ou contínuas, quando não se conhece um algoritmo exato rápido capaz de localizar o ótimo global com precisão finita. Quanto às restrições, podendo ser restritas, quando há restrições ou irrestritas, quando não há restrições (Azuma, 2011).

Na otimização mono-objetivo, como o próprio nome sugere, existe somente um único objetivo a ser alcançado. Dessa forma, encontra-se somente uma única solução ótima para o problema.

A otimização multiobjetivo é o ramo da otimização matemática que busca desenvolver métodos de resolução para problemas reais que possuem dois ou mais objetivos que devem ser atingidos simultaneamente, podendo ou não possuir restrições. Geralmente esses objetivos concorrem entre si de tal modo que a melhoria de um pode levar à degradação dos demais. Dessa forma, normalmente, não existe uma solução única que otimize todos eles ao mesmo tempo mas sim um conjunto de soluções eficientes.

2.4. Algoritmos Genéticos Mono-Objetivo

Os algoritmos genéticos (AGs) foram desenvolvidos pelo americano John Henry Holland na Universidade de Michigan na década de 60. Ele desenvolveu simulações em computador que retinham os mecanismos originais encontrados em sistemas evolutivos naturais baseados na teoria evolutiva de Darwin (Silva, 2008)

Goldberg (1989) citou quatro razões que fazem os AGs atrativos para aplicações:

- AGs podem resolver problemas difíceis em um pequeno espaço de tempo de forma confiável;
- A interface para a construção de AGs com os modelos de problemas existentes é, geralmente, simples;
- AGs são extensíveis; e
- AGs são fáceis de hibridizar;

A Estrutura básica de um algoritmo genético é apresentada pela Figura 2.3 a seguir.

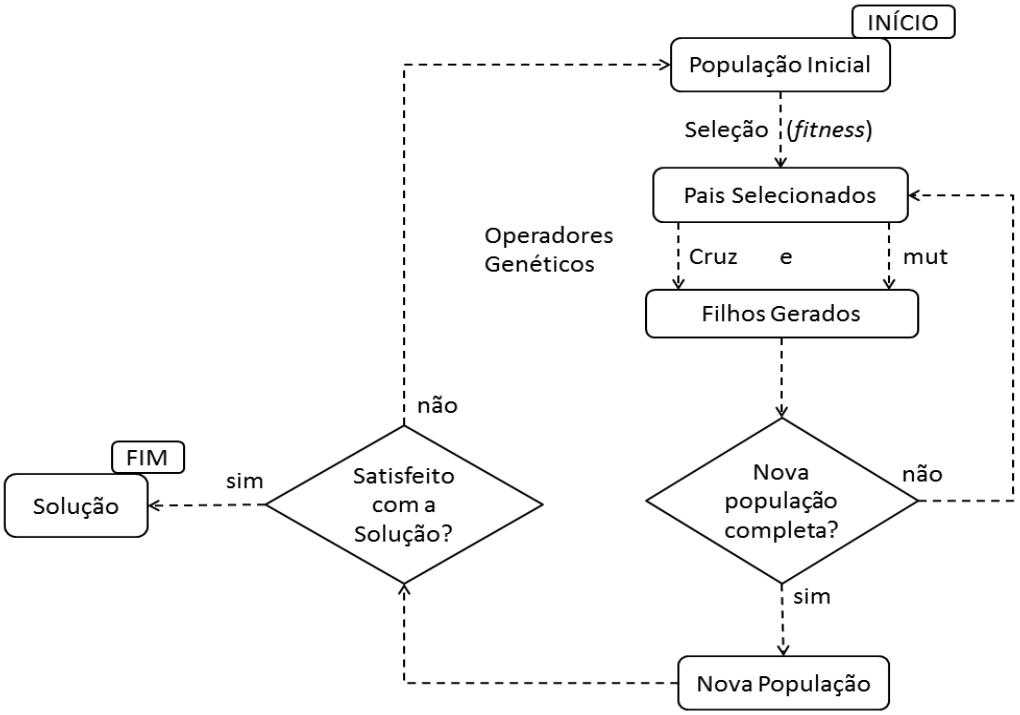

Figura 2.3 – Estrutura de um algoritmo genético

Inicialmente uma população de indivíduos é gerada aleatoriamente. Os indivíduos são portadores dos códigos genéticos que são uma representação do espaço de busca do problema a ser solucionado. Ou seja, um indivíduo é uma possível solução do problema. Geralmente, o código genético é representado na forma sequencial de bits.

Cada indivíduo da população inicial é avaliado de acordo com a função objetivo que define o problema. Os valores da função objetivo calculados atribuem probabilidades dos indivíduos serem selecionados: quanto melhor for o indivíduo maior é chance que ele tem de sobreviver. Esses valores são chamados de aptidão ou *fitness*.

A seleção desses indivíduos é geralmente feita de forma estocástica. A mais utilizada é a roleta estocástica (do inglês, *Roulette Wheel Selection ou RWS*) Goldberg (1989). Nesse método cada slot da roleta possui tamanho proporcional à *fitness* de cada indivíduo. Roda-se a roleta e o indivíduo selecionado será aquele referente ao *slot* em que a agulha parar. Repete-se o procedimento até que a nova população possua o mesmo número de indivíduos que a população inicial.

Caso um mesmo indivíduo seja escolhido várias vezes, o método descrito anteriormente pode gerar uma nova população com variabilidade genética baixa, o

que levaria o algoritmo à uma convergência prematura. Para solucionar o problema, Baker (1987) propôs o método de amostragem universal estocástico, no qual o número de agulhas que a roleta possui é igual ao número de indivíduos da população. Essas agulhas são igualmente espaçadas na roleta que é girada uma única vez.

O Torneio é outro método utilizado na seleção dos indivíduos. São escolhidos n indivíduos aleatoriamente da população inicial (geralmente n=2) e o melhor deles é selecionado.

Após a nova população estar completa, são feitos os procedimentos de cruzamento e mutação. No cruzamento, dois (ou mais) indivíduos são escolhidos aleatoriamente e seus materiais genéticos são permutados. Dois novos indivíduos são então criados, ou seja, novos pontos no espaço de otimização são obtidos.

Na mutação, um indivíduo é escolhido e seu código genético é modificado de acordo com uma regra probabilística determinada. Haverá, nesse caso, a inserção de novas características genéticas.

O processo de cruzamento e mutação é realizado até que a população de filhos possua o mesmo número de indivíduos que a população inicial. Avalia-se essa nova população e o processo cessa quanto um critério de parada é atendido.

Alguns critérios de parada em GAs são:

- Obtenção do ótimo conhecido.
- Estabilização do valor da função objetivo.
- Execução de um número pré-determinado de gerações.

2.5. Problemas de Otimização MultiObjetivo

Problemas que envolvem a otimização de vários objetivos simultaneamente, que, em geral, são conflitantes, são chamados de problemas de otimização multiobjetivo (MOOP, do inglês *Multi-Objective Optimization Problem*). Um MOOP possui um conjunto de funções objetivo a serem atimizadas (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):

$$\left.\begin{array}{rll} \text{Maximizar / minimizar } f_m(x), & & m = 1,2, \ \dots \ , N_{obj} \\ \text{restrita a} & g_j(x) \geq 0 & j = 1,2, \ \dots \ , NR_{des} \\ & h_k(x) = 0, & k = 1,2, \dots , NR_{igu} \\ & x_i^{(inf)} \leq x_i \leq x_i^{(sup)}, & \text{i} = 1,2, \dots , N_{var} \end{array}\right\}$$

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 $NR_{igu} + NR_{des}$ 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. O vetor funções objetivo $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots . , f_{N_{obj}}(\boldsymbol{x})]$ compõe um espaço multidimensional chamado espaço de objetivos S_{obj}. Para cada solução **x** no espaço de decisão, existe um $\boldsymbol{f}(\boldsymbol{x})$ 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 $\boldsymbol{f}(\boldsymbol{x})$ (de dimensão N_{obj}). (Delbém, 2008).

2.5.1. Soluções Pareto Ótimas

As funções objetivo empregadas nos MOOPs são em geral conflitantes entre sim. Dessa forma, não é possível melhorar o valor de uma sem piorar o da outra. Um exemplo prático de objetivos conflitantes são preço e qualidade de um produto. Os produtos de melhor qualidade geralmente apresentam um custo maior. Assim, em uma compra devem ser considerados vários modelos do produto com diversos valores nos objetivos de preço e qualidade. Se ambos os objetivos possuem a mesma importância (ou prioridade), não há como afirmar, por exemplo, que certa redução de preço compensa determinada perda de qualidade (Mansour, 2009).

Em otimização multiobjetivo, o conceito de otimalidade baseia-se na noção introduzida por Francis Ysidro Edgeworth em 1881 (Edgeworth, 1881) e depois generalizada por Vilfredo Pareto em 1896 (Pareto, 1896). É chamado de Edgeworth-Pareto ótimo ou, simplesmente, Pareto-ótimo. (Azuma, 2011)

O conjunto que engloba as soluções ótimas obtidas é denominado conjunto Pareto-ótimo se, para cada solução desse conjunto, não existe nenhuma outra solução que seja melhor que elas, ou seja, se não existe nenhuma outra solução factível capaz de reduzir o valor de um dos objetivos do problema sem que, simultaneamente, cause o aumento em pelo menos um dos outros objetivos.

A avaliação das soluções factíveis contidas no conjunto Pareto-ótimo é feita através da análise de dominância de Pareto. Ao se comparar duas soluções é dito que uma é melhor que a outra quando a primeira Pareto domina a segunda, e é dito que elas são incomparáveis quando nenhuma exerce Pareto dominância sobre a outra. Em uma linguagem mais técnica, é dito que x_1 Pareto domina x_2 se $f(x_1) \leq f(x_2)$ e $f(x_1) \neq f(x_2)$, ou seja, em pelo menos um dos objetivos a desigualdade é atendida de forma estrita.

2.6. Algoritmos Genéticos Multiobjetivo

A otimização multiobjetivo é o ramo da otimização matemática que busca desenvolver métodos de resolução para problemas reais que possuem dois ou mais objetivos que devem ser considerados simultaneamente, podendo ou não possuir restrições. Geralmente esses objetivos concorrem entre si de tal modo que a melhoria de um leva à degradação dos demais. Dessa forma, normalmente não existe uma solução única que otimize todos eles ao mesmo tempo, mas sim um conjunto de soluções eficientes. Devido a existência desses objetivos conflitantes, encontrar soluções viáveis que otimizem simultaneamente todos os objetivos é o maior desafio.

Em otimização multiobjetivo, o conceito de otimalidade baseia-se na noção introduzida por Francis Ysidro Edgeworth em 1881 (Edgeworth (1881)) e depois generalizada por Vilfredo Pareto em 1896 (Pareto (1896)). É chamado de Edgeworth-Pareto ótimo ou, simplesmente, Pareto-ótimo.

O conjunto que engloba as soluções ótimas obtidas é denominado conjunto Pareto-ótimo se, para cada solução desse conjunto, não existe nenhuma outra solução que seja melhor ou igual à alguma solução desse conjunto em todos os objetivos simultaneamente e melhor em ao menos um.

O primeiro Algoritmo Genético multiobjetivo implementado e testado foi o chamado *Vector Evaluated Genetic Algorithm* (VEGA) desenvolvido por David Schaffer em 1984. Schaffer modificou o programa GENESIS (Gredenstette, 1984) para incluir múltiplas funções objetivos. Neste algoritmo, em cada geração, um número de sub-populações é gerado através de uma seleção proporcional de acordo com cada objetivo em questão. Assim, para um problema com 'k' objetivos, 'k' sub-populações de tamanho N/k são geradas, sendo N o número de indivíduos da população. Esses indivíduos são agrupados a fim de se obter uma nova população de tamanho N para então serem aplicados os processos de cruzamento e mutação, (Ashish Ghosh 2004, Satchidananda Dehuri, 2004). Durante algumas gerações o algoritmo é eficiente, mas, em alguns casos, ele se torna tendencioso em direção a alguns indivíduos ou regiões.

Goldberg em 1989 sugeriu um método para minimizar o problema encontrado no VEGA chamado de *nondominated sorting*. A técnica constite em dividir a população em '*fronts*' baseada na relação de dominância entre os indivíduos. Goldberg propôs um esquema de classificação de cada indivíduo em 'ranks'. O rank 1 era atribuído a todos os indivíduos que não eram dominados por nenhum outro. Para os demais indivíduos o valor do rank era atribuído de acordo com o número de pontos que os dominam mais um. A seleção é então realizada através dos ranks e não mais dos pontos individualmente. Esse processo pode levar a uma convergência prematura do algoritmo.

Em 1993, Fonseca e Fleming implementaram as sugestões de Goldberg de uma maneira diferente, propuseram então, o *Multiobjective Optimization Genetic Algorithm* (MOGA). Nesse método um processo de formação de nicho é usado para distribuir a população na região do Pareto-Ótimo. Entretanto, em vez de executar o processo de *sharing* no espaço de parâmetros, eles o executam no espaço de objetivos. Isso pode levar a uma não diversidade no espaço de parâmetros. Mesmo tendo uma fácil implementação, para Srinivas, 1994, MOGA pode não encontrar

múltiplas soluções em problemas nos quais diferentes pontos do conjunto Pareto-Ótimo correspondem ao mesmo valor de função objetivo.

Horn, Nafploitis e Goldberg, 2009, propuseram o NPGA baseado no *Pareto domination tournament* e *equivalence class sharing.* O primeiro método é basicamente um sistema de seleção de torneio baseado na Pareto dominância. Neste esquema de seleção, um conjunto de indivíduos da população inicial é escolhido aleatoriamente em cada processo de seleção. O método NPGA consiste na escolha aleatória de dois indivíduos que são comparados com um subconjunto da população, geralmente um conjunto contendo um número de indivíduos de 10% do total da população. Se um deles é dominado pelos indivíduos desse subconjunto e o outro não, então o indivíduo não dominado é escolhido. Quando os dois indivíduos são dominados ou não dominados a escolha é feita através do método do torneio.

Erickson et al. (2001), propôs o NPGA II, que utiliza o "rank" Pareto mas mantém a seleção pelo torneio, como no NPGA original. A contagem de nicho é calculado usando indivíduos da próxima geração que está parcialmente completa, ao invés de utilizar a geração atual. Esse método foi chamado de *continuously updated fitness sharing,* como proposto por Oei et al. em 1991.

Em 1994, Srinivas e Deb propuseram o *Nondominated Sorting Genetic Algoritm* (NSGA). Este método se baseia em várias camadas de classificação dos indivíduos. Os indivíduos não dominados obtém um determinado valor de fitness modelo e, em seguida, são removidos da população. O processo é repetido até que toda a população seja classificada. Buscando manter a diversidade genética da população classificada, os indivíduos são distribuídos, no espaço factível de decisão, com o seu valor de *fitness*. O método NSGA é relativamente fácil de implementar, entretanto aparenta ser muito sensível ao valor do fator de partilha e apresenta falta de elitismo (Ashish Ghosh 2004, Satchidananda Dehuri, 2004).

Tendo em vista tais falhas e problemas encontrados no NSGA, em 2002 Deb propôs uma nova versão do algoritmo, o NSGA II. O NSGA II implementa o conceito de dominância, ou seja, classifica-se a população total em *front* de acordo com o grau de dominância. Aqueles indivíduos localizados no primeiro *front* são as melhores soluções da geração considerada, enquanto que no último *front* encontram-se as piores.

2.6.1. SPEA2 – Strenght Pareto Evolutionary Algorithm

O SPEA (Zitzler & Tjiele 1999) é uma técnica para encontrar ou aproximar o conjunto Pareto-ótimo para problemas de otimização multiobjetivo que combina várias características de algoritmos evolutivos multiobjetivo anteriores de uma maneira única.

Essa técnica se baseia nas ideias básicas de um AG clássico, porém sua estrutura apresenta algumas diferenças. Esse algoritmo utiliza duas populações. Uma população regular e um arquivo externo, ou população externa, que contém as soluções não dominadas encontradas pelo algoritmo. Assim, a cada geração, as soluções não dominadas são armazenadas nessa população externa. A aptidão de cada indivíduo é dada por um valor de força (*strength*), que é proporcional ao número de soluções por ele dominadas.

Um aspecto importante deste algoritmo são as estratégias utilizadas durante o processo, que envolvem o elitismo e a diversidade. Com o elitismo, o algoritmo não permite que sejam perdidas boas soluções durante as gerações seguintes, e com a diversidade, permite que, ao longo das gerações, se obtenha uma boa diversificação do conjunto solução.

O SPEA 2 (Zitzler et al., 2001) se baseia nas ideias básicas do SPEA, as principais diferenças entre eles são:

- Um esquema de atribuição da *fitness* melhoradas é utilizada, considerando para cada indivíduo quantos outros ele domina e é dominado.
- A técnica de estimativa de densidade do vizinho mais próximo é incorporada o que permite uma orientação mais precisa do processo de busca.
- O tamanho do arquivo é fixo, ou seja, sempre que o número de indivíduos não dominados é menor que o tamanho do arquivo, este é preenchido por indivíduos dominados. No SPEA o tamanho do arquivo pode variar ao longo do tempo.
- Os novos métodos de truncamento de arquivos são utilizados, o que garante a preservação da fronteira Pareto ótimo.

A Figura 2.4 (Azuma, 2011) apresenta um diagrama da estrutura geral do SPEA 2

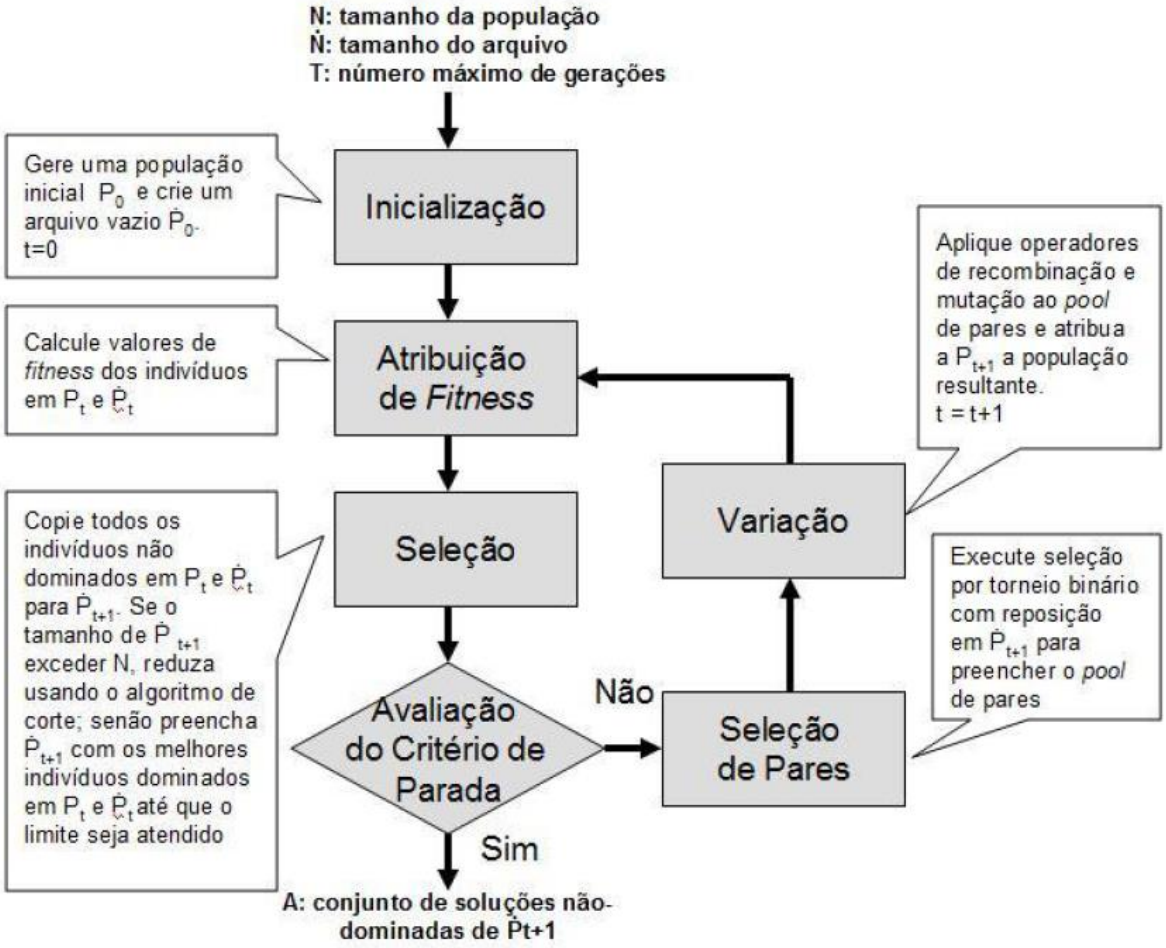

Figura 2.4 – Fluxograma Básico do SPEA 2

O método emprega uma população P_t e um arquivo externo $\overline{P_t}$. No arquivo externo $\overline{P_t}$. são armazenadas as soluções não-dominadas até o momento. São fornecidos como parâmetros iniciais do algoritmo o tamanho da população P_t, denominado N, o tamanho do arquivo externo $\overline{P_t}$, denominado $\overline{N}$, e o número máximo de gerações, dado por T.

O algoritmo inicia criando uma população inicial P_0 e um arquivo externo $\overline{P_0}$ inicialmente vazio. A cada iteração, a função de *fitness* é calculada para cada solução $i \in P_t \cup \overline{P_t}$. Em seguida, executa-se a seleção: as soluções não-dominadas de $i \in P_t \cup \overline{P_t}$ são copiadas para o novo arquivo externo $\overline{P_{t+1}}$. Se o conjunto de soluções não-dominadas tem o tamanho exato do arquivo externo, a seleção está finalizada. Senão, pode-se ter duas situações:

• Tamanho do arquivo externo menor que o número de soluções não-dominadas. Neste caso, ordenam-se as soluções de $P_t \cup \overline{P_t}$, de acordo com a função de *fitness*, e preenche-se o arquivo externo com as soluções dominadas com maior valor da função de *fitness*.

• Tamanho do arquivo externo excede o tamanho $\overline{N}$. Nesse caso, é usado um algoritmo de corte.

Por último, realiza-se o processo de seleção por torneio com reposição, recombinação e mutação $\overline{P_{t+1}}$ sobre para se obter a nova população P_{t+1} (Azuma, 2011).

2.7. Síntese do Capítulo

Neste capítulo, foram apresentados os conceitos básicos de grafos e árvores, otimização mono e multiobjectivo e os principais algoritmos genéticos mono e multiobjetivo. Fez-se uma revisão bibliográfica de restauração e reconfiguração de redes de distribuição. Foi descrito também o algoritmo adotado nesse trabalho, o SPEA2.

Capítulo 3

Abordagem Proposta

Neste capítulo é apresentado a formulação do problema de otimização multiobjetivo a ser tratado que consiste na restauração do sistema de distribuição de energia elétrica, detalhando-se a arquitetura do algoritmo implementado.

3.1. Modelagem do Problema

O problema de restauração do sistema de distribuição de energia elétrica proposto foi modelado como um problema de otimização multiobjetivo, cujas funções objetivo são minimizar o tempo das manobras e a porcentagem de cargas não restauradas (1).

$$\min\{f_1, f_2\} \qquad (1)$$

onde:

f_1 = tempo de manobras (em minutos);

$f_2 = \sum \frac{Nu}{Nl} \times 100\%,$

onde:

$Nu =$ número de cargas não religadas;

$Nl =$ número total de cagas que foram desligadas devido a contingência.

A porcentagem de cargas não restauradas se refere somente aos blocos onde não ocorreram falhas. Dessa forma, é possível alcançar 100% de cargas restauradas em alguns casos. No entanto, essa taxa nem sempre é possível pois, devido à característica radial do sistema, podem não haver rotas entre a parte ativa do sistema e uma carga potencialmente restaurável.

Todas as soluções devem atender três restrições:

- topologia radial da parte ativa da rede;
- limite máximo de corrente nos condutores;
- limite mínimo de tensão nas barras de carga.

Existem dois limites para os valores de corrente e de tensão nas barras: o limite desejável, no qual a corrente deve ser menor ou igual a 1,05 p.u. e a tensão deve ser maior ou igual a 0,95 p.u.; e o limite aceitável, no qual a corrente máxima pode chegar até 1,10 p.u. e a tensão não pode ser menor que 0,90 p.u.. Soluções que se encontrem abaixo do limite aceitável são consideradas infactíveis e desconsideradas pelo algoritmo. Já soluções entre os limites aceitável e desejável têm seus valores de função penalizadas, mas são aceitáveis. A escolha dessa estratégia se baseia nos princípios: muitas vezes é melhor atender o cliente com menor qualidade que deixar de atendê-lo; e não se espera manter o sistema trabalhando por muito tempo em uma configuração considerada emergencial. Vale ressaltar que estes parâmetros são configuráveis, sendo possível desabilitar a aceitação de soluções abaixo do limite desejável caso esse seja o desejo do operador.

3.2. Arquitetura do algoritmo

Na arquitetura do algoritmo todo o instanciamento foi dividido em três camadas: camada C, constituída de todos os elementos do sistema detalhado (avaliação microscópica do sistema ou de baixo nível); camada B, constituída do sistema em blocos (avaliação macroscópica do sistema ou de alto nível); e a camada A, que é constituída dos parâmetros do algoritmo. A Figura 3.1 ilustra essa arquitetura em camadas.

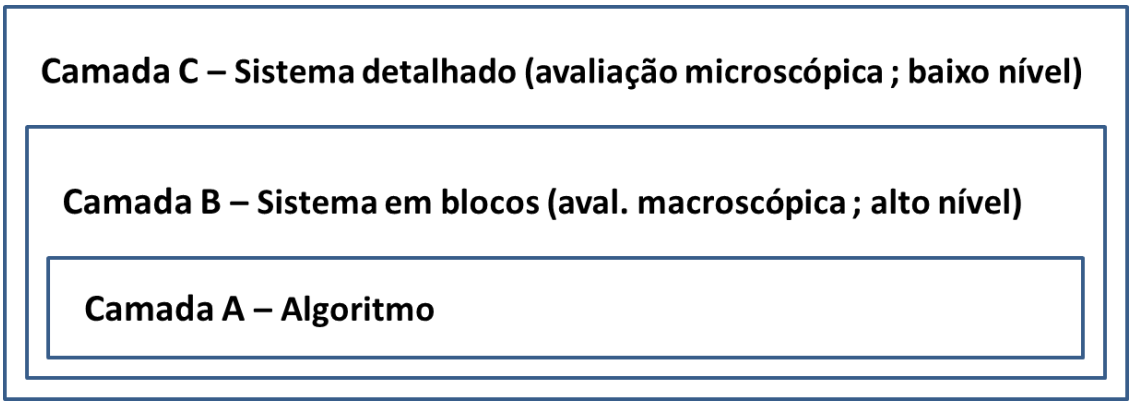

Figura 3.1. Arquitetura do algoritmo em camadas.

Essa divisão em camadas é de extrema importância pois, a alteração na estrutura da rede, como por exemplo a inclusão de geração distribuída (GD) não altera em nada as camadas mais internas, não sendo necessárias modificações no algoritmo.

A camada C, como mencionado anteriormente, é uma visão microscópica do sistema. Nela estão contidas as instâncias da rede: os tipos, carga, tensão, coordenadas das barras; capacidade das subestações; capacidade de outras fontes; o tipo, capacidade de corrente, impedância das linhas; o tipo, estado, capacidade de corrente, impedância das chaves, etc. A Figura 3.2 apresenta uma ilustração do sistema visto na camada C, nível de barras.

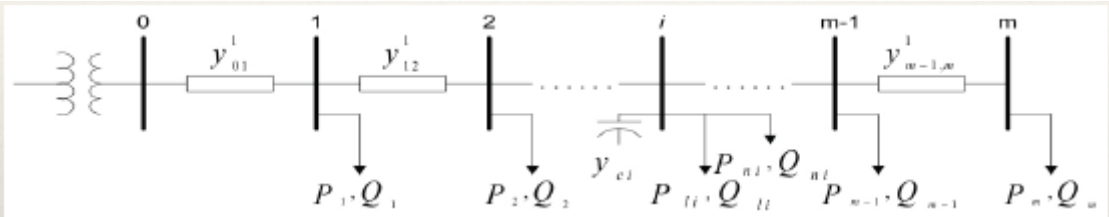

Figura 3.2. Sistema na camada C

Com o objetivo de facilitar a implementação do algoritmo e torná-lo mais eficaz, as barras entre chaves do sistema foram agrupadas em blocos. Os blocos são interligados através dessas chaves, cujo status é definido com base na condição inicial, ou seja, no cenário pré-falta, como chaves normalmente abertas (NA) e chaves normalmente fechadas (NF). As chaves são também classificadas como manobráveis e proteção, somente manobráveis e somente proteção. Esse sistema em blocos constitui a camada B. Nessa camada estão contemplados os equipamentos que atuaram na falta, as cargas dos blocos, os estados das chaves, os limites das correntes, e os limites das tensões nas barras. A Figura 3.3 apresenta uma ilustração do sistema visto na camada B, nível de blocos.

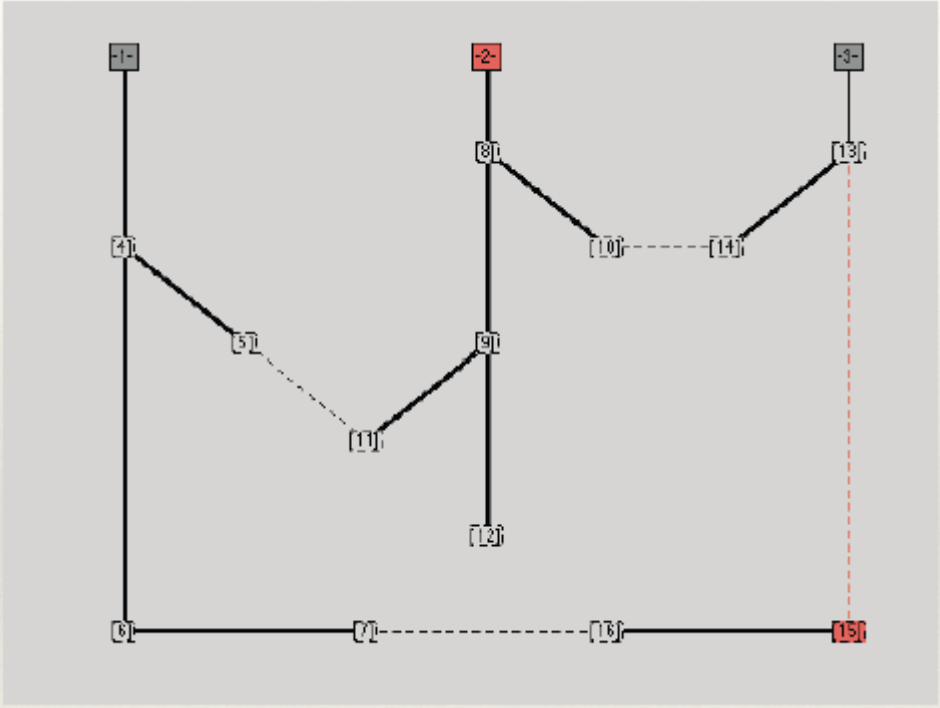

Figura 3.3. Sistema na camada B. A linha pontilhada indicam as chaves normalmente abertas (NA). E as linhas contínuas indicam as chaves normalmente fechadas (NF).

3.3. Classificação dos Blocos e Chaves

Na situação esperada, na qual a proteção está devidamente coordenada, a ocorrência de uma contingência implica na abertura automática da chave de proteção mais próxima da falta, no caminho entre esta e a origem do alimentador. Isso faz com que outras cargas que antes eram ligadas após o trecho da falha, no mesmo alimentador, também sejam desconectadas, uma vez que a parte ativa do sistema é geralmente radial.

Inicialmente, cada bloco é associado a um grupo correspondente ao alimentador a que está conectado (se o sistema possui 5 alimentadores, existem 5 grupos). No cenário pós falta, os grupos dos blocos são modificados, como segue:

- Blocos onde ocorreram as falhas são associados ao grupo 0.
- Blocos que foram desconectados do alimentador i são associados ao grupo –i (blocos não operacionais).
- Blocos que não foram afetados por nenhuma falha, chamados de blocos operacionais, mantém seu grupo original.

As chaves também são classificadas de acordo com o tipo de operação, como segue:

- Tipo 1 (aberta): indica a chave desligada pela proteção.
- Tipo 2 (fechada): indica a chave que conecta um bloco não-operacional com um bloco em falta, e deve ser aberta caso o bloco em falta seja religado.
- Tipo 3 (fechada): indica a chave que conecta dois blocos de mesmo grupo, operacionais ou não.
- Tipo 4 (aberta): indica a chave que conecta dois blocos de grupos operacionais distintos.
- Tipo 5 (aberta): indica a chave que conecta um bloco operacional e um bloco desconectado (chave de restauração).

Ao se religar um trecho desconectado deve-se abrir primeiro uma chave do tipo 2 para que a falta não seja realimentada, ou uma chave do tipo 3 a montante da chave do tipo 2 correspondente (interrupção antes da chave).

O objetivo de separar as chaves em tipos é criar subpermutações da permutação original. Uma vez que o importante para o algoritmo são as permutações relativas a cada grupo, o espaço de busca do problema é reduzido drasticamente.

Na condição de pós falta, chaves de manobras devem ser acionadas de forma a reconectar o maior número de consumidores possíveis respeitando as restrições de tensão, corrente e topologia do sistema.

3.4. Codificação e Decodificação

Para solucionar o problema de restauração foi proposto um algoritmo genético multiobjetivo, baseado no Strength Pareto Evolutionary Algorithm 2 (SPEA2) [Zitzler, Laumanns, Thiele, 2001], cujo fluxograma é ilustrado na Figura 3.4.

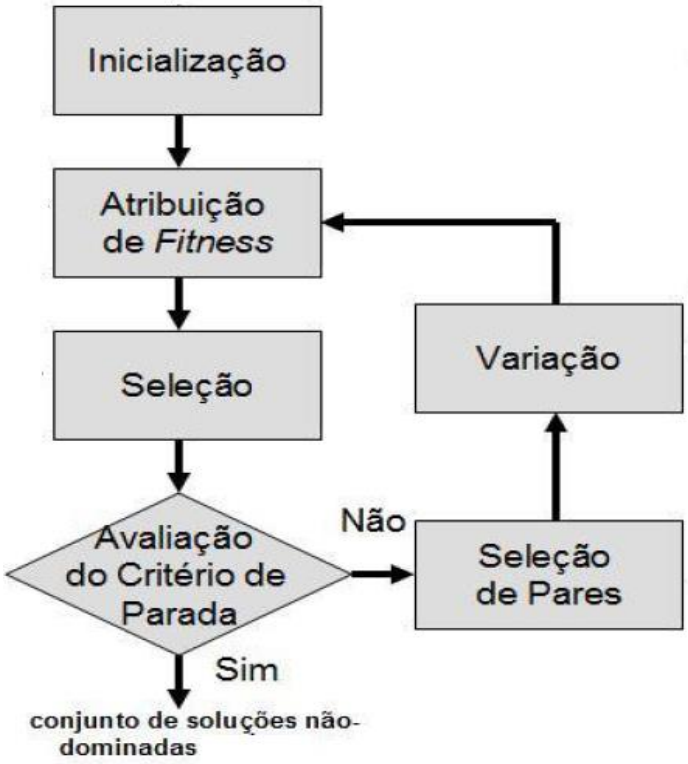

Figura 3.4. Fluxograma do algoritmo implementado.

As soluções do algoritmo são codificadas como permutações de NC elementos, onde NC é o número de chaves disponíveis para manobra. A permutação indica a posição relativa das manobras candidatas em cada tipo de operação.

O indivíduo é então decodificado em várias soluções, com manobras sequenciais, seguindo a permutação das chaves. A Figura 3.5 apresenta um pseudocódigo do processo de decodificação.

- (A) Para a primeira chave de operação tipo 5 manobrável no indivíduo:
    - (B) Feche a chave e remova da lista de chaves manobráveis.
    - (C) Abra a chave de operação tipo 2.
    - Execute o fluxo de carga.
    - Se a rede obtida em (C) obedece às restrições, então salve o indivíduo parcial
    - Caso Contrário:
        - Cancele a abertura realizada em (C).
        - Construa uma lista com todas as chaves entre a chave fechada em (B) e a tentativa de abertura em (C).
        - (D) Abra a próxima chave da lista e execute o fluxo de carga..
            - Se a rede obtida em (D) obedece às restrições, então salve o indivíduo parcial.
            - Caso contrário:
                - Cancele a abertura realizada em (D) e retorne ao passo (D).
    - Atualize o tipo de operação de cada chave.
    - Avalie a solução parcial utilizando a função objetivo.
    - Caso não seja atingido um critério de parada, retorne ao passo (a).

Figura 3.5. Pseudocódigo do processo de decodificação.

O algoritmo vai repetir o processo enquanto existirem chaves do tipo 5 manobráveis, ou seja, enquanto existirem chaves de restauração no sistema. Primeiramente uma chave de restauração manobrável é fechada. Dessa forma, como a falta provavelmente voltará a ser realimentada, uma chave do tipo 2 adequada será aberta para fazer o isolamento do trecho. Feito isso, o fluxo de carga é executado para verificar se a rede é ou não factível. Caso a rede atenda às restrições, o indivíduo será salvo. Caso não atenda uma ou mais restrições, será construída uma lista constituída de chaves do tipo 3 que estão a jusante da chave de restauração que foi fechada anteriormente. Dessa forma, serão feitas podas nas árvores que foram religadas. Isso se torna interessante pois, muitas vezes, não será possível realimentar um trecho inteiro de um alimentador fechando somente uma chave de restauração. O mesmo deverá ser reconectado ao sistema através do fechamento de uma segunda chave, garantindo a factibilidade da solução.

Caso a solução seja factível, as manobras redundantes são canceladas, o tipo de operação das chaves e os grupos são atualizados e as soluções avaliadas utilizando a função objetivo. Caso contrário, todas as alterações realizadas são canceladas e todo o processo será realizado utilizando a próxima chave de restauração manobrável.

Dessa forma, tem-se um indivíduo que decodifica em várias soluções sequenciais, ou seja, um indivíduo será decodificado em várias sequências de manobras a serem realizadas e cada uma delas terá um nível de restauração alcançado.

Para exemplificar o processo de decodificação, considere o sistema teste de 16 blocos (Zhu, 2002) ilustrado na Figura 3.6.

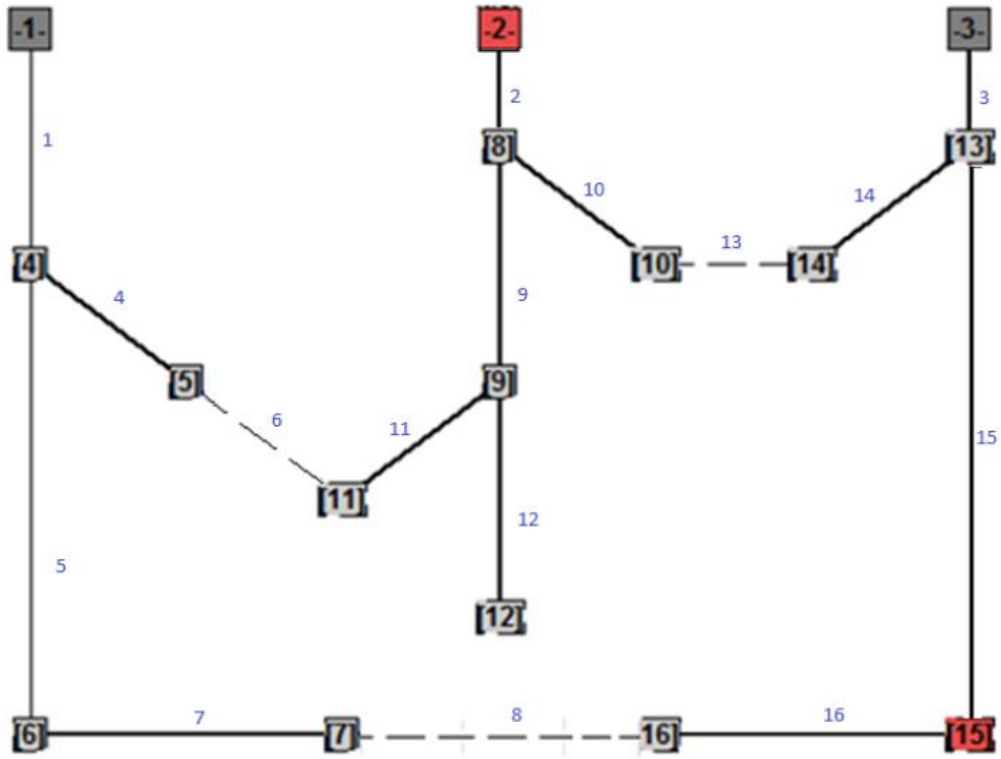

Figura 3.6. Esquema ilustrativo para o sistema teste de 16 blocos.

As chaves foram indexadas da seguinte forma:

- 1: chave que liga o bloco 1 ao 4 (1-4)
- 2: chave que liga o bloco 2 ao 8 (2-8)
- 3: chave que liga o bloco 3 ao 13 (3-13)
- 4: 4-5
- 5: 4-6
- 6: 5-11
- 7: 6-7
- 8: 7-16
- 9: 8-9
- 10: 8-10
- 11: 9-11
- 12: 9-12
- 13: 10-14
- 14: 13-14
- 15: 13-15
- 16:15-16

Um exemplo de indivíduo pode ser escrito como uma permutação dessas 16 chaves:

$$I = [14\ 10\ 13\ 16\ 7\ 6\ 3\ 12\ 5\ 11\ 9\ 4\ 1\ 15\ 2\ 8]$$

Foram simuladas duas falhas no sistema, sendo uma no bloco 2 e outra no 15. Após a ocorrência das falhas, o sistema de proteção atua abrindo a chave 13-15 e isolando o alimentador 2. A Tabela 1 apresenta a classificação dos tipos das chaves para o exemplo.

Tabela 1 – Tipos das chaves para o sistema teste de 16 blocos.

TIPO	Chaves
1	15
2	16, 2
3	14, 10, 7, 3, 12, 5, 11, 9, 4, 1
4	-
5	13, 6, 8

A ordem das chaves apresentada na Tabela 1 é a ordem em que elas aparecem no indivíduo original e que será seguida pelo algoritmo. Dessa forma, o importante não é a ordem em que elas aparecem no indivíduo com um todo, mas sim como elas são ordenadas nos seus respectivos grupos.

A primeira manobra a ser testada, na primeira iteração, será com a chave de restauração 13. Fechando a chave 13, o bloco 2 volta a ser realimentado e a chave 2 será aberta.

É realizada a atualização das chaves, onde as mesmas são reprocessadas quanto ao tipo, como apresentado na Tabela 2, onde o número sublinhado indica que a chave já foi manobrada.

Tabela 2 – Tipos das chaves para o sistema teste de 16 blocos após a primeira iteração.

TIPO	Chaves
1	15, 2
2	16
3	14, 10, 13, 7, 3, 12, 5, 11, 9, 4, 1,
4	6
5	8

Seguindo a ordem das chaves, na segunda iteração, será fechada a chave de restauração 8. Assim, o bloco 15, em falta, volta a ser realimentado e a chave 16 é aberta. Novamente é realizada a atualização das chaves e o reprocessamento das mesmas quanto ao tipo, como apresentado na Tabela 3.

Tabela 3 – Tipos das chaves para o sistema teste de 16 blocos após a segunda iteração.

TIPO	Chaves
1	16, 2
2	15
3	14, 10, 13, 7, 3, 12, 5, 11, 9, 4, 1, 8
4	6
5	

A Figura 3.7 ilustra as manobras executadas nas duas iterações.

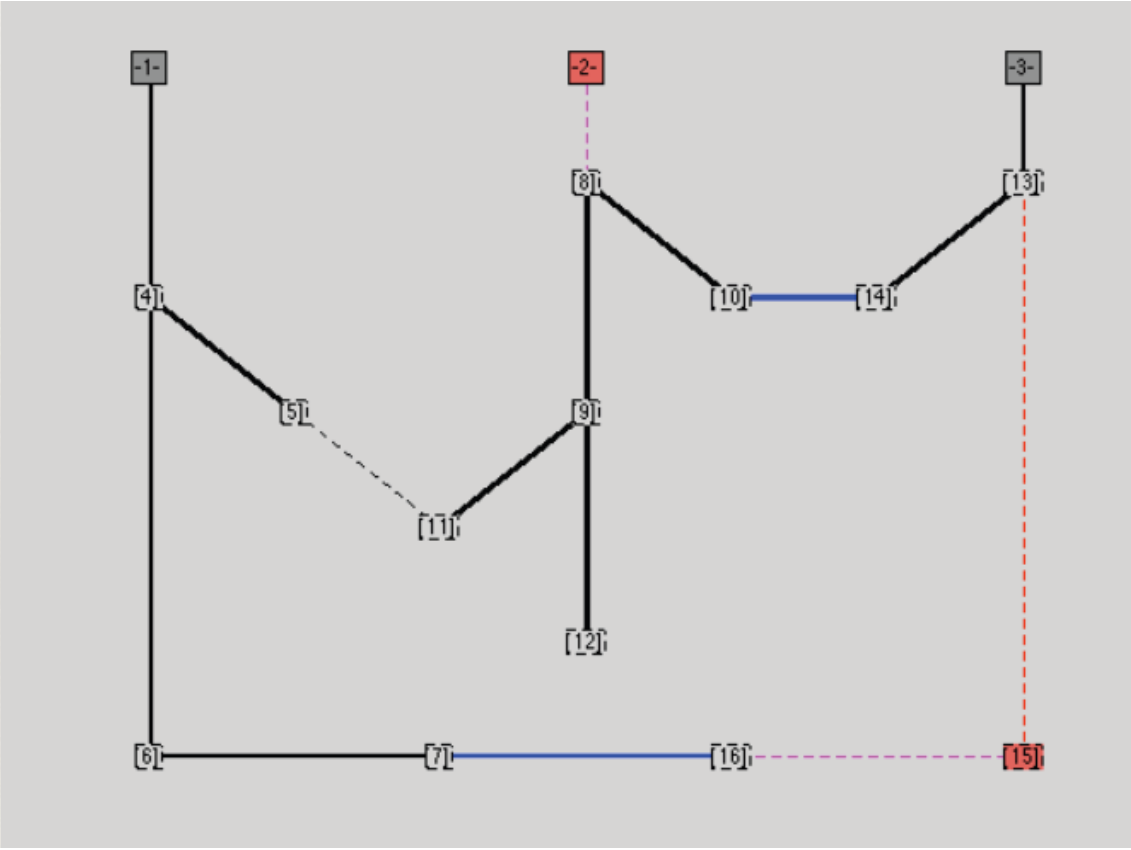

Figura 3.7. Exemplo de solução para o SD de 16 blocos. A linha pontilhada vermelha indica a chave que foi aberta pela atuação da proteção. As linhas pontilhadas magentas indicam as chaves abertas durante as manobras. E as linhas contínuas azuis, indicam as chaves fechadas durante as manobras.

O indivíduo foi decodificado em 2 soluções: a primeira com duas manobras, restaurando 87% da carga recuperável; e a segunda com quatro manobras (duas da primeira solução mais duas adicionais, característica da segunda solução) restaurando 100% da carga recuperável.

Através do mecanismo de decodificação, um mesmo indivíduo pode se decodificar em várias soluções. Essas soluções são sequenciais, ou seja, a manobra da solução três, por exemplo, só poderá ser executada após a execução das manobras indicadas pelas soluções um e dois.

As soluções são apresentadas ao decisor que irá escolher, de acordo com a conveniência da situação, qual é a melhor. Ele pode optar também por não realizar todas as manobras, tendo em vista que muitas vezes isso implicará em um tempo gasto maior e um aumento pequeno na porcentagem de cargas restauradas. Como a proposta do algoritmo é, não só minimizar a carga desconectada ao fim do conjunto de manobras, mas também minimizar a carga desconectada ao fim de cada iteração, o decisor terá certeza que a sequência de manobras apresentada é a que recuperará a maior quantidade de carga.

Para ilustrar o que foi mencionado anteriormente, consideremos dois indivíduos decodificados em 7 soluções, mostradas na Figura 3.8. Ambas as soluções apresentam o mesmo resultado final, recuperam 100% das cargas recuperáveis. Entretanto, nota-se que a área abaixo da curva de f_2 é menor que a da curva de f_1. Portanto, o indivíduo 2, a cada iteração, recupera mais carga que o indivíduo 1, acumulando assim menos DIC.

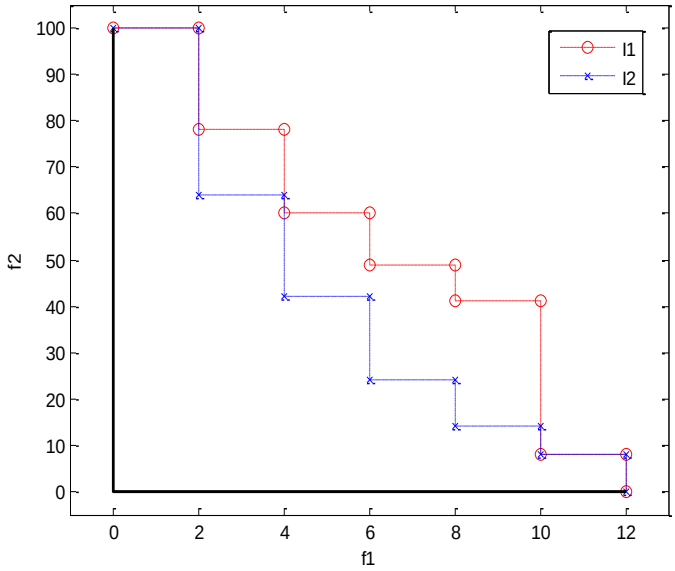

Figura 3.8. Esquema ilustrativo de Fronteira Pareto de dois indivíduos.

No fluxo do algoritmo, inicialmente é criada uma população de N indivíduos aleatórios. O arquivo externo é inicialmente vazio, entretanto, durante a evolução deverá conter também N indivíduos. Cada indivíduo da população é avaliado utilizando o processo de decodificação e sua *fitness* final é calculada pela soma ponderada do valor da "*raw fitness"* ($\mathrm{R^n}$), que é a soma das forças dos indivíduos que dominam o outro, e do seu indicador de diversidade ($\mathrm{D^n}$), que é a distância para todos os indivíduos, como em (2).

$$F(I) = \alpha R^n(I) + (1-\alpha) * D^n(I) \qquad (2)$$

onde:

$$R^n(I) = 1 - \left[\frac{R(I) - min(R)}{max(R) - min(R)}\right];$$

$$D^n(I) = \left[\frac{D(I) - min(D)}{max(D) - min(D)}\right];$$

$$\alpha = 0{,}5 = parâmetro\ de\ diversificação.$$

onde:

$$R(I) = \sum_{i=1}^{N(I)} \sum_{j \in P, j \prec i} S(j)$$

$$D(I) = \sum_{i}^{N(I)} dis(I, I_i)$$

onde:

$$P = conjunto\ de\ indivíduos\ da\ população.$$

$$N(I) = número\ de\ soluções\ contidas\ pelo\ indivíduo\ I.$$

$$S = cardinalidade\ dos\ conjuntos\ de\ soluções\ que\ dominam\ S$$

$$S(I) = |\{j | j\ \in P \wedge j \prec i\}|;$$

A distância entre dois indivíduos A e B ($\mathrm{dis}(\mathrm{A},\mathrm{B})$), é definida pela soma das diferenças no estado das chaves ao fim da execução de cada conjunto de manobras (utilizando distância *Hamming*), até atingir o total de estágios de A ou B, dependendo de qual for maior.

A fim de exemplificar o cálculo da distância, considere dois indivíduos A e B. O indivíduo A contém três soluções e o B quatro, conforme a decodificação mostrada na Figura 3.9.

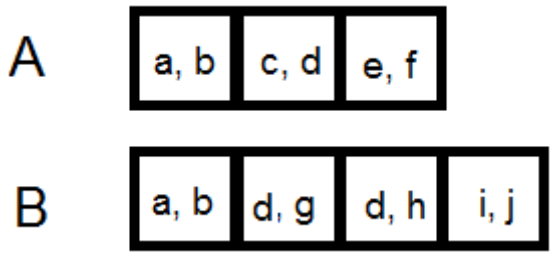

Figura 3.9 – Decodificação de dois indivíduos.

Para encontrar a distância entre A e B, $\mathrm{dis}(\mathrm{A},\mathrm{B})$, primeiramente compara-se os estados das chaves, ao fim de cada conjunto de manobras. A priori não existem diferenças, pois a configuração pós falta é a mesma para ambas. Para o primeiro estágio, não há diferença entre as soluções, pois as manobras são as mesmas, fazendo com que a distância seja zero. Para o segundo estágio, há uma diferença, logo a distância é 1. Na terceira posição, há diferença nas duas soluções, a distância, então, será 2 desse estágio acrescido de 1 acumulado do estágio anterior, resultando em 3. Logo a distância total, $\mathrm{dis}(\mathrm{A},\mathrm{B})$ é a soma das distâncias de cada estágio individualmente (0+1+3) resultando em 4.

A diversidade do indivíduo A, D(A), é a soma das distâncias entre ele e os outros indivíduos, conforme (3).

$$D(A) = dis(A,B) + dis(A,C) + dis(A,D) \quad (3)$$

3.5. Seleção, cruzamento e mutação

A seleção dos indivíduos é feita por meio de torneio binário (Chakraborty, 1997). Comparam-se duas soluções: se uma é não dominada em relação à população atual e a outra é dominada, a primeira é escolhida. Caso ambas as soluções sejam dominadas ou não dominadas, é escolhida aquela que possuir melhor valor de *fitness*. Esse procedimento é repetido até que uma nova população de N indivíduos seja criada.

Os indivíduos são, então, divididos em pares e, para cada par, é verificado se ocorrerá ou não cruzamento com base em uma probabilidade *pcr*, dada a priori. O operador de cruzamento utilizado é baseado no *cycle crossover* (Yoshikawa, 2008) a fim de melhorar o desempenho do algoritmo e evitar a geração de permutações infactíveis. Esse operador foi concebido para o problema do caixeiro viajante, como segue.

No problema do caixeiro viajante, dado um conjunto de n cidades e os custos de viagem entre cada par de cidades, o caixeiro deve visitar cada uma e, finalmente, voltar para a cidade de partida, com a menor distância possível. Modelando os indivíduos do GA como permutações, cada indivíduo consiste em uma ordem de cidades a serem visitadas. Os cruzamentos clássicos aplicados a este tipo de representação geralmente levam à soluções infactíveis, como pode ser visto a seguir. Sejam dois indivíduos factíveis, X e Y, apresentados na Figura 3.10 (Yoshikawa, 2008). A aplicação do cruzamento de um ponto a essas rotas geralmente leva à rotas infactíveis, como pode ser visto na Figura 3.11 (Yoshikawa, 2008).

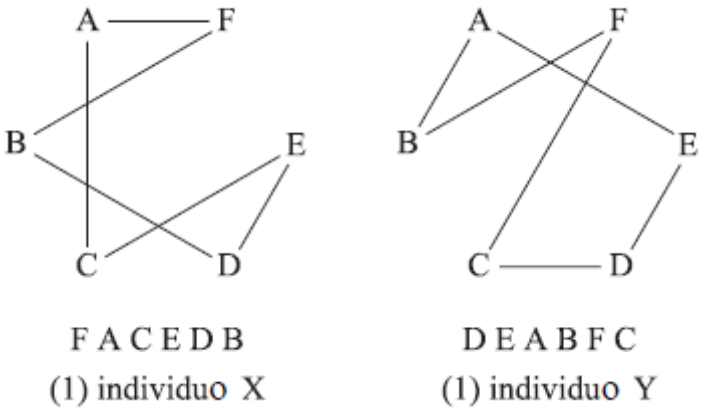

Figura 3.10 – Exemplo de decodificação das cidades.

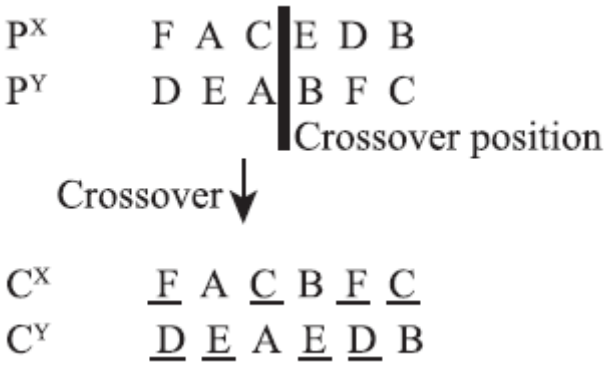

Figura 3.11 – Exemplo de permutação infactível.

A fim de evitar essas soluções infactíveis e melhorar o desempenho do algoritmo, Oliver et al. [7] propôs o *cycle crossover*, que consiste em repetir o processamento que examina a qual gene dos outros pais cada gene corresponde. A arquitetura proposta cria a tabela de genes correspondentes de cada pai como um pré-processamento e armazena-o em registro.

O *cycle crossover* realiza operações de cruzamento trocando genes incluídos em um ciclo mutante. Na Figura 3.12 (Yoshikawa, 2008) é mostrado um exemplo de como é feito o processamento do operador. Primeiramente, o gene que está na posição 1 do pai X ocupará a posição 1 do filho X. A posição 1 é armazenada na lista de registros. Num segundo passo, o gene do pai X que é igual ao gene na posição 1 do pai Y é examinada, esse gene ocupará no filho a mesma posição em que aparece no pai X e sua posição é guardada na lista de registros. Posteriormente, repete-se o processo para as demais posições.

O segundo filho é gerado de forma semelhante, basta trocar o pai X pelo pai Y.

locus	1	2	3	4	5	6
P^X	F	A	C	E	D	B
P^Y	D	E	A	B	F	C
C^X	F	-	-	-	-	-

(1) $i = 1, l = \{ 1 \}$

locus	1	2	3	4	5	6
P^X	F	A	C	E	D	B
P^Y	D	E	A	B	F	C
C^X	F	-	-	-	D	-

(2) $i = 5, l = \{ 1, 5 \}$

locus	1	2	3	4	5	6
P^X	F	A	C	E	D	B
P^Y	D	E	A	B	F	C
C^X	F	E	A	B	D	C

(3) $C^X{}_k = P^X{}_k \quad (k \notin l)$

Figura 3.12 – Exemplo de procedimento de *cycle crossover*. Figura adaptada de (Yoshikawa, 2008).

Na aplicação proposta, este operador de cruzamento é aplicado em partes da permutação que representa o indivíduo original. São considerados três subgrupos. :

- Subgrupo 1: chaves tipos 2 e 3
- Subgrupo 2: chaves tipo 4;
- Subgrupo 3: chaves tipo 5;

As permutações resultantes são reinseridas nos pais nas mesmas posições originais (a nova permutação do subgrupo 1 ocupa as posições originalmente ocupadas pela permutação do subgrupo 1 e assim por diante).

Sejam os indivíduos apresentados na Tabela 4. Aplicando o cycle crossover para cada um dos sub-grupos, têm-se as proles apresentadas na Tabela 5.

Tabela 4 – Exemplo de indivíduo subdividido em grupos conforme os tipos de chaves para o sistema teste de 16 blocos.

Subgrupo	PAI X	PAI Y
1	2,16, 3, 4, 14, 11, 12, 7, 9, 1, 10, 5	2, 16, 11, 3, 5, 10, 4, 1, 12, 7, 9, 14
2	-	-
3	13, 8, 6	8, 6, 13

Tabela 5 – Exemplo de indivíduo subdividido em grupos conforme os tipos de chaves para o sistema teste de 16 blocos.

Subgrupo	FILHO X	FILHO Y
1	14, 4, 1, 3, 16, 9, 11, 12, 7, 10, 5, 2	7, 2, 5, 3, 9, 11, 4, 1, 10, 12, 16, 14
2	-	-
3	13, 8, 6	13, 8, 6

Após os ciclos de cruzamento, cada indivíduo da população de filhos pode sofrer mutação, cuja probabilidade de ocorrência *pmut* também é dada a priori. Caso ocorra, a mutação pode ser do tipo *swap* ou *reverse* (Suliman, 2013).

Na mutação de *swap* dois genes são escolhidos aleatoriamente e trocados de lugar. Seja o exemplo de indivíduo do caixeiro viajante da Figura 3.13, no qual os genes A e D foram escolhidos para a mutação.

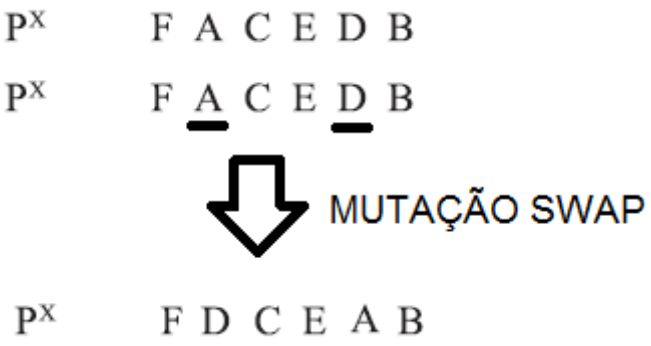

Figura 3.13 – Exemplo de mutação *Swap*.

Da mesma forma, na mutação *reverse* dois genes são escolhidos aleatoriamente. Entretanto a troca de posição ocorre entre todos os genes que estão entre os dois escolhidos. Seja o mesmo exemplo de indivíduo do caixeiro viajante apresentado anteriormente. A Figura 3.14 ilustra a mutação *reverse,* no qual os mesmo dois genes, A e D foram escolhidos para a mutação.

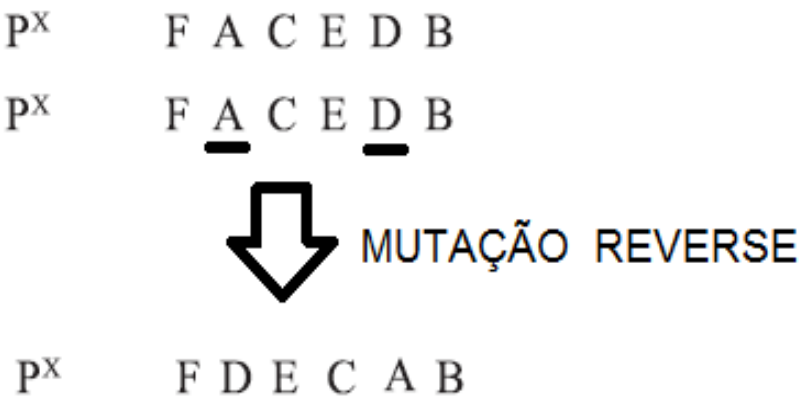

Figura 3.14 – Exemplo de mutação *Reverse*.

No escopo deste trabalho, estes operadores de mutação são executados para as sub-permutações definidas pelos mesmos grupos considerados no cruzamento. A cada operação, uma, duas ou três mutações serão executadas, sendo que uma operação ocorre necessariamente no sub-grupo 3.

Após seleção, cruzamento e mutação, o arquivo externo é então atualizado. O arquivo criado é composto das soluções não-dominadas e das soluções dominadas com melhor *fitness*. Caso o número de soluções não-dominadas seja maior que o arquivo, as N soluções não-dominadas com melhor *fitness* são copiadas para o arquivo.

Foram considerados dois critérios de parada, sendo que o primeiro a ocorrer implica no fim do algoritmo: número máximo de avaliações de função e tempo limite de execução.

O algoritmo gera como resposta um arquivo contendo uma aproximação do conjunto Pareto, que contém várias soluções, no qual cada uma delas contém um conjunto de manobras e um percurso a ser seguido.

3.6. Síntese do Capítulo

Neste capítulo foi apresentado o problema de otimização tratado e descrito os passos da implementação do algoritmo utilizado, o SPEA2.

Capítulo 4

Resultados e Discussões

Neste capítulo são apresentados e comentados os resultados de restauração para um sistema teste de 16 blocos (Zhu, 2002) e para dois sistemas de grande porte da Cemig Distribuição. Em ambos os casos são simuladas faltas em alguns blocos do sistema.

O sistema teste de 16 blocos é utilizado para mostrar que a ferramenta proposta funciona, devido à facilidade ilustrativa dos blocos. Serão analisados o número de manobras a serem realizadas e a quantidade de carga que as mesmas conseguem recuperar.

O primeiro sistema da Cemig, chamado de Cemig A é composto por 703 barras agrupadas em 121 blocos, 132 linhas, 5 alimentadores, 116 chaves NF e 16 chaves NA. O segundo sistema, Cemig B, é composto por 484 barras agrupadas em 125 blocos, 168 linhas, 5 alimentadores, 120 chaves NF e 48 NA. Em ambos os casos são analisados o tempo estimado para a realização das manobras, tendo em vista que há diferença entre as chaves telecomandadas e as manuais, e a quantidade de carga recuperada.

Nos dois sistemas são analisados três exemplos de conjuntos de falhas. Para o terceiro exemplo de cada um, foram utilizadas duas topologias distintas, ou seja, dois cenários aleatórios distintos.

4.1. Sistema de 16 Blocos

A Figura 4.1 apresenta um esquema ilustrativo do sistema de 16 blocos, onde 1, 2 e 3 são alimentadores.

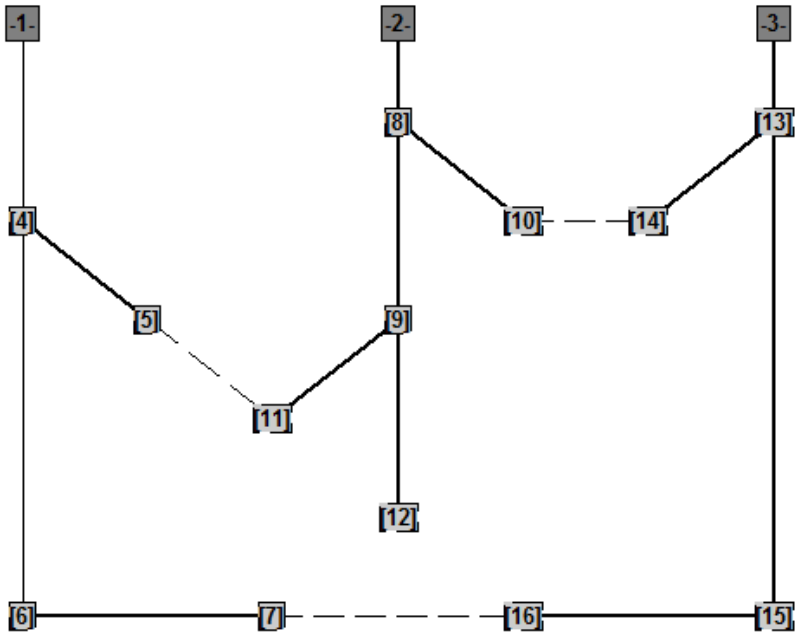

Figura 4.1. Esquema ilustrativo para o sistema de 16 blocos.

Foram simuladas duas falhas no sistema, sendo uma na barra 2 e outra na 15. Após a ocorrência das falhas, o sistema de proteção atua abrindo a chave 13-15 e isolando o alimentador 2. A Figura 4.2 apresenta a fronteira Pareto encontrada pelo algoritmo. Essa fronteira contém um indivíduo que decodifica em duas soluções, uma com duas e outra com quatro manobras.

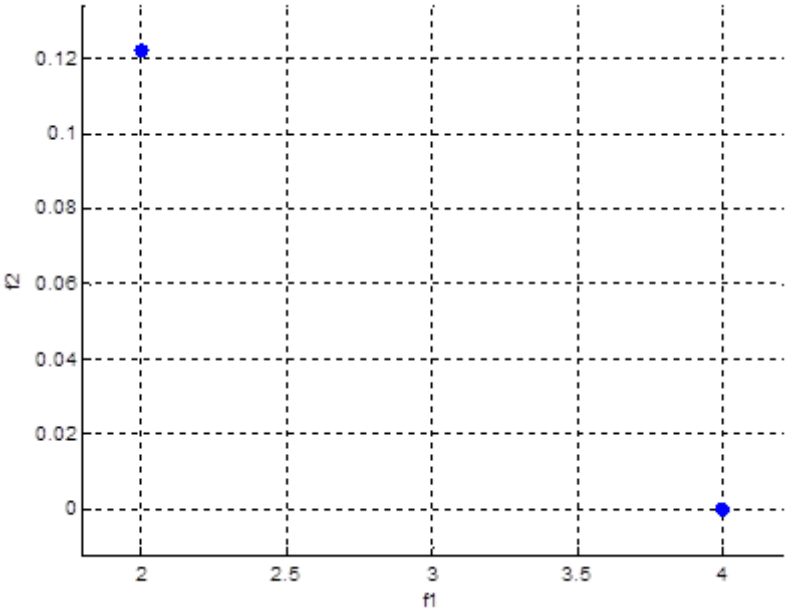

Figura 4.2. Aproximação da fronteira Pareto para o sistema de 16 blocos.

A Figura 4.3 mostra o indivíduo que contém as duas soluções da fronteira Pareto. Nessa figura, as linhas pontilhadas vermelhas indicam as chaves que foram abertas pela atuação da proteção. As linhas pontilhadas magentas indicam as chaves abertas durante as manobras. Já as linhas contínuas azuis, indicam as chaves fechadas durante as manobras. Os blocos vermelhos indicam as falhas e os blocos escuros os blocos não restaurados.

Esse indivíduo contém duas soluções sequenciais, ou dois estágios. No primeiro estágio, onde são realizadas duas manobras, a chave 2-8 abre e a chave 5-11 fecha, restaurando 88% das cargas desconectadas. No segundo estágio, onde são realizadas mais duas manobras (totalizando quatro com as duas realizadas no primeiro estágio), a chave 13-15 abre e a chave 7-16 fecha, eliminando o impacto das falhas.

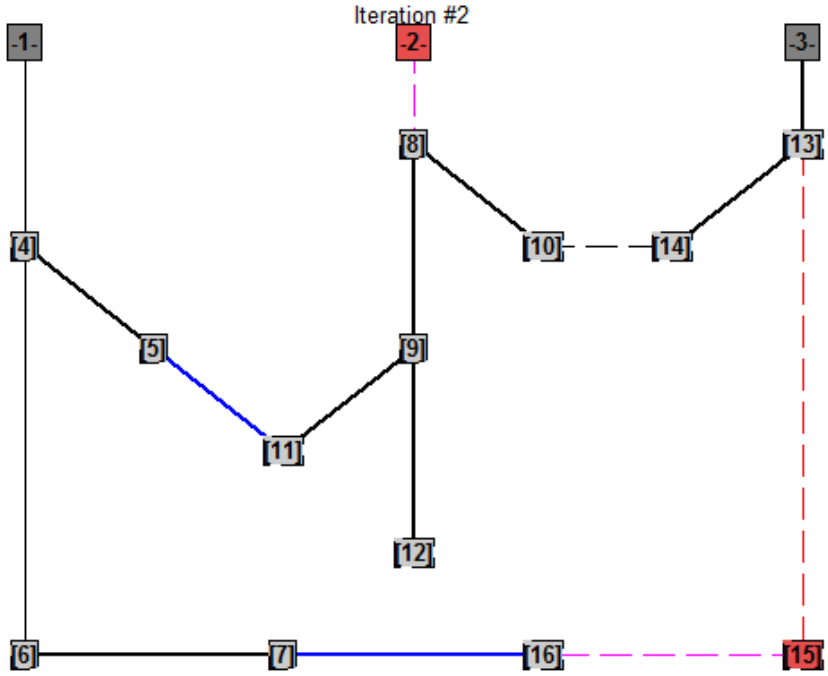

Figura 4.3. Exemplo de solução para o SD de 16 blocos. A linha pontilhada vermelha indica a chave que foi aberta pela atuação da proteção. As linhas pontilhadas magentas indicam as chaves abertas durante as manobras. E as linhas contínuas azuis, indicam as chaves fechadas durante as manobras. Os blocos vermelhos indicam as falhas.

4.2. Sistema Cemig A

A Figura 4.4 apresenta um esquema ilustrativo do sistema da Cemig A, onde os blocos 1, 14, 56, 90 e 101 são alimentadores. Vale destacar que a ordenação dos blocos nessa figura foi feita de forma hierárquica, sem qualquer aderência com suas posições no plano cartesiano.

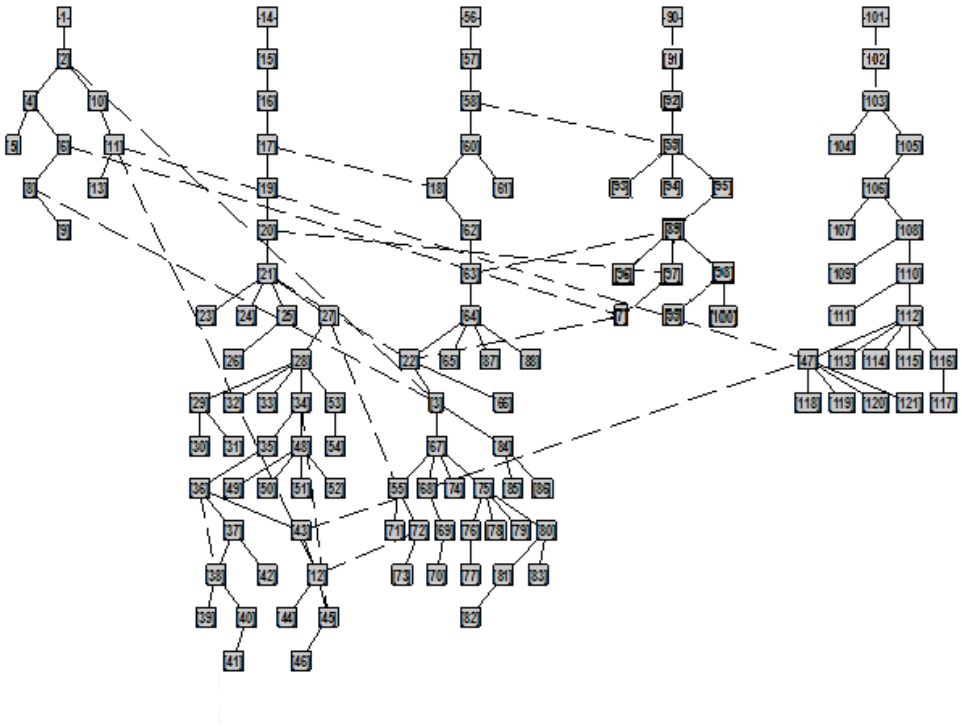

Figura 4.4. Esquema ilustrativo para o sistema Cemig A.

4.2.1. Exemplo 1 - Sistema Cemig A

No primeiro exemplo do sistema Cemig A foram simuladas duas falhas, sendo uma no bloco 91 e outra no bloco 102. A Figura 4.5 apresenta a fronteira Pareto encontrada pelo algoritmo. Essa fronteira contém dois indivíduos que decodificam em duas soluções, e dois indivíduos que decodificam em três soluções.

Para ilustração das manobras escolheu-se o indivíduo que se decodifica em duas soluções sendo que a primeira delas recupera aproximadamente 68% das cargas reconectáveis em 16 minutos e a segunda recupera totalmente o sistema em 46 minutos.

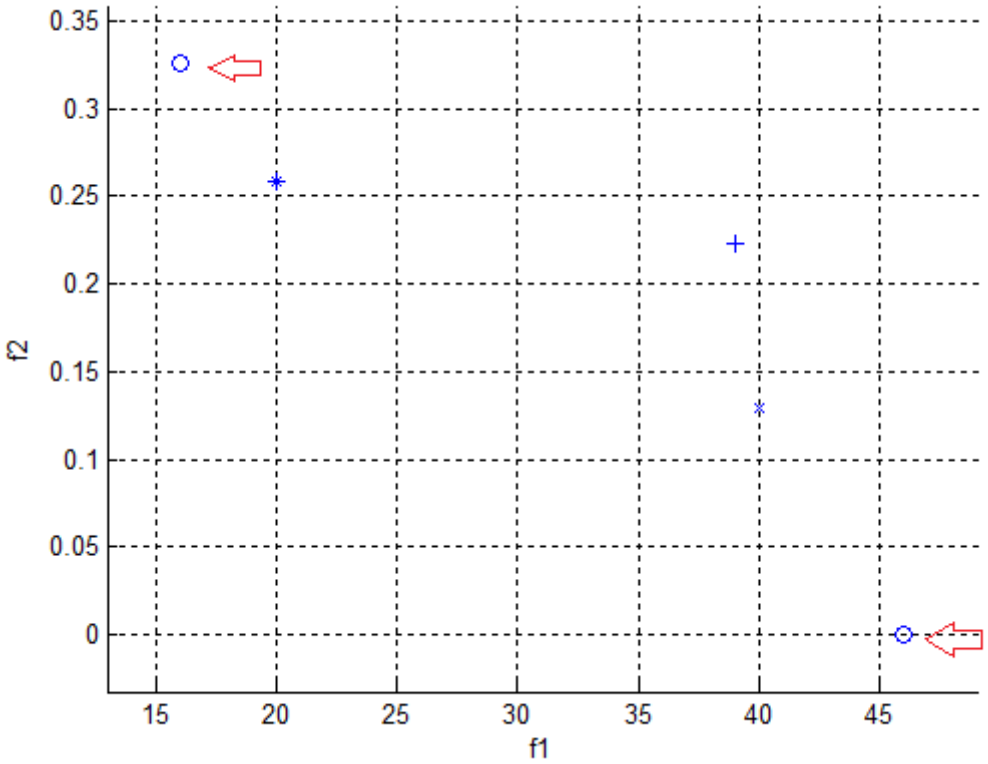

Figura 4.5. Aproximação da fronteira Pareto para o sistema da Cemig A do exemplo 1. As setas indicam as soluções do indivíduo escolhido para a ilustração do exemplo.

Após a ocorrência das falhas, o sistema de proteção atua abrindo as chaves 101-102 e 90-91 isolando os alimentadores 101 e 90, respectivamente. A Figura 4.6 mostra o primeiro estágio do indivíduo mencionado anteriormente, onde são realizadas duas manobras, a chave 102-103 abre e a chave 11-47 fecha. No segundo estágio, apresentado na Figura 4.7, são realizadas mais duas manobras (totalizando quatro com as duas realizadas no primeiro estágio), a chave 81-82 abre e a chave 6-7 fecha, reduzindo o impacto das falhas para 0%.

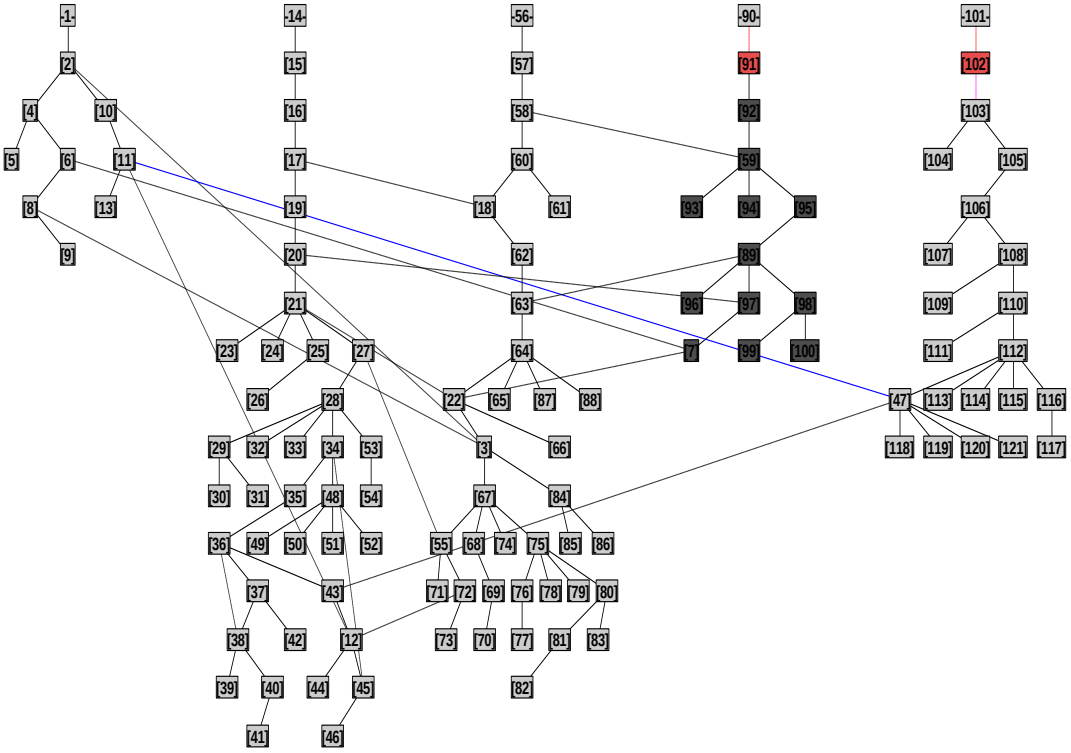

Figura 4.6. Exemplo de solução para o SD da Cemig A Exemplo 1 – primeiro estágio.

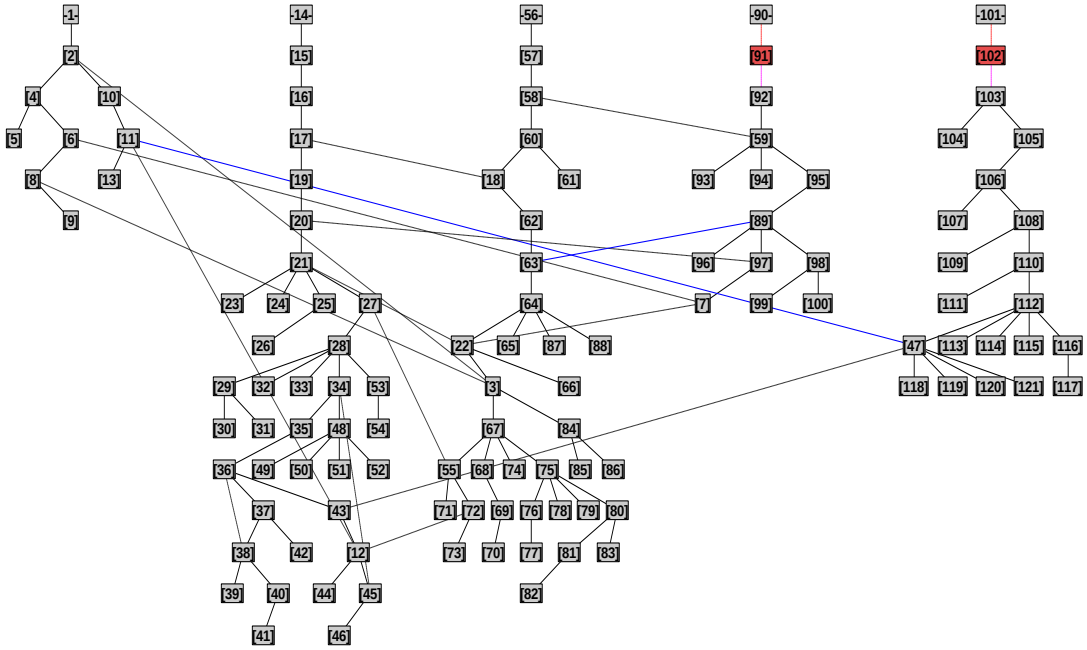

Figura 4.7. Exemplo de solução para o SD da Cemig A Exemplo 1 – segundo estágio.

4.2.2. Exemplo 2 - Sistema Cemig A

Neste segundo exemplo do sistema Cemig A, foi simulada uma única falha, no bloco 15. A Figura 4.8 apresenta a fronteira Pareto encontrada pelo algoritmo. Essa fronteira contém dois indivíduos que decodificam em duas soluções, e um indivíduo que decodifica em uma solução.

Para ilustração das manobras realizadas escolheu-se o indivíduo que se decodifica em uma única solução recuperando 100% das cargas reconectáveis em 27 minutos.

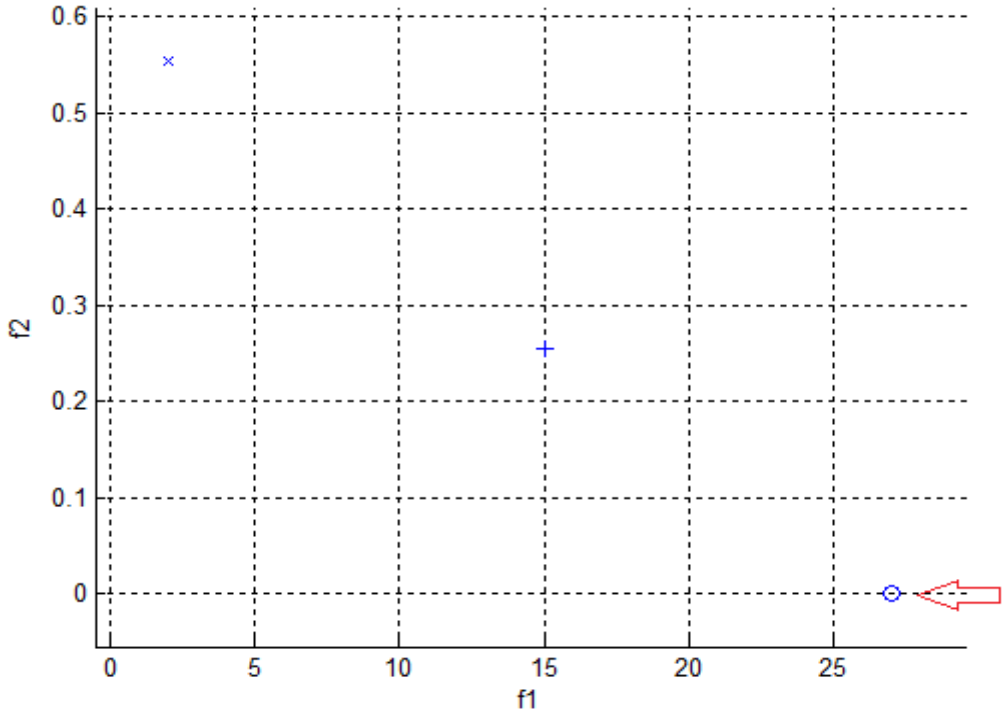

Figura 4.8. Aproximação da fronteira Pareto para o sistema da Cemig A do exemplo 2. A seta indica a solução do indivíduo escolhido para a ilustração do exemplo.

A Figura 4.9 apresenta o sistema após a restauração. Após a ocorrência da falha, o sistema de proteção atua abrindo a chave 14-15 isolando o alimentador 14, a chave 15-16 abre e a chave 20-96 fecha.

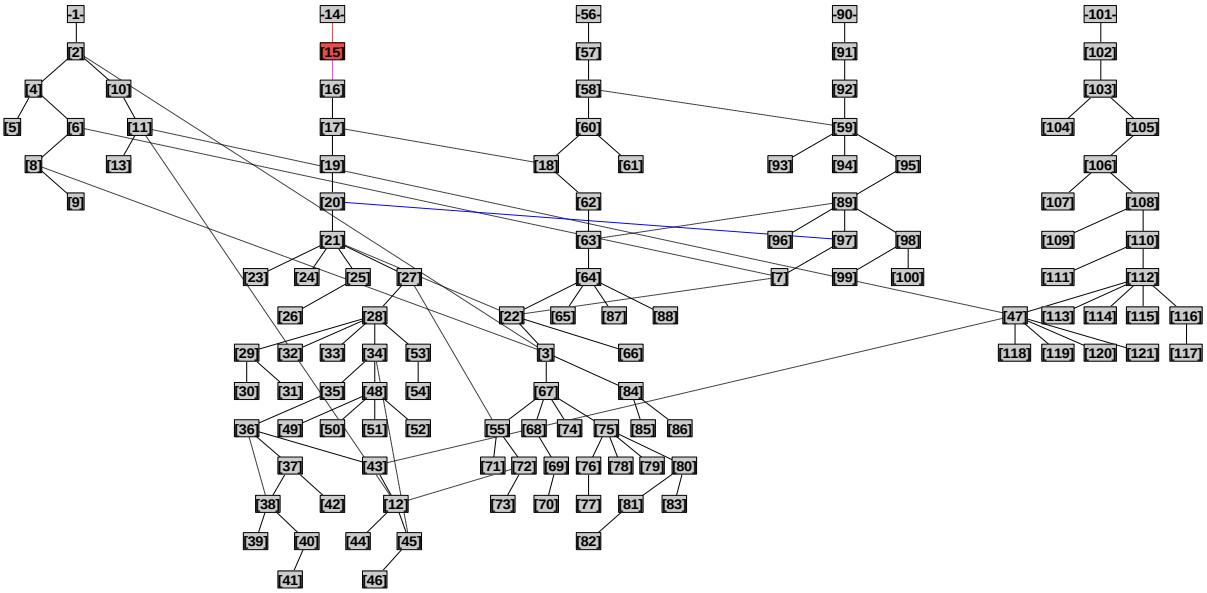

Figura 4.9. Exemplo de solução para o SD da Cemig A Exemplo.

4.2.3. Exemplo 3 – Topologia 1 - Cemig A

O terceiro exemplo do sistema Cemig A partiu inicialmente da geração de duas topologias aleatórias factíveis para a rede e, posteriormente, a simulação de sete falhas, nos blocos 27, 32, 37, 91, 102, 117 e 121. É importante ressaltar que, neste exemplo, foi considerado que a localização das falhas não é conhecida a priori (simulando uma notificação por telefone por exemplo). Com isso, é necessário considerar que todos os blocos dentro do trecho de proteção da falha estão também com falha, fazendo com que essa se expanda.

A Figura 4.10 apresenta a fronteira Pareto encontrada pelo algoritmo para a topologia 1. Essa fronteira contém treze indivíduos, sendo que dois se decodificam em três soluções, três decodificam em quatro soluções, cinco em cinco soluções e três decodificam em seis soluções.

Para ilustração das manobras escolheu-se o indivíduo que se decodifica em cinco soluções. Na Tabela 6 estão indicadas a porcentagem de carga restaurada e o tempo gasto para cada uma das soluções.

Tabela 6 – Resultado para o exemplo 3 do sistema Cemig A, topologia 1.

SOLUÇÃO	CARGA RESTAURADA (%)	TEMPO (min)
1	34	16
2	46	29
3	59	59
4	77	87
5	79	144

Neste exemplo não é possível recuperar todas as cargas. Isso ocorre devido à ausência de rota para esses blocos.

Interessante ressaltar que para esse exemplo o decisor pode optar por não realizar a última manobra, tendo em vista que o tempo estimado para sua execução quase dobra quando comparada à quarta solução aumentando apenas 2% da carga final restaurada. Nesse caso a concessionária pode preferir focar na solução do problema que ocasionou o problema.

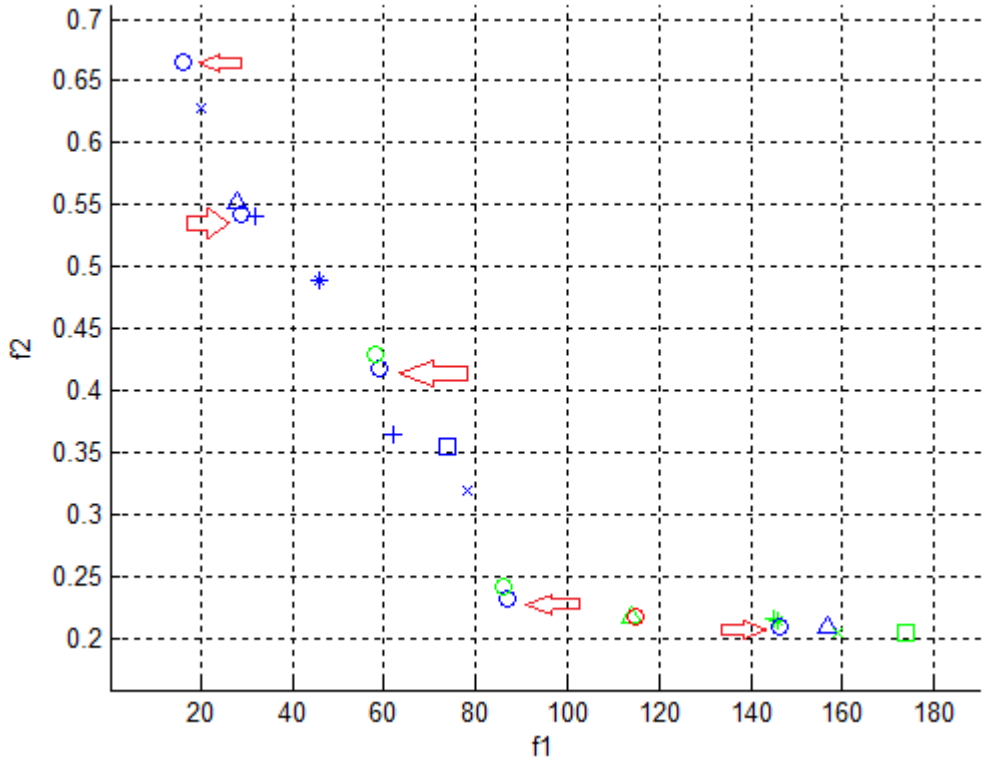

Figura 4.10. Aproximação da fronteira Pareto para o sistema da Cemig A do exemplo 3, topologia 1. As setas indicam as soluções do indivíduo escolhido para a ilustração do exemplo.

A Figura 4.11 apresenta o sistema após a restauração. Após a ocorrência das falhas, o sistema de proteção atua abrindo as chaves 14-15 isolando o alimentador 14, 90-91 isolando o alimentador 90 e a chave 101-102 isolando o alimentador 101.

A Tabela 7 apresenta as chaves que foram abertas e fechadas em cada estágio.

Tabela 7 – Chaves abertas e fechadas por estágio para o exemplo 3 do sistema Cemig A, topologia 1.

Estágio	Chave(s) Aberta(s)	Chave(s) Fechada(s)
1	102-103	11-47
2	59-95	6-7
3	36-43	12-72
4	27-28	34-45

5	59-92, 59-93 e 59-94	59-58

A abertura de mais de uma chave no último estágio ocorre devido à violação de alguma das restrições.

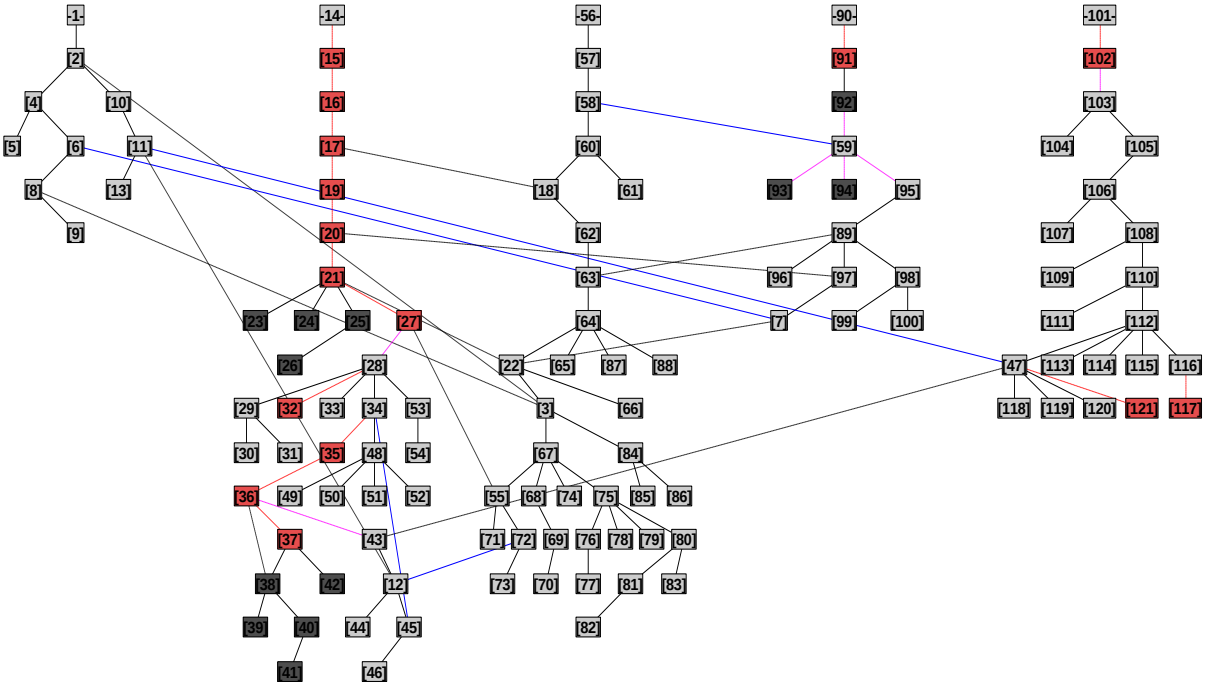

Figura 4.11. Exemplo de solução para o SD da Cemig A Exemplo 3, topologia 1.

4.2.4. Exemplo 3 – Topologia 2 - Cemig A

A Figura 4.12 apresenta a fronteira Pareto encontrada pelo algoritmo para a topologia 2. Essa fronteira contém onze indivíduos, sendo que dois se decodificam em três soluções, quatro decodificam em quatro soluções, três em cinco soluções e dois decodificam em seis soluções.

Para ilustração das manobras escolheu-se o indivíduo que se decodifica em quatro soluções. Na Tabela 8 estão indicadas a porcentagem de carga restaurada e o tempo gasto para cada uma das soluções.

Tabela 8 – Resultado para o exemplo 3 do sistema Cemig A, topologia 2.

SOLUÇÃO	CARGA RESTAURADA (%)	TEMPO (min)
1	34	16
2	46	46
3	65	74
4	79	129

Para essa topologia também não é possível recuperar todas as cargas devido à ausência de rota para esses blocos.

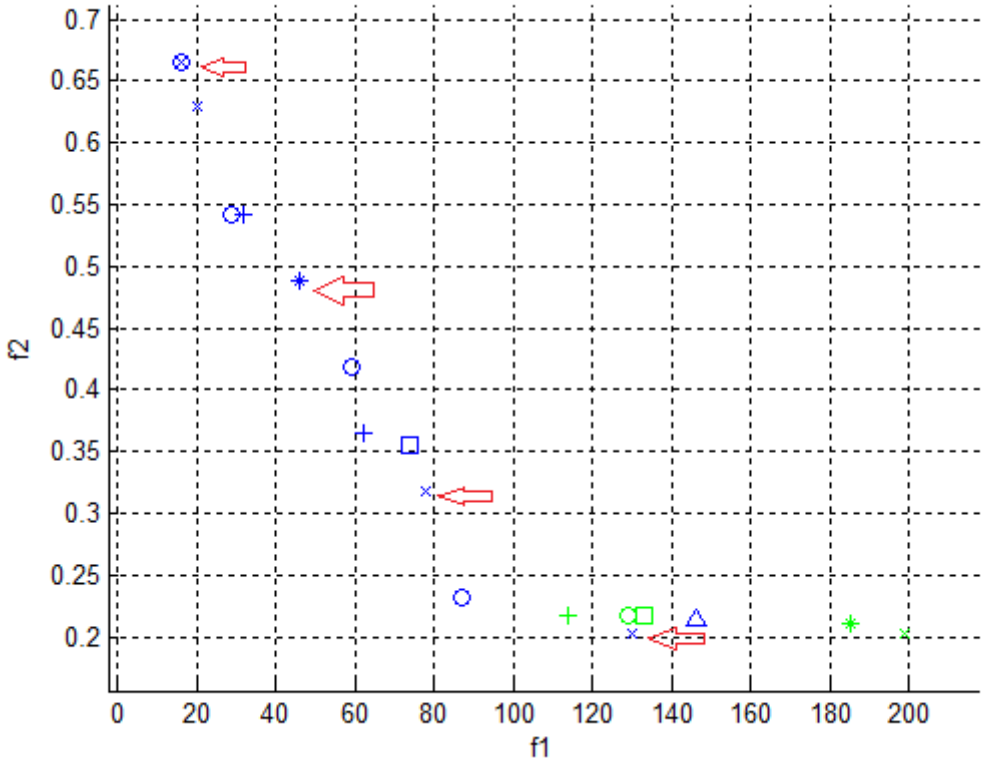

Figura 4.12. Aproximação da fronteira Pareto para o sistema da Cemig A do exemplo 3, topologia 2. As setas indicam as soluções do indivíduo escolhido para a ilustração do exemplo.

A Figura 4.13 apresenta o sistema após a restauração. Após a ocorrência das falhas, o sistema de proteção atua abrindo as chaves 14-15 isolando o alimentador 14, 90-91 isolando o alimentador 90 e a chave 101-102 isolando o alimentador 101.

A Tabela 9 apresenta as chaves que foram abertas e fechadas em cada estágio.

Tabela 9 – Chaves abertas e fechadas por estágio para o exemplo 3 do sistema Cemig A, topologia 2.

Estágio	Chave(s) Aberta(s)	Chave(s) Fechada(s)
1	102-103	11-47
2	36-43	12-72
3	27-28	34-45
4	59-92, 59-93 e 59-94	6-7

A abertura de mais de uma chave no último estágio ocorre devido à violação a alguma das restrições.

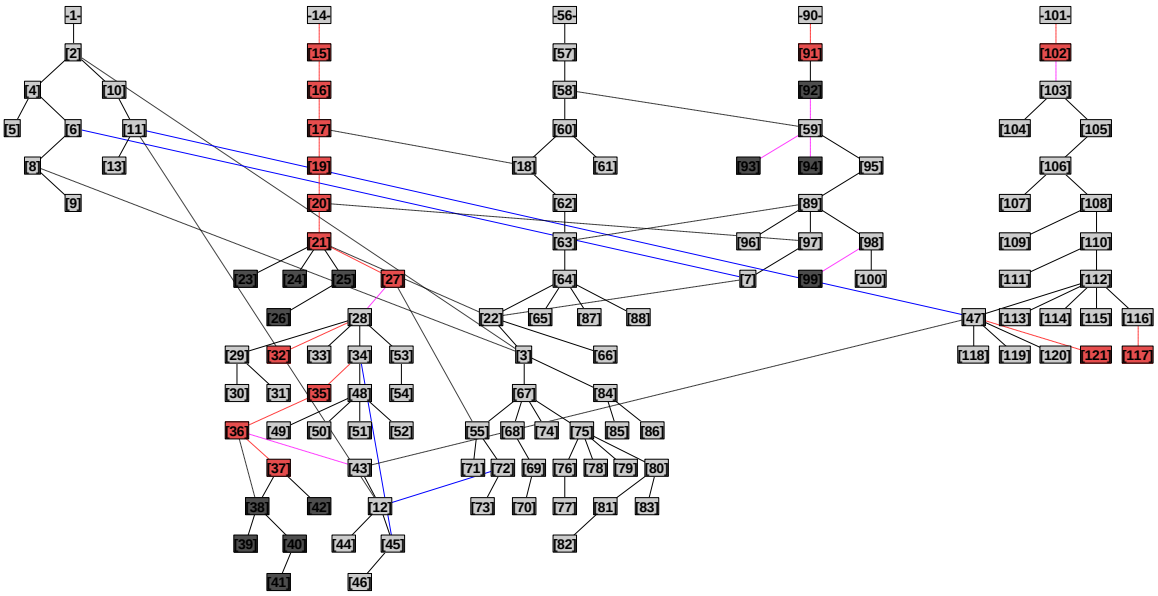

Figura 4.13. Exemplo de solução para o SD da Cemig A Exemplo 3, topologia 2.

4.3. Sistema Cemig B

O sistema Cemig B representa uma área de alta prioridade o que justifica a existência de muitas chaves NA. A Figura 4.14 apresenta um esquema ilustrativo do sistema da Cemig B, onde os blocos 1, 12, 50, 84 e 118 são alimentadores.

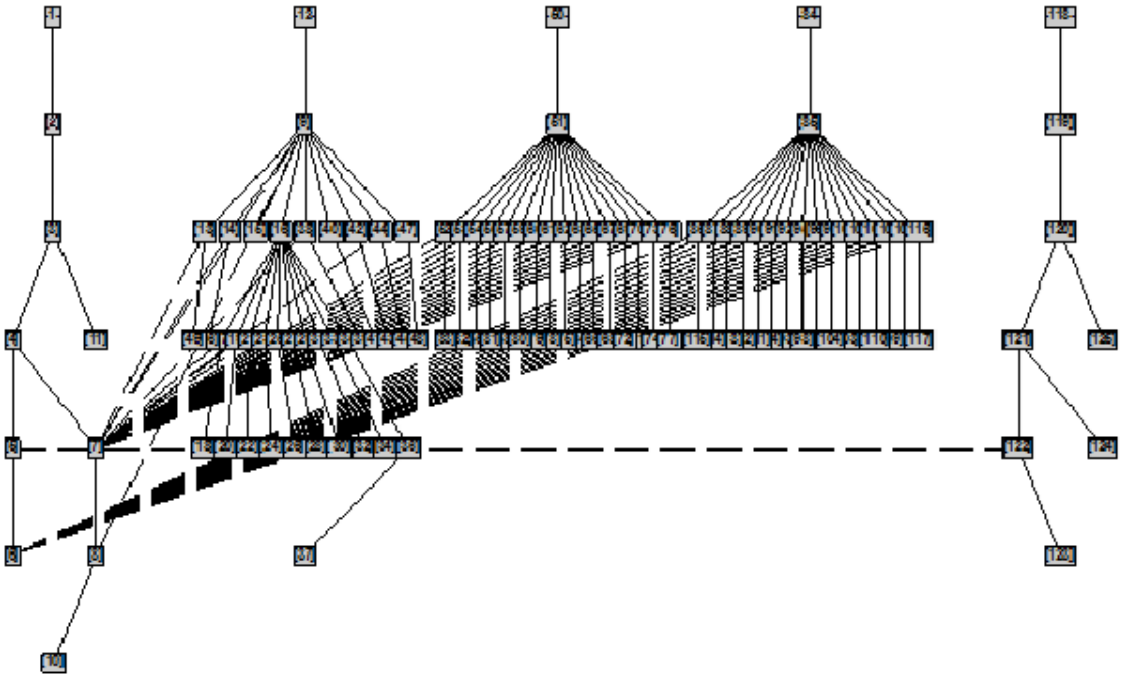

Figura 4.14. Esquema ilustrativo para o sistema Cemig B.

4.3.1. Exemplo 1 - Sistema Cemig B

No primeiro exemplo do sistema Cemig B foi simulada uma única falha, no bloco 2. A Figura 4.15 apresenta a fronteira Pareto encontrada pelo algoritmo. Essa fronteira contém um indivíduo que decodifica em uma solução.

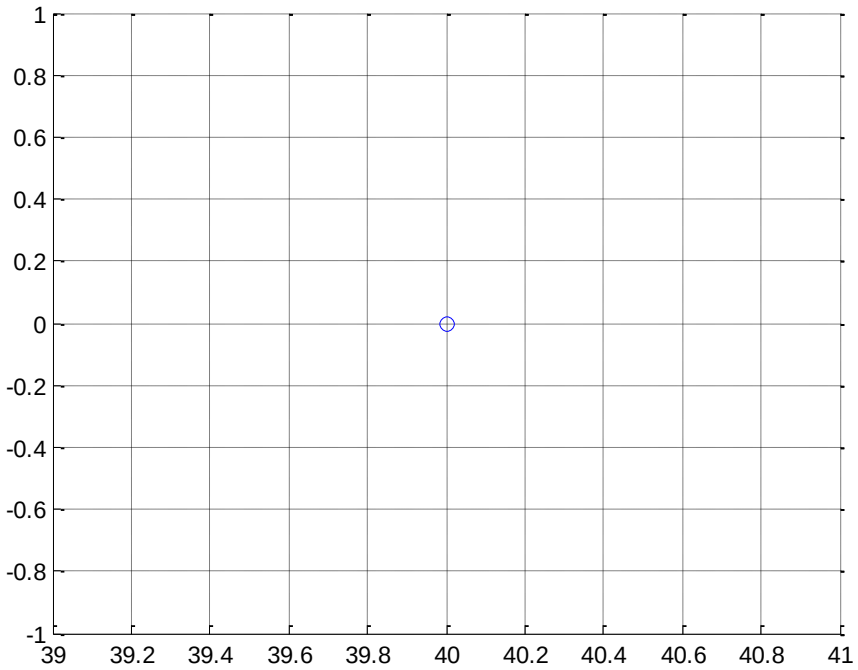

Figura 4.15. Aproximação da fronteira Pareto para o sistema da Cemig B - exemplo 1.

A Figura 4.16 apresenta o sistema após a restauração. Após a ocorrência da falha, o sistema de proteção atua abrindo a chave 1-2 isolando o alimentador 1 e, então, a chave 2-3 abre e a chave 7- 60 fecha.

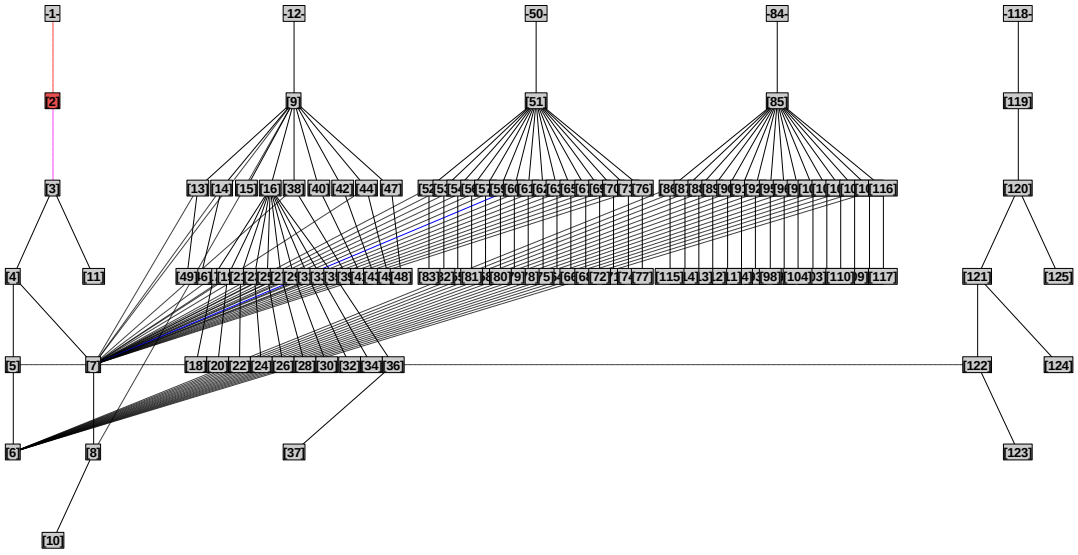

Figura 4.16. Exemplo de solução para o SD da Cemig B Exemplo.

4.3.2. Exemplo 2 - Sistema Cemig B

No segundo exemplo do sistema Cemig B foi simulada uma única falha, no bloco 51. A Figura 4.17 apresenta a fronteira Pareto encontrada pelo algoritmo. Essa fronteira contém vinte e cinco indivíduos que se decodificam em dezesseis soluções cada.

Para ilustração das manobras escolheu-se o indivíduo que se decodifica em 16 soluções. Na Tabela 10 estão indicadas a porcentagem de carga restaurada e o tempo gasto para cada uma das soluções.

Tabela 10 – Resultado para o exemplo 2 do sistema Cemig B.

SOLUÇÃO	CARGA RESTAURADA (%)	TEMPO (min)
1	15	24
2	26	50
3	36	78
4	46	107
5	52	137
6	59	178
7	64	219
8	69	250
9	71	273
10	77	311
11	81	354
12	85	384
13	91	425
14	92	447
15	96	493
16	100	545

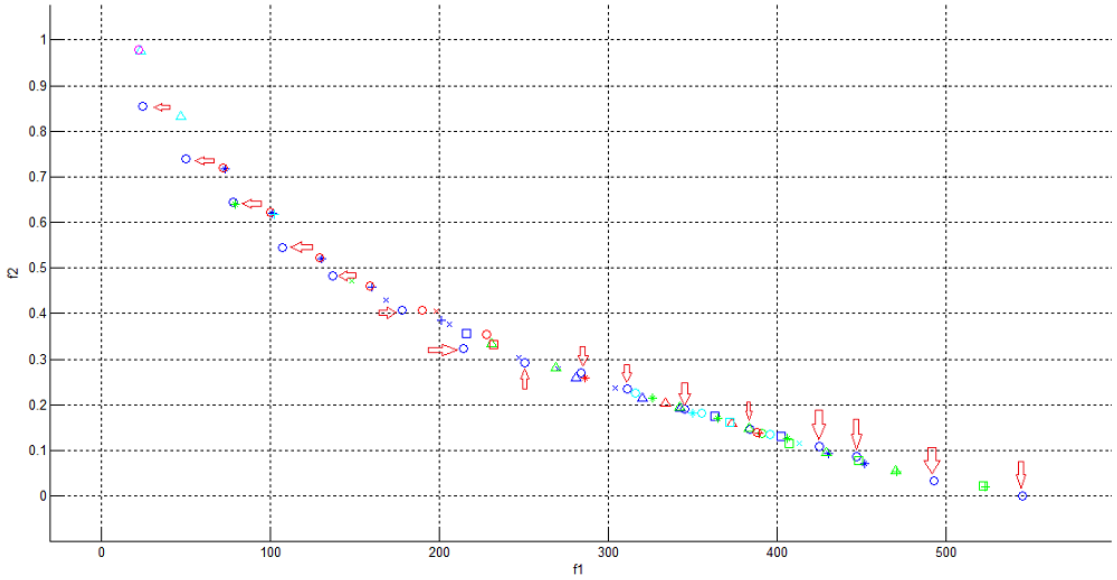

Figura 4.17. Aproximação da fronteira Pareto para o sistema da Cemig B do exemplo 2. As setas indicam as soluções do indivíduo escolhido para a ilustração do exemplo.

A Figura 4.18 apresenta o sistema após a restauração. Após a ocorrência das falhas, o sistema de proteção atua abrindo a chave 50-51 isolando o alimentador 50.

A Tabela 11 apresenta as chaves que foram abertas e fechadas em cada estágio.

Tabela 11 – Chaves abertas e fechadas por estágio para o exemplo 2 do sistema Cemig B.

Estágio	Chave(s) Aberta(s)	Chave(s) Fechada(s)
1	51-59	7 - 59
2	51 – 57	7 – 57
3	51 – 76	7 – 76
4	51 – 62	7 – 62
5	51 – 54	7 – 54
6	51 – 53	7 – 53
7	51 – 75	7 – 75
8	51 – 56	7 – 56
9	51 – 61	7 – 61
10	51 – 63	7 – 63
11	51 – 70	7 – 70
12	51 – 65	7 – 65
13	51 – 69	7 – 69

14	51 – 60	7 – 60
15	51 – 67	7 – 67
16	51 – 52	7 – 52

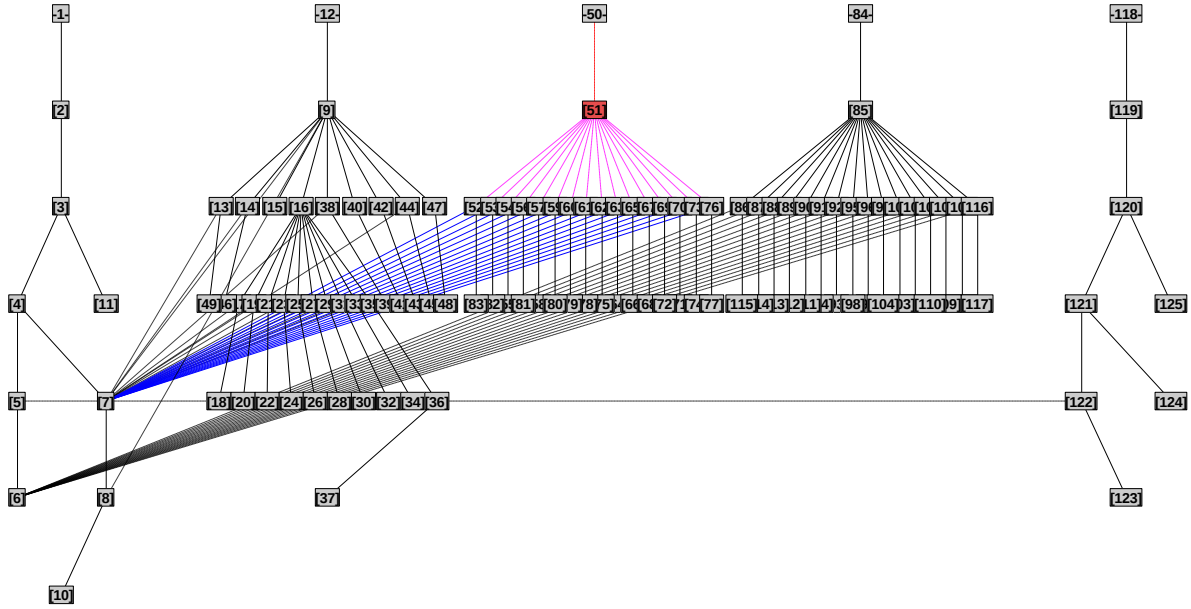

Figura 4.18. Exemplo de solução para o SD da Cemig B Exemplo.

4.3.3. Exemplo 3 – Topologia 1 - Cemig B

No terceiro exemplo do sistema Cemig B foram geradas duas topologias iniciais aleatórias e factíveis e posteriormente foram simuladas duas falhas, nos blocos 9 e 85. Nesse caso também foi considerado que a localização das falhas não é exatamente conhecida a priori.

A Figura 4.19 apresenta a fronteira Pareto encontrada pelo algoritmo para a topologia 1. Essa fronteira contém vinte e três indivíduos que se decodificam em vinte e uma soluções.

Para ilustração das manobras escolheu-se o indivíduo que Para ilustração das manobras escolheu-se o indivíduo que se decodifica em 21 soluções. Na Tabela 12 estão indicadas a porcentagem de carga restaurada e o tempo gasto para cada uma das soluções.

Tabela 12 – Resultado para o exemplo 3 do sistema Cemig B, topologia 1.

SOLUÇÃO	CARGA RESTAURADA (%)	TEMPO (min)
1	3	37
2	6	85
3	8	111
4	44	137
5	46	183
6	40	208
7	43	243
8	48	276
9	55	314
10	59	351
11	60	391
12	61	425
13	66	457
14	73	504
15	76	549
16	79	573
17	80	605
18	83	649
19	86	680
20	87	716
21	82	763

Neste exemplo não é possível recuperar todas as cargas, isso ocorre devido à ausência de rota para esses blocos.

É interessante ressaltar que para esse exemplo o decisor pode optar por não realizar todas as manobras tendo em vista que o tempo estimado para sua execução aumenta muito enquanto a quantidade de carga restaurada varia pouco. Dessa forma, a equipe de campo pode conseguir chegar ao local da falta e resolver o problema mais rapidamente do que a realização de todas as manobras indicadas.

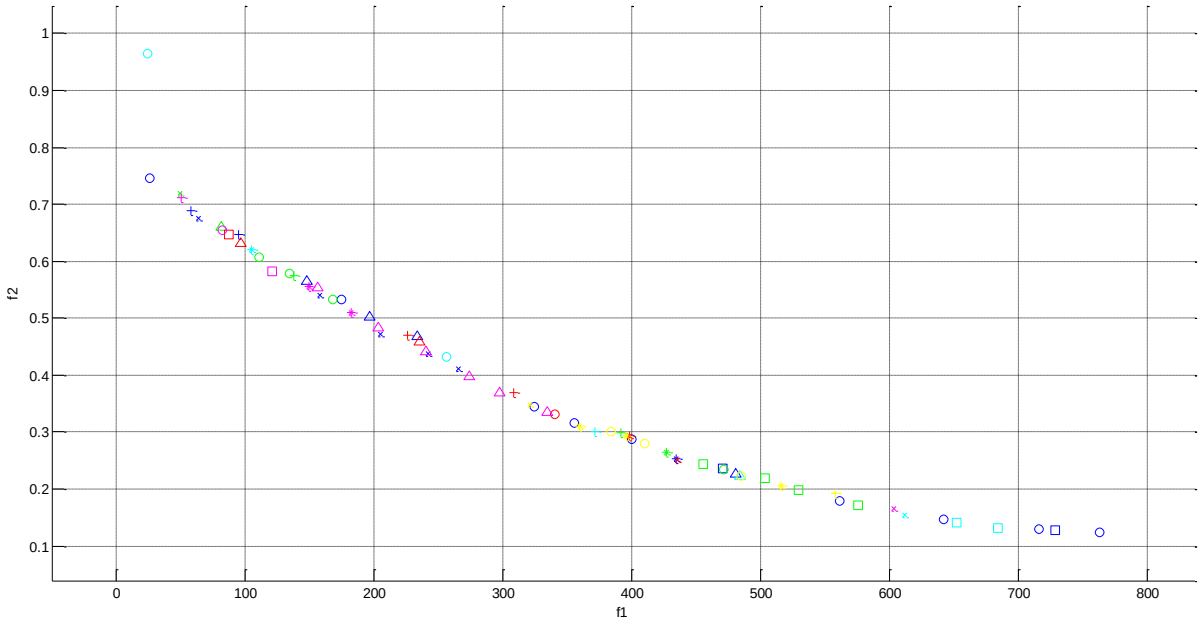

Figura 4.19. Aproximação da fronteira Pareto para o sistema da Cemig B do exemplo 3, topologia 1.

A Figura 4.20 apresenta o sistema após a restauração. Após a ocorrência das falhas, o sistema de proteção atua abrindo as chaves 12 – 9 isolando o alimentador 12 e 84 – 85 isolando o alimentador 12.

A Tabela 13 apresenta as chaves que foram abertas e fechadas em cada estágio.

Tabela 13 – Chaves abertas e fechadas por estágio para o exemplo 3 do sistema Cemig B, Topologia 1.

Estágio	Chave(s) Aberta(s)	Chave(s) Fechada(s)
1	85 – 105	6 – 105
2	85 – 92	6 – 92
3	9 – 16	7 – 17
4	85 – 87	6 – 87
5	85 – 101	6 – 101
6	85 – 107	6 – 107
7	85 – 99	6 – 99
8	85 – 88	6 – 88
9	85 – 95	6 – 95
10	9 – 13	7 – 13

11	85 – 96	6 – 96
12	85 – 86	6 – 86
13	9 – 44	7 – 44
14	9 – 38	7 – 38
15	85 – 106	6 – 106
16	85 – 91	6 – 91
17	85 – 90	6 – 90
18	85 – 100	6 – 100
19	9 – 14	7 – 14
20	85 – 116	6 – 116
21	85 – 89	6 – 89

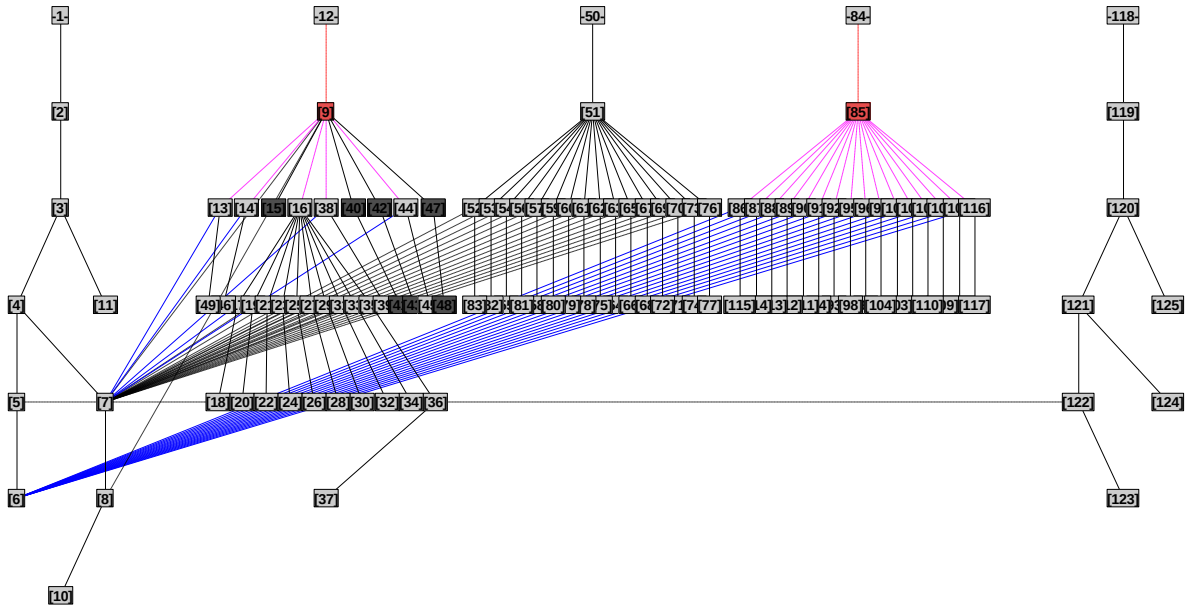

Figura 4.20. Exemplo de solução para o SD da Cemig B Exemplo 3 – Topologia 1

4.3.4. Exemplo 3 – Topologia 2 - Cemig B

A Figura 4.21 apresenta a fronteira Pareto encontrada pelo algoritmo para a topologia 2. Essa fronteira contém trinta e quatro indivíduos que se decodificam em vinte e uma soluções.

Para ilustração das manobras escolheu-se o indivíduo que se decodifica em 21 soluções. Na Tabela 14 estão indicadas a porcentagem de carga restaurada e o tempo gasto para cada uma das soluções.

Tabela 14 – Resultado para o exemplo 3 do sistema Cemig B, topologia 2.

SOLUÇÃO	CARGA RESTAURADA (%)	TEMPO (min)
1	6	32
2	31	63
3	38	101
4	43	134
5	50	181
6	53	205
7	56	236
8	59	273
9	62	319
10	64	363
11	66	399
12	70	446
13	74	461
14	75	493
15	76	519
16	80	554
17	82	602
18	85	647
19	86	694
20	87	734
21	88	768

Neste exemplo não é possível recuperar todas as cargas, isso ocorre devido à ausência de rota para esses blocos.

É interessante ressaltar novamente que para esse exemplo o decisor pode optar por não realizar todas as manobras tendo em vista que o tempo estimado para sua execução aumenta muito enquanto a quantidade de carga restaurada varia pouco. Dessa forma, a equipe de campo pode conseguir

chegar ao local da falta e resolver o problema mais rapidamente do que a realização de todas as manobras indicadas.

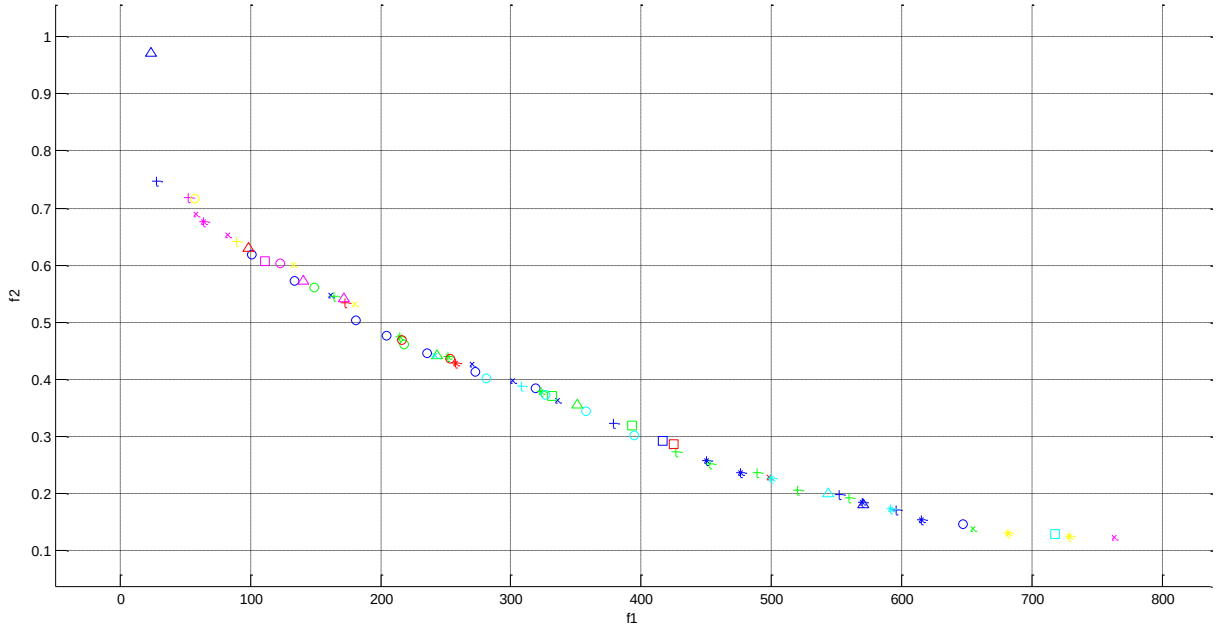

Figura 4.21. Aproximação da fronteira Pareto para o sistema da Cemig B do exemplo 3, topologia 2.

A Figura 4.22 apresenta o sistema após a restauração. Após a ocorrência das falhas, o sistema de proteção atua abrindo as chaves 12 – 9 isolando o alimentador 12 e 84 – 85 isolando o alimentador 12.

A Tabela 15 apresenta as chaves que foram abertas e fechadas em cada estágio.

Tabela 15 – Chaves abertas e fechadas por estágio para o exemplo 3 do sistema Cemig B, Topologia 2.

Estágio	Chave(s) Aberta(s)	Chave(s) Fechada(s)
1	85 – 105	6 – 105
2	9 – 16	7 – 33
3	85 – 87	6 – 87
4	85 – 107	6 – 107
5	85 – 101	6 – 101
6	9 – 38	7 – 38
7	9 – 44	7 – 44

8	85 – 96	6 – 96
9	85 – 100	6 – 100
10	85 – 99	6 –99
11	85 – 90	6 – 90
12	85 – 88	6 – 88
13	85 – 92	6 – 92
14	85 – 107	6 – 107
15	85 – 91	6 – 91
16	9 – 14	7 – 14
17	9 – 13	7 – 13
18	85 – 86	6 – 86
19	85 – 116	6 – 116
20	85 – 95	6 – 95
21	85 – 89	6 – 89

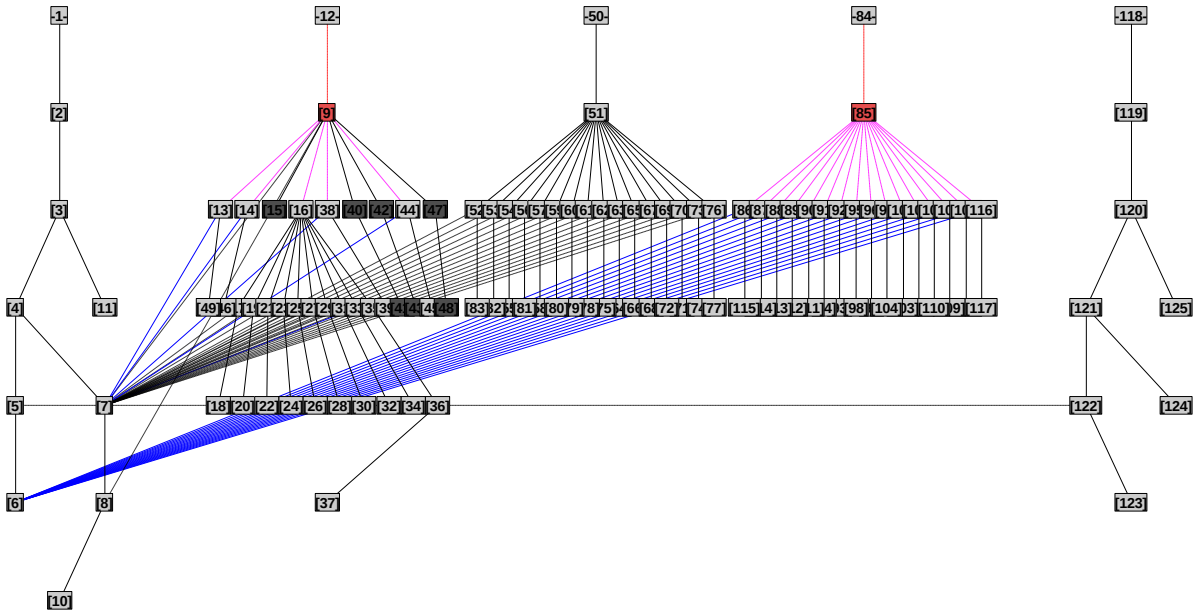

Figura 4.22. Exemplo de solução para o SD da Cemig B Exemplo 3 – Topologia 2.

4.4. Síntese do Capítulo

Este capítulo apresentou e discutiu os resultados de restauração obtidos no teste realizado com um sistema de 16 blocos e na aplicação do algoritmo implementado para dois sistemas da Cemig D, chamados sistema A

e sistema B. Para ambos os casos o teste foi realizado em três exemplos e variou-se, no terceiro caso de cada um, a topologia da rede.

Capítulo 5

Conclusão

Este capítulo sintetiza as principais conclusões obtidas e apresenta algumas considerações sobre perspectivas de pesquisas futuras relativas à área de estudo deste trabalho.

A cada dia o problema de reconfiguração de rede vem se tornando mais crucial para as concessionárias de energia elétrica. Isto ocorre devido à pressão sofrida por elas para atingirem as metas dos indicadores de continuidade estabelecidas pela ANEEL.

Foi discutido o problema de otimização não-linear de restauração de um sistema teste de 16 blocos no qual foram simuladas falhas a fim de comprovar a eficiência e funcionamento da ferramenta. Um algoritmo genético multiobjetivo SPEA2 foi implementado com o objetivo de realizar varreduras e gerar soluções eficientes e não repetidas. O algoritmo foi então aplicado a dois sistemas de grande porte disponibilizados pela Cemig distribuição, um de 703 barras e outro de 484.

O algoritmo gera como resultado indivíduos decodificados em várias soluções sequenciais. Como a proposta do algoritmo é não só minimizar a carga desconectada ao fim do conjunto de manobras, mas também minimizar a carga desconectada ao fim de cada iteração, o decisor terá confiança que a sequência de manobras apresentada é a que recuperará a maior quantidade de carga e, dessa forma, pode optar por não realizar todas.

Em ambos os sistemas, o algoritmo gerou soluções satisfatórias em tempo médio de execução baixo, cerca de 20 segundos.

Os resultados obtidos neste trabalho sugerem que, devido à radialidade dos sistemas, nem sempre é possível restaurar toda a carga desconectada proveniente das falhas. Isso geralmente ocorre por não haver rota até essas cargas ou pela violação a alguma restrição do sistema. Constata-se também, que nem sempre é interessante a realização de todas as

manobras, tento em vista que o tempo estimado para a realização das manobras é muito grande em relação a quantidade de carga restaurada.

Em trabalhos futuros o algoritmo implementado poderá ser utilizado com outras funções objetivo, como por exemplo, minimização das perdas do sistema e minimização do tempo de execução do algoritmo.

Referências Bibliográficas

Ali Chowdhury & Don Koval “Practical Methods and Applications” IEEE Series on Power Engineering, Power Distribution System Reliability, 2009.

Azuma, R. M. “Otimização multiobjetivo em problemas de estoque e roteamento gerenciados pelo fornecedor”. Dissertação de Mestrado, Campinas, SP, 2011.

Barbosa, C. H. N. R.; Ribeiro, R. S.; Vasconcelos, J. E. O.; Silva, M. G.; Silveira, T. A. G.; Cunha, T. B. ; Pereira, E. C.; Vasconcelos, J. A.. “Reconfiguração Autônoma de Sistemas Primários de Distribuição de Energia Elétrica”. In: Simpósio Brasileiro de Sistemas Elétricos, 2012, Goiânia. IV Simpósio Brasileiro de Sistemas Elétricos. Goiânia: UFG, 2012. v. 4. p. 1-6.

Bezdek J. C., “Pattern Recognition With Fuzzy Objective Function Algorithms”. New York: Plenum, 1981.

Bezdek J. C., Pal M. R., Keller J., Krisnapuram R., “Models and Algorithms for Pattern Recognition and Image Processing”. Norwell, MA: Kluwer, 1999, p. 792.

Bondy, J. A. and Murty, U. S. R. (1976). Graph Theory with Applications. MacMillan.

Chakraborty, M.; Chakraborty, U. K., “An Analysis of Linear Ranking and Binary Tournament Selection in Genetic Algorithms”. In International Conference, IEEE pp. 407-411 vol. 1. Sep. 1997.

Civanlar, S.; Grainger, J. J.; Yin, H. and Lee, S. S. H. (1988). Distribution feeder reconfiguration for loss reduction. IEEE Trans. Power Delivery, Vol. III, No. 3, pp. 1217- 1223.

Coelho, M., Oliveira, M.X., Wanner, E.F., Souza, M.J.F., Souza, S.R., Metaheurística GRASP Aplicada ao Problema de Corte com Dimensão

Aberta Guilhotinado, X Simpósio Brasileiro de Automação Inteligente, 2011, São João Del Rey.

Deb, K. “Multi-objective optimization using evolutionary algorithms.”, 2001.

Deb, K.; Pratap, A.; Agarwal, S.; Meyariva, T., “A Fast and Elitist Multiobjective Gentic Algorithm NSGA-II”. In IEEE Transactions on Evolutionary Computation, Vol. 6, no. 2, April 2002.

Delbém, A. C. B.; Ticona, W. G. C. “Algoritmos Evolutivos para otimização multi-objetivo”, São Carlos, 2008.

Edgeworth, F. Y. 1881. Mathematical physics. P. Keagan, London, England.

El-werfelli, M.; Dunn, R.; Iravani, P.; “Bachbone-network Reconfiguration for Power System Restoration using Genetic Algorithm and Expert System”, In Sustainable Power Generation and Supply, pages 1-6, April 2009.

Fonseca, C. M. and Fleming P. J. “Genetic Algorithms for Multi-Objective Optimization: Formulation, Discussion, and Generalization”. Proceedings of the Fifth International Conference on Gentic Algorithms, 416-423, 1993.Hou, Y., Liu, C. Zhang, P. Sun, K.; “Constructing Power System Restoration Strategies” Electrical and Electronics Engeneering, International Conference on Bursa, ELECO 2009, pp. I-8-I-13, Nov. 2009.

Ghosh, A. and Dehiri, S. “Evolucionary Algorithms for Multi-Criterion Optimization: A Survey”. International Journal of Computing e Information Sciences, vol 2, No. 1, April 2004.

Horn, J. and Nafploitis, N. and Goldberg, D.E. “A Niched Pareto Genetic Algorithm for Multi-Objective Optimization”. Proceedings of the fist IEEE Conference on Evolutionary Computation.82-87. 1994.

Mansour, M. R.; Santos, A.C.; London, J.B.; Delbem, A.C.B.; Bretas, N.G., "Energy restoration in distribution systems using multi-objective evolutionary algorithm and an efficient data structure," PowerTech, 2009 IEEE Bucharest , vol., no., pp.1,7, June 28 2009-July 2 2009.

Mantovani, R. S. José; Casari, Fernando; Romero, A., Rubén. “Reconfiguração de Sistemas de Distribuição Radiais Utilizando o Critério de Queda de Tensão.” Controle & Automação vol. 11 no. 03/Set, Out, Nov, Dezembro de 2000.

Mendoza, F.; Bernal-Agustin, J.L.; Dominguez-Navarro, J.A., "NSGA and SPEA Applied to Multiobjective Design of Power Distribution Systems," Power Systems, IEEE Transactions on , vol.21, no.4, pp.1938,1945, Nov. 2006.

Miu, K. N., Chiang, H. D., Yuan, B., Darling, G.; “Fast service restoration for large-scale distribution systems with priority customers and constraints” Power Systems, IEEE Transaction on, vol. 13, pp. 798-795, Aug 1988.

Moussa. Et al. 2010 “Representação nó-profundidade e algoritmos evolutivos aplicados ao problema de restabelecimento de energia em sistemas de distribuição de energia elétrica.” CBA 2010 XVIII Congresso Brasileiro de Automática – pp. 1215-1221, 2010.

Nouri- Zadeh, S.; Ranjbar, A. M.; “Multi Objective Power System Restoration” Electric Power Conference, 2008. EPEC 2008. IEEE Canada, pp. 1-5, Oct 2008.

Pareto, V. 1896. Cours d’economie politique, volume i and ii. F. Rouge, Lausanne.

Peponis, G.; Papadopoulos, M. Reconfiguration of radial distribution networks: application of heuristic methods on large-scale networks. IEE Proceedings on Generation, Transmission, and Distribution, v. 142, n. 6, p. 631-638, November 1995.

Ramaswamy, P.C.; Deconinck, G., "Smart grid reconfiguration using simple genetic algorithm and NSGA-II," Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on, vol., no., pp.1,8, 14-17 Oct. 2012.

Sanches, D.S. et al., 2011. “Integrating relevant aspects of moeas to solve reduction problem in large-scale” Distribution Systems. In PowerTech, Trondhein. pp. 1-6, 2011.

Santos, A. C., Delbem, A. e Bretas, N. G. Energy restoration for large-scale distribuition system using EA and new data structure. In Power and Energy Society General Meeting – Conversion and Delivery of Electrical Energy in the 21st Century, IEEE pp. 1-8, 2008.

Simone, L. F. C. “Reconfiguração de Redes para Restabelecimento de Energia em um Sistema de Distribuição Real.” Trabalho de conclusão de curso, USP- São Carlos, 2012.

Silva, R. R. “Estudo e Aplicação de um Algoritmo Genético Compacto Usando Elitismo e Mutação” Dissertação de mestrado, Curitiba, 2008.

Srinivas N.; Deb K., “Multi-objective optimization using non-dominated sorting in genetic algorithms,” Evol. Comput., vol. 2, no. 3, pp. 221–248, Fall 1994.

Suliman, S. I.; Kendall, G.; Musirin, I., “Optimizing Channel Allocation in Wireless Communication Using Single-Swap Mutation Based Heuristic”. In Advanced Communication Technology, in 15th International Conference, IEEE pp. 774-778 Jan. 2013.

Thakur, T., & Jaswanti, “Study and characterization of power distribution system network reconfiguration” IEEE PES Transmission and Distribution conference & Exposition, Latin America, 2006.

www.aneel.gov.br (ANEEL, 2013)

Yoshikawa M.; Otsuka, K.; Terai H., “Dedicated Hardware Architecture for Cycle Crossover Operation”. In Applications of Digital Information and Web Technologies, in First International Conference, IEEE pp. 338-341 Aug. 2008.

Zhu, J. Z. “Optimal Reconfiguration of Electrical Distribuition Network Using the Refined Genetic Algorithm” Electric Power Systems Research 62, pp. 37-42, 2002.

Zitzler E., Thiele L., “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach,” IEEE Trans. Evol. Comput., vol. 3, no. 4, pp. 257–271, Nov. 1999.

Zitzler E., Laumanns M., ThieleL.; “SPEA2: Improving the Strength Pareto Evolutionary Algorithm” Evolutionary Methods for Design, Optimisation and Control, CIMNE, Barcelona, Spain 2002.