



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Orlando Júnior Nascimento Ferreira

**Algoritmo Genético de Chaves Aleatórias Viciadas e Recozimento Simulado
aplicados ao Problema da Reconfiguração de Redes de Distribuição de Energia
Elétrica considerando Técnicas de Reinicialização e Híbridização**

Florianópolis
2022

Orlando Júnior Nascimento Ferreira

**Algoritmo Genético de Chaves Aleatórias Viciadas e Recozimento Simulado
aplicados ao Problema da Reconfiguração de Redes de Distribuição de Energia
Elétrica considerando Técnicas de Reinicialização e Híbridização**

Dissertação submetida ao Programa de Pós-Graduação
em Engenharia Elétrica da Universidade Federal de
Santa Catarina para a obtenção do título de mestre
em Engenharia Elétrica.

Orientador: Prof. Miguel Moreto, Dr.

Florianópolis

2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Ferreira, Orlando Júnior Nascimento
Algoritmo Genético de Chaves Aleatórias Viciadas e
Recozimento Simulado aplicados ao problema da
reconfiguração de redes de distribuição de energia elétrica
considerando técnicas de reinicialização e hibridização /
Orlando Júnior Nascimento Ferreira ; orientador, Miguel
Moreto, 2022.
124 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico, Programa de Pós-Graduação em
Engenharia Elétrica, Florianópolis, 2022.

Inclui referências.

1. Engenharia Elétrica. 2. Reconfiguração de sistemas de
distribuição de energia elétrica. 3. Algoritmo genético de
chaves aleatórias viciadas. 4. Recozimento simulado. 5.
Estratégias de reinicialização e hibridização. I. Moreto,
Miguel . II. Universidade Federal de Santa Catarina.
Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

Orlando Júnior Nascimento Ferreira

Algoritmo Genético de Chaves Aleatórias Viciadas e Recozimento Simulado aplicados ao Problema da Reconfiguração de Redes de Distribuição de Energia Elétrica considerando Técnicas de Reinicialização e Hibridização

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Mauro Augusto da Rosa, PhD.
Universidade Federal de Santa Catarina

Prof. João Paulo Abreu Vieira, Dr.
Universidade Federal do Pará

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia Elétrica.

Prof. Telles Brunelli Lazzarin, Dr.
Coordenação do Programa de
Pós-Graduação

Prof. Miguel Moreto, Dr.
Orientador

Florianópolis, 2022.

Este trabalho é dedicado aos meus pais que sempre me apoiaram e acreditaram em mim.

AGRADECIMENTOS

Agradeço a Deus por todas as oportunidades, por me ajudar a superar dificuldades, e sempre cuidar de mim e da minha família.

Aos meus pais, Orlando de Jesus e Maria das Graças, por sempre me apoiarem em todas as decisões, por toda a educação e conhecimento, por sempre estarem ao meu lado e por serem os melhores pais que poderia ter.

A todos os professores do Programa de Pós-Graduação em Engenharia Elétrica da UFSC por todo o conhecimento compartilhado, em especial aos professores Mauro Augusto da Rosa, Roberto de Souza Salgado, Kátia Campos de Almeida e Diego Issicaba por toda a paciência e ajuda nas disciplinas e ao professor Miguel Moreto, por me aceitar como orientado, por toda a paciência e auxílio na elaboração desta dissertação.

Agradeço aos meus amigos e colegas da Celesc Distribuição por todo o conhecimento compartilhado e pelos momentos de descontração, em especial ao André Ikeda, André Leonardo, Daniel Bez, Rafael Paludo, Daniel Pelaquim, Daniel Schmitt, Felipe Moreira, Leonardo, Luiz Fernando, Marcos, Miriam, Rogério e Otávio Wilbert.

Também agradeço aos amigos que fiz durante o mestrado Diego Ferreira, Fernando Nast, Gabriel Kahler, Renan Cardoso, Israel Félix, David Santos, Irlana Costa, Fernanda Tonial, Juan Vidal, Eloisa Guerreiro e Glenda Machado por toda a ajuda e aos momentos de descontração.

E, por fim, ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro durante o curso de mestrado.

RESUMO

A reconfiguração de sistemas de distribuição de energia elétrica (RSDEE) é baseada na mudança dos estados (aberto ou fechado) das chaves de laço e seccionadoras alterando a topologia da rede de forma a minimizar as perdas de potência ativa. Desta forma, viabiliza-se a transferência de cargas entre alimentadores, sob as restrições operacionais da rede, tais como, níveis de tensão adequados, permanência da estrutura radial da rede elétrica e o fornecimento de energia para todas as cargas. Conseqüentemente, tal procedimento pode ocasionar uma redução das perdas técnicas e melhoria no perfil de tensão. A redução de perdas via reconfiguração de redes é um problema de natureza combinatória, em que se deve determinar a combinação dos estados das chaves que correspondem a configuração radial da rede com o menor nível de perdas. Nesta dissertação são apresentadas duas metaheurísticas, o Algoritmo Genético de Chaves Aleatórias Viciadas e o Recozimento Simulado, para resolver o problema de RSDEE e, para minimizar o tempo esperado exigido para obter a solução, estratégias de reinicialização foram adicionadas à técnica baseada em algoritmo genético e também um novo método híbrido foi proposto. Para viabilizar a modelagem radial dos sistemas de distribuição e possibilitar uma redução do espaço de busca de soluções, para obter melhores soluções, foram combinados ao algoritmo genético os decodificadores algoritmo de Kruskal e o algoritmo baseado em *links* da co-árvore (árvore complementar da rede). Além disso, em relação às estratégias de *restart* no algoritmo genético o problema foi estudado em três casos distintos: o primeiro sem o uso de *restart*, o segundo com *restart* a cada κ iterações sem melhoria da função objetivo e o terceiro que utiliza um monitoramento da função objetivo dos indivíduos da população para efetuar um *restart* com elitismo. Por fim, no algoritmo híbrido foram utilizados o recozimento simulado e o algoritmo genético de chaves aleatórias viciadas na codificação baseada na co-árvore. Para a análise destas propostas de solução foram utilizados cinco sistemas bem conhecidos na literatura, os sistemas 14, 33, 69, 84 e 136 barras, e a partir dos resultados obtidos, foi possível obter soluções de qualidade em tempo significativamente reduzido quando comparado à outras abordagens.

Palavras-chave: Reconfiguração de sistemas de distribuição de energia elétrica. Algoritmo genético de chaves aleatórias viciadas. Recozimento simulado. Estratégias de reinicialização. Algoritmo híbrido.

ABSTRACT

The reconfiguration of electric power distribution systems (RSDEE) is based on changing the states (open or closed) of tie and sectionalizing switches changing the network topology in order to minimize active power losses. In this way, it is possible to transfer loads between feeders, subject to the operational constraints of the grid, such as adequate voltage levels, maintenance of the grid's radial structure, and the supply of power to all loads. Consequently, such a procedure can lead to a reduction in technical losses and improvement in voltage profile. Loss reduction through grid reconfiguration is a combinatorial problem, in which it is necessary to determine the combination of switch states that correspond to the radial configuration of the grid with the lowest level of losses. In this dissertation, two metaheuristics are presented, the Biased Random-Key Genetic Algorithm and Simulated Annealing, to solve the RSDEE problem and, to minimize the expected time required to obtain the solution, restart strategies have been added to the genetic algorithm-based technique and also a new hybrid method has been proposed. To achieve the radial modeling of distribution systems and to enable a reduction of the solution search space to obtain better solutions, the decoders Kruskal's algorithm and the link-based algorithm of the co-tree (complementary network tree) were combined to the genetic algorithm. Moreover, regarding the restart strategies in the genetic algorithm the problem was studied in three distinct cases: the first without the use of restart, the second with restart every κ iterations without improvement of the objective function and the third that uses a monitoring of the objective function of the individuals of the population to perform a restart with elitism. Finally, in the hybrid algorithm, simulated annealing and the biased random-key genetic algorithm with the decode based on the co-tree were used. For the analysis of these proposed solutions five well-known systems in the literature were used, the systems 14, 33, 69, 84 and 136-bus, and from the results obtained, it was possible to achieve quality solutions in a significantly reduced time when compared to other approaches.

Keywords: Reconfiguration of distribution systems. Biased random-key genetic algorithm. Simulated annealing. Restart strategies. Hybrid algorithm.

LISTA DE FIGURAS

Figura 1 – Perdas Técnicas/Energia injetada 2018.	16
Figura 2 – Topologia inicial e topologia após a reconfiguração do sistema de 14 barras.	23
Figura 3 – Decodificador usado para mapear soluções no hipercubo para soluções no espaço solução onde o <i>fitness</i> é calculado.	40
Figura 4 – Transição da geração k para a geração $k + 1$ no BRKGA.	42
Figura 5 – <i>Crossover uniforme parametrizado</i> no BRKGA2 para a rede 14 barras.	44
Figura 6 – Fluxograma do BRKGA.	45
Figura 7 – Grafo de 14 nós obtido do sistema teste de 14 barras.	46
Figura 8 – Exemplo para ilustrar segundo decodificador utilizando a rede teste 14 nós.	51
Figura 9 – Decodificação do primeiro caminho fechado para a regra 1 na rede 14 barras.	52
Figura 10 – Decodificação do segundo caminho fechado para a regra 1 na rede 14 barras.	52
Figura 11 – Decodificação do terceiro caminho fechado para a regra 1 na rede 14 barras.	52
Figura 12 – Exemplo para cálculo do nível da chave na rede.	60
Figura 13 – Fluxograma do algoritmo de recozimento simulado.	66
Figura 14 – Possíveis locais para incorporar conhecimento ou outros operadores dentro de uma única geração.	76
Figura 15 – Fluxograma do método híbrido BRKGA2/SA proposto.	79
Figura 16 – Topologia ótima da rede 14 barras.	81
Figura 17 – Perfil de tensão na rede 14 barras.	81
Figura 18 – Evolução das perdas na rede 14 barras.	84
Figura 19 – Evolução do desvio de tensão na rede 14 barras.	84
Figura 20 – Topologia ótima da rede 33 barras.	85
Figura 21 – Perfil de tensão na rede 33 barras.	86
Figura 22 – Evolução das perdas na rede 33 barras.	88
Figura 23 – Evolução do desvio de tensão na rede 33 barras.	88
Figura 24 – Topologia ótima da rede 69 barras.	89
Figura 25 – Perfil de tensão na rede 69 barras.	90
Figura 26 – Evolução das perdas na rede 69 barras.	92
Figura 27 – Evolução do desvio de tensão na rede 69 barras.	92
Figura 28 – Topologia ótima da rede 84 barras.	93
Figura 29 – Perfil de tensão na rede 84 barras.	94
Figura 30 – Evolução das perdas na rede 84 barras.	95

Figura 31 – Evolução do desvio de tensão na rede 84 barras.	97
Figura 32 – Topologia ótima da rede 136 barras.	99
Figura 33 – Perfil de tensão na rede 136 barras.	99
Figura 34 – Evolução das perdas na rede 136 barras.	102
Figura 35 – Evolução do desvio de tensão na rede 136 barras.	102

LISTA DE QUADROS

Quadro 1 – Comparação dos métodos clássicos e algoritmos genéticos no problema de RSDEE.	19
Quadro 2 – Valores de parâmetros recomendados para o BRKGA.	41
Quadro 3 – Pseudocódigo do BRKGA.	44
Quadro 4 – Pseudocódigo do algoritmo de Kruskal.	48
Quadro 5 – Pseudocódigo da codificação baseada na co-árvore.	53
Quadro 6 – Analogia entre o sistema físico e o recozimento simulado.	56
Quadro 7 – Pseudocódigo para reconfiguração usando SA.	65

LISTA DE TABELAS

Tabela 1 – Parâmetros de controle dos algoritmos propostos no sistema 14 barras.	82
Tabela 2 – Parâmetros de reinicialização aplicados ao BRKGA1 e BRKGA2 na rede 14 barras.	83
Tabela 3 – Resultados das metodologias propostas no sistema 14 barras. . . .	83
Tabela 4 – Parâmetros de controle dos algoritmos propostos no sistema 33 barras.	86
Tabela 5 – Parâmetros de reinicialização aplicados ao BRKGA1 e BRKGA2 na rede 33 barras.	87
Tabela 6 – Resultados das metodologias propostas no sistema 33 barras. . . .	87
Tabela 7 – Parâmetros de controle dos algoritmos propostos no sistema 69 barras.	90
Tabela 8 – Parâmetros de reinicialização aplicados ao BRKGA1 e BRKGA2 na rede 69 barras.	91
Tabela 9 – Resultados das metodologias propostas no sistema 69 barras. . . .	91
Tabela 10 – Parâmetros de controle dos algoritmos propostos no sistema 84 barras.	94
Tabela 11 – Parâmetros de reinicialização aplicados ao BRKGA1 e BRKGA2 na rede 84 barras.	95
Tabela 12 – Resultados das metodologias propostas no sistema 84 barras. . . .	96
Tabela 13 – Parâmetros de controle dos algoritmos propostos no sistema 136 barras.	100
Tabela 14 – Parâmetros de reinicialização aplicados ao BRKGA1 e BRKGA2 na rede 136 barras.	100
Tabela 15 – Resultados das metodologias propostas no sistema 136 barras. . .	101
Tabela 16 – Dados do sistema de 14 barras.	113
Tabela 17 – Dados do sistema de 33 barras.	114
Tabela 18 – Dados do sistema de 69 barras.	115
Tabela 19 – Dados do sistema de 69 barras (continuação).	116
Tabela 20 – Dados do sistema de 84 barras.	117
Tabela 21 – Dados do sistema de 84 barras (continuação 1).	118
Tabela 22 – Dados do sistema de 84 barras (continuação 2).	119
Tabela 23 – Dados do sistema de 136 barras.	120
Tabela 24 – Dados do sistema de 136 barras (continuação 1).	121
Tabela 25 – Dados do sistema de 136 barras (continuação 2).	122
Tabela 26 – Dados do sistema de 136 barras (continuação 3).	123

LISTA DE ABREVIATURAS E SIGLAS

ANEEL	Agência Nacional de Energia Elétrica
BRKGA	Biased Random-Key Genetic Algorithm
BRKGA1	Algoritmo genético de chaves aleatórias viciadas com decodificador baseado em algoritmo de Kruskal
BRKGA2	Algoritmo genético de chaves aleatórias viciadas com decodificador baseado em <i>links</i> da co-árvore
COM	Component Object Model
MATLAB	Matrix Laboratory
OpenDSS	Open Distribution System Simulator
PNLIM	Programação Não Linear Inteira Mista
RKGA	Random Key Genetic Algorithm
RSDEE	Reconfiguração de Sistemas de Distribuição de Energia Elétrica
SA	Simulated Annealing
SDBT	Sistema de Distribuição de Baixa Tensão
SDMT	Sistema de Distribuição de Média Tensão
VBA	Visual Basic for Applications

SUMÁRIO

1	INTRODUÇÃO	15
1.1	MOTIVAÇÃO E JUSTIFICATIVAS	15
1.2	OBJETIVOS	20
1.2.1	Objetivo Geral	20
1.2.2	Objetivos Específicos	20
1.3	ESTRUTURA DO TRABALHO	20
2	O PROBLEMA DE RECONFIGURAÇÃO DE SISTEMA DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA	22
2.1	VISÃO GERAL DO PROBLEMA DE RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA	22
2.2	REVISÃO BIBLIOGRÁFICA	24
2.2.1	PRIMEIRAS PESQUISAS EM RECONFIGURAÇÃO DE REDES	25
2.2.2	MÉTODOS DE OTIMIZAÇÃO CONVENCIONAIS APLICADOS À RECONFIGURAÇÃO	26
2.2.3	MÉTODOS DE BUSCA INTELIGENTE APLICADOS À RECONFIGURAÇÃO	28
2.3	CONTRIBUIÇÕES	31
3	MÉTODOS DE OTIMIZAÇÃO APLICADOS NO PROBLEMA DE RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO	33
3.1	MODELAGEM MATEMÁTICA PROBLEMA DE RECONFIGURAÇÃO	33
3.2	REPRESENTAÇÕES DE PROBLEMAS DE OTIMIZAÇÃO ATRAVÉS DE ALGORITMOS GENÉTICOS	36
3.2.1	Genótipos e Fenótipos	37
3.2.2	Decomposição da função <i>fitness</i>	38
3.3	ALGORITMO GENÉTICO DE CHAVES ALEATÓRIAS VICIADAS	39
3.3.1	Parte Independente do Problema: Estrutura geral do BRKGA	41
3.3.2	Parte Dependente do Problema: Decodificadores	45
3.3.2.1	Algoritmo de Kruskal	46
3.3.2.2	Algoritmo baseado na co-árvore	48
3.4	SIMULATED ANNEALING PARA OTIMIZAÇÃO COMBINATÓRIA	53
3.4.1	ALGORITMO DE METROPOLIS	54
3.4.2	ALGORITMO DE RECOZIMENTO SIMULADO	55
3.4.3	MECANISMO DE PERTURBAÇÃO	57
3.4.4	COOLING SCHEDULE	60
3.4.4.1	Temperatura Inicial	61
3.4.4.2	Função Decremento de Temperatura	62
3.4.4.3	Temperatura Final	62

3.4.4.4	Comprimento da Cadeia de Markov	63
3.4.5	RECOZIMENTO SIMULADO NO PROBLEMA DE RECONFIGURAÇÃO	64
3.5	ESTRATÉGIAS DE APLICAÇÃO DE RESTRIÇÕES	67
3.5.1	Eliminação	67
3.5.2	Funções Penalidade	67
3.6	METODOLOGIAS DE RESTART APLICADAS AO BRKGA	68
3.6.1	ESTRATÉGIA DE RESTART SEM ELITISMO	70
3.6.2	ESTRATÉGIA DE RESTART COM ELITISMO	71
3.7	METODOLOGIA HÍBRIDA BASEADA EM BRKGA E SA	73
3.7.1	PRESERVAÇÃO DA DIVERSIDADE	74
3.7.2	INICIALIZAÇÃO INTELIGENTE	75
3.7.3	ALGORITMO HÍBRIDO PROPOSTO	77
4	RESULTADOS	80
4.1	REDE DE DISTRIBUIÇÃO DE 14 BARRAS	80
4.2	REDE DE DISTRIBUIÇÃO DE 33 BARRAS	85
4.3	REDE DE DISTRIBUIÇÃO DE 69 BARRAS	89
4.4	REDE DE DISTRIBUIÇÃO DE 84 BARRAS	93
4.5	REDE DE DISTRIBUIÇÃO DE 136 BARRAS	98
5	CONCLUSÃO	103
5.1	SUGESTÕES PARA TRABALHOS FUTUROS	103
	REFERÊNCIAS	105
	ANEXO A – DADOS DOS SISTEMAS DE DISTRIBUIÇÃO	113
A.1	SISTEMA DE 14 BARRAS	113
A.2	SISTEMA DE 33 BARRAS	114
A.3	SISTEMA DE 69 BARRAS	115
A.4	SISTEMA DE 84 BARRAS	117
A.5	SISTEMA DE 136 BARRAS	120

1 INTRODUÇÃO

1.1 MOTIVAÇÃO E JUSTIFICATIVAS

Segundo o relatório Perdas de Energia Elétrica na Distribuição da ANEEL (CARRAZZA *et al.*, 2019), as perdas na distribuição correspondem à diferença entre a energia injetada e a energia fornecida. Sendo definida, neste mesmo relatório, a energia injetada como a energia ativa medida proveniente dos agentes supridores (transmissores, outras distribuidoras e geradores) e da geração própria necessária para atendimento do mercado da distribuidora. Enquanto a energia fornecida é definida como a energia ativa entregue, medida ou estimada, nos casos previstos pela legislação, a outras distribuidoras, às unidades consumidoras cativas e livres, mais o consumo próprio. Estas perdas podem ser divididas em duas categorias: perdas não técnicas e perdas técnicas.

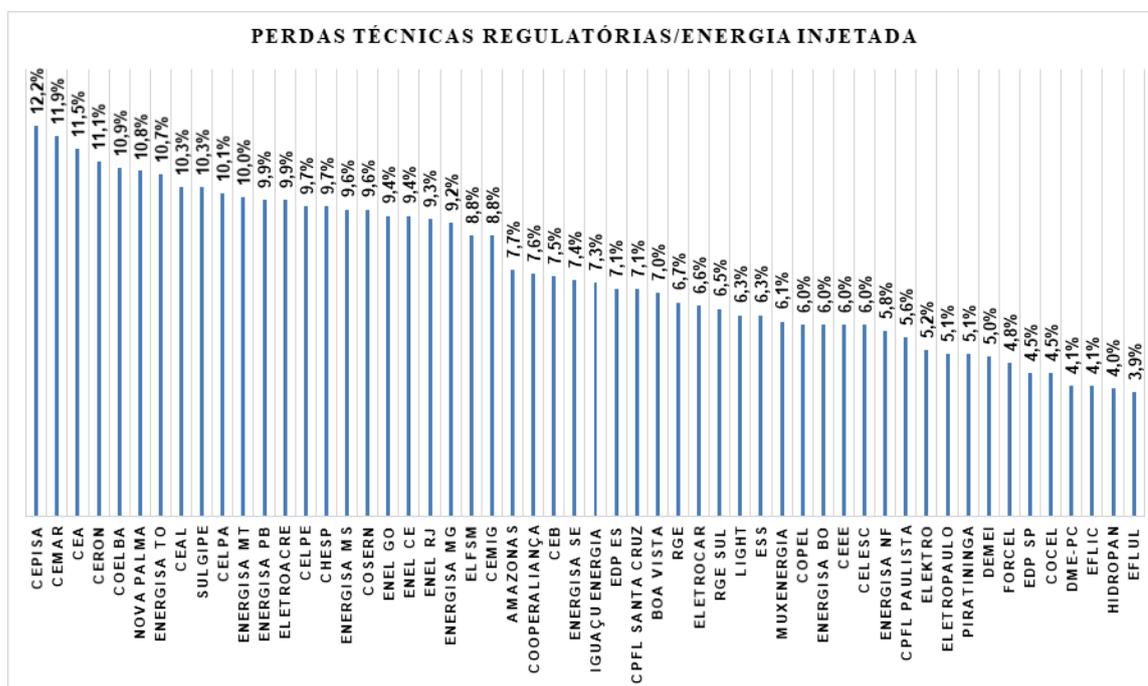
As perdas não técnicas decorrem principalmente de furtos (ligação clandestina e desvio direto da rede), fraudes de energia (adulteração no medidor), erros de medição e faturamento, entre outros. Essas perdas não técnicas correspondem à diferença entre as perdas na distribuição e as perdas técnicas. Enquanto as perdas técnicas correspondem à energia dissipada naturalmente no transporte através do sistema de distribuição devido a fenômenos físicos, desse modo, estão associadas as características de carregamento (perdas por efeito Joule nos condutores e perdas nos núcleos dos transformadores) e a configuração das redes das concessionárias de distribuição (perdas devido às correntes de fuga no ar e nos isoladores) (CARRAZZA *et al.*, 2019).

Estas perdas técnicas variam entre as diferentes características das redes de cada área de concessão, como ilustra a Figura 1 sobre os percentuais regulatórios de perdas técnicas sobre a energia injetada de 54 concessionárias no Brasil em 2018. Ainda segundo este relatório de Carrazza *et al.* (2019), no Brasil em 2018, o custo das perdas técnicas referente ao processo de geração, transmissão e distribuição era da ordem de R\$ 7,1 bilhões. Enquanto os custos das perdas na rede de distribuição considerados nas tarifas somaram aproximadamente R\$ 1,5 bilhão.

Diante deste cenário, a aplicação de processos de automação inteligente na rede elétrica, melhorias físicas (recondutoramento de cabos e substituição de equipamentos) e o uso de redes inteligentes (Smart Grids), mostram-se promissores para a melhoria da qualidade, confiabilidade e redução destas perdas na rede elétrica. Entre os processos de automação inteligente destaca-se a reconfiguração das redes de distribuição, uma vez que estas redes inicialmente produzem um grande número de perdas por causa do baixo nível de tensão do sistema, o procedimento da reconfiguração é capaz de melhorar o perfil de tensão no alimentador e reduzir os custos operacionais da rede elétrica, sob condições de operação normais.

Segundo o módulo 7 dos Procedimentos de Distribuição de Energia Elétrica no

Figura 1 – Perdas Técnicas/Energia injetada 2018.



Fonte: (CARRAZZA *et al.*, 2019).

Sistema Elétrico Nacional (PRODIST) (ANEEL, 2019) o cálculo das perdas técnicas é realizado de modo a considerar os equipamentos dos sistemas de distribuição (segmentos de rede, ramais, transformadores, reguladores e medidores) e os subgrupos de tensão (A1, A2, A3, A3a, A4 e B) aos quais esses segmentos e equipamentos pertencem. De acordo com o método empregado para o cálculo de perdas, o sistema de distribuição é segmentado em:

- Sistema de Distribuição de Alta Tensão (SDAT);
- Transformadores de potência;
- Sistema de Distribuição de Média Tensão (SDMT);
- Sistema de Distribuição de Baixa Tensão (SDBT);
- Transformadores de distribuição;
- Ramais de ligação e;
- Medidores de energia das unidades consumidoras do SDBT.

Conforme descrito em (ANEEL, 2019), as perdas de energia nas redes e equipamentos associados ao SDMT e ao SDBT são obtidas pela aplicação do método de fluxo de potência. Na nota técnica n° 0057/2014 (CAPELI; QUEIROZ, L. M.; SOUSA, 2014) da ANEEL é apresentada uma proposta de apuração das perdas através da execução do fluxo de carga nas redes de distribuição (média tensão e baixa tensão) através do *software Open Distribution System Simulator* (OpenDSS). Segundo esta

nota esses estudos já vêm sendo realizados pela maioria das distribuidoras como ferramenta de planejamento e gestão do carregamento do sistema de distribuição, tendo como característica a obtenção das perdas técnicas de maneira mais precisa do que as metodologias que a ANEEL têm empregado. Destaca-se ainda nesta nota técnica que essa ferramenta permitirá que os parâmetros regulatórios sejam definidos de modo mais objetivo e transparente, com o intuito de sinalizar às empresas a importância da perda técnica ser considerada como variável relevante no planejamento e gerenciamento do sistema de distribuição.

O problema da reconfiguração é normalmente modelado como um problema de Programação Não linear Inteira Mista (PNLIM), que pode ser resolvido usando diferentes abordagens, tais como, algoritmos heurísticos, metaheurísticas, otimização clássica e métodos híbridos que empregam diferentes estratégias de otimização para providenciar uma solução. De acordo com o teorema *No Free Lunch* (WOLPERT; MACREADY, 1997), existe um método de otimização que trabalha melhor para cada classe de problema de otimização, contudo não existe um único método que será melhor para todas as classes.

No caso de problemas de reconfiguração de redes de distribuição, onde muitos métodos foram propostos, a comparação entre as várias técnicas para o problema de RSDEE pode ser feita a partir das informações disponíveis sobre a qualidade e o esforço computacional das soluções. Entretanto, esta avaliação é difícil devido as diversas condições de simulação, como diferentes equipamentos do computador (tipo de processador, memória, etc.), linguagens de programação, sistemas teste de diferentes dimensões e funções objetivo, por exemplo. Levando em conta esses fatores, comparações diretas de métodos de reconfiguração de sistemas de distribuição são injustas, desse modo, poucas comparações indiretas tem sido propostas na literatura. A análise comparativa deve considerar o esforço computacional de diferentes métodos para resolver problemas de RSDEE de acordo com o tempo de processamento e o número de fluxos de carga necessários para encontrar a melhor solução (SILVEIRA *et al.*, 2021).

Nesse sentido, em (SILVEIRA, 2020) e (SILVEIRA *et al.*, 2021), foi desenvolvido um método para avaliar o desempenho de diferentes algoritmos, metaheurísticos e de otimização matemática, na solução do problema de RSDEE, com o objetivo de reduzir as perdas de potência ativa e ponderar o esforço computacional do processo de solução de cada técnica. Dada a natureza combinatória, existe um número muito grande de possíveis soluções para o problema e cada tipo de rede deve atender a características particulares. Esta característica das soluções no projeto de redes reduz o número de algoritmos que podem ser utilizados. Assim, nestes trabalhos foi proposta uma estratégia para comparar o desempenho dos métodos, considerando tanto o esforço computacional como a qualidade da solução do problema de RSDEE

onde sistemas teste padrões de diferentes dimensões foram usados na avaliação. No Quadro 1 são mostradas algumas conclusões obtidas deste trabalho bem como a apresentação das vantagens do algoritmo genético em relação aos métodos de otimização matemática.

Um procedimento não especializado e robusto (algoritmos genéticos e recozimento simulado, por exemplo) pode ser bem úteis nesta classe de problema. Pois, a linearização ou a simplificação do problema podem tornar o método tradicional aplicável, contudo, mesmo uma solução globalmente ideal garantida para a tarefa simplificada pode estar muito distante e, portanto, muito inferior a uma solução aproximada para o verdadeiro problema (SILVEIRA, 2020).

Assim, tanto os métodos de otimização determinísticos quanto os estocásticos tem seus próprios pontos fortes e fraquezas. Métodos determinísticos têm a garantia de alcançar soluções ótimas globalmente para certos problemas. Entretanto, podem algumas vezes não terem sucesso, quando lidam com funções bem condicionadas e problemas de grande escala. Métodos estocásticos, por outro lado, podem lidar com uma variedade de problemas, mas eles têm uma garantia teórica pobre de obter soluções globalmente ótimas. Esses métodos podem oferecer soluções ótimas globalmente, mas com uma garantia probabilística de que o tempo para encontrar a solução não seja uma quantidade de tempo infinito (DAS, A. K.; PRATIHAR, 2019).

O algoritmo genético, por exemplo, é robusto, fácil de implementar e aplicável para resolver os problemas de otimização contínuos e discretos. Entretanto, nota-se que oferece uma garantia probabilística de alcançar a solução ótima globalmente como outros métodos estocásticos. Além disso, o algoritmo genético possui uma pobre capacidade de busca local e possui um tempo lento até atingir o critério de parada.

Para aumentar as capacidades de busca global e local de um algoritmo genético, um modo pode ser trabalhar com uma população de tamanho grande. Entretanto, com esse tipo de processo de busca poderia ser computacionalmente muito mais caro devido ao tamanho grande da população (DAS, A. K.; PRATIHAR, 2019).

Muitos pesquisadores têm tentado combinar uma técnica de busca local, tais como *hill climbing*, otimização por enxame de partículas, otimização por colônia de formigas, busca tabu, recozimento simulado, entre outros, com as estruturas tradicionais de um algoritmo genético para melhorar sua performance global. Esses métodos híbridos dos algoritmos genéticos melhoram suas capacidades de exploração e aproveitamento das soluções, outro modo que pode melhorar a performance do algoritmo genético é através do uso de métodos *multi-start* (estratégias de *restart*) do algoritmo genético que nada mais é do que executar o algoritmo genético algumas vezes sob diferentes condições iniciais e a melhor solução obtida entre todas as execuções é considerada como solução ótima global (DAS, A. K.; PRATIHAR, 2019).

Desse modo, nesta dissertação foram avaliados os desempenhos de algumas

Quadro 1 – Comparação dos métodos clássicos e algoritmos genéticos no problema de RSDEE.

Métodos	Características
Programação Linear Inteira Mista (PLIM) e Programação Linear Inteira Mista Retangular (PLIMR)	À medida que o valor do parâmetro usado na linearização e os passos de discretizações aumentam no processo de linearização, o esforço computacional sofre agravamentos, devido ao aumento do tamanho do problema. São os modelos que apresentaram os esforços computacionais de maior eficiência (em relação aos modelos PNLIM, PNLIMR e PCSOIM), no geral, são capazes de apresentar soluções ótimas apresentando esforços computacionais mais baixos que os modelos não lineares e o modelo relaxado considerados para a resolver o problema de RSDEE.
Programação Não Linear Inteira Mista (PNLIM) e Programação Não Linear Inteira Mista Retangular (PNLIMR)	Apresentaram desempenhos muito caros computacionalmente na obtenção das soluções para os sistemas de pequeno e médio porte. Mostraram-se incapazes de obter uma solução ótima para sistemas de médio a grande porte.
Programação Cônica de Segunda Ordem Inteira Mista (PCSOIM)	Mesmo após o relaxamento da não linearidade o problema ainda exige um esforço computacional notável, principalmente em sistemas de grande porte. Apresenta esforços computacionais melhores que os modelos não convexos (PNLIM e PNLIMR), apresentando ótimas soluções para o problema de otimização.
Algoritmos Genéticos (Algoritmo Genético de Chaves Aleatórias Viciadas com decodificador Prim (BRKGA) e Algoritmo Genético de Chu-Beasley Especializado (AGCBE))	Embora não possam garantir matematicamente as soluções ótimas, ainda são adequados para resolver o problema de RSDEE independente do porte dos sistemas, pois encontram soluções de excelente qualidade (geralmente a ótima) em um baixo tempo de processamento. Além disso, possuem como vantagem: <ul style="list-style-type: none"> • Otimiza com variáveis contínuas e discretas; • Não requer informações de derivadas; • Pesquisas simultâneas a partir de uma ampla amostragem da superfície de custo; • Lida com um grande número de variáveis; • É bem adequado para computação paralela; • Otimiza variáveis com superfícies de custo extremamente complexas (podem saltar de um mínimo local); • Providencia uma lista de variáveis ótimas, não apenas uma única solução e; • Pode codificar as variáveis para que a otimização seja feita com as variáveis codificadas.

Fonte: (SILVEIRA, 2020), (HAUPT, R. L.; HAUPT, S. E., 2004)

técnicas de solução no problema da reconfiguração de rede de distribuição de energia elétrica, assim, foram aplicados um *Biased Random-Key Genetic Algorithm* (BRKGA)

em conjunto com duas codificações independentes do algoritmo principal, responsáveis por manter a estrutura radial da rede, sendo a primeira baseada no algoritmo de Kruskal e segunda baseada na co-árvore da rede, ambas as codificações foram executadas em três casos: sem o uso de *restart*, *restart* a cada κ iterações sem melhoria da função objetivo e, por último, uma estrutura que utiliza um monitoramento do *fitness* de alguns indivíduos da população para efetuar um *restart* com elitismo. Por fim, também foram utilizados o algoritmo *Simulated Annealing* (SA) e um algoritmo híbrido baseado em BRKGA e SA.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral desta dissertação consiste em avaliar o desempenho de diferentes ferramentas para a solução do problema de reconfiguração de sistemas de distribuição de energia elétrica e propor novas metodologias para minimização do tempo de execução e melhoria da qualidade dessas soluções.

1.2.2 Objetivos Específicos

- Aplicar os algoritmos BRKGA e SA ao problema da reconfiguração de redes de distribuição;
- Analisar os resultados obtidos de diferentes procedimentos de *restart* aplicados ao BRKGA no problema de reconfiguração de redes de distribuição para redução de perdas;
- Desenvolver um algoritmo híbrido baseado em BRKGA e SA aplicado ao problema da reconfiguração de redes elétricas para redução de perdas de energia;
- Comparar os resultados obtidos entre as diferentes metodologias desenvolvidas.

1.3 ESTRUTURA DO TRABALHO

Esta dissertação está organizada da seguinte forma:

- **Capítulo 1:** São apresentados os casos motivadores que fundamentam o desenvolvimento deste trabalho, os objetivos e a organização desta dissertação.
- **Capítulo 2:** É apresentada uma visão geral do problema, a revisão bibliográfica dos principais trabalhos desenvolvidos para a solução do problema da reconfiguração e as contribuições desta dissertação.

- **Capítulo 3:** Apresenta a formulação do problema da RSDEE, bem como os algoritmos *Biased Random-Key Genetic Algorithm* (com os decodificadores algoritmo de Kruskal e a codificação em co-árvore) e *Simulated Annealing*, técnicas de penalização das restrições utilizadas, estratégias de *restart* e um algoritmo híbrido para a solução do problema da RSDEE.
- **Capítulo 4:** São comparados os resultados obtidos das perdas de cada uma das técnicas propostas, também são mostrados o ganho no perfil de tensão de cada rede e o esforço computacional dessas ferramentas.
- **Capítulo 5:** Nesta seção encontram-se as conclusões deste trabalho e sugestões de trabalhos futuros.
- **Anexo A:** Neste anexo são mostrados os dados dos sistemas teste escolhidos para a abordagem do problema de reconfiguração.

2 O PROBLEMA DE RECONFIGURAÇÃO DE SISTEMA DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

2.1 VISÃO GERAL DO PROBLEMA DE RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

Os sistemas de distribuição de energia elétrica normalmente são construídos como redes interconectadas, porém mantêm uma estrutura radial de árvore. Isto significa que os sistemas de distribuição são divididos em subsistemas de alimentadores radiais, que contém um número considerável de chaves normalmente fechadas e outras normalmente abertas. Através da reconfiguração é possível redistribuir cargas entre alimentadores, alterando a topologia para uma outra estrutura, ainda radial, e mantendo todas as cargas conectadas, porém com redução de perdas e melhoria no perfil de tensão ao longo dos alimentadores.

Assim, o problema da reconfiguração de redes de distribuição consiste na alteração da topologia da rede em operação, realizada pela mudança de um conjunto de chaves normalmente abertas/fechadas a fim de encontrar uma nova topologia radial, este procedimento pode ser utilizado para diversos propósitos, tais como, redução de perdas, balanceamento de cargas, restauração de serviço, e assim por diante. Embora em diferentes propósitos, todos os problemas de reconfiguração têm em comum a característica de otimização combinatória e modelos matemáticos não lineares. Em todos os casos, o método de reconfiguração deve lidar internamente com algumas restrições, como conectividade da rede, configuração radial, limites de afundamento de tensão, fluxo de potência nos ramos, entre outros (SILVA; FALCÃO; LEE, K. Y., 2008).

O problema de RSDEE pode ser fundamentado como um problema de otimização da busca por uma solução factível que, ao mesmo tempo, satisfaz um grupo de restrições operacionais de igualdade e desigualdade. Sua complexidade combinatória é causada pela natureza discreta das variáveis de controle associadas às chaves, que estão colocadas em regiões estratégicas da rede para manter a estrutura radial da rede elétrica. Alternativamente, o espaço de busca aumenta exponencialmente de tamanho conforme o número de chaves no sistema aumenta (SILVEIRA *et al.*, 2021).

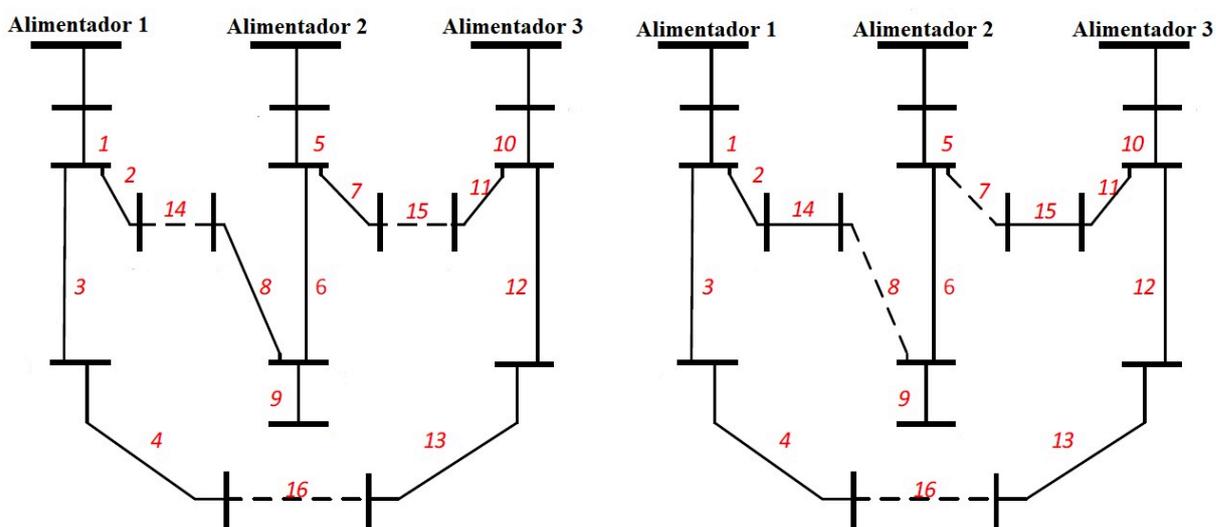
A complexidade do problema está diretamente ligada à quantidade de chaves de interconexão e o estado de cada chave de laço e seccionadora. Assim, dado um sistema com N chaves de manobra, esse sistema fornecerá 2^N configurações possíveis, contudo, a maior partes destas configurações são inviáveis, pois não atendem às restrições de radialidade ou pelo fato de possuírem barras desconectadas ao sistema. A combinação do número e estado dos dispositivos o caracteriza como um problema de natureza combinatória complexa de crescimento exponencial. Portanto, quanto maior a quantidade de chaves para controlar dentro da rede, maior será a dificuldade e a

complexidade para encontrar uma topologia radial que seja a solução ótima para o problema de RSDEE (SILVEIRA, 2020).

Uma rede de distribuição pode ser representada, de acordo com a teoria de grafos, como um grafo $G(N,E)$ que contém um conjunto N de nós e um conjunto E de ramos. Onde todos os nós representam ou um nó fonte ou um nó dissipador, considerando que um ramo representa uma seção do alimentador que pode ser, ou carregada (chave fechada) ou descarregada (chave aberta). Uma vez que a rede é radial, para que as seções do alimentador formem um conjunto de árvores cada nó dissipador deve ser suprido por exatamente um nó fonte. Desta forma, a reconfiguração de redes de distribuição pode ser vista como um problema de busca de uma árvore ótima com o menor valor de perdas de potência ativa de um dado grafo, respeitando a restrições como balanço de potência (leis de Kirchhoff), limites de tensão nas barras, limites de corrente nos ramos, além da condição de radialidade, entre outras restrições que complementam a representação do problema (ZHU, 2015).

Para exemplificar o que acontece em um processo de RSDEE, a Figura 2 ilustra o caso de uma topologia radial em um sistema de 14 barras, obtido de (CIVANLAR *et al.*, 1988). No lado esquerdo desta figura encontra-se o sistema original antes da reconfiguração, em que as linha contínuas representam as chaves fechadas e as linhas tracejadas representam as chaves abertas (14, 15 e 16), neste caso produzindo uma perda de potência ativa de 511,43 kW. Após a reconfiguração da rede, mostrada no lado direito da Figura 2, a nova topologia da rede apresenta as chaves 7, 8 e 16 abertas, mantendo a radialidade do sistema, entretanto, a nova topologia possui como perdas de potência ativa o valor de 468,1 kW.

Figura 2 – Topologia inicial e topologia após a reconfiguração do sistema de 14 barras.



Fonte: (CIVANLAR *et al.*, 1988), (SILVEIRA, 2020).

O processo de busca desta solução ótima, no caso das metaheurísticas, é feito através da exploração do espaço solução, simplificada, é um conjunto que contém todas as configurações possíveis para o sistema de distribuição de energia analisado. Assim, o processo de busca consiste em procurar a melhor configuração da rede em um conjunto de alternativas disponíveis, dentro do espaço solução, por meio de uma função objetivo específica, neste caso se trata da perda de potência ativa, que apresenta (ou se aproxima) de um valor ótimo desejado, a mínima perda de energia.

Essas metaheurísticas guiam o processo de busca, objetivando explorar eficientemente o espaço solução a fim de encontrar essas soluções, utilizando desde técnicas simples de busca local a procedimentos complexos de aprendizagem. Essas técnicas empregam um certo grau de aleatoriedade para encontrar soluções tão ótimas quanto possível para problemas reconhecidamente difíceis. A estrutura desses algoritmos pode ser aplicada para resolver tais problemas de um modo aproximado, pela combinação de métodos contrutivos com estratégias de busca local ou baseadas em população, bem como estratégias para escapar de ótimos locais e realizar uma busca robusta no espaço de soluções de um problema.

Esses métodos são aplicados para encontrar respostas de problemas sobre os quais há poucas informações e métodos do tipo força bruta são desconsiderados devido ao espaço de solução ser muito grande. Desse modo, seu processo de busca não é feito as "cegas", como nos métodos de força bruta, sua busca é guiada através de processos de avaliação da função objetivo que para o caso deste trabalho são obtidos através do fluxo de carga da rede elétrica. Assim, cada vez que um novo valor de perda é obtido deste fluxo de carga a metaheurística irá determinar o processo de busca através da avaliação de cada solução candidata do problema, verificando sua otimizabilidade e guiando novos caminhos de busca, caso não seja encontrada a solução ótima.

2.2 REVISÃO BIBLIOGRÁFICA

Nesta seção é apresentada uma síntese das principais pesquisas desenvolvidas, relacionadas ao problema da reconfiguração de redes de distribuição de energia. As várias técnicas de otimização utilizadas nessas operações incluem os métodos tradicionais e modernos de otimização que, em geral, estão dentro destes dois grupos (ZHU, 2015):

- a) Métodos de otimização convencionais:
 - Abordagens de otimização irrestritas;
 - Programação não-linear;
 - Programação linear;

- Programação quadrática;
 - Método do gradiente reduzido generalizado;
 - Método de Newton;
 - Programação de fluxo de rede;
 - Programação inteira mista;
 - Método dos pontos interiores.
- b) Métodos de busca inteligente:
- Algoritmos evolucionários;
 - Busca Tabu;
 - Recozimento simulado;
 - Otimização por enxame de partículas.

2.2.1 PRIMEIRAS PESQUISAS EM RECONFIGURAÇÃO DE REDES

Em 1975, os engenheiros Merlin e Back foram os primeiros a abordar o problema da reconfiguração de redes de distribuição para redução de perdas como um problema de natureza combinatória de otimização, não diferenciável e com restrições rígidas. Usando a técnica de otimização *branch and bound* para redes de distribuição de grande porte e uma abordagem exata para redes de pequeno porte. Além de uma representação baseada na árvore de geração mínima para modelagem da estrutura da rede, com seções de linha representadas por arcos de grafo e as barras pelos nós da rede. Assim, Merlin e Back (1975) introduzem a possibilidade de reduzir perdas na rede apenas pela análise de métodos matemáticos de exploração do espaço de soluções das configurações factíveis da rede sem a necessidade de investimentos em novos equipamentos.

Desde então, foram desenvolvidas novas técnicas de otimização para solução do problema de reconfiguração de redes, tais como, o trabalho de Civanlar *et al.* (1988) que desenvolveram uma técnica denominada *branch exchange*. Nessa técnica heurística, cada etapa realiza uma permutação entre pares de chaves, uma aberta e outra fechada, eliminando as opções de manobra indesejáveis até o ponto em que não seja mais possível reduzir as perdas, reduzindo assim até as soluções candidatas, opções de manobras viáveis que não violam as restrições, convergindo ao estado em que não seja mais possível reduzir perdas, geralmente alcançando uma solução local.

Ainda no mesmo ano (LEE, R. E.; BROOKS, 1988), desenvolveram um método que avalia os efeitos da reconfiguração do sistema e a alocação de banco de capacitores, considerando dois cenários, o primeiro para um modelo dos tipos e localização de cargas para tempos representativos do ano, e o segundo com um modelo elétrico

do sistema de distribuição para cálculo do impacto de várias distribuições de carga em diferentes períodos do ano.

Em Baran e Wu (1989) foi dada continuidade ao trabalho desenvolvido por Civanlar *et al.* (1988) com a introdução de dois diferentes métodos de fluxo de carga, cada um com diferentes graus de precisão, esses novos métodos consideravam ambas potências real e a reativa. Com os resultados dos testes, foi observado que os métodos de estimação desenvolvidos eram computacionalmente eficientes, podendo serem usados em buscas da reconfiguração de um dado sistema. Esta abordagem de busca introduzida no artigo não é exaustiva e seu tempo de execução é aceitável, porém não garante encontrar o ótimo global.

Por último, em Shirmohammadi e Hong (1989) é desenvolvido um método heurístico baseado na pesquisa de Merlin e Back (1975). Nessa técnica o encontro do ponto de ótimo (ou próximo do ótimo) e a solução final são independentes do estado inicial das chaves da rede. Essa metodologia propõe melhorias em relação as desvantagens de Merlin e Back (1975), assim, cargas deixaram de ser assumidas como puramente ativas e fixas indiferente da mudança de configuração da rede, e passaram a ser considerados os ângulos da tensão e as restrições. Além disso, uma nova técnica de fluxo de carga AC baseada na compensação para a solução eficiente de rede de distribuição fracamente malhadas é implementada na metodologia de redução de perdas.

2.2.2 MÉTODOS DE OTIMIZAÇÃO CONVENCIONAIS APLICADOS À RECONFIGURAÇÃO

A partir das décadas seguintes diversos trabalhos abordando o problema da reconfiguração de redes foram surgindo, explorando novos métodos de busca, novas heurísticas, metaheurísticas, inteligência de enxames, comportamento de grupos na busca por comida, entre outros, além de adaptações dos métodos clássicos baseados em otimização numérica.

Em Schmidt *et al.* (2005), foi desenvolvido uma busca através do método de otimização não-linear inteira mista denominado *Best-First Search* para estabelecer o estado de todas as chaves combinado com o método padrão de Newton para cálculo das correntes de ramo. O estudo tem como desvantagem a não garantia da otimalidade da solução, mas a alta qualidade das soluções subótimas encontradas, junto com a alta velocidade de processamento, faz dessa abordagem atrativa para sistemas de distribuição em tempo real.

A proposta de Khodr *et al.* (2009), foi estudar o problema da reconfiguração de redes de distribuição através da integração do fluxo de carga ótimo e na abordagem de decomposição de Benders. A formulação é incorporada em dois estágios: o primeiro é o problema mestre e é formulado como um problema de programação não-linear

inteira, onde é determinado a topologia radial da rede. No segundo estágio, chamado de estágio escravo, também é formulado como um problema de programação não linear, e é usado para determinar a viabilidade da solução do problema mestre por meio de um fluxo de potência ótimo e fornece a informação para o corte linear de Benders que conecta ambos os problemas. Contudo, a não convexidade de tal formulação não-linear não garante o encontro da solução ótima global, desta forma, caso haja problemas de convergência ações remediadoras devem ser implementadas.

Em Taylor e Hover (2012), foi utilizado um método de programação quadrática inteira mista, com programação restrita quadraticamente, modelos de programação de cone de segunda ordem e também formulações do fluxo de carga AC com convexidade e relaxações contínuas. Este estudo se assemelha ao trabalho de Khodr *et al.* (2009), onde uma programação não linear inteira mista foi resolvida usando decomposição de Bender, porém é resolvido o inteiro misto, não linear, através de formulações convexas. No estudo, três novas formulações convexas foram introduzidas e representam as primeiras aproximações convexas para a versão de fluxo de potência AC do problema. Os algoritmos empregados garantem o ótimo local dentro de cada aproximação e se mostram eficientes obtendo boas soluções e praticidade em grandes sistemas.

Na proposta de Zheng Huang, Hara e Kita (2014), um Algoritmo de Fluxo Inteligente é desenvolvido para o problema de reconfiguração de redes de distribuição. A ideia deste algoritmo é baseada no método heurístico para o balanço do suprimento de cargas da distribuição para minimização de perdas de linha. Neste método de otimização a geração da topologia da rede é direta ao invés de efetuar uma pesquisa numérica não ordenada. Além disso, múltiplas informações de fluxo de carga, tais como, tensão, perdas de potência ativa e grau de quebra de conectividade são utilizadas para a geração das topologias. Assim, as regras de geração de fluxo garantem a validação topológica das soluções candidatas e nenhum método estocástico é usado, apenas um eficiente parâmetro é utilizado. Desta forma, o algoritmo encontra a configuração de mínima perda de linha baseado na revisão das configurações balanceadas.

Por fim, Teshome e Lian (2018) propõem um método de programação linear inteira mista usado para a reconfiguração de redes de distribuição com e sem geração distribuída. Nesta formulação poucas variáveis e restrições são exigidas para um modelo preciso da reconfiguração de redes de distribuição e uma considerável redução do esforço computacional é obtido em relação as abordagens anteriores de métodos de programação linear inteira mista. Também integra vários geradores distribuídos na reconfiguração, incorpora as perdas de linha e considera restrições de operação típicas. A localização dos geradores distribuídos e seus dimensionamentos são simultaneamente otimizados no método, enquanto a restrição de radialidade é mantida.

2.2.3 MÉTODOS DE BUSCA INTELIGENTE APLICADOS À RECONFIGURAÇÃO

Os métodos de busca inteligente, em geral, são desenvolvidos especificamente para um determinado problema, encontrando soluções de boa qualidade para problemas complexos e com baixo esforço computacional em comparação com os métodos exatos. Estes métodos podem ser baseados em experiência (treinamento), na natureza ou ainda no senso comum, por exemplo. Podem ainda serem baseados em população ou em uma única função sob sucessivas iterações para encontrar a solução da função objetivo.

Entre os trabalhos que usam métodos de busca inteligente, está Hong-Chan Chang e Kuo (1994) que utiliza um método baseado em busca em vizinhança, denominado recozimento simulado, para a reconfiguração de redes para redução de perdas em sistemas de distribuição. Embora o recozimento simulado evite ficar preso em um mínimo local e tenha uma qualidade da solução encontrada, este exige um excessivo esforço computacional para encontrar o mínimo global. Para contornar isso, são propostos um esquema de perturbação para seleção da abertura das chaves, um conjunto de equações simplificadas de fluxo de potência e um procedimento de inicialização para determinação de uma melhor temperatura inicial para o recozimento simulado. Assim, o esforço computacional é reduzido sem degradação da qualidade da solução.

No estudo de Jeon *et al.* (2002) foi desenvolvido um recozimento simulado para a solução do problema de reconfiguração de redes, porém, diferente do mecanismo de perturbação aleatório proposto em Hong-Chan Chang e Kuo (1994), foi proposto um melhorado mecanismo de perturbação que relaciona a topologia da rede e a temperatura para a escolha das configurações, além disso, também foi aplicado um *cooling schedule* de tempo polinomial que se baseia em métodos estatísticos durante o processo de busca para definir os parâmetros de controle do algoritmo. O novo método de perturbação permite que o espaço solução seja diversificado em altas temperaturas e intensificado na região de atração em baixas temperaturas, mais do que isso, permite que o recozimento simulado escape mais facilmente de mínimos locais e alcance mais rapidamente a vizinhança da solução ótima.

Em Leonardo Queiroz e Lyra (2009) são aplicados algoritmos genéticos para resolver o problema de redução de perdas nos sistemas de distribuição considerando variações temporais de cargas. Para evitar soluções não factíveis, os autores adotam o algoritmo de Kruskal para gerar árvores de geração mínima e descartar os indivíduos não factíveis. Nesse estudo são realizados ainda uma análise comparativa de um algoritmo genético simples, algoritmo genético híbrido, algoritmo genético com mutação proporcional, algoritmo genético adaptativo com aumento da taxa de mutação, algoritmo genético adaptativo com diminuição da taxa de mutação, e um algoritmo genético dinâmico adaptativo. Por fim, comparando os resultados do algoritmo genético simples e do algoritmo genético híbrido é notado que o uso da busca local no método

híbrido apresentou uma grande melhoria no *fitness* devido à aplicação da busca local na primeira geração.

Em Zhong-hui, Jian-yong e Heng (2010), foi desenvolvida uma metodologia para combinar a capacidade de busca em larga escala do algoritmo *Artificial Fish Swarm* e a capacidade de busca local através da busca Tabu, melhorando a habilidade de otimização global pelo uso do homeomorfismo para comprimir o espaço de busca global. O algoritmo proposto é capaz de melhorar a eficiência da busca do algoritmo através da redução do espaço de busca de soluções ótimas, enquanto melhora a precisão pela capacidade de busca local da busca Tabu, superando a falha que a busca Tabu possui a sensibilidade ao valor inicial.

A proposta de Aman *et al.* (2014), foi aplicar um algoritmo Colônia de Abelhas Artificial Discreto para encontrar o posicionamento ótimo das chaves de laço, combinado com estruturas de teoria de grafos e a análise do fluxo de carga radial Thukaram. O método de fluxo de potência radial é executado para cada combinação de manobra de sucesso e as perdas do sistema de potência são computadas atualizando a posição das chaves de laço até encontrar a posição ótima das chaves.

No estudo de Gu, Ji e Liu (2014), um algoritmo genético imune é usado para diminuição de perdas na rede e melhoria da qualidade de tensão em redes de distribuição. No processo, alguns anticorpos como células de memória são preservados, quando o mesmo antígeno é invadido de novo, células de memória são ativadas e produzem anticorpos novamente para fazer a resposta mais rápida do que a resposta inicial, a função de memória do sistema imune. O algoritmo genético imune comparado com o algoritmo genético básico pode acelerar a velocidade da busca, melhorar a habilidade de busca global e assegurar rápida obtenção da solução do ótimo global. Além disso, mantém a diversidade do anticorpo melhorando a habilidade de busca global, evitando a convergência prematura, e também pode melhorar a habilidade de busca local.

Em Mohamed Imran e Kowsalya (2014) é apresentado o *Fireworks Algorithm* para resolver o problema de reconfiguração com os objetivos reduzir perdas e de melhorar o perfil de tensão. Este é um algoritmo de inteligência de enxame baseado no conceito do processo de explosão de fogos de artifício para a busca para a melhor localização de faíscas, é uma busca em espaço local em torno de um ponto específico onde o fogo de artifício explode por meio das faíscas geradas na explosão. Em cada geração o algoritmo seleciona alguns pontos de qualidade como fogos de artifício, o que gera um número de faíscas para a busca do espaço local em torno dele. O processo de busca continua até que pelo menos uma faísca alcance uma solução ótima adequada, ou que o critério de parada seja alcançado.

A proposta de Ruan *et al.* (2016), foi desenvolvida a partir de uma melhorada otimização quântica de enxame de partículas para a redução de perdas em redes

de distribuição com geração distribuída. A topologia da rede é analisada na base da estrutura da matriz incidência de ramo e matriz incidência de ramo de laço, essas matrizes identificam trechos ilhados e laços. Depois, a perda da rede é calculada pelo método *forward-backward sweep*. Também, um fator de distúrbio é introduzido e em conjunto com o esquema de codificação decimal, assegura-se o encontro da solução ótima e o aumento na performance computacional.

No estudo de (FARIA; RESENDE, M. G.; ERNST, 2017) um algoritmo genético de chaves aleatórias viciadas foi aplicado ao problema de reconfiguração, neste algoritmo soluções são codificadas pelo uso de *random keys*, que são vetores de números reais no intervalo de zero a um, que possibilitam uma busca indireta da solução dentro de um espaço de busca proprietário. Além disso, os operadores genéticos não precisam ser modificados para gerar apenas soluções factíveis e, também, uma estratégia de elitismo é aplicada. Um decodificador, responsável pelo atendimento da restrição da configuração radial, é construído aplicando um conjunto de regras da teoria de grafos que garantem a geração de apenas soluções factíveis ao problema.

Em Singh, Chauhan e Reddy (2017), uma técnica híbrida baseada na heurística *branch exchange* e na metaheurística Otimização do Sistema Imunológico Artificial é usada para encontrar a configuração ótima de uma rede para a minimização de perdas e, então, é comparada com a Otimização de Exame de Partículas Seletiva e Otimização de Sistema Imune Artificial através do tempo de execução efetivo. As topologias aleatórias iniciais são geradas pela Otimização do Sistema Imunológico Artificial e cromossomos adicionais são melhorados pela técnica heurística *branch exchange*. A aplicação de ambas as técnicas reduz o tempo de execução do algoritmo e a reconfiguração ótima é obtida em um número menor de iterações.

Em Hasanpour *et al.* (2017), um *Cuckoo Search Algorithm* é aplicado para reduzir as perdas de potência e melhorar o perfil de tensão em uma rede, com diferentes unidades de geração distribuída e variação de carga. O *Cuckoo Search Algorithm* é uma metaheurística inspirada no comportamento parasita de algumas espécies de pássaros cuco que depositam ovos em ninhos de outras espécies de aves hospedeiras, combinando uma busca aleatória local e uma busca exploratória global. No estudo, o perfil de variação de carga ao longo de 24 horas e as fontes eólica, solar e pequenas unidades de geração hídrica são considerados na reconfiguração.

Em López *et al.* (2018), uma simulação de Monte Carlo e um Algoritmo Genético são utilizados a fim de estruturar uma abordagem mono-objetivo de programação estocástica para tratamento da demanda como variável aleatória. O modelo baseado na técnica de Monte Carlo é desenvolvido para atribuir um nível aleatório em cada nó do sistema, e um comportamento estatístico das cargas é atribuído usando funções de densidade de probabilidade uniforme. Assim, o sorteio aleatório de Monte Carlo atribui os níveis de carga para o modelo de reconfiguração determinística e o Algoritmo Gené-

tico encontra a melhor solução, o processo é repetido por um certo número de vezes, de acordo com a técnica de Monte Carlo. Também são usadas as técnicas Diakoptic e compressão de grafo que permitem identificar um reduzido grupo de topologias ótimas, para uma grande faixa de valores de carga, reduzindo drasticamente espaço de busca.

No trabalho de Cavalheiro, Vergílio e Lyra (2018), também foi aplicado um algoritmo genético de chaves aleatórias viciadas para o problema da reconfiguração de redes como em (FARIA; RESENDE, M. G.; ERNST, 2017), porém o problema possui uma abordagem adicional com fontes de energia renováveis variáveis, todavia, para a construção do decodificador foi utilizado o algoritmo de Kruskal para a geração de soluções factíveis ao problema, atendimento da restrição da radialidade da rede.

Em Vargas, Romero e Franco (2018), também foi aplicado um algoritmo genético de chaves aleatórias viciadas para a solução do problema de reconfiguração, assim como feito por Cavalheiro, Vergílio e Lyra (2018) e (FARIA; RESENDE, M. G.; ERNST, 2017), porém, como decodificador foi proposto um algoritmo de Prim para a geração de soluções factíveis ao problema, mantendo a restrição da radialidade da rede.

Por fim, em Abbaskhani-Davanloo *et al.* (2019), foi desenvolvido um método de reconfiguração de rede baseado em Algoritmos Genéticos e *Fuzzy* para melhoria das perdas horárias nos sistemas de distribuição incorporando incertezas de carga e gerações renováveis. A função objetivo nesse problema inclui duas partes, a primeira é a minimização de perdas e a segunda parte é o ajuste do índice de tensão para regular a tensão em cada barra. Em cada tempo de reconfiguração, um certo número de chaves é aberto. Para encontrar uma solução ótima considerando incertezas, o problema é resolvido sob diversos cenários gerados.

2.3 CONTRIBUIÇÕES

Com base na revisão bibliográfica descrita acima, esta dissertação apresenta dois novos métodos para a solução do problema de reconfiguração de redes de distribuição de energia, cujas contribuições para a aceleração e melhoria da qualidade das soluções encontradas são apresentadas, como segue:

- O primeiro método proposto segue os estudos de (CAVALHEIRO; VERGÍLIO; LYRA, 2018) e (FARIA; RESENDE, M. G.; ERNST, 2017) utilizando o algoritmo genético de chaves aleatórias viciadas e os decodificadores propostos, porém para minimizar o tempo de execução do algoritmo duas estratégias de *restart* são adicionadas a estes dois algoritmos genéticos.
- O segundo método proposto é baseado nos estudos de Jeon *et al.* (2002), porém com a alteração da função decremento baseada em (HUANG, M. D.; ROMEO; SANGIOVANNI-VINCENTELLI, 1986), a fim de acelerar a obtenção da solução e melhorar a qualidade das soluções obtidas.

- O terceiro método proposto é um algoritmo híbrido utilizando as mesmas ferramentas de (FARIA; RESENDE, M. G.; ERNST, 2017) para a exploração da busca global, e adicionando o algoritmo de recozimento simulado para a busca local, derivado de Jeon *et al.* (2002), a fim de minimizar o esforço computacional e melhorar a qualidade das soluções obtidas.
- Considerando os mesmos sistemas testes, os métodos propostos apresentaram melhores resultados do que os trabalhos de Aman *et al.* (2014), Zhonghui, Jian-yong e Heng (2010), Mohamed Imran e Kowsalya (2014), Singh, Chauhan e Reddy (2017) e (SILVEIRA, 2020), por exemplo.

3 MÉTODOS DE OTIMIZAÇÃO APLICADOS NO PROBLEMA DE RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO

3.1 MODELAGEM MATEMÁTICA PROBLEMA DE RECONFIGURAÇÃO

Neste capítulo são apresentadas as formulações matemáticas do problema de RSDEE. O problema de RSDEE pode ser representado usando um modelo de PNLIM com característica não convexa, onde as variáveis inteiras representam o estado de operação das chaves de manobra (chaves de laço e seccionadoras) e as variáveis contínuas estão associadas a processos operacionais (corrente que passa pelos ramos, tensão nas barras, entre outros). O intuito é obter uma configuração radial, dentre todas as configurações possíveis, que minimize as perdas de potência ativa, satisfazendo restrições operacionais através de mudanças do estado das chaves de laço e seccionadoras para alterar a topologia da rede (SILVEIRA, 2020).

O modelo matemático da função objetivo do problema de reconfiguração de redes de distribuição de energia pode ser expressado pelo uso da corrente de ramo, equação (1). A formulação matemática está sujeita ainda as restrições de características operacionais (topologia radial e limites de tensão e corrente) e físicas (leis de Kirchhoff). Desta forma, o problema de RSDEE é representado como um problema de minimização de perdas de potência ativa da rede, considerando os restrições descritas nas equações (2) a (16), segundo a proposta de (SILVEIRA, 2020) (BORGES; FRANCO; RIDER, 2013).

$$\min v = \sum_{mn \in \Omega_b} R_{mn} I_{mn}^{sqr} \quad (1)$$

Sujeito a,

$$\sum_{km \in \Omega_b} P_{km} - \sum_{mn \in \Omega_b} (P_{mn} + R_{mn} I_{mn}^{sqr}) + P_m^G = P_m^D \quad \forall m \in \Omega_n \quad (2)$$

$$\sum_{km \in \Omega_b} Q_{km} - \sum_{mn \in \Omega_b} (Q_{mn} + X_{mn} I_{mn}^{sqr}) + Q_m^G = Q_m^D \quad \forall m \in \Omega_n \quad (3)$$

$$V_m^{sqr} - V_n^{sqr} = 2(R_{mn} P_{mn} + X_{mn} Q_{mn}) + Z_{mn}^2 I_{mn}^{sqr} - w_{mn} \quad \forall mn \in \Omega_b \quad (4)$$

$$V_n^{sqr} I_{mn}^{sqr} = P_{mn}^2 + Q_{mn}^2 \quad \forall mn \in \Omega_b \quad (5)$$

$$\underline{V}^2 \leq V_m^{sqr} \leq \overline{V}^2 \quad \forall m \in \Omega_n \quad (6)$$

$$I_m^{sqr} \leq \overline{I}_{mn}^2 (y_{mn}^+ + y_{mn}^-) \quad \forall mn \in \Omega_b \quad (7)$$

$$P_{mn}^+ - P_{mn}^- = P_{mn} \quad \forall mn \in \Omega_b \quad (8)$$

$$0 \leq P_{mn}^+ \leq \overline{V} I_{mn} y_{mn}^+ \quad \forall mn \in \Omega_b \quad (9)$$

$$0 \leq P_{mn}^- \leq \overline{V} I_{mn} y_{mn}^- \quad \forall mn \in \Omega_b \quad (10)$$

$$-\overline{V} I_{mn}(y_{mn}^+ + y_{mn}^-) \leq Q_{mn} \leq \overline{V} I_{mn}(y_{mn}^+ + y_{mn}^-) \quad \forall mn \in \Omega_b \quad (11)$$

$$|w_{mn}| \leq (\overline{V}^2 - \underline{V}^2)(1 - (y_{mn}^+ + y_{mn}^-)) \quad \forall mn \in \Omega_b \quad (12)$$

$$\sum_{mn \in \Omega_b} (y_{mn}^+ + y_{mn}^-) = |N| - 1 \quad \forall mn \in \Omega_b \quad (13)$$

$$y_{mn}^+ + y_{mn}^- \leq 1 \quad \forall mn \in \Omega_b \quad (14)$$

$$y_{mn}^+, y_{mn}^- \in \{0,1\} \quad \forall mn \in \Omega_b \quad (15)$$

$$\sum_{mn \in \Omega_b} y_{mn}^+ + \sum_{nm \in \Omega_b} y_{nm}^- = 1 \quad \forall m \in \Omega_n \quad (16)$$

Em que,

- \overline{I}_{mn} : Corrente máxima do ramo mn ;
- I_{mn}^{sqr} : Quadrado da corrente do ramo mn ;
- Ω_b : Conjunto de ramos;
- Ω_n : Conjunto de nós;
- P_{km} : Fluxo de potência ativa no ramo km ;
- P_{mn} : Fluxo de potência ativa no ramo mn ;
- P_{mn}^+ : Fluxo de potência ativa no ramo mn em direção direta;
- P_{mn}^- : Fluxo de potência ativa no ramo mn em direção inversa;
- P_m^D : Potência ativa demandada no nó m ;
- P_m^G : Potência ativa gerada no nó m ;
- Q_{km} : Fluxo de potência reativa no ramo km ;
- Q_{mn} : Fluxo de potência reativa no ramo mn ;
- Q_m^D : Potência reativa demandada no nó m ;

- Q_m^G : Potência reativa gerada no nó m ;
- R_{mn} : Resistência do ramo mn ;
- \bar{V} : Limite superior da tensão;
- \underline{V} : Limite inferior da tensão;
- V_n^{sqr} : Quadrado da tensão no nó n ;
- w_{mn} : Variável auxiliar usada na modelagem do estado do ramo mn ;
- X_{mn} : Reatância do ramo mn ;
- y_{mn}^+ : Variável binária associada à direção direta do fluxo de carga no ramo mn ;
- y_{mn}^- : Variável binária associada à direção inversa do fluxo de carga no ramo mn ;
- Z_{mn} : Impedância do ramo mn .

Com o objetivo de obter os mesmos resultados das literaturas, as cargas dos sistemas teste foram modeladas como potência ativa e reativa constantes e não no modelo ZIP proposto no módulo 7 do prodist (ANEEL, 2019).

No modelo de Programação Não Linear Inteira Mista proposto, apresentado nas equações (1) a (16), todos os ramos possuem uma chave de interconexão que pode estar aberta ou fechada, isto é, o ramo pode estar em operação ou não.

A função objetivo equação (1) representa as perdas de potência ativa do problema de RSDEE. As restrições (2) e (3) correspondem ao balanço de potência ativa e reativa, respectivamente, garantindo o cumprimento da primeira lei de *Kirchhoff*. A queda da magnitude de tensão nos ramos é representada pela Equação (4) e o cálculo da magnitude do fluxo de corrente é expressado pela Equação (5). A equação (4) possui uma variável adicional w_{mn} que depende do estado das chaves de interconexão, esta equação permite obter a magnitude da tensão nodal final em termos da magnitude da tensão nodal inicial, da potência ativa e reativa, da magnitude da corrente e dos parâmetros elétricos do ramo (resistência e reatância). Os limites de magnitude de tensão e a capacidade de fluxo de corrente no ramo são representadas pelas equações (6) e (7), respectivamente (BORGES; FRANCO; RIDER, 2013) (SILVEIRA, 2020).

O fluxo de potência ativa do sistema de distribuição elétrica é representado usando duas variáveis na Equação (8) para identificar a direção do fluxo de potência ativa. Assim, P_{mn}^+ indica que o fluxo segue a direção positiva (de m para n), enquanto P_{mn}^- indica que o fluxo segue a direção negativa (de n para m). Essas direções são definidas pelas equações (9) e (10), conforme as variáveis binárias y_{mn}^+ e y_{mn}^- , e podem ser incluídas como restrições adicionais ao modelo. O limite de potência reativa é dado pela Equação (11). A operação da variável auxiliar w_{mn} é limitada pela Equação (12) de acordo com o estado de operação da chave de interligação, assim, se a chave

estiver fechada tem-se que $w_{mn} = 0$, caso contrário, se a chave estiver aberta, w_{mn} é livre para variar no intervalo estabelecido pela Equação (12), devendo satisfazer a Equação (4) (BORGES; FRANCO; RIDER, 2013) (SILVEIRA, 2020).

A restrição da Equação (13) é necessária, embora não suficiente, para assegurar a configuração radial da rede. A fim de garantir a radialidade, considera-se que todas as barras possuem carga, assim o uso da restrição, Equação (13), combinado com as restrições das equações (2) e (3) geram a radialidade. A restrição dada pela Equação (14) define que apenas uma variável binária relacionada com a direção do fluxo de potência esteja ativa, enquanto a Equação (15) estabelece a natureza binária do estado de operação da chave de interconexão com o ramo mn . Por fim, juntamente com as restrições que garantem a radialidade do sistema, a Equação (16) certifica que cada barra na solução final seja ligada por somente um ramo. L e N representam os conjuntos de ramos e de barras (nós), respectivamente (BORGES; FRANCO; RIDER, 2013) (SILVEIRA, 2020).

O uso de variáveis binárias no problema é necessário para representar o estado de conexão dos ramos e também definir a direção do fluxo de potência ativa (y_{mn}^+ representa a direção mn (direta) e y_{nm}^- representa a direção nm (inversa)). Assim, se um das duas variáveis assumir o valor 1, isso significa que o ramo está ativo e a chave de manobra está fechada. Caso contrário, ambas as variáveis assumem o valor 0, isto significa que o ramo não está ativo, indicando que a chave de manobra está aberta. O modelo definido pelas equações (1) a (16) é uma formulação de PNLIM devido à restrição dada pela Equação (5). Essa restrição não linear determina a corrente nos ramos do sistema e é constituída pelo produto de duas variáveis (V_n^{sqr} e I_{mn}^{sqr}) e a soma dos quadrados do fluxo de potência ativa e reativa (P_{mn}^2 e Q_{mn}^2), respectivamente (SILVEIRA, 2020).

3.2 REPRESENTAÇÕES DE PROBLEMAS DE OTIMIZAÇÃO ATRAVÉS DE ALGORITMOS GENÉTICOS

Algoritmos genéticos são métodos de otimização inspirados na natureza que podem ser utilizados para muitos problemas de otimização combinatória. Estes métodos imitam os princípios básicos da vida e aplicam operadores genéticos como mutação, *crossover* ou seleção para obter uma sequência de alelos e também aplicam o conceito de sobrevivência do mais adaptado para encontrar soluções ótimas, ou próximas do ótimo, para problemas de otimização combinatória. Cada indivíduo tem um correspondente cromossomo que codifica a solução, e consiste de uma sequência de genes sendo que cada gene pode tomar um valor, denominado alelo. Este cromossomo está associado a um nível *fitness* que está correlacionado a um correspondente valor da função objetivo da solução que codifica (ROTHLAUF, 2006).

Os algoritmos genéticos envolvem um conjunto de indivíduos que constituem

uma população que se altera em um número de gerações. Em cada geração, uma nova população é criada pela combinação de elementos da população atual para produzir descendentes que fazem a próxima geração. Mutações aleatórias também tomam lugar como um meio de escapar de aprisionamento em mínimos locais. O conceito de sobrevivência do mais adaptado dentro do algoritmo genético ocorre durante o processo de seleção dos indivíduos para reprodução e produção de descendentes, quando os indivíduos são selecionados, aqueles com melhor valor *fitness* são preferidos sobre os de menor valor (ROTHLAUF, 2006).

Quando se usa esses algoritmos para propósitos de otimização, representações são exigidas para codificar potenciais soluções. Sem as representações, nenhum uso desses algoritmos é possível. Nas seções seguintes é introduzida a teoria da representação genética descrevendo a noção de genótipos e fenótipos, ilustrando como a função *fitness* pode ser decomposta dentro de um genótipo-fenótipo e no mapeamento *fitness*-fenótipo.

3.2.1 Genótipos e Fenótipos

Da teoria genética de Gregor Mendel, observou-se que a natureza distingue entre o código genético (genótipo) de um indivíduo e sua aparência externa (fenótipo). O genótipo representa todas as informações armazenadas nos cromossomos e nos permite descrever um indivíduo à nível de genes, todavia, o fenótipo descreve a aparência externa de um indivíduo. Para construir um fenótipo a partir de uma informação genotípica uma transformação deve ser executada, ou seja, um mapeamento ou representação genótipo-fenótipo é realizado a fim de se reproduzir as características externas do indivíduo.

Um cromossomo descreve uma sequência de alelos de certo comprimento onde todas as informações genéticas de um indivíduo são armazenadas. Embora a natureza frequentemente use mais do que um cromossomo, a maioria das aplicações dos algoritmos genéticos apenas usam um cromossomo para codificar a informação do genótipo, sendo que cada cromossomo consiste de vários alelos, que são as menores unidades de informação em um cromossomo. A aparência de um fenótipo de um indivíduo determina seu sucesso na vida. Desta forma, quando se compara as habilidades de diferentes indivíduos, deve-se julgá-los no nível do fenótipo. Entretanto, durante a reprodução deve-se observar indivíduos no nível do genótipo, porque durante a reprodução sexual, os descendentes não herdam as propriedades do fenótipo de seus pais, mas apenas uma informação genotípica relativa à propriedade fenotípica, e os descendentes herdam material genético de ambos os pais. Desta forma, operadores genéticos trabalham no nível do genótipo, enquanto a avaliação dos indivíduos é realizada no nível do fenótipo (ROTHLAUF, 2006).

3.2.2 Decomposição da função *fitness*

Todo o problema de otimização, que pode ser resolvido usando algoritmos genéticos, pode ser decomposto dentro de um mapeamento genótipo-fenótipo f_g e um mapeamento fenótipo-*fitness* f_p . Assim, cada função *fitness* atribui um valor ao genótipo x^g que pode ser decomposta dentro de um mapeamento genótipo-fenótipo f_g e, por conseguinte, no mapeamento fenótipo-*fitness* f_p . A introdução de uma representação explícita é necessária caso o fenótipo de um problema não seja acessível ao algoritmo genético. Além disso, a introdução de uma representação pode se tornar muito útil nos casos em que há restrições no espaço de busca, podendo ser modelado por uma codificação específica. Pode-se ainda usar os mesmos genótipos para diferentes tipos de problemas e apenas os interpretando diferentemente pelo uso do mapeamento genótipo-fenótipo, desta forma, permitindo o uso de operadores genéticos padrões sem grandes alterações nas propriedades (ROTHLAUF, 2006).

Define-se φ_g como o espaço de busca do genótipo onde os operadores genéticos, tais como, recombinação e mutação são aplicadas. Matematicamente, um problema de otimização formulado como mostra a equação (17).

$$\mathbf{x} = \min_{x \in \varphi_g} f(x) \quad (17)$$

Onde x é um vetor de variáveis de decisão (ou alelos), $f(x)$ é a função *fitness* (ou objetivo) e o vetor \mathbf{x} é o mínimo global. Pode ser construído como um espaço de busca φ_g associado a uma relação de transformação ao espaço \mathfrak{R} do problema, tal como mostra a equação (18).

$$f(x) : \varphi_g \rightarrow \mathfrak{R} \quad (18)$$

Desta forma a função *fitness* f é decomposta em duas partes. A primeira mapeia o espaço genótipo φ_g para o espaço fenótipo φ_p e o segundo mapeia φ_p para o espaço *fitness* \mathfrak{R} (ROTHLAUF, 2006).

$$f_g(x^g) = \varphi_g \rightarrow \varphi_p \quad (19)$$

$$f_p(x^p) = \varphi_p \rightarrow \mathfrak{R} \quad (20)$$

Assim, o mapeamento fenótipo-genótipo f_g é determinado pelo tipo de representação usada, f_p representa a função *fitness* e atribui um valor de *fitness* $f_p(x^p)$ para cada indivíduo $x^p \in \varphi_p$, e os operadores genéticos são aplicados aos genótipos $x^g \in \varphi_g$. Desse modo, pode-se obter uma decomposição dentro de um mapeamento genótipo-fenótipo f_g e um fenótipo-*fitness* f_p .

3.3 ALGORITMO GENÉTICO DE CHAVES ALEATÓRIAS VICIADAS

O *Biased Random Key Genetic Algorithm* (BRKGA) é uma metaheurística inspirada na natureza desenvolvida por (GONÇALVES; RESENDE, M. G. C., 2011) e é derivada do *Random Key Genetic Algorithm* (RKGA), introduzido por Bean (1994). Tanto no método RKGA quanto BRKGA são geradas populações de p indivíduos, sendo cada indivíduo representado por um vetor de n elementos, denominados cromossomos. Em outras palavras, cada cromossomo representa um vetor contendo n números reais no intervalo contínuo $[0, 1]$, que representam os alelos dentro do cromossomo. Esta representação do cromossomo é denominada *Network Random Keys*, ou simplesmente *Random Keys*, onde cada alelo contido nestes vetores representa, no espaço fenótipo, um peso para cada arco da rede no grafo ponderado para a geração da árvore geradora mínima ou, ainda, uma posição para encontrar as chaves abertas dentro de um determinado percurso fechado, será retornado a esse assunto na seção sobre os decodificadores.

Ambos os métodos, RKGA e BRKGA, buscam por soluções dos problemas no módulo do hiper-cubo contínuo n -dimensional e não diretamente no espaço de busca do problema, Figura 3. Desta forma, esses algoritmos podem ser separados em módulos independente e dependente do problema sem afetar a solução. Isso possibilita projetar um algoritmo de propósito geral que pode ser reutilizado para implementar diferentes heurísticas. Deste modo, quando projetada uma nova heurística para um problema de otimização combinatória específico, necessita-se apenas implementar a parte dependente do problema, denominada decodificador, cuja função é mapear as soluções no hiper-cubo para o espaço de busca do problema. A porção independente do problema não tem conhecimento do problema sendo resolvido, pois é limitado a busca no hiper-cubo. A única conexão para o problema de otimização combinatória sendo resolvido é a porção dependente do problema do algoritmo, onde tomam como entrada os alelos dos cromossomos e os associam com uma solução de um problema de otimização combinatória para que um valor objetivo (ou *fitness*) possa ser calculado (GONÇALVES; RESENDE, M. G. C., 2011) (MARTÍ; PARDALOS; RESENDE, M. G. C., 2018).

Assim, o hiper-cubo é um universo de n elementos, definidos sob o intervalo $(0,1)$, é uma extrapolação do cubo ou quadrado para n dimensões. Em outras palavras, um hiper-cubo é uma cadeia de valores reais no intervalo $(0,1)$ de comprimento n , são interpretadas como vetores em um hiper-cubo n -dimensional. Fazendo uma analogia com o cubo, cada coordenada no espaço tridimensional dentro deste cubo é definido por 3 variáveis x , y e z , sendo cada uma destas coordenadas limitadas a um valor no intervalo $(0,1)$. Em relação ao problema da reconfiguração, cada ponto obtido destas coordenadas representa uma topologia da rede elétrica no espaço solução.

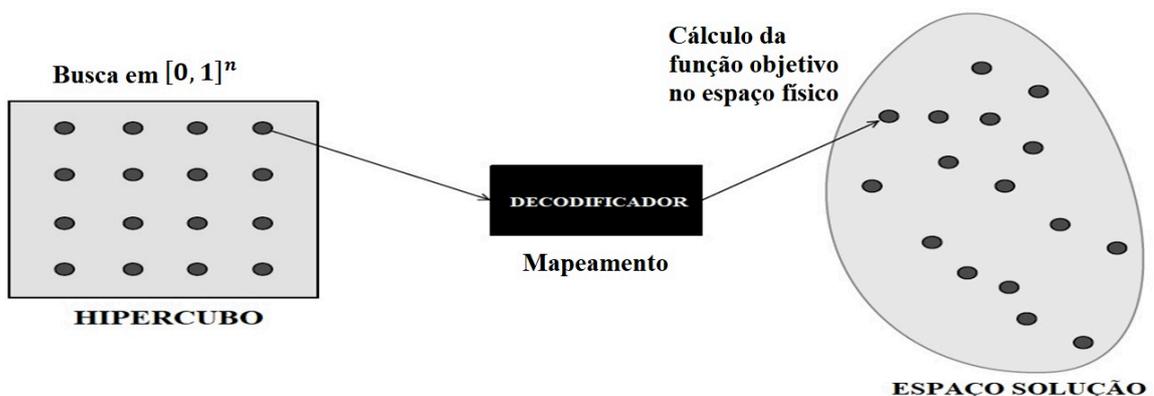
A representação por *Random Keys* codifica uma solução com números alea-

tórios e esses valores são usados como chaves de ordenação para decodificar uma solução. *Random Keys* eliminam o problema da factibilidade dos descendentes pelo uso de codificações cromossômicas que representam soluções de uma maneira suave. Essas codificações são interpretadas na rotina de avaliação da função objetivo de um modo que evita o problema da factibilidade (ANDRADE *et al.*, 2021).

A diferença primária entre essa codificação e a das demais literaturas é o uso dos números aleatórios como identificadores das soluções representadas. Números aleatórios são amostrados do mesmo espaço, tipicamente $[0,1]^n$. O algoritmo genético busca nesse espaço como um substituto para o espaço literal. Pontos no espaço de chaves aleatórias $[0,1]^n$ são mapeados para pontos no espaço literal (físico) por uma função de avaliação (decodificador), essa função decodificadora $f : [0,1]^n \rightarrow S$ é necessária para mapear indivíduos do espaço do BRKGA para o espaço do problema S, a Figura 3 representa esse processo (ANDRADE *et al.*, 2021).

Uma vez que todos os operadores evolucionários são aplicados no espaço do BRKGA (hipercubo), operadores personalizados baseados na estrutura do problema são desnecessários. Por essa razão, a abordagem de *Random Keys* não é similar a codificação binária. A geração de números aleatórios no espaço das chaves emprega um sentido de busca aleatória na associação com o algoritmo genético, uma busca não tão aleatória ocorre na codificação binária. Isso permite prototipagem e testes rápidos, reduzindo assim os custos de desenvolvimento. Um aspecto crucial do decodificador é que deve ser uma função determinística, ou seja, dado um cromossomo, o decodificador deve sempre gerar a mesma solução. Este pré-requisito garante a reprodutibilidade dos resultados sem proibir o uso de sinais aleatórios por decodificadores. Assim, uma vantagem dessa codificação é a robustez e precisão na estrutura do problema (ANDRADE *et al.*, 2021).

Figura 3 – Decodificador usado para mapear soluções no hiper-cubo para soluções no espaço solução onde o *fitness* é calculado.



O RKGA e o BRKGA diferem apenas no modo como selecionam os pais para reprodução (*crossover*), pois, a variante não viciada (RKGA) seleciona ambos os pais da população inteira, enquanto na variante viciada (BRKGA) um progenitor é sempre escolhido do conjunto elite, que introduz o princípio do elitismo no processo de reprodução. Essa modificação é suficiente para fazer a versão viciada do algoritmo genético superar a versão não viciada. Isso provavelmente é mais enfatizado através do *crossover uniforme parametrizado* usado em ambos os métodos para combinar os pais e produzir descendentes. Essa diferença, quase sempre conduz a uma grande diferença na performance dessas variantes, em geral, um BRKGA tende a encontrar melhores soluções do que RKGA dado o mesmo tempo de execução, além disso, possui uma probabilidade muito maior de encontrar uma solução em menor tempo (GONÇALVES; RESENDE, M. G. C., 2011) (MARTÍ; PARDALOS; RESENDE, M. G. C., 2018).

Desse modo, a principal vantagem do BRKGA sobre outras variantes do algoritmo genético é talvez as soluções de alta qualidade obtidas rapidamente. Isso é principalmente apoiado pela adoção do duplo elitismo quando se gera descendentes através da reprodução. Inicialmente, um progenitor é sorteado do conjunto elite que mantém os melhores cromossomos encontrados até então, em seguida os genes dos descendentes são herdados do progenitor elite com alta probabilidade. De uma perspectiva prática, o BRKGA permite rápida prototipagem, uma vez que o espaço solução do BRKGA e o espaço solução do problema são mantidos separados (ANDRADE *et al.*, 2021).

3.3.1 Parte Independente do Problema: Estrutura geral do BRKGA

O algoritmo inicia com a definição do número de genes (n) do cromossomo de um indivíduo, o número de indivíduos na população (p), o número de indivíduos elite na população (p_e), o número de mutantes (p_m) introduzidos em cada geração dentro da população, e a probabilidade de que os descendentes herdem o alelo de seu progenitor elite (ρ_e). Esses parâmetros de configuração estão relacionados entre si conforme ilustra o Quadro 2 (GONÇALVES; RESENDE, M. G. C., 2011).

Quadro 2 – Valores de parâmetros recomendados para o BRKGA.

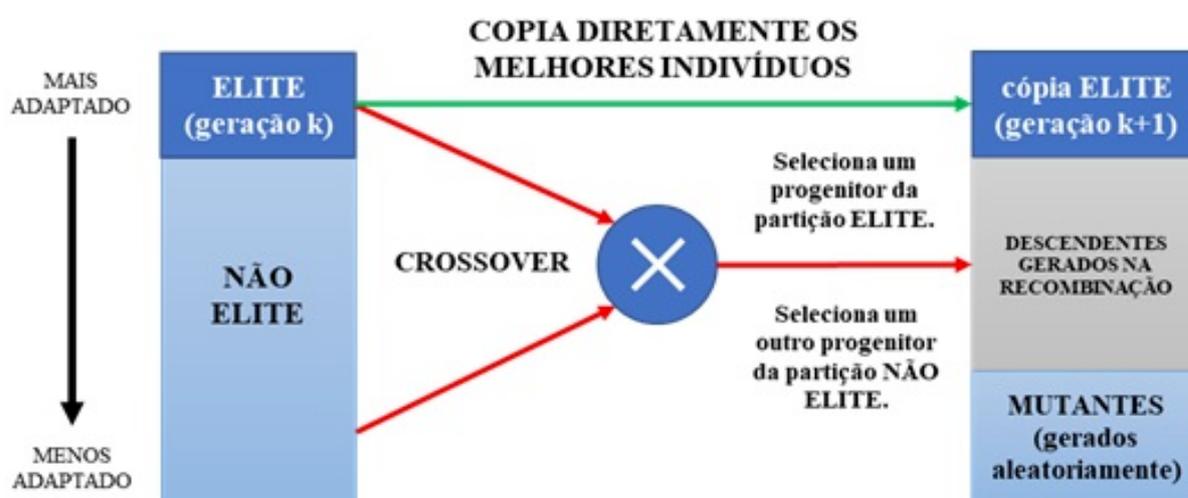
Parâmetro	Restrição	Valor Recomendado
p	Tamanho da população	$p = a.n$, onde $a \geq 1 \in \mathbb{R}$ é uma constante e n é o comprimento do cromossomo.
p_e	Tamanho da população elite	$0,10p \leq p_e \leq 0,25p$
p_m	Tamanho da população mutante	$0,10p \leq p_m \leq 0,30p$
ρ_e	Probabilidade de herdar alelo elite	$0,5 < \rho_e \leq 0,8$

Fonte: (GONÇALVES; RESENDE, M. G. C., 2011)

Para criar a população inicial, sorteios uniformes são realizados para gerar os indivíduos da primeira população, que será envolvida pelo processo evolutivo do BRKGA sobre um certo número de iterações, denominada gerações. Cada geração dessa população será decodificada para que o *fitness* de cada indivíduo possa ser calculado. Depois do *fitness* de cada indivíduo ser calculado pelo decodificador, estes indivíduos serão ordenados por esses valores, os melhores p_e indivíduos mais adaptados são alocados na partição rotulada como ELITE e os indivíduos remanescentes ($p - p_e$) são alocados na partição rotulada como NÃO ELITE (GONÇALVES; RESENDE, M. G. C., 2011) (MARTÍ; PARDALOS; RESENDE, M. G. C., 2018).

Para construir a população da próxima geração, o conjunto p_e (ELITE) da geração (k) atual é copiado, sem mudança, para a próxima geração. Em seguida, o algoritmo insere p_m mutantes dentro da próxima população e os alocam na partição denominada MUTANTES. O restante da população é completado com ($p - p_e - p_m$) cromossomos filhos usando o *crossover uniforme parametrizado*, por conseguinte, estes novos indivíduos da geração ($k + 1$) têm seus *fitness* calculados pelo decodificador e são novamente ordenados, separando em novas partições ELITE e NÃO ELITE, este processo é repetido até que um critério de parada seja alcançado. A mutação no BRKGA é feita através da introdução de indivíduos mutantes dentro da população. Um elemento mutante é simplesmente um vetor de *random keys* gerado da mesma maneira que a população inicial. Assim, em cada geração um pequeno número p_m de mutantes são introduzidos dentro da população corrente, a Figura 4 ilustra esse procedimento (GONÇALVES; RESENDE, M. G. C., 2011) (MARTÍ; PARDALOS; RESENDE, M. G. C., 2018).

Figura 4 – Transição da geração k para a geração $k + 1$ no BRKGA.



Fonte: Adaptado de (GONÇALVES; RESENDE, M. G. C., 2011)

Com o vetor p_e de indivíduos elite e p_m de indivíduos mutantes contabilizados para dentro da população $k+1$, $(p-p_e-p_m)$ indivíduos adicionais devem ser produzidos para completar os p indivíduos que irão fazer parte da população da geração $k+1$, isto é feito produzindo $p-p_e-p_m$ descendentes através do processo de cruzamento ou reprodução dos indivíduos. No BRKGA, cada elemento é gerado combinando um progenitor selecionado de forma aleatória da partição ELITE e o outro progenitor selecionado da partição NÃO ELITE. Um mesmo indivíduo pode produzir mais do que um descendente, pois, a repetição na seleção do par para a reprodução é permitida. (GONÇALVES; RESENDE, M. G. C., 2011) (MARTÍ; PARDALOS; RESENDE, M. G. C., 2018)

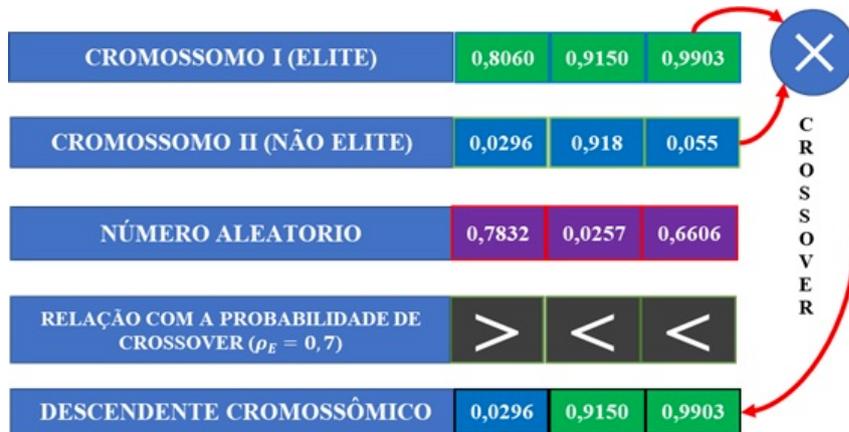
No *crossover uniforme parametrizado* proposto por Spears e Jong (1991), primeiramente, dois sorteios uniformes são realizados, o primeiro sorteio é realizado para a escolha de um progenitor elite, entre todos os elementos da partição ELITE, e segundo sorteio uniforme é realizado para a escolha de um outro progenitor, escolhido na partição NÃO ELITE. Depois de escolhidos os pais, n sorteios uniformes são realizados para definir se cada alelo que irá compor o cromossomo filho virá do progenitor elite ou do não elite, sendo n o número de alelos dos cromossomos. Desse modo, cada valor obtido no sorteio será comparado com um parâmetro ρ_e (probabilidade de o filho herdar um gene do progenitor elite), se este valor randômico for menor do que ρ_e o filho irá herdar o alelo do progenitor elite, caso contrário, irá herdar do não elite.

Em outras palavras, seja n o número de genes no cromossomo de um indivíduo. Para $i=1,2,\dots,n$, o i -ésimo alelo $c(i)$ do descendente c , recebe o valor do i -ésimo alelo $e(i)$ do progenitor elite e , com probabilidade ρ_e , e o valor do i -ésimo alelo $e(i)$ de um progenitor não elite e , com probabilidade $1-\rho_e$, desse modo, o descendente é mais provável de herdar características do progenitor elite do que aquele não elite, para os valores de ρ_e do Quadro 2.

A Figura 5 ilustra o procedimento de *crossover* para dois vetores de cromossomos com três genes cada. O cromossomo I refere-se ao indivíduo elite e o cromossomo II ao não-elite. No exemplo abaixo, a probabilidade de o descendente herdar um alelo do indivíduo elite é $\rho_e = 0,7$, isto é, em um sorteio uniforme no intervalo $[0, 1]$ os descendentes herdam o alelo de um progenitor elite caso o resultado seja abaixo da probabilidade 0,7 e, caso contrário, seja superior ao valor 0,7, o descendente herdará o alelo do indivíduo não elite. Como ilustra a Figura 5, o descendente herda o alelo do cromossomo I (elite) no segundo e terceiro genes. Assim, o descendente se assemelha mais ao progenitor elite do que o não elite (GONÇALVES; RESENDE, M. G. C., 2011).

Quando a população $k+1$ está completa, isto é, quando os p indivíduos estão todos gerados, os valores *fitness* são calculados para todos os recém-criados vetores de cromossomos e, em seguida, a nova população será particionada novamente dentro dos conjuntos de indivíduos elite e não-elite para iniciar a próxima população, e assim

Figura 5 – Crossover uniforme parametrizado no BRKGA2 para a rede 14 barras.



Fonte: Elaborado pelo autor.

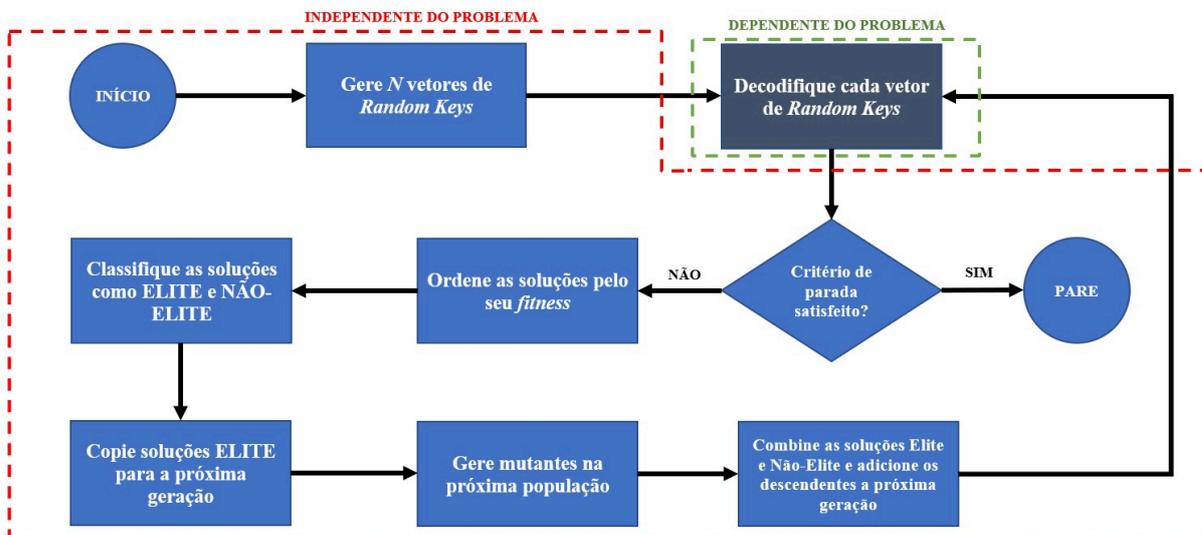
o processo é repetido até que um critério de parada seja estabelecido. No Quadro 3 e na Figura 6 estão ilustrados o pseudocódigo e o fluxograma para a construção do BRKGA.

Quadro 3 – Pseudocódigo do BRKGA.

Parâmetros de entrada do BRKGA ($ P $, $ P_e $, $ P_m $, p , ρ_e)
Inicialize o valor da melhor solução encontrada: $f^* \leftarrow \infty$;
Gere uma população P com p vetores de <i>random keys</i> , cada uma de tamanho n ;
ENQUANTO critério de parada; Faça
Decodifique e avalie o custo de cada nova população em P ;
Particione P dentro de dois conjuntos: P_e e $P_{\bar{e}}$;
Encontre a melhor solução x^+ em P ;
$x^+ \leftarrow \operatorname{argmin}(f(x) x \in P)$
SE $f(x^+) < f^*$ ENTÃO
$x^* \leftarrow x^+$;
$f^* \leftarrow f(x^+)$;
FIM(SE)
Inicialize a população da próxima geração: $P^+ \leftarrow P_e$;
Gere um conjunto P_m de mutantes, cada mutação com n chaves aleatórias;
Adicione P_m a população da próxima geração: $P^+ \leftarrow P^+ \cup P_m$
PARA CADA $i \leftarrow 1$ até $ P - P_e - P_m $ FAÇA
Selecione o progenitor “a” aleatoriamente de P_e ;
Selecione o progenitor “b” aleatoriamente de $P_{\bar{e}}$;
PARA CADA $j \leftarrow 1$ até n FAÇA
Lance uma moeda viciada com probabilidade $\rho_e > 0.7$ do resultado dar cara;
SE cara então $c[j] \leftarrow a[j]$;
CASO CONTRÁRIO $c[j] \leftarrow b[j]$;
FIM(SE)
Adicione os descendentes c à população da próxima geração: $P^+ \leftarrow P^+ \cup c$
FAÇA
Atualize a população: $P \leftarrow P^+$
FIM(ENQUANTO)
RETORNE x^*

Fonte: (GONÇALVES; RESENDE, M. G. C., 2011)(MARTÍ; PARDALOS; RESENDE, M. G. C., 2018)

Figura 6 – Fluxograma do BRKGA.



Fonte: (GONÇALVES; RESENDE, M. G. C., 2011)

3.3.2 Parte Dependente do Problema: Decodificadores

Depois de ser decodificado, o *fitness* da função deve ser calculado para que uma dada estrutura possa usar essa informação no processo de busca. O BRKGA permite a decodificação das soluções codificadas para resultar nas soluções factíveis exclusivas, sem necessidade de alteração nos operadores do algoritmo devido a independência do BRKGA e do decodificador (FARIA; RESENDE, M. G.; ERNST, 2017).

Segundo Rothlauf (2006) a codificação baseada em *random keys* pertence à classe de codificações ponderadas. Em contraste com outras representações, tais como, codificação de Vetores Característicos que podem apenas indicar se um laço é estabelecido ou não, codificações ponderadas usam pesos para o genótipo e podem assim codificar a importância de arcos de um grafo, por exemplo. Consequentemente, uma construção adicional do algoritmo é necessária para que se construa uma árvore validando os pesos do genótipo.

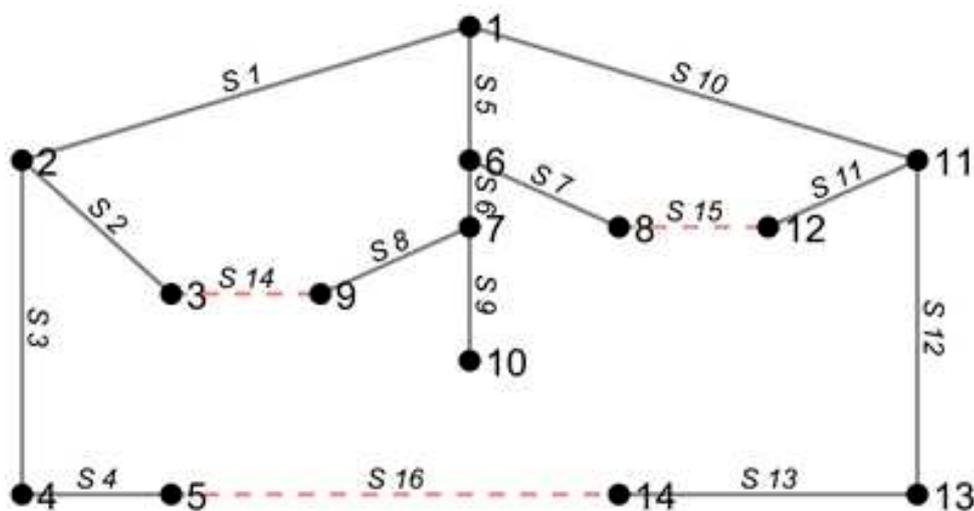
Assim, *random keys* armazenam a importância dos arcos como vetores de pesos e um algoritmo de construção desenvolve uma árvore a partir dessa sequência de valores, tais como, o algoritmo de Kruskal e o algoritmo baseado em *links* da co-árvore utilizados neste trabalho. Os decodificadores fazem a conexão entre as soluções no hiper-cubo e o *fitness* de suas correspondentes soluções no espaço solução do problema. Métodos de codificação de topologias de redes de distribuição são fundamentais para encontrar a solução ótima do algoritmo, por um lado, uma estratégia complexa pode aumentar consideravelmente o tempo execução e, por outro, uma estratégia simples não permite uma efetiva exploração do campo de busca.

3.3.2.1 Algoritmo de Kruskal

Um grafo é uma estrutura de abstração bastante útil na representação e solução de diversos tipos de problemas, também possui uma representação gráfica bastante confortável. Além disso, um grafo formaliza relações de interdependência existente entre os elementos de um conjunto como, por exemplo, os elementos da rede elétrica. Para descrever a estrutura geométrica da rede elétrica basta substituir os componentes da rede por segmentos de linha, independente das características desses componentes. As relações entre os elementos da rede elétrica serão caracterizadas por traços denominados arestas (ou arcos), enquanto seus terminais serão caracterizados por nós (ou vértices), sendo que este nó pode ser incidente a um ou mais elementos (STAGG; EL-ABIAD, 1968) (GOLDBARG, M. C.; GOLDBARG, E., 2012).

Um exemplo desta representação é a rede teste de 14 nós (barras), mostrada na Figura 7. Nesta figura, 'S' representa a palavra *switch*, as linhas tracejadas em vermelho representam as chaves de laço (chaves abertas), as demais são as chaves seccionadoras da rede (chaves fechadas).

Figura 7 – Grafo de 14 nós obtido do sistema teste de 14 barras.



Fonte: Elaborado pelo autor.

Matematicamente, pode-se definir um grafo G como uma tripla ordenada consistindo de um conjunto não vazio de nós (ou vértices) $N(G)$, um conjunto de arestas $E(G)$, e uma relação que associa cada aresta a um par de nós de G , não necessariamente distintos. Um grafo $H(W,F)$ é um subgrafo de um grafo $G(N,E)$, se W é um subconjunto de N e F é um subconjunto de E , em outras palavras, um subgrafo é qualquer subconjunto de elementos de um grafo. Um grafo pode ainda conter informações associadas tanto a seus vértices quanto às suas arestas, algumas dessas informações (numéricas ou alfabéticas) podem ser escritas junto aos seus elementos associados,

um caso especial destas informações numéricas é o caso de grafos ponderados que são grafos com valores numéricos (pesos) associados às suas arestas ou vértices (GOLDBARG, M. C.; GOLDBARG, E., 2012) (BALAKRISHNAN, 1997).

Pode-se definir ainda o conceito de árvore que é um grafo conexo T em que existe um, e somente um, caminho entre qualquer par de vértices de T . Um caminho é um subgrafo de elementos conectados com não mais do que dois elementos conectados a qualquer nó e, por fim, um grafo é conectado se, e apenas se, há um caminho entre todo par de nós. Em outras palavras, um subgrafo conectado e que contém todos os nós de um grafo, mas sem caminhos fechados, é denominado de árvore, ou seja, uma árvore é um grafo acíclico conectado (GOLDBARG, M. C.; GOLDBARG, E., 2012).

Algumas propriedades das árvores: todo grafo G conexo possui pelo menos uma subárvore que contém todos os seus vértices; toda árvore é um grafo planar R^2 ; toda árvore finita com $E > 1$ possui pelo menos dois vértices terminais; as arestas de uma árvore são chamadas de ramos e formam um subconjunto dos elementos do grafo conectado. A partir dessas propriedades, pode-se definir que o número de ramos exigidos para formar uma árvore é dado pela equação (21) (STAGG; EL-ABIAD, 1968).

$$b = n - 1 \quad (21)$$

Onde n é o número de nós em um grafo e b o número de ramos. Os elementos de um grafo conectado que não estão incluídos na árvore são chamados de *links* (elos) e formam um subgrafo, não necessariamente conectado, denominado co-árvore. A co-árvore é o complemento da árvore. O número de *links* l de um grafo conectado com e arestas é:

$$l = e - n + 1 \quad (22)$$

Se um *link* é adicionado a árvore, o grafo resultante contém um caminho fechado, denominado *loop*, se este contém apenas um *link*, então, são independentes e formam *loops* básicos (ou ciclos fundamentais) do grafo G . Consequentemente, o número de *loops* básicos é igual ao número de *links*, calculados pela equação (22).

Uma árvore geradora mínima T_{min} é uma árvore geradora de menor custo, dentre todas as possíveis do grafo G . O custo de uma árvore geradora T de um grafo ponderado G é dado pelo somatório dos custos das arestas de T . Existem vários algoritmos capazes de formar uma árvore geradora como mostrado em (GOLDBARG, M. C.; GOLDBARG, E., 2012), para esta dissertação será utilizado o algoritmo de Kruskal. O algoritmo foi proposto por Joseph B. Kruskal (1956) e seu raciocínio é voltado para a formação da árvore através de inclusões das arestas de menor peso do grafo ponderado, no Quadro 4 pode-se ilustrar o pseudocódigo utilizado para criar uma árvore de mínima expansão a partir do algoritmo de Kruskal.

Quadro 4 – Pseudocódigo do algoritmo de Kruskal.

<p>Ler $G(V,E)$ e $P = [p_{ij}]$, onde P representa uma matriz de pesos de G.</p> <p>Ordene as arestas em ordem não-crescente de pesos p_{ij} no vetor $H = [h_i]$</p> <p>$T \leftarrow h_1$</p> <p>$i \leftarrow 2$</p> <p>ENQUANTO $T < n$ Faça</p> <p>Se $T \cup h_i$ é um grafo acíclico, então</p> <p>$i \leftarrow i + 1$</p> <p>FIM (se)</p> <p>FIM (enquanto)</p> <p>Escrever T arestas da árvore geradora mínima</p>
--

Fonte: (BALAKRISHNAN, 1997)(GOLDBARG, M. C.; GOLDBARG, E., 2012)

Desta forma, a árvore que se forma é, de fato, uma árvore geradora mínima. Onde H representa o vetor ordenado dos pesos das arestas, T é a árvore em formação, n é o número de arestas a serem preenchidas em T e h_i é um elemento de H , ou ainda, h_i é um peso de G .

Assim, o algoritmo de Kruskal gera árvores de peso mínimo a partir de cada indivíduo da população, ou seja, cada genótipo da sequência de alelos do cromossomo gera uma árvore de mínimo peso (fenótipo) associada a esse cromossomo. De outro modo, o algoritmo de Kruskal decodifica um vetor de n *random keys* para obter uma árvore associada ao grafo da rede de distribuição. O uso deste algoritmo na decodificação garante que cada cromossomo seja sempre capaz de ser associado a uma configuração radial para a rede de distribuição. A entrada deste algoritmo é a sequência de alelos do cromossomo, que representam um conjunto de pesos das arestas de um grafo ponderado e a saída é um conjunto de arestas de uma árvore geradora mínima T da rede de distribuição.

3.3.2.2 Algoritmo baseado na co-árvore

Em Swarnkar, Gupta e Niazi (2011), foi proposto um outro decodificador que pode ser utilizado em problemas de reconfiguração de redes de distribuição e que pode ser usado em conjunto com qualquer técnica metaheurística. Como o algoritmo de Kruskal, essa técnica também se baseia na teoria de grafos, onde a rede de distribuição é vista como um grafo $G(N,E)$, e usa um conjunto de regras para corrigir indivíduos não factíveis e, assim, gerar apenas vetores de indivíduos factíveis durante a decodificação.

Nesta técnica os elementos usados para representar a solução do problema de reconfiguração são os *links* da co-árvore, isto é, as chaves normalmente abertas da rede elétrica. O vetor solução representa uma co-árvore que deve ter uma correspondente árvore que é factível, ou seja, uma vez que tenhamos obtido os *links* da co-árvore, podemos obter facilmente a correspondente árvore do grafo da rede elétrica. Com o uso de *links* da co-árvore da rede de distribuição para representar uma solução, através de vetores de *loop*, o espaço de busca da solução do problema é reduzido dras-

ticamente. Assim, se a representação usando vetores de *loop* é usada, o comprimento de cada cromossomo é reduzido do número total de chaves da rede elétrica, como usado no algoritmo de Kruskal, para o número total de *loops* fundamentais. Na rede 14 barras, por exemplo, um vetor cromossomo irá possuir 16 elementos (número de ramos da rede) no algoritmo de Kruskal, porém terá 3 elementos (número de chaves abertas ou *links* da co-árvore) no vetor cromossomo usando o segundo decodificador, o número de *links* de um grafo de rede é calculado pela equação (22).

Um *locus* gênico (posição onde o gene está alocado no vetor cromossomo) corresponde a um vetor de *loop* fundamental e o valor do alelo é a chave a ser aberta contida nesse *loop* fundamental. Como manter a radialidade é o principal obstáculo nos problemas de reconfiguração de redes de distribuição, as regras desenvolvidas têm como principal motivação eliminar quaisquer configurações que não satisfaçam a radialidade da rede de distribuição. Uma vez que a restrição de radialidade é satisfeita, o fluxo de carga utilizado na reconfiguração da rede pode ser executado. Portanto, essas regras garantem a produção de indivíduos com topologias radiais factíveis, sem envolver constantes checagens de malha. Para apresentar estas regras, que compõe a decodificação, é necessário definir alguns termos (SWARNKAR; GUPTA; NIAZI, 2011):

- **Nó Principal:** A junção de três ou mais elementos do grafo de rede de distribuição;
- **Nó Exterior:** O nó localizado no perímetro do grafo de rede de distribuição;
- **Nó Interior:** O nó localizado dentro do perímetro do grafo da rede de distribuição;
- **Vetor de Loop:** O conjunto de elementos constituindo um caminho fechado em um grafo de rede de distribuição. Esse caminho fechado não pode conter em seu interior outro caminho fechado;
- **Vetor de Ramos Comuns:** O conjunto de elementos que são comuns entre quaisquer dois vetores de *loop* de um grafo de rede de distribuição;
- **Vetor de Grupo Proibido:** O conjunto de vetores de ramos comuns. De cada um deles, se um está aberto, então um ou mais nós interiores do grafo de redes de distribuição estará isolado. O tamanho do vetor de grupos proibidos não pode ser maior do que l , número de *loops* fundamentais. Em outras palavras, é um conjunto com todos os vetores de ramos comuns que podem tornar um ou mais nós isolados.

Para ilustrar estes conceitos, toma-se novamente como exemplo a rede teste de 14 nós, mostrada na Figura 8. Fechando todas as chaves dessa rede podemos obter três percursos fechados no grafo da rede (L1, L2 e L3), esses formam três vetores de *loops* fundamentais, ou básicos, que contém todas as chaves dentro de seus respectivos caminhos fechados e que não contém em seu interior outro caminho

fechado, definidos como $L_1=\{S1, S2, S5, S6, S8, S14\}$, $L_2=\{S5, S7, S10, S11, S15\}$ e $L_3=\{S2, S3, S4, S6, S7, S8, S11, S12, S13, S14, S15, S16\}$. O vetor de ramos comuns desta mesma rede é formado pelas chaves que são comuns a dois *loops* fundamentais, como ilustram os vetores $C_{(L_1-L_2)}=\{S5\}$, $C_{(L_1-L_3)}=\{S2, S6, S8, S14\}$ e $C_{(L_2-L_3)}=\{S7, S11, S15\}$.

Por fim, o vetor de grupo proibido para esta rede é definido como $P=\{C_{(L_1-L_2)}, C_{(L_1-L_3)}, C_{(L_2-L_3)}\}$, pois possui apenas três chaves abertas, ou *loops* fundamentais, $l = 3$, e não há quaisquer outras possíveis combinações de vetores comuns que gerem outros vetores de ramos proibidos. Assim, caso a rede teste tenha como chaves abertas os ramos S5, S2 e S11, então a rede tem os nós 3, 6, 7, 8, 9, 10 e 12 isolados. Seja, L_j , C_k e P_h que denotam os vetores de *loop*, vetores de ramos comuns e os vetores de grupos proibidos, respectivamente, para uma dada topologia da rede de distribuição. Para um indivíduo ser factível, ou seja, forneça uma topologia radial sem ilhamento de qualquer nó, ou grupo de nós, da rede de distribuição, as seguintes regras são analisadas para selecionar um conjunto de chaves candidatas para abertura (SWARNKAR; GUPTA; NIAZI, 2011):

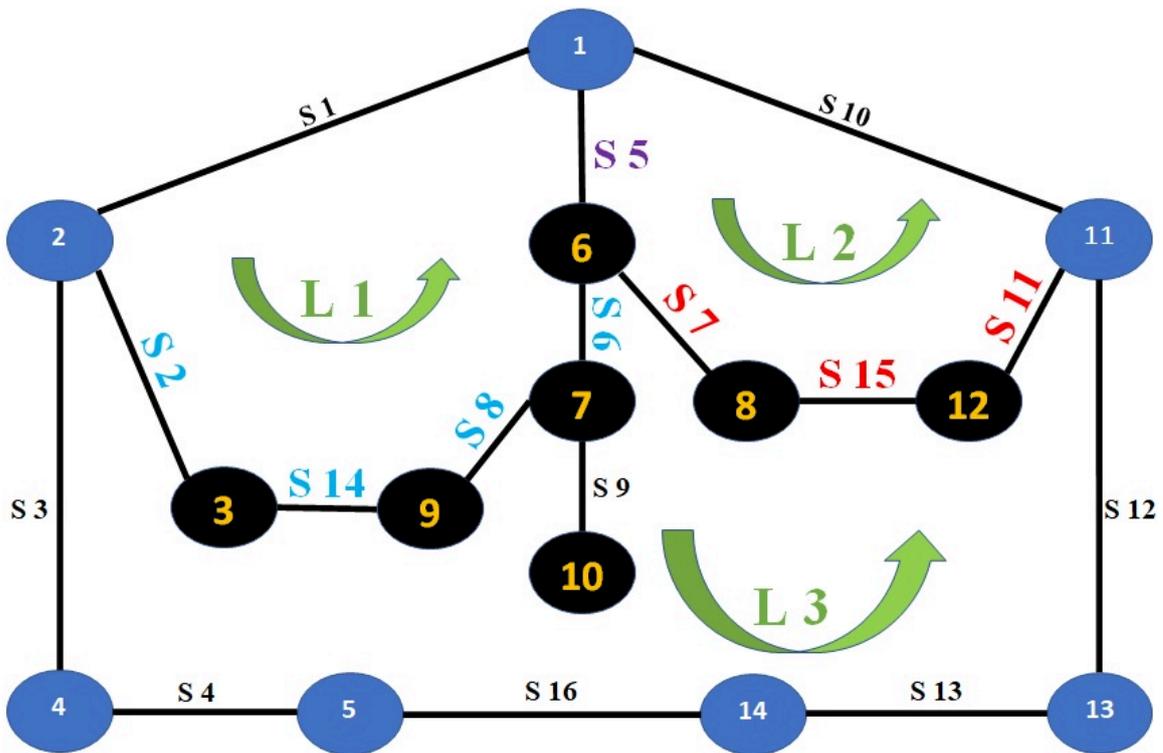
- **Regra 1:** Cada chave candidata deve pertencer ao seu correspondente vetor de *loop*. Assim, para uma decodificação de indivíduos, cada cromossomo candidato é formado por um conjunto de chaves a serem abertas nos correspondentes percursos fechados, através de números reais no intervalo [0,1] definem uma chave através da sua posição no seu correspondente vetor de *loop*, ou seja, cada elemento no vetor cromossomo candidato será associado a uma chave a ser aberta no seu respectivo *loop* básico.
- **Regra 2:** Apenas uma chave candidata pode ser selecionada de um *vetor de ramos comuns*.
- **Regra 3:** Todos os *vetores de ramos comuns* do *vetor de grupos proibidos* não podem participar simultaneamente para formar um indivíduo.

Desta forma, a regra 1 previne o ilhamento de nós exteriores, a regra 2 previne o ilhamento de nós interiores e a regra 3 previne o ilhamento de nós principais da rede de distribuição. A regra 1 é um procedimento que recebe um vetor de *random keys* como entrada e decodifica uma solução, tal que o primeiro alelo pertence ao primeiro vetor de *loop*, o segundo alelo pertence ao segundo vetor de *loop* e assim por diante. As figuras 9, 10 e 11 mostram como é feito o processo de decodificação para a regra 1 na rede teste de 14 barras. As regras 2 e 3 são procedimentos que analisam o vetor decodificador produzido pela regra 1 para checar por violações. Se uma violação é detectada, um substituto para o valor do primeiro alelo que não está obedecendo a regra correspondente é procurado no correspondente vetor de *loop*. Se a violação não é eliminada, o segundo alelo de seu correspondente vetor de *loop* é usado para a

correção do processo e, assim por diante, até que essas violações sejam corrigidas, a aplicação dessas regras é *sequencial*.

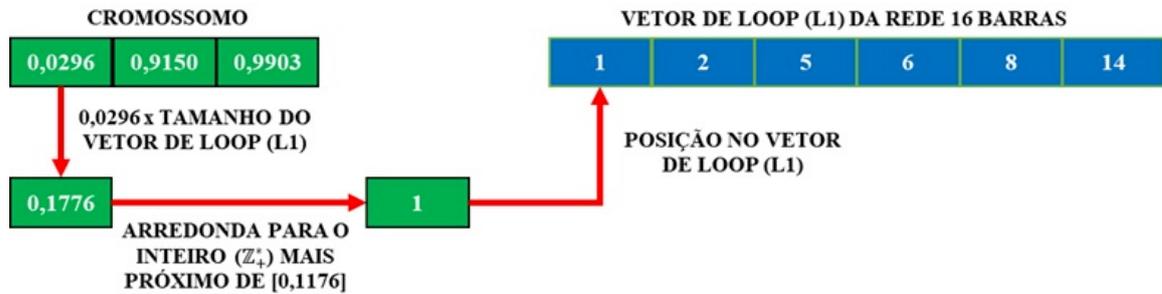
Para ilustrar estas regras, toma-se como exemplo novamente a rede 14 barras, onde o cromossomo candidato gerou as chaves abertas S1, S5 e S16. Na Figura 9, o valor [0,1776] é arredondado para 1 e não para 0, isto depende da linguagem de programação adotada, pois um vetor pode começar pela posição 0, como na linguagem C, ou pela posição 1, como na linguagem MATLAB (MATrix LABoratory). O Quadro 5 apresenta o pseudocódigo desta codificação. Para simplificação do texto os termos BRKGA1 e BRKGA2 serão utilizados daqui em diante para especificar os algoritmos genéticos de chaves aleatórias viciadas com decodificador baseado no algoritmo de Kruskal e decodificador baseado nos *links* da co-árvore da rede, respectivamente.

Figura 8 – Exemplo para ilustrar segundo decodificador utilizando a rede teste 14 nós.



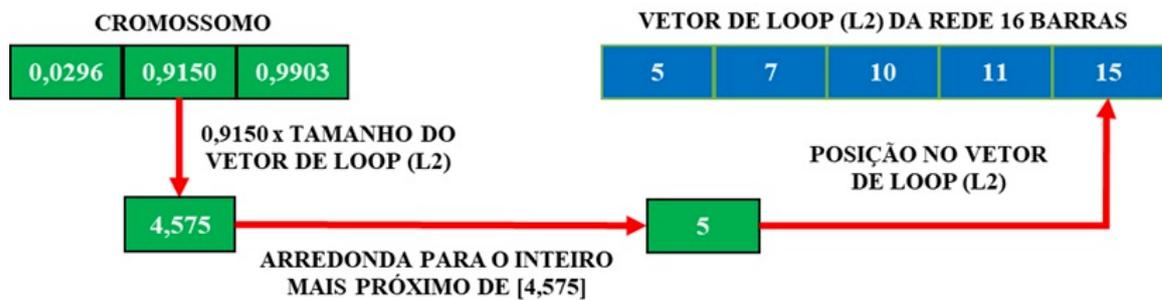
Fonte: Elaborado pelo autor.

Figura 9 – Decodificação do primeiro caminho fechado para a regra 1 na rede 14 barras.



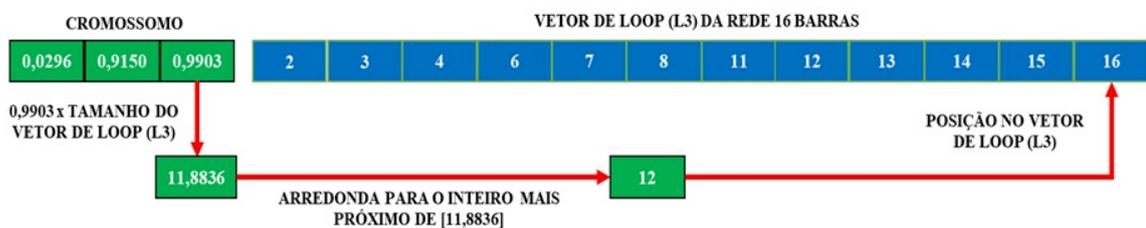
Fonte: Elaborado pelo autor.

Figura 10 – Decodificação do segundo caminho fechado para a regra 1 na rede 14 barras.



Fonte: Elaborado pelo autor.

Figura 11 – Decodificação do terceiro caminho fechado para a regra 1 na rede 14 barras.



Fonte: Elaborado pelo autor.

Quadro 5 – Pseudocódigo da codificação baseada na co-árvore.

<p>ENTRADA: Vetor de <i>random keys</i>, variáveis booleanas regra2, regra3 = falso;</p> <p>FAÇA $i=1:p$ (preencha os p cromossomos candidatos da população) Adicione as n chaves abertas a um vetor de chaves candidatas a partir do vetor <i>random keys</i>, conforme regra 1; ENQUANTO regra2 ou regra3 falsas SE há mais de um elemento do <i>vetor de ramo comum</i> em chaves candidatas regra2=verdadeiro; CASO CONTRÁRIO regra2=falso; FIM (SE) SE $regra2 == verdadeiro$; Encontre a posição do elemento que viola a <i>regra 2</i> e troque pelo elemento seguinte no correspondente vetor de <i>loop</i>; FIM (SE) SE todos os <i>vetores de ramos comuns</i> de algum <i>vetor de grupo proibido</i> participam simultaneamente Regra3=verdadeiro; CASO CONTRÁRIO Regra3=falso; FIM (SE) SE $regra3 == verdadeiro$ Encontre esses elementos do vetor de grupo proibido no vetor cromossomo e troque pelo elemento seguinte no correspondente vetor de loop básico; FIM (SE) FIM (ENQUANTO) FIM (FAÇA)</p>

Fonte: Elaborado pelo autor.

3.4 SIMULATED ANNEALING PARA OTIMIZAÇÃO COMBINATÓRIA

Algoritmos de busca local constituem uma classe de algoritmos de aproximação geral que são baseados na melhoria progressiva sob o valor da função de custo pela exploração de vizinhanças. Desta forma, seja (S, f) uma instância de um problema de otimização combinatória e \mathcal{N} uma estrutura de vizinhança. Então, essa estrutura de vizinhança é um mapeamento que define para cada solução $i \in S$ um conjunto $S_i \subset S$ de soluções que são “próximas” a i em algum sentido. O conjunto S_i é chamado de vizinhança de uma solução i , e cada $j \in S_i$ é chamado de uma solução vizinha de i (AARTS; KORST, 1991).

Define-se, um mecanismo de geração como um meio capaz de selecionar uma solução j da vizinhança S_i de uma solução i . Em outras palavras, dada uma instância de um problema de otimização combinatória e uma estrutura de vizinhança, então, um algoritmo de busca local é um algoritmo que itera (repete) sobre um número de soluções. Começa com uma dada solução inicial, que é frequentemente escolhida aleatoriamente e, em seguida, pelo mecanismo de geração tentando continuamente encontrar uma melhor solução pela busca da vizinhança da solução corrente para uma solução com menor custo. Se tal solução é encontrada, a solução corrente é substituída

por essa solução, caso contrário, o algoritmo continua com a solução corrente. O algoritmo termina quando nenhuma melhoria pode ser obtida (AARTS; KORST, 1991).

Outro conceito importante é a otimalidade local, seja novamente (S, f) uma instância de um problema de otimização combinatória e \mathfrak{N} uma estrutura de vizinhança, então $i \in S$ é chamado uma solução ótima localmente ou simplesmente um ótimo local, em relação a \mathfrak{N} , se i é melhor do que, ou igual a, todas as suas soluções vizinhas em relação ao seu custo. Mais especificamente, no caso da minimização, i é chamado de uma solução mínima localmente, ou simplesmente mínimo local, se a equação (23) é satisfeita (AARTS; KORST, 1991).

$$f(i) \leq f(j), \quad \forall j \in S_i \quad (23)$$

Assim, o princípio básico de algoritmos de busca local é buscar soluções através de iterações entre soluções vizinhas.

3.4.1 ALGORITMO DE METROPOLIS

O *Simulated Annealing* se origina da analogia com o processo físico de recozimento de sólidos. Na física da matéria condensada, recozer é conhecido como um processo térmico para obter um estado de mínima energia livre. No processo, um sólido em banho quente é aquecido pelo aumento da temperatura do banho até que este sólido derreta, em seguida é diminuída lentamente a temperatura do banho quente até as partículas deste sólido se rearranjarem em um estado base. Na fase líquida todas as partículas do sólido se arranjam aleatoriamente. No estado base, as partículas são arranjadas em uma estrutura altamente reticulada e a energia do sistema é mínima. O estado base do sólido é obtido apenas se a temperatura máxima é suficientemente alta e o resfriamento é feito suficientemente lento. Caso contrário, o sólido irá congelar dentro de um estado metaestável ao invés do estado base (AARTS; KORST, 1991) (LAARHOVEN; AARTS, 1987).

Em 1953, Metropolis *et al.* (1953) introduziram um método geral para investigação das propriedades das iterações individuais das moléculas de um sólido como equações de estado. O método introduzido por esses autores é baseado nas técnicas de Monte Carlo modificadas e gera uma sequência de estados do sólido. Assim, dado um estado corrente i do sólido de energia E_i , então um subsequente estado j é gerado pela aplicação de um mecanismo de perturbação que transforma o estado corrente em um próximo estado através de uma pequena perturbação e, assim, obter a energia do próximo estado (E_j). Se a diferença de energia $E_j - E_i$, é menor ou igual a zero, o estado j é aceito como um estado corrente. Todavia, se a diferença de energia é maior do que zero, o estado j é aceito com uma certa probabilidade definida pela equação (24).

$$f(x) = \exp\left(\frac{E_i - E_j}{k_b T}\right) \quad (24)$$

Onde T denota a temperatura do banho quente e k_b é uma constante física conhecida como constante de Boltzmann. A regra de aceitação descrita acima é conhecida como critério de Metropolis. Se a diminuição da temperatura é feita suficientemente lenta, o sólido pode alcançar o equilíbrio térmico em cada temperatura. No algoritmo de Metropolis isso é alcançado pela geração de um grande número de transições em um dado valor de temperatura.

Se um estado é definido pelo conjunto de posições das partículas, então, no equilíbrio térmico, a probabilidade de o sistema estar no estado i com energia E_i , em uma temperatura T , é representada pela distribuição de Boltzmann, equações (25) e (26) (LAARHOVEN; AARTS, 1987).

$$\pi_i = Pr(s = i) = \frac{1}{Z(T)} \exp\left(-\frac{E_i}{k_b T}\right) \quad (25)$$

$$Z(T) = \sum_{i \in S} \exp\left(-\frac{E_i}{k_b T}\right) \quad (26)$$

Onde a equação (26) é conhecida como uma função de partição, k_b é a constante de Boltzmann, T é a temperatura, $E(i)$ é a energia do estado i , e S é o espaço de estados.

Em Laarhoven e Aarts (1987) é demonstrado que para uma alta temperatura todos os estados são igualmente prováveis de serem obtidos. Neste caso, a energia desses estados não afeta a probabilidade do estado, equação (25). Por outro lado, em uma baixa temperatura, o valor da função exponencial é fortemente afetado pela energia do estado, assim, quando a temperatura se aproxima de zero, o sistema irá convergir ao estado com mínima energia, E_{min} uma vez que a probabilidade do estado, apenas com a mínima energia, é diferente de zero, enquanto a probabilidade dos outros é zero.

3.4.2 ALGORITMO DE RECOZIMENTO SIMULADO

O algoritmo *simulated annealing* foi independentemente desenvolvido por Kirkpatrick, Gelatt e Vecchi (1983) para resolver problemas de otimização combinatória, baseado nos estudos de Metropolis *et al.* (1953), foi observado que existe uma analogia entre a função de minimização de custo de um problema de otimização combinatória e o lento resfriamento de um sólido até que este alcance um estado base de baixa energia, assim, este processo de otimização pode ser executado pela aplicação do algoritmo de Metropolis e a geração de uma sequência de lentos decrementos de valores de temperatura, resultando em uma analogia entre o sistema físico multipartículas e um problema de otimização, mostrada no Quadro 6.

Desse modo, soluções em um problema de otimização combinatória são equivalentes a estados em um sistema físico e o custo de uma solução é equivalente ao estado da energia. Assim, o algoritmo de recozimento simulado pode ser visto como uma iteração dos algoritmos de Metropolis, avaliados em valores decrescentes dos parâmetros de controle, para esta dissertação o parâmetro de controle será a temperatura.

Quadro 6 – Analogia entre o sistema físico e o recozimento simulado.

Sistema Físico	Problema de Otimização
Estado	Solução (configuração da rede)
Energia	Custo (perdas na rede elétrica)
Estado Base	Solução Ótima
Resfriamento Rápido	Busca Local
Resfriamento Lento	Recozimento Simulado

Fonte: (KIRKPATRICK; GELATT; VECCHI, 1983)

Como definido no início da seção, em algoritmos de busca local existem uma estrutura de vizinhança e um mecanismo de geração. Assim, seja (S, f) denotando uma instância de um problema de otimização combinatória e i e j duas soluções com custo $f(i)$ e $f(j)$, respectivamente. Então, o critério de aceitação determina se j é aceito de i pela aplicação da seguinte probabilidade de aceitação, equações (27) e (28).

$$P_T(j \text{ aceito}) = 1, \quad \text{se } f(j) \leq f(i) \quad (27)$$

$$P_T(j \text{ aceito}) = \exp\left(\frac{f(i) - f(j)}{T}\right), \quad \text{se } f(j) > f(i) \quad (28)$$

Onde $T \in \mathbb{R}^+$ e denota o parâmetro de controle. Desta forma, pode-se notar que o mecanismo de geração corresponde ao mecanismo de perturbação e o critério de aceitação corresponde ao critério de Metropolis.

Deve-se definir ainda uma *transição* que é uma ação combinada resultante de uma transformação de uma solução corrente em uma subsequente, esta ação consiste na aplicação de um mecanismo de perturbação e na aplicação do critério de aceitação. Uma propriedade típica do algoritmo de recozimento simulado é que, além de aceitar melhorias na função objetivo (função custo), também aceita deteriorações até certo ponto, entende-se por deteriorações as soluções com desempenho pior do que a encontrada na solução corrente.

Inicialmente, em grandes valores de T várias deteriorações serão aceitas, conforme T decresce, apenas pequenas deteriorações serão aceitas e finalmente com o valor de T se aproximando de zero, nenhuma deterioração será mais aceita. Essa configuração significa que o algoritmo do recozimento simulado, em contraste com outros algoritmos de busca locais, pode escapar de mínimos locais. A probabilidade de aceitar deteriorações é implementada pela comparação do valor na equação (28) com

um número aleatório gerado de uma distribuição uniforme no intervalo $[0,1]$. O reconhecimento simulado pode ser modelado matematicamente usando a teoria de cadeias de Markov, isto é, uma sequência de tentativas, onde a probabilidade do resultado de uma dada tentativa depende apenas do resultado da tentativa anterior (AARTS; KORST, 1991).

3.4.3 MECANISMO DE PERTURBAÇÃO

Em problemas de otimização combinatória de grande porte, um apropriado mecanismo de geração de novas vizinhanças deve ser estabelecido para que se possa explorar corretamente o espaço solução do problema. Assim, no problema de reconfiguração de redes de distribuição como o estado aberto/fechado das chaves seccionadoras e de laço determinam a configuração da rede do sistema, um mecanismo de perturbação deve ser capaz de alterar este estado das chaves e gerar uma nova configuração do sistema a partir da configuração corrente do sistema. Nesta seção, será mostrado o primeiro trabalho no desenvolvimento de um mecanismo de perturbação para o problema de reconfiguração de redes de distribuição de energia e, em seguida, o mecanismo de perturbação adotado nesta dissertação.

O primeiro mecanismo de perturbação para problemas de reconfiguração foi proposto por (CHIANG; JEAN-JUMEAU, 1990a) (CHIANG; JEAN-JUMEAU, 1990b) e é baseado na adição/remoção de chaves sem considerar a topologia da rede. Este mecanismo de perturbação pode ser descrito como:

1. Escolha randomicamente uma chave t de um conjunto de chaves de laço Ω_{laco} e então a feche. A chave t é removida de Ω_{laco} e alocada no conjunto de chaves seccionadoras Ω_{sec} . Isso cria um percurso fechado no sistema, e as chaves seccionadoras neste caminho fechado são incluídas em um conjunto *loop* Ω_{loop} .
2. Randomicamente escolha uma chave do conjunto Ω_{loop} e então a abra. Isso irá restaurar o sistema de volta a estrutura radial. A chave s é removida do conjunto Ω_{sec} e adicionada ao conjunto Ω_{laco} .

Para ilustrar este mecanismo, toma-se como exemplo a rede teste de 14 barras na Figura 7, onde as chaves de laço e seccionadoras são representadas pela linha pontilhada (vermelho) e a contínua (preta), respectivamente. Quando fechamos a chave de laço S14, um percurso fechado é criado e as chaves seccionadoras S1, S2, S5, S6, e S8 são incluídas no conjunto Ω_{loop} . Se qualquer chave seccionadora em Ω_{loop} é aberta, a configuração do sistema é restaurada à sua configuração radial. Como resultado desse chaveamento, uma nova configuração do sistema pode ser gerada, que satisfaz as restrições de configuração radial e suprimento de energia. Esse mecanismo de perturbação é adequado para o problema de reconfiguração da

rede, mas sua representação não utiliza o conhecimento da topologia do sistema. As chaves de laço e seccionadoras são randomicamente selecionadas e isto implica que a topologia do sistema e a temperatura não afetam o processo de busca e, desta forma, a propriedade de busca do recozimento simulado não pode ser influenciada.

Ainda na Figura 7, é possível notar que o espaço solução local do *loop* 1 formado pela chave de laço S14 possui menos elementos (chaves seccionadoras) do que o *loop* 3 formado pelo fechamento da chave de laço S16. Dessa forma, se todas as chaves de laço são selecionadas com igual probabilidade, no *loop* 3 a busca pode não ser realizada suficientemente, pois, a busca realizada no *loop* 1 é realizada mais frequentemente, dessa forma, retornando muitas vezes a uma configuração já visitada, ou seja, a busca é realizada mais do que é necessário. Portanto, é essencial que as chaves de laço sejam selecionadas em relação tamanho dos percursos fechados formados, porque o tamanho de cada *loop* pode ser significativamente diferente nos sistemas de distribuição.

No método de seleção proposto por Jeon *et al.* (2002), um mecanismo de perturbação baseado na topologia da rede e na temperatura é introduzido, neste método para que o *loop* 3 seja selecionado com mais frequência do que o *loop* 1, uma normalização do tamanho de cada *loop* é realizada, isto é, uma seleção proporcional ao tamanho do *loop* é efetuada. Uma vez que o percurso fechado é selecionado, são atribuídas as chaves seccionadoras um nível de chave associado ao valor de temperatura corrente, isto é feito para conduzir a uma diversidade na busca.

Normalmente, ao selecionarmos uma chave seccionadora que cause uma larga perturbação no sistema, esta configuração pode ser aceita pelo critério de Metropolis em altas temperaturas, causando deterioração na função de custo. No entanto, a maioria dessas transições largamente perturbadas são rejeitadas em baixas temperaturas. Desta forma, esse novo mecanismo de perturbação é capaz de gerar configurações aceitáveis em baixas temperaturas pelo aumento na busca por configurações que causem pequenas perturbações, conforme a temperatura corrente diminui. Por conseguinte, o *Mecanismo de Perturbação Baseado em Topologia* pode ser descrito como (JEON *et al.*, 2002):

1. Dada uma configuração radial inicial e um conjunto de chaves de laço, para cada chave de laço existe um caminho fechado formado pelo fechamento da chave. Então, o tamanho deste *loop* (isto é, o número de chaves seccionadoras contidas neste percurso fechado) pode ser determinado. Determine o tamanho dos *loops* correspondentes a todas as chaves de laço e normalize cada tamanho destes com a soma de todos os tamanhos de cada *loop*, desta forma, o resultado é o tamanho relativo de cada *loop* básico.
2. Execute uma seleção proporcional ao tamanho do *loop* de uma chave de laço correspondente através do giro da roleta ponderada. A roleta é viciada

com os tamanhos relativos dos *loops* fundamentais associados as chaves de laço. Feche a chave de laço selecionada, criando um percurso fechado no sistema. Forme o conjunto *loop* Ω_{loop} , com todas as chaves seccionadoras dentro do *loop* básico. Assim, as chaves de laço serão selecionadas em relação ao tamanho do *loop*.

3. Para todas as chaves seccionadoras no Ω_{loop} , calcule o *Switch Level* (SL), isto é, a distância entre a chave de laço t selecionada e cada chave seccionadora contida no *loop* fundamental. Escolha e abra uma chave seccionadora com a probabilidade de seleção dependente do nível da chave em Ω_{loop} . Isso irá restaurar o sistema de volta a estrutura radial.

A probabilidade de seleção das chaves seccionadoras é determinada pela distribuição uniforme viciada com o inverso do nível da chave, equação (29).

$$SW_i = \frac{1}{n} \gamma_k + (1 - \gamma_k) \left(\frac{\frac{1}{SL_i}}{\sum_{j=1}^n \frac{1}{SL_j}} \right) \quad (29)$$

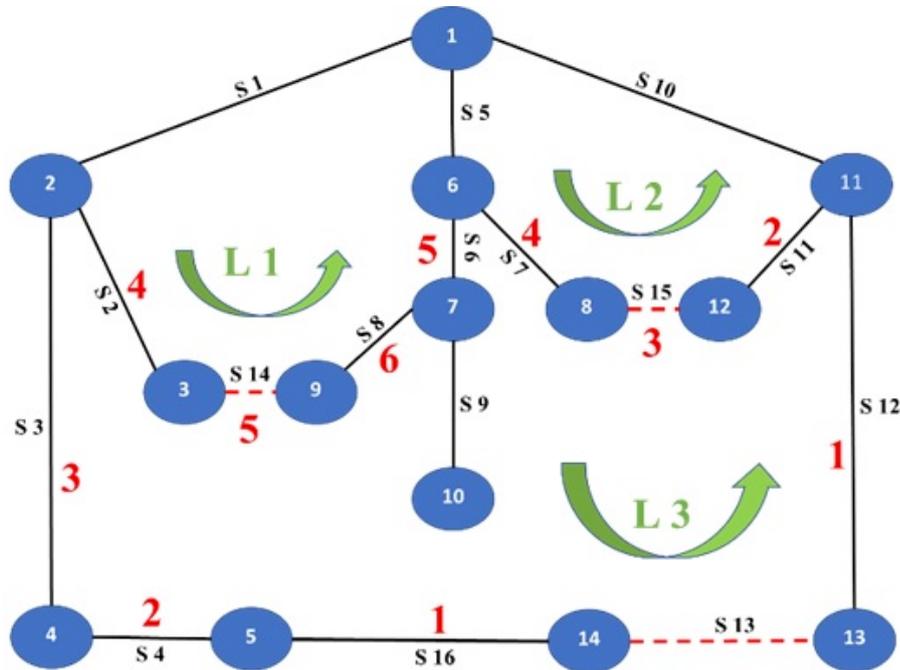
Onde SW_i e SL_i são as probabilidades de seleção e o nível da chave seccionadora i , respectivamente, n é o número de chaves seccionadoras em Ω_{loop} , e γ_k é um número positivo na faixa $[0,1]$, que é calculado pela equação (30).

$$\gamma_k = \lambda + \lambda e^{(T_k - T_0)} \quad (30)$$

Onde λ é um número positivo na faixa de $[0, 0.5]$, e T_0 e T_k são as temperaturas inicial e corrente, respectivamente. Para esta dissertação, é definido $\lambda = 0.5$, assim, na temperatura inicial T_0 todas as chaves seccionadoras são randomicamente selecionadas, isto é, $\gamma_k = 1$ e cada chave seccionadora pode ser selecionada com a mesma probabilidade. Pela diminuição da temperatura T_k , γ_k é diminuído, o que aumenta a probabilidade de seleção da chave com o menor nível de chave. Isso implica que uma configuração suavemente perturbada do proposto mecanismo de perturbação baseado em topologia é mais provável e, portanto, pode ser facilmente aceito como uma nova configuração em baixa temperatura.

Para exemplificar o procedimento de cálculo da probabilidade de seleção das chaves seccionadoras, toma-se como base a rede teste 14 barras. Após uma seleção proporcional ao tamanho do *loop*, onde a chave de laço S13 foi selecionada, é construído o conjunto de chaves seccionadoras obtidas do *loop* fundamental (L3) definido como $\Omega_{sec}=\{S2,S3,S4,S6,S7,S8,S11,S12,S14,S15,S16\}$, e definimos os pesos para estas chaves seccionadoras, conforme ilustra a Figura 12, assim, o nível de chave em S13 é $SL_{13}=\{4,3,2,5,4,6,2,1,5,3,1\}$, em seguida, utiliza-se as equações (29) e (30) para calcular a probabilidade de seleção das chaves seccionadoras.

Figura 12 – Exemplo para cálculo do nível da chave na rede.



Fonte: Elaborado pelo autor.

3.4.4 COOLING SCHEDULE

Uma implementação de tempo finito do recozimento simulado pode ser realizada pela geração de cadeias de Markov homogêneas de comprimento finito com uma sequência finita de parâmetros de controle decrescentes. Para isso deve-se especificar um conjunto de parâmetros que governam a convergência do algoritmo, esses parâmetros combinados formam o chamado *Cooling Schedules*. Portanto, *cooling schedules* especificam uma sequência finita de valores de parâmetros de controle, um parâmetro de controle frequentemente usado é a temperatura. Assim, têm-se um valor de temperatura inicial T_0 , uma função decremento para diminuição da temperatura, uma temperatura final especificada por um critério de parada e um comprimento finito de cada cadeia de Markov homogênea, isto é, um número finito de transições para cada temperatura (AARTS; KORST, 1991).

A pesquisa por adequados *cooling schedules* tem sido assunto de diversos trabalhos como em Laarhoven e Aarts (1987), Youssef, M. Sait e Adiche (2001), Triki, Collette e Siarry (2005) e Schneider e Puchta (2010), alguns desses trabalhos com abordagens bem diferentes das tradicionais como em Schneider e Puchta (2010) onde é utilizado como parâmetro de controle a probabilidade de Metropolis, equação (28), ao invés da temperatura.

A construção de *cooling schedules* é baseada no conceito de *quasi equilibrium*, onde o comprimento da cadeia de Markov e a função decremento devem ser escolhidos

tal que o *quasi equilibrium* seja restaurado no fim de cada cadeia de Markov. Assim, para grandes decrementos em T_k serão exigidos maiores comprimentos das cadeias de Markov a fim de restaurar o *quasi equilibrium* no próximo valor do parâmetro de controle $T_{(k+1)}$. Desse modo, existe um compromisso entre grandes decrementos de parâmetros de controle e pequenos comprimentos de cadeia de Markov. Normalmente, a escolha de pequenos decrementos em T_k evitam cadeias extremamente longas, porém, alternativamente, poderíamos usar grandes valores de tentativas das cadeias de Markov a fim de realizar de fazer grandes decrementos em T_k (AARTS; KORST, 1991).

3.4.4.1 Temperatura Inicial

O valor da temperatura inicial deve ser um valor suficientemente grande, permitindo a aceitação de virtualmente todas as transições propostas, uma vez que em altas temperaturas todas as soluções ocorrem com igual probabilidade. Assim, a melhor escolha para a temperatura inicial deve ser uma temperatura infinita ou pelo menos uma temperatura muito elevada. Por outro lado, não devemos utilizar um SA com temperaturas muito altas, uma vez que o tempo consumido e a qualidade média das soluções obtidas nas altas temperaturas são obviamente ruins. Matematicamente, isso significa que T_0 deve ser escolhido para que $\forall(x,y) : \exp(-\frac{f(x)-f(y)}{T_0}) \simeq 1$.

Desta forma, para obter a temperatura inicial partimos da regra demonstrada em Laarhoven e Aarts (1987) que é baseada na taxa de aceitação inicial χ_0 , definida como o número de transições ruins aceitas dividida pelo número de tentativas de transições ruins, sob o aumento médio no custo $\overline{\Delta f^{(+)}}$. Outra fórmula equivalente é proposta em Sait e Youssef (2000) e é definida como o número de transições aceitas dividido pelo número de tentativas de transições. Seja Δ^+ o conjunto de todos os movimentos de subida, então a mudança média no custo sobre todos os movimentos de subida é dada pela equação (31) (LAARHOVEN; AARTS, 1987) (TRIKI; COLLETTE; SIARRY, 2005):

$$\overline{\Delta f^{(+)}} = \frac{1}{|\Delta^+|} \sum_{\vec{xy} \in \Delta^+} f(y) - f(x) \quad (31)$$

Em seguida, assume-se que uma sequência de tentativas na forma de *Random Walk* é realizada em um certo valor T do parâmetro de controle. Seja m_1 que denota o número de transições propostas de i para j para o qual $f(j) \leq f(i)$, e m_2 o número de transições para que $f(j) > f(i)$. Além disso, seja $\overline{\Delta f^{(+)}}$ a diferença média no custo sobre m_2 transições de aumento de custo. Então, a taxa de aceitação χ pode ser aproximada pela equação (32) (LAARHOVEN; AARTS, 1987).

$$\chi \approx \frac{m_1 + m_2 \cdot \exp\left(-\frac{\overline{\Delta f^{(+)}}}{T}\right)}{m_1 + m_2} \quad (32)$$

Do qual é obtido:

$$T = \frac{\overline{\Delta f}^{(+)}}{\ln\left(\frac{m_2}{m_2 \cdot \chi - m_1(1-\chi)}\right)} \quad (33)$$

A temperatura inicial de T_0 pode ser calculada da equação (33) do seguinte modo. Inicialmente, T_0 é configurado igual a zero. Depois, uma sequência de m tentativas de *Random Walk* é gerada. Em seguida, em cada tentativa um novo valor de T_0 é calculado da equação (33), onde $\chi = \chi_0$, denominada taxa de aceitação inicial, para esta dissertação será utilizado o valor $\chi_0 = 0,6$ obtido de Jeon *et al.* (2002). Os valores de m_1 e m_2 correspondem aos números de transições com diminuição no custo e transições com aumento no custo, respectivamente. Além disso, $m = m_1 + m_2$, desta forma é obtido o valor inicial da temperatura (LAARHOVEN; AARTS, 1987).

3.4.4.2 Função Decremento de Temperatura

O comprimento da cadeia de Markov e a função decremento para mudança de T_k em $T_{(k+1)}$ são fortemente relacionadas através do conceito de *quasi equilibrium*. Se o decremento em T_k é grande, leva mais tempo para estabelecer o *quasi equilibrium* em $T_{(k+1)}$. Assim, há uma forte relação entre o rápido decremento de T_k e o comprimento da cadeia de Markov. Nesta dissertação, a função de decremento utilizada é baseada na proposta de M. D. Huang, Romeo e Sangiovanni-Vincentelli (1986), onde a regra de decremento é baseada nos valores de custo médio das cadeias de Markov consecutivas, equação (34) (TRIKI; COLLETTE; SIARRY, 2005).

$$T_{k+1} = T_k \cdot \exp\left(-\frac{\delta T_k}{\sigma(T_k)}\right) \quad (34)$$

Onde $\sigma(T_k)$ é o desvio padrão do custo na k -ésima cadeia de Markov e δ é um parâmetro constante definido pelo usuário e está contido no intervalo ($0 < \delta \leq 1$), um valor típico de δ é 0,7.

3.4.4.3 Temperatura Final

A terminação do algoritmo é baseada na extrapolação do custo esperado na temperatura final $\langle f_{T_k} \rangle$ para $T_k \rightarrow 0$. Seja a equação (35) (LAARHOVEN; AARTS, 1987):

$$\Delta\langle f \rangle_T = \langle f \rangle_T - f_{opt} \quad (35)$$

Então, a execução do algoritmo é terminada se $\Delta\langle f \rangle_T$ é pequena comparada a $\langle f \rangle_{T_0}$, o custo esperado em T_0 . Para valores suficientemente grandes de T_0 , temos $\langle f \rangle_{T_0} \approx \langle f \rangle_{\infty}$. Portanto, pode-se aproximar $\Delta\langle f \rangle_T$ para $T \ll 1$ por (LAARHOVEN; AARTS, 1987):

$$\Delta \langle f \rangle_T \approx T \frac{\partial \langle f \rangle_T}{\partial T} \quad (36)$$

Portanto, pode-se terminar o algoritmo para algum k , tal que:

$$\frac{T_k}{\langle f \rangle_\infty} \left. \frac{\partial \langle f \rangle_T}{\partial T} \right|_{T=T_k} < \varepsilon_s \quad (37)$$

Onde ε_s é um número positivo pequeno, denominado parâmetro de parada, sendo a equação (37) o critério de parada. Sendo $\langle f \rangle_{T_k}$ o custo médio na k -ésima cadeia de Markov, então, $\partial \langle f \rangle_{T_k} / \partial T$ em $T = T_k$ é a taxa de mudança no custo médio em T_k . Para o propósito desta dissertação, fazemos o uso da equivalência, equação (38) (TRIKI; COLLETTE; SIARRY, 2005) (SCHNEIDER; PUCHTA, 2010).

$$\frac{\partial \langle f(T) \rangle}{\partial T} = \frac{Var(f)}{T^2} \quad (38)$$

No equilíbrio térmico, com $\langle f(T) \rangle$ sendo o valor esperado da energia. Muitos autores escrevem $Var(T)$ ao invés de $Var(f)$ para a variância dos valores de energia a fim de enfatizar que o tamanho das flutuações dos vários valores de energia em torno de seu valor médio fortemente depende da temperatura T (SCHNEIDER; PUCHTA, 2010).

Assim, a execução do algoritmo é terminada se o valor da função custo da solução obtida na última tentativa da cadeia de Markov permanece sem alteração para um número consecutivo de cadeias.

3.4.4.4 Comprimento da Cadeia de Markov

Uma cadeia de Markov é uma sequência de tentativas, onde a probabilidade do resultado de uma dada tentativa depende apenas do resultado da tentativa anterior. Uma vez que já definimos que as cadeias de Markov neste recozimento simulado são homogêneas, então, por definição, a probabilidade de transição é independente do número de tentativas.

O comprimento das cadeias de Markov são baseados na exigência de que cada valor de T_k do parâmetro de controle o *quasi equilibrium* seja restabelecido. O número de transições necessárias para alcançar isso é calculado do argumento intuitivo de que o *quasi equilibrium* será restabelecido depois da aceitação de pelo menos algum número fixo de transições. Entretanto, as transições são aceitas com probabilidade decrescente, obtendo um comprimento da cadeia de Markov tendendo ao infinito quanto a temperatura T_k tende a zero. Conseqüentemente, o comprimento da cadeia de Markov é limitado a alguma constante \bar{L} para evitar cadeias de Markov extremamente longas para pequenos valores de T_k (AARTS; KORST, 1991).

A função de decremento da temperatura exige apenas “pequenos” números de tentativas de soluções. De acordo com Laarhoven e Aarts (1987) um comprimento

da cadeia de Markov satisfatório deve ser igual ao tamanho das vizinhanças. Para esta dissertação, o comprimento da cadeia de Markov escolhido depende do tamanho da rede teste utilizada, assim, o comprimento de Markov será o número de chaves controladas na rede teste estudada, desta forma, conforme aumentamos a dimensão das redes de distribuição maiores serão os valores exigidos para o comprimento das cadeias de Markov.

3.4.5 RECOZIMENTO SIMULADO NO PROBLEMA DE RECONFIGURAÇÃO

No problema de reconfiguração de redes, o espaço solução no recozimento simulado é o conjunto de todos os possíveis estados aberto/fechado das chaves. A nova configuração do sistema é gerada pelo uso do mecanismo de perturbação baseado em topologia, todavia, em algumas situações violações das restrições aparecem ao longo do processo de otimização, e para obter soluções factíveis o uso de técnicas de penalização devem ser utilizadas. Nesta dissertação, a restrição de afundamento de tensão é abordada segundo o critério da eliminação do indivíduo não factível, já descrito na seção 3.5.1. Assim, qualquer violação das restrições de suprimento de energia ou configuração radial podem ser evitadas pelo uso do mecanismo de perturbação antes do cálculo do fluxo de carga, enquanto as violações de afundamento de tensão são eliminadas através do método da pena de morte após a aplicação do cálculo de fluxo de carga.

Além disso, para avaliar a radialidade e ocorrência de caminhos fechados nos sistemas teste uma técnica deve ser implementada, em Stojanović e Rajić (2017), uma matriz de checagem de malha, com dimensão igual ao número de nós n da rede teste, é usada a cada iteração para verificar se a rede é conectada e se possui percursos fechados, porém o uso desta técnica aumenta o esforço computacional. No caso desta dissertação o OpenDSS já fornece automaticamente uma série de informações após a execução do fluxo de carga, entre essas informações estão variáveis com informações sobre os percursos fechados, cargas desconectadas e ramos abertos contidas na classe *DSSCircuit.Topology* que retornam informações que substituem a matriz de checagem de malha e, portanto, diminuindo o esforço computacional ao retirar a necessidade desta matriz.

Por fim, as implementações do algoritmo recozimento simulado usualmente contêm duas estruturas de repetição (laços) sobrepostos: no laço externo ocorre o controle do decremento da temperatura e no laço interno a temperatura permanece constante e consiste da aplicação do algoritmo de Metropolis. Assim, na simulação do SA ocorre inicialmente a exploração da vizinhança e o armazenamento de cada ponto encontrado em uma lista. Então, em cada iteração do laço interno um procedimento que simula o algoritmo de Metropolis é chamado, depois de decorrido L tentativas do comprimento de Markov, o algoritmo segue para o laço externo onde a temperatura

é reduzida seguindo a função decréscimo escolhida (TRIKI; COLLETTE; SIARRY, 2005).

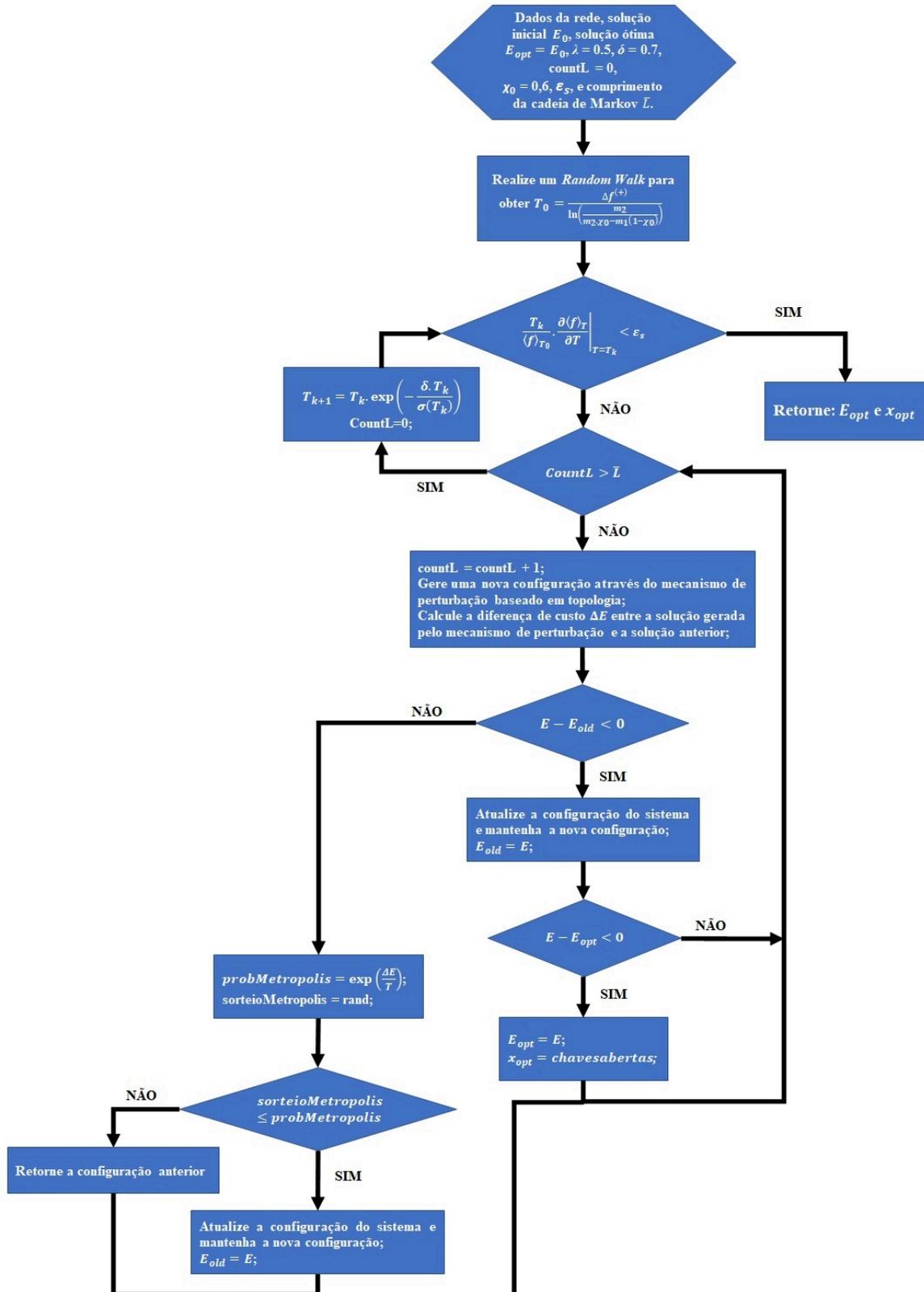
O pseudocódigo e o fluxograma para geração do recozimento simulado na reconfiguração de redes de distribuição de energia é mostrado no Quadro 7 e na Figura 13, respectivamente.

Quadro 7 – Pseudocódigo para reconfiguração usando SA.

<p>Entrada: dados da rede, solução inicial E_0, solução ótima $E_{opt} = E_0$, $\lambda = 0.5$, $\delta = 0.7$, $countL = 0$, taxa de aceitação inicial $\chi_0 = 0,6$, parâmetro de parada ϵ_s, e comprimento da cadeia de Markov \bar{L}.</p> <p><i>Random Walk</i> para obter $T_{inicial}$ a partir da Equação (33)</p> <p>ENQUANTO $\left. \frac{T_k}{\langle f \rangle_\infty} \frac{\partial \langle f \rangle_T}{\partial T} \right _{T=T_k} < \epsilon_s$</p> <p>ENQUANTO $countL < \bar{L}$ $countL = countL + 1$; Gere uma nova configuração através do mecanismo de perturbação baseado em topologia; Calcule a diferença de custo ΔE entre a solução gerada pelo mecanismo de perturbação e a solução anterior; SE classe <i>DSSCircuit.Topology</i> retorna restrições de afundamento de tensão e configuração radial factíveis SE $(E - E_{old}) < 0$ Atualize a configuração do sistema e mantenha a nova configuração; $E_{old} = E$; SE $E < E_{opt}$ $E_{opt} = E$; $x_{opt} = chavesabertas$; FIM (SE) CASO CONTRÁRIO $probMetropolis = \exp\left(\frac{\Delta E}{T}\right)$; sorteioMetropolis = rand; SE sorteioMetropolis \leq probMetropolis $E_{old} = E$; CASO CONTRÁRIO Retorne a configuração anterior; FIM (SE) FIM (SE) CASO CONTRÁRIO Retorne a configuração anterior; FIM (SE) FIM (ENQUANTO) $T_{k+1} = T_k \cdot \exp\left(-\frac{\lambda T_k}{\sigma(T_k)}\right)$; $countL = 0$; FIM (ENQUANTO) RETORNE E_{opt} e x_{opt}</p>
--

Fonte: Elaborado pelo autor.

Figura 13 – Fluxograma do algoritmo de recozimento simulado.



Fonte: Elaborado pelo autor.

3.5 ESTRATÉGIAS DE APLICAÇÃO DE RESTRIÇÕES

Existem várias formas de lidar com as restrições em problemas que envolvem algoritmos genéticos, algumas foram estudadas no trabalho de Ponsich *et al.* (2008), que revisaram as técnicas: eliminação de indivíduos não factíveis, penalização da função objetivo, conceitos de dominância, preservação da factibilidade, reparo de indivíduos inviáveis e métodos híbridos. Algumas destas técnicas também podem ser aplicadas a outras metaheurísticas como o recozimento simulado. As mais tradicionais e utilizadas nesta dissertação, são as técnicas eliminação de indivíduos não factíveis e penalização da função objetivo.

3.5.1 Eliminação

Esse método, também chamado “método da pena de morte”, consiste em rejeitar indivíduos não factíveis. O método é muito simples de implementar, mas tem como principal fraqueza a retirada de informações importantes do espaço das soluções não factíveis, o que poderia ajudar a guiar a busca rumo ao ótimo global. Entretanto, essa técnica constitui uma primeira abordagem válida quando nenhum recurso permite previamente especificar um método ajustado ao problema. No caso desta dissertação a aplicação deste método foi aplicada apenas no algoritmo de recozimento simulado. Observou-se nos testes que no algoritmo genético de chaves aleatórias viciadas para encontrar a solução ótima utilizando este método foi exigido um número consideravelmente maior de gerações para obter o ótimo global do que o método de funções penalidade, descrito a seguir, devido a esta perda de informações do espaço das soluções não factíveis.

3.5.2 Funções Penalidade

Neste método, o problema com restrição é transformado em um problema irrestrito pela introdução das restrições na função objetivo através dos termos de penalidade. Assim, é possível formular os termos da penalidade de acordo com uma larga diversidade de técnicas. Sendo que uma penalização será eficiente se sua expressão está mais relacionada a quantidade de violação da restrição do que ao número da restrição violada (PONSICH *et al.*, 2008).

A partir da formulação clássica do problema de otimização:

$$\min f(x) \quad \text{s.a.} \quad g_j(x) \leq 0, \quad j = 1, \dots, m \quad (39)$$

Então, com a formulação irrestrita incluindo o termo de penalidade, o novo critério F para minimizar pode ser geralmente escrito dessa forma:

$$\min F(x) = f(x) + \sum_{j=1}^m R_j \max[0, g_j(x)]^\beta \quad (40)$$

Na maioria das vezes, a penalidade é expressada sobre uma forma quadrática, correspondendo a $\beta = 2$. O fator R_j pode ser expresso de muitos modos, mostrando várias complexidades e melhorando a eficiência da solução para o problema abordado. Assim, na maioria dos problemas, o ótimo global está localizado no limite do espaço factível. Então, se a influência do fator penalidade é muito importante, a pressão exercida para deslocar os indivíduos de dentro do espaço factível será muito forte, impedindo-os de irem para regiões mais promissoras. Além disso, no caso de espaços factíveis separados, um fator de penalidade muito alto pode confinar a população a uma região de factibilidade sem permitir que indivíduos cruzem zonas não factíveis e se dirijam por outras regiões factíveis (onde o ótimo global poderia estar localizado) (PONSICH *et al.*, 2008).

Por outro lado, um fator de penalidade muito baixo pode conduzir a uma busca exaustiva no espaço não factível, visitando regiões onde a função objetivo é muito baixa, mas que são fortemente inviáveis. Além disso, é normalmente admitido que o termo de penalidade deve ser preferencialmente muito baixo no início da busca, a fim de explorar a larga região do espaço de busca. No fim de uma execução, regiões promissoras devem ainda serem determinadas. É então mais relevante ter um termo de uma alta penalidade, para intensificar a busca nessas zonas forçando os indivíduos a satisfazer as restrições. No estado da arte uma grande variedade de métodos de penalização já foi implementada, uma das mais simples é a penalidade estática: um valor numérico, que não varia durante toda a busca, é alocado em cada fator R_j (PONSICH *et al.*, 2008).

No caso desta dissertação, a restrição em que foi aplicada essa metodologia foi o afundamento de tensão, a equação (39) foi utilizada de forma que o desvio de tensão fosse multiplicado por um fator que levasse o *fitness*, do indivíduo que violou a restrição, a um termo próximo das perdas iniciais, desta forma a escolha desses fatores dependia apenas das perdas iniciais. Evitando a escolha de fatores de penalidade muito altos ou baixos conforme explicado anteriormente.

3.6 METODOLOGIAS DE RESTART APLICADAS AO BRKGA

Quando se avalia a performance de um método de otimização, a maioria dos relatórios envolvem o tempo de execução e a qualidade da solução, ambos são sensíveis as condições iniciais e aos parâmetros dos algoritmos, especialmente em algoritmos randomizados. Essa variabilidade nos tempos de execução pode ser frequentemente atribuída aos passos randomizados do algoritmo, tal como regras de seleção, soluções estocásticas iniciais e movimentos de busca local randômicos em metaheurísticas.

Diferentemente dos algoritmos de Monte Carlo, que garantem providenciar respostas aproximadas ou soluções dentro de um tempo fixo, os algoritmos do tipo *Las Vegas* são algoritmos que sempre encontram “soluções corretas”, mas o tempo computacional exigido é incerto (MARTÍ; PARDALOS; RESENDE, M. G. C., 2018).

Segundo Luby, Sinclair e Zuckerman (1993), um algoritmo do tipo *Las Vegas* é um algoritmo randomizado que sempre fornece a resposta correta quando termina, mas cujo tempo de execução é uma variável aleatória. Assim, seja $A(x)$ um algoritmo randomizado do tipo *Las Vegas*, em qualquer entrada x , a saída de $A(x)$ é sempre correta, mas seu tempo de execução $T_A(x)$ é uma variável aleatória. Dessa forma, o objetivo é minimizar, para cada entrada x , o tempo esperado exigido para obter a resposta de $A(x)$. Em geral, o tempo de execução de um algoritmo pode ser medido em iterações ou em tempo computacional. Ainda em Luby, Sinclair e Zuckerman (1993) uma proposta de estratégia de reinicialização é elaborada para se associar a um dado algoritmo e é definida como uma sequência, $S = \{t_1, t_2, \dots\}$, que representa os tempos de execução entre essas reinicializações. Desta forma, em uma estratégia S , a primeira execução continua por t_1 unidades de tempo ou até a resposta correta seja encontrada. A segunda execução dura t_2 unidades de tempo, ou até a resposta correta seja encontrada, e assim por diante. Esses tempos de execução são independentes um do outro, pois nenhuma informação é passada entre esses diferentes tempos computacionais. Por último, diferentes estratégias de *restart* podem conduzir a diferentes tempos médios de execução.

Em algumas aplicações, a distribuição do tempo de execução desses algoritmos tem uma grande dispersão, e há uma probabilidade relativamente alta de obter um tempo de execução que é distante do valor médio. Tais distribuições podem ser exploradas para acelerar a busca via estratégia multi-start, assim, cada execução tem uma duração limitada, e o algoritmo é repetidamente reiniciado com diferentes condições iniciais ou sementes aleatórias (*random seeds*). Dependendo da aplicação, alguns geradores dessas sementes podem ser mais rápidos ou retornar valores com maior precisão. Todos os geradores de números pseudoaleatórios são baseados em algoritmos determinísticos, e todos os geradores passam por um teste estatístico específico para a aleatoriedade. Nesta dissertação, o gerador utilizado foi o *Mersenne Twister*, proposto em Matsumoto e Nishimura (1998), que gera números pseudoaleatórios uniformes. O algoritmo fornece um período de $2^{19937} - 1$ e é 623-dimensional equidistribuído e cada valor $U(0,1)$ é criado usando dois inteiros de 32 bits, assim, o *Mersenne Twister* é capaz de gerar de 0 a $2^{32} - 1$ possibilidades de sementes geradoras distintas.

Estratégias de *restart* em otimização frequentemente providenciam uma aceleração do algoritmo para procedimentos de otimização randomizados. Em Sergienko, Shilo e Roshchin (2000), os autores providenciam uma definição formal sobre *restart*

e também estabelecem uma explicação teórica da eficiência da estratégia de *restart*, mostrando como a aplicação dessas estratégias podem reduzir significativamente o tempo médio para a solução do problema.

Os algoritmos genéticos são robustos, fáceis de implementar e aplicáveis para resolver ambos os problemas de otimização contínuos e discretos. Como outras técnicas de otimização heurísticas, tem uma garantia probabilística de alcançar a solução de ótimo global em um número finito de gerações. Entretanto, sofrem de uma pobre capacidade de busca local e, portanto, têm um tempo de execução lento. Para superar essas desvantagens várias ferramentas foram desenvolvidas por diversos pesquisadores. Uma dessas ferramentas é a estratégia de *restart* que pode ser simplificada como uma técnica que reinicializa o algoritmo com uma nova população de soluções, sempre que não for capaz de encontrar qualquer melhor solução que satisfaça alguma condição pré-especificada pelo usuário. Nas seções seguintes, duas estratégias de *restart* serão apresentadas cada uma com seus méritos e deméritos (DAS, A. K.; PRATIHAR, 2019).

3.6.1 ESTRATÉGIA DE RESTART SEM ELITISMO

Como introduzido na seção anterior, estratégias de *restart* para aceleração de algoritmos estocásticos foram primeiro propostos por Luby, Sinclair e Zuckerman (1993). Neste artigo, os autores definem uma estratégia de *restart* como uma sequência infinita de intervalos de tempo $S = \{t_1, t_2, t_3, \dots\}$ que definem os tempos de execução t_1 , $t_1 + t_2$, $t_1 + t_2 + t_3, \dots$, quando o algoritmo é reiniciado do princípio, isto é, o algoritmo é reiniciado usando uma nova semente do gerador de números aleatórios.

Assim, Luby, Sinclair e Zuckerman (1993) propõem uma estratégia ótima de *restart* que utiliza intervalos de tempo iguais $t_1 = t_2 = t_3 = t^*$ entre as reinicializações, onde t^* é uma constante. Contudo, implementar tal estratégia pode ser difícil na prática, porque exige inserir um valor constante de tempo t^* . Uma vez que não há informação a priori sobre a distribuição do tempo da execução da heurística para o problema de otimização sob consideração, corre-se o risco de escolher um valor de t^* muito pequeno ou muito grande. Desta forma, um valor que é muito pequeno pode fazer com que a variante com *restart* da heurística demore muito mais para convergir do que a variante sem *restart*. Por outro lado, um valor que é muito grande pode nunca reiniciar, fazendo com que a variante com *restart* da heurística seja tão longa para convergir quando a variante sem *restart*.

Uma característica que pode ser controlada mais facilmente e com menos variação do que o tempo de execução é o número de iterações entre melhorias da solução corrente, ou da melhor solução encontrada. Assim, em (RESENDE, M.; RIBEIRO, 2011) é proposto uma estratégia de *restart* que monitora a última iteração em que a solução foi melhorada, neste estudo uma estratégia de *restart* é aplicada em conjunto

com o algoritmo GRASP e a heurística path-relinking, nesta estratégia o algoritmo é reiniciado a cada κ iterações realizadas sem melhoria, sendo denominada estratégia de restart(κ).

Desta forma, reiniciar o algoritmo GRASP com path-relinking exige um conjunto de iterações $\{i_1, i_2, \dots\}$ na qual a heurística é reiniciada do princípio, além disso, exige esvaziar o conjunto elite, descartando a solução corrente, e iniciar uma nova iteração com uma nova semente para o gerador de números aleatórios. Essa estratégia também exige como entrada um valor para o parâmetro κ , assim, um limitado número de valores de κ quase sempre alcançam o resultado desejado, isto é, reduzem a contagem de tempo médio bem como seu desvio padrão. Nesta dissertação, essa estratégia de *restart* será aplicada ao *Biased Random Keys Genetic Algorithm* em ambos os decodificadores Kruskal e co-árvore para resolver o problema da reconfiguração de redes de distribuição de energia.

3.6.2 ESTRATÉGIA DE RESTART COM ELITISMO

Em Amit Kumar Das e Pratihari (2019), uma nova estratégia de *restart* com princípio de elitismo foi proposta de forma a promover um equilíbrio adequado entre exploração e aproveitamento durante a busca. Além disso, o esquema de reinicialização proposto é capaz de detectar uma situação de convergência prematura de um algoritmo genético e ativar uma reinicialização do algoritmo para evitar tal situação. A estratégia proposta é composta de três diferentes condições e sempre que qualquer uma dessas condições seja satisfeita, o algoritmo será reiniciado.

Antes de definirmos essas condições, alguns conceitos precisam ser definidos:

- **Quociente de Diversidade (DQ):** é definido como a taxa da diferença absoluta entre os valores do melhor *fitness* (f_{melhor}) e o *fitness* médio (f_{medio}) da população dividido entre o melhor *fitness* e o pior *fitness* (f_{pior}) da população, equação (41). Esse parâmetro deve ser calculado em cada geração e seu valor é encontrado variando entre 0 e 1, no geral.

$$DQ = \left(\frac{|f_{melhor} - f_{medio}|}{|f_{melhor} - f_{pior}|} \right) \quad (41)$$

- **Máximo Quociente de Diversidade (DQ_{max}):** é o máximo de todos os valores medidos de DQ e é também calculado em cada geração.
- **Limite do Quociente de Diversidade (DQ_{limite}):** é o valor limite para o quociente de diversidade e é calculado conforme mostra a equação (42).

$$DQ_{limite} = DQ_{max} \times (1 - DQ_{max}) \quad (42)$$

Na primeira condição, no fim de qualquer geração (exceto a primeira), quando todos os operadores evolucionários forem aplicados e não houver melhoria encontrada

da melhor solução disponível, comparada aquela da geração anterior, será incrementado o valor do parâmetro denominado m por um, ou seja, $m = m + 1$ a cada geração sem encontrar uma solução com o *fitness* menor (no caso da minimização da função objetivo) do que a melhor solução encontrada. O valor inicial desse parâmetro m é configurado para ser igual a zero no início de uma evolução, e sempre que não houver uma melhoria da melhor solução, será incrementado por um. Entretanto, se qualquer melhoria é vista na melhor solução em uma geração, o valor de m será novamente configurado igual a zero naquela geração em particular.

Se em uma certa condição, o valor de m é encontrado para ser maior do que um limite pré-definido, denominado m_{limite} , o algoritmo irá se preparar para uma reinicialização, isto é, um algoritmo genético reinicia se $m > m_{limite}$. O objetivo de selecionar um valor adequado para m_{limite} é difícil, pois depende da natureza do problema. Um valor muito baixo desse parâmetro iria conduzir a um frequente reinício do algoritmo e conseqüentemente, o algoritmo não estaria apto a convergir a uma solução ótima ou próxima do ótimo. Por outro lado, um valor muito alto de m_{limite} pode ser a razão para uma menor eficiência da busca global do algoritmo.

Na segunda condição durante uma particular evolução, um algoritmo genético irá contar o número de gerações, parâmetro denominado c_{gen} , onde o parâmetro m é visto tendo um valor maior do que zero. Em outras palavras, o parâmetro c_{gen} será incrementado por um, isto é, $c_{gen} = c_{gen} + 1$, sempre que m é encontrado maior do que zero em uma evolução. Da mesma forma que m , o valor inicial para c_{gen} é configurado igual a zero durante o início de uma evolução. Desta forma, em uma certa geração, se c_{gen} é encontrado sendo maior do que uma contagem limite pré-especificada do valor do parâmetro, denominado th_{count} , e o valor de DQ é também encontrado menor do que DQ_{limite} , então o algoritmo será reiniciado.

Essa condição de *restart* pode ser escrita como “Se ($c_{gen} > th_{count}$ && $DQ < DQ_{limite}$)” e o algoritmo se prepara para uma reinicialização. Depois de cada *restart*, o valor de c_{gen} é novamente configurado para ser igual a zero. Em geral, o parâmetro DQ é visto para ter um elevado valor no estágio inicial de uma evolução e posteriormente, aumenta gradualmente. Agora, um ponto importante é notado aqui sobre o tempo de implementação dessa segunda condição para um *restart*. Essa condição é aplicada, apenas quando há um suficiente decremento na diversidade da população. Desta forma, inerentemente mantém um adequado balanço entre as capacidades de exploração e aproveitamento de um algoritmo genético e faz o processo de busca mais eficiente.

Na terceira condição: o valor de DQ é encontrado maior ou igual a um ($DQ \geq 1$), ou igual a zero ($DQ = 0$), ou igual ao infinito ($DQ \rightarrow \infty$), ou um valor ‘NaN’ ($DQ = NaN$), então é assumido que é um caso de convergência prematura do algoritmo. Nesse caso em particular, atribuímos o valor de DQ igual ao valor de DQ_{max} . Além disso, outro

parâmetro, chamado *flag*, é atribuído valor igual a um. Caso contrário, o parâmetro *flag* será assumido ter um valor igual a zero. Entretanto, sempre que o algoritmo enfrenta esse tipo especial de situação para a ocorrência de convergência prematura (isto é, *flag* = 1), será reiniciado e uma nova evolução será iniciada. Nota-se que depois de cada reinicialização, o parâmetro *flag* será atribuído novamente o valor igual a zero. Assim, essa condição de *restart* irá ajudar o algoritmo a superar a situação de convergência prematura.

Dessa forma, sempre que qualquer uma dessas condições é cumprida, o algoritmo é reiniciado. Durante a reinicialização, uma cópia da solução elite disponível na última geração é transferida para a próxima geração e o resto das soluções são criadas randomicamente. A diferença significativa dessa estratégia de *restart* é a utilização do princípio do elitismo, que é nada mais do que diretamente copiar as soluções elite para a próxima iteração da população, assim a melhor solução disponível no momento não é perdida como acontece na estratégia de reinicialização abordada na seção anterior, como será mostrado adiante isso provocará uma considerável melhoria em relação a proposta anterior (DAS, A. K.; PRATIHAR, 2019).

3.7 METODOLOGIA HÍBRIDA BASEADA EM BRKGA E SA

Muitos autores consideram que nem sempre os algoritmos genéticos são a melhor solução para problemas de otimização específicos, pois, esses algoritmos têm como desvantagem uma pobre capacidade de busca local e a introdução de um “overhead computacional”, isto é, um gasto excessivo de tempo computacional e memória, devido à busca baseada em populações, característica dos algoritmos genéticos. Assim, para contornar essas desvantagens muitos autores propõem a união das técnicas de busca local com os algoritmos genéticos, desta forma o novo algoritmo pode aumentar as capacidades de busca global e local, aumentando a sua capacidade de exploração e aproveitamento das soluções.

Um tipo de algoritmo híbrido é o algoritmo memético que segundo Neri, Cotta e Moscato (2012) são metaheurísticas baseadas em população compostas de uma estrutura evolucionária e um conjunto de algoritmos de busca local que são ativados dentro de um ciclo de geração da estrutura externa. Neste caso, o conceito de meme é emprestado da filosofia e é entendido como a unidade de transmissão de cultura.

Em outras palavras, ideias complexas podem ser decompostas em memes que se propagam e sofrem mutação dentro da população. A cultura, deste modo, constantemente sofre evolução e tende rumo a melhorias progressivas. Desta forma, ideias fortes tendem a resistir e serem propagadas dentro da comunidade enquanto ideias fracas tendem a não serem selecionadas e, portanto, tendem a desaparecer. Uma vez que os algoritmos meméticos consistem do conceito de combinar algoritmos de busca global e local, estes representam uma classe ampla e flexível de algoritmos que

constroem novas estruturas a partir dos algoritmos evolucionários e assim, constituem uma nova filosofia de otimização.

Algoritmos meméticos tipicamente tentam usar soluções de alta qualidade como pontos de partida. Isso pode ser feito ou usando um mecanismo mais sofisticado, como por exemplo uma heurística construtiva, para injetar boas soluções na população inicial, ou pelo uso de procedimentos de busca local para melhorar soluções aleatórias. Assim, a ideia de hibridização consiste em combinar as vantagens de diferentes componentes, mas se os componentes dominam um outro, a hibridização pode mais atrapalhar do que ser útil e o esforço computacional pode ser desperdiçado. Para o caso de algoritmos meméticos, se o efeito da busca local é muito forte, o algoritmo pode rapidamente ficar preso em um ótimo local de má qualidade. Além disso, o algoritmo é provavelmente redescoberto no mesmo ótimo local repetidas vezes. Por último, uma busca local excessiva pode rapidamente conduzir a uma perda de diversidade dentro da população (NERI; COTTA; MOSCATO, 2012).

Segundo William Eugene Hart (1994) algoritmos genéticos com grandes populações são mais efetivos quando a busca local é usada raramente. Também afirma que uma larga frequência de busca local é necessária se o algoritmo não é capaz de identificar regiões que são prováveis de conter o ótimo local. Como a introdução de elitismo aumenta o grau de aproveitamento, comparado a exploração, menos busca local é necessária quando usamos o elitismo. Além disso, o tipo de algoritmo genético usado com a busca local tem um forte impacto na performance. William Eugene Hart (1994) também observa que o uso de busca local tem restringido muitas aplicações de uso de pequenas populações por causa do aumento do esforço computacional, isso se mantém em particular quando a busca local é aplicada a cada indivíduo na população.

3.7.1 PRESERVAÇÃO DA DIVERSIDADE

O problema da convergência prematura, pelo qual a população converge em torno de um ponto subótimo, pode ser particularmente problemático para os algoritmos meméticos. Se a busca local é aplicada até cada ponto ter sido movido a um ótimo local, então isso pode conduzir para perdas de diversidade dentro da população a menos que novos ótimos locais sejam constantemente identificados. Alternativamente, mesmo se a busca local é terminada antes da otimalidade local, um espaço de busca induzido com grandes vales de atração pode também resultar na convergência prematura para a solução subótima no centro de grandes vales de atração. Um número de abordagens tem sido desenvolvidas para combater esse problema (HART, William E.; KRASNOGOR; SMITH, 2005):

- Quando inicializar a população com indivíduos bem conhecidos, apenas use uma relativamente pequena porção deles;

- Aplique a busca local a uma pequena fração da população (o que ajuda a assegurar que o resto da população esteja diversa);
- Usando operadores de recombinação que são designados para preservar diversidade;
- Usando múltiplos métodos de busca local, onde cada um induz a um diferente espaço de busca com ótimos locais distintos;
- Modificando o operador seleção para prevenir duplicadas;
- Usando um critério *fuzzy* que explicitamente controla duplicatas, como a regra pivô no estágio de busca;
- Modificando o operador seleção, ou o critério e aceitação da busca local, para usar o método de Boltzmann e assim preservar a diversidade.

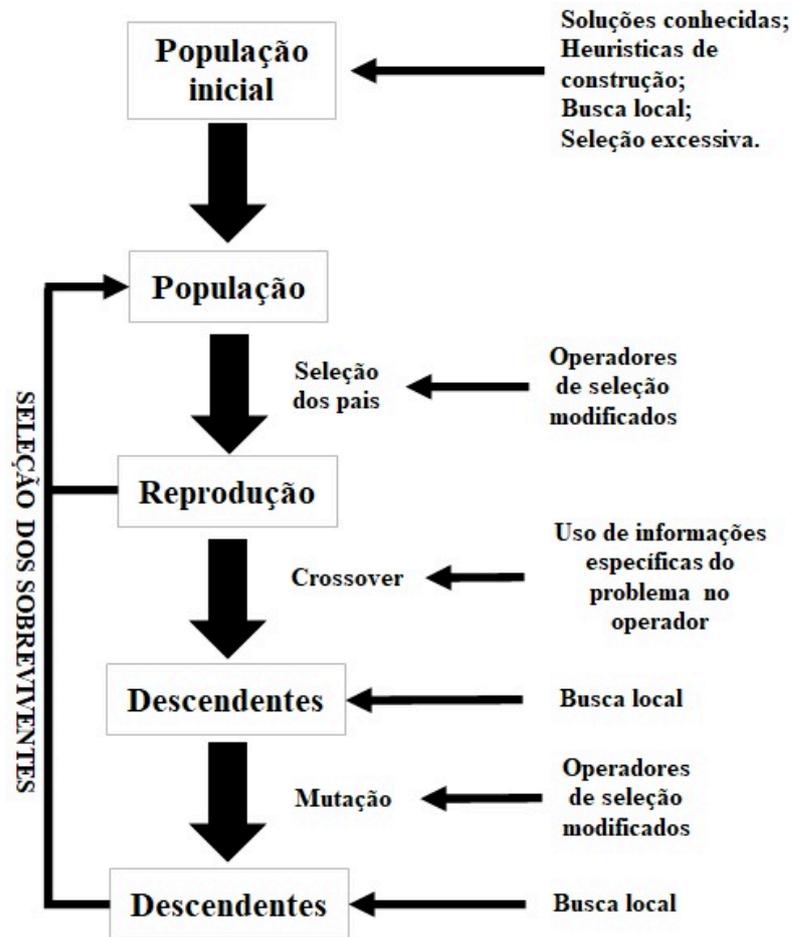
O último método leva a analogia natural do recozimento simulado, onde os piores movimentos podem ser aceitos com probabilidade não nula para ajudar a escapar de ótimos locais. Este método ajuda a resolver a questão da diversidade durante a fase de busca local, assim um vizinho menos adaptado pode ser aceito com uma probabilidade que aumenta exponencialmente conforme a faixa de valores *fitness* da população diminui, equação (28). Embora os trabalhos acima sejam restritos aos métodos que incorporam alguma forma de mecanismo de melhoria agindo no resultado das variantes dos operadores evolucionários, há vários outros modos em que algoritmos evolucionários ou algoritmos meméticos podem ser usados em conjunto com outros operadores e/ou conhecimento específico do domínio, isso é ilustrado na Figura 14.

3.7.2 INICIALIZAÇÃO INTELIGENTE

A maneira mais trivial pela qual o conhecimento existente sobre a estrutura de um problema ou potenciais soluções possam ser incorporadas dentro de um algoritmo evolucionário é na fase de inicialização. Em muitos casos o algoritmo evolutivo irá fazer um progresso inicial rápido, entretanto iniciar o algoritmo evolutivo pelo uso de soluções existentes pode oferecer benefícios interessantes (SURRY; RADCLIFFE, 1996):

- É possível prevenir o desperdício de esforço computacional podendo gerar aumento na eficiência (velocidade).
- Uma população inicial não aleatória pode direcionar a busca dentro de uma particular região do espaço de busca que contém boas soluções. A tendência da busca pode resultar no aumento da efetividade (qualidade da solução final).
- De modo geral, uma dada quantidade total de esforço computacional dividido sobre a inicialização da heurística e a busca evolucionária pode fornecer

Figura 14 – Possíveis locais para incorporar conhecimento ou outros operadores dentro de uma única geração.



Fonte: (HART, William E.; KRASNOGOR; SMITH, 2005).

melhores resultados do que gastar tudo em busca evolutiva “pura”, ou em uma heurística multistart equivalente.

Há um número de possíveis modos em que a função de inicialização pode ser mudada da simples criação de números aleatórios, tais como (SURRY; RADCLIFFE, 1996):

- Semear a população com uma ou mais boas soluções previamente conhecidas, decorrentes de outras técnicas.
- Na inicialização seletiva um grande número de soluções aleatórias são criadas, então a população inicial é selecionada desses. Outras alternativas incluem selecionar um conjunto baseado não apenas no *fitness*, mas também na diversidade e assim maximizar a cobertura do espaço de busca.
- Realizar uma busca local iniciando de cada membro da população inicial, de modo que a população inicial consiste de um conjunto de pontos que são

localmente ótimos em relação a algum operador de movimento.

- Usando um ou mais dos métodos acima para identificar um (ou possivelmente mais) boas soluções, e, em seguida, cloná-los e aplicar a mutação em uma alta taxa (mutação em massa) para produzir um número de indivíduos na vizinhança do ponto inicial.

Esses métodos tem sido testados e tendo exibido bom ganho de performance para certos problemas. Entretanto, a questão importante é fornecer ao algoritmo evolutivo uma diversidade suficiente para que a evolução possa ocorrer. Em Surry e Radcliffe (1996) foi examinado o efeito da variar a proporção da população inicial de um algoritmo genético que foi derivado de boas soluções. Suas conclusões foram:

- O uso de uma pequena proporção de soluções derivadas na população inicial ajudou a busca genética;
- À medida que a proporção foi aumentada, o desempenho médio foi melhorado;
- A melhor performance veio da população inicial mais aleatória.

Em outras palavras, à medida que a proporção das soluções derivadas das heurísticas usadas aumentaram, o mesmo aconteceu com o desempenho médio, porém a variação do desempenho diminuiu. Isso significa que não havia as execuções ocasionais realmente boas resultantes do algoritmo evolutivo pesquisando regiões completamente novas do espaço e surgindo novas soluções. Para certos tipos de problemas, esta é uma propriedade indesejável.

3.7.3 ALGORITMO HÍBRIDO PROPOSTO

No algoritmo híbrido proposto, ilustrado no fluxograma da Figura 15, a estrutura principal é construída através de um BRKGA com decodificador baseado na co-árvore da rede, e a busca local é introduzida na população através dos indivíduos mutantes, ou seja, ao invés de introduzir na população indivíduos mutantes gerados aleatoriamente são as soluções aceitas dentro da busca local do recozimento simulado que irão compor os indivíduos mutantes.

Além disso, para introduzir diversidade e manter as melhores características o critério de Metropolis é introduzido ao processo de reprodução. Assim, um indivíduo descendente do *crossover* irá compor a população $k + 1$ se seu *fitness* for menor do que o do progenitor não elite (caso da minimização), caso contrário, o indivíduo será aceito na população $k + 1$ com probabilidade de Metropolis, equação (28), onde a temperatura T é obtida da temperatura corrente do último SA realizado. Caso o indivíduo descendente não seja aceito no critério de Metropolis, o progenitor não elite da população k irá tomar o lugar deste indivíduo descendente. Por fim, uma

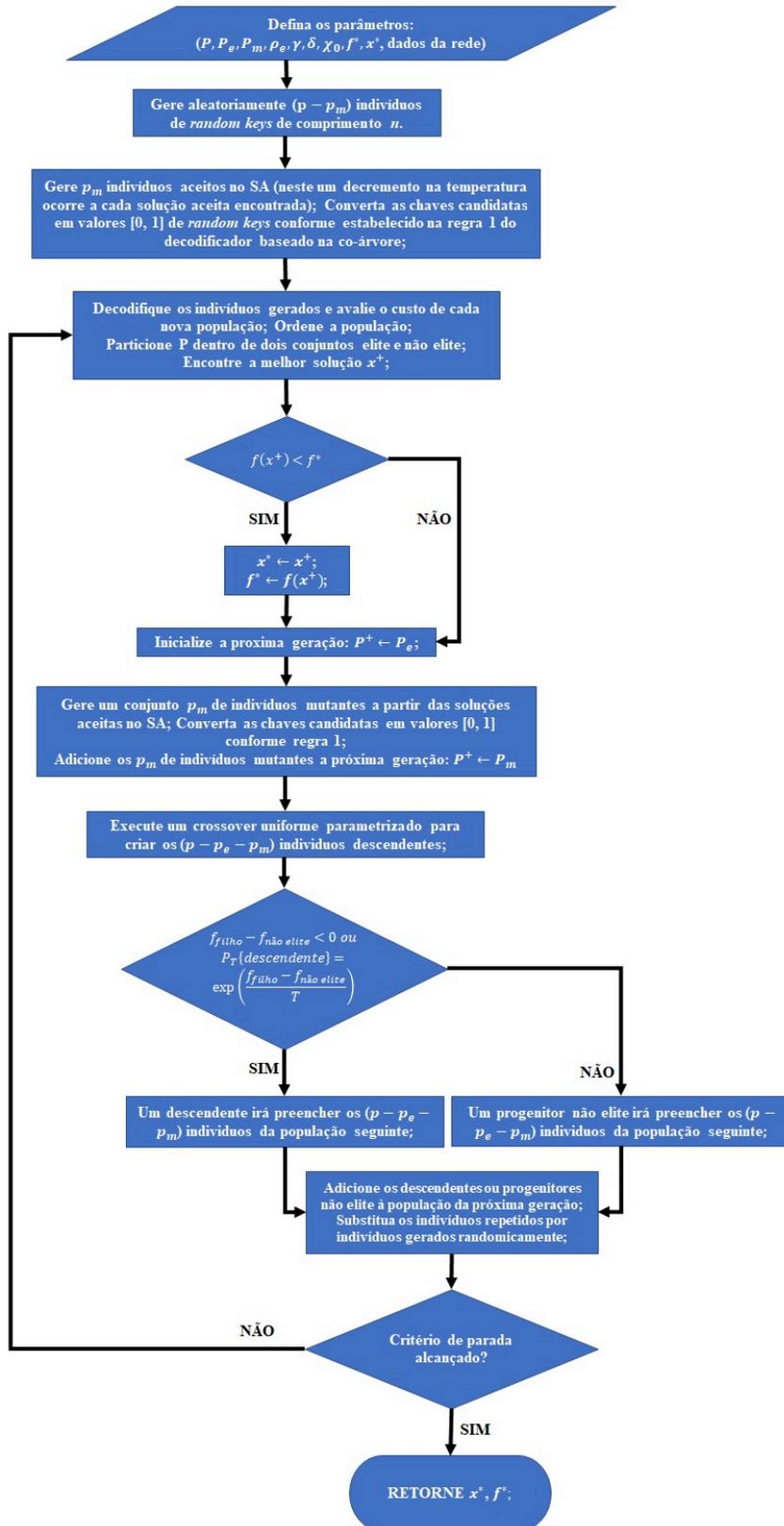
rotina simples para substituição de cromossomos repetidos na população corrente por indivíduos gerados aleatoriamente é implementada.

Na população inicial, testes com a população composta somente de indivíduos gerados aleatoriamente, população gerada das soluções aceitas do recozimento simulado e uma inicialização híbrida com diferentes proporções entre indivíduos gerados aleatoriamente e as soluções aceitas do recozimento simulado foram realizados, e as melhores soluções encontradas foram com a utilização da inicialização híbrida com $p - p_m$ indivíduos gerados aleatoriamente e p_m indivíduos gerados das soluções aceitas do recozimento simulado.

Assim, no método proposto a primeira população é criada através de um processo híbrido entre indivíduos gerados aleatoriamente e indivíduos gerados das soluções aceitas do recozimento simulado. Em seguida, na geração seguinte p_e soluções elite são copiadas diretamente (partição elite), $p - p_e - p_m$ indivíduos são gerados através do *crossover uniforme parametrizado* com a adição do critério de Metropolis para avaliar se o indivíduo da próxima população será o indivíduo descendente do processo de reprodução ou o pai não elite. Para preencher os p_m indivíduos restantes o refriamento simulado com a última solução corrente é realizado e todas as soluções aceitas irão preencher a partição da população mutante. Dentro do SA uma diminuição de temperatura é implementada a cada solução aceita, ao invés de ser a cada cadeia de Markov de comprimento L .

Por fim, soluções candidatas repetidas são substituídas por indivíduos aleatórios. Conforme será observado na próxima seção, esta metodologia obteve resultados de boa qualidade em todas as simulações, porém em sistemas teste de pequena dimensão o esforço computacional exigido foi maior do que as demais técnicas.

Figura 15 – Fluxograma do método híbrido BRKGA2/SA proposto.



Fonte: Elaborado pelo autor.

4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos através da implementação dos algoritmos abordados nas seções anteriores, os testes foram realizados em um computador DELL com processador Intel Core™ i5-10500 CPU@ 3.10 GHz e 16 GB de memória RAM. O *software* utilizado para a execução do fluxo de carga foi o Open Distribution System Simulator (OpenDSS) versão 9.3.0.2. Este *software* possibilita que muitas aplicações sejam realizadas através de *scripts* de texto padrões, que são executados a partir do painel de comando para o cálculo de perdas na distribuição. Este simulador pode ainda implementar um processo iterativo através de outros programas por meio da interface *Component Object Model* (COM) que comanda o OpenDSS nos casos onde a realização de um procedimento não é possível de ser implementado dentro do mesmo. Assim, esta interface COM é responsável pela implementação do processo iterativo utilizado para a redução de perdas técnicas através da reconfiguração de redes (CAPELI; QUEIROZ, L. M.; SOUSA, 2014).

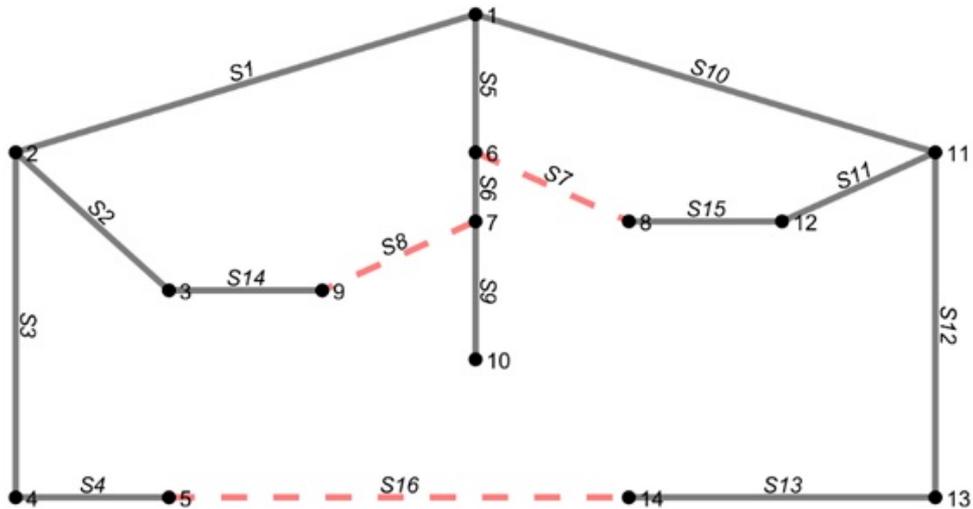
Este programa possui ainda um suporte ao programador em Visual Basic for Applications (VBA) que expõem as classes, propriedades e métodos para o usuário. Também possui um corretor de código, útil para o desenvolvimento do código para acessar o OpenDSS através da interface COM, desse modo, é possível primeiramente buscar um código em VBA e depois utilizá-lo no problema de reconfiguração no editor do MATLAB versão R2021a, utilizado nesta dissertação. Os testes foram realizados em cinco sistemas de distribuição bem conhecidos na literatura, as redes de distribuição de 14 barras apresentado em (CIVANLAR *et al.*, 1988), 33 barras apresentado em (BARAN; WU, 1989), 69 barras em (SAVIER; DAS, D., 2007), 84 barras em (CHIOU; CHANG, C.-F.; SU, 2005) e 136 barras proposto em (MANTOVANI; CASARI; ROMERO, 2000), os dados desses sistemas testes também podem ser encontrados no anexo A desta dissertação.

4.1 REDE DE DISTRIBUIÇÃO DE 14 BARRAS

Esse sistema possui 13 barras de carga e 16 ramos, dos quais 13 são chaves seccionadoras normalmente fechadas e 3 são chaves de interconexão normalmente abertas (chaves de laço). Na topologia original as chaves de laço que desconectam os ramos são as chaves S14, S15 e S16. O sistema originalmente possui uma tensão de operação de 23 kV e potência base de 100 MVA, as demandas totais de potências ativa e reativa são 28,7 MW e 17,3 MVar, respectivamente, com as perdas ativas em 513,715 kW e tensão mínima de 0,96828 p.u.. Após a realização da reconfiguração, todos os algoritmos propostos obtiveram a mesma topologia com as chaves abertas S7, S8 e S16, mostrada na Figura 16. Na Figura 17 pode-se notar uma melhoria do perfil de tensão da rede do sistema original para a nova configuração, gerada através

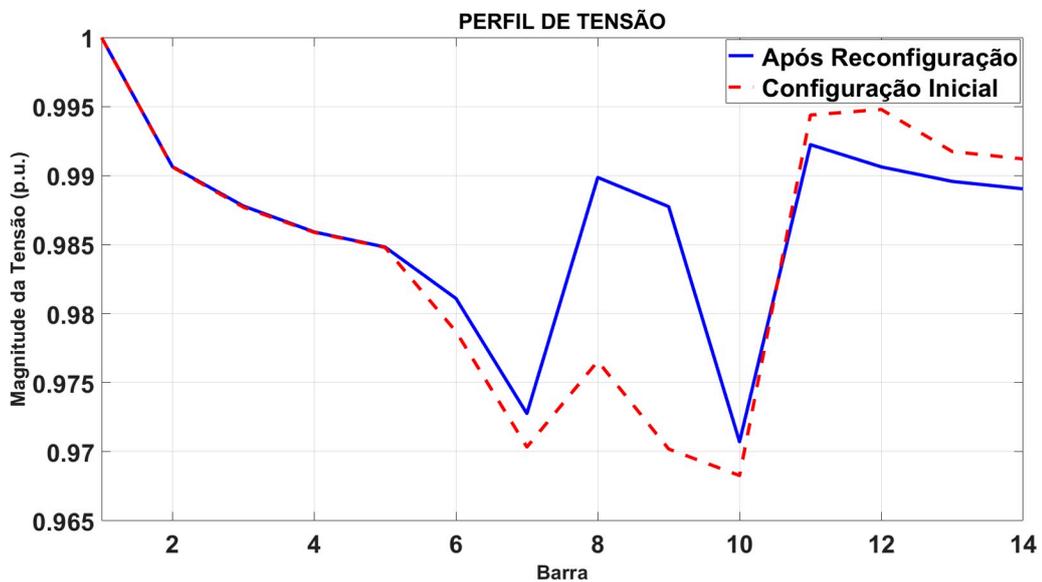
do procedimento de reconfiguração.

Figura 16 – Topologia ótima da rede 14 barras.



Fonte: Elaborado pelo autor.

Figura 17 – Perfil de tensão na rede 14 barras.



Fonte: Elaborado pelo autor.

Na Tabela 1 são mostrados os parâmetros utilizados na inicialização e critério de parada dos algoritmos propostos e na Tabela 2 são mostrados os parâmetros utilizados para os procedimentos de *restart* nos algoritmos BRKGA com decodificador Kruskal e decodificador baseado em co-árvore. Além disso, a semente geradora de números aleatórios também é especificada na Tabela 1, a escolha deste valor foi selecionada

ao acaso e mantida em todos os algoritmos de cada sistema teste para demonstrar o comportamento de cada método utilizado. Nesta dissertação, a escolha dos valores dos parâmetros das estratégias de *restart*, tiveram como regra geral, para decidir o valor desses parâmetros, a dimensão dos sistemas teste e o tamanho da população, assim, se a rede teste do problema de reconfiguração for simples, então κ , m_{limite} e th_{count} são configurados com valores baixos, conforme as redes se tornem mais complexas maiores serão esses valores.

Tabela 1 – Parâmetros de controle dos algoritmos propostos no sistema 14 barras.

Parâmetros	BRKGA/ Kruskal	BRKGA/ Co-árvore	SA	Híbrido BRKGA/SA
População	20	20	-	20
População elite	4	4	-	4
População mutante	2	2	-	2
Probabilidade (ρ_e)	0,7	0,7	-	0,7
Parâmetro de parada (ϵ_s)	-	-	1	-
Comprimento da cadeia de Markov	-	-	16	-
Semente geradora de números aleatórios	2315	2315	2315	2315

Fonte: Elaborado pelo autor.

A evolução das perdas da função objetivo bem como seu desvio de tensão ao longo das gerações ou iterações são mostrados nas Figuras 18 e 19, respectivamente. Nestas figuras pode-se notar que o BRKGA1 e o BRKGA2, em relação as suas versões com reinicialização, possuem curvas idênticas inicialmente e que decorridas algumas gerações uma alteração na curva é efetuada, isso ocorre devido a atuação da reinicialização, em alguns casos essa alteração na curva pode ser mais visível como no caso do BRKGA2 ilustrado na Figura 18. No caso do algoritmo híbrido pode-se notar na curva da evolução das perdas mostrada na Figura 18 que este método converge com menos iterações que os demais, devido ao acréscimo da busca local ao BRKGA2.

Por fim, na Tabela 3 pode-se observar o tempo de execução de cada algoritmo até encontrar a solução ótima. Como já referido em seções anteriores, o decodificador baseado em co-árvore possui um espaço de busca reduzido em relação ao decodificador baseado no algoritmo de Kruskal, apesar deste fato, o algoritmo demorou teve um tempo de execução superior, isto ocorre em razão da semente geradora gerar melhores soluções para um decodificador e piores para o outro. Em outros testes com diferentes valores de semente geradora o algoritmo baseado em co-árvore obteve solu-

ções com um esforço computacional significativamente menor em relação ao algoritmo de Kruskal. Em relação ao algoritmo híbrido proposto pode-se notar um maior valor no tempo de execução, apesar de encontrar a solução ótima na segunda geração, devido a melhoria no processo de busca ocorrido com o acréscimo da busca local.

Tabela 2 – Parâmetros de reinicialização aplicados ao BRKGA1 e BRKGA2 na rede 14 barras.

Parâmetros	BRKGA1		BRKGA2	
	Restart sem Elitismo	Restart com Elitismo	Restart sem Elitismo	Restart com Elitismo
κ	2	-	2	-
m	-	2	-	2
th_{count}	-	2	-	2

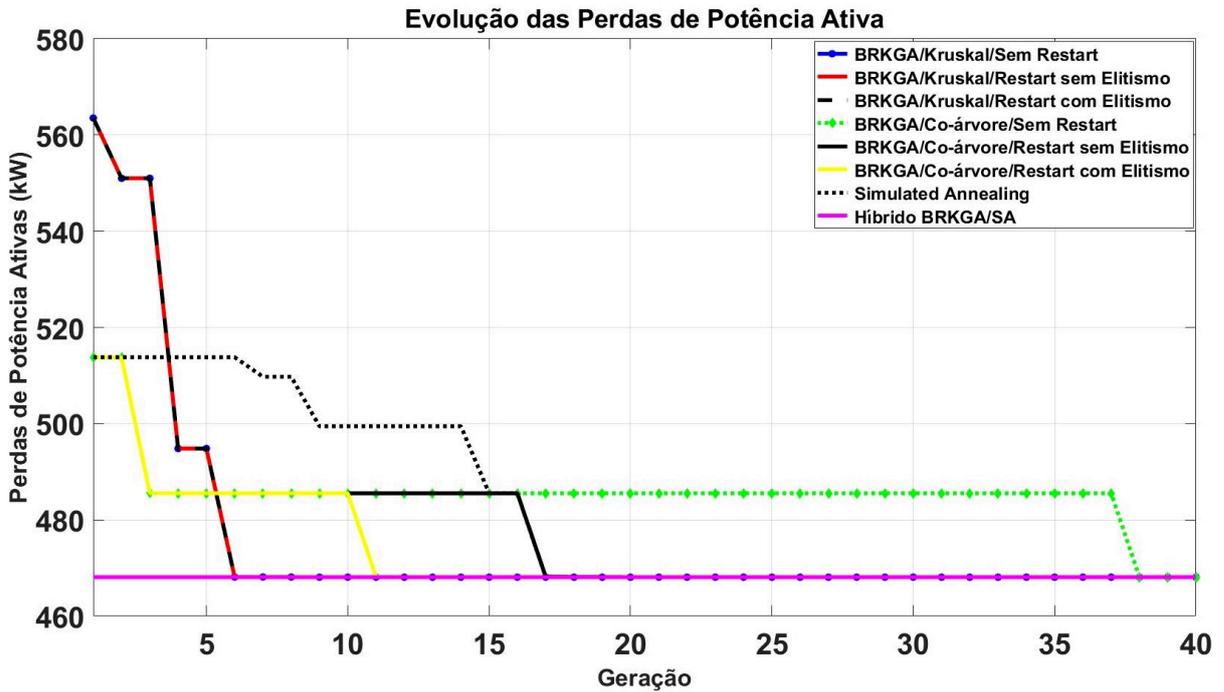
Fonte: Elaborado pelo autor.

Tabela 3 – Resultados das metodologias propostas no sistema 14 barras.

Algoritmo	Chaves abertas	Gerações/ Iterações até a solução	Tempo de Execução (seg.)	Perdas (kW)	Tensão mínima (p.u.)
BRKGA1/ Sem Restart	S7, S8, S16	6	0,379254	468,1031	0,970722
BRKGA1/ Restart sem Elitismo	S7, S8, S16	6	0,364311	468,1031	0,97072
BRKGA1/ Restart com Elitismo	S7, S8, S16	4	0,339647	468,1796	0,970721
BRKGA2/ Sem Restart	S7, S8, S16	38	0,854193	468,0936	0,960593
BRKGA2/ Restart sem Elitismo	S7, S8, S16	17	0,581096	468,0848	0,970717
BRKGA2/ Restart com Elitismo	S7, S8, S16	12	0,435017	468,0936	0,970724
Simulated Annealing	S7, S8, S16	160	1,595347	468,0878	0,970724
Híbrido BRKGA2/ SA	S7, S8, S16	2	0,751501	468,0848	0,970721

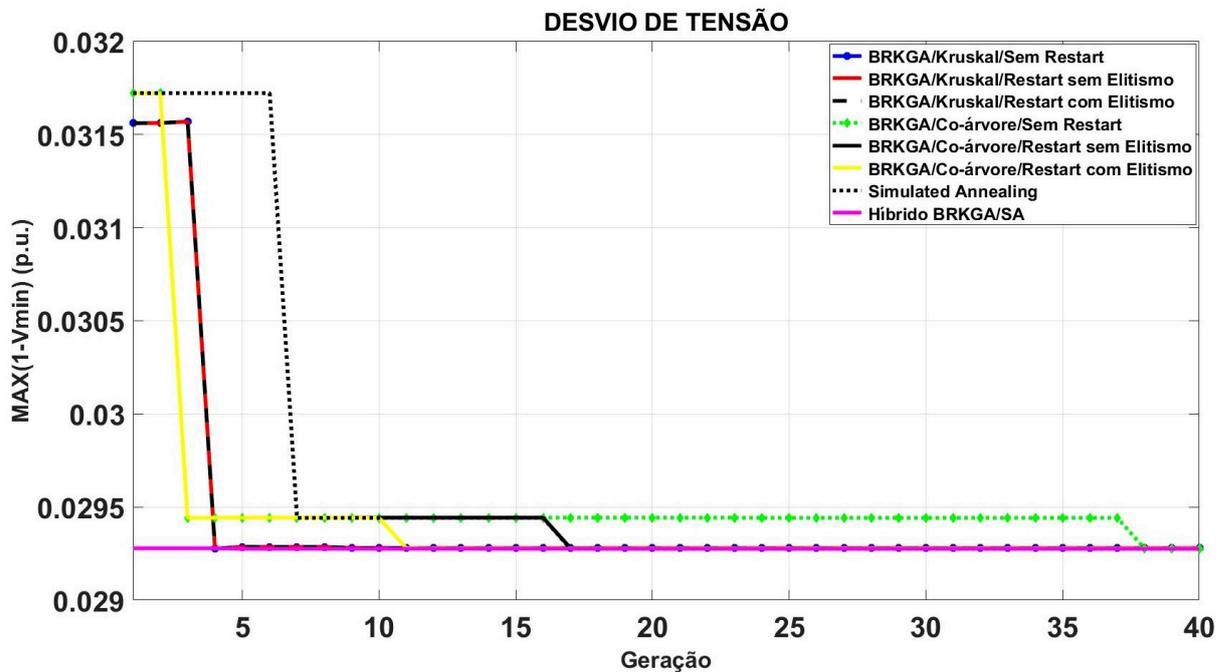
Fonte: Elaborado pelo autor.

Figura 18 – Evolução das perdas na rede 14 barras.



Fonte: Elaborado pelo autor.

Figura 19 – Evolução do desvio de tensão na rede 14 barras.

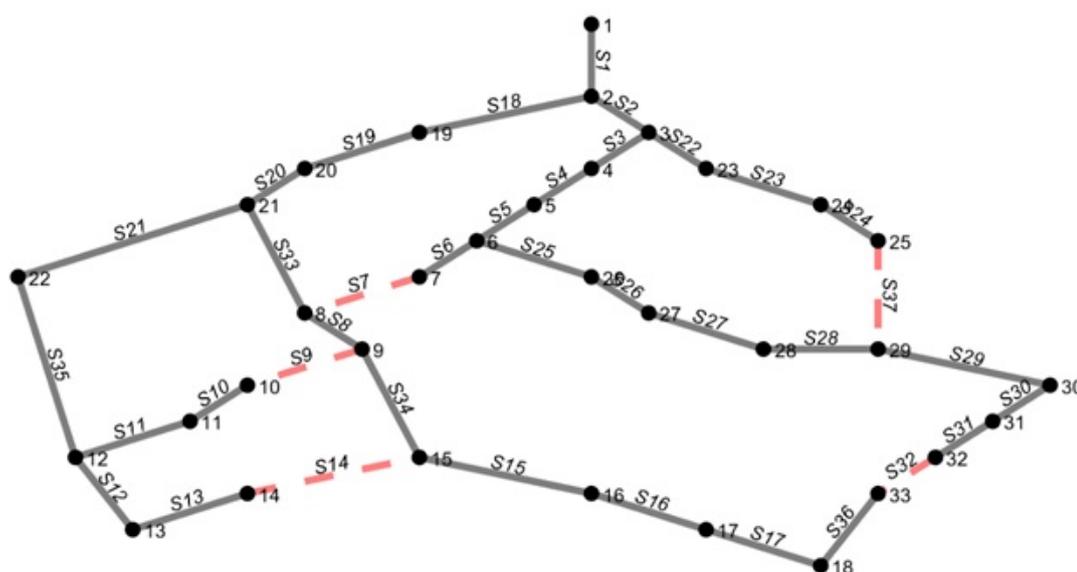


Fonte: Elaborado pelo autor.

4.2 REDE DE DISTRIBUIÇÃO DE 33 BARRAS

Esse sistema possui 32 barras de carga e 37 ramos que conectam as 32 barras, das quais 32 ramos são chaves seccionadoras normalmente fechadas e 5 são chaves de interconexão normalmente abertas. Na topologia original as chaves de laço que desconectam os ramos são S33, S34, S35, S36 e S37. Esse sistema possui uma tensão de operação de 12,66 kV e potência base de 100 MVA, as demandas totais de potências ativa e reativa são 3715 kW e 2300 kVAr, respectivamente, com as perdas ativas em 201,44 kW e tensão mínima de 0,91344 p.u.. Após a realização da reconfiguração, todos os algoritmos propostos obtiveram a mesma topologia com as chaves abertas S7, S9, S14, S32 e S37, ilustrada na Figura 20. Na Figura 21 é mostrada a melhoria do perfil de tensão da nova configuração em relação ao sistema original.

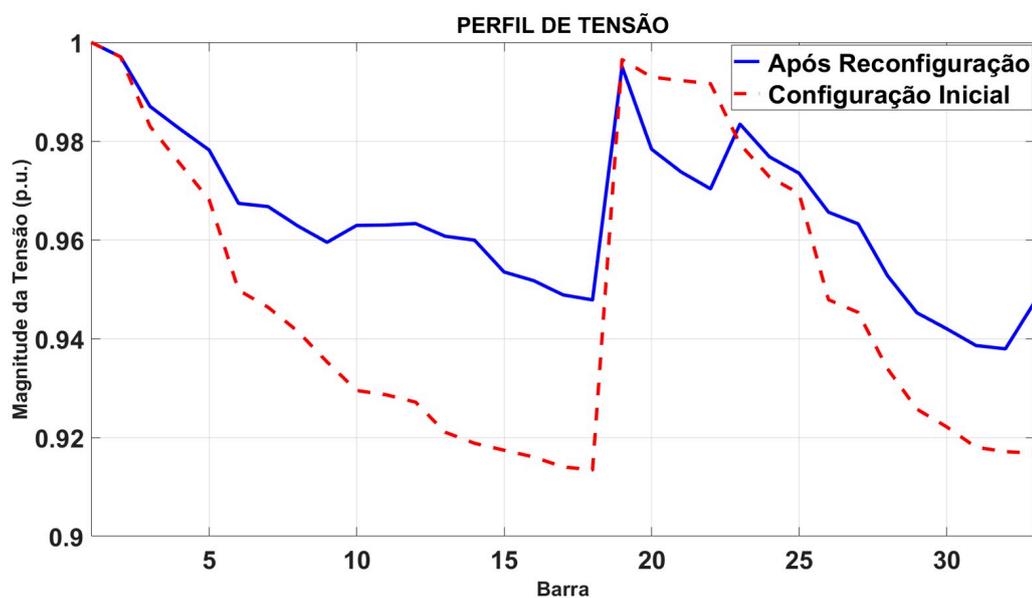
Figura 20 – Topologia ótima da rede 33 barras.



Fonte: Elaborado pelo autor.

Na Tabela 4 são mostrados os parâmetros utilizados na inicialização, semente geradora e critério de parada dos algoritmos propostos e na Tabela 5 são mostrados os parâmetros utilizados para os procedimentos de *restart* nos algoritmos BRKGA1 e BRKGA2. Novamente, como no caso da rede 14 barras, devido a semente escolhida temos o BRKGA com decodificador baseado no algoritmo de Kruskal sem a possibilidade de que a reinicialização possa minimizar o esforço computacional, conduzindo o algoritmo ao mesmo tempo de execução.

Figura 21 – Perfil de tensão na rede 33 barras.



Fonte: Elaborado pelo autor.

Tabela 4 – Parâmetros de controle dos algoritmos propostos no sistema 33 barras.

Parâmetros	BRKGA/ Kruskal	BRKGA/ Co-árvore	SA	Híbrido BRKGA/SA
População	50	50	-	50
População elite	10	10	-	10
População mutante	5	5	-	5
Probabilidade (ρ_e)	0,7	0,7	-	0,7
Parâmetro de parada (ϵ_s)	-	-	0,2	-
Comprimento da cadeia de Markov	-	-	37	-
Semente geradora de números aleatórios	103501	103501	103501	103501

Fonte: Elaborado pelo autor.

Analisando o tempo de execução na Tabela 6 dos algoritmos propostos, pode-se observar as mesmas características do sistema anterior em relação as estratégias de *restart*, contudo, em relação ao SA e ao algoritmo híbrido há um considerável aumento no esforço computacional.

Tabela 5 – Parâmetros de reinicialização aplicados ao BRKGA1 e BRKGA2 na rede 33 barras.

Parâmetros	BRKGA1		BRKGA2	
	Restart sem Elitismo	Restart com Elitismo	Restart sem Elitismo	Restart com Elitismo
κ	2	-	2	-
m	-	2	-	1
th_{count}	-	2	-	1

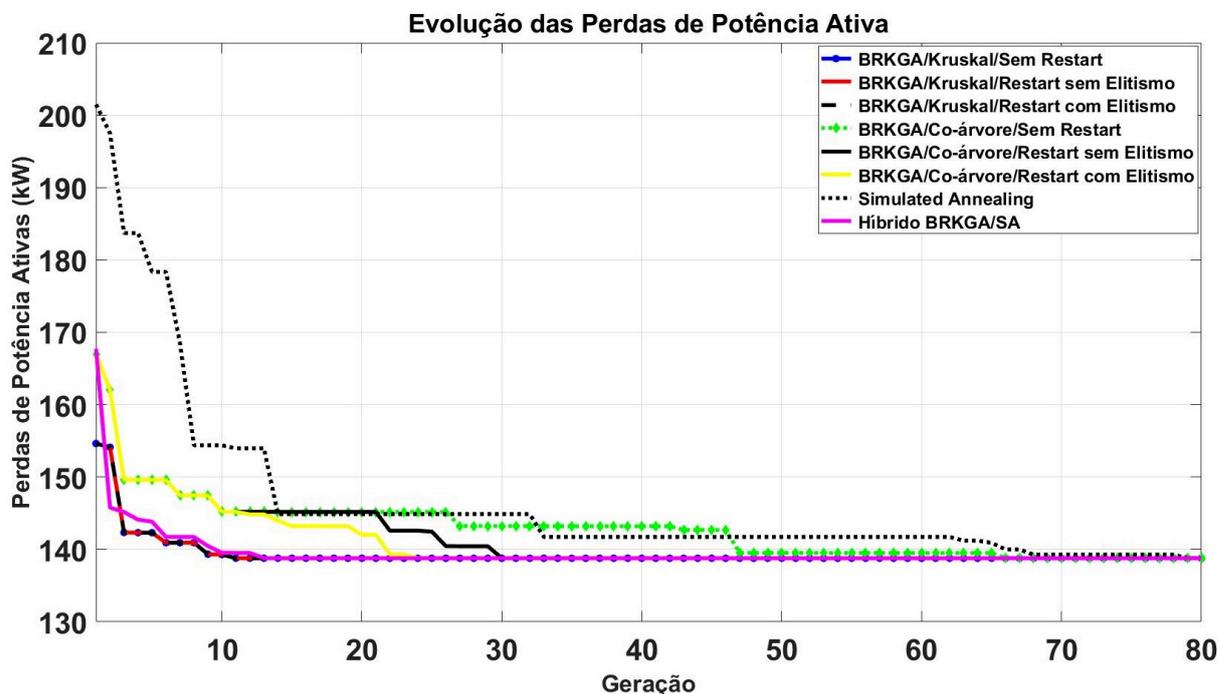
Fonte: Elaborado pelo autor.

Tabela 6 – Resultados das metodologias propostas no sistema 33 barras.

Algoritmo	Chaves abertas	Gerações/Iterações até a solução	Tempo de Execução (seg.)	Perdas (kW)	Tensão mínima (p.u.)
BRKGA1/ Sem Restart	S7, S9, S14, S32, S37	11	1,285397	138,7782	0,937942
BRKGA1/ Restart sem Elitismo	S7, S9, S14, S32, S37	11	1,282223	138,7867	0,937942
BRKGA1/ Restart com Elitismo	S7, S9, S14, S32, S37	11	1,277141	138,7884	0,937942
BRKGA2/ Sem Restart	S7, S9, S14, S32, S37	66	4,952422	138,7755	0,937935
BRKGA2/ Restart sem Elitismo	S7, S9, S14, S32, S37	30	2,510883	138,7884	0,937942
BRKGA2/ Restart com Elitismo	S7, S9, S14, S32, S37	24	2,12021	138,8004	0,937942
Simulated annealing	S7, S9, S14, S32, S37	666	6,810549	138,7957	0,937945
Híbrido BRKGA2/ SA	S7, S9, S14, S32, S37	13	7,217844	138,7883	0,937945

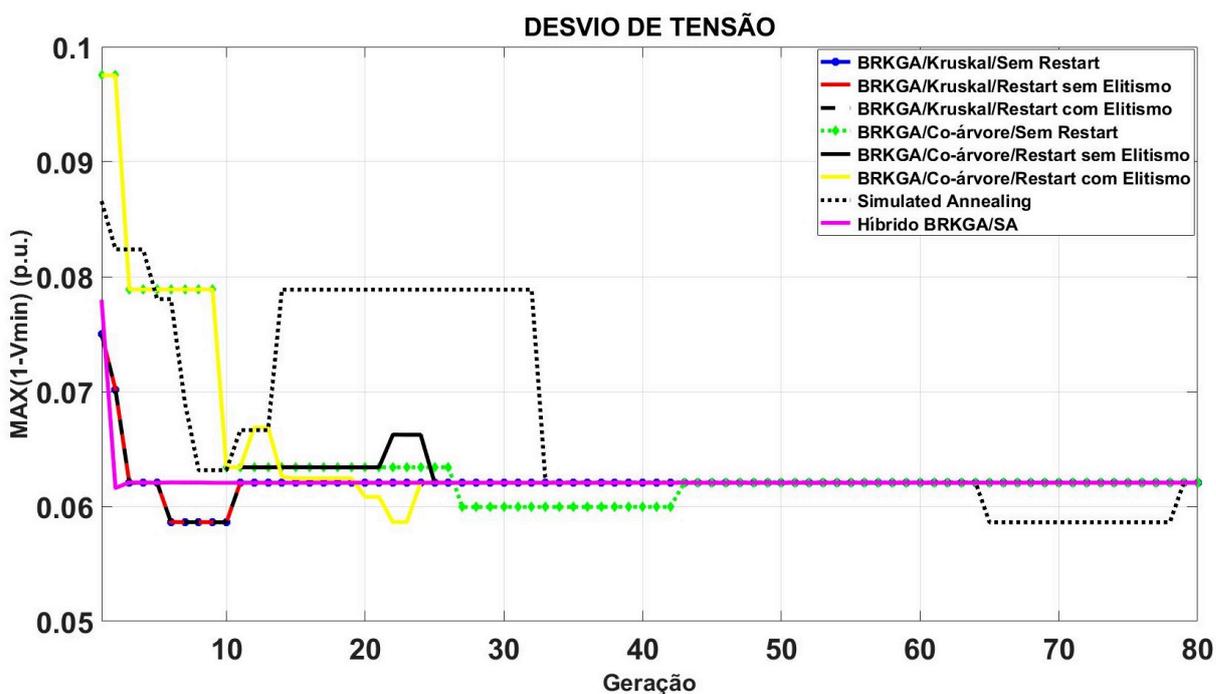
Fonte: Elaborado pelo autor.

Figura 22 – Evolução das perdas na rede 33 barras.



Fonte: Elaborado pelo autor.

Figura 23 – Evolução do desvio de tensão na rede 33 barras.

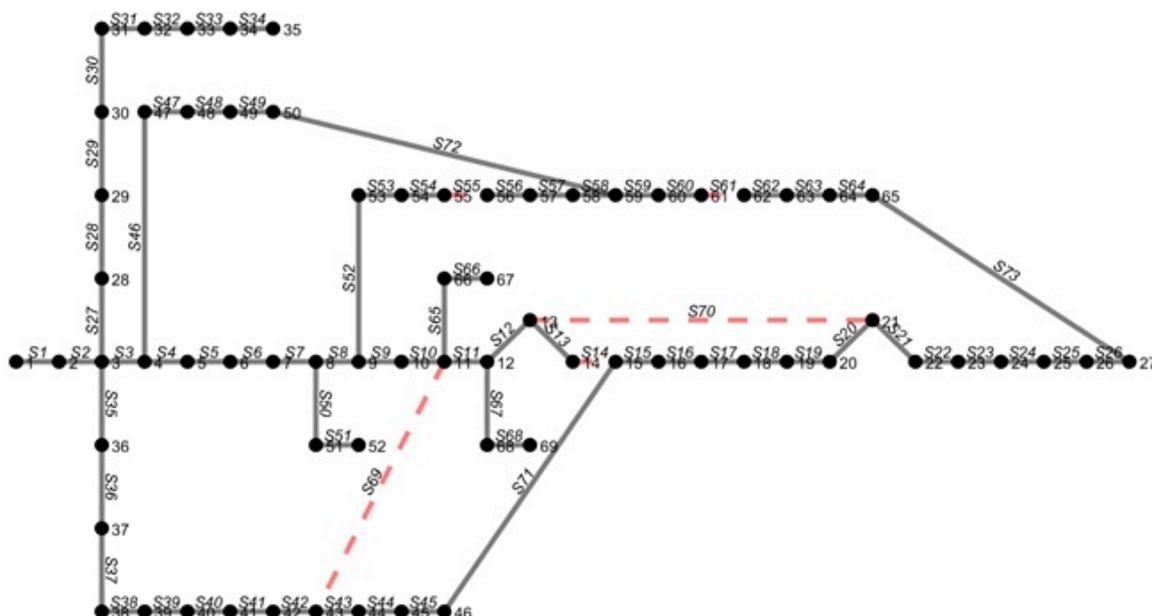


Fonte: Elaborado pelo autor.

4.3 REDE DE DISTRIBUIÇÃO DE 69 BARRAS

O sistema possui 69 barras, sendo 68 barras de carga, 1 subestação e 73 ramos, dos quais 68 são chaves seccionadoras normalmente fechadas e 5 são chaves de interconexão normalmente abertas. Na topologia original as chaves de laço que desconnectam os ramos são S69, S70, S71, S72 e S73. O sistema possui uma tensão de operação de 12,66 kV e potência base de 100 MVA, as demandas totais de potências ativa e reativa são 3802,19 kW e 2694,6 kVAr, respectivamente, com as perdas ativas em 223,521 kW e tensão mínima de 0,90938 p.u.. Após a realização da reconfiguração, todos os algoritmos propostos obtiveram a mesma topologia mostrada na Figura 24, obtendo as chaves abertas S14, S55, S61, S69 e S70. Na Figura 25 novamente é observado uma melhoria no perfil de tensão em relação ao sistema original.

Figura 24 – Topologia ótima da rede 69 barras.

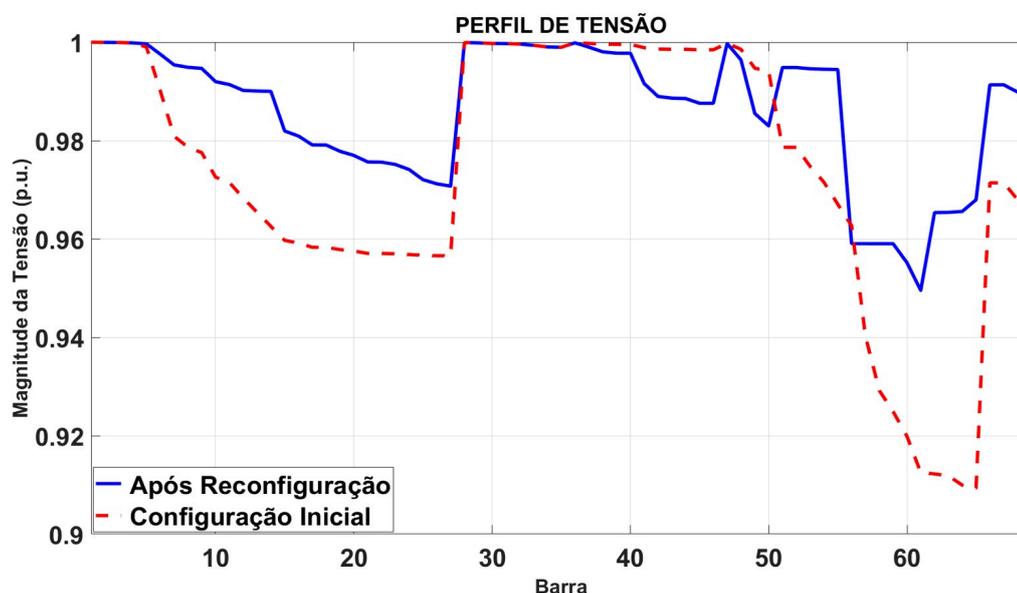


Fonte: Elaborado pelo autor.

Na Tabela 7 são mostrados os parâmetros iniciais utilizados nas simulações para o sistema de 69 barras e na Tabela 8 estão os parâmetros utilizados nos processos de reinicialização do BRKGA1 e BRKGA2, respectivamente. A evolução das perdas ativas e desvio de tensão nesta rede são mostrados nas Figuras 26 e 27, nesta ordem. Pode-se notar na Figura 26 as mesmas características obtidas nos sistemas anteriores em relação aos procedimentos de reinicialização, assim, novamente a reinicialização com elitismo se mostrou mais eficiente na rede teste, conforme pode-se notar nos gráficos e na iteração em que as soluções foram encontradas, conforme ilustra a Tabela 9.

Como observado na Tabela 9, diferentemente dos casos anteriores, o BRKGA2 obteve um melhor desempenho em relação ao BRKGA1, uma vez que o primeiro possui

Figura 25 – Perfil de tensão na rede 69 barras.



Fonte: Elaborado pelo autor.

um espaço de busca reduzido, seria esperado uma redução no tempo de simulação. No caso do método híbrido apesar da curva de evolução das perdas encontrar a solução ótima em menos iterações, ainda há um valor elevado do tempo de execução em relação aos demais métodos.

Tabela 7 – Parâmetros de controle dos algoritmos propostos no sistema 69 barras.

Parâmetros	BRKGA/ Kruskal	BRKGA/ Co-árvore	SA	Híbrido BRKGA/SA
População	100	100	-	100
População elite	20	20	-	20
População mutante	10	10	-	10
Probabilidade (ρ_e)	0,7	0,7	-	0,7
Parâmetro de parada (ϵ_s)	-	-	0,02	-
Comprimento da cadeia de Markov	-	-	73	-
Semente geradora de números aleatórios	31546	31546	31546	31546

Fonte: Elaborado pelo autor.

Tabela 8 – Parâmetros de reinicialização aplicados ao BRKGA1 e BRKGA2 na rede 69 barras.

Parâmetros	BRKGA1		BRKGA2	
	Restart sem Elitismo	Restart com Elitismo	Restart sem Elitismo	Restart com Elitismo
κ	7	-	6	-
m	-	2	-	2
th_{count}	-	3	-	3

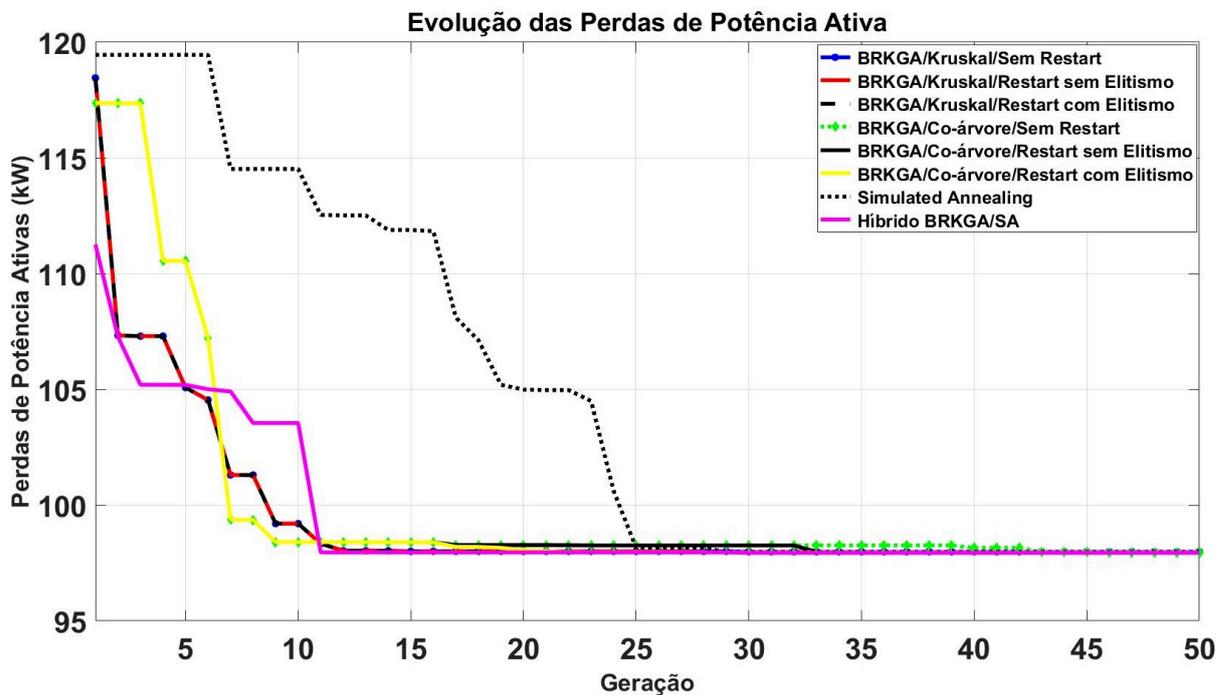
Fonte: Elaborado pelo autor.

Tabela 9 – Resultados das metodologias propostas no sistema 69 barras.

Algoritmo	Chaves abertas	Gerações/Iterações até a solução	Tempo de Execução (seg.)	Perdas (kW)	Tensão mínima (p.u.)
BRKGA1/ Sem Restart	S14, S55, S61, S69, S70	133	37,474425	97,93962	0,94954
BRKGA1/ Restart sem Elitismo	S14, S55, S61, S69, S70	34	10,520165	97,94633	0,94954
BRKGA1/ Restart com Elitismo	S14, S55, S61, S69, S70	23	7,189757	97,98538	0,949541
BRKGA2/ Sem Restart	S14, S55, S61, S69, S70	54	14,339952	97,94633	0,949544
BRKGA2/ Restart sem Elitismo	S14, S55, S61, S69, S70	39	10,483804	97,94633	0,949544
BRKGA2/ Restart com Elitismo	S14, S55, S61, S69, S70	27	7,284688	97,97332	0,949541
Simulated Annealing	S14, S55, S61, S69, S70	876	11,310879	97,97099	0,949542
Híbrido BRKGA2/ SA	S14, S55, S61, S69, S70	11	17,655326	97,97493	0,949542

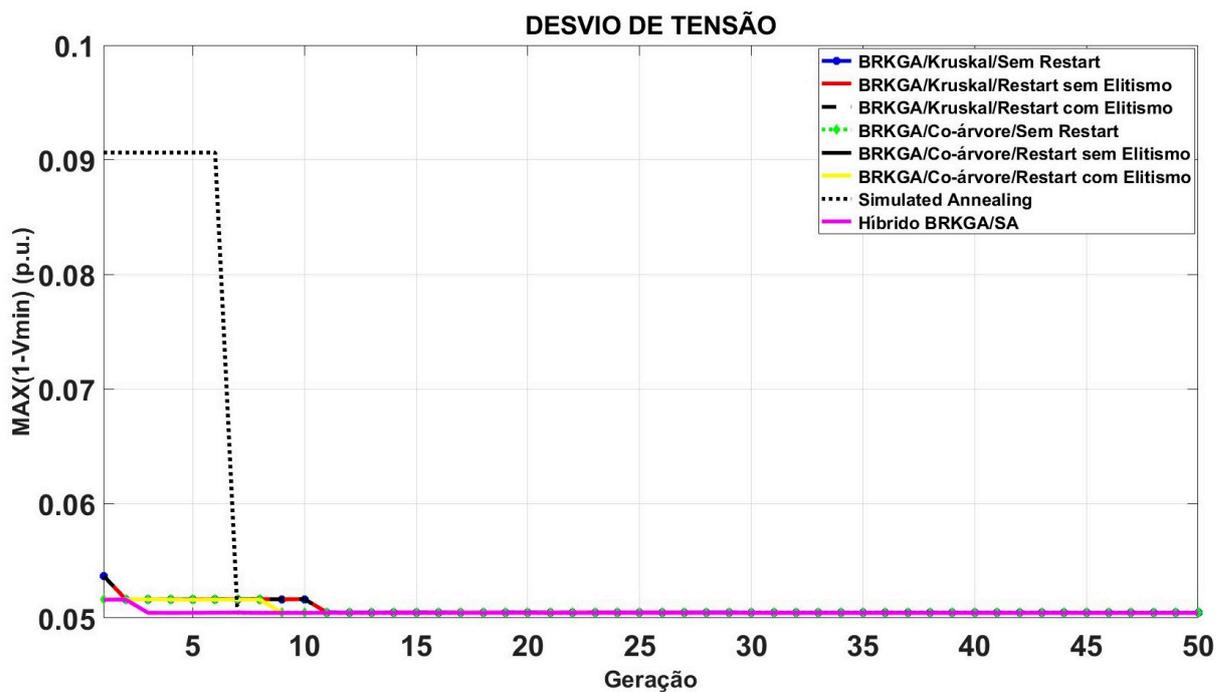
Fonte: Elaborado pelo autor.

Figura 26 – Evolução das perdas na rede 69 barras.



Fonte: Elaborado pelo autor.

Figura 27 – Evolução do desvio de tensão na rede 69 barras.

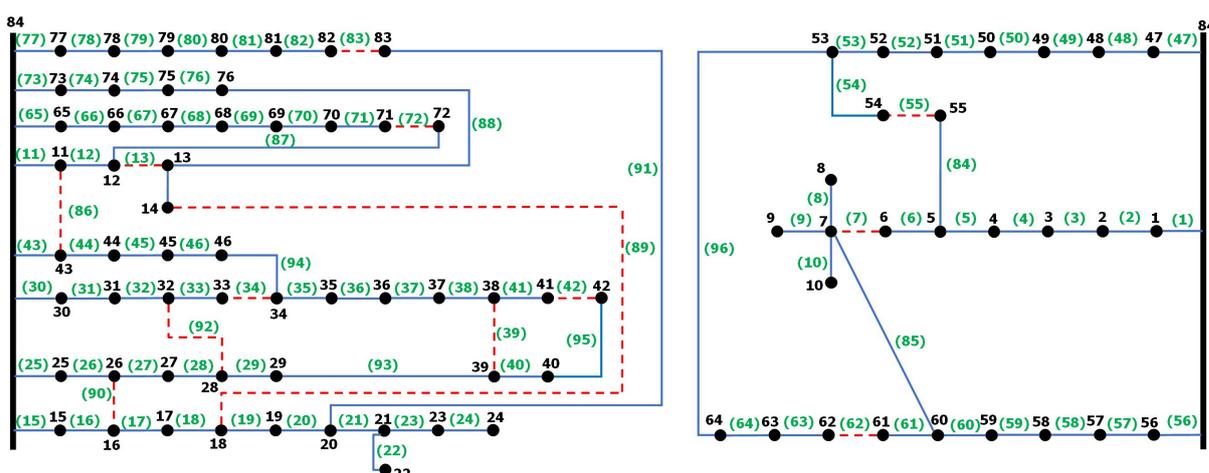


Fonte: Elaborado pelo autor.

4.4 REDE DE DISTRIBUIÇÃO DE 84 BARRAS

Este sistema teste possui 84 barras, sendo 83 barras de carga, 2 subestações e 96 ramos, dos quais 83 são chaves seccionadoras normalmente fechadas e 13 são chaves de interconexão normalmente abertas. Na topologia original as chaves de laço que desconectam os ramos são S84, S85, S86, S87, S88, S89, S90, S91, S92, S93, S94, S95 e S96. A tensão na subestação é 11,40 kV e a potência base 100 MVA, as demandas totais de potências ativa e reativa são 28,351 MW e 20,700 MVar, respectivamente, as perdas ativas e a tensão mínima antes da reconfiguração são 531,722 kW e 0,92855 p.u., nesta ordem. Após a realização da reconfiguração, todos os algoritmos propostos obtiveram a mesma topologia com as chaves abertas S7, S13, S34, S39, S42, S55, S62, S72, S83, S86, S89, S90 e S92, conforme mostra a Figura 28. Na Figura 29 pode-se observar a melhoria no perfil de tensão em relação ao sistema original.

Figura 28 – Topologia ótima da rede 84 barras.

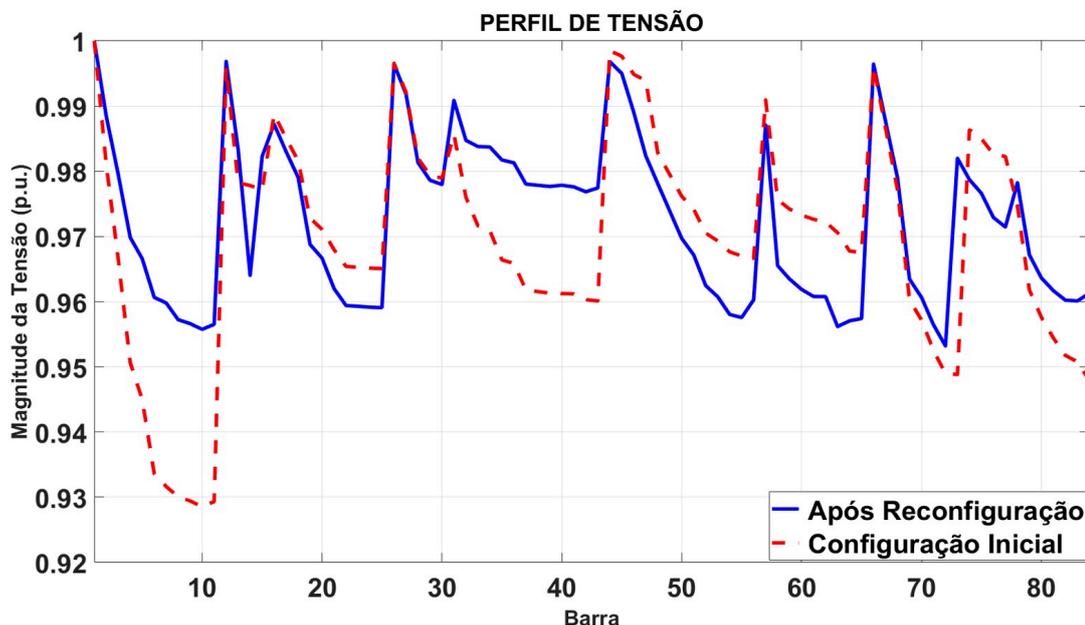


Fonte: Elaborado pelo autor.

Na Tabela 10 são mostrados os parâmetros utilizados na inicialização dos métodos propostos para as simulações do sistema teste 84 barras e, na Tabela 11 são mostrados os parâmetros utilizados para os procedimentos de *restart* nos algoritmos BRKGA1 e BRKGA2, nesta ordem. Conforme mostrado na Tabela 10, os valores da população dos algoritmos genéticos permaneceram os mesmos, em relação a rede 69 barras, pois um aumento na população a partir deste ponto poderia aumentar consideravelmente o esforço computacional. Como nos casos anteriores as estratégias de reinicialização possibilitaram a diminuição do tempo de execução dos algoritmos BRKGA1 e BRKGA2.

Analisando o tempo de execução dos algoritmos propostos na Tabela 12, pode-se observar algumas mudanças em relação aos sistemas anteriores, para este sistema

Figura 29 – Perfil de tensão na rede 84 barras.



Fonte: Elaborado pelo autor.

Tabela 10 – Parâmetros de controle dos algoritmos propostos no sistema 84 barras.

Parâmetros	BRKGA/ Kruskal	BRKGA/ Co-árvore	SA	Híbrido BRKGA/SA
População	100	100	-	100
População elite	20	20	-	20
População mutante	10	10	-	10
Probabilidade (ρ_e)	0,7	0,7	-	0,7
Parâmetro de parada (ϵ_s)	-	-	2	-
Comprimento da cadeia de Markov	-	-	96	-
Semente geradora de números aleatórios	124578	124578	124578	124578

Fonte: Elaborado pelo autor.

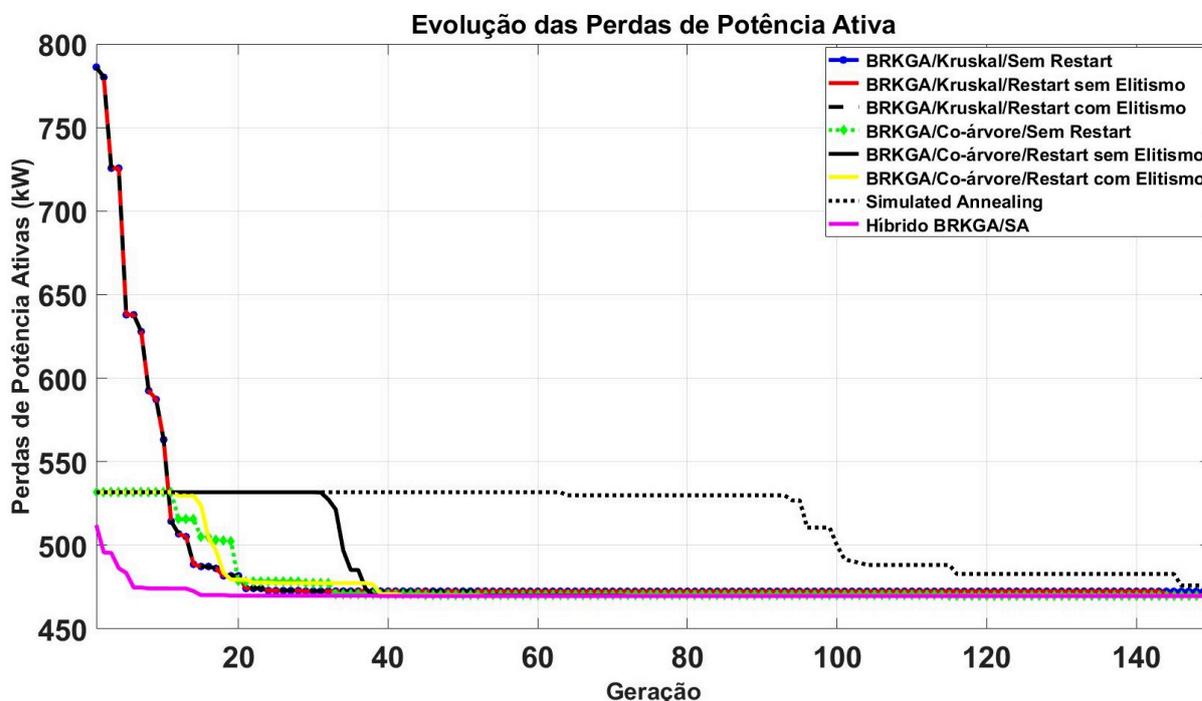
teste o BRKGA2 teve um desempenho melhor em relação ao BRKGA1 devido ao espaço de busca reduzido, além disso, as variantes com estratégias de reinicialização tiveram os valores do tempo de execução reduzidos e novamente a estratégia de *restart* com elitismo se mostrou mais eficiente. Quanto ao algoritmo híbrido, houve uma considerável diminuição no tempo de execução em relação as demais técnicas de solução.

Tabela 11 – Parâmetros de reinicialização aplicados ao BRKGA1 e BRKGA2 na rede 84 barras.

Parâmetros	BRKGA1		BRKGA2	
	Restart sem Elitismo	Restart com Elitismo	Restart sem Elitismo	Restart com Elitismo
κ	28	-	10	-
m	-	10	-	6
th_{count}	-	10	-	8

Fonte: Elaborado pelo autor.

Figura 30 – Evolução das perdas na rede 84 barras.



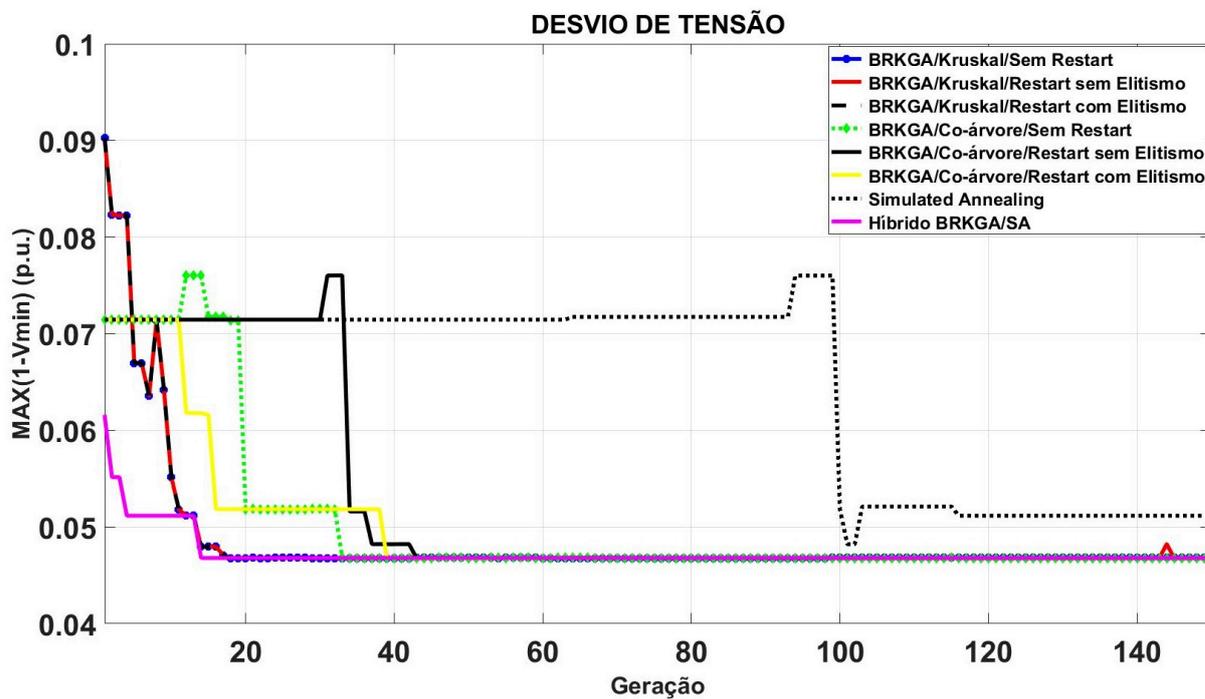
Fonte: Elaborado pelo autor.

Tabela 12 – Resultados das metodologias propostas no sistema 84 barras.

Algoritmo	Chaves abertas	Gerações/ Iterações até a solução	Tempo de Execução (seg.)	Perdas (kW)	Tensão mínima (p.u.)
BRKGA1/ Sem Restart	S7, S13, S34, S39, S42, S55, S62, S72, S83, S86, S89, S90, S92	206	78,105002	469,6674	0,953198
BRKGA1/ Restart sem Elitismo	S7, S13, S34, S39, S42, S55, S62, S72, S83, S86, S89, S90, S92	150	57,539766	469,6658	0,953198
BRKGA1/ Restart com Elitismo	S7, S13, S34, S39, S42, S55, S62, S72, S83, S86, S89, S90, S92	67	25,788045	469,6683	0,953198
BRKGA2/ Sem Restart	S7, S13, S34, S39, S42, S55, S62, S72, S83, S86, S89, S90, S92	115	41,639967	469,6655	0,953198
BRKGA2/ Restart sem Elitismo	S7, S13, S34, S39, S42, S55, S62, S72, S83, S86, S89, S90, S92	72	27,270545	469,6649	0,953198
BRKGA2/ Restart com Elitismo	S7, S13, S34, S39, S42, S55, S62, S72, S83, S86, S89, S90, S92	53	20,42272	469,6665	0,953198
Simulated Annealing	S7, S13, S34, S39, S42, S55, S62, S72, S83, S86, S89, S90, S92	7200	95,789313	469,6629	0,953199
Híbrido BRKGA2/ SA	S7, S13, S34, S39, S42, S55, S62, S72, S83, S86, S89, S90, S92	39	80,941191	469,6651	0,953199

Fonte: Elaborado pelo autor.

Figura 31 – Evolução do desvio de tensão na rede 84 barras.



Fonte: Elaborado pelo autor.

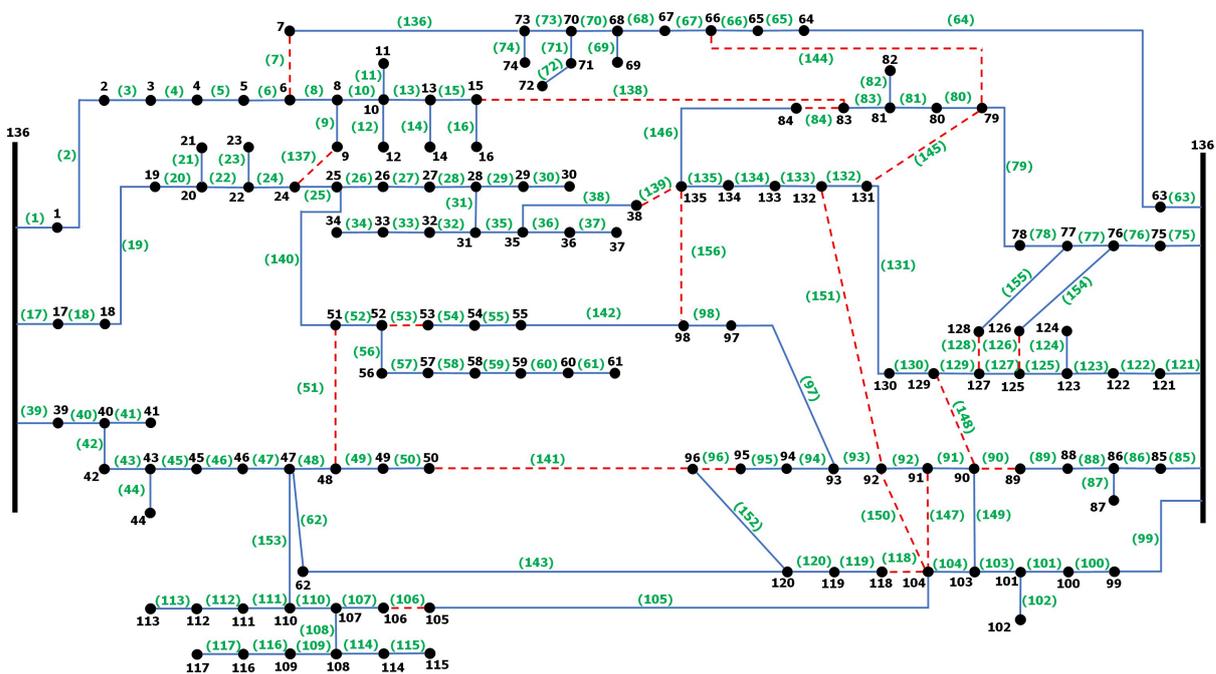
4.5 REDE DE DISTRIBUIÇÃO DE 136 BARRAS

Este sistema teste possui 136 barras, sendo 135 barras de carga, 2 subestações e 156 ramos, dos quais 135 são chaves seccionadoras normalmente fechadas e 21 são chaves de interconexão normalmente abertas. Na topologia original as chaves de laço que desconectam os ramos são S136, S137, S138, S139, S140, S141, S142, S143, S144, S145, S146, S147, S148, S149, S150, S151, S152, S153, S154, S155 e S156. A tensão na subestação é 11,40 kV e potência base de 100 MVA, as demandas totais de potências ativa e reativa são 18313,809 kW e 7932,533 kVAr, respectivamente, as perdas de potência ativa e a tensão mínima antes da reconfiguração são 319,897 kW e 0,93078 p.u., nesta ordem.

Na Tabela 13 são mostrados os parâmetros utilizados de inicialização, semente geradora e critério de parada usados nas simulações do sistema teste 136 barras. Na Tabela 14 são mostrados os parâmetros utilizados para as estratégias de *restart* nos algoritmos BRKGA1 e BRKGA2. Como nos casos anteriores as estratégias de reinicialização possibilitaram a redução do tempo de execução dos algoritmos BRKGA1 e BRKGA2, com melhor desempenho atribuído à reinicialização com elitismo. Após a realização da reconfiguração, quase todos os algoritmos propostos obtiveram a mesma topologia mostrada na Figura 32, com exceção do *simulated annealing*, conforme pode ser observado na Tabela 15 apenas algumas poucas chaves abertas são diferentes, porém os resultados foram muito próximos aos dos demais métodos. Quanto as pequenas diferenças de resultados apresentadas nos demais casos, estes ocorrem devido aos métodos de aproximação do OpenDSS que podem obter uma pequena diferença no resultados, mesmo obtendo as mesmas chaves na solução. Na Figura 33 é observada a melhoria no perfil de tensão em relação ao sistema original.

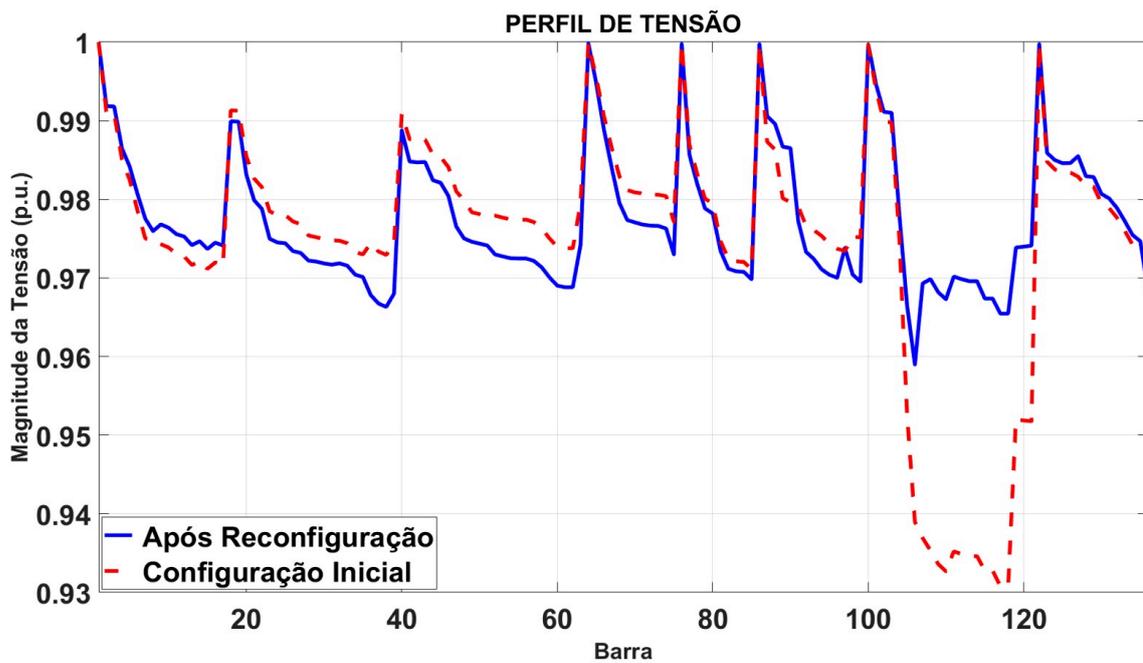
Por fim, analisando o tempo de execução dos algoritmos propostos na Tabela 15, pode-se notar a tendência de melhoria da performance do método híbrido em relação aos demais métodos, conforme o aumento das redes de teste. Novamente, tem-se o BRKGA2 com um desempenho melhor em relação ao BRKGA1 devido ao espaço de busca reduzido. Além disso, entre as estratégias de reinicialização estudadas, a estratégia baseada em elitismo teve os menores valores de tempo de execução, entre todos os casos estudados.

Figura 32 – Topologia ótima da rede 136 barras.



Fonte: Elaborado pelo autor.

Figura 33 – Perfil de tensão na rede 136 barras.



Fonte: Elaborado pelo autor.

Tabela 13 – Parâmetros de controle dos algoritmos propostos no sistema 136 barras.

Parâmetros	BRKGA/ Kruskal	BRKGA/ Co-árvore	SA	Híbrido BRKGA/SA
População	100	100	-	100
População elite	20	20	-	20
População mutante	10	10	-	10
Probabilidade (ρ_e)	0,7	0,7	-	0,7
Parâmetro de parada (ϵ_s)	-	-	2	-
Comprimento da cadeia de Markov	-	-	156	-
Semente geradora de números aleatórios	87469	87469	87469	87469

Fonte: Elaborado pelo autor.

Tabela 14 – Parâmetros de reinicialização aplicados ao BRKGA1 e BRKGA2 na rede 136 barras.

Parâmetros	BRKGA1		BRKGA2	
	Restart sem Elitismo	Restart com Elitismo	Restart sem Elitismo	Restart com Elitismo
κ	130	-	160	-
m	-	20	-	20
th_{count}	-	40	-	40

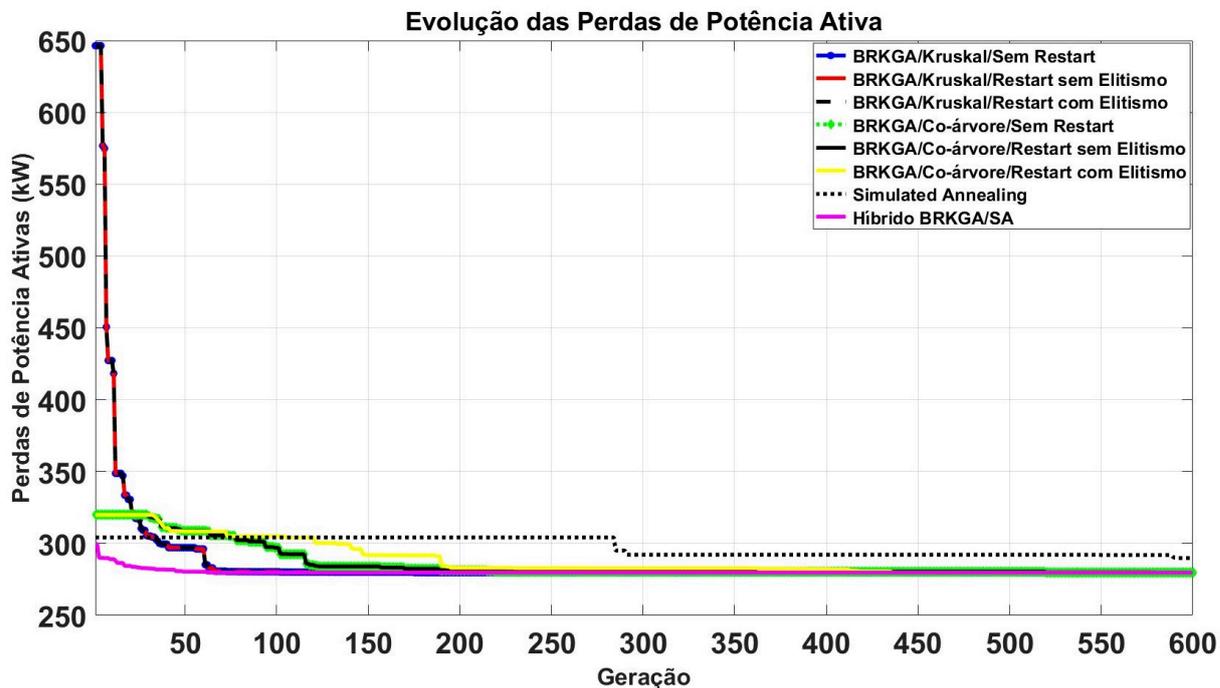
Fonte: Elaborado pelo autor.

Tabela 15 – Resultados das metodologias propostas no sistema 136 barras.

Algoritmo	Chaves abertas	Gerações/ Iterações até a solução	Tempo de Execução (seg.)	Perdas (kW)	Tensão mínima (p.u.)
BRKGA1/ Sem Restart	S7, S51, S53, S84, S90, S96, S106, S118, S126, S128, S137, S138, S139, S141, S144, S145, S147, S148, S150, S151, S156	857	496,993538	279,7736	0,960593
BRKGA1/ Restart sem Elitismo	S7, S51, S53, S84, S90, S96, S106, S118, S126, S128, S137, S138, S139, S141, S144, S145, S147, S148, S150, S151, S156	658	364,484545	279,7806	0,960591
BRKGA1/ Restart com Elitismo	S7, S51, S53, S84, S90, S96, S106, S118, S126, S128, S137, S138, S139, S141, S144, S145, S147, S148, S150, S151, S156	465	280,698642	279,7779	0,960594
BRKGA2/ Sem Restart	S7, S51, S53, S84, S90, S96, S106, S118, S126, S128, S137, S138, S139, S141, S144, S145, S147, S148, S150, S151, S156	593	361,066581	279,7828	0,960593
BRKGA2/ Restart sem Elitismo	S7, S51, S53, S84, S90, S96, S106, S118, S126, S128, S137, S138, S139, S141, S144, S145, S147, S148, S150, S151, S156	578	338,178355	279,7761	0,960594
BRKGA2/ Restart com Elitismo	S7, S51, S53, S84, S90, S96, S106, S118, S126, S128, S137, S138, S139, S141, S144, S145, S147, S148, S150, S151, S156	443	272,456008	279,7735	0,960594
Simulated Annealing	S7, S35, S51, S90, S96, S106, S118, S126, S135, S137, S138, S141, S142, S144, S145, S146, S147, S148, S150, S151, S155	17940	345,191878	279,7871	0,958969
Híbrido BRKGA2/ SA	S7, S51, S53, S84, S90, S96, S106, S118, S126, S128, S137, S138, S139, S141, S144, S145, S147, S148, S150, S151, S156	69	223,482045	279,7922	0,960596

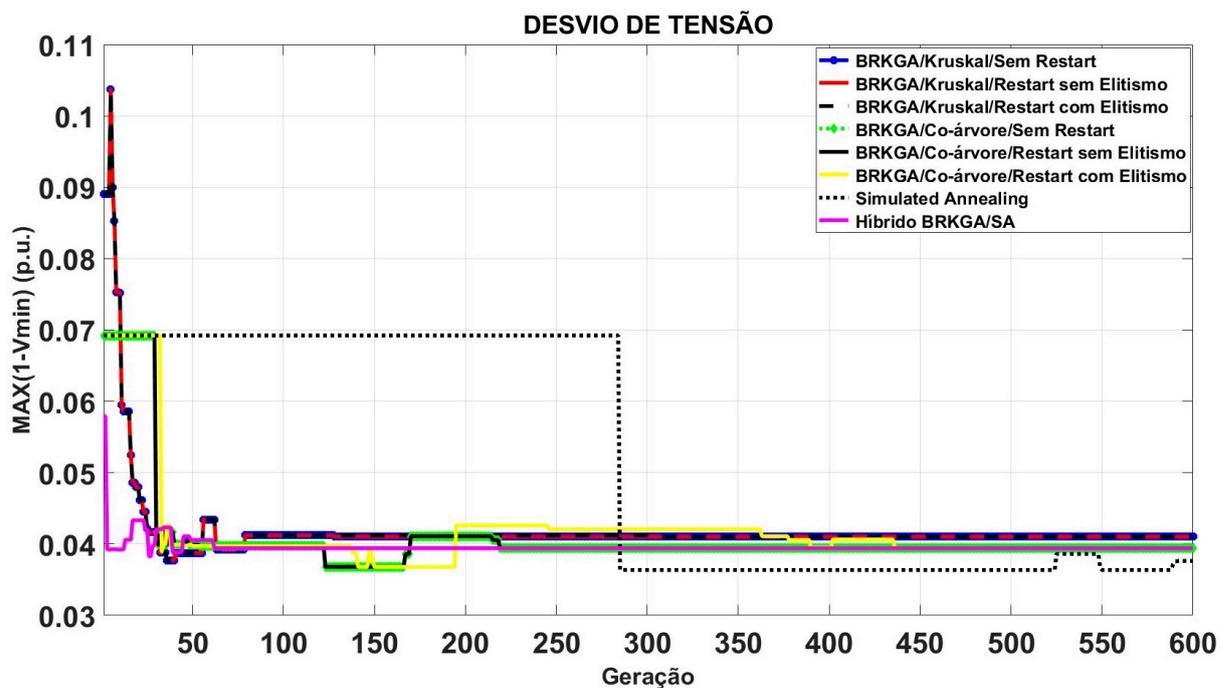
Fonte: Elaborado pelo autor.

Figura 34 – Evolução das perdas na rede 136 barras.



Fonte: Elaborado pelo autor.

Figura 35 – Evolução do desvio de tensão na rede 136 barras.



Fonte: Elaborado pelo autor.

5 CONCLUSÃO

Neste trabalho foram apresentadas cinco diferentes abordagens para resolver o problema da reconfiguração em redes de distribuição para minimização de perdas de energia, a primeira abordagem utiliza um algoritmo genético de chaves aleatórias viciadas com decodificador baseado no algoritmo de Kruskal (BRKGA1), a segunda um algoritmo genético de chaves aleatórias viciadas com decodificador baseado nos *links* da co-árvore (BRKGA2), a terceira um recozimento simulado, a quarta um algoritmo híbrido construído a partir do BRKGA2 e do recozimento simulado, e por fim, estratégias de reinicialização aplicadas ao BRKGA1 e BRKGA2. Todos os algoritmos apresentados encontraram a melhor solução disponível e diferindo um do outro apenas na quantidade de iterações e no tempo computacional para encontrar a solução ótima. Além disso, todas as soluções encontradas foram idênticas, em alguns casos superior, as literaturas consultadas.

Nas simulações com as estratégias de reinicialização aplicadas as técnicas BRKGA1 e BRKGA2 a técnica de reinicialização com elitismo foi a que obteve o melhor desempenho para a minimização do tempo de execução dos algoritmos em todos os casos estudados. Quanto ao algoritmo híbrido, este obteve uma baixa quantidade de iterações para encontrar a solução ótima em todos os casos, porém para sistemas de pequeno porte, 14, 33 e 69 barras o algoritmo teve um maior tempo de execução em relação aos demais algoritmos, porém para as redes de porte médio 84 e 136 barras o algoritmo híbrido teve um desempenho superior em relação as demais técnicas. Quanto ao recozimento simulado, este realizou uma quantidade maior de iterações para encontrar a solução ótima, no entanto, suas operações são mais simples e, desta forma, seu tempo de execução se mostrou mais baixo em alguns casos.

Durante as simulações, foi observado que os resultados dependem fortemente das condições iniciais do problema, principalmente da semente geradora de números aleatórios, pois uma mesma semente pode conduzir um algoritmo a obter um pior desempenho em relação a outro, mesmo com o primeiro tendo um espaço de busca reduzido, como aconteceu entre os casos do BRKGA1 e BRKGA2.

Desta forma, a pesquisa conduziu uma comparação entre métodos de busca global, busca local, estratégias de *restart* e métodos híbridos, conforme os resultados encontrados a pesquisa conduziu a um melhor desempenho atribuído ao método híbrido para os sistemas de distribuição de médio a grande porte e quanto as estratégias de reinicialização o método com elitismo foi o que obteve o melhor desempenho.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

As propostas a seguir são sugestões de trabalhos futuros que visam o aperfeiçoamento das metodologias apresentadas neste trabalho, bem como propostas

complementares de continuidade de estudos para a abordagem do problema da reconfiguração de redes de distribuição:

- Incluir outras formas de *cooling schedules* aos algoritmos de recozimento simulado e híbrido, incluindo principalmente o uso de *cooling schedules* adaptativos.
- Estender a utilização dos métodos propostos a sistemas de maior porte para verificar se o desempenho dos algoritmos propostos permanece o mesmo.
- Analisar o desempenho de outros decodificadores baseados em teoria de grafos para construção da topologia radial.
- Desenvolver uma abordagem de processamento paralelo entre as partes dependente e independente do BRKGA e para a metodologia híbrida, uma vez que o BRKGA e o SA podem ser processados de maneira independente.
- Adicionar outras heurísticas para a construção da geração inicial.
- Utilizar métodos de penalização mais eficientes para eliminação de soluções não factíveis.
- Adicionar uma abordagem multiobjetivo com redução do número de chaveamentos, ajustes de proteção (self-healing), dimensionamento de geradores distribuídos, reguladores de tensão e bancos de capacitores (controle Volt/Var).

REFERÊNCIAS

- AARTS, Emile; KORST, Jan. **Simulated annealing and boltzmann machines: A stochastic approach to combinatorial optimization and neural computing**. Chichester: John Wiley & Sons, 1991. ISBN 0471921467.
- ABBASKHANI-DAVANLOO, Arash; AMINI, Mahraz; MODARRESI, M. Sadegh; JAFARISHIADEH, Fatemeh. Distribution System Reconfiguration for Loss Reduction Incorporating Load and Renewable Generation Uncertainties. *In: 2019 IEEE Texas Power and Energy Conference (TPEC)*. [S.l.: s.n.], 2019. P. 1–6.
- AMAN, M. M.; JASMON, G. B.; BAKAR, A. H. A.; NAIDU, K.; MOKHLIS, H. Discrete artificial bee colony algorithm for solving network reconfiguration problem. *In: 3RD IET International Conference on Clean Energy and Technology (CEAT) 2014*. [S.l.: s.n.], 2014. P. 1–6.
- ANDRADE, Carlos E.; TOSO, Rodrigo F.; GONÇALVES, José F.; RESENDE, Mauricio G.C. The Multi-Parent Biased Random-Key Genetic Algorithm with Implicit Path-Relinking and its real-world applications. **European Journal of Operational Research**, v. 289, n. 1, p. 17–30, 2021. ISSN 0377-2217.
- ANEEL, AGENCIA NACIONAL DE ENERGIA ELÉTRICA -. **Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional – PRODIST: Módulo 7 - Cálculo de Perdas na Distribuição**. [S.l.], 2019. Disponível em: https://www.gov.br/aneeldocuments/656827/14866914/M%5C%c3%5C%b3duLo7_Revisao_5+-+Retificado/669bf2b6-7fb4-07e8-f5fd-0bea4d83ad34. Acesso em: 4 jun. 2020.
- BALAKRISHNAN, V. K. **Schaum's Outline of Graph Theory: Including Hundreds of Solved Problems**. 1. ed. New York: McGraw-Hill, mar. 1997. ISBN 0070054894.
- BARAN, M.E.; WU, F.F. Network reconfiguration in distribution systems for loss reduction and load balancing. **IEEE Transactions on Power Delivery**, v. 4, n. 2, p. 1401–1407, 1989.
- BEAN, James C. Genetic Algorithms and Random Keys for Sequencing and Optimization. **INFORMS J. Comput.**, v. 6, n. 2, p. 154–160, 1994.

- BORGES, Marlon C. O.; FRANCO, John F.; RIDER, Marcos J. Optimal Reconfiguration of Electrical Distribution Systems. **Journal of Control, Automation and Electrical Systems**, v. 1, n. 25, p. 103–111, 2013.
- CAPELI, Luiz Henrique; QUEIROZ, Leonardo Mendonça; SOUSA, Renato Eduardo Farias de. **Aprimoramento da metodologia de cálculo de perdas na distribuição regulamentada no Módulo 7 – Cálculo de Perdas na Distribuição do PRODIST: Nota Técnica n° 0057/2014-SRD/ANEEL**. [S.l.], 2014. Disponível em: <https://fdocumentos.tips/document/nota-tecnica-n-00572014-srdaneel-processos-48500-em-11-de-junho-de.html?page=1>. Acesso em: 4 jun. 2020.
- CARRAZZA, Luís C.; SOUZA, Marcelo H.; CARVALHO, Claudio E.; LIMA, Davi A. **Relatório de Perdas de Energia Elétrica na Distribuição: AGENCIA NACIONAL DE ENERGIA ELÉTRICA (ANEEL): Superintendência de Gestão Tarifária**. [S.l.], 2019. Disponível em: https://www.aneel.gov.br/documents/654800/18766993/Relat%5C%C3%5C%B3rio+Perdas+de+Energia_+Edi%5C%C3%5C%A7%5C%C3%5C%A3o+1-2019-02-07.pdf/d7cc619e-0f85-2556-17ff-f84ad74f1c8d. Acesso em: 4 jun. 2020.
- CAVALHEIRO, Ellen M.B.; VERGÍLIO, André H.B.; LYRA, Christiano. Optimal configuration of power distribution networks with variable renewable energy resources. **Computers & Operations Research**, v. 96, p. 272–280, 2018. ISSN 0305-0548.
- CHANG, Hong-Chan; KUO, Cheng-Chien. Network reconfiguration in distribution systems using simulated annealing. **Electric Power Systems Research**, v. 29, n. 3, p. 227–238, 1994. ISSN 0378-7796.
- CHIANG, H. D.; JEAN-JUMEAU, R. Optimal network reconfigurations in distribution systems. I. A new formulation and a solution methodology. **IEEE Transactions on Power Delivery**, v. 5, n. 4, p. 1902–1909, 1990.
- CHIANG, H. D.; JEAN-JUMEAU, R. Optimal network reconfigurations in distribution systems. II. Solution algorithms and numerical results. **IEEE Transactions on Power Delivery**, v. 5, n. 3, p. 1568–1574, 1990.
- CHIOU, Ji-Pyng; CHANG, Chung-Fu; SU, Ching-Tzong. Variable scaling hybrid differential evolution for solving network reconfiguration of distribution systems. **IEEE Transactions on Power Systems**, v. 20, n. 2, p. 668–674, 2005.

- CIVANLAR, S.; GRAINGER, J.J.; YIN, H.; LEE, S.S.H. Distribution feeder reconfiguration for loss reduction. **IEEE Transactions on Power Delivery**, v. 3, n. 3, p. 1217–1223, 1988.
- DAS, Amit Kumar; PRATIHAR, Dilip Kumar. Performance improvement of a genetic algorithm using a novel restart strategy with elitism principle. eng. **International journal of hybrid intelligent systems**, v. 15, n. 1, p. 1–15, 2019. ISSN 1448-5869.
- FARIA, H.; RESENDE, M. G.; ERNST, D. A biased random key genetic algorithm applied to the electric distribution network reconfiguration problem. **Journal of Heuristics**, v. 23, n. 6, p. 533–550, dez. 2017.
- GOLDBARG, Marco C.; GOLDBARG, Elizabeth. **Grafos: Conceitos, algoritmos e aplicações**. 1. ed. Rio de Janeiro: Elsevier, jul. 2012. ISBN 8535257160.
- GONÇALVES, José Fernando; RESENDE, Mauricio G. C. Biased random-key genetic algorithms for combinatorial optimization. **Journal of Heuristics**, v. 17, n. 5, p. 487–525, 2011.
- GU, Cailian; JI, Jianwei; LIU, Li. Research of immune algorithms for reconfiguration of distribution network with distributed generations. *In*: THE 26th Chinese Control and Decision Conference (2014 CCDC). [S.l.: s.n.], 2014. P. 2156–2160.
- HART, William E.; KRASNOGOR, N.; SMITH, J. E. Memetic Evolutionary Algorithms. *In*: **Recent Advances in Memetic Algorithms**. Edição: William E. Hart, N. Krasnogor e J. E. Smith. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. P. 3–27. ISBN 9783540323631.
- HART, William Eugene. **Adaptive Global Optimization with Local Search**. 1994. Tese (Doutorado) – USA. UMI Order No. GAX94-32928.
- HASANPOUR, Raof; KALESAR, Belal Mohamadi; NOSHAHR, Javad Behkesh; FARHADI, Payam. Reconfiguration of smart distribution network considering variation of load and local renewable generation. *In*: 2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe). [S.l.: s.n.], 2017. P. 1–5.
- HAUPT, Randy L.; HAUPT, Sue Ellen. **Practical Genetic Algorithms**. 2. ed. New Jersey: Wiley-Interscience, jan. 2004.

HUANG, M. D.; ROMEO, F.; SANGIOVANNI-VINCENTELLI, A. L. An efficient general cooling schedule for simulated annealing. *In: PROCEEDINGS of the IEEE International Conference on Computer-Aided Design*. Santa Clara: [s.n.], 1986. P. 381–384.

HUANG, Zheng; HARA, Ryoichi; KITA, Hiroyuki. Intelligent Flow Algorithm for Online Reconfiguration of Distribution System. **Journal of International Council on Electrical Engineering**, Taylor & Francis, v. 4, n. 1, p. 16–22, 2014. eprint: <https://doi.org/10.5370/JICEE.2014.4.1.016>.

JEON, Young-Jae; KIM, Jae-Chul; KIM, Jin-O; SHIN, Joong-Rin; LEE, K.Y. An efficient simulated annealing algorithm for network reconfiguration in large-scale distribution systems. **IEEE Transactions on Power Delivery**, v. 17, n. 4, p. 1070–1078, 2002.

KHODR, H. M.; MARTINEZ-CRESPO, J.; MATOS, M. A.; PEREIRA, J. Distribution Systems Reconfiguration Based on OPF Using Benders Decomposition. **IEEE Transactions on Power Delivery**, v. 24, n. 4, p. 2166–2176, 2009.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. **Science**, American Association for the Advancement of Science, v. 220, n. 4598, p. 671–680, 1983. ISSN 00368075.

LAARHOVEN, Peter J. M. van; AARTS, Emile. **Simulated Annealing: Theory and Applications**. 1. ed. Eindhoven: Springer: Series Mathematics e its Applications, 1987.

LEE, R. E.; BROOKS, C. L. A method and its application to evaluate automated distribution control. **IEEE Transactions on Power Delivery**, v. 3, n. 3, p. 1232–1240, 1988.

LÓPEZ, Rodrigo; LOPEZ, Miguel; MENDOZA, Jorge; LÓPEZ, Enrique; LEFRANC, Gastón. Probabilistic Minimal Loss Reconfiguration for Electric Power Distribution Control. *In: 2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*. [S.l.: s.n.], 2018. P. 1–7.

LUBY, Michael; SINCLAIR, Alistair; ZUCKERMAN, David. Optimal speedup of Las Vegas algorithms. **Information Processing Letters**, v. 47, n. 4, p. 173–180, 1993. ISSN 0020-0190.

MANTOVANI, José R. S.; CASARI, Fernando; ROMERO, Rubén A. Reconfiguração de sistemas de distribuição radiais utilizando o critério de queda de tensão.

Sociedade Brasileira de Automática, v. 11, n. 2, p. 150–159, 2000.

MARTÍ, Rafael; PARDALOS, Panos M.; RESENDE, Mauricio G. C. (Ed.). **Handbook of Heuristics**. [S.l.]: Springer, 2018. ISBN 978-3-319-07123-7.

MATSUMOTO, Makoto; NISHIMURA, Takuji. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. **ACM Trans. Model. Comput. Simul.**, Association for Computing Machinery, New York, NY, USA, v. 8, n. 1, p. 3–30, jan. 1998. ISSN 1049-3301.

MERLIN, A.; BACK, H. Search for a minimal-loss operating spanning tree configuration in an urban power distribution system. **Proceeding's 5th Power System Computation Conference**, v. 5, n. 1, p. 1–18, 1975.

METROPOLIS, Nicholas; ROSENBLUTH, Arianna W.; ROSENBLUTH, Marshall N.; TELLER, Augusta H.; TELLER, Edward. Equation of state calculations by fast computing machines. **Journal of Chemical Physics**, v. 21, p. 1087–1092, 1953.

MOHAMED IMRAN, A.; KOWSALYA, M. A new power system reconfiguration scheme for power loss minimization and voltage profile enhancement using Fireworks Algorithm. **International Journal of Electrical Power & Energy Systems**, v. 62, p. 312–322, 2014. ISSN 0142-0615.

NERI, Ferrante; COTTA, Carlos; MOSCATO, Pablo (Ed.). **Handbook of Memetic Algorithms**. [S.l.]: Springer, 2012. v. 379. (Studies in Computational Intelligence). ISBN 978-3-642-23246-6.

PONSICH, A.; AZZARO-PANTEL, C.; DOMENECH, S.; PIBOULEAU, L. Constraint handling strategies in Genetic Algorithms application to optimal batch plant design. **Chemical Engineering and Processing: Process Intensification**, v. 47, n. 3, p. 420–434, 2008. 10th French Congress on Chemical Engineering. ISSN 0255-2701.

QUEIROZ, Leonardo; LYRA, Christiano. Adaptive Hybrid Genetic Algorithm for Technical Loss Reduction in Distribution Networks Under Variable Demands. **IEEE Transactions on Power Systems**, v. 24, n. 1, p. 445–453, 2009.

RESENDE, Maurício; RIBEIRO, Celso. Restart strategies for GRASP with path-relinking heuristics. eng. **Optimization Letters**, v. 5, p. 467–478, 2011.

ROTHLAUF, Franz. **Representations for Genetic and Evolutionary Algorithms**. 2. ed. Berlin: Springer, Berlin, Heidelberg, 2006. ISBN 9783540324447.

RUAN, Bo; CHEN, Xi; HUANG, Jun; MEI, Zhen; LI, Yu. Network reconfiguration for loss reduction in distribution network with distributed generation. *In*: 2016 IEEE International Conference on Power and Renewable Energy (ICPRE). [S.l.: s.n.], 2016. P. 446–450.

SAIT, Sadiq M.; YOUSSEF, Habib. **Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems**. New Jersey: Wiley-IEEE Computer Society, jan. 2000. v. 27.

SAVIER, J. S.; DAS, Debapriya. Impact of Network Reconfiguration on Loss Allocation of Radial Distribution Systems. **IEEE Transactions on Power Delivery**, v. 22, n. 4, p. 2473–2480, 2007.

SCHMIDT, H.P.; IDA, N.; KAGAN, N.; GUARALDO, J.C. Fast reconfiguration of distribution systems considering loss minimization. **IEEE Transactions on Power Systems**, v. 20, n. 3, p. 1311–1319, 2005.

SCHNEIDER, Johannes J.; PUCHTA, Markus. Investigation of acceptance simulated annealing — A simplified approach to adaptive cooling schedules. **Physica A: Statistical Mechanics and its Applications**, v. 389, n. 24, p. 5822–5831, 2010.

SERGIENKO, I. V.; SHILO, V. P.; ROSHCHIN, V. A. Restart Technology for Solving Discrete Optimization Problems. **Cybernetics and System Analysis**, v. 36, n. 5, p. 659–666, 2000.

SHIRMOHAMMADI, D.; HONG, H.W. Reconfiguration of electric distribution networks for resistive line losses reduction. **IEEE Transactions on Power Delivery**, v. 4, n. 2, p. 1492–1498, 1989.

SILVA, Alexandre P. Alves da; FALCÃO, Djalma M.; LEE, Kwang Y. Overview of Applications in Power Systems. *In*: MODERN Heuristic Optimization Techniques. [S.l.]: John Wiley & Sons, Ltd, 2008. cap. 12, p. 235–259. ISBN 9780470225868. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470225868.ch12>.

SILVEIRA, Christoffer L. Bezão. **Análise comparativa de métodos de otimização para a solução do problema de reconfiguração de sistemas de distribuição de energia elétrica**. 2020. Diss. (Mestrado) – Ilha Solteira.

SILVEIRA, Christoffer L. Bezão; TABARES, Alejandra; FARIA, Lucas Teles; FRANCO, John F. Mathematical optimization versus Metaheuristic techniques: A performance comparison for reconfiguration of distribution systems. eng. **Electric power systems research**, Elsevier B.V, v. 196, p. 107272, 2021. ISSN 0378-7796.

SINGH, Birender; CHAUHAN, Sushil; REDDY, C.C. Power loss minimization in electrical power distribution networks by use of hybrid reconfiguration method. *In*: 2017 3rd International Conference on Condition Assessment Techniques in Electrical Systems (CATCON). [S.l.: s.n.], 2017. P. 349–354.

SPEARS, Villiam M.; JONG, Kenneth A. De. On the virtues of parameterized uniform crossover. *In*: IN Proceedings of the Fourth International Conference on Genetic Algorithms. [S.l.: s.n.], 1991. P. 230–236.

STAGG, Glenn W.; EL-ABIAD, Ahmed H. **Computer Methods in Power System Analysis**. 1. ed. New York: McGraw-Hill, abr. 1968.

STOJANOVIĆ, Branko; RAJIĆ, Tomislav. Novel approach to reconfiguration power loss reduction problem by simulated annealing technique. **International Transactions on Electrical Energy Systems**, v. 27, n. 12, e2464, 2017. e2464 ITEES-17-0236.R1. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/etep.2464>.

SURRY, Patrick D.; RADCLIFFE, Nicholas J. Inoculation to initialise evolutionary search. *In*: FOGARTY, Terence C. (Ed.). **Evolutionary Computing**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. P. 269–285.

SWARNKAR, Anil; GUPTA, Nikhil; NIAZI, K.r. A novel codification for meta-heuristic techniques used in distribution network reconfiguration. **Electric Power Systems Research**, v. 81, n. 7, p. 1619–1626, 2011.

TAYLOR, Joshua A.; HOVER, Franz S. Convex Models of Distribution System Reconfiguration. **IEEE Transactions on Power Systems**, v. 27, n. 3, p. 1407–1413, 2012.

- TESHOME, Dawit F.; LIAN, Kuo Lung. Comprehensive mixed-integer linear programming model for distribution system reconfiguration considering DGs. **IET Generation, Transmission & Distribution**, v. 12, n. 20, p. 4515–4523, 2018. ISSN 1751-8687.
- TRIKI, E.; COLLETTE, Y.; SIARRY, P. A theoretical study on the behavior of simulated annealing leading to a new cooling schedule. **European Journal of Operational Research**, v. 166, n. 1, p. 77–92, out. 2005.
- VARGAS, Rommel; ROMERO, Rubén; FRANCO, John F. Biased Random-Key Genetic Algorithm Applied to the Optimal Reconfiguration of Radial Distribution Systems. *In: 2018 IEEE PES Transmission Distribution Conference and Exhibition - Latin America (T D-LA)*. [S.l.: s.n.], 2018. P. 1–5.
- WOLPERT, David H.; MACREADY, William G. No free lunch theorems for optimization. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 1, p. 67–82, 1997.
- YOUSSEF, Habib; M. SAIT, Sadiq; ADICHE, Hakim. Evolutionary algorithms, simulated annealing and tabu search: a comparative study. **Engineering Applications of Artificial Intelligence**, v. 14, n. 2, p. 167–181, 2001. ISSN 0952-1976.
- ZHONG-HUI, Gao; JIAN-YONG, Ding; HENG, Liu. Hybrid algorithm based on artificial fish swarm algorithm and tabu search in distribution network reconfiguration. *In: 2010 International Conference On Computer Design and Applications*. [S.l.: s.n.], 2010. v. 5, p. v5-415-v5-418.
- ZHU, Jizhong. Optimal Reconfiguration of Electrical Distribution Network. *In: OPTIMIZATION OF POWER SYSTEM OPERATION*. [S.l.]: John Wiley & Sons, Ltd, 2015. cap. 12, p. 483–528. ISBN 9781118887004. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118887004.ch12>.

ANEXO A – DADOS DOS SISTEMAS DE DISTRIBUIÇÃO

Este anexo apresenta os dados dos sistemas de distribuição utilizados para a verificação da funcionalidade dos algoritmos propostos nesta pesquisa.

A.1 SISTEMA DE 14 BARRAS

- Barra da Subestação: 1
- Tensão de Base: 23 kV
- Potência de Base: 100 MVA
- Demanda Potência Ativa: 28,7 MW
- Demanda Potência Reativa: 17,3 MVar

Tabela 16 – Dados do sistema de 14 barras.

Nº do ramo	Barra Inicial	Barra Final	Resistência (p.u.)	Reatância (p.u.)	Demanda Ativa (MW)	Demanda Reativa (MVar)
1	1	4	0,075	0,1	2	1,6
2	4	5	0,08	0,11	3	1,5
3	4	6	0,09	0,18	2	0,8
4	6	7	0,04	0,04	1,5	1,2
5	2	8	0,11	0,11	4	2,7
6	8	9	0,08	0,11	5	3
7	8	10	0,11	0,11	1	0,9
8	9	11	0,11	0,11	0,6	0,1
9	9	12	0,08	0,11	4,5	2
10	3	13	0,11	0,11	1	0,9
11	13	14	0,09	0,12	1	0,7
12	13	15	0,08	0,11	1	0,9
13	15	16	0,04	0,04	2,1	1
14	5	11	0,04	0,04		
15	10	14	0,04	0,04		
16	7	16	0,09	0,12		

Fonte: (CIVANLAR *et al.*, 1988).

A.2 SISTEMA DE 33 BARRAS

- Barra da Subestação: 1
- Tensão de Base: 12,66 kV
- Potência de Base: 100 MVA
- Demanda Potência Ativa: 3715 kW
- Demanda Potência Reativa: 2300 kVAr

Tabela 17 – Dados do sistema de 33 barras.

Nº do ramo	Barra Inicial	Barra Final	Resistência (Ω)	Reatância (Ω)	Demanda Ativa (kW)	Demanda Reativa (kVAr)
1	1	2	0,0922	0,0470	100	60
2	2	3	0,4930	0,2511	90	40
3	3	4	0,3660	0,1864	120	80
4	4	5	0,3811	0,1941	60	30
5	5	6	0,8190	0,7070	60	20
6	6	7	0,1872	0,6188	200	100
7	7	8	0,7114	0,2351	200	100
8	8	9	1,0300	0,7400	60	20
9	9	10	1,0440	0,7400	60	20
10	10	11	0,1966	0,0650	45	30
11	11	12	0,3744	0,1238	60	35
12	12	13	1,4680	1,1550	60	35
13	13	14	0,5416	0,7129	120	80
14	14	15	0,5910	0,5260	60	10
15	15	16	0,7463	0,5450	60	20
16	16	17	1,2890	1,7210	60	20
17	17	18	0,7320	0,5740	90	40
18	2	19	0,1640	0,1565	90	40
19	19	20	1,5042	1,3554	90	40
20	20	21	0,4095	0,4784	90	40
21	21	22	0,7089	0,9373	90	40
22	3	23	0,4512	0,3083	90	50
23	23	24	0,8980	0,7091	420	200
24	24	25	0,8960	0,7011	420	200
25	6	26	0,2030	0,1034	60	25
26	26	27	0,2842	0,1447	60	25
27	27	28	1,0590	0,9337	60	20
28	28	29	0,8042	0,7006	120	70
29	29	30	0,5075	0,2585	200	600
30	30	31	0,9744	0,9630	150	70
31	31	32	0,3105	0,3619	210	100
32	32	33	0,3410	0,5302	60	40
33	8	21	2,0000	2,0000		
34	9	15	2,0000	2,0000		
35	12	22	2,0000	2,0000		
36	18	33	0,5000	0,5000		
37	25	29	0,5000	0,5000		

Fonte: (BARAN; WU, 1989).

Tabela 19 – Dados do sistema de 69 barras (continuação).

Nº do ramo	Barra Inicial	Barra Final	Resistência (Ω)	Reatância (Ω)	Demanda Ativa (kW)	Demanda Reativa (kVAr)
40	40	41	0,7283	0,8509	1,2	1
41	41	42	0,31	0,3623	0	0
42	42	43	0,041	0,0478	6	4,3
43	43	44	0,0092	0,0116	0	0
44	44	45	0,1089	0,1373	39,22	26,3
45	45	46	0,0009	0,0012	39,22	26,3
46	4	47	0,0034	0,0084	0	0
47	47	48	0,0851	0,2083	79	56,4
48	48	49	0,2898	0,7091	384,7	274,5
49	49	50	0,0822	0,2011	384,7	274,5
50	8	51	0,0928	0,0473	40,5	28,3
51	51	52	0,3319	0,1114	3,6	2,7
52	9	53	0,174	0,0886	4,35	3,5
53	53	54	0,203	0,1034	26,4	19
54	54	55	0,2842	0,1447	24	17,2
55	55	56	0,2813	0,1433	0	0
56	56	57	1,59	0,5337	0	0
57	57	58	0,7837	0,263	0	0
58	58	59	0,3042	0,1006	100	72
59	59	60	0,3861	0,1172	0	0
60	60	61	0,5075	0,2585	1244	888
61	61	62	0,0974	0,0496	32	23
62	62	63	0,145	0,0738	0	0
63	63	64	0,7105	0,3619	227	162
64	64	65	1,041	0,5302	59	42
65	11	66	0,2012	0,0611	18	13
66	66	67	0,0047	0,0014	18	13
67	12	68	0,7394	0,2444	28	20
68	68	69	0,0047	0,0016	28	20
69	11	43	0,5	0,5		
70	13	21	0,5	0,5		
71	15	46	1	0,5		
72	50	59	2	1		
73	27	65	1	2,5		

Fonte: (SAVIER; DAS, D., 2007)

Tabela 22 – Dados do sistema de 84 barras (continuação 2).

Nº do ramo	Barra Inicial	Barra Final	Resistência (Ω)	Reatância (Ω)	Demanda Ativa (kW)	Demanda Reativa (kVAr)
84	5	55	0,131	0,269	0	0
85	7	60	0,131	0,269		
86	11	43	0,131	0,269		
87	12	72	0,3406	0,6994		
88	13	76	0,4585	0,9415		
89	14	18	0,5371	1,0824		
90	16	26	0,0917	0,1883		
91	20	83	0,0786	0,1614		
92	28	32	0,0524	0,1076		
93	29	39	0,0786	0,1614		
94	34	46	0,0262	0,0538		
95	40	42	0,1965	0,4035		
96	53	64	0,0393	0,0807		

Fonte: (CHIOU; CHANG, C.-F.; SU, 2005)

Tabela 26 – Dados do sistema de 136 barras (continuação 3).

Nº do ramo	Barra Inicial	Barra Final	Resistência (Ω)	Reatância (Ω)	Demanda Ativa (kW)	Demanda Reativa (kVAr)
140	33	61	0,04502	0,10394		
141	59	145	0,14187	0,14166		
142	65	147	0,14187	0,14166		
143	73	206	0,0394	0,09094		
144	78	125	0,12944	0,29882		
145	125	219	0,01688	0,03898		
146	131	223	0,3317	0,17448		
147	139	154	0,14187	0,14166		
148	138	217	0,07692	0,17756		
149	138	153	0,07692	0,17756		
150	141	154	0,07692	0,17756		
151	141	220	0,07692	0,17756		
152	145	206	0,26482	0,26443		
153	160	56	0,49696	0,64567		
154	212	122	0,17059	0,08973		
155	215	123	0,05253	0,12126		
156	223	147	0,2932	0,29276		

Fonte: (MANTOVANI; CASARI; ROMERO, 2000)