



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Vitor dos Santos Batista

Análise de Desempenho de Meta-heurísticas Aplicadas ao Problema de Restauração de Redes de Distribuição

Belém

2020

Vitor dos Santos Batista

Análise de Desempenho de Meta-heurísticas Aplicadas ao Problema de Restauração de Redes de Distribuição

Dissertação de Mestrado apresentado à coordenação do Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) do Instituto de Tecnologia da Universidade Federal do Pará (UFPA) como parte dos requisitos finais para a obtenção do grau de Meste em Engenharia Elétrica.

Universidade Federal do Pará

Orientador: Ubiratan Holanda Bezerra

Belém

2020

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

B333a Batista, Vitor dos Santos
Análise de Desempenho de Meta-heurísticas Aplicadas ao
Problema de Restauração de Redes de Distribuição / Vitor dos
Santos Batista. — 2020.
74 f. : il. color.

Orientador(a): Prof. Dr. Ubiratan Holanda Bezerra
Dissertação (Mestrado) - Programa de Pós-Graduação em
Engenharia Elétrica, Instituto de Tecnologia, Universidade Federal
do Pará, Belém, 2020.

1. Restauração de Sistemas Elétricos. 2. Meta-heurísticas. 3.
Representação Nó-Profundidade. 4. Análise de Desempenho.
I. Título.

CDD 621.3191

**"ANÁLISE DE DESEMPENHO DE METAHEURÍSTICAS
APLICADAS AO PROBLEMA DE RESTAURAÇÃO DE REDES
DE DISTRIBUIÇÃO"**

AUTOR: VITOR DOS SANTOS BATISTA

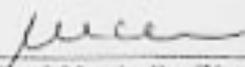
DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO
COLÉGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO
JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA
ELÉTRICA NA ÁREA DE SISTEMAS DE ENERGIA ELÉTRICA.

APROVADA EM: 20/02/2020

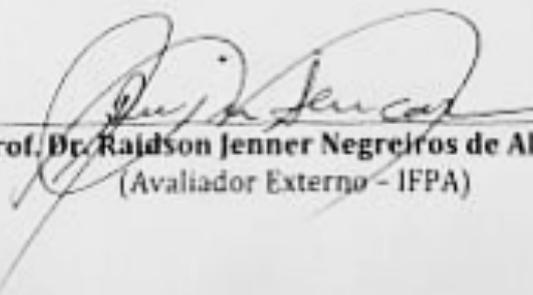
BANCA EXAMINADORA:



Prof. Dr. Ubiratan Holanda Bezerra
(Orientador - PPGEE/UFPA)



Prof.ª Dr.ª Maria Emilia de Lima Tostes
(Avaliadora Interna - PPGEE/UFPA)



Prof. Dr. Rajdson Jenner Negreiros de Alencar
(Avaliador Externo - IFPA)

VISTO:

Prof.ª Dr.ª Maria Emilia de Lima Tostes
(Coordenadora do PPGEE/TEC/UFPA)

Resumo

Nos últimos anos diversas meta-heurísticas foram utilizadas para solucionar o problema de restauração de Sistemas de Distribuição de Energia (SDE) de forma eficiente. Dentre elas destaca-se o AEMT - Algoritmo Evolutivo Multiobjetivo por Tabela, que juntamente com a utilização da RNP - Representação Nó-Profundidade, trouxeram um grande avanço na área pois a RNP sendo utilizada como estrutura de dados é possível realizar modificações na topologia do SDE sem perder a radialidade e restabelecer o fornecimento de energia para todas as cargas desligadas após a falta. Devido a falta de exploração de outras meta-heurísticas que utilizam a RNP como estrutura de dados, este trabalho visa realizar uma análise de desempenho comparando o AEMT com outras três meta-heurísticas, a Busca Tabu, Colônia de Abelhas e Estratégias evolutivas. A análise foi realizada em 3 SDEs com 84, 119 e 135 barras e foram simuladas 3 faltas em cada sistema.

Palavras-chave: Restauração de Sistemas Elétricos. Meta-heurísticas. Representação Nó-Profundidade. Análise de Desempenho.

Abstract

In the last years, several meta-heuristics have been used to solve the problem of restoring distribution systems efficiently. Among them, the Multiobjective Evolutionary Algorithm with Node-depth encoding (MEAN) stands out, which together with Node-Depth Encoding (NDE) makes a great advance in the field. All this because the NDE makes changes on the topology of the distribution system without losing radiality and reestablishing the power supply for all disconnected loads after the fault. Due to the lack of exploration of other meta-heuristics that use NDE as a data structure, this work aims to evaluate a performance analysis comparing MEAN with three other meta-heuristics, Tabu Search, Artificial Bee Colony and Evolutionary Strategies. The analysis was performed in three distribution systems the84-bus, 119-bus and 135-bus.s

Keywords: Electric System Restoration. Meta-heuristics. Node-Depth Encoding. Performance Analysis.

Lista de ilustrações

Figura 1.	Exemplo de um Sistema Elétrico de Potência	16
Figura 2.	Sistema de Distribuição com uma falta na barra 14 e setores a jusante desconectados	17
Figura 3.	Sistema de Distribuição restaurado	18
Figura 4.	Exemplo de um Grafo	22
Figura 5.	Exemplo de uma Floresta	23
Figura 6.	Representação de um SDE por meio de um grafo	24
Figura 7.	Produção de Soluções Infactíveis	25
Figura 8.	Codificação de um SDE num conjunto de RNPs	26
Figura 9.	T_{de} , T_{para} e suas respectivas RNPs	28
Figura 10.	T_{tmp} e sua RNP	28
Figura 11.	T'_{de} e sua RNP	29
Figura 12.	T'_{para} e sua RNP.	29
Figura 13.	T_{de} e suas respectivas RNPs	30
Figura 14.	Subárvores enraizadas nos nós do caminho de r até p	31
Figura 15.	T'_{tmp} e sua RNP	31
Figura 16.	T'_{para} e sua RNP	32
Figura 17.	Sistema IEEE de 33 barras	33
Figura 18.	Possibilidade de reconexão entre as subárvores isoladas do sistemas	34
Figura 19.	Etapa de varredura inversa (<i>backward</i>)	36
Figura 20.	Diferença entre recombinação e vizinhança	39
Figura 21.	Fluxograma da restauração utilizando Meta-heurísticas	40
Figura 22.	Dados estatísticos presentes no <i>boxplot</i>	47
Figura 23.	SDE de 84 barras	49
Figura 24.	SDE de 119 barras	50
Figura 25.	SDE de 135 barras	51
Figura 26.	Resultados da falta na barra 18	52
Figura 27.	Resultados da falta na barra 36	53
Figura 28.	Resultados da falta na barra 66	54
Figura 29.	Tempo de execução de cada algoritmo para o Sistema 84-barras	55
Figura 30.	Resultados da falta na barra 10	56
Figura 31.	Resultados da falta na barra 28	57
Figura 32.	Resultados da falta na barra 63	58
Figura 33.	Resultados da falta na barra 100	59
Figura 34.	Tempo de execução de cada algoritmo para o Sistema 119-barras	61
Figura 35.	Resultados da falta na barra 21	62

Figura 36.	Resultados da falta na barra 49	63
Figura 37.	Resultados da falta na barra 95	64
Figura 38.	Resultados da falta na barra 107	65
Figura 39.	Tempo de execução de cada algoritmo para o Sistema 135-barras . . .	67

Lista de tabelas

Tabela 1.	Lista de Adjacência do grafo da Figura 4.	23
Tabela 2.	Características dos sistemas de distribuição avaliados.	48
Tabela 3.	Teste de Dunn para a falta na barra 18	51
Tabela 4.	Resultado do AEMT falta na barra 18	52
Tabela 5.	Teste de Funn para a falta na barra 36	52
Tabela 6.	Resultado do ES falta na barra 36	53
Tabela 7.	Teste de Dunn para a falta na barra 66	54
Tabela 8.	Resultado do AEMT falta na barra 66	54
Tabela 9.	Teste de Dunn para a falta na barra 10	56
Tabela 10.	Resultado do ES falta na barra 10	56
Tabela 11.	Teste de Dunn para a falta na barra 28	57
Tabela 12.	Resultado do ES falta na barra 28	58
Tabela 13.	Teste de Dunn para a falta na barra 63	58
Tabela 14.	Resultado do AEMT falta na barra 63	59
Tabela 15.	Teste de Dunn para a falta na barra 100	60
Tabela 16.	Resultado do AEMT falta na barra 100	60
Tabela 17.	Teste de Dunn para a falta na barra 21	62
Tabela 18.	Resultado do ES falta na barra 21	62
Tabela 19.	Teste de Dunn para a falta na barra 49	63
Tabela 20.	Resultado do AEMT falta na barra 49	63
Tabela 21.	Teste de Dunn para a falta na barra 95	64
Tabela 22.	Resultado do ES falta na barra 95	64
Tabela 23.	Teste de Dunn para a falta na barra 107	66
Tabela 24.	Resultado do AEMT falta na barra 107	66
Tabela 25.	Médias e desvios-padrões dos testes executados	73
Tabela 26.	Testes estatísticos	73
Tabela 27.	Teste de Dunn para o SDE 84-barras	74
Tabela 28.	Teste de Dunn para o SDE 119-barras	74
Tabela 29.	Teste de Dunn para o SDE 135-barras	74

Lista de algoritmos

1	Cálculo do Número de Manobras	37
2	Busca Tabu	41
3	ABC	42
4	Estratégias Evolutivas $\mu + \lambda$	43
5	AEMT - Algoritmo Evolutivo Multiobjetivo por Tabelas	44

Sumário

1	INTRODUÇÃO	12
1.1	Motivação	12
1.2	Objetivos	13
1.2.1	Geral	13
1.2.2	Específicos	13
1.3	Revisão Bibliográfica	13
1.4	Estrutura do Trabalhos	15
2	RESTAURAÇÃO DE ENERGIA EM REDES DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA	16
2.1	Sistemas de Distribuição de Energia Elétrica (SDEE)	16
2.1.1	Restauração de Sistemas de Distribuição	17
2.2	Formulação Matemática	18
3	ESTRUTURA DE DADOS	21
3.1	Fundamentos da Teoria dos Grafos	21
3.2	Representação de um SDEE por meio de Grafos	24
3.3	Representação Nó-Profundidade	25
3.3.1	Operador PAO	27
3.3.2	Operador CAO	27
3.3.3	Determinação dos nós de poda, raiz, adjacente	32
3.4	Criação da Primeira Configuração Factível	33
3.5	Fluxo de Carga para SDE	35
3.6	Cálculo do Número de Manobras	36
4	ALGORITMOS DE OTIMIZAÇÃO PARA O PROBLEMA DE RESTAURAÇÃO	38
4.1	Meta-heurísticas	38
4.2	Busca Tabu	41
4.3	<i>Artificial Bee Colony</i> (ABC)	42
4.4	<i>Evolutionary Strategies</i> (ES)	43
4.5	AEMT - Algoritmo Evolutivo Multiobjetivo por Tabelas	44
5	RESULTADOS COMPUTACIONAIS	46
5.0.1	Análise Estatística	46
5.1	Sistemas Testes	47

5.2	Resultados	50
5.2.1	Falta Única no Sistema 84-barras	51
5.2.1.1	Falta na barra 18	51
5.2.1.2	Falta na barra 36	52
5.2.1.3	Falta na barra 66	53
5.2.1.4	Tempo Computacional para o Sistema 84-barras	54
5.2.2	Falta Única no Sistema 119-barras	55
5.2.2.1	Falta na barra 10	55
5.2.2.2	Falta na barra 28	57
5.2.2.3	Falta na barra 63	58
5.2.2.4	Falta na barra 100	59
5.2.2.5	Tempo Computacional para o Sistema 119-barras	60
5.2.3	Falta Única no Sistema 135-barras	60
5.2.3.1	Falta na barra 21	60
5.2.3.2	Falta na barra 49	62
5.2.3.3	Falta na barra 95	63
5.2.3.4	Falta na barra 107	64
5.2.3.5	Tempo Computacional para o Sistema 135-barras	66
5.3	Análise de Desempenho	67
6	CONCLUSÕES E TRABALHOS FUTUROS	69
	REFERÊNCIAS	70
	APÊNDICE A – RESULTADOS NUMÉRICOS DAS ANÁLISES ESTADÍSTICAS	73

1 Introdução

1.1 Motivação

Investimentos tem sido feito ao longo dos anos por concessionárias de energia elétrica para uma melhor distribuição de energia. Mesmo que o Sistema de Distribuição (SD) seja bem projetado, ele não está isento de falhas que possam interromper a distribuição de energia elétrica. Essas interrupções podem ser provenientes de falhas em equipamentos, erros humanos, acidentes ou desastres naturais.

Por isso é de grande importância que as concessionárias tenham estratégias para lidar com essas possíveis interrupções na rede elétrica. Para isso existem diversos equipamentos que lidam com essas falhas automaticamente (como os religadores), porém algumas falhas não podem ser supridas por eles, gerando faltas permanentes, mostrando que algo mais grave ocorreu no SD.

Nos casos de uma falta permanente, deve-se isolar o trecho onde ocorreu a falta e deslocar uma equipe da concessionária de energia para averiguar o ocorrido e resolver o problema. Esta resolução pode demorar de 4 a 14 horas, dependendo do que ocasionou a falta (ADIBI; MILANICZ, 1999). Devido as características radiais dos SD, após o isolamento da falta o trecho a jusante desta fica desenergizado, causando prejuízos tanto para as concessionárias de energia quanto para os seus clientes. Para religar temporariamente os clientes afetados pela falta, chaves seccionadoras Normalmente Abertas (NA) e Normalmente Fechadas (NF) são distribuídas pela rede elétrica com a finalidade de reconfigurar o SD para reestabelecer o fornecimento de energia para os clientes afetados, até que o problema seja resolvido.

Com o aumento dos grandes centros urbanos, os SD se tornaram cada vez maiores e mais complexos, aumentando o número de chaves NA e NF que podem ser manipuladas, elevando o grau de complexidade da escolha da nova topologia temporária. Diante disso diversas meta-heurísticas foram utilizadas ao longo dos anos a fim de encontrar a melhor topologia para a restauração temporária de energia após uma falta permanente.

Um dos maiores problemas relacionados a utilização das meta-heurísticas está em como representar computacionalmente o SD. Inicialmente essa representação era composta de um vetor contendo os estados das chaves NA e NF, contudo essa estratégia acabava gerando diversas soluções não-radiais, sendo necessário tratar essas soluções ineficazes, aumentando o custo computacional e o tempo de execução das meta-heurísticas.

Um dos maiores avanços nessa área foi a utilização da Representação Nó-Profundidade (RNP) como estrutura de dados para o SD (DELBEM et al., 2004). Com essa estrutura de

dados e seus operadores foi possível modificar da topologia dos SD sem perder a radialidade, gerando uma grande economia de tempo e recursos computacionais.

Uma das meta-heurísticas que foi extensamente utilizada juntamente a RNP foi AEMT - Algoritmo Evolutivo Multi-objetivo por Tabelas (SANTOS, 2009). Essa união trouxe excelentes resultados para a restauração de SD, pois o AEMT consegue resolver problemas multi-objetivos e multi-restritivos com uma boa eficiência.

Desde a união do AEMT com a RNP surgiram diversas modificações nessa combinação que possibilitaram resultados ainda melhores, contudo a exploração de outras meta-heurísticas utilizando a RNP como estrutura de dados foi pouco realizada.

1.2 Objetivos

1.2.1 Geral

O objetivo deste trabalho é realizar uma análise de desempenho com quatro meta-heurísticas para o problema de restauração de energia elétrica em sistemas de distribuição, utilizando a RNP como estrutura de dados, são elas:

- *Artificial Bee Colony* (ABC) (KARABOGA, 2005);
- Algoritmo Evolutivo Multiobjetivo por Tabelas (AEMT) (SANTOS, 2009);
- Busca Tabu (GLOVER, 1989);
- *Evolutionary Strategies* (ES) (BACK; HOFFMEISTER; SCHWEFEL, 1991).

1.2.2 Específicos

Este trabalho visa avaliar as meta-heurísticas quanto ao tempo de execução de cada uma delas e a função objetivo do problema de restauração.

Além disso este trabalho apresenta novos algoritmos de manipulação da RNP para que seja possível realizar modificações em Sistemas de Distribuição que apresentem apenas uma subestação fornecedora de energia.

1.3 Revisão Bibliográfica

Diversos trabalhos utilizam de meta-heurísticas para a restauração da rede elétrica.

Em (HSIAO; CHIEN, 2000) a restauração da rede elétrica ocorre através da combinação Fuzzy-GA (*Fuzzy-Genetic Algorithm*), cujo objetivos são minimizar a área fora de serviço, o número de operações, queda de tensão e o carregamento dos trechos e

dos transformadores, mantendo a radialidade. O sistema foi codificado utilizando as chaves NA e NF do sistema, se a radialidade for perdida quando uma chave for alterada pelo *Genetic Algorithm* (GA), o algoritmo tenta reparar os indivíduos que violaram a restrição.

Em (AUGUGLIARO; DUSONCHET; SANSEVERINO, 2000), o processo de restauração utiliza-se de ES juntamente com Conjuntos Fuzzy. Cada solução é codificada como um vetor binário contendo as informações das chaves NA, banco de capacitores e cargas. Não sendo mantida a radialidade do sistema.

Em (TOUNE et al., 2002) é realizado um estudo comparativo de 4 meta-heurísticas para o problema de restauração de SD, sendo elas o Algoritmo Genético, PSA (*Parallel Simulated Annealing*), Busca Tabu e Busca Tabu Reativa. Nele o SD é representado através de uma matriz contendo numa linha as cargas e na outra o nó que fornece energia para aquela carga, realizando alguns procedimentos para a manutenção da radialidade do sistema.

Em (SHIN et al., 2004) é utilizado o GA juntamente com a Busca Tabu na restauração. Neste caso a Busca Tabu foi implementada na etapa de Mutação do GA para acrescentar uma maior diversidade no processo de busca do GA. Um ponto interessante deste trabalho é a minimização das perdas financeiras provocadas pelas faltas como função objetivo, mantendo a radialidade do sistema.

Em (INAGAKI; NAKAJIMA; HASEYAMA, 2006) é utilizado o GA com o *Simulated Annealings* (SA) para o problema de restauração. O sistema é representado pelo estado das chaves NA e NF e o algoritmo tenta minimizar o número de chaveamentos e os setores fora de serviço. Neste trabalho é gerado uma fronteira de Pareto com as soluções finais além de se evitar a geração de soluções não radiais.

Em (SANTOS; DELBEM; BRETAS, 2008) é apresentado um algoritmo evolutivo (posteriormente denominado de AEMT - Algoritmo Evolutivo Multi-objetivo por Tabelas) que utiliza a RNP - Representação Nó-Profundidade, proposta por (DELBEM et al., 2004), como estrutura de dados e de dois operadores para a modificação da rede. As principais vantagens da utilização da RNP e de seus operadores é a garantia da restauração do setor a jusante da falta e de sempre gerar configurações radiais.

Em (MANSOUR, 2009) foi utilizada a RNP juntamente com o *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) (DEB et al., 2002), surgindo o NS2R, onde as ideias de (DELBEM et al., 2004) e do famoso NSGA-II foram unidas para resolver o problema de restauração com redução de perdas.

Em (SANCHES et al., 2013) é apresentado Algoritmo Evolutivo Multi-Objetivo em Tabelas com Soluções Não-Dominadas e *Strength Pareto* (AEMT-SP), que utiliza as características do AEMT, NSGA-II e SPEA2 - *Strength Pareto Evolutionary Algorithm 2* (ZITZLER; LAUMANN; THIELE, 2001) e da RNP como estrutura de dados, além da

Evolução Diferencial Multi-Objetivo em Tabelas (EDMT) a qual se baseia no algoritmo da Evolução Diferencial (STORN; PRICE, 1997) e num operador de Recombinação baseado no histórico evolutivo das Soluções.

Em (MATHIAS-NETO, 2016) é utilizada a Busca Tabu juntamente com a RNP no processo de restauração, incluindo um novo operador para a RNP, denominado CUT, que permite o corte seletivo de carga no para a utilização de geradores distribuídos no processo de restauração.

Em (MARQUES, 2018) é utilizada o AEMT juntamente com uma busca local exaustiva, no processo de restauração, levando em consideração os aspectos práticos dos Sistemas de Distribuição.

1.4 Estrutura do Trabalhos

Os capítulos seguintes estão organizados da seguinte maneira:

- Capítulo 2: É apresentado o problema de restauração das redes de distribuição de energia elétrica;
- Capítulo 3: É apresentada o funcionamento da RNP - Representação Nó-Profundidade e as modificações realizadas para o seu funcionamento em redes com apenas uma subestação;
- Capítulo 4: São apresentados os algoritmos de otimização que serão avaliados;
- Capítulo 5: É apresentado os resultados computacionais e a análise de desempenho dos algoritmos;
- Capítulo 6: São apresentadas as conclusões deste trabalho e a possibilidade de trabalhos futuros.

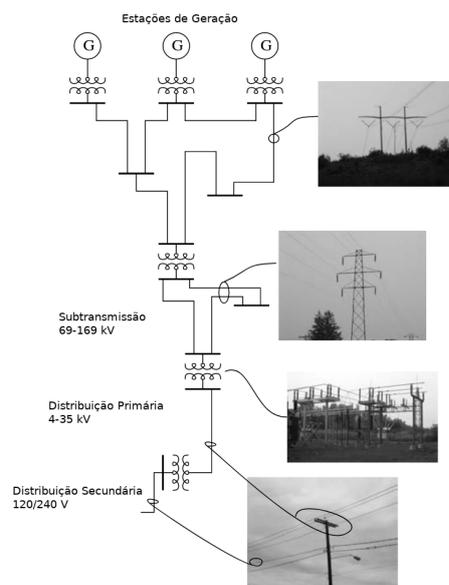
2 Restauração de Energia em Redes de Distribuição de Energia Elétrica

Neste capítulo será apresentado uma pequena introdução aos sistemas de distribuição e ao problema de restauração. Também será apresentada a formulação matemática do problema que será utilizada na avaliação dos algoritmos testados.

2.1 Sistemas de Distribuição de Energia Elétrica (SDEE)

O Sistema Elétrico de Potência pode ser definido como uma rede conectada de equipamentos, capaz de gerar energia elétrica e levar essa energia até os consumidores com qualidade e segurança. Esse sistema é dividido em três segmentos: a Geração, a Transmissão e a Distribuição. A Geração é o segmento responsável por converter algum tipo de energia presente na natureza (proveniente de rios, dos ventos, vinda do sol, de reações químicas, entre outras) para energia elétrica. A Transmissão é responsável por transportar essa energia do local da geração para o local onde ela será consumida, que geralmente fica distante da geração. A Distribuição fica responsável por distribuir a energia elétrica vinda das linhas de transmissão para o consumidor. Essa relação entre a Geração, Transmissão e Distribuição de energia pode ser observada na Figura 1.

Figura 1. Exemplo de um Sistema Elétrico de Potência



Fonte: (SHORT, 2018)

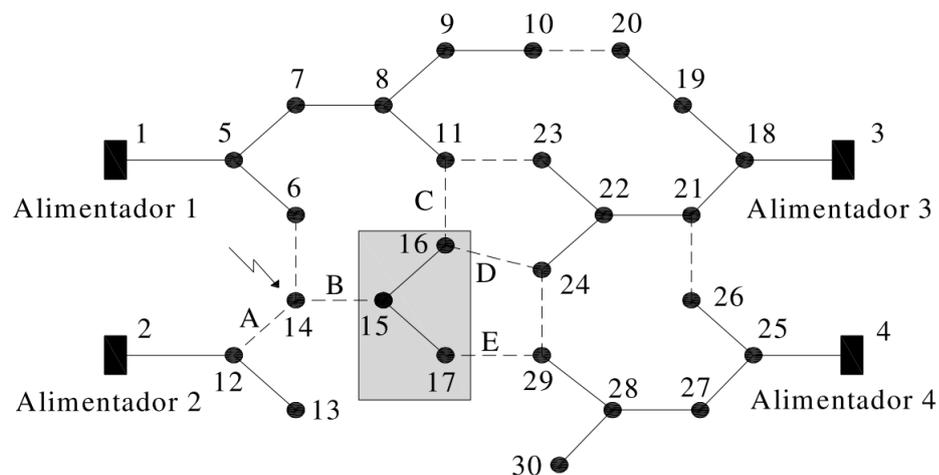
A Distribuição de energia é subdividida em dois setores, a Distribuição Primária e a Distribuição Secundária. O sistema de Distribuição Primário opera em médias tensões e em redes radiais aéreas. Ela atende transformadores de distribuição que são utilizados para baixar a tensão para alimentar o Sistema de Distribuição Secundário e aos consumidores primários que dependem de uma alta demanda de energia, como indústrias, hospitais, *shopping centers*, entre outros (MANSOUR, 2009). É nela em que se encontram recursos instalados para a reconfiguração do Sistema de Distribuição após uma falta permanente. A Distribuição Secundária pode operar em redes em malha ou radiais, ela atende consumidores de baixa tensão, pequenas indústrias e comércios, além de atender os consumidores domésticos (MANSOUR, 2009).

No Sistema de Distribuição existem chaves seccionadoras espalhadas pela rede e através da abertura e do fechamento dessas chaves é possível alterar a topologia da rede elétrica. Elas podem se apresentar em dois estados, Normalmente Abertas (NA) e Normalmente Fechadas (NF).

2.1.1 Restauração de Sistemas de Distribuição

A restauração de sistemas elétricos acontece quando uma falta de caráter permanente ocorre na rede elétrica. Neste contexto primeiramente é necessário encontrar o local onde esta falta ocorreu. Após a localização a falta deve ser isolada, abrindo todas as chaves seccionadoras NFs conectadas a ela (SANTOS, 2009). A Figura 2 mostra este processo, onde a barra 14 foi isolada do sistema, abrindo a chave *A* e *B*, o que ocasionou um desligamento de todas as barras a jusante a falta (destacado em cinza).

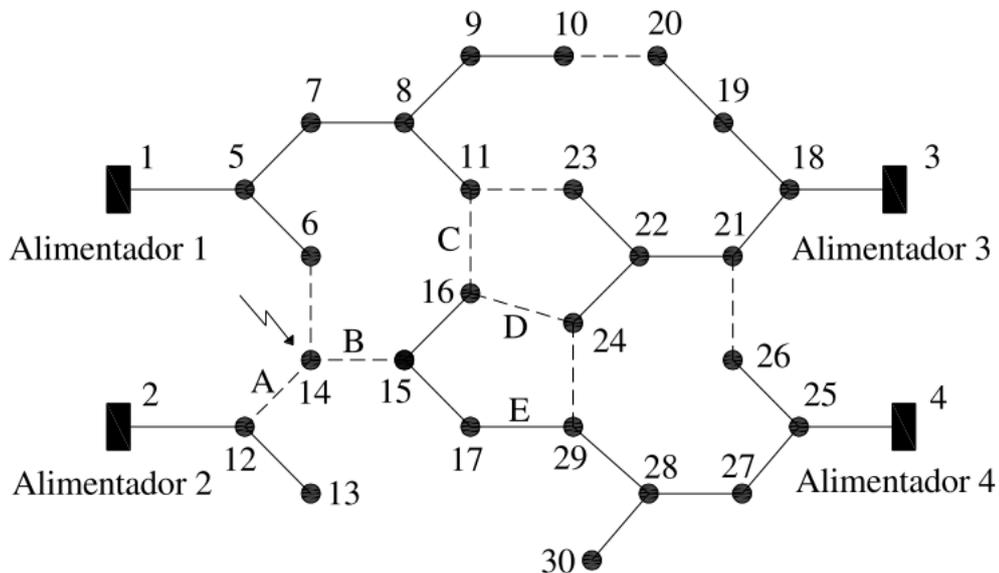
Figura 2. Sistema de Distribuição com uma falta na barra 14 e setores a jusante desconectados



Fonte: (SANTOS, 2009)

A restauração de energia para o trecho desligado na Figura 2 pode ser realizado através do fechamento das chaves NAs conectadas nele, sendo elas as chaves *C*, *D* e *E*. Um exemplo de uma possível topologia temporária é apresentada na Figura 3, onde o setor foi religado temporariamente através da chave *E*. Quando o problema na barra 14 for solucionado o sistema volta a sua topologia original.

Figura 3. Sistema de Distribuição restaurado



Fonte: (SANTOS, 2009)

2.2 Formulação Matemática

A escolha da nova topologia após o isolamento da falta deve ser realizada tentando maximizar o número de consumidores desligados e minimizar número de chaveamentos necessários na restauração. Esta nova topologia deve ser capaz de cumprir com as restrições de operação existentes nos SD, sendo elas as restrições de carregamento de linhas, potência demandada da subestação, níveis de tensão estabelecidos e radialidade do sistema. Com isso o problema de restauração de SDE pode ser compreendido como um problema multi-objetivo e multi-restritivo (MARQUES, 2013).

Com a utilização da RNP e de seus operadores, desenvolvidos por (DELBEM et al., 2004), é garantido o reestabelecimento de todos os consumidores desligados após a falta. Cada alteração da topologia mantém os consumidores conectados sem perder a radialidade do sistema, com isso um dos objetivos e uma das restrições do problema de restauração foram cumpridos apenas por utilizar uma estrutura de dados adequada, diminuindo a complexidade do problema, portanto a restauração de SDE com a utilização da RNP pode

ser modelado matematicamente utilizando a Função de Agregação, que contém o objetivo do problema (redução do número de manobras) agregada as penalidades das violações das restrições, sendo apresentada na Equação 2.1.

$$\begin{aligned}
 & \text{Min } f(T) + X(T)P_1 + B(T)P_2 + V(T)P_3 \\
 & P_1 = \begin{cases} 100 & \text{se } X(T) > 1 \\ 0 & \text{se } X(T) \leq 1 \end{cases} \\
 & P_2 = \begin{cases} 100 & \text{se } B(T) > 1 \\ 0 & \text{se } B(T) \leq 1 \end{cases} \\
 & P_3 = \begin{cases} 1000 & \text{se } V(T) > 1.05 \\ 1000 & \text{se } V(T) < 0.95 \\ 0 & \text{se } 0.95 \leq V(T) \leq 1.05 \end{cases}
 \end{aligned} \tag{2.1}$$

Onde T é a rede elétrica representada pela RNP, f é o número de manobras e o único objetivo do problema que deverá ser minimizado (o cálculo do número de manobras é visto na Seção 3.6), X , B e V são as restrições do problema, sendo que X é o maior carregamento da rede, B é o maior carregamento da subestação, V é a maior queda de tensão na rede e P_1 , P_2 e P_3 são as penalidades das restrições.

As restrições X e V são obtidas através dos níveis de corrente nos trechos e dos níveis de tensão nas barras, respectivamente. Esses dados provém do cálculo de Fluxo de Carga, apresentado na Seção 3.5, que é calculado para cada nova topologia produzida pelas meta-heurísticas.

O carregamento da rede é calculada utilizando a Fórmula 2.2, sendo $j = \{1, 2, \dots, m\}$ com m sendo o número total de trechos na rede, x_j é o valor de corrente de um determinado trecho j , calculada utilizando o Fluxo de Carga, e \bar{x}_j o valor de corrente máxima admitido por este trecho.

$$X(T) = \max \left(\frac{x_j}{\bar{x}_j} \right) \tag{2.2}$$

Uma determinada rede violará essa restrição se ao menos um trecho estiver sobrecarregado. Esta restrição não verifica quantos trechos ficaram sobrecarregados, pois se o trecho mais carregado estiver com o valor abaixo de um (1), isto garantirá que todos os trechos do sistema não ficaram sobrecarregados e a restrição é zerada, conforma a Equação 2.1.

Os níveis de tensão são restritos por um limite superior e um limite inferior: $V_k^{min} < V_k < V_k^{max}$, sendo $k = \{1, 2, \dots, n\}$ com n sendo o número total de barras na rede.

Por se tratar de um problema de minimização, acaba ocorrendo uma incompatibilidade entre a restrição de tensão e as outras restrições do problema, tendo em vista que os valores da tensão (em p.u.) tendem a ir para o valor um (1) até alcançar os limites de tensão, enquanto isso as outras restrições tendem a ir para zero (0). Por isso foi utilizado o valor da queda de tensão para o cálculo da restrição V , conforme a Equação 2.3, sendo v_k o nível de tensão na barra em p.u. Este valor será utilizado somente se os níveis de tensão na barra estiverem fora dos limites estabelecidos, caso contrário o valor de $V(T)$ é zerado, conforme a Equação 2.1.

$$V(T) = \max(1 - v_k) \quad (2.3)$$

Os limites V_k^{min} e V_k^{max} inicialmente foram definidos como sendo, respectivamente, 0,95 p.u. e 1,05 p.u. Contudo, em alguns testes, os algoritmos realizavam chaveamentos em regiões que não tinham conexão com a falta, devido alguns valores de tensão da configuração pré-falta serem abaixo de 0,95 p.u. Por isso, os limites são definidos como sendo os valores mínimos e máximos de tensão nas barras do sistema pré-falta.

O carregamento das subestações é calculado utilizando a razão entre o somatório de cargas conectadas numa determinada subestação e a potência máxima desta, caso haja mais de uma subestação num sistema, o valor atribuído a B corresponde a subestação mais carregada, conforme a Equação 2.4, sendo s o número de subestações, k o número de barras conectadas numa subestação s e \bar{b}_s o carregamento máximo da subestação s .

$$B(T) = \max \left(\frac{\sum_{i=1}^k b_s^i}{\bar{b}_s} \right) \quad (2.4)$$

Da mesma forma que a restrição do carregamento da rede, se a subestação com o maior carregamento não ficar sobrecarregada, todas as subestações da rede também não ficarão.

Os valores escolhidos para as penalidades foram escolhidos com base em (SANCHES, 2013), sendo $P_1 = 100$ e $P_2 = 100$. A única exceção está em $P_3 = 1000$ pois os valores da queda de tensão são muito pequenos em comparação as outras restrições, por isso o valor de P_3 foi aumentado.

3 Estrutura de dados

Neste Capítulo será apresentado alguns conceitos básicos sobre a Teoria dos Grafos necessários para o entendimento do funcionamento da RNP e de seus operadores. Também serão apresentadas as extensões realizadas nesses operadores, um novo algoritmo para encontrar os nós de poda, raiz e adjacente, utilizados por eles, e um novo algoritmo para a obtenção da Primeira Configuração Factível.

3.1 Fundamentos da Teoria dos Grafos

Um grafo pode ser definido como uma abstração de um objeto de estudo existente no mundo real, para simplificar e facilitar o entendimento e a resolução de problemas envolvendo este objeto. Os grafos podem representar desde redes de transportes, relacionamento de pessoas em redes sociais, redes de fibra óptica e sistemas de distribuição de energia elétrica.

Matematicamente, um grafo G é representado por $G = (V, E)$, onde $V(G)$ é um conjunto finito dos elementos de G , denominado de vértices ou nós e $E(G)$ é um conjunto finito de pares (ordenados ou não) do conjunto $V(G)$ denominado de elos ou arestas. Estas arestas formam um relacionamento entre os itens que o compõe, tendo a possibilidade de um nó ser conectado a nenhum, um ou mais de um nó, formando assim um grafo conectado.

Na Figura 4 é apresentado um exemplo de grafo $G = (V, E)$, sendo o conjunto de nós $V = \{A, B, C, D, E, F, G, H\}$ e o conjunto de arestas $E = \{(A, B), (B, C), (B, D), (C, D), (D, E), (D, F), (D, G), (E, F), (G, H)\}$. Esta figura servirá de exemplo na explicação de algumas propriedades e conceitos importantes sobre grafos, importantes para o entendimento da RNP e de seus operadores, que serão apresentados a seguir.

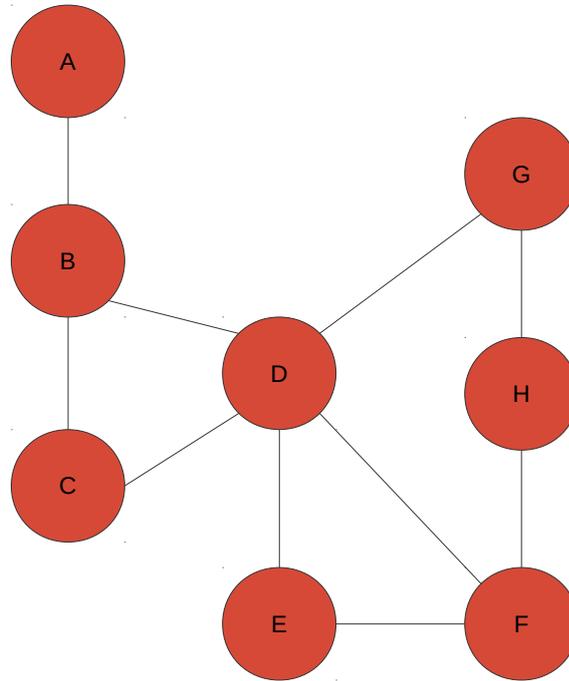
A ordem (n) de um grafo é o número de nós presentes nele, sendo $n = |V|$ e o tamanho (m) de um grafo é o número de arestas contidas no gráfico, sendo $m = |E|$ (SOARES, 2009). O grafo da Figura 4 possui $n = 8$ e $m = 10$.

Os grafos podem ser classificados de acordo com a orientação de suas arestas, se elas existirem o grafo é classificado como grafo orientado ou dígrafo ¹, ou seja $(A, B) \neq (B, A)$. Caso a orientação das arestas não tenha importância para o problema, o grafo é classificado como grafo simples, ou simplesmente grafos, sendo $(A, B) = (B, A)$ (SOARES, 2009).

O relacionamento de vizinhança ou adjacência ocorre quando dois nós são conectados por uma aresta. Na Figura 4 a aresta ab é responsável pela conexão entre os nós A e B ,

¹ neologismo do Inglês *Digraph - Directed Graph*

Figura 4. Exemplo de um Grafo



Fonte: Produzido pelo autor

logo A é adjacente de B e vice-versa (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011).

O caminho entre dois nós num grafo é dado pela sequência de nós $\{v_1, v_2, v_3, \dots, v_n\}$ sendo que dois nós consecutivos são adjacentes (SOARES, 2009). Como exemplo um caminho partindo do nó A para o nó G , presentes na Figura 4, é composto pelos nós $\{A, B, D, G\}$.

Um ciclo num grafo ocorre quando o ponto de partida de um caminho é igual ao final deste, caso um grafo não possua ciclos ele é denominado de grafo acíclico. O caminho $\{D, E, F, H, G, D\}$ da Figura 4 é um ciclo. Um grafo conexo é aquele que para qualquer conjunto de nós (v, w) existe um caminho com as extremidades v e w (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011).

Uma das formas de se representar um grafo é através de uma Lista de Adjacência (LA), nela estão presentes todas as relações de adjacências existentes num grafo. A LA do grafo da Figura 4 é apresentada na Tabela 1.

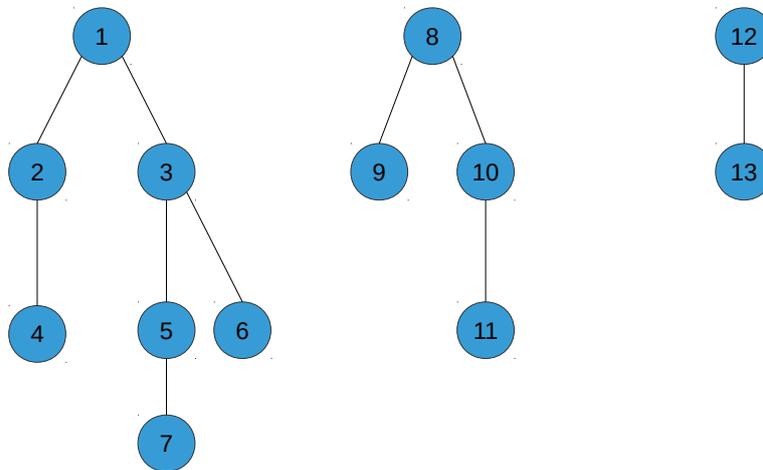
Uma Árvore é um grafo acíclico, não-orientado e conexo. Caso um grafo seja acíclico e não-conexo ele é denominado de Floresta, sendo assim, uma Floresta é composta de

Tabela 1. Lista de Adjacência do grafo da Figura 4.

Nó	Adjacentes				
A	B				
B	A	C	D		
C	B	D			
D	B	C	E	F	G
E	D	F			
F	D	E	H		
G	D	H			
H	F	G			

Árvores. Uma árvore é dita enraizada quando um dos nós desta árvore é denominado de raiz (SOARES, 2009). A profundidade dos nós corresponde ao tamanho do caminho deste nó até o nó raiz, ou seja, é a distância entre esses dois nós. Na Figura 5 é apresentado um exemplo de uma Floresta contendo 3 Árvores, sendo as raízes delas os nós 1, 8 e 12.

Figura 5. Exemplo de uma Floresta

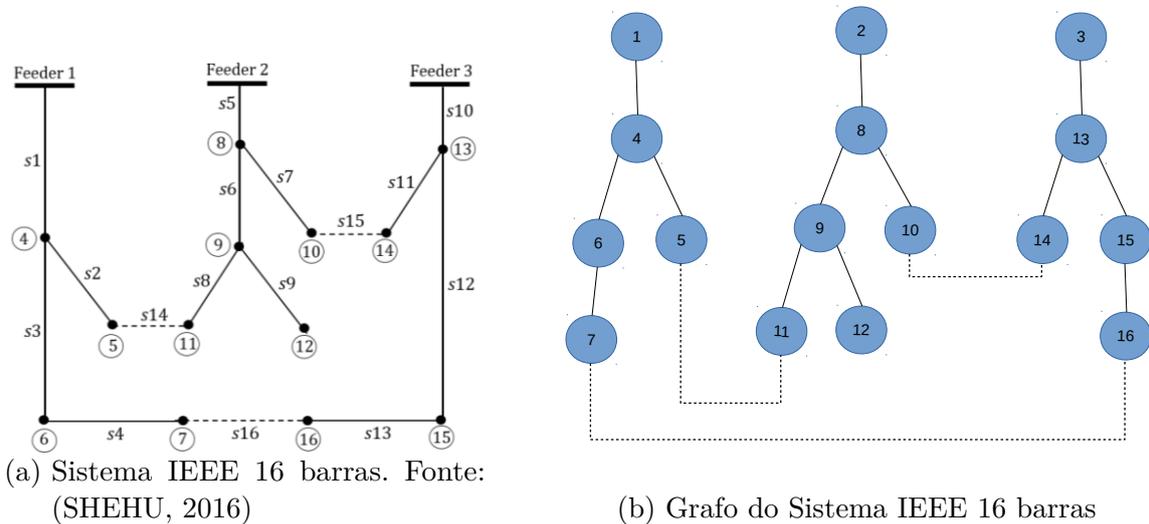


Fonte: Produzido pelo autor

3.2 Representação de um SDEE por meio de Grafos

Nos problemas de restauração os SDE são apresentados como um conjunto de subestações e cargas conectadas através de chaves seccionadores NA e NF. Este SDE pode ser representado por meio de uma floresta, sendo que cada árvore desta floresta representa uma rede conectada a uma subestação, cada barra da subestação é a raiz de uma árvore, cada nó do grafo representa uma barra do sistema elétrico e cada chave, seja ela NA ou NF, é representada por uma aresta tracejada e contínua, respectivamente. A Figura 6 mostra um exemplo típico dessa representação, onde o sistema IEEE 16-barras, com 3 subestações é representado numa floresta com três árvores.

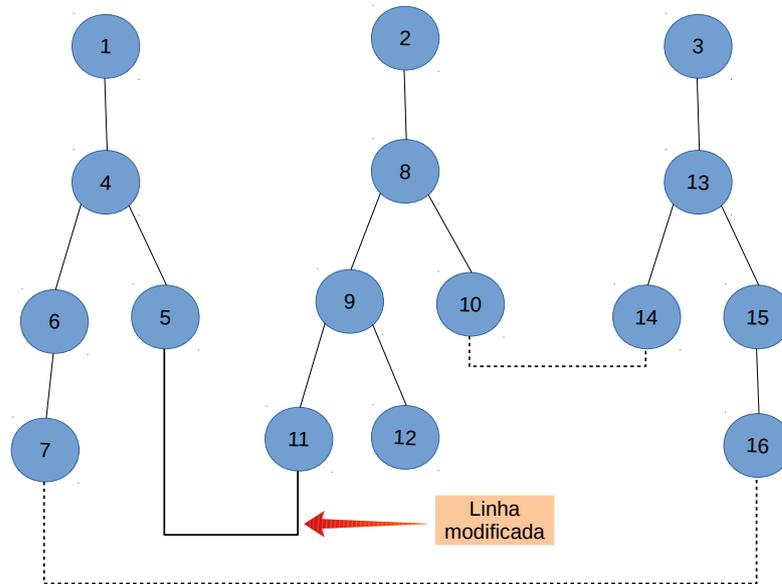
Figura 6. Representação de um SDEE por meio de um grafo



Fonte: Produzido pelo autor

Existem diversas formas de se representar computacionalmente um SDE, a mais comum é a utilização de um vetor binário contendo os estados das chaves NA e NF (HSIAO; CHIEN, 2000), (AUGUGLIARO; DUSONCHET; SANSEVERINO, 2000), sendo 1 se a chave estiver fechada e 0 se ela estiver aberta. A modificação deste vetor provocava uma alteração na topologia da rede. No decorrer do algoritmo essas modificações podem gerar soluções não-radiais. A Figura 7 exemplifica esta ideia, mostrando que a alteração da chave (5, 11) acabou ocasionando a geração de um ciclo no sistema. Mesmo que para engenheiros esta modificação seja algo impensável, a codificação por meio dos estados das chaves NA e NF permite esta ação.

Figura 7. Produção de Soluções Infactíveis



(a) Sistema IEEE-16 barras apresentando um ciclo

De	Para	Conf. Original	Conf. Modificada
1	4	1	1
4	5	1	1
4	6	1	1
6	7	1	1
2	8	1	1
8	10	1	1
8	9	1	1
9	11	1	1
9	12	1	1
3	13	1	1
13	14	1	1
13	15	1	1
15	16	1	1
5	11	0	1
7	16	0	0
10	14	0	0

(b) Chaves do Sistema IEEE 16-barras

Fonte: Produzido pelo autor

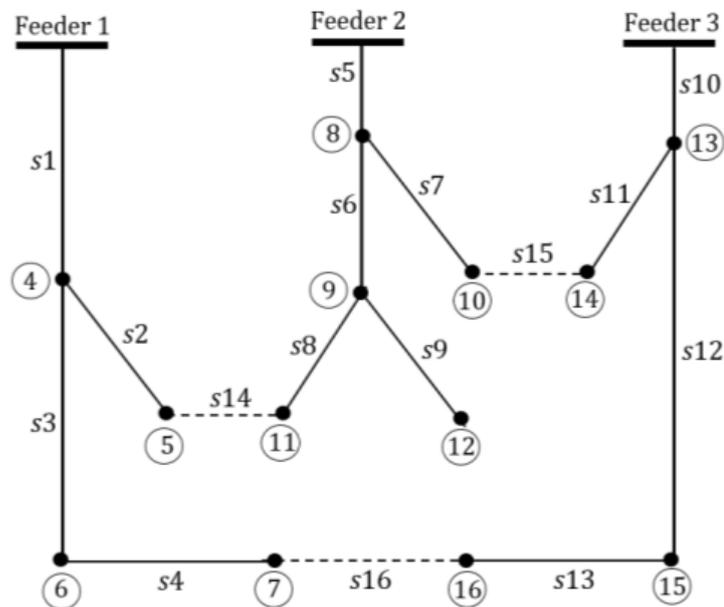
3.3 Representação Nó-Profundidade

A RNP - Representação Nó-Profundidade é uma estrutura de dados utilizada para representar árvores, proposta por (DELBEM et al., 2004). A maior vantagem da utilização

da RNP é de realizar modificações na topologia do SDE sem que a radialidade deles seja perdida, ou seja, produzir novas florestar a partir de uma existente sem que essas florestas geradas apresentem ciclos nelas.

A RNP é composta por uma matriz $2 \times N$ onde N é o número de nós da árvore. Para uma floresta, composta de várias árvores, existe uma RNP para cada árvore, como mostra a Figura 8. A primeira linha dessa matriz corresponde ao nós da árvore e a segunda relacionada a profundidade desses nós. Esses nós estão ordenados de acordo com a busca em profundidade realizada na árvore, partindo do nó da subestação, como mostra a Figura 8.

Figura 8. Codificação de um SDE num conjunto de RNPs



(a) Sistema IEEE 16 barras. Fonte: (SHEHU, 2016)

Alimentador 1					
Nó	1	4	6	7	5
Prof.	1	2	3	4	3

Alimentador 2						
Nó	2	8	9	11	12	10
Prof.	1	2	3	4	4	3

Alimentador 3					
Nó	3	13	14	15	16
Prof.	1	2	3	3	4

(b) Representação Nó-Profundidade do Sistema IEEE 16-barras

Fonte: Produzido pelo autor

As modificações realizadas na RNP são realizadas através de dois operadores principais, o PAO (*Preserve Ancestor Operator*) e o CAO (*Change Ancestor Operator*). Para o operador PAO é necessário que já se tenham escolhido previamente os nós de poda p e adjacente a e para o operador CAO, além dos nós p e a , é necessário escolher um novo nó raiz r . O algoritmo para encontrar esses nós na floresta é visto na Seção 3.3.3.

3.3.1 Operador PAO

O operador PAO consiste em gerar uma floresta F' a partir de uma floresta F . Este processo corresponde em transferir uma subárvore de uma árvore T_{de} para outra árvore T_{para} , mantendo a estrutura da subárvore que será transferida, ou seja, o nó raiz p da subárvore é o mesmo nó que será enxertado em outra árvore da floresta num nó adjacente a .

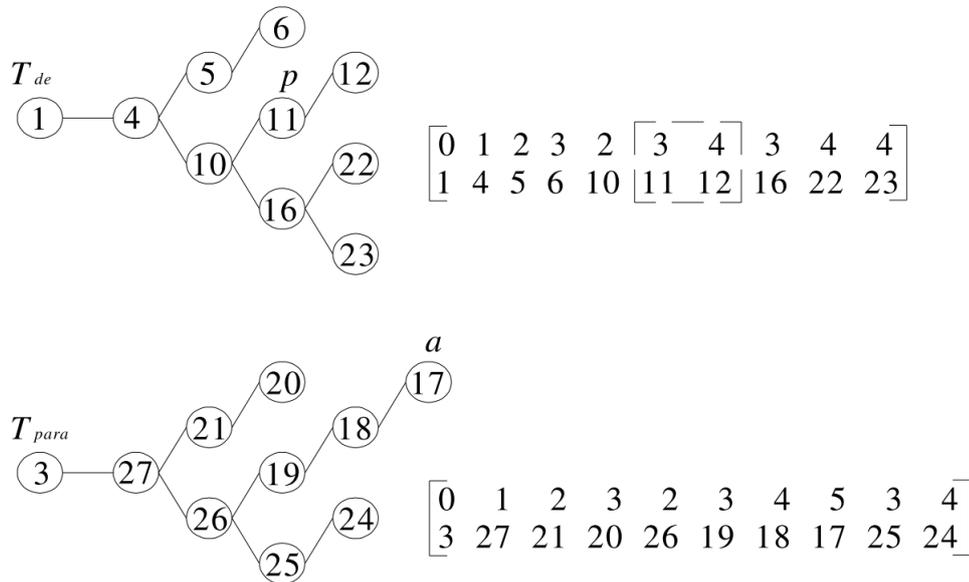
Foram utilizados os trabalhos de (SANTOS, 2009) e (SANCHES, 2013) na descrição dos passos dos passos a seguir. Considerando que p e a foram previamente determinados, i_p e i_a como sendo os índices de p e a nas árvores T_{de} e T_{para} , respectivamente e considerando $p = 11$ e $a = 17$, o operador PAO pode ser descrito através dos seguintes passos:

1. Determinar as posições $\{i_p - i_u\}$ dos índices em T_{de} correspondente a subárvore que contém o nó p como raiz. Sendo i_p conhecido, é necessário encontrar apenas i_u que corresponde ao índice do último nó da subárvore que contém o nó p como raiz. O conjunto $\{i_p - i_u\}$ corresponde ao nó p em i_p e aos consecutivos nós x no vetor T_{de} de forma que $i_x > i_p$ e $p_x > p_p$ (retângulo tracejado na Figura 9), onde p_x é a profundidade do nó x e p_p é a profundidade do nó p ;
2. Copiar os dados do conjunto $\{i_p - i_u\}$ de T_{de} para um vetor temporário T_{tmp} (subárvore que está sendo transferida), ver Figura 10.
3. Atualizar a profundidade de cada nó x do conjunto $\{i_p - i_u\}$ utilizando a Equação 3.1:

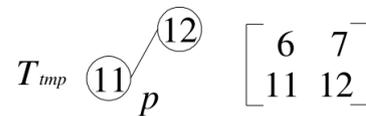
$$p_x = p_x - p_p + p_a + 1 \quad (3.1)$$
4. Armazenar num vetor T'_{para} os dados de T_{para} e T_{tmp} , inserindo T_{tmp} a partir da posição ia , ver Figura 11;
5. Armazenar em T'_{de} os nós de T_{de} sem os nós de T_{tmp} , ver Figura 12;

3.3.2 Operador CAO

O CAO segue os mesmos princípios do PAO, contudo a estrutura da subárvore que será transferida é alterada, o nó raiz da subárvore não é mais o nó p , como no operador

Figura 9. T_{de} , T_{para} e suas respectivas RNPs

Fonte: (SANTOS, 2009)

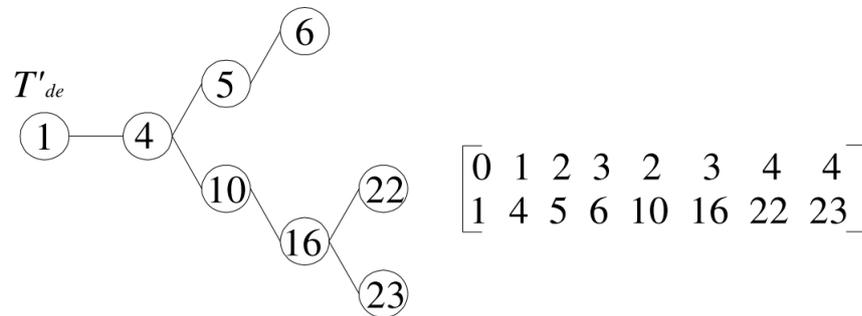
Figura 10. T_{tmp} e sua RNP

Fonte: (SANTOS, 2009)

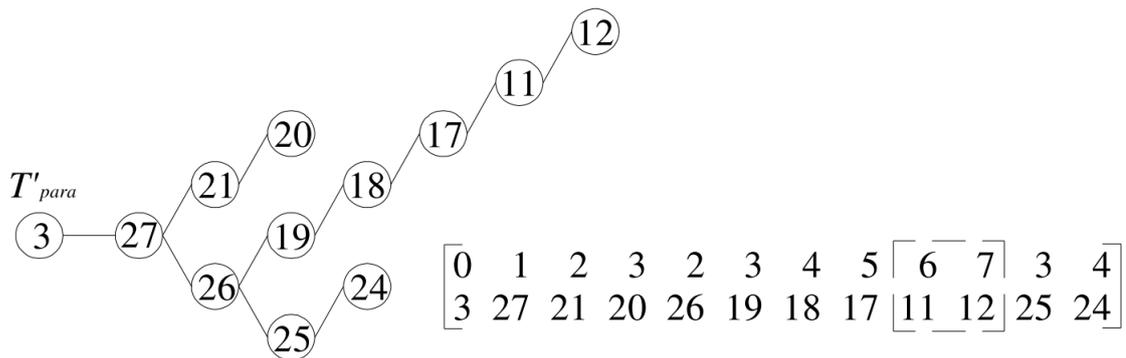
PAO, mas sim o nó r (previamente determinado), sendo este agora o nó que será enxertado no nó a .

Foram utilizados os trabalhos de (SANTOS, 2009) e (SANCHES, 2013) na descrição dos passos dos passos a seguir, sendo considerado que p , r e a foram previamente determinados, i_p e i_r como sendo os índices de p e r da árvore T_{de} e i_a sendo o índice do nó a da árvore T_{para} , e considerando $p = 10$, $r = 16$ e $a = 17$. As diferenças entre os operadores PAO e CAO consiste numa modificação no passo 3, como mostrado a seguir:

1. Determinar as posições $\{i_p - i_u\}$ dos índices em T_{de} correspondente a subárvore que contém o nó p como raiz. Sendo i_p conhecido, é necessário encontrar apenas i_u que corresponde ao índice do último nó da subárvore que contém o nó p como raiz. O conjunto $\{i_p - i_u\}$ corresponde ao nó p em i_p e aos consecutivos nós x no vetor T_{de}

Figura 11. T'_{de} e sua RNP

Fonte: (SANTOS, 2009)

Figura 12. T'_{para} e sua RNP.

Fonte: (SANTOS, 2009)

de forma que $i_x > i_p$ e $p_x > p_p$ (retângulo tracejado na Figura 13), onde p_x é a profundidade do nó x e p_p é a profundidade do nó p ;

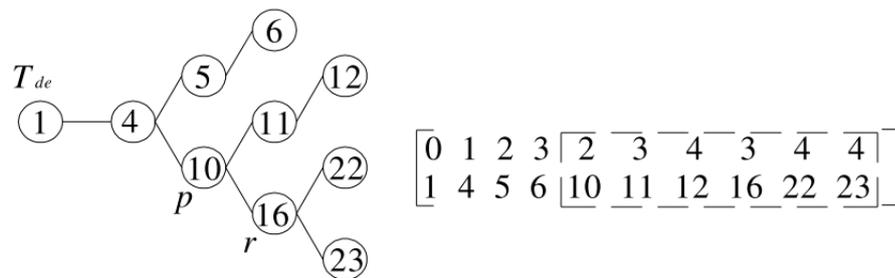
2. Copiar os dados do conjunto $\{i_p - i_u\}$ de T_{de} para um vetor temporário T_{tmp} (subárvore que está sendo transferida).
3. **Passo modificado:** Sendo considerado os nós no caminho de r a p ($r_0, r_1, r_2, \dots, r_n$, onde $r_0 = r$ e $r_n = p$) como sendo raízes de subárvores, a subárvore com a raiz r_i , $i = 1, \dots, n$, sem a subárvore com raiz r_{i-1} (Figura 14) deverá ser armazenada em T'_{tmp} , ver Figura 15. Considerando x como cada nó da subárvore com raiz r_i , a profundidade de cada nó da subárvore com raiz r_i é atualizada através da seguinte fórmula:

$$p_x = p_x - p_{ri} + p_{r(i-1)} + 1,$$

sendo que para r_0 o valor de $p_{r(i-1)} = p_a$;

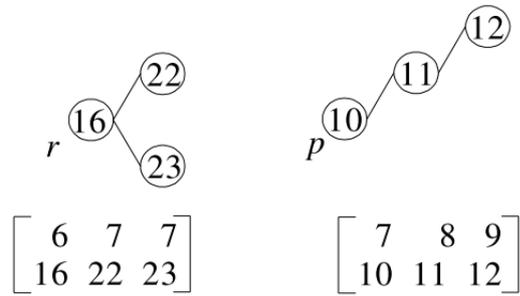
4. Armazenar num vetor T'_{para} os dados de T_{para} e T_{tmp} , inserindo T'_{tmp} a partir da posição ia , ver Figura 16;
5. Armazenar em T'_{de} os nós de T_{de} sem os nós de T_{tmp} ;

Figura 13. T_{de} e suas respectivas RNPs



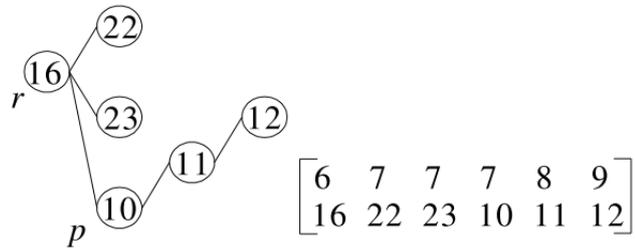
Fonte: (SANTOS, 2009)

Figura 14. Subárvores enraizadas nos nós do caminho de r até p

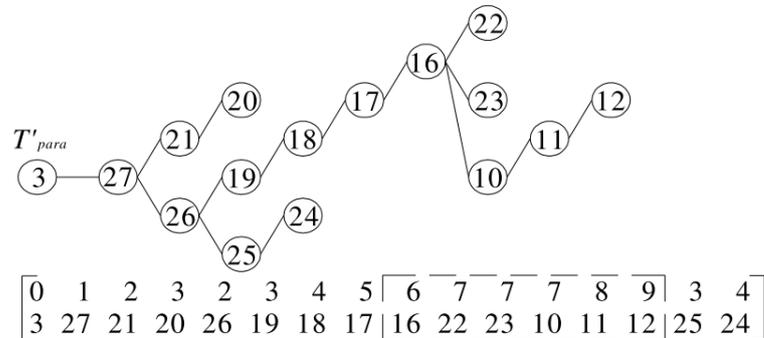


Fonte: (SANTOS, 2009)

Figura 15. T'_{tmp} e sua RNP



Fonte: (SANTOS, 2009)

Figura 16. T'_{para} e sua RNP

Fonte: (SANTOS, 2009)

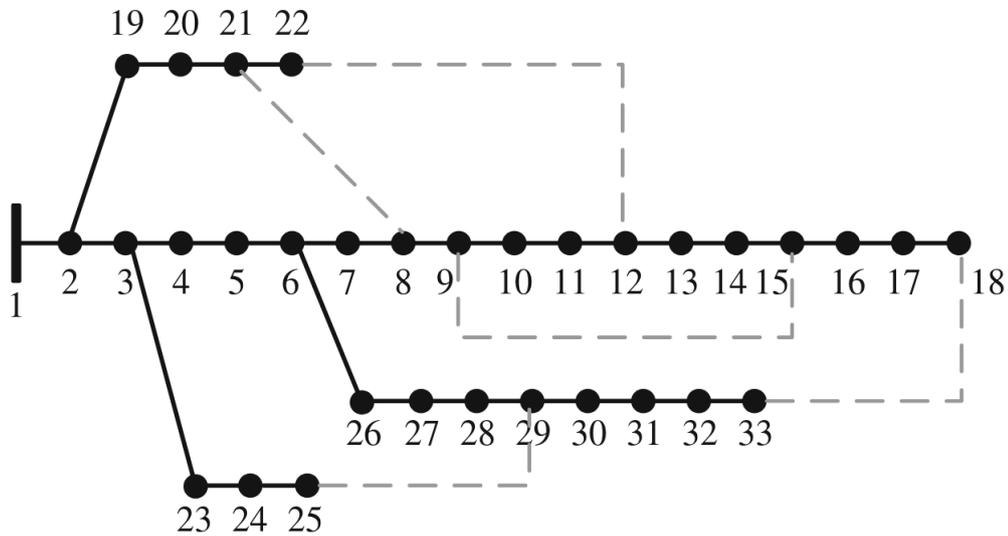
3.3.3 Determinação dos nós de poda, raiz, adjacente

Durante os estudos sobre dos operadores da RNP observou-se uma certa limitação. Nenhum deles conseguem realizar modificações dentro de uma mesma RNP, impossibilitando a transferência de carga dentro de uma mesma rede. Isto ficou mais evidente quando o programa de restauração foi executado no sistema IEEE de 33 barras, apresentado na Figura 17. A floresta que representa este sistema é uma floresta única, composta de apenas uma árvore, impossibilitando a utilização dos operadores. Isso ocorre pois na definição dos nós p , r e a o algoritmo sempre busca um nó adjacente numa árvore diferente da árvore dos nós raiz e poda.

Por esses motivos houve a necessidade de desenvolver uma nova forma de definir os nós p , r e a para que seja possível utilizar a RNP e seus operadores nesses tipos de sistemas. Ao invés de utilizar a lista de adjacência, como o algoritmo de (DELBEM et al., 2004), é utilizada a lista das chaves abertas para encontrar tais nós. Os passos deste novo algoritmo são apresentados a seguir:

- **Passo 1:** Escolher aleatoriamente uma chave aberta e define aleatoriamente um dos nós dessa chave como o nó pr e o outro como o nó a . Caso o nó pr seja o nó raiz de uma árvore é escolhida outra chave aberta pois não será possível desligar os nós das subestações. O nó pr recebe essa designação pois ele pode se tornar o nó p ou o nó r , dependendo de qual operador for escolhido;
- **Passo 2:** Localizar o índice das árvores onde os nós pr e a estão presentes. Se ambos estiverem em árvores diferentes, então vá para o **Passo 3**. Caso contrário, ou seja, ambos os nós estão presentes numa mesma árvore, verifica-se se o nó pr está no caminho do nó a até a subestação, se for verdadeiro atribui-se o valor do nó pr para

Figura 17. Sistema IEEE de 33 barras



Fonte: (HONG et al., 2017)

o nó a e vice-versa. Este passo é necessário pois o desligamento do nó pr implicará no desligamento do nó a ;

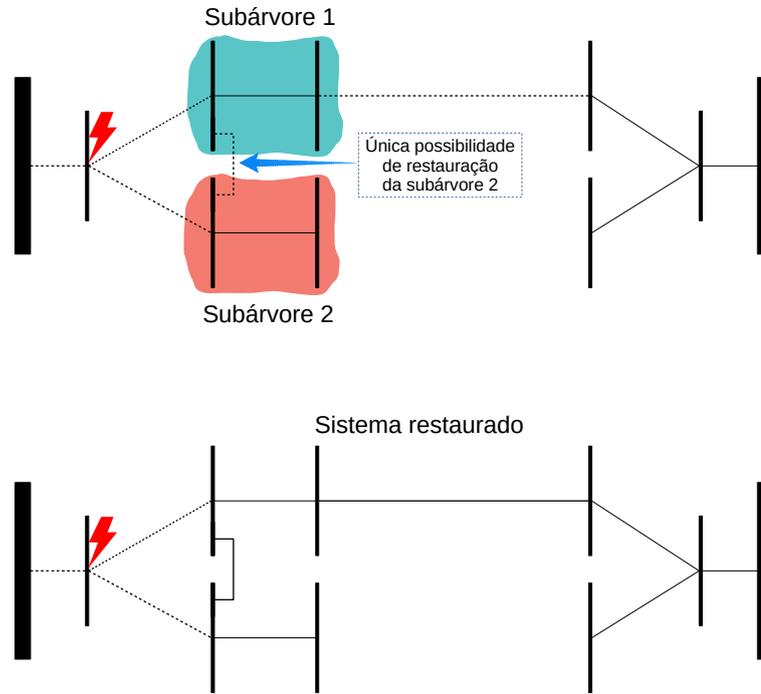
- **Passo 3:** Escolher se será aplicado o operador PAO ou o operador CAO. Se o operador PAO for escolhido, então $p = pr$. Caso o operador CAO for escolhido, então $r = pr$. Para a escolha do nó p primeiramente deve se encontrar o caminho de r até a subestação, excluindo estes dois nós do caminho. Se o caminho resultante for vazio então o nó pai de r é o nó da subestação, não sendo possível aplicar o operador CAO neste caso, somente o operador PAO com $p = r$. Se o tamanho do caminho for maior que zero então é escolhido aleatoriamente um nó deste caminho para ser o nó p .

3.4 Criação da Primeira Configuração Factível

A primeira configuração factível (PCF) é gerada a partir da reconexão do setor desenergizado após a falta, utilizando os operadores da RNP (SANTOS, 2009). Dela será derivada todas as soluções utilizadas no processo de busca.

O problema em utilizar esta abordagem existe quando uma determinada falta gera duas ou mais subárvores isoladas e uma dessas árvores só poderá ser reconectada na rede através de outra subárvore isolada. Um exemplo pode ser observado na Figura 18, onde o único caminho possível para reconectar a subárvore 2 é através da chave que conecta ela a subárvore 1, a qual também foi desconectada pela falta.

Figura 18. Possibilidade de reconexão entre as subárvores isoladas do sistemas



Fonte: Produzido pelo autor

Considerando $S = \{s_1, s_2, \dots, s_n\}$ o conjunto de subárvores desenergizadas após o isolamento da falta, $P = \{p_1, p_2, \dots, p_n\}$ o conjunto de chaves NA conectadas em S e L a lista de nós presentes em S , os passos a seguir apresentam o algoritmo capaz de criar a PCF contornando esse problema:

- Enquanto existirem chaves NA em P , faça:
 - Passo 1: Selecionar uma chave NA de P e armazenar em sw , se ambos os nós de sw estiverem presentes em L , outra chave NA é selecionada;
 - Passo 2: Definir o nó de poda p como o nó de sw presente em L e o outro nó como sendo o nó adjacente a ;
 - Passo 3: Definir s_i como sendo a subárvore em que p está presente, e aplicar o operador PAO utilizando s_i como T_{tmp} (ver Seção 3.3.1);
 - Passo 4: Remover s_i de S , remover as chaves NA conectadas em s_i de P deixando somente as chaves que estejam conectadas em outras subárvores e remover os nós presentes em s_i de L .

3.5 Fluxo de Carga para SDE

O estudo do fluco de carga consiste em obter as condições de operação dela, ou seja, os valores de tensão nas barras, correntes, perdas do sistema e a potência que flui nas linhas energizadas (SANTOS, 2009). Com esses valores é possível obter os dados necessários para calcular as restrições do problema de restauração.

O método utilizado para os estudos de fluxo de carga é o *backward/forward* por serem rápidos e bastante utilizados em sistemas radiais. Ele possui duas versões, a soma de potências (BARAN; WU, 1989) (CESPEDES, 1990) e a soma de correntes (SHIRMOHAMMADI et al., 1988), sendo esta última a versão utilizada neste trabalho.

O método *backward/forward* com soma de correntes consistem em calcular as tensões indo dos nós mais externos de uma rede (nós folhas) até a subestação e vice-versa. Inicialmente é definido valores para as tensão nodais, em seguida é realizada uma varredura inversa (*backward*), onde os valores das correntes nas linhas são calculados. Em seguida é realizada a varredura direta (*forward*), em que os valores das tensões nodais são calculados partindo da subestação até os nós folhas. Estes passos são repetidos até que algum critério de convergência seja satisfeito (SANCHES, 2013).

O fluxo de carga é executado para cada árvore da floresta após uma modificação na topologia da rede, sendo necessário realizar esse estudo somente nas árvores afetadas. Foi utilizado o trabalho de (MANSOUR, 2009) para a escrita dos passos mais detalhados deste método, apresentado a seguir:

1. **Pré-processamento:** Considerando m o número de nós do sistema e n o número de ramos, inicialmente é definido as tensões complexas em todos os nós iguais a $1 + 0j$;
2. **Cálculo da corrente nodal:** Na iteração k , as correntes complexas injetadas nos nós são calculadas utilizando a fórmula:

$$I_i^{(k)} = \left(\frac{S_i}{V_i^{(k-1)}} \right)^* - Y_i^{sh} V_i^{k-1}, i = 1, 2, \dots, n$$

sendo V_i^{k-1} a tensão na barra i , calculada na iteração $k - 1$, S_i a potência complexa da barra i , Y_i^{sh} a soma de todos os elementos *shunt* da barra i e n o número de nós da árvore;

3. **Backward:** Na iteração k , partindo das linhas mas extremas da rede até as linhas conectadas ao nó raiz, calcula-se a corrente J_L na linha L , que conecta a barra $L2$ à $L1$, como mostra a Figura 19, sendo:

$$J_L^{(k)} = I_{L2}^{(k)} + \sum(\text{Correntes nas linhas que saem do nó } L2)$$

onde $L = [p, p - 1, \dots, 1]$, $I_{L2}^{(k)}$ é a corrente no nó $L2$ e p é o número de linhas do sistema;

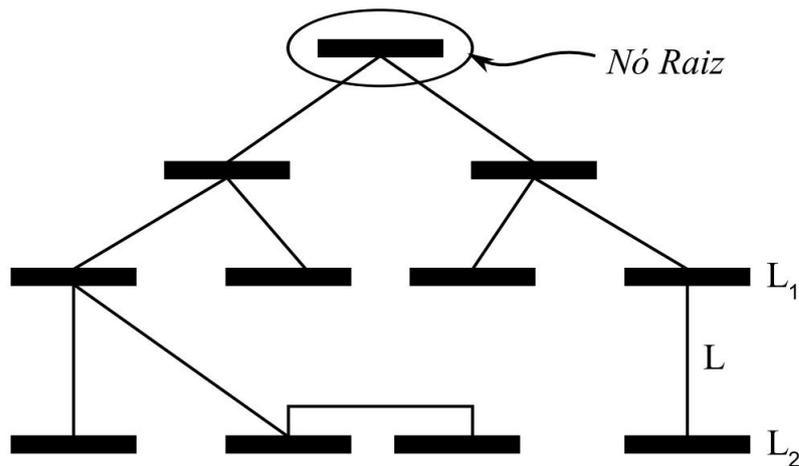
4. **Forward:** Atualiza as tensões nas barras a partir daquelas que estão conectadas à raiz da árvore (subestação) até as barras mais extremas do sistema. A tensão de $L2$ é calculada utilizando a tensão de $L1$ calculada na iteração k e o fluxo de corrente na linha calculado no passo 3:

$$V_{L2}^{(k)} = V_{L1}^{(k)} - Z_L J_L^{(k)}, \quad L = 1, 2, \dots, p$$

sendo Z_L a impedância série da linha L ;

5. Repete os passos 2, 3 e 4 até que um critério de convergência seja satisfeito.

Figura 19. Etapa de varredura inversa (*backward*)



Fonte: (MANSOUR, 2009)

O critério de parada é satisfeito quando a diferença do valor de tensão entre duas iterações consecutivas for menor que um valor de erro $e = 10^{-8}$. Para a prevenção de possíveis loops infinitos que possam ocorrer durante a execução dos algoritmos, foi adicionado um limite de 100 iterações que o estudo de fluxo de carga pode realizar, caso esse valor seja atingido sem que a convergência tenha sido estabelecida o programa é parado, retornando um erro de execução.

3.6 Cálculo do Número de Manobras

O objetivo dos algoritmos avaliados é diminuir o número de manobras após a restauração, cumprindo as devidas restrições. Essas restrições são obtidas através do

cálculo do Fluxo de Carga, visto na Seção 3.5, já o número de Manobras pode ser obtido através da comparação entre os estados das chaves NAs e NFs da configuração que está sendo avaliada com a configuração original, entretanto essa estratégia possui um alto custo computacional, pois a cada nova solução produzida pelas meta-heurísticas é necessário a comparação entre dois vetores do tamanho do número de nós existentes no sistema.

Em (SANTOS, 2009) foi proposto um algoritmo capaz de determinar o número de manobras apenas comparando os estados das chaves que foram alteradas. As modificações na rede realizadas pelos algoritmos CAO e PAO sempre geram alterações de chaves em pares, onde uma chave é aberta e outra fechada, diante desta característica é possível determinar o número de manobras comparando o par de chaves modificadas da configuração original com a configuração gerada.

Considerando O a configuração original, Q a configuração avaliada e P a configuração pai de Q , ou seja, Q foi gerada ao aplicar os operadores da RNP em P , i e j como sendo as chaves modificadas e NM o número de manobras. O Algoritmo 1 mostra o funcionamento do cálculo do número de manobras da configuração avaliada. Caso o valor das chaves modificadas sejam os mesmos da configuração original, o número de manobras de Q é o número de manobras de sua configuração pai acrescido de 2, se ambos forem diferentes este número é decrescido de 2 e para qualquer outro caso o número de manobra é o mesmo da configuração pai.

Algoritmo 1: Cálculo do Número de Manobras

```

1 if  $O(i) = Q(i)$  and  $O(j) = Q(j)$  then
2   |  $NM(Q) \leftarrow NM(P) - 2$ 
3 else if  $O(i) \neq Q(i)$  and  $O(j) \neq Q(j)$  then
4   |  $NM(Q) \leftarrow NM(P) + 2$ 
5 else
6   |  $NM(Q) \leftarrow NM(P)$ 

```

4 Algoritmos de Otimização para o Problema de Restauração

Neste Capítulo é apresentada uma pequena introdução sobre as meta-heurísticas e suas divisões, seguidos dos algoritmos a serem avaliados por este trabalho.

4.1 Meta-heurísticas

As meta-heurísticas podem ser definidas como o conjunto de algoritmos de otimização que possuem a capacidade de encontrar boas soluções para alguns problemas (GENDREAU; POTVIN et al., 2010) (LUKE, 2013). Elas podem ser divididas e agrupadas baseadas em suas características, por exemplo os Algoritmos Evolutivos utilizam a Evolução Natural proposta por Darwin através da disputa entre indivíduos e a sobrevivência do mais apto, já a Inteligência de Enxames utiliza a cooperação entre os indivíduos no processo de busca.

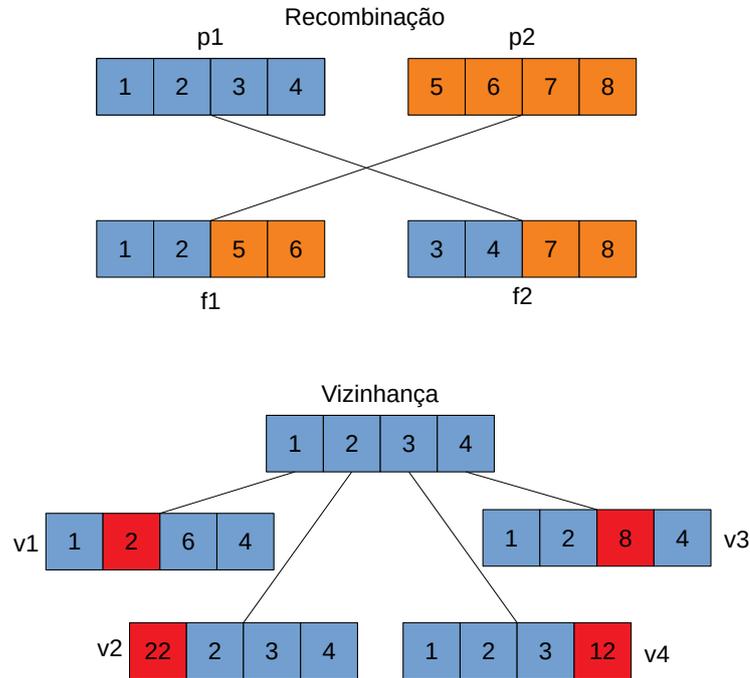
Outra divisão possível é relacionada ao processo de criação de novas soluções. No processo de busca das meta-heurísticas, elas precisam ter algum mecanismo capaz de produzir novas soluções com base nas existentes. Este processo pode ser realizado através da recombinação, ou cruzamento, de duas ou mais soluções, etapa que ocorre nos Algoritmos Genéticos, onde dois indivíduos são mesclados para gerar novas soluções.

Outra forma de produzir novas soluções é através da criação de uma vizinhança, onde um conjunto de soluções é gerada a partir de pequenas modificações numa determinada solução. Esse conjunto é denominado de vizinhança pois seus elementos apresentam características bem próximas da solução originária do conjunto.

Esta estratégia é amplamente utilizada por algoritmos de busca local e também pelo Algoritmo Genético na etapa de Mutação. Diferentemente da recombinação, a criação da vizinhança é realizada a partir de uma única solução. A Figura 20 exemplifica essas diferenças, na etapa de Recombinação, dois pais (p1 e p2) geram dois filhos (f1 e f2), ambos os filhos apresentam características parecidas com as dos pais ao mesmo tempo em que um filho é bem diferente de um pai. Já na etapa de Vizinhança, todos os vizinhos possuem grande semelhança com a solução original, apresentando apenas uma pequena diferença com esta.

Os operadores computacionais da RNP são algoritmos de vizinhança, gerando apenas uma única solução a partir de outra, sendo a única forma de produzir novas soluções. Por esse motivo, os algoritmos capazes de solucionar o problema de restauração

Figura 20. Diferença entre recombinação e vizinhança



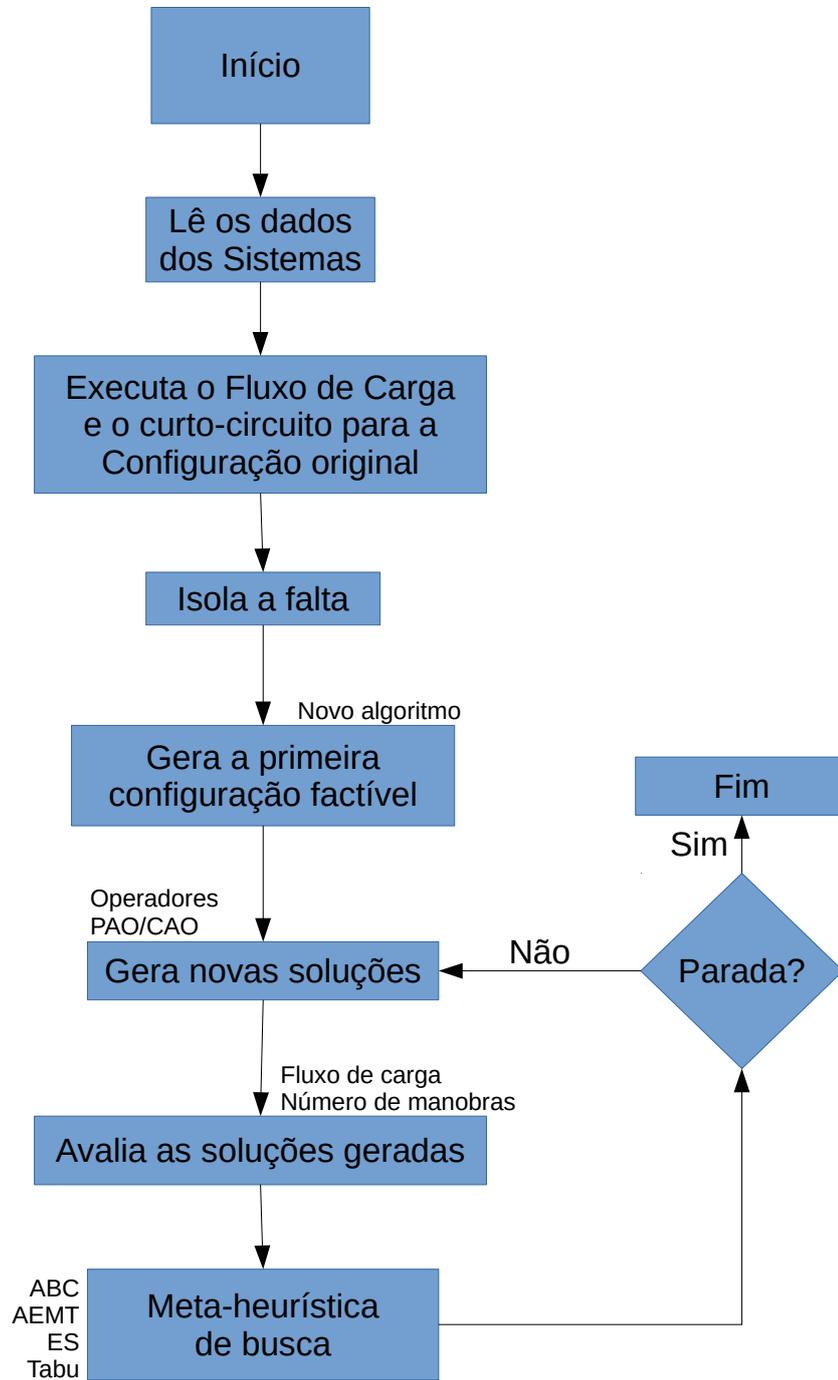
Fonte: Produzido pelo autor

ficam limitados àqueles que utilizem da vizinhança como estratégia de diversificação. Alguns deles foram escolhidos para a análise de desempenho deste trabalho, sendo eles: Busca Tabu, ES e ABC e AEMT. Esses algoritmos foram escolhidos por serem bem explorados na literatura, apresentando bons resultados para outros tipos de problemas de otimização.

Das meta-heurísticas avaliadas, o ABC e o ES nunca foram implementadas para o problema de restauração utilizando a RNP. É importante realizar esta avaliação para analisar o comportamento de outras meta-heurísticas para este problema e verificar se é possível obter resultados melhores que o AEMT.

Uma visão geral das meta-heurísticas utilizadas pode ser observado na Figura 21. Este fluxograma apresenta do processo de restauração utilizando a RNP com as quatro meta-heurísticas aqui avaliadas. Primeiramente, após a falta, é gerada a primeira configuração factível. Após isso o algoritmo entra num *loop* realizando a busca de uma boa solução através de uma meta-heurística, utilizando os operadores da RNP para produzir novas soluções e o Fluxo de Carga para avaliá-las. Este processo é executado até que alguma condição de parada seja satisfeita.

Figura 21. Fluxograma da restauração utilizando Meta-heurísticas



Fonte: Produzido pelo autor

4.2 Busca Tabu

A Busca Tabu foi proposta por Glover em 1986 (GLOVER, 1989). Ela é uma meta-heurística de busca local onde as alterações que proporcionam novas soluções são armazenadas numa lista, denominada de Lista Tabu e caso uma nova solução apresente alterações que estejam presentes nesta lista ela será descartada, como mostra o Algoritmo 2.

A lista tabu possui um número máximo de alterações que poderão ser armazenadas, quando ela ficar totalmente preenchida, as alterações mais antigas são removidas da lista e dão lugar a novas alterações. Neste trabalho, a lista tabu será composta das chaves que foram modificadas na utilização do PAO/CAO.

Algoritmo 2: Busca Tabu

```

1 pcf = Primeira Solução Factível;
2 MelhorSolução = pcf;
3 SoluçãoAtual = pcf;
4 tabu =  $\emptyset$ ; // Lista Tabu
5 maxTabu = n; // Tamanho máximo da Lista Tabu
6 while Condição de parada não é satisfeita do
7   V = Vizinhaça(MelhorSolução); // Gera a vizinhaça com PAO/CAO
8   V = sort(V); // Ordena a vizinhaça
9   if  $V_0 < \text{MelhorSolução}$  then
10    | MelhorSolução =  $V_0$ ;
11  end
12  // Atualização da Lista Tabu
13  for  $i=1$  until length(V) do
14    | if  $V_i$  não está na lista tabu then
15      | SoluçãoAtual =  $V_i$ ;
16      | if length(tabu) == maxTabu then
17        | tabu.remove(0)
18      | end
19      | tabu.append( $V_i$ );
20      | break;
21    | end
22  end
23 Retorna MelhorSolução;

```

4.3 Artificial Bee Colony (ABC)

O ABC é um algoritmo de inteligência de enxames desenvolvido por (KARABOGA, 2005). O algoritmo ABC é baseado na busca das abelhas por novas fontes de alimento. Ele é constituído basicamente de uma sequência de gerações de vizinhança e seleções, alternando entre gulosa e por torneio ou roleta. Por último, é realizada uma etapa de exploração, onde é verificada se as soluções mantidas estão a um determinado número preestabelecido de iterações sem melhora, ou seja, se ela não for substituído por um vizinho nas seleções gulosas esta solução é descartada e outra solução é gerada aleatoriamente.

Um dos principais pontos do ABC está na sua fácil implementação e dos poucos parâmetros presentes, basicamente só será necessário definir o número de soluções mantidas em O e o número de iterações que uma solução poderá ser armazenada no algoritmo (SHUKRAN et al., 2011). O algoritmo do ABC é apresentado no Algoritmo 3

Algoritmo 3: ABC

```

1  $pcf$  = Primeira Solução Factível;
2 MelhorSolução =  $pcf$ ;
3  $O$  = PopulaçãoInicial ; // Gerada através dos operadores PAO/CAO em  $pcf$ 
4 while Condição de parada não é satisfeita do
5    $O'$  = Vizinhança( $O$ );
6    $B$  = Gulosa( $O, O'$ ) ; // Seleção gulosa entre os vetores  $O$  e  $O'$ 
7    $V$  = torneio( $B$ ) ; // Seleção por torneio em  $B$ 
8    $V'$  = Vizinhança( $V$ );
9    $F$  = Gulosa( $V, V'$ ) ; // Seleção gulosa entre os vetores  $V$  e  $V'$ 
10  MelhorSolução = best( $O, O', B, V, V', F$ ) ; // Melhor solução da iteração
11  for  $i=1$  until length( $O$ ) do
12    if  $O_i$  não apresentou melhora then
13      |  $O_i$  = Vizinhança( $O_i$ );
14    end
15  end
16 end
17 Retorna MelhorSolução;

```

4.4 Evolutionary Strategies (ES)

O ES é um dos primeiros algoritmos baseados na teoria da evolução. Ele foi criado por Rechenberg na década de 1960 e sua primeira versão era um algoritmo de busca local. No decorrer dos anos ele foi evoluindo e seus formatos mais usados são ES - $(\mu + \lambda)$ e ES (μ, λ) , onde o símbolo μ denota o número de pais de todas as gerações, e λ refere-se à quantia de descendentes de gerados por μ (BACK; HOFFMEISTER; SCHWEFEL, 1991).

Os símbolos " + " e ", " representam o tipo de seleção aplicado aos indivíduos gerados. De acordo com o (BEYER; SCHWEFEL, 2002), na seleção de tipo ", μ pais geram λ descendentes, sendo $\lambda > \mu$ e a partir do λ filhos, os melhores μ serão selecionados para a próxima geração. Na seleção do tipo " + " λ filhos são gerados, contudo $\lambda < \mu$ e seleção dos melhores indivíduos será realizada com a união de filhos e pais de cada geração. Segundo (BEYER; SCHWEFEL, 2002) cada uma das formas de seleção de estratégias evolutivas tem sua aplicação característica. Tipo de seleção ", " é mais indicado para aplicações com espaços de pesquisa ilimitados, e o tipo de seleção " + " é recomendado para espaços de busca discretos e finitos, por isso a seleção do tipo " + " O Algoritmo 4 representa o fluxograma do ES- $(\mu + \lambda)$.

Algoritmo 4: Estratégias Evolutivas $\mu + \lambda$

```

1 Gera uma população inicial de  $\mu$  soluções;
2  $pcf$  = Primeira Solução Factível;
3  $pais_{\mu}$  = População Inicial ; // Gerada através do PAO/CAO em  $pcf$ 
4 while Condição de parada não é satisfeita do
5   |  $filhos_{\lambda}$  = Vizinhaça( $pais_{\mu}$ );
6   |  $pop_{\mu+\lambda}$  =  $pais_{\mu}$  +  $filhos_{\lambda}$  ; // Uni os pais com os filhos
7   |  $pop_{\mu+\lambda}$  =  $sort(pop_{\mu+\lambda})$  ; // Ordena a população
8   |  $pais_{\mu}$  =  $pop_{\mu}$  ; // Seleciona as melhores  $\mu$  soluções dos pais
9 end
10 Retorna a melhor solução de  $pais_{\mu}$ ;
```

4.5 AEMT - Algoritmo Evolutivo Multiobjetivo por Tabelas

O AEMT utiliza subpopulações de soluções distribuídas em tabelas no processo de otimização (SANTOS, 2009). Ele foi desenvolvido com o intuito de solucionar problemas multi-objetivos e multi-restritivos. Cada tabela do AEMT armazena as melhores soluções segundo cada objetivo e/ou restrição (se mensurável), sendo que uma das tabela deverá conter as melhores soluções com base na função de agregação, onde serão agregados todos os objetivos e restrições, bem como a punição por violação destas (??). Mesmo sendo um algoritmo multiobjetivo, o AEMT pode ser aplicado a problemas moo-objetivos e multi-restritivos, como o problema de restauração.

O Algoritmo 5 apresenta o pseudocódigo do AEMT. Primeiramente as tabelas são geradas a partir da aplicação da função de vizinhança de uma solução inicial. Em seguida um indivíduo é selecionado aleatoriamente de qualquer umas das tabelas e é aplicado a função de vizinhança nele para gerar uma solução candidata que é comparada com as piores soluções de cada tabela. Caso seja melhor que uma delas, a solução candidata a substitui, permitindo que uma solução possa estar presente em mais de uma tabela.

Algoritmo 5: AEMT - Algoritmo Evolutivo Multiobjetivo por Tabelas

```

1 Gera uma população inicial de  $\mu$  soluções;
2  $pcf$  = Primeira Solução Factível;
3  $pop$  = PopulaçãoInicial ; // Gerada através dos operadores PAO/CAO em  $pcf$ 
4  $tabelas$  =  $dist(pop)$  ; // Distribui as soluções de  $pop$  nas tabelas
5 while Condição de parada não é satisfeita do
6   |  $tabelas$  =  $sort(tabelas)$  ; // Ordena as tabelas
7   |  $ind$  =  $random(tabelas)$  ; // Seleciona um indivíduo aleatório de uma
   |   tabela aleatória
8   |  $newInd$  =  $Vizinhança(ind)$  ; // Aplica os operadores em  $ind$ 
9   | for  $i=1$  until  $length(tabelas)$  do
10  |   | if  $newInd$  for melhor que a pior solução de  $tabelas_i$  then
11  |   |   | Substitui a pior solução por  $newInd$ ;
12  |   |   | end
13  |   | end
14 end
15 Retorna a melhor solução da tabela da função de agregação;

```

São utilizadas cinco tabelas no AEMT, sendo uma para a função de agregação, uma para o número de manobras e outras três para as restrições do carregamento das linhas, carregamento das subestações e níveis de tensões.

5 Resultados Computacionais

Nesta seção serão apresentadas as análises estatísticas utilizadas para validar a análise de desempenho, os sistemas testes e os resultados computacionais com uma discussão sobre a eficiência dos algoritmos avaliados.

Além de analisar os resultados finais da Função de Agregações, que contém o número de manobras e as restrições operacionais gerados pelos algoritmos, também será analisado o tempo de execução de cada algoritmo.

Todos os algoritmos e o fluxo de carga foram implementados na linguagem Python 2.7 e executados no interpretador PyPy, sem o uso de bibliotecas adicionais. Os testes foram executados num computador i5-8400, 16 GB de memória RAM e sistema operacional Ubuntu 18.04.3. Cada falta foi simulada 100 vezes para cada algoritmo, totalizando 4400 execuções.

5.0.1 Análise Estatística

Para a validação dos algoritmos selecionados é necessário uma análise estatística capaz de lidar com mais de um conjunto de dados. Inicialmente pretendia-se utilizar o teste de ANOVA (*Analysis of Variance*) para verificar se existia diferença significativa entre os resultados obtidos.

A ANOVA só pode ser aplicada em conjuntos de dados que seguem a distribuição normal, por isso foi executado o teste de Kolmogorov-Smirnov para verificar se os dados da Função de Agregação seguem essa distribuição. O resultado deste teste mostrou que os dados analisados não seguem a distribuição normal, impossibilitando o uso da ANOVA e sendo necessário executar o teste de Kruskal-Wallis, que é a versão não-paramétrica da ANOVA, para validar os dados.

O teste de Kruskal-Wallis foi executado com uma significância $\alpha = 0.05$ e com as seguintes hipóteses:

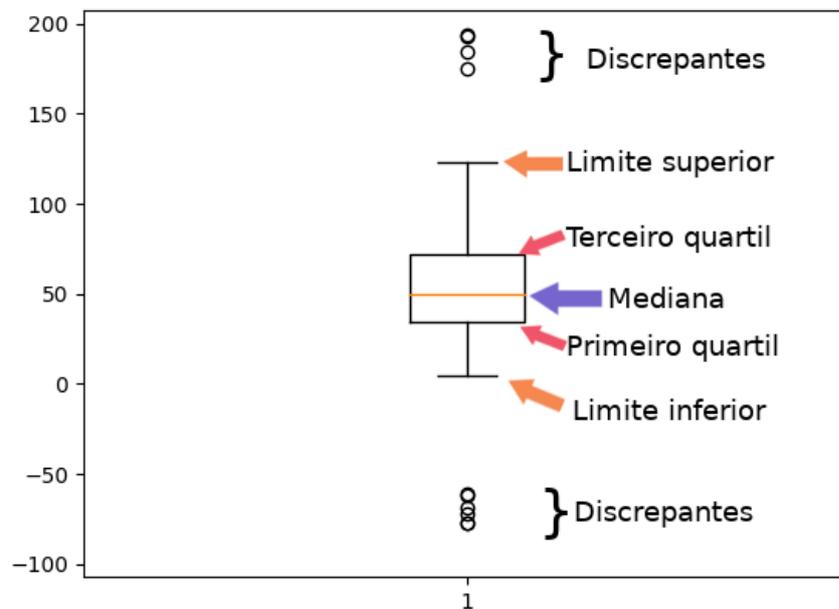
- H_0 ($\rho > \alpha$): Há diferença significativa entre os resultados dos algoritmos;
- H_1 ($\rho < \alpha$): Não existe diferença significativa entre os resultados dos algoritmos.

Como o teste de Kruskal-Wallis não avalia qual algoritmo apresenta resultados melhores do que outro, caso H_1 seja verdadeiro, é necessário utilizar o teste de Dunn, para verificar qual algoritmo apresentou os menores valores da Função de Agregação, que realiza comparações de todos os resultados avaliados dois a dois, utilizando as mesmas hipóteses do teste de Kruskal-Wallis. Se H_0 for aceito então não há diferença entre os

seus resultados, caso contrário utiliza-se o *boxplot* para verificar qual algoritmo apresentou o melhor resultado. Todos os dados dos resultados dos testes de Kolmogorov-Smirnov, Kruskal-Wallis e Dunn são apresentados no Apêndice A.

O *boxplot* é utilizado por apresentar diversos dados estatísticos numa só imagem, sendo eles o Limite Inferior, Limite Superior, Primeiro e Terceiro Quartil, Mediana e os dados discrepantes, como mostra a Figura 22.

Figura 22. Dados estatísticos presentes no *boxplot*



Fonte: Produzido pelo autor

5.1 Sistemas Testes

As faltas foram simuladas em três SDEs: Sistemas 84-barras, 119-barras e 135-barras. O sistema 119-barras (ZHANG; FU; ZHANG, 2007) foi escolhido por apresentar apenas uma subestação, logo sua representação computacional utilizando a RNP é composta de uma floresta única, ideal para avaliar o desempenho dos algoritmos com o novo algoritmo de obtenção dos nós de poda, raiz e adjacente, vistos na Seção 3.3.3, e com o algoritmo para a obtenção da Primeira Configuração Factível, visto na Seção 3.4. Neste sistema não será possível avaliar a restrição de subestação pois quaisquer alteração na sua topologia não irá alterar o carregamento da subestação.

O sistema 84-barras (SU; LEE, 2003) também apresenta uma única subestação, por isso ele foi adaptado para ser possível avaliar a restrição de subestação. Nele cada

alimentador é considerado uma subestação, totalizando 11 subestações (*A* a *K*), quando na realidade existe apenas uma subestação.

O sistema 135-barras (MANTOVANI; CASARI; ROMERO, 2000) foi escolhido por apresentar duas subestações e onze alimentadores, possibilitando a transferência de cargas internamente e externamente as subestações, podendo ser aplicadas tanto as extensões dos operadores da RNP, propostas por este trabalho, quanto as desenvolvidas por (DELBEM et al., 2004).

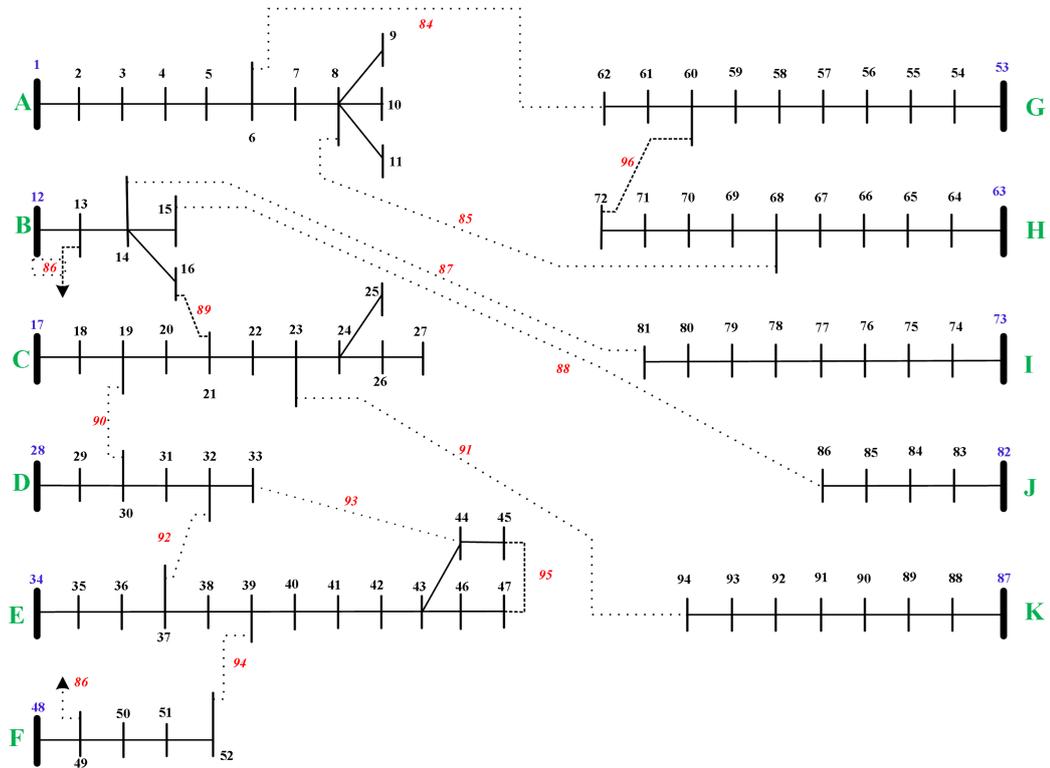
A Tabela 2 apresenta as características deles e as barras onde foram simuladas as faltas e as Figuras 23, 24 e 25 mostram a topologia inicial de cada sistema, onde as linhas contínuas são as chaves NF e as tracejadas as chaves NA. As barras de falta foram escolhidas por deixarem um grande setor desconectado após o isolamento da falta, tornando assim o problema mais complexo. Foi considerado também que todos os trechos possuem chaves seccionadoras automáticas.

Tabela 2. Características dos sistemas de distribuição avaliados.

	Sistemas		
	84	119	135
Tensão Base (kV)	11,4	11	13,8
Carga Ativa Total (MVA)	28,35	22,71	18,28
Carga Reativa Total (MVA _r)	20,70	17,04	7,91
Qtd. de Alimentadores	11	3	8
Qtd. de Subestações	11	1	2
Qtd. da Chaves NA	13	15	21
Qtd. de Chaves NF	83	118	134
Tensão Máxima Pré-falta (p.u.)	1	1	1
Tensão Mínima Pré-falta (p.u.)	0,928	0,869	0,931
Barras de Falta	18, 36, 66	10, 28, 63, 100	21, 49, 95, 107

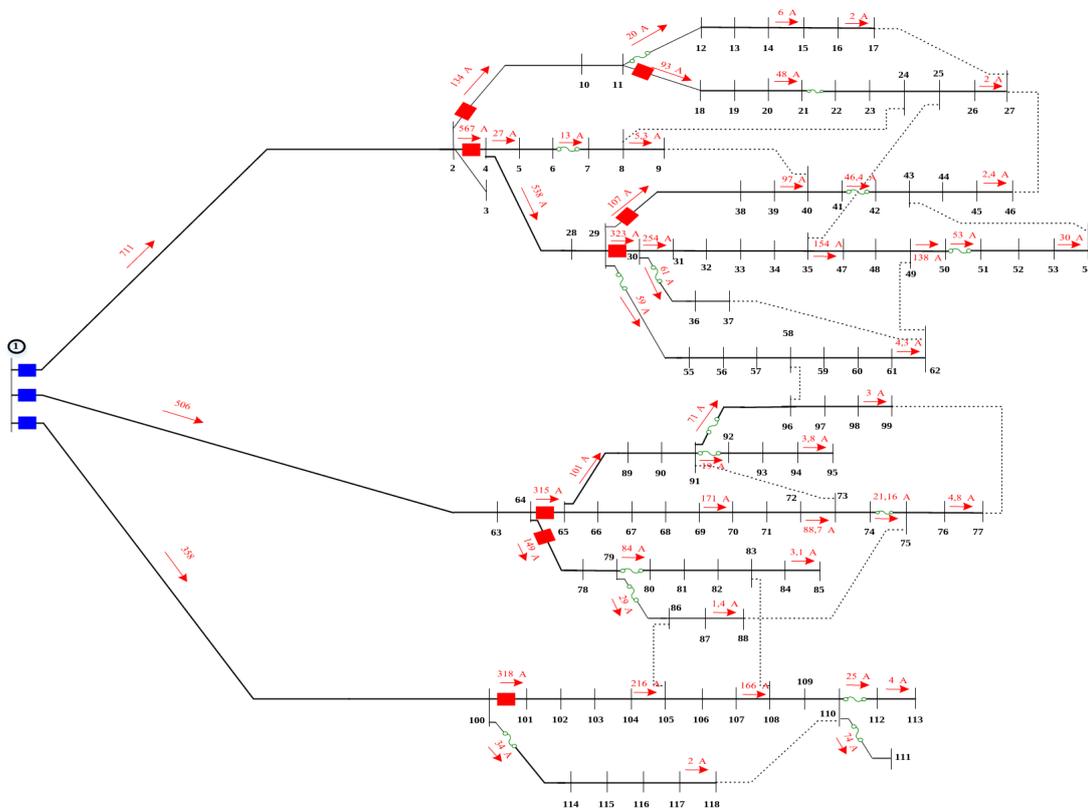
Fonte: Produzido pelo autor

Figura 23. SDE de 84 barras



Fonte: Produzido pelo autor

Figura 24. SDE de 119 barras



Fonte: Produzido pelo autor

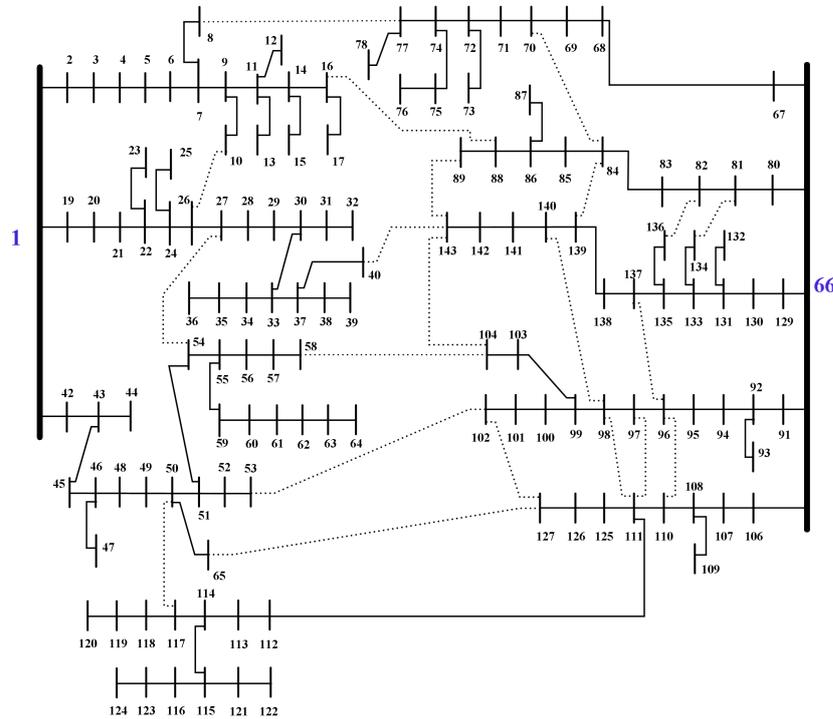
5.2 Resultados

O teste de Krustall-Wallis foi executado com o objetivo de verificar se a hipótese nula será rejeitada ou aceita. Em todos os resultados a hipótese nula (H_0) foi rejeitada, portanto, em todos os testes, existe pelo menos um grupo que se difere do restante.

O teste de Dunn de cada teste é apresentado em suas respectivas seções. Para facilitar a análise dos resultados, o teste de Dunn informam apenas se a hipótese nula é aceita ou rejeitada. Os valores numéricos de todos os testes estão presentes no Apêndice A.

Também são apresentados os melhores resultados encontrados pelo algoritmo que encontrou os menores valores da Função de Agregação. Se dois ou mais algoritmos conseguirem chegar em soluções ótimas, apenas um deles será escolhido para a exibição da melhor solução.

Figura 25. SDE de 135 barras



Fonte: Produzido pelo autor

5.2.1 Falta Única no Sistema 84-barras

5.2.1.1 Falta na barra 18

A Figura 26 apresenta os resultados da Função de Agregação para a falta na barra 18. O AEMT apresentou os melhores resultados em relação aos outros algoritmos, seguido pelo ABC, ES e a Busca Tabu. Aparentemente o ABC e o AEMT apresentam resultados parecidos, contudo o teste de Dunn, apresentado na Tabela 3, mostra que existe sim uma diferença significativa entre esses dois algoritmos e que essa diferença não existe entre o ABC e o ES, logo esses algoritmos apresentam resultados semelhantes.

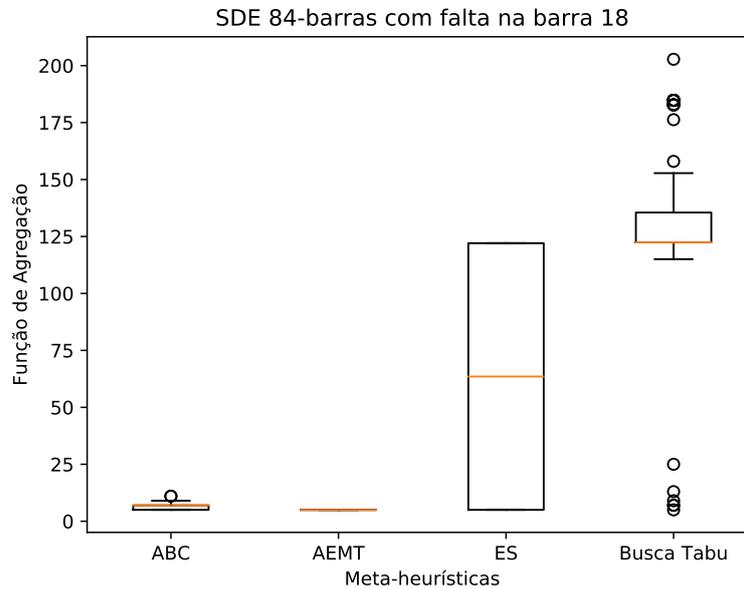
Tabela 3. Teste de Dunn para a falta na barra 18

	AEMT	ES	TS
ABC	Rejeita	Aceita	Rejeita
AEMT	-	Rejeita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

A Tabela 4 apresenta os resultados da melhor solução encontrada pelo AEMT, nesta solução pode-se observar o cumprimento de todas as restrições do sistema com pouquíssimas manobras necessárias para realizar esta tarefa.

Figura 26. Resultados da falta na barra 18



Fonte: Produzido pelo autor

Tabela 4. Resultado do AEMT falta na barra 18

Função de agregação	Número de manoras	Carr. dos trechos (%)	Carr. das subestações (%)	Menor tensão
5	5	58,80	76,24	0,928

5.2.1.2 Falta na barra 36

A Figura 27 apresenta os resultados da Função de Agregação para a falta na barra 36. O AEMT e o ES conseguiram convergir para um ótimo local, seguidos do ABC e da Busca Tabu. A Tabela 5 mostra que não há diferença entre os resultados do AEMT e do ES pois a hipótese nula foi aceita.

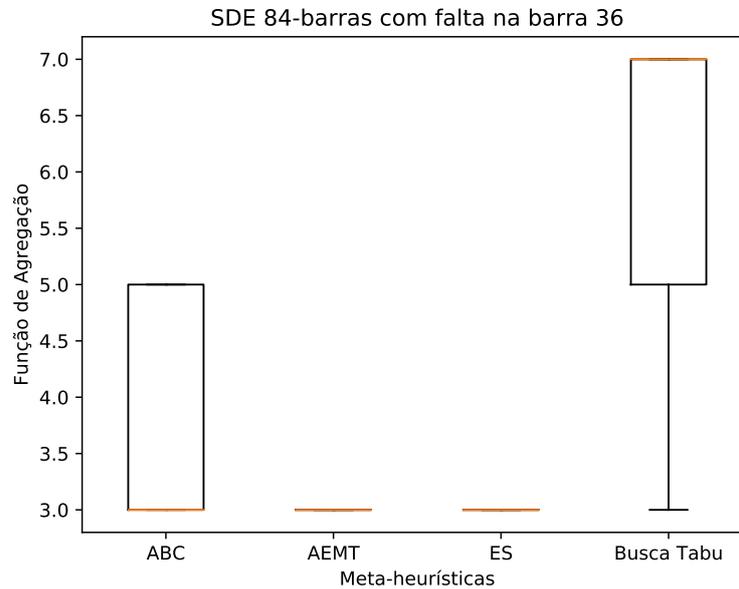
Tabela 5. Teste de Funn para a falta na barra 36

	AEMT	ES	TS
ABC	Rejeita	Rejeita	Rejeita
AEMT	-	Aceita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

Nesta simulação todos os algoritmos apresentaram um bom desempenho, mesmo que o ABC e a Busca Tabu não atingindo o ótimo global, os resultados destes algoritmos apresentaram cumpriram todas as restrições se mostrando eficazes para esta simulação.

Figura 27. Resultados da falta na barra 36



Fonte: Produzido pelo autor

A Tabela 6 apresenta os resultados da melhor solução encontrada pelo ES, nesta solução pode-se observar o cumprimento de todas as restrições do sistema com pouquíssimas manobras necessárias para realizar esta tarefa.

Tabela 6. Resultado do ES falta na barra 36

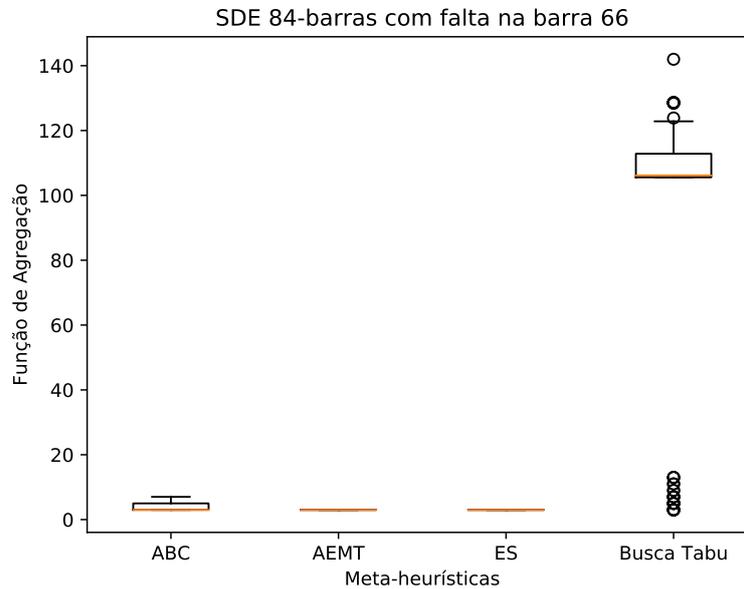
Função de agregação	Número de manoras	Carr. dos trechos (%)	Carr. das subestações (%)	Menor tensão
3	3	40,35	80,91	0,928

5.2.1.3 Falta na barra 66

A Figura 28 apresenta os resultados da Função de Agregação para a falta na barra 66. Assim como a falta na barra 36, o AEMT e o ES apresentaram resultados similares, aceitando a hipótese nula apresentada na Tabela 7 e convergindo para um ótimo global, seguidos do ABC e da Busca Tabu. Nesta Simulação a Busca Tabu apresentou uma maior diferença com relação aos outros algoritmos, mesmo que algumas soluções apresentassem resultados próximos deles, a maior parte das execuções apresentaram soluções ineficazes e distantes das soluções geradas pelos outros algoritmos.

A Tabela 8 apresenta os resultados da melhor solução encontrada pelo AEMT, nesta solução pode-se observar o cumprimento de todas as restrições do sistema com pouquíssimas manobras necessárias para realizar esta tarefa.

Figura 28. Resultados da falta na barra 66



Fonte: Produzido pelo autor

Tabela 7. Teste de Dunn para a falta na barra 66

	AEMT	ES	TS
ABC	Rejeita	Rejeita	Rejeita
AEMT	-	Aceita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

Tabela 8. Resultado do AEMT falta na barra 66

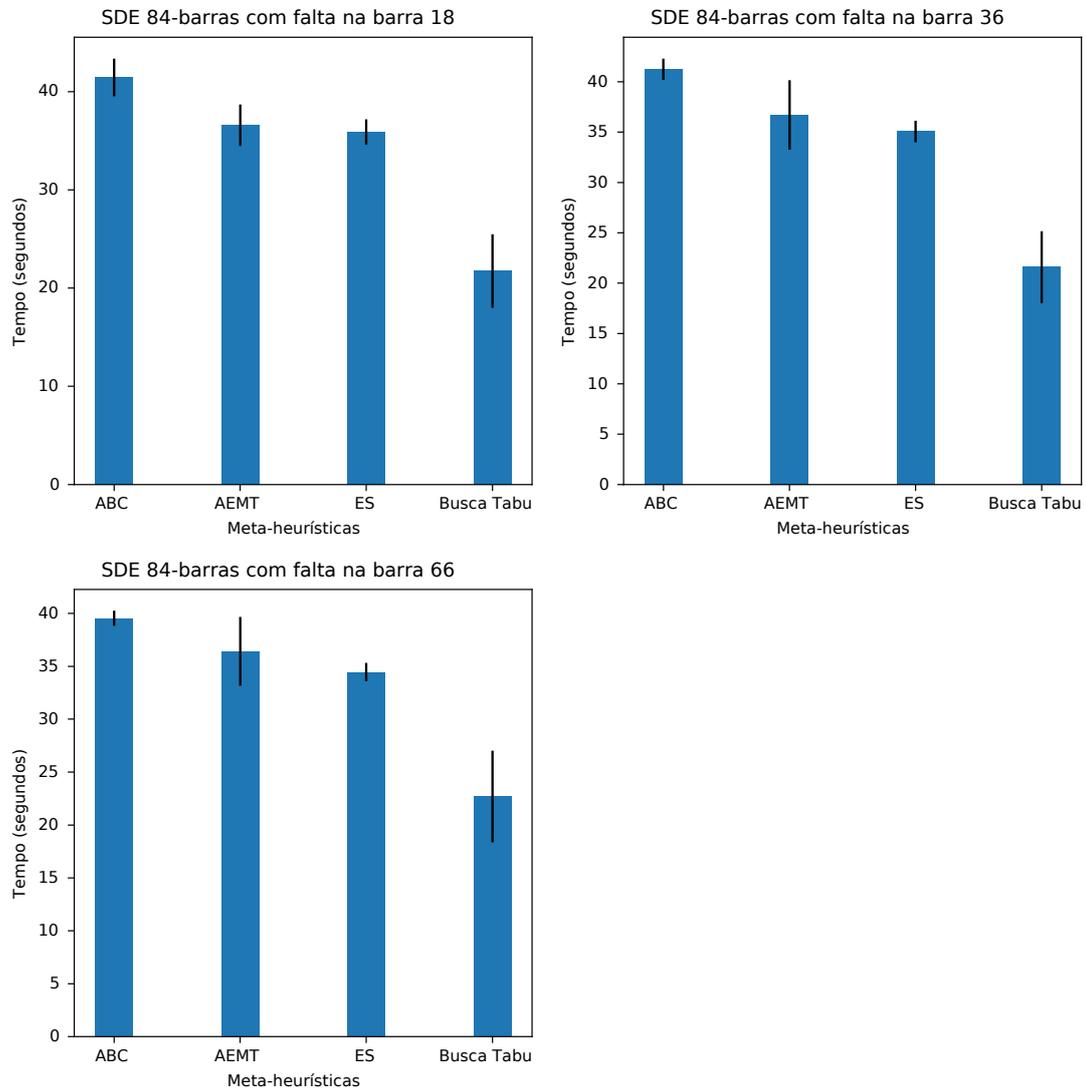
Função de agregação	Número de manoras	Carr. dos trechos (%)	Carr. das subestações (%)	Menor tensão
3	3	36,71	55,14	0,928

5.2.1.4 Tempo Computacional para o Sistema 84-barras

Com relação ao tempo de execução dos algoritmos, para o Sistema 84-barras, a Busca Tabu foi o algoritmo que apresentou o menor tempo de execução, seguido do ES e AEMT, e por último o ABC, como mostra a Figura 29.

Apesar de apresentar um tempo médio bem abaixo dos demais algoritmos, a Busca Tabu apresentou um elevado desvio padrão (linha preta na Figura 29), assim como o AEMT. Neste quesito o ES foi o algoritmo com o melhor tempo de execução, por apresentar valores mais constantes em relação aos outros algoritmos.

Figura 29. Tempo de execução de cada algoritmo para o Sistema 84-barras



Fonte: Produzido pelo autor

5.2.2 Falta Única no Sistema 119-barras

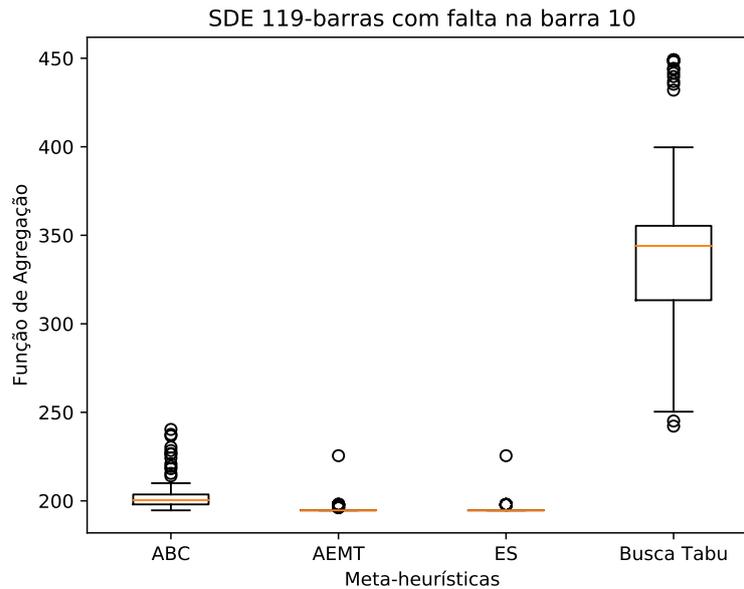
Os resultados dos testes no sistema 119-barras não levam em consideração o carregamento da subestação, por só apresentar uma.

5.2.2.1 Falta na barra 10

A Figura 30 apresenta os resultados da Função de Agregação para a falta na barra 10. Neste caso nenhum algoritmo conseguiu encontrar uma solução que cumprisse todas as restrições. Os algoritmos mais efetivos foram o AEMT e o ES que apresentaram resultados bem parecidos, aceitando a hipótese nula, como mostra a Tabela 9. A distância da Busca

Tabu para os outros algoritmos aumentou, provavelmente devido a este caso apresentar um alto grau de complexidade e a Busca Tabu, por ser um algoritmo de busca local, não conseguir lidar com esse aumento.

Figura 30. Resultados da falta na barra 10



Fonte: Produzido pelo autor

Tabela 9. Teste de Dunn para a falta na barra 10

	AEMT	ES	TS
ABC	Rejeita	Rejeita	Rejeita
AEMT	-	Aceita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

A Tabela 10 apresenta uma das soluções encontradas pelo ES, nesta solução existem trechos sobrecarregados, o que torna esta solução impraticável.

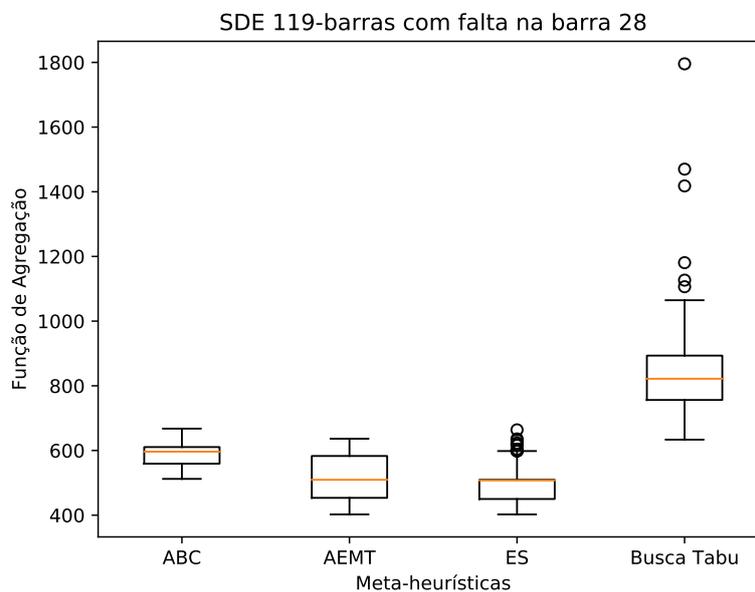
Tabela 10. Resultado do ES falta na barra 10

Função de agregação	Número de manoras	Carr. dos trechos (%)	Menor tensão
194,63	9	185,63	0,869

5.2.2.2 Falta na barra 28

A Figura 31 apresenta os resultados da Função de Agregação para a falta na barra 28. Os algoritmos continuaram não convergindo para uma boa solução. O AEMT e o ES apresentaram os melhores resultados, sem apresentarem uma diferença significativa entre eles, como mostra a Tabela 11. A Busca Tabu continua apresentando resultados bem insatisfatórios e o ABC apresentou resultados próximos aos do AEMT e ES. Esse tipo de situação começa a ser uma tendência nas simulações, mostrando uma provável ineficiência da Busca Tabu para esse tipo de problema.

Figura 31. Resultados da falta na barra 28



Fonte: Produzido pelo autor

Tabela 11. Teste de Dunn para a falta na barra 28

	AEMT	ES	TS
ABC	Rejeita	Rejeita	Rejeita
AEMT	-	Aceita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

A Tabela 12 apresenta uma das soluções encontradas pelo ES, esta solução apresentou trechos super carregados com 25 manobras, sendo totalmente impraticável.

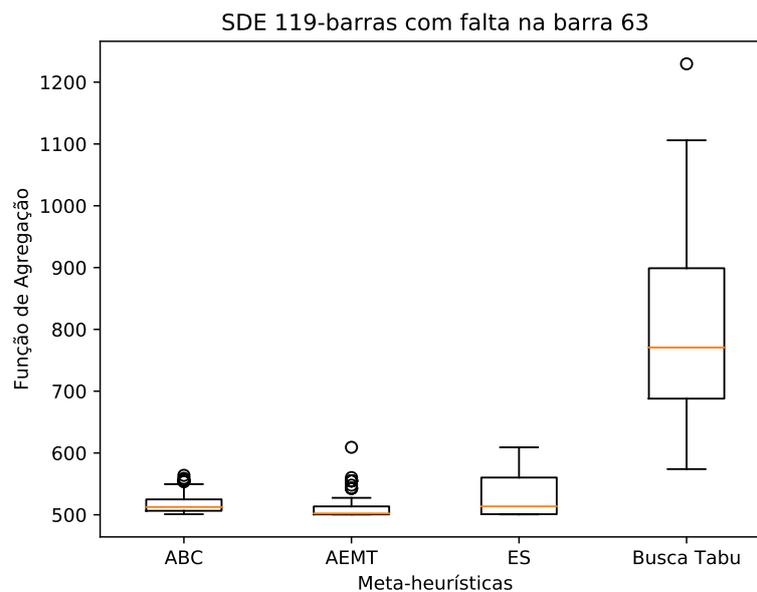
Tabela 12. Resultado do ES falta na barra 28

Função de agregação	Número de manoras	Carr. dos trechos (%)	Menor tensão
404,40	25	379,40	0,869

5.2.2.3 Falta na barra 63

A Figura 32 apresenta os resultados da Função de Agregação para a falta na barra 63. Neste caso nenhum algoritmo conseguiu gerar alguma solução que cumprisse as restrições, contudo apenas o AEMT apresentou os melhores resultados, seguido do ABC e do ES que apresentaram resultados semelhantes entre si, como mostra a Tabela 13. A Busca Tabu continuou apresentando os piores resultados.

Figura 32. Resultados da falta na barra 63



Fonte: Produzido pelo autor

Tabela 13. Teste de Dunn para a falta na barra 63

	AEMT	ES	TS
ABC	Rejeita	Aceita	Rejeita
AEMT	-	Rejeita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

A Tabela 14 apresenta uma das soluções encontradas pelo AEMT, esta solução apresentou trechos super carregados com 13 manobras, além de tensões abaixo do limite

inferior, tornando esta solução impraticável.

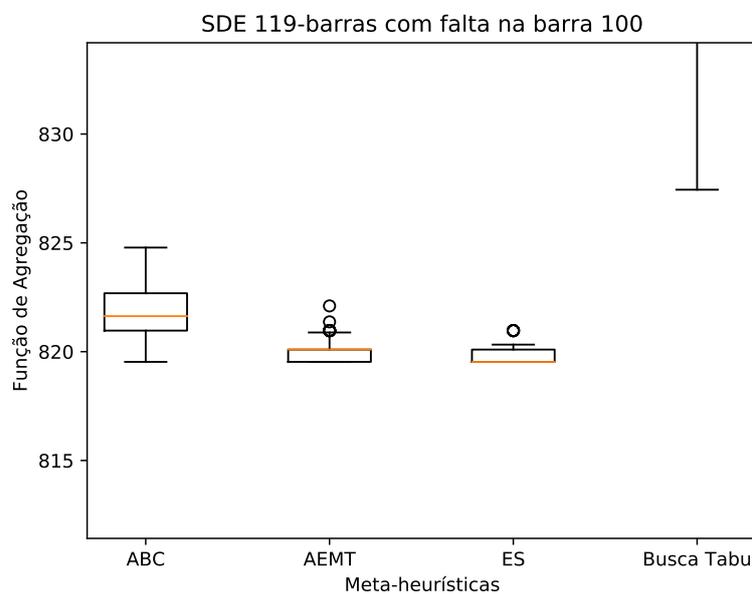
Tabela 14. Resultado do AEMT falta na barra 63

Função de agregação	Número de manoras	Carr. dos trechos (%)	Menor tensão
513,56	13	374,95	0,8351

5.2.2.4 Falta na barra 100

A Figura 33 apresenta os resultados da Função de Agregação para a falta na barra 100. Devido a discrepância entre os resultados da Busca Tabu e dos outros algoritmos, a Figura 33 foi ampliada para facilitar a visualização dos dados. Nota-se que ocorreu o mesmo padrão visto em outras simulações: O AEMT e ES apresentando os melhores resultados e sem uma diferença significativa entre eles, como mostra a Tabela 15, o ABC consegue bons resultados, mas não tão bons quanto o AEMT e o ES, e a Busca Tabu apresentando os piores resultados. Neste caso também nenhum algoritmo conseguiu convergir para boas soluções.

Figura 33. Resultados da falta na barra 100



Fonte: Produzido pelo autor

Tabela 15. Teste de Dunn para a falta na barra 100

	AEMT	ES	TS
ABC	Rejeita	Rejeita	Rejeita
AEMT	-	Aceita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

5.2.2.5 Tempo Computacional para o Sistema 119-barras

Com relação ao tempo de execução dos algoritmos, para o Sistema 119-barras, a Busca Tabu e o ES foram os algoritmos que apresentaram os menores tempos computacionais, como mostra a Figura 34.

Assim como nas simulações executadas no Sistema 84-barras, ocorreu uma alta variação nos resultados da Busca Tabu, mostrando que para este Sistema o ES é o algoritmo mais indicado com relação ao tempo de execução por ser mais constante que a Busca Tabu.

A Tabela 16 apresenta uma das soluções encontradas pelo AEMT, esta solução apresentou trechos super carregados com 11 manobras, além de tensões bem abaixo do limite inferior, tornando esta solução impraticável.

Tabela 16. Resultado do AEMT falta na barra 100

Função de agregação	Número de manoras	Carr. dos trechos (%)	Menor tensão
820,97	11	656,80	0,799

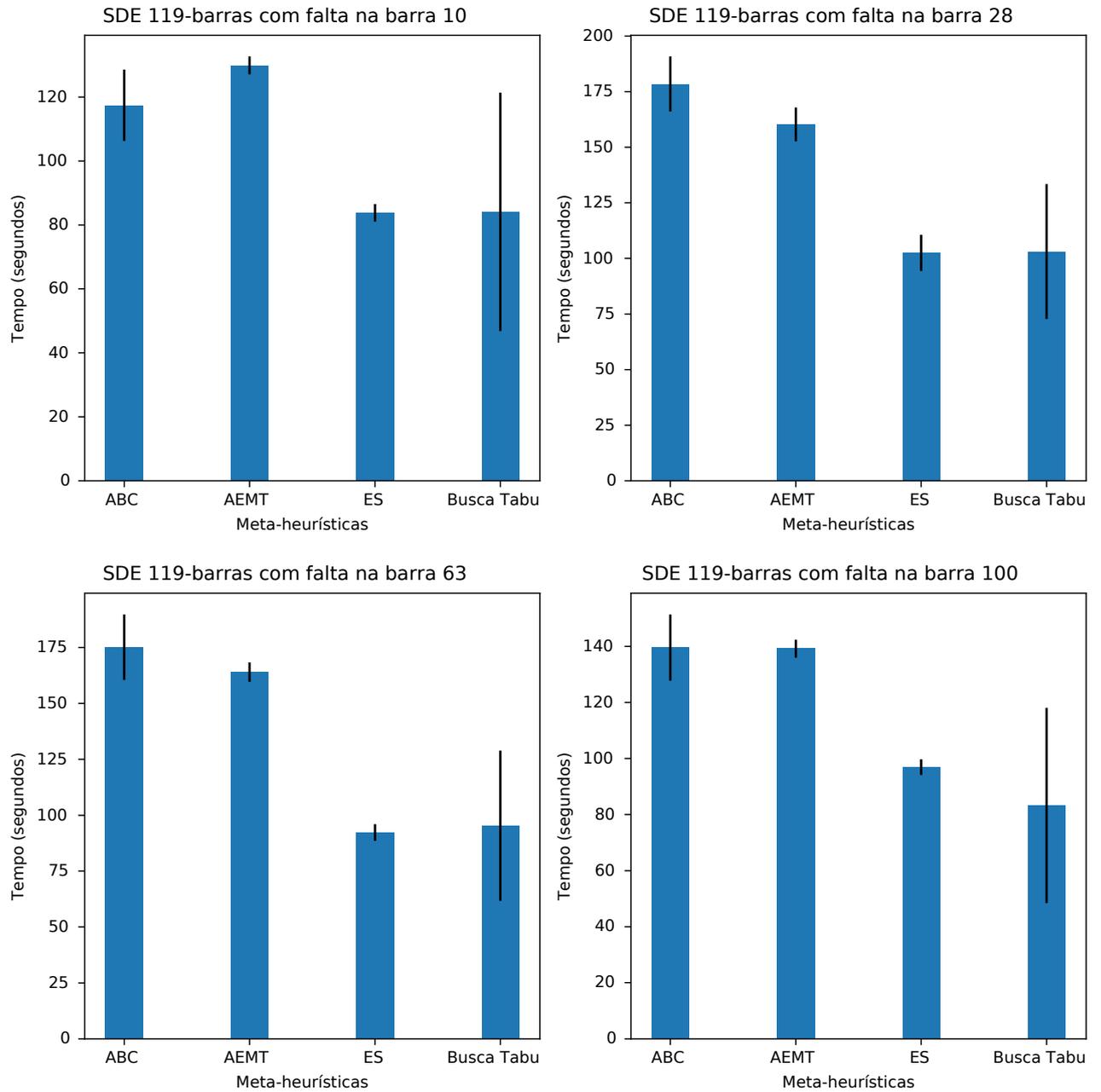
5.2.3 Falta Única no Sistema 135-barras

Assim como ocorreu no caso da falta da barra 100 do Sistema 119-barras, os dados da Busca Tabu apresentaram uma grande variação, dificultando a visualização dos dados no *bolxplot*. Por esse motivo, todas as Figuras foram ampliadas.

5.2.3.1 Falta na barra 21

A Figura 35 apresenta os resultados da Função de Agregação para a falta na barra 21. Os algoritmos AEMT e ES conseguiram convergir para o ótimo global. O ABC e a Busca Tabu apresentaram resultados semelhantes, como mostra a Tabela 17, não havendo diferença significativa entre eles. Neste caso todos os algoritmos conseguiram gerar boas soluções, exceto a Busca Tabu que em algumas execuções não conseguiu gerar soluções que cumprisse as restrições do sistema.

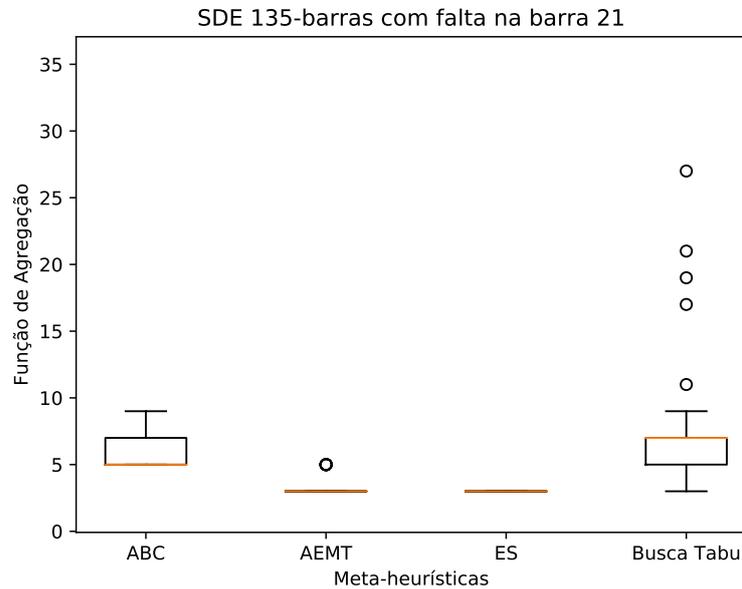
Figura 34. Tempo de execução de cada algoritmo para o Sistema 119-barras



Fonte: Produzido pelo autor

A Tabela 18 apresenta os resultados da melhor solução encontrada pelo ES, nesta solução pode-se observar o cumprimento de todas as restrições do sistema com pouquíssimas manobras necessárias para realizar esta tarefa.

Figura 35. Resultados da falta na barra 21



Fonte: Produzido pelo autor

Tabela 17. Teste de Dunn para a falta na barra 21

	AEMT	ES	TS
ABC	Rejeita	Rejeita	Aceita
AEMT	-	Aceita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

Tabela 18. Resultado do ES falta na barra 21

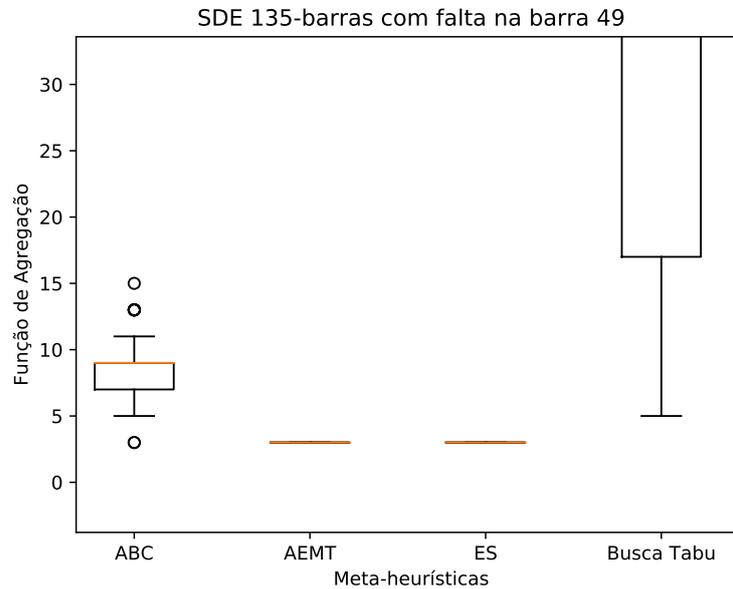
Função de agregação	Número de manoras	Carr. dos trechos (%)	Carr. das subestações (%)	Menor tensão
3	3	80,14	67,19	0,931

5.2.3.2 Falta na barra 49

A Figura 36 apresenta os resultados da Função de Agregação para a falta na barra 49. Neste caso a tendência volta a se repetir, o AEMT e o ES convergiram em todas as execuções (neste caso para um ótimo global), seguidos do ABC que, mesmo não convergindo conseguiu apresentar boas soluções e por último a Busca Tabu.

A Tabela 20 apresenta os resultados da melhor solução encontrada pelo AEMT, nesta solução pode-se observar o cumprimento de todas as restrições do sistema com pouquíssimas manobras necessárias para realizar esta tarefa.

Figura 36. Resultados da falta na barra 49



Fonte: Produzido pelo autor

Tabela 19. Teste de Dunn para a falta na barra 49

	AEMT	ES	TS
ABC	Rejeita	Rejeita	Rejeita
AEMT	-	Aceita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

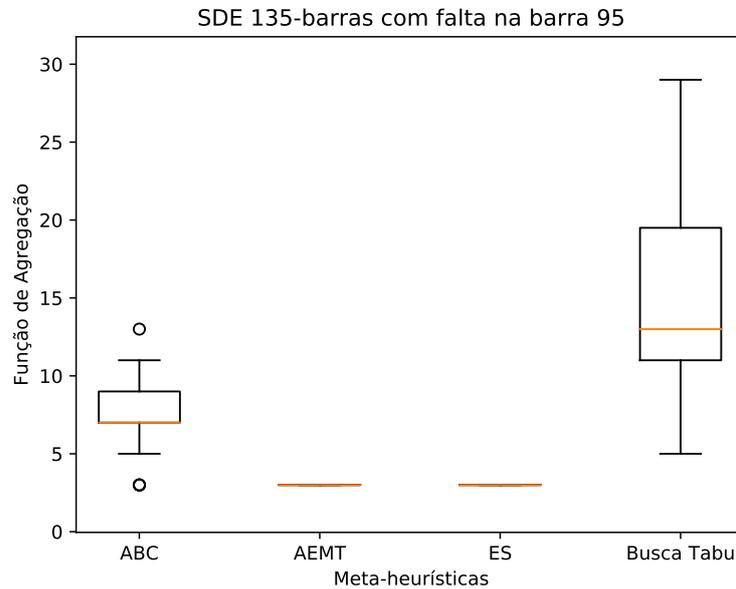
Tabela 20. Resultado do AEMT falta na barra 49

Função de agregação	Número de manoras	Carr. dos trechos (%)	Carr. das subestações (%)	Menor tensão
3	3	73,44	68,92	0,931

5.2.3.3 Falta na barra 95

A Figura 37 apresenta os resultados da Função de Agregação para a falta na barra 95. Resultados parecidos com os que já foram apresentados. AEMT e ES convergiram para o ótimo global, ABC apresenta bons resultados e a Busca Tabu não consegue apresentar resultados satisfatórios nas execuções.

Figura 37. Resultados da falta na barra 95



Fonte: Produzido pelo autor

Tabela 21. Teste de Dunn para a falta na barra 95

	AEMT	ES	TS
ABC	Rejeita	Rejeita	Rejeita
AEMT	-	Aceita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

A Tabela 22 apresenta os resultados da melhor solução encontrada pelo ES, nesta solução pode-se observar o cumprimento de todas as restrições do sistema com pouquíssimas manobras necessárias para realizar esta tarefa.

Tabela 22. Resultado do ES falta na barra 95

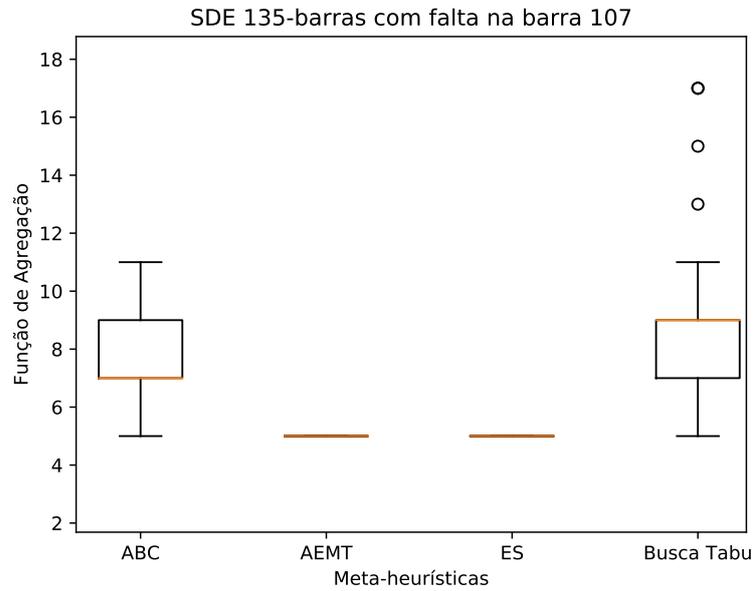
Função de agregação	Número de manoras	Carr. dos trechos (%)	Carr. das subestações (%)	Menor tensão
3	3	72,39	67,19	0,931

5.2.3.4 Falta na barra 107

A Figura 38 apresenta os resultados da Função de Agregação para a falta na barra 107. Neste caso, mesmo que siga a tendência apresentada nos resultado anteriores, ao analisar a Figura 38, o ABC teria um comportamento parecido com a Busca Tabu, contudo

o teste de Dunn, apresentado na Tabela 23, impede esta conclusão, logo, o ABC apresentou resultados melhores que a Busca Tabu.

Figura 38. Resultados da falta na barra 107



Fonte: Produzido pelo autor

Tabela 23. Teste de Dunn para a falta na barra 107

	AEMT	ES	TS
ABC	Rejeita	Rejeita	Rejeita
AEMT	-	Aceita	Rejeita
ES	-	-	Rejeita

Fonte: Produzido pelo autor

A Tabela 24 apresenta os resultados da melhor solução encontrada pelo AEMT, nesta solução pode-se observar o cumprimento de todas as restrições do sistema com pouquíssimas manobras necessárias para realizar esta tarefa.

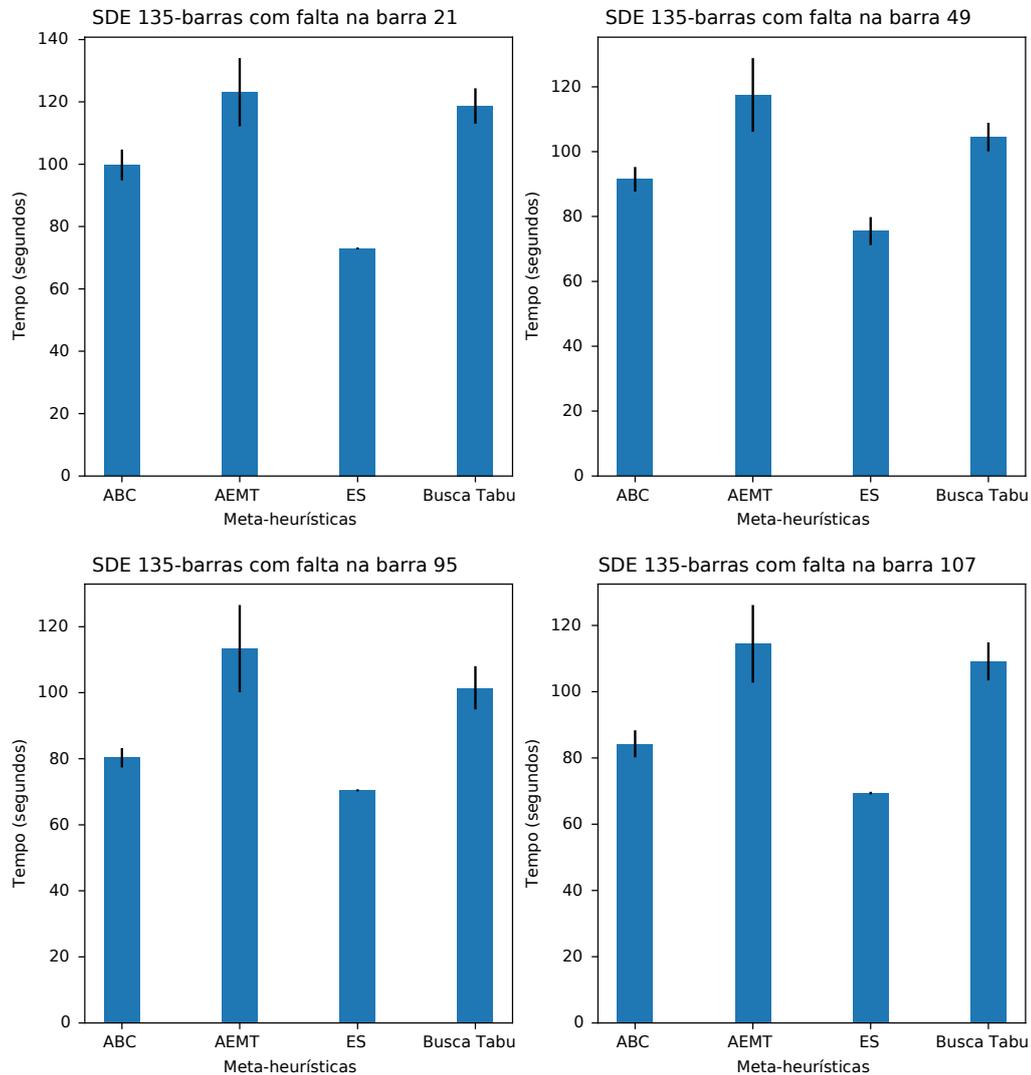
Tabela 24. Resultado do AEMT falta na barra 107

Função de agregação	Número de manoras	Carr. dos trechos (%)	Carr. das subestações (%)	Menor tensão
5	5	73,44	77,70	0,936

5.2.3.5 Tempo Computacional para o Sistema 135-barras

Com relação ao tempo de execução dos algoritmos, para o Sistema 135-barras, o ES foi o algoritmo com o menor tempo computacional, seguido do ABC, Busca Tabu e AEMT, como mostra a Figura 39, consolidando o ES como o algoritmo mais rápido e estável dos quatro, pois houve poucas variações no seu desvio padrão, com relação aos outros três.

Figura 39. Tempo de execução de cada algoritmo para o Sistema 135-barras



Fonte: Produzido pelo autor

5.3 Análise de Desempenho

Nos testes apresentados, o AEMT foi o que apresentou os melhores resultados. Nos Sistemas 84-barras e 135-barras este algoritmo conseguiu convergir para o ótimo global sem muita variação. Mesmo sem encontrar uma solução ótima para o Sistema 119-barras, ele se mostrou um algoritmo bem robusto em todos os casos, por apresentar resultados consistentes em todos os testes.

Apesar de ser robusto e eficiente, o AEMT apresentou um elevado tempo de execução nas simulações. No Sistema 84-barras ele foi o segundo algoritmo mais demorado, atrás apenas do ABC e com resultados bem próximos do ES. No Sistema 119-barras,

o AEMT continua sendo o segundo mais demorado, contudo a diferença entre ele e o ES aumentou bastante e na falta na barra 10 ele foi o algoritmo com maior tempo computacional, em média 2 minutos. No Sistema 135-barras essa diferença aumenta ainda mais, ele se torna o algoritmo mais demorado em todos os casos.

Em contraste com o AEMT, a Busca Tabu foi o algoritmo com os piores resultados apresentados. No Sistema 84-barras, apesar de encontrar algumas soluções ótimas, a maior parte dos resultados produzidos por ela não foram bons. Isto se repete em todos os sistemas testados. Além de não ser eficiente, a Busca Tabu também não foi robusta por apresentar uma variação muito grande nos resultados produzidos.

Apesar de não ser eficiente, a Busca Tabu foi o algoritmo mais rápido no Sistema 84-barras. No Sistema 119-barras, apesar de ter um tempo médio pequeno, ela apresentou uma grande variação nos resultados se comparada com a ES. No Sistema 135-barras ocorreu algo interessante, ela apresentou resultados tão demorados quanto o AEMT, contudo os resultados finais não foram tão bons quanto dele. Isto mostra uma falta de robustez presente neste algoritmo tanto nos resultados finais quanto no tempo de execução. Isto mostra que este algoritmo não é indicado para este tipo de problema.

Com relação ao ABC, ele apresentou resultados satisfatórios. Não eram os melhores resultados, porém eram bons resultados e que resolvem o problema cumprindo as restrições, como nos Sistemas 84-barras e 135-barras. Mesmo apresentando bons resultados, nas simulações dos Sistemas 84-barras e 119-barras, ele apresentou o maior tempo computacional. O pior caso foi a falta na barra 28 do Sistema 119-barras, onde o algoritmo levou em média cerca de 3 minutos para encontrar as soluções. No Sistema 135-barras o tempo de execução diminuiu se comparado com outros algoritmos.

Por fim o ES foi o algoritmo que mais bem se adaptou ao problema de restauração. Dos 11 testes, 9 deles ele conseguiu resultados tão bons quanto o AEMT, que já é um algoritmo consolidado na literatura. Além de ser um algoritmo robustos, sem muitas variações nos resultados encontrados, o ES se apresentou um algoritmo bem rápido e consistente nos tempos computacionais. Nos Sistemas 119-barras e 135-barras ele foi o algoritmo mais rápido e no Sistema 84-barras ele foi o segundo. Esses motivos tornam ele o algoritmo com o melhor *custo x benefício*, por ter resultados tão bons quanto o AEMT, sendo mais rápido que ele.

Nos testes do Sistema 119-barras, nenhum algoritmo conseguiu encontrar uma solução que cumprisse com todas as restrições. Acredita-se não ser uma falha dos algoritmos mas sim não existir uma solução que cumpra tais restrições. Uma possível solução seria a utilização do operador CUT, desenvolvido por (MATHIAS-NETO, 2016), para a rejeição de carga, o que elevaria o custo computacional pois o problema passa a ser multi-objetivo, minimizando o número de manobras e maximizando o número de consumidores desligados, sendo necessário outros algoritmos para a resolução deste problema.

6 Conclusões e Trabalhos Futuros

Neste trabalho foi apresentado uma análise de desempenho de quatro meta-heurísticas para o problema de restauração de redes de distribuição de energia elétrica, utilizando a RNP como estrutura de dados. Dos algoritmos testados, o AEMT foi que apresentou os melhores e os mais consistentes resultados, já o ES apresentou resultados tão bons quanto o do AEMT, porém com um tempo de execução bem inferior. Isso mostra que é possível utilizar outras meta-heurísticas de forma eficiente para o problema de restauração, tendo em vista que o ES apresentou resultados satisfatórios.

Mesmo sendo possível utilizar outras meta-heurísticas de forma eficiência para o problema de restauração de SD, este trabalho mostrou que é possível obter resultados razoáveis com algoritmos baseados na inteligência de enxames, como o ABC e também mostrou que é impraticável utilizar algoritmos de busca local, como a Busca Tabu, devido a alta complexidade do problema, sendo necessário meta-heurísticas mais robustas.

Os testes foram realizados em 3 sistemas reais, que mostram a eficiência dos algoritmos AEMT e ES, pois conseguiram restaurar os sistemas 84-barras e 135-barras com robustez.

Também foram apresentados um novo algoritmo de definição dos nós de poda, raiz e adjacente, além de um novo algoritmo para encontrar a Primeira Configuração Factível utilizando a RNP. Estes algoritmos foram elaborados visando a restauração em sistemas de distribuição com poucas subestações, onde a transferência de carga entre elas fica limitada, permitindo a utilização da RNP em sistemas como o 119-barras, que apresenta uma única subestação, e no sistema 135-barras que apresenta duas subestações e a possibilidade de transferência de carga dentro de uma mesma subestação, o que antes era impossível utilizando a RNP.

A seguir, são listados alguns trabalhos futuros:

- Realizar mais testes em outros sistemas e avaliando faltas múltiplas;
- Realizar modificações no ES com a finalidade de melhorar ainda mais o seu desempenho;
- Utilizar o operador CUT e avaliar algoritmos multi-objetivos, analisando seu desempenho em sistemas que apresentem uma única árvore em suas florestas;
- Avaliar os algoritmos, porém com a inclusão da geração distribuída;
- Avaliar os algoritmos levando em consideração os dispositivos de proteção dos sistemas de distribuição.

Referências

- ADIBI, M.; MILANICZ, D. Estimating restoration duration. *IEEE Transactions on Power Systems*, IEEE, v. 14, n. 4, p. 1493–1498, 1999. Citado na página 12.
- AUGUGLIARO, A.; DUSONCHET, L.; SANSEVERINO, E. R. Multiobjective service restoration in distribution networks using an evolutionary approach and fuzzy sets. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 22, n. 2, p. 103–110, 2000. Citado 2 vezes nas páginas 14 e 24.
- BACK, T.; HOFFMEISTER, F.; SCHWEFEL, H.-P. A survey of evolution strategies. In: MORGAN KAUFMANN PUBLISHERS SAN MATEO, CA. *Proceedings of the fourth international conference on genetic algorithms*. [S.l.], 1991. v. 2, n. 9. Citado 2 vezes nas páginas 13 e 43.
- BARAN, M.; WU, F. F. Optimal sizing of capacitors placed on a radial distribution system. *IEEE Transactions on power Delivery*, IEEE, v. 4, n. 1, p. 735–743, 1989. Citado na página 35.
- BEYER, H.-G.; SCHWEFEL, H.-P. Evolution strategies—a comprehensive introduction. *Natural computing*, Springer, v. 1, n. 1, p. 3–52, 2002. Citado na página 43.
- CESPEDES, R. New method for the analysis of distribution networks. *IEEE Transactions on Power Delivery*, IEEE, v. 5, n. 1, p. 391–396, 1990. Citado na página 35.
- DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, IEEE, v. 6, n. 2, p. 182–197, 2002. Citado na página 14.
- DELBEM, A. C.; CARVALHO, A. de; POLICASTRO, C. A.; PINTO, A. K.; HONDA, K.; GARCIA, A. C. Node-depth encoding for evolutionary algorithms applied to network design. In: SPRINGER. *Genetic and Evolutionary Computation Conference*. [S.l.], 2004. p. 678–687. Citado 6 vezes nas páginas 12, 14, 18, 25, 32 e 48.
- FEOFILOFF, P.; KOHAYAKAWA, Y.; WAKABAYASHI, Y. Uma introdução sucinta à teoria dos grafos. 2011. Citado na página 22.
- GENDREAU, M.; POTVIN, J.-Y. et al. *Handbook of metaheuristics*. [S.l.]: Springer, 2010. v. 2. Citado na página 38.
- GLOVER, F. Tabu search—part i. *ORSA Journal on computing*, INFORMS, v. 1, n. 3, p. 190–206, 1989. Citado 2 vezes nas páginas 13 e 41.
- HONG, H.; HU, Z.; GUO, R.; MA, J.; TIAN, J. Directed graph-based distribution network reconfiguration for operation mode adjustment and service restoration considering distributed generation. *Journal of Modern Power Systems and Clean Energy*, SGEPRI, v. 5, n. 1, p. 142–149, 2017. Citado na página 33.
- HSIAO, Y.-T.; CHIEN, C.-Y. Enhancement of restoration service in distribution systems using a combination fuzzy-ga method. *IEEE Transactions on Power Systems*, IEEE, v. 15, n. 4, p. 1394–1400, 2000. Citado 2 vezes nas páginas 13 e 24.

- INAGAKI, J.; NAKAJIMA, J.; HASEYAMA, M. A multiobjective service restoration method for power distribution systems. In: IEEE. *2006 IEEE International Symposium on Circuits and Systems*. [S.l.], 2006. p. 4–pp. Citado na página 14.
- KARABOGA, D. *An idea based on honey bee swarm for numerical optimization*. [S.l.], 2005. Citado 2 vezes nas páginas 13 e 42.
- LUKE, S. *Essentials of metaheuristics*. [S.l.]: Lulu Raleigh, 2013. v. 2. Citado na página 38.
- MANSOUR, M. R. *Algoritmo para obtenção de planos de restabelecimento para sistemas de distribuição de grande porte*. Tese (Doutorado) — Universidade de São Paulo, 2009. Citado 4 vezes nas páginas 14, 17, 35 e 36.
- MANTOVANI, J. R.; CASARI, F.; ROMERO, R. A. Reconfiguração de sistemas de distribuição radiais utilizando o critério de queda de tensão. *Controle and Automacao*, p. 150–159, 2000. Citado na página 48.
- MARQUES, L. T. *Restabelecimento de energia por reconfiguração de redes em sistemas de distribuição de grande porte com priorização de chaves, consumidores e definição de sequência de chaveamento*. Tese (Doutorado) — Universidade de São Paulo, 2013. Citado na página 18.
- MARQUES, L. T. *Restabelecimento de energia em sistemas de distribuição considerando aspectos práticos*. Tese (Doutorado) — Universidade de São Paulo, 2018. Citado na página 15.
- MATHIAS-NETO, W. P. Proposta de um algoritmo eficiente baseado em busca tabu e representação nó-profundidade para a restauração de redes de distribuição de energia elétrica. Universidade Estadual Paulista (UNESP), 2016. Citado 2 vezes nas páginas 15 e 68.
- SANCHES, D.; MAZUCATO, S. J.; CASTOLDI, M.; DELBEM, A.; LONDON, J. Algoritmo evolutivo multi-objetivo baseado em tabelas de subpopulações de soluções não dominadas e força de pareto para tratar do problema de restabelecimento de energia em sistemas de distribuição. p. 1–6, 2013. Citado na página 14.
- SANCHES, D. S. *Algoritmos evolutivos multi-objetivo para reconfiguração de redes em sistemas de distribuição de energia elétrica*. Tese (Doutorado) — Universidade de São Paulo, 2013. Citado 4 vezes nas páginas 20, 27, 28 e 35.
- SANTOS, A. C. d. *Algoritmo evolutivo computacionalmente eficiente para reconfiguração de sistemas de distribuição*. Tese (Doutorado) — Universidade de São Paulo, 2009. Citado 13 vezes nas páginas 13, 17, 18, 27, 28, 29, 30, 31, 32, 33, 35, 37 e 44.
- SANTOS, A. C. dos; DELBEM, A. C.; BRETAS, N. G. A multiobjective evolutionary algorithm with node-depth encoding for energy restoration. In: IEEE. *2008 Fourth International Conference on Natural Computation*. [S.l.], 2008. v. 6, p. 417–422. Citado na página 14.
- SHEHU, G. Distribution network reconfiguration for loss reduction and voltage profile improvement using b-pso. In: . [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 24 e 26.

- SHIN, D.-J.; KIM, J.-O.; KIM, T.-K.; CHOO, J.-B.; SINGH, C. Optimal service restoration and reconfiguration of network using genetic-tabu algorithm. *Electric Power Systems Research*, Elsevier, v. 71, n. 2, p. 145–152, 2004. Citado na página 14.
- SHIRMOHAMMADI, D.; HONG, H. W.; SEMLYEN, A.; LUO, G. A compensation-based power flow method for weakly meshed distribution and transmission networks. *IEEE Transactions on power systems*, IEEE, v. 3, n. 2, p. 753–762, 1988. Citado na página 35.
- SHORT, T. A. *Electric power distribution handbook*. [S.l.]: CRC press, 2018. Citado na página 16.
- SHUKRAN, M. A. M.; CHUNG, Y. Y.; YEH, W.-C.; WAHID, N.; ZAIDI, A. M. A. Artificial bee colony based data mining algorithms for classification tasks. *Modern Applied Science*, v. 5, n. 4, p. 217, 2011. Citado na página 42.
- SOARES, T. W. d. L. *Estruturas de dados eficientes para algoritmos evolutivos aplicados a projeto de redes*. Tese (Doutorado) — Universidade de São Paulo, 2009. Citado 3 vezes nas páginas 21, 22 e 23.
- STORN, R.; PRICE, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, Springer, v. 11, n. 4, p. 341–359, 1997. Citado na página 15.
- SU, C.-T.; LEE, C.-S. Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution. *IEEE Transactions on power delivery*, IEEE, v. 18, n. 3, p. 1022–1027, 2003. Citado na página 47.
- TOUNE, S.; FUDO, H.; GENJI, T.; FUKUYAMA, Y.; NAKANISHI, Y. Comparative study of modern heuristic algorithms to service restoration in distribution systems. *IEEE Transactions on Power Delivery*, IEEE, v. 17, n. 1, p. 173–181, 2002. Citado na página 14.
- ZHANG, D.; FU, Z.; ZHANG, L. An improved ts algorithm for loss-minimum reconfiguration in large-scale distribution systems. *Electric Power Systems Research*, Elsevier, v. 77, n. 5-6, p. 685–694, 2007. Citado na página 47.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische . . . , v. 103, 2001. Citado na página 14.

APÊNDICE A . Resultados numéricos das análises estatísticas

Neste apêndice serão apresentadas as médias e desvios-padrões e os resultados dos testes estatísticos apresentados no Capítulo 5.

Tabela 25. Médias e desvios-padrões dos testes executados

SDE	Barra de falta	ABC		AEMT		ES		TS	
		Média	Desvio Padrão						
84	18	6,54	1,44	5,00	0,00	63,53	58,53	129,51	38,15
84	36	3,52	0,88	3,00	0,00	3,00	0,00	6,00	1,08
84	66	3,66	0,98	3,00	0,00	3,00	0,00	88,19	43,77
119	10	203,38	9,71	195,42	3,22	195,46	3,25	343,36	48,41
119	28	586,53	34,03	516,52	63,54	495,96	64,75	848,07	170,04
119	63	518,38	15,29	512,26	19,80	539,29	42,22	787,15	126,09
119	100	821,82	1,21	819,95	0,51	819,82	0,40	883,62	99,41
135	21	5,66	0,98	3,12	0,47	3,00	0,00	12,28	25,34
135	49	8,56	2,18	3,00	0,00	3,00	0,00	90,84	55,47
135	95	7,60	1,75	3,00	0,00	3,00	0,00	29,55	38,36
135	107	7,18	1,50	5,00	0,00	5,00	0,00	21,73	36,82

Tabela 26. Testes estatísticos

SDE	Barra de falta	Testes Estatísticos	
		KS-test	KW-test
84	18	0,00	$1,07^{-57}$
84	36	0,00	$1,33^{-67}$
84	66	0,00	$1,67^{-65}$
119	10	0,00	$3,14^{-72}$
119	28	0,00	$3,48^{-61}$
119	63	0,00	$4,20^{-52}$
119	100	0,00	$5,13^{-70}$
135	21	0,00	$4,32^{-73}$
135	49	0,00	$7,63^{-80}$
135	95	0,00	$1,42^{-78}$
135	107	0,00	$3,81^{-65}$

Tabela 27. Teste de Dunn para o SDE 84-barras

Falta na barra 18				Falta na barra 36			
	AEMT	ES	Tabu		AEMT	ES	Tabu
ABC	$3,35^{-07}$	$2,33^{-01}$	$1,38^{-27}$	ABC	$6,81^{-04}$	$6,81^{-04}$	$2,81^{-32}$
AEMT	-	$3,09^{-10}$	$1,61^{-57}$	AEMT	-	1,00	$2,43^{-52}$
ES	-	-	$3,27^{-22}$	ES	-	-	$2,43^{-52}$

Falta na barra 66			
	AEMT	ES	Tabu
ABC	$2,15^{-04}$	$2,15^{-04}$	$9,94^{-30}$
AEMT	-	1,00	$5,00^{-51}$
ES	-	-	$5,00^{-51}$

Tabela 28. Teste de Dunn para o SDE 119-barras

Falta na barra 10				Falta na barra 28			
	AEMT	ES	Tabu		AEMT	ES	Tabu
ABC	$1,14^{-16}$	$1,29^{-16}$	$5,53^{-13}$	ABC	$1,42^{-08}$	$1,80^{-13}$	$3,30^{-15}$
AEMT	-	$9,89^{-01}$	$3,42^{-54}$	AEMT	-	$9,08^{-02}$	$7,87^{-42}$
ES	-	-	$4,28^{-54}$	ES	-	-	$1,87^{-52}$

Falta na barra 63				Falta na barra 100			
	AEMT	ES	Tabu		AEMT	ES	Tabu
ABC	$2,39^{-04}$	$8,25^{-01}$	$2,80^{-28}$	ABC	$3,00^{-15}$	$8,57^{-19}$	$1,02^{-11}$
AEMT	-	$9,81^{-05}$	$6,23^{-49}$	AEMT	-	$3,36^{-01}$	$7,04^{-49}$
ES	-	-	$3,19^{-27}$	ES	-	-	$3,04^{-55}$

Tabela 29. Teste de Dunn para o SDE 135-barras

Falta na barra 21				Falta na barra 49			
	AEMT	ES	Tabu		AEMT	ES	Tabu
ABC	$6,10^{-33}$	$2,52^{-36}$	$1,64^{-01}$	ABC	$3,42^{-24}$	$3,42^{-24}$	$1,30^{-08}$
AEMT	-	$5,28^{-01}$	$1,21^{-40}$	AEMT	-	1,00	$1,84^{-56}$
ES	-	-	$2,10^{-44}$	ES	-	-	$1,84^{-56}$

Falta na barra 95				Falta na barra 107			
	AEMT	ES	Tabu		AEMT	ES	Tabu
ABC	$7,22^{-25}$	$7,22^{-25}$	$1,06^{-07}$	ABC	$1,58^{-22}$	$1,58^{-22}$	$1,84^{-05}$
AEMT	-	1,00	$5,90^{-55}$	AEMT	-	1,00	$7,79^{-45}$
ES	-	-	$5,90^{-55}$	ES	-	-	$7,79^{-45}$