

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
FACULDADE DE ENGENHARIA
CAMPUS DE ILHA SOLTEIRA

ROMMEL GREGORIO VARGAS PERALTA

RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO ATRAVÉS DO
ALGORITMO GENÉTICO DE CHAVES ALEATÓRIAS VICIADAS

Ilha Solteira

2018

ROMMEL GREGORIO VARGAS PERALTA

**RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO ATRAVÉS DO
ALGORITMO GENÉTICO DE CHAVES ALEATÓRIAS VICIADAS**

Dissertação apresentada à Faculdade de Engenharia do Câmpus de Ilha Solteira – UNESP como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.
Área de concentração: Automação.

Prof. Dr. John Fredy Franco Baquero
Orientador

Ilha Solteira

2018

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

V297r Vargas Peralta, Rommel Gregorio.
Reconfiguração de sistemas de distribuição através do algoritmo genético de chaves aleatórias viciadas / Rommel Gregorio Vargas Peralta. -- Ilha Solteira: [s.n.], 2018
77 f. : il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia . Área de conhecimento: Automação, 2018

Orientador: John Fredy Franco Baquero
Inclui bibliografia

1. Reconfiguração de sistemas de distribuição. 2. Algoritmo genético das chaves aleatórias viciadas. 3. Algoritmo de Prim. 4. Fluxo de carga de varredura para sistemas radiais.



UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Ilha Solteira

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Reconfiguração de sistemas de distribuição através do algoritmo genético de chaves aleatórias viciadas

AUTOR: ROMMEL GREGORIO VARGAS PERALTA

ORIENTADOR: JOHN FREDY FRANCO BAQUERO

Aprovado como parte das exigências para obtenção do Título de Mestre em ENGENHARIA ELÉTRICA, área: AUTOMAÇÃO pela Comissão Examinadora:

Prof. Dr. JOHN FREDY FRANCO BAQUERO
Departamento de Engenharia Elétrica / Universidade Estadual Paulista Júlio de Mesquita Filho, Câmpus Experimental Rosana

Prof. Dr. RUBEN AUGUSTO ROMERO LAZARO
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. LUIS ALFONSO GALLEGO PAREJA
Departamento de Engenharia Elétrica / Universidade Estadual de Londrina

Ilha Solteira, 20 de abril de 2018

DEDICO

À minha família

AGREDECIMIENTOS

Aos meus pais pelo apoio financeiro durante a realização deste trabalho.

Aos meus irmãos, a Alonso por seu apoio e amizade e a Joel por seu apoio, apesar da distância

Ao professor Ruben pela confiança, paciência e amizade oferecidas desde o início deste trabalho.

À família do professor Ruben, Sra. Lucila e César Augusto, pela solidariedade e amizade durante todo esse tempo.

Ao professor Fredy pela paciência e apoio durante o desenvolvimento deste trabalho.

Aos professores e companheiros do LaPSEE.

À CAPES pelo suporte financeiro.

“Ser más para servir mejor”.

San Ignacio de Loyola

RESUMO

Nesta dissertação é proposta a aplicação do algoritmo genético de chaves aleatórias viciadas para a solução do problema de reconfiguração de sistemas de distribuição. Esse problema de otimização consiste em encontrar a configuração radial que apresenta perdas mínimas, satisfazendo as restrições topológicas e as restrições operacionais, sendo modelado como um problema de Programação Não Linear Inteira Mista. O método proposto utiliza o algoritmo de Prim na geração de configurações radiais e emprega um algoritmo de fluxo de carga de varredura para avaliar cada proposta de solução. O algoritmo genético de chaves aleatórias viciadas foi desenvolvido na linguagem de programação FORTRAN e foi testado em quatro sistemas de distribuição da literatura especializada (14 barras, 33 barras, 84 barras e 136 barras). Os resultados obtidos da aplicação do algoritmo permitem avaliar o seu desempenho e eficiência em comparação com a melhor solução encontrada na literatura especializada.

Palavras-chave: Reconfiguração de sistemas de distribuição. Algoritmo genético das chaves aleatórias viciadas. Algoritmo de Prim. Fluxo de carga de varredura para sistemas radiais.

ABSTRACT

The application of the biased random-key genetic algorithm for the reconfiguration of distribution systems is proposed in this Dissertation. The problem of reconfiguration in distribution systems consists of finding the radial configuration that presents the minimum losses, satisfying topological and operating constraints and is commonly modeled as a mixed-integer nonlinear programming problem. The proposed method uses the Prim's algorithm to generate radial configurations that are evaluated through a backward/forward sweep power flow method. The biased random-key genetic algorithm used was developed in the programming language FORTRAN and was tested in four systems (14-bus, 33-bus, 84-bus and 136-bus). The obtained results show the performance and efficiency of the proposed method in comparison to the best solution found in the specialized literature.

Keywords: Distribution system reconfiguration. Biased random-key genetic algorithm. Prim's algorithm. Backward/forward sweep power flow.

LISTA DE FIGURAS

Figura 1	– Codificação utilizada em Nara.	25
Figura 2	– Codificação utilizada em Mendoza.	25
Figura 3	– Codificação utilizada em Carreño	26
Figura 4	– Codificação utilizada no algoritmo BRKGA.	26
Figura 5	– Exemplo de um grafo com 9 vértices e 8 arestas.	27
Figura 6	– Representação de um gráfo não direcionado.	27
Figura 8	– Representação de um grafo direcionado.	28
Figura 7	– Floresta de 6 nós formado por 2 árvores.	28
Figura 9	– Árvore de 9 nós com 5 folhas.	29
Figura 10	– Execução do algoritmo Kruskal para um grafo com 7 vértices e 12 arestas.	30
Figura 11	– Execução do algoritmo Kruskal para um grafo com 7 vértices e 12 arestas.	31
Figura 12	– Execução do algoritmo do Prim para um grafo com 7 vértices e 12 arestas.	32
Figura 13	– Execução do algoritmo do Prim para um grafo com 7 vértices e 12 arestas.	33
Figura 14	– Sistema de duas barras.	35
Figura 15	– Vetor de chaves com valores entre 0 e 1.	43
Figura 16	– Processo de recombinação de dos cromossomos.	45
Figura 17	– Mutação no BRKGA	46
Figura 18	– Fluxograma do BRKGA.....	47
Figura 19	– Função realizada pelo decodificador.	48
Figura 20	– Sistema de 14 barras com pesos atribuídos a cada um dos seus ramos.	52
Figura 21	– Vetor de chaves aleatórias para o sistema de 14 barras.	52
Figura 22	– Vetor binário que mostra os ramos ligados e desligados.	53
Figura 23	– Configuração radial obtida a partir do vetor da Figura 20.	54
Figura 24	– Processo de recombinação no sistema de 14 barras.	55
Figura 25	– Processo de mutação no sistema de 14 barras.	56
Figura 26	– Configuração ótima do sistema teste de 14 barras.	58
Figura 27	– Configuração ótima do sistema teste de 33 barras.	59
Figura 28	– Níveis de tensão antes e depois da reconfiguração no sistema de 33 barras.	60
Figura 29	– Evolução do BRKGA para o sistema de 33 barras.	60
Figura 30	– Configuração ótima do sistema teste de 84 barras.	61
Figura 31	– Evolução do BRKGA para o sistema de 84 barras.	62
Figura 32	– Níveis de tensão antes e depois da reconfiguração no sistema de 84 barras.	62

Figura 33 – Níveis de tensão antes e depois da reconfiguração no sistema de 136 barras.....63

Figura 34 – Evolução do BRKGA para o sistema de 136 barras.64

LISTA DE TABELAS

Tabela 1	– Parâmetros sugeridos no BRKGA.	49
Tabela 2	– Resultados para o sistema teste de 14 barras.	59
Tabela 3	– Resultados para o sistema teste de 33 barras.	60
Tabela 4	– Resultados para o sistema teste de 84 barras.	61
Tabela 5	– Resultados para o sistema teste de 136 barras.	63
Tabela 6	– Parâmetros escolhidos no algoritmo BRKGA para os sistemas teste.	65
Tabela 7	– Número de iterações feitas e número de problemas de fluxo de carga resolvidos.	65
Tabela 8	– Resultados obtidos depois da aplicação do algoritmo BRKGA.	65
Tabela 9	– Dados do sistema de 14 barras.	71
Tabela 10	– Dados do sistema de 33 barras.	71
Tabela 11	– Dados do sistema de 84 barras.	72
Tabela 12	– Dados do sistema de 136 barras.	74

LISTA DE ABREVIATURAS

AG	Algoritmo Genético
BRKGA	Biased Random Keys Genetic Algorithm
FC	Fluxo de Carga
PCIM	Programação Cônica Inteira Mista
PLIM	Programação linear Inteira Mista
PNLIM	Programação Não Linear Inteira Mista
RKGA	Random Keys Genetic Algorithm
RSDEE	Reconfiguração de Sistemas de Distribuição Elétrica
VNS	Variable Neighborhood Search

LISTA DE SIMBOLOS

Ω_i	Conjunto de ramos
g_{ij}	Condutância do ramo ij
n_{ij}	Variável binária que indica se a chave está aberta (0) ou fechada (1)
V_i	Tensão na barra i
V_j	Tensão na barra j
θ_{ij}	Diferencia angular entre as tensões das barras ij
P_{S_i}	Potencia ativa fornecida pela subestação na barra i
P_{D_i}	Demanda de potência ativa na barra i
P_{ij}	Fluxo de potência ativa pelo ramo ij
Q_{S_i}	Potencia reativa fornecida pela subestação na barra i
Q_{D_i}	Demanda de potência reativa na barra i
Q_{ij}	Fluxo de potência reativa pelo ramo ij
\underline{V}	Limite inferior de tensão
\bar{V}	Limite superior de tensão
I_{rij}	Parte real do fluxo de corrente no ramo ij
I_{xij}	Parte imaginária do fluxo de corrente no ramo ij
I_{ij}	Fluxo máximo de corrente no ramo ij
n_b	Número de barras do sistema
b_{ij}	Susceptância do ramo ij
θ_i	Ângulo da tensão da barra i
θ_j	Ângulo da tensão da barra j
n_l	Número de linhas do sistema
V_{ref}	Tensão da subestação
V_{kr}	Parte real da tensão na barra k
V_{ki}	Parte real da tensão na barra k
V_{mr}	Parte real da tensão na barra m
V_{mi}	Parte real da tensão na barra m

I_{kmr}	Parte real do fluxo de corrente no ramo km
I_{kmi}	Parte imaginaria do fluxo de corrente no ramo km
Z_{km}	Impedância do ramo km
R_{km}	Resistencia do ramo km
X_{km}	Reatância do ramo km
nr	Número de ramos do sistema
p	Tamanho da população
p_e	Tamanho da população de elite
p_m	Tamanho da população de elite
ρ_e	Probabilidade de herança do alelo de elite

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVOS E CONTRIBUIÇÕES	16
1.2	ESTRUTURA DA DISSERTAÇÃO	16
2	O PROBLEMA DA RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA	17
2.1	DESCRIÇÃO DO PROBLEMA.....	17
2.2	MODELAGEM MATEMÁTICA DO PROBLEMA	17
2.3	REVISÃO DA LITERATURA ESPECIALIZADA.....	19
2.3.1	Métodos aproximados	19
2.3.2	Métodos exatos	22
3	CARACTERÍSTICAS ESPECÍFICAS DO PROBLEMA DE RECONFIGURAÇÃO QUANDO É RESOLVIDO USANDO META-HEURÍSTICAS	24
3.1	CODIFICAÇÃO DE UMA PROPOSTA DE SOLUÇÃO	24
3.2	GERAÇÃO DE PROPOSTAS DE SOLUÇÕES RADIAIS	26
3.2.1	Conceitos básicos	26
3.2.1.1	<i>Floresta e árvore</i>	28
3.2.1.2	<i>Árvore geradora</i>	29
3.2.2	Algoritmo de Kruskal	29
3.2.3	Algoritmo de Prim	31
3.3	NECESSIDADE DE RESOLVER PROBLEMAS DE FLUXO DE CARGA RADIAL	33
3.3.1	Método de varredura	34
3.4	A RESTRIÇÃO DE RADIALIDADE.....	37
3.5	ESPAÇO DE BUSCA REDUZIDO.....	38

4	O ALGORITMO GENÉTICO ESPECIALIZADO DE CHAVES ALEATÓRIAS VICIADAS	39
4.1	O ALGORITMO GENÉTICO	39
4.1.1	Conceitos básicos	39
4.1.2	Codificação da proposta de solução	40
4.1.3	Seleção no algoritmo genético	40
4.1.4	Operadores no algoritmo genético	41
4.1.5	Parâmetros dos algoritmos genéticos	42
4.2	O ALGORITMO GENÉTICO DE CHAVES ALEATÓRIAS VICIADAS.....	43
4.2.1	Visão geral	43
4.2.2	Seleção	44
4.2.3	Operadores genéticos no BRKGA	44
4.2.4	Componentes do BRKGA	47
5	O ALGORITMO GENÉTICO DE CHAVES ALEATÓRIAS VICIADAS APLICADO AO PROBLEMA DE RECONFIGURAÇÃO	50
5.1	ESTRUTURA GERAL DO ALGORITMO	50
5.2	SELEÇÃO DE PARÂMETROS.....	51
5.3	GERAÇÃO DA POPULAÇÃO INICIAL	51
5.4	DECODIFICAÇÃO DAS PROPOSTAS DE SOLUÇÃO	51
5.5	AValiação da função objetivo	54
5.6	CLASSIFICAÇÃO DAS PROPOSTAS DE SOLUÇÃO.....	54
5.7	IMPLEMENTAÇÃO DE OPERADORES GENÉTICOS.....	55
5.7.1	Recombinação	55
5.7.2	Mutação	56
5.8	MODIFICAÇÃO DA POPULAÇÃO INICIAL	56
6	RESULTADOS	58

6.1	SISTEMA DE 14 BARRAS	58
6.2	SISTEMA DE 33 BARRAS	59
6.3	SISTEMA DE 84 BARRAS	61
6.4	SISTEMA DE 136 BARRAS	63
6.5	RESUMO DO RESULTADOS.....	65
7	CONCLUSÕES	66
	REFERÊNCIAS	67
	ANEXO A - Dados dos sistemas de distribuição	71
A.1	Sistema de 14 barras	71
A.2	Sistema de 33 barras	71
A.3	Sistema de 84 barras	72
A.4	Sistema de 136 barras	74

1 INTRODUÇÃO

Os sistemas de distribuição de energia elétrica são planejados em forma malhada, mas operam de forma radial, pois isso facilita a coordenação das proteções e também reduz as correntes de curto-circuito. O problema de Reconfiguração de Sistemas de Distribuição de Energia Elétrica (RSDEE) consiste em encontrar a configuração radial que apresenta perdas mínimas e que atende às restrições operacionais, como as leis de Kirchhoff, níveis de tensão e limites de corrente em cada um dos ramos. Nesta dissertação, propõe-se o uso do Algoritmo Genético de Chaves Aleatórias Viciadas (BRKGA) para resolver o problema de reconfiguração. O método proposto permite explorar eficientemente o espaço de busca, de forma que apenas configurações radiais são avaliadas em termos da função objetivo (perdas de potência) e da satisfação das restrições de operação (limites de tensão e de corrente).

1.1 OBJETIVOS E CONTRIBUIÇÕES

Este trabalho tem os seguintes objetivos e contribuições:

- Apresentar o algoritmo genético das chaves aleatórias viciadas e sua aplicação no problema da reconfiguração de sistemas de distribuição;
- Propor um método para gerar apenas configurações radiais, que funciona em um espaço de busca menor;
- Mostrar uma comparação dos resultados obtidos pela aplicação do algoritmo BRKGA com os resultados obtidos na literatura especializada para o problema de reconfiguração.

1.2 ESTRUTURA DA DISSERTAÇÃO

Além deste capítulo introdutório, no Capítulo 2, o problema da RSDEE é apresentado de forma mais ampla, descrevendo o modelo matemático do problema e a revisão da literatura especializada. O Capítulo 3 discute as características do problema de reconfiguração quando é resolvido usando meta-heurísticas. O Capítulo 4 apresenta o algoritmo genético das chaves aleatórias viciadas, suas características, seus componentes, assim como alguns detalhes de sua implementação computacional. No Capítulo 5 é proposta a aplicação do algoritmo BRKGA ao problema da RSDEE, enquanto que no Capítulo 6 são apresentados os resultados obtidos em sistemas teste da literatura especializada. Finalmente, o Capítulo 7 apresenta as conclusões alcançadas e no Anexo são incluídos os dados dos sistemas testados.

2 O PROBLEMA DA RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

Neste capítulo serão discutidos o problema da reconfiguração de sistemas de energia elétrica, o modelo matemático do problema, e também é feita uma revisão da literatura especializada, descrevendo as técnicas de otimização utilizadas na sua solução.

2.1 DESCRIÇÃO DO PROBLEMA

Nos sistemas de distribuição, procura-se cada vez mais ter redes que operem de forma eficiente. Para atingir esse objetivo, uma das alternativas comumente usadas é a reconfiguração de sistemas de distribuição, isto é, a mudança da topologia do sistema por meio da conexão ou desconexão de circuitos.

A reconfiguração dos sistemas de distribuição é um problema de otimização combinatória complexa de solução difícil, que procura encontrar a configuração radial em que as perdas mínimas são obtidas, e para isso, chaves seccionadoras das linhas são abertas e/ou fechadas dentro do sistema, de modo que a configuração radial de operação seja mantida. A configuração radial constitui a restrição topológica do problema, mas, além dessa condição, existem restrições adicionais entre as quais estão:

- Leis de Kirchhoff;
- Limites de tensão nas barras;
- Limites de corrente nas linhas.

O problema de reconfiguração tem sido amplamente estudado na literatura especializada. Diferentes trabalhos podem ser encontrados nos quais métodos de otimização são aplicados para resolver este problema. Esses métodos variam desde o uso de técnicas exatas até a aplicação de métodos aproximados, como heurísticas e meta-heurísticas.

2.2 MODELAGEM MATEMÁTICA DO PROBLEMA

O problema de RSDEE para redução de perdas de potência é um problema que pode ser modelado como um problema de Programação Não Linear Inteiro Misto (PNLIM). O modelo

matemático para o problema RSDEE pode ser escrito segundo o conjunto de equações (1) – (9), como proposto em (LAVORATO et al., 2012).

$$\min v = \sum_{(ij) \in \Omega_l} (g_{ij} n_{ij} (V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij})) \quad (1)$$

s. t

$$P_{S_i} - P_{D_i} + \sum_{j \in \Omega_{b_i}} (n_{ij} P_{ij}) = 0 \quad \forall i \in \Omega_b \quad (2)$$

$$Q_{S_i} - Q_{D_i} + \sum_{j \in \Omega_{b_i}} (n_{ij} Q_{ij}) = 0 \quad \forall i \in \Omega_b \quad (3)$$

$$\underline{V} \leq V_i \leq \bar{V} \quad \forall i \in \Omega_b \quad (4)$$

$$n_{ij} (I_{r_{ij}}^2 + I_{x_{ij}}^2) \leq \bar{I}_{ij}^2 \quad \forall (ij) \in \Omega_l \quad (5)$$

$$\sum_{(ij) \in \Omega_l} n_{ij} = n_b - 1 \quad (6)$$

$$n_{ij} \in \{0,1\} \quad \forall (ij) \in \Omega_l \quad (7)$$

A equação (1) representa a função objetivo do problema, que corresponde à minimização das perdas de potência ativa.

As restrições (2) e (3) representam as equações convencionais de balanço de potência; as variáveis de fluxo de potência P_{ij} e Q_{ij} são definidas por (8) e (9):

$$P_{ij} = V_i^2 g_{ij} - V_i V_j (g_{ij} \cos \theta_{ij} + b_{ij} \sin \theta_{ij}) \quad (8)$$

$$Q_{ij} = -V_i^2 b_{ij} - V_i V_j (g_{ij} \sin \theta_{ij} - b_{ij} \cos \theta_{ij}) \quad (9)$$

A restrição (4) representa os limites de tensão em cada barra do sistema, onde \underline{V} e \bar{V} correspondem aos limites superior e inferior do módulo da tensão.

A restrição (5) representa a corrente máxima em cada ramo. Os elementos $I_{r_{ij}}$ e $I_{x_{ij}}$ são dados por (10) e (11):

$$I_{r_{ij}} = g_{ij} (V_i \cos \theta_i - V_j \cos \theta_j) - b_{ij} (V_i \sin \theta_i - V_j \sin \theta_j) \quad (10)$$

$$I_{x_{ij}} = g_{ij} (V_i \sin \theta_i - V_j \sin \theta_j) + b_{ij} (V_i \cos \theta_i - V_j \cos \theta_j) \quad (11)$$

A equação (6) representa a condição necessária para garantir a radialidade do sistema. De acordo com Lavorato et al. (2012), se as demandas em todas as barras de carga fossem diferentes

de zero, as restrições (2) e (3), juntamente com (6), garantem que uma solução factível do problema seja radial. Dado que a solução deve satisfazer (2), a demanda de energia de cada barra de carga deve ser satisfeita, então há um caminho entre a subestação e cada barra. Assim, cada barra está conectada à subestação, formando um gráfico relacionado. Como a rede que deseja-se reconfigurar já está operando, a restrição (5) deve ser cumprida pela topologia atual.

A equação (7) representa a natureza binária de n_{ij} . O circuito entre as barras i - j está conectado se o valor da variável binária correspondente (n_{ij}) for igual a 1 e está desconectado se o valor for igual a 0.

2.3 REVISÃO DA LITERATURA ESPECIALIZADA

Para resolver o problema da reconfiguração de sistemas de reconfiguração, várias técnicas foram desenvolvidas ao longo dos anos, as quais podem ser classificadas em dois grandes grupos: 1) métodos exatos, e 2) métodos aproximados. Primeiro, serão discutidos os métodos aproximados e, em seguida, os métodos exatos desenvolvidos para a solução do problema de RSDEE.

2.3.1 Métodos aproximados

Os métodos aproximados incluem heurísticas e meta-heurísticas. Embora estes métodos não possam comprovar a otimalidade da solução encontrada, geralmente eles encontram soluções de boa qualidade em um tempo muito menor do que os métodos exatos. Nas próximas seções é feita a revisão da literatura especializada que inclui heurísticas e meta-heurísticas que foram aplicadas ao problema da reconfiguração.

2.3.1.1 Heurísticas

As heurísticas são técnicas que são desenvolvidas especificamente para um determinado problema, encontrando soluções de boa qualidade para problemas complexos em pequenos tempos computacionais em comparação com os métodos exatos. Elas são baseadas na experiência e no senso comum. Para o problema da reconfiguração dos sistemas de distribuição, podem ser encontradas as propostas heurísticas discutidas a continuação.

(MERLIN; BACK, 1975), foi um dos primeiros trabalhos na literatura que abordam o problema da RSDEE. Este trabalho apresenta duas metodologias. A primeira é uma heurística construtiva, na qual todas as chaves estão fechadas, transformando o sistema em uma configuração malhada e, em seguida, um problema de fluxo de carga é resolvido para conhecer os estados do sistema. Depois de ter o fluxo de potência em cada ramo, aquele com o menor fluxo torna-se um candidato a ser desligado, dessa forma continua-se até obter uma configuração radial. A segunda metodologia é um algoritmo *branch and bound* que usa as perdas da configuração malhada como um limite inferior, sendo usado como limite superior o valor das perdas obtidas com a primeira metodologia.

Na proposta de (CIVANLAR et al., 1988) são feitas trocas entre os ramos ligados e desligados, de forma que a topologia radial do sistema é mantida. A variação das perdas é calculada com uma fórmula matemática aproximada.

(SHIRMOHAMMADI; HONG, 1989), com base no trabalho de Merlin e Back (1975), inclui na sua proposta as restrições dos limites de corrente nas linhas e os limites de tensão nas barras, considerando também as perdas reativas. Se um dos limites for violado, a chave do ramo é fechada e a chave do ramo através da qual o menor valor do fluxo de corrente passa é aberto.

O trabalho de (BARAN; WU, 1989), baseado na metodologia de Civanlar (1988) de troca de ramos ligados e desligados, introduz dois métodos para o cálculo de fluxos de energia para configurações radiais após a mudança de ramos, buscando reduzir as perdas de potência.

(GOSWAMI; BASU, 1992) estenderam a metodologia proposta por Shirmohammadi e Hong (1989), de forma que em vez de fechar todos os ramos do sistema, apenas um deles é fechado, formando um único laço. Depois disso, o fluxo de energia em cada ramo é calculado para finalmente abrir o ramo pelo qual o fluxo de energia é menor.

(GOMES et al., 2006) propuseram uma metodologia em que a partir de uma configuração malhada são abertas chaves até obter uma configuração radial. Dos ramos candidatos a serem desligados, é aberto aquele ramo que leva a um menor aumento das perdas. Depois disso, segue um processo de troca de ramos ligados por desligados para tentar reduzir as perdas.

2.3.1.2 Meta-heurísticas

As meta-heurísticas são definidas como métodos de solução que realizam uma interação entre procedimentos de melhoria local e estratégias de nível superior para criar um processo

capaz de escapar da solução ótima local e realizar uma busca robusta de um espaço de solução (GLOVER; KOCHENBERGER, 2003).

Diversos trabalhos que resolvem o problema de reconfiguração utilizando meta-heurísticas podem ser encontrados na literatura especializada. Esses métodos aplicam *Tabu Search*, Algoritmos Genéticos, *Simulating Annealing*, VNS, etc. Uma pequena revisão de alguns desses trabalhos é feita a seguir.

Nara et al. (1992) propuseram pela primeira vez a solução do problema da reconfiguração de sistemas de distribuição por meio de um algoritmo genético, utilizando uma codificação binária e operadores de recombinação e mutação; entretanto não foi discutido como lidar com a aparição de configurações não radiais.

A proposta de Zhu (2002), baseada no trabalho de Nara et al. (1992), propõe o uso de um algoritmo genético "refinado", no qual o tamanho da codificação é reduzido. Além disso, propõe-se um método de fluxo de carga radial para avaliar propostas de soluções e modifica o operador de mutação de acordo com algumas regras probabilísticas.

(MENDOZA et al., 2006), para a otimização do problema de reconfiguração, emprega um algoritmo genético em que a codificação binária não é mais utilizada e a análise topológica é utilizada para a criação de propostas de soluções radiais. Introduce o conceito de laços fundamentais para evitar a aparição de laços fechados e apresenta operadores genéticos especializados.

(CARRENO; ROMERO; PADILHA-FELTRIN, 2008) propuseram uma nova forma de codificação da solução do problema de RSDEE. Para a geração da população inicial utilizaram uma adaptação do algoritmo de Prim, além de um eficiente operador de recombinação que permite a obtenção de configurações radiais. Para a otimização do problema, um algoritmo genético Chu-Beasley é usado.

(ZHANG; FU; ZHANG, 2007) desenvolveram um algoritmo Tabu Search melhorado em que uma operação de mutação é introduzida para escapar dos mínimos locais. Propõe um método para gerar uma vizinhança candidata, composta por elementos de alta qualidade, e com isso realizar uma busca local eficiente. Este último método baseia-se na equação de variação de perdas propostas no trabalho de Baran e Wu (1989).

(FRANCO et al., 2012) empregaram uma meta-heurística Tabu Search para a solução do problema de reconfiguração. O algoritmo de Prim foi usado para encontrar uma solução inicial;

além disso, métodos especializados foram utilizados para calcular perdas e uma técnica para reduzir a vizinhança foi aplicada visando acelerar a convergência e reduzir o tempo computacional.

A proposta de Possagnolo (2015) resolve o problema da reconfiguração de sistemas de distribuição usando uma meta-heurística de busca de vizinhança variável (VNS) contemplando todas as suas variantes. Métodos especializados são usados para calcular as perdas de potência e uma técnica para reduzir a vizinhança é proposta. Além disso, o problema é tratado com cenários múltiplos de demanda.

Souza, Romero e Franco (2015) estudaram a aplicação de três algoritmos imunológicos artificiais ao problema de RSDEE. Esse trabalho utiliza a codificação proposta por Carreño et al. (2006) e usa o algoritmo de Prim para gerar a população inicial. Nesse trabalho foram considerados cenários com diferentes níveis de demanda.

2.3.2 Métodos exatos

(MERLIN; BACK, 1975) apresentaram uma metodologia em que inicialmente todas as chaves dos circuitos estão fechadas e formam uma configuração malhada; depois as chaves são abertas sequencialmente, de modo que, usando uma medida aproximada, as perdas da rede sejam minimizadas em cada etapa até que uma configuração radial seja alcançada. Para a resolução do problema foi usado o algoritmo de Branch and Bound.

(GLAMOCANIN, 1990) formulou o problema da reconfiguração como um problema de transporte com custos quadráticos. Para obter a configuração inicial as perdas são linearizadas e a solução do problema é obtida por meio do método Simplex.

(JABR; SINGH; PAL, 2012) propõem um modelo exato de programação não linear inteira mista para a resolução do problema de RSDEE considerando a geração distribuída. Em primeiro lugar, propõe-se um modelo convexo de programação cônica inteira mista (PCIM) para encontrar as perdas exatas e sua solução global é garantida. Além disso, apresenta-se um modelo de programação linear inteira mista (PLIM) que é o resultado da aproximação poliédrica das restrições cônicas. Restrições da árvore geradora são usadas para garantir a radialidade das configurações e o modelo também limita o número de ações de comutação das chaves.

Franco et al. (FRANCO et al., 2013) apresentaram um modelo de programação linear inteira mista (PLIM) para a resolução do problema de reconfiguração de sistemas de distribuição

considerando geração distribuída. Esse modelo é o resultado da linearização das restrições não-lineares de um modelo de programação não linear (PNLIM) definido nesse artigo. Esse modelo garante a convergência para a solução ótima por meio de um *solver* comercial.

3 CARACTERÍSTICAS ESPECÍFICAS DO PROBLEMA DE RECONFIGURAÇÃO QUANDO É RESOLVIDO USANDO META-HEURÍSTICAS

Como pode ser visto ao analisar a literatura especializada sobre o problema da reconfiguração de sistemas de distribuição, no momento da aplicação das diferentes meta-heurísticas, é possível apreciar certas características e considerações que os métodos meta-heurísticos compartilham ao otimizar este problema.

Uma das primeiras e principais etapas é escolher a codificação das propostas de solução que são usadas. Neste capítulo se apresentará uma breve visão das codificações usadas em diferentes trabalhos para otimizar este problema. Após essa breve explicação, será discutida a codificação usada no algoritmo genético das chaves aleatórias viciadas, que é uma codificação particular desta meta-heurística.

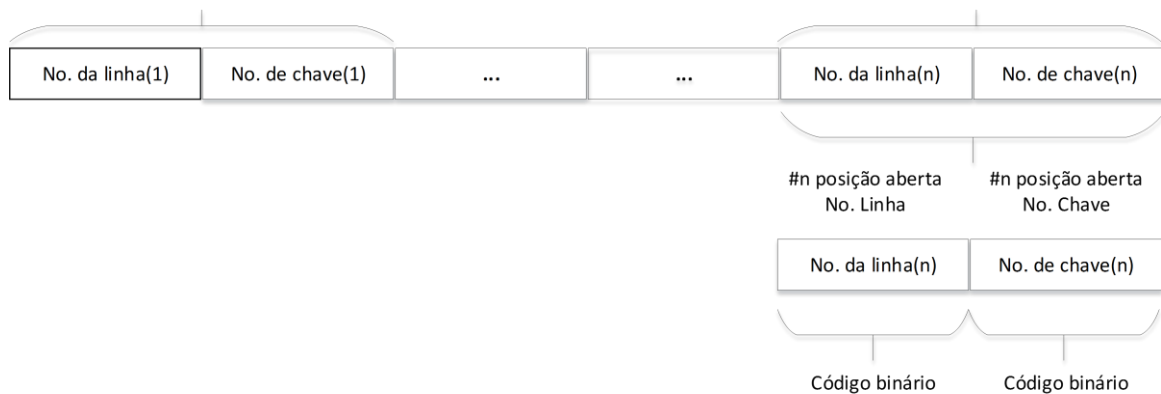
Depois, será abordada a geração de propostas de soluções radiais, de grande importância na otimização deste problema uma vez que as soluções devem apresentar uma configuração radial. Além disso, o algoritmo de fluxo de carga usado para a avaliação das propostas de soluções radiais será tratado.

3.1 CODIFICAÇÃO DE UMA PROPOSTA DE SOLUÇÃO

A codificação é uma maneira adequada de representar uma solução proposta pertencente ao espaço de busca do problema a ser otimizado, sendo o objeto desta dissertação o problema da reconfiguração de sistemas de distribuição para minimização das perdas de potência.

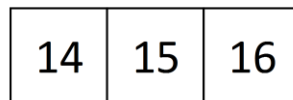
Revisando a literatura especializada sobre a otimização do problema da reconfiguração através do uso de meta-heurísticas, pode-se observar que as soluções podem ser codificadas de várias formas. Neste tópico, serão analisadas as formas de decodificação usadas em trabalhos anteriores e depois se explicará a codificação utilizada neste trabalho.

No trabalho de (NARA et al., 1992), no qual é usado um algoritmo genético, foi proposto o uso de uma codificação binária, em que foram especificados os ramos desligados, bem como o número da chave aberta pertencente a esse ramo (considerando várias chaves por ramo). Neste trabalho, não todas as propostas de solução produto da aplicação dos operadores genéticos são configurações radiais. A Figura 1 mostra a codificação utilizada nesse trabalho.

Figura 1 – Codificação utilizada em Nara (1992).

Fonte: Nara (1992).

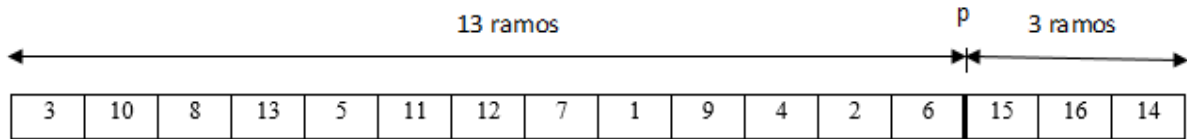
No trabalho de Mendoza (2006), que propõe um algoritmo genético, a codificação binária não é mais utilizada. Nesse trabalho, uma solução proposta é codificada usando um vetor de comprimento $(n_l - n_b + 1)$ que indica quais ramos do sistema estão desligados, onde n_l é o número total de ramos e n_b é o número de barras. O uso de operadores genéticos especializados procura que os produtos da recombinação e a mutação sejam configurações radiais. Assim, a Figura 2 mostra o exemplo da codificação de uma proposta de solução.

Figura 2 – Codificação utilizada em Mendoza (2006).

Fonte: Próprio autor.

No trabalho de Carreño, Romero e Padilha-Feltrin (CARRENO; ROMERO; PADILHA-FELTRIN, 2008), no qual é usado um algoritmo genético de Chu-Beasley, propõe-se uma nova codificação que consiste em um vetor de comprimento n_l no qual as primeiras $(n_b - 1)$ posições incluem todos os ramos ligados no sistema e as restantes $(n_l - n_b + 1)$ posições incluem os ramos desvinculados do sistema. Com isso, é possível trabalhar com apenas topologias radiais. Na Figura 3, a codificação proposta nesse trabalho é mostrada.

Tendo feito uma breve revisão das codificações utilizadas na otimização do problema da reconfiguração de sistemas de distribuição usando algoritmos genéticos, a codificação utilizada nesta dissertação é explicada.

Figura 3 – Codificação utilizada em Carreño (2010)

Fonte: Carreño (2010).

Neste trabalho é utilizada uma codificação específica que depende do Algoritmo Genético de Chaves Aleatórias Viciadas (BRKGA), que é a meta-heurística que será usada. Para codificar uma solução proposta, propõe-se um vetor de comprimento n_l , formado por números reais entre 0 e 1, onde o valor do i -ésimo elemento do vetor corresponde a um peso atribuído ao i -ésimo ramo do sistema. A Figura 4 mostra um exemplo dessa codificação. Esse vetor será decodificado e retornará uma configuração radial. Este processo de decodificação será feito com o algoritmo de Prim, que será explicado na próxima seção.

Figura 4 – Codificação utilizada no algoritmo BRKGA.

0.64	0.85	0.36	0.95	0.46	0.82	0.66	0.28	0.52	0.37	0.78	0.45	0.65	0.23	0.16	0.31
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

Fonte: Próprio autor.

3.2 GERAÇÃO DE PROPOSTAS DE SOLUÇÕES RADIAIS

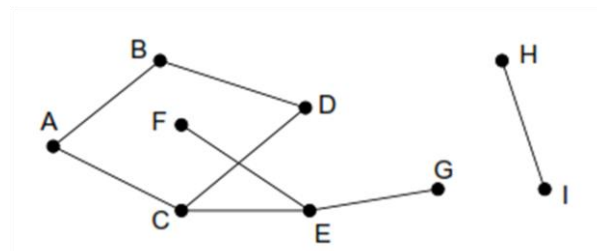
Para a geração de apenas propostas de soluções radiais pode ser usado um algoritmo de árvore geradora. Nesta seção serão explicados alguns conceitos sobre a teoria dos grafos e depois se explicará o algoritmo que foi usado na geração das propostas de soluções radiais.

3.2.1 Conceitos básicos

Informalmente, um grafo é um conjunto de pontos conectados por linhas, mas do ponto de vista matemático um grafo é formalmente um par de conjuntos (V, E) (LEHMAN, ERIC, F THOMSON, 2017), onde:

- V é um conjunto não vazio cujos elementos são chamados vértices;
- E é uma coleção de subconjuntos de dois elementos chamados arestas.

Figura 5 – Exemplo de um grafo com 9 vértices e 8 arestas.



Fonte: Lehman (2017).

Da figura 5, obtemos os conjuntos:

$$V = \{A, B, C, D, E, F, G, H, I\}$$

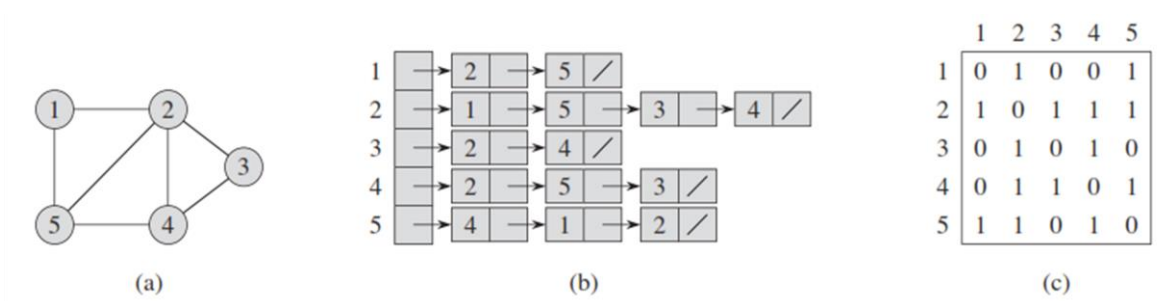
$$E = \{\{A, B\}, \{A, C\}, \{B, D\}, \{C, D\}, \{C, E\}, \{E, F\}, \{E, G\}, \{H, I\}\}$$

Se forem eliminados alguns vértices ou arestas de um grafo, obtém-se um subgrafo. Formalmente, um subgrafo de $G = (V, E)$ é um grafo $G' = (V', E')$ onde V' é um subconjunto não vazio de V e E' é um subconjunto de E .

Podem-se dividir os grafos em duas grandes classes: os grafos direcionados (digrafos) ou os grafos não direcionados (grafos simples).

Para representar um grafo existem duas formas padrão. A primeira é como uma lista de adjacência e a outra como uma matriz de adjacência, e ambos podem ser aplicados aos grafos direcionados ou não direcionados (grafos simples), como ilustrado nas Figuras 6 e 7.

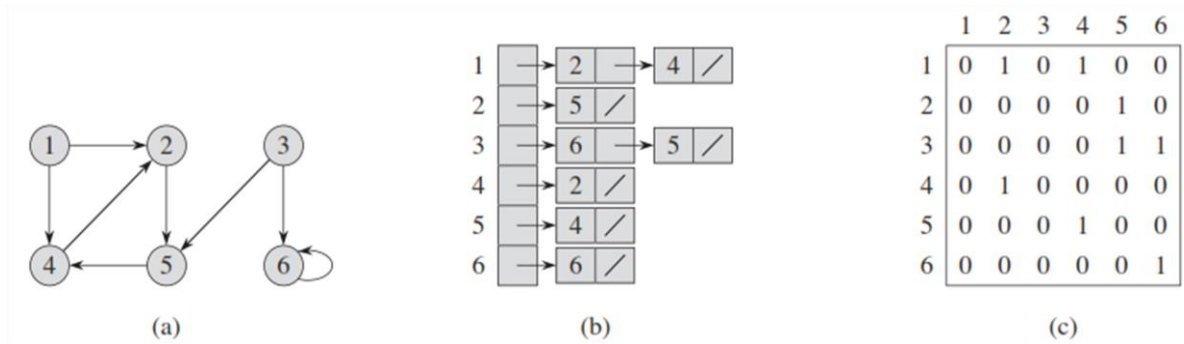
Figura 6 – Representação de um grafo não direcionado. (a) Grafo não direcionado G. (b) Representação por lista de adjacências. (c) Representação por matriz de adjacência



Fonte: Cormen (2013).

Em um grafo, dois vértices estão conectados quando há um caminho que começa em um deles e termina no outro. Por convenção, cada vértice está conectado a si mesmo por um caminho de comprimento zero. Considerando isso, um grafo é conexo se cada par de vértices estiverem conectados.

Figura 7 – Representação de um grafo direcionado. (a) Grafo direcionado G. (b) Representação por lista de adjacências. (c) Representação por matriz de adjacência.



Fonte: Cormen (2013).

Figura 8 – Floresta de 6 nós formado por 2 árvores.



Fonte: Lehman (2017).

3.2.1.1 Floresta e árvore

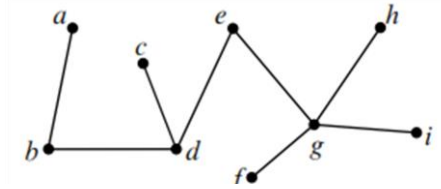
Dentro da teoria dos grafos, um grafo acíclico é chamado de floresta, e um grafo acíclico conexo é chamado de uma árvore.

Em uma floresta, um nó de grau 1 é chamado de folha. Por exemplo, no grafo da Figura 8, a floresta tem 4 folhas, enquanto que o grafo da Figura 9 tem 5 folhas.

Uma árvore tem várias propriedades únicas, entre as quais: (1) é um grafo não direcionado; (2) existe apenas um caminho entre um par de vértices; (3) adicionar uma aresta

entre dois nós não adjacentes cria um grafo com um laço; (4) remover uma aresta desconecta o grafo; (5) o número de arestas é igual ao número de vértices menos 1.

Figura 9 – Árvore de 9 nós com 5 folhas.



Fonte: Cormen (2013).

3.2.1.2 Árvore geradora

Todos os grafos conexos contêm um subgrafo que é uma árvore com os mesmos vértices do grafo, isso é chamado de árvore geradora para o grafo. Essas árvores geradoras são importantes para os grafos com peso, nos quais valores numéricos são atribuídos a todas suas arestas. O peso total do grafo é a soma dos pesos de todas as suas arestas. O problema de encontrar o peso total mínimo para a árvore geradora é chamado de problema mínimo da árvore geradora (LEHMAN; ERIC; F THOMSON, 2017).

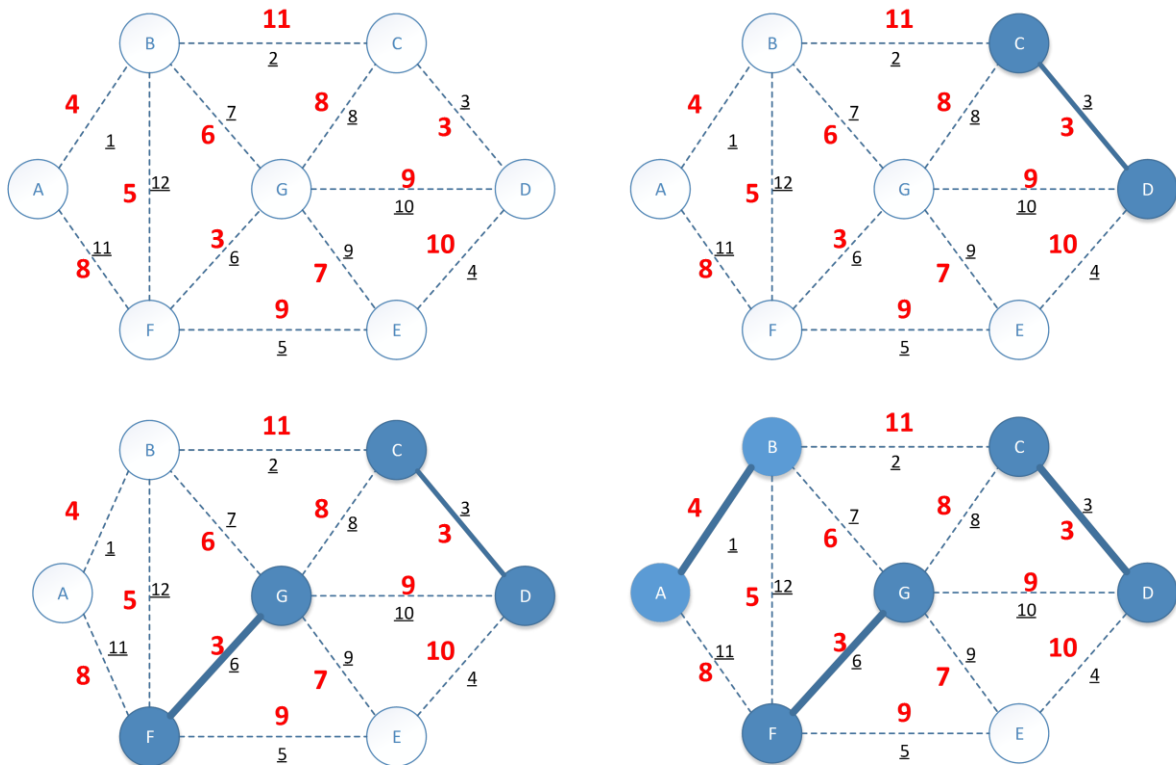
Para resolver esse problema, serão explicados brevemente dois algoritmos: o algoritmo de Kruskal e o algoritmo de Prim. Ambos os algoritmos são gulosos e compartilham o mesmo método genérico, que consiste em adicionar na árvore geradora mínima uma aresta cada vez (iteração). Esses métodos manipulam um conjunto de arestas A , mantendo o seguinte loop invariante: “antes de cada iteração, A é um subconjunto de alguma árvore geradora mínima”. Desta forma, determina-se uma aresta (u, v) que pode ser adicionada ao conjunto A sem violar esse loop invariante e essa aresta chama-se uma aresta segura.

3.2.2 Algoritmo de Kruskal

O algoritmo de Kruskal tem como ponto de partida a aresta do menor peso e, a partir desse ponto, adiciona-se uma aresta segura à floresta depois de procurar entre todas as arestas que conectam qualquer par de árvores na floresta. A aresta segura selecionada é aquela de menor peso. As Figuras 10 e 11 ilustram a execução do algoritmo de Kruskal para um grafo com 7

vértices e 12 arestas com pesos, o número da aresta é mostrado em cor preta e o peso da aresta em vermelho.

Figura 10 – Execução do algoritmo Kruskal para um grafo com 7 vértices e 12 arestas (parte 1).

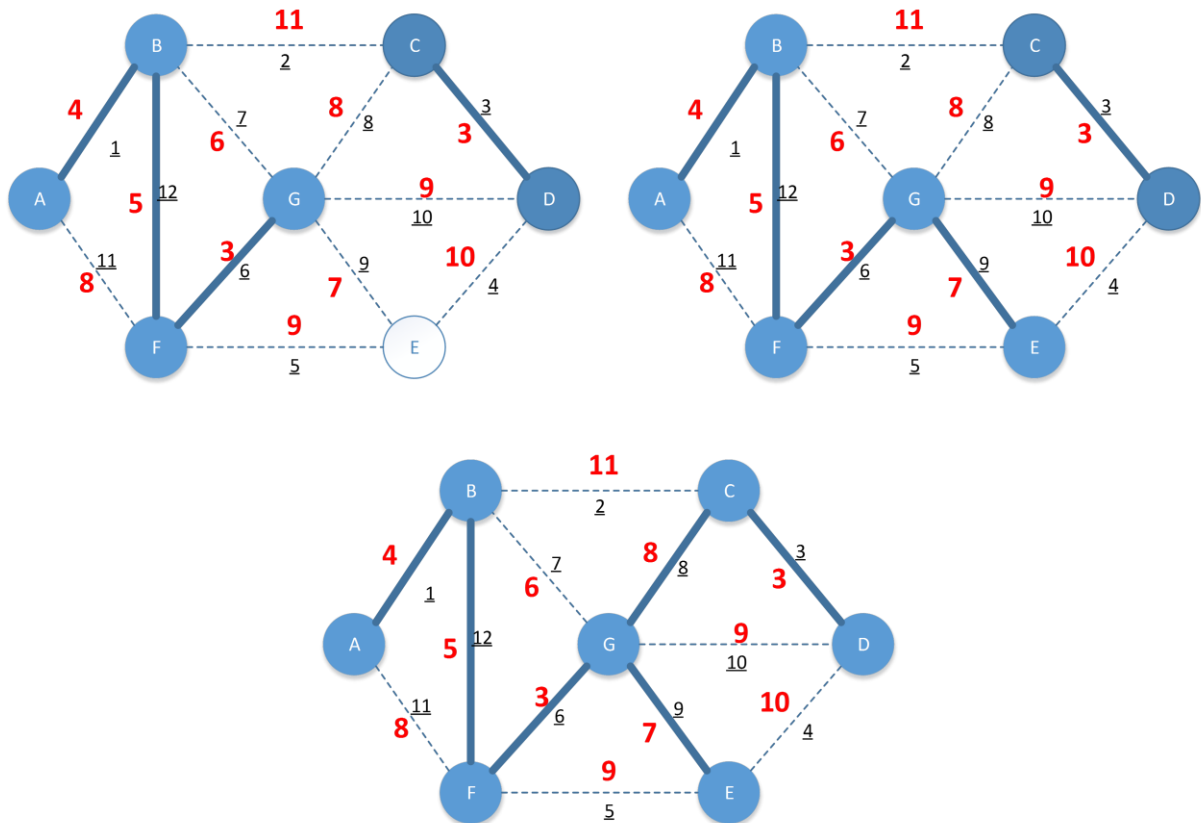


Fonte: Próprio autor.

A partir do grafo com pesos, a aresta com o peso menor é escolhida, neste caso temos duas arestas com o mesmo peso menor, decide-se escolher a aresta 3, que une os vértices C e D. Depois a outra aresta com mesmo peso menor é escolhida, que é a aresta 6 que une os vértices F e G. Em seguida, escolhe-se a próxima aresta segura de menor peso, que é a aresta 1, que une os vértices A e B. A seguir, são analisadas as arestas seguras candidatas e escolhe-se a aresta candidata com o menor peso, que é a aresta 12, que une os vértices B e F. Até este ponto, adicionamos 4 arestas, mas note que até este ponto não formam um grafo conexo. As arestas seguras candidatas são analisadas novamente e a de menor peso é escolhida, a escolhida é a aresta 9, que une os vértices E e G. Finalmente, as últimas arestas seguras candidatas são analisadas, essas arestas são a aresta 8, a aresta 10 e a aresta 4, entre estes 3 aquele com o peso menor é

escolhida, isto é a aresta 8, que une os vértices C e G. Como resultado, uma árvore com um peso mínimo é obtida.

Figura 11 – Execução do algoritmo Kruskal para um grafo com 7 vértices e 12 arestas (parte 2).



Fonte: Próprio autor.

3.2.3 Algoritmo de Prim

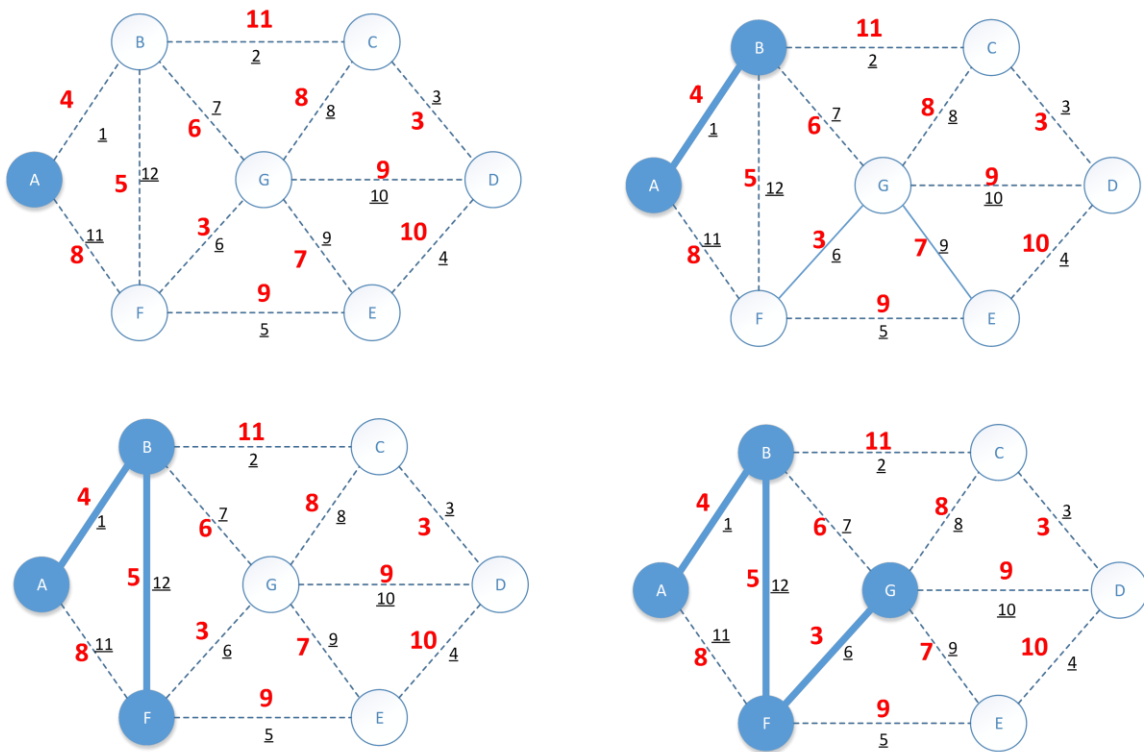
O algoritmo de Prim, ao contrário do algoritmo de Kruskal, forma sempre uma única árvore e tem um ponto de partida arbitrário.

O algoritmo de Prim, a partir do nó arbitrário e com os pesos atribuídos a cada aresta, adiciona à árvore a aresta segura com menor peso, sendo uma aresta por iteração até formar a árvore com o peso mínimo. As etapas seguidas por este algoritmo são as seguintes:

- (1) Determinar o vértice inicial;
- (2) Procurar, entre as arestas seguras, a aresta de menor peso e adicioná-la à árvore;
- (3) Continuar este processo até conectar todos os vértices.

As Figuras 12 e 13 mostram o procedimento que realiza o algoritmo de Prim; os números em preto são a numeração arbitraria atribuída a cada aresta e os números em vermelho são o peso de cada aresta.

Figura 12 – Execução do algoritmo do Prim para um grafo com 7 vértices e 12 arestas.

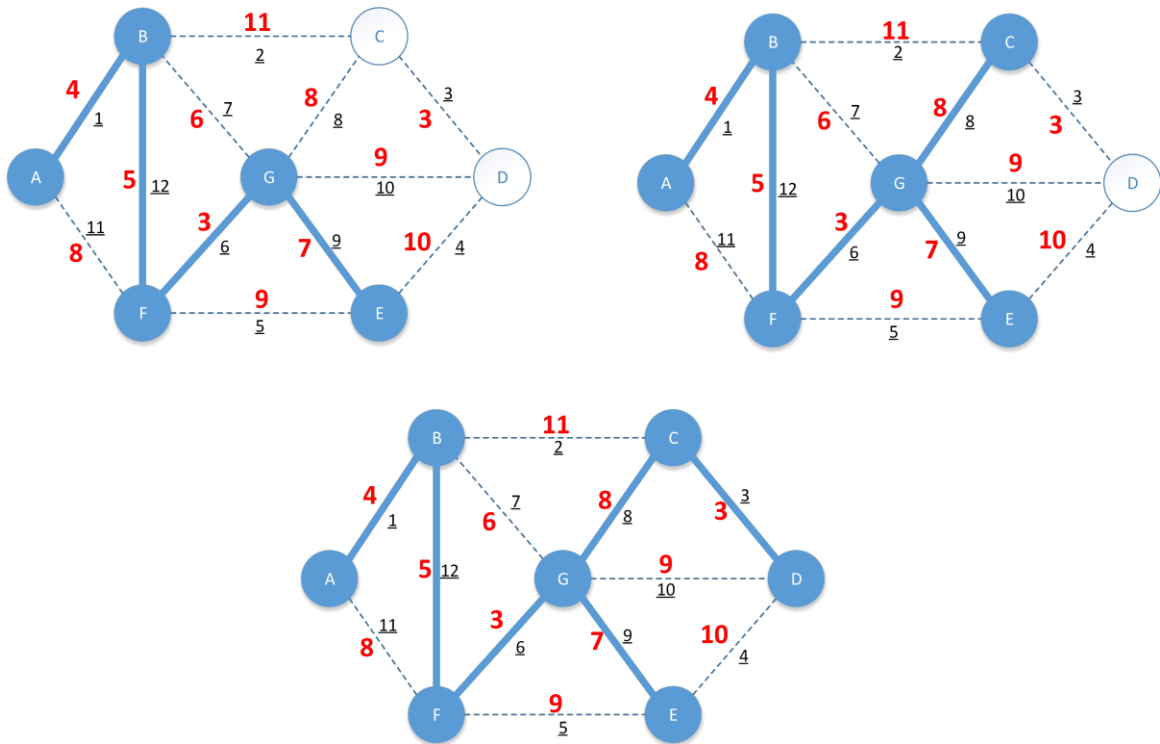


Fonte: Próprio autor.

No primeiro passo é escolhido arbitrariamente o vértice A. Depois, a partir desse vértice analisam-se as arestas candidatas: 1 e 11. Como a aresta 1 tem menor peso, então é escolhida para formar parte da árvore. A aresta 1 conecta o vértice A com B, então se analisam as novas arestas candidatas: 11, 12, 7 e 2. Assim, a aresta escolhida é a aresta 12, que tem o menor peso. A continuação, são analisadas as novas arestas candidatas: 2, 7 e 6; neste momento, a aresta 11 não é uma aresta segura porque se é adicionada ao arvore geraria um laço junto com as arestas adicionada até agora. A aresta escolhida e a aresta 6. Novamente analisam-se as novas arestas candidatas: 2, 8, 10, 9 e 5, sendo que a aresta escolhida é a 9. Depois de adicionar 9 à arvore, analisam-se as novas arestas cândidas: 8, 10 e 4. A aresta com menor peso é escolhida, assim 8 é

selecionada. Finalmente analisam-se as novas arestas candidatas, escolhendo a 3. No final obtém-se a árvore que conecta todos os vértices do grafo.

Figura 13 – Execução do algoritmo do Prim para um grafo com 7 vértices e 12 arestas



Fonte: Próprio autor.

3.3 NECESSIDADE DE RESOLVER PROBLEMAS DE FLUXO DE CARGA RADIAL

Uma vez que são geradas as propostas de solução, é preciso avaliar sua qualidade e viabilidade em relação às restrições operacionais. Assim, é necessário um algoritmo de fluxo de carga rápido e eficiente. Primeiro, será discutido de forma geral o problema de fluxo de carga e, em seguida, o processo de cálculo do algoritmo de fluxo de carga usado neste trabalho será explicado.

O cálculo do fluxo de carga (ou fluxo de potência) em uma rede de energia elétrica consiste essencialmente na determinação dos estados da rede, da distribuição dos fluxos e de outras grandezas de interesse. Para a resolução deste tipo de problema é utilizado um modelo

estático da rede, representado por meio de equações e desigualdades algébricas. As variações no tempo são consideradas lentas o suficiente para ignorar os efeitos transitórios.

As equações básicas do fluxo de carga são obtidas impondo a conservação das potências ativas e reativas em cada nó da rede, isto é, a energia líquida injetada deve ser igual à soma das potências que fluem através dos componentes internos que tem esse nó com um dos seus terminais, isso equivale a impor a primeira Lei Kirchhoff. A segunda lei de Kirchhoff é usada para expressar os fluxos de energia nos componentes internos como funções das tensões (estados) de seus nós terminais (MONTICELLI, 1983).

Existem vários tipos de algoritmos de fluxo de carga como o método de Newton e suas versões desacopladas, e o algoritmo Gauss-Seidel. No caso dos sistemas de distribuição, devido à topologia radial que apresentam quando operam, foram desenvolvidos algoritmos de fluxo de carga especializados. Esses métodos especializados buscam ser rápidos e eficientes. Para os sistemas de distribuição, existem os seguintes métodos: algoritmo de varredura, método de soma de potência (CESPEDES, 1990), algoritmo de Goswami-Basu (GOSWAMI; BASU, 1992) e algoritmo apresentado por Baran-Wu (BARAN; WU, 1989).

Para a avaliação das soluções propostas neste trabalho e para verificar a conformidade com as restrições operacionais, foi utilizado o método de varredura (SHIRMOHAMMADI et al., 1988) aplicado aos sistemas de distribuição radial. Este método é apresentado a continuação.

3.3.1 Método de varredura

Este método é chamado de varredura porque é percorrido um caminho a partir das barras mais afastas até a barra da subestação (processo “backward”) e, depois, da barra da subestação para as barras mais afastadas (processo “forward”). O processo é repetido até atingir um critério de parada. Neste trabalho, a parada desse algoritmo é feita quando a variação das perdas ativas em duas iterações consecutivas é menor que uma tolerância especificada.

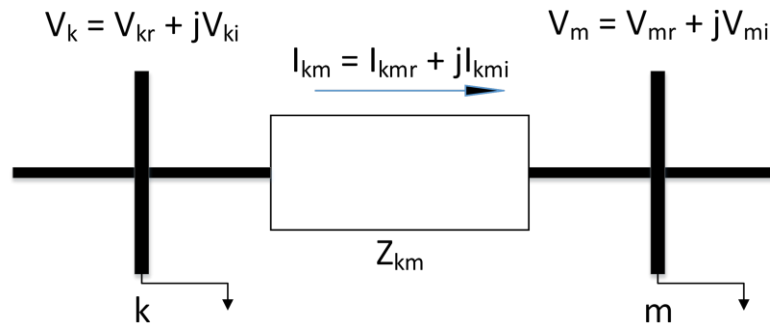
Antes de iniciar os cálculos para a aplicação deste método, deve ser realizada uma renumeração das barras e ramos de acordo com camadas segundo sua proximidade com a subestação. Todas as barras e ramos de uma camada devem ser renumerados antes de passar para a próxima camada.

Depois de fazer a renumeração das barras e os ramos do sistema, começa o processo de cálculo. Como primeiro passo, é escolhido um valor de tensão para todas as barras do sistema, esse valor é igual à tensão da subestação (V_{ref}), o que significa que $V_k = V_{ref} + j0$, para cada barra k do sistema.

3.3.1.1 Cálculo de correntes de carga

Com os valores de tensão atribuídos a todas as barras do sistema, os valores das correntes de carga podem ser calculados, a partir das barras mais afastadas e seguindo na direção da barra de subestação (processo “backward”). O cálculo das correntes de carga é feito a partir das expressões para a potência da carga e a tensão, dadas pelas equações (12) e (13) para uma barra k , como ilustrado na Figura 14.

Figura 14 – Sistema de duas barras.



Fonte: Próprio autor.

$$S_k = P_k + jQ_k \quad (12)$$

$$V_k = V_{kr} + jV_{ki} \quad (13)$$

Por definição, a potência da carga e dada por (14).

$$S_k = V_k I_k^* \quad (14)$$

A partir de (14) chegasse-se à equação para a corrente de carga (15).

$$I_k^* = \frac{P_k + jQ_k}{V_{kr} + jV_{ki}} \quad (15)$$

Da equação (15) pode ser derivada uma expressão para a parte real da corrente da carga (16) e uma expressão para a parte imaginária (17).

$$I_k^* = \frac{P_k + jQ_k}{V_{kr} + jV_{ki}} = \frac{P_k + jQ_k}{V_{kr} + jV_{ki}} \cdot \frac{V_{kr} - jV_{ki}}{V_{kr} - jV_{ki}} = \frac{(P_k V_{kr} + Q_k V_{ki}) - j(P_k V_{ki} - Q_k V_{kr})}{(V_{kr}^2 + V_{ki}^2)}$$

$$I_{kr} = \frac{(P_k V_{kr} + Q_k V_{ki})}{(V_{kr}^2 + V_{ki}^2)} \quad (16)$$

$$I_{ki} = \frac{(P_k V_{ki} - Q_k V_{kr})}{(V_{kr}^2 + V_{ki}^2)} \quad (17)$$

Da mesma forma, pode-se calcular a corrente de carga da barra m , nas suas partes real (18) e imaginária (19).

$$I_{mr} = \frac{(P_m V_{mr} + Q_m V_{mi})}{(V_{mr}^2 + V_{mi}^2)} \quad (18)$$

$$I_{mi} = \frac{(P_m V_{mi} - Q_m V_{mr})}{(V_{mr}^2 + V_{mi}^2)} \quad (19)$$

3.3.1.2 Cálculo de correntes em cada ramo do sistema

Considerando a barra m como uma barra terminal, então a corrente I_{km} , que passa pelo ramo que une as barras k e m , é igual a:

$$I_{km} = I_{kmr} + jI_{kmi} = I_{mr} + jI_{mi} \quad (20)$$

Dessa forma, a corrente I que entra na barra k é a soma da corrente de carga I_k mais a corrente I_{km} que passa pelo ramo $k - m$.

Fazendo essa suposição e generalizando para todo o sistema, uma vez que se conhecem todas as correntes de carga em cada barra do sistema, as correntes nos ramos são dadas por (21).

$$\vec{I}_{km} = \vec{I}_m + \sum_{ml \in \Omega_l} \vec{I}_{ml} \quad (21)$$

Onde m é a barra mais afastada da subestação no ramo $k - m$ e a soma corresponde à soma das correntes dos ramos vizinhos à barra m que pertencem às camadas inferiores.

3.3.1.3 Cálculo das tensões em cada barra do sistema

Partindo da barra da subestação para as barras mais afastadas, por meio de um processo para frente (forward) e conhecendo as correntes que passam por todos os ramos do sistema, assim, podem ser calculadas as tensões em cada barra usando (22). Desenvolvendo a equação (22), obtemos as expressões (23) e (24).

$$\vec{V}_k = \vec{V}_m + Z_{km} \cdot \vec{I}_{km} \quad (22)$$

$$\vec{V}_k = \vec{V}_m + (R_{km} + jX_{km})(I_{kmr} + jI_{kmi}) \quad (23)$$

$$\vec{V}_k = V_{mr} + jV_{mi} + (R_{km}I_{kmr} - X_{km}I_{kmi}) + j(X_{km}I_{kmr} + R_{km}I_{kmi}) \quad (24)$$

Conhecendo os valores de corrente (\vec{I}_{km}) e impedância (Z_{km}) para o ramo $k - m$, pode-se obter o valor da tensão na barra m (\vec{V}_m), expresso nas equações (25) e (26).

$$V_{mr} = V_{kr} - R_{km}I_{kmr} + X_{km}I_{kmi} \quad (25)$$

$$V_{mi} = V_{ki} - R_{km}I_{kmi} - X_{km}I_{kmr} \quad (26)$$

3.3.1.4 Cálculo de perdas no sistema

Depois de ter os valores das correntes para todos os ramos do sistema, podem-se calcular as perdas ativas e reativas da configuração radial que está sendo avaliada. As perdas ativas (P_{kmp}) no ramo $k - m$ são calculadas com a (27) e as perdas reativas (Q_{kmp}) por meio de (28).

$$P_{kmp} = R_{km}I_{km}^2 \quad (27)$$

$$Q_{kmp} = X_{km}I_{km}^2 \quad (28)$$

As perdas totais de potência em todos os ramos do sistema são calculadas segundo (29) e (30). Após calcular as perdas totais, verifica-se se o critério de parada foi satisfeito; em caso contrário, o processo iterativo continua até chegar à convergência do fluxo de carga.

$$P_{tr} = \sum_{(k,m) \in \Omega} R_{km}I_{km}^2 \quad (29)$$

$$Q_{ti} = \sum_{(k,m) \in \Omega} X_{km}I_{km}^2 \quad (30)$$

3.4 A RESTRIÇÃO DE RADIALIDADE

Como mencionado anteriormente, os sistemas de distribuição são planejados em forma de malha, mas sua operação é feita de forma radial devido a razões técnicas de operação. De acordo com este motivo, Lavorato et al. (LAVORATO et al., 2012) propõem que todos os modelos de otimização em sistemas de distribuição considerem a radialidade na sua formulação. A formulação matemática empregada neste trabalho, particularmente o uso das equações (2), (3) e (6) da seção 2.2, garante a topologia radial das configurações obtidas.

Em outras modelagens do problema de reconfiguração, essas condições radiais também estão incluídas na formulação matemática. No trabalho de (JABR; SINGH; PAL, 2012), para

garantir a radialidade das configurações encontradas, restrições da árvore geradora são introduzidas na modelagem. Da mesma forma, na proposta de (BORGHETTI, 2012) há também restrições que garantem a radialidade das configurações avaliadas.

3.5 ESPAÇO DE BUSCA REDUZIDO

No trabalho de Possagnolo (2015) é realizado um estudo sobre o tamanho do espaço de busca do problema de reconfiguração, isto, a partir do número de configurações radiais que podem ser geradas a partir do número de barras (nb) e do número de ramos (nr) de cada sistema. Esse estudo começa com o caso mais geral, em que cada ramo pode ser aberto ou fechado, pelo qual um total de 2^{nr} configurações são obtidas. Depois, é explorado o caso em que de todos os ramos (nr), sempre permanecem fechados ($nb - 1$) ramos, que é o número de ramos que devem ser fechados de modo que uma configuração radial seja obtida; este número de ramos fechados é uma condição necessária, mas não suficiente. Então, para encontrar esse número de configurações, calcula-se a combinatória de nr em $nb - 1$, com a expressão:

$$C(nr, nb - 1) = \frac{nr!}{(nb - 1)! (nr - nb + 1)!}$$

Em seguida, com base no teorema da matriz-árvore é identificado o número de árvores geradoras. O teorema da matriz-árvore diz que para um grafo G conexo, com a matriz de adjacência A e matriz de grau D e Laplaciana L , o número de árvores geradoras de G é igual a valor de qualquer cofator da matriz Laplaciana L . Assim, o número de árvores geradoras é calculado como:

$$L_{ij} = (-1)^{i+j} \det(L(i \setminus j)), \quad \text{em que} \quad L = D - A$$

É necessário destacar que a seguinte relação é satisfeita: $L_{ij} \ll C(nr, nb - 1) \ll 2^{nr}$, isto é, o número de árvores geradoras é muito menor que as combinações de ramos fechados. Conclui-se então que uma estratégia que sempre gera uma proposta de solução viável do ponto de vista da radialidade, considera um espaço de busca reduzido e, em consequência, a busca será mais eficiente (POSSAGNOLO, 2015).

Esta dissertação procura alcançar esse objetivo usando um algoritmo de árvore geradora para obter cada uma das propostas de solução a serem avaliadas, garantindo a radialidade de todas as configurações a serem avaliadas.

4 O ALGORITMO GENÉTICO ESPECIALIZADO DE CHAVES ALEATÓRIAS VICIADAS

Neste capítulo será apresentado o algoritmo genético a ser utilizado na otimização do problema de reconfiguração dos sistemas de distribuição visando a redução das perdas. Antes de explicar o algoritmo genético especializado usado neste trabalho, se analisará o algoritmo genético tradicional, com suas características e operadores. Então, as características e particularidades do algoritmo genético das chaves aleatórias viciadas serão discutidas, o que permitirá apreciar as diferenças marcantes entre os algoritmos.

4.1 O ALGORITMO GENÉTICO

O algoritmo genético é uma meta-heurística que John Holland introduziu em seu livro *Adaptação em sistemas naturais e artificiais* (HOLLAND, 1975). Holland capturou em seu livro as semelhanças que ele encontrou entre o processo de seleção natural e evolução das espécies e a maneira de resolver um problema de otimização matemática. Na natureza, o processo de seleção natural e a evolução das espécies são consequência de um processo de otimização estocástica que ocorre em um determinado ambiente e em tempo real; então um algoritmo genético apenas imita uma parte dos componentes desse processo.

Com base no texto de Holland, e nos trabalhos de seus colaboradores Ken De Jong e David Goldberg, foi criado um campo de pesquisa e aplicação para resolver problemas complexos de pesquisa operacional.

De Jong foi um dos primeiros autores que mostrou interesse na aplicação do algoritmo genético (AG) no campo da otimização, mas no artigo (DE JONG, 1993) estabeleceu que os algoritmos genéticos não são realmente otimizadores de funções, mas que são acessórios no assunto da adaptação. Entretanto, atualmente o uso de algoritmos genéticos é bastante popular e bem-sucedido na otimização de problemas combinatórios.

4.1.1 Conceitos básicos

Uma das características dos algoritmos genéticos que os diferenciam de outras heurísticas é que permitem separar a representação do problema em relação às variáveis do problema. Isso é

feito através do uso de dois termos biológicos: o genótipo, que é a representação codificada das variáveis, e o fenótipo, que é o conjunto de variáveis específicas do problema. Nos algoritmos genéticos existe uma população de vetores e cada vetor é chamado de cromossomo. Cada elemento de um cromossomo é um gene e cada valor que esse gene leva é chamado de alelo. Assim, um algoritmo genético através de várias gerações busca a evolução de uma população formada por cromossomos que representam uma solução proposta para o problema que está sendo otimizado. O processo seguido por um algoritmo genético básico é o seguinte:

- 1) Etapa preliminar: Escolher uma forma de codificação para as propostas de solução e estabelecer a maneira de avaliar a função objetivo; escolher os parâmetros do algoritmo, como o tamanho da população, a taxa de recombinação, a taxa de mutação e o tipo de seleção; determinar o critério de parada;
- 2) Gerar a população inicial;
- 3) Avaliar a qualidade de todos os elementos da população e atualizar a incumbente;
- 4) Pare, se o critério de parada foi satisfeito. Caso contrário ir para o passo 5;
- 5) Implementar a etapa de seleção de indivíduos que entrarão no processo de recombinação;
- 6) Implementar o operador de recombinação;
- 7) Implementar o operador de mutação, atualizar a população e retornar ao passo 3.

4.1.2 Codificação da proposta de solução

Para a codificação das propostas de solução no espaço de busca do problema, inicialmente foi proposto usar uma codificação binária, mas atualmente a codificação segue a proposta de (MICHALEWICZ, 1996) que sugere que esta codificação deve seguir a natureza e as características do problema.

4.1.3 Seleção no algoritmo genético

A seleção é um processo pelo qual uma parte da população é escolhida para participar do processo para gerar a próxima geração. A ideia básica da seleção é que ela deve estar relacionada à adequação dos indivíduos. Os métodos de seleção são usados para direcionar o processo de seleção para melhores regiões no espaço de busca (BÄCK; FOGEL; MICHALEWICZ, 2000).

No esquema original, sua implementação era conhecida como o método da roleta. Este método usa uma distribuição de probabilidade para seleção, na qual a probabilidade de seleção de

um cromossomo é proporcional ao seu fitness (função objetivo). Assim, cada um dos candidatos da população recebe uma parte proporcional, onde os melhores indivíduos receberão uma maior parcela da roleta e de tal forma que a soma de todas as percentagens seja a unidade. Para selecionar um indivíduo, um valor aleatório entre 0 e 1 é gerado para simular uma probabilidade e, usando esse valor, localize o candidato de acordo com sua posição na roleta, onde a probabilidade cumulativa é mostrada.

Outro método de seleção utilizado é o método do torneio, que busca fazer a seleção com base em comparações diretas entre candidatos escolhidos aleatoriamente. Neste método, os descendentes são escolhidos para executar n_p jogos, em cada jogo são escolhidos aleatoriamente k indivíduos, e o candidato vencedor desses jogos, terá o direito de fazer parte do processo de formação de uma nova população. Sendo escolhidos aleatoriamente para competir, não há nenhum favoritismo para os indivíduos com melhor fitness.

Além dos dois métodos mencionados, pode-se usar uma seleção elitista, que consiste em transferir os melhores elementos da população atual para a população da próxima geração e, assim, evitar a perda de informações valiosas durante a aplicação dos operadores genéticos

Além desses métodos, existem outros, que podem procurar a participação de candidatos de maior qualidade ou, por exemplo, um método que melhore a eficiência computacional.

4.1.4 Operadores no algoritmo genético

4.1.4.1 Recombinação

O processo de recombinação consiste em combinar alguns genes de um dos pais com os genes de o outro pai e assim formar novos indivíduos. Ao aplicar este operador iterativamente, geração após geração, espera-se que os genes pertencentes aos indivíduos de melhor qualidade apareçam com maior frequência e que, eventualmente, eles convirjam para uma solução de excelente qualidade. Existem várias formas de recombinação, entre as quais estão:

- **Recombinação de um ponto:** neste tipo de recombinação, um ponto de corte é escolhido aleatoriamente e, a partir desse ponto, a informação genética dos pais será trocada. Assim, os descendentes terão os primeiros genes iguais a um dos pais e os outros genes iguais ao segundo.

- Recombinação de dois pontos: neste caso, 2 pontos de corte são escolhidos aleatoriamente e, a partir destes, a informação genética dos pais é trocada. A informação genética incluída entre os pontos de corte permanecerá invariante, tendo um descendente a informação genética de um dos pais e o outro descendente a informação genética do outro pai.

Dessa forma, as recombinações podem ser feitas em X pontos. De fato, há evidências de que uma recombinação de 8 pontos seja a melhor, em termos do número de avaliações de função necessárias para alcançar um ótimo global, como um media em uma variedade de tipos de problemas (GENDREAU; POTVIN, 2010).

- Recombinação uniforme: é uma forma de recombinação que busca fazer a recombinação um processo completamente aleatório. Nesta forma de recombinação, para cada gene de um descendente, existe uma certa probabilidade de herdar o gene do primeiro pai como herdando o gene do segundo pai.
- Recombinação não linear: este tipo de recombinação pode ser aplicado em certos tipos de problemas. Um caso de aplicação está nos problemas em que o espaço da solução é o espaço de permutação dos números, como é o caso do problema do agente viajante. Neste problema, são selecionados dois pontos de corte e os elementos compreendidos entre esses pontos, permutam em ordem.

4.1.4.2 Mutação

O operador de mutação geralmente tem uma probabilidade de ocorrência por isto uma taxa de mutação pode ser aplicada. A mutação introduz variações aleatórias em alguns genes de um indivíduo, com o objetivo de contribuir com uma diversidade genética para a população, além de ajudar o algoritmo genético a escapar dos mínimos locais.

4.1.5 Parâmetros dos algoritmos genéticos

O algoritmo genético tradicional tem poucos parâmetros, mas estes podem ser muito sensíveis e sua seleção correta pode ajudar a uma convergência mais rápida. Esses parâmetros são: o tamanho da população, o número de gerações, a taxa de recombinação e a taxa de mutação.

4.2 O ALGORITMO GENÉTICO DE CHAVES ALEATÓRIAS VICIADAS

Nesta parte do capítulo é apresentada a meta-heurística usada na otimização do problema de reconfiguração, uma visão geral, seus componentes, a aplicação de operadores genéticos.

4.2.1 Visão geral

O Algoritmo Genético de Chaves Aleatórias Viciadas (BRKGA) é uma meta-heurística evolutiva de propósito geral desenvolvida por J.F. Gonçalves e Resende (GONÇALVES; RESENDE, 2011). É uma variante do algoritmo genético de chaves aleatórias (RKGA) desenvolvida por J.C. Bean (BEAN, 1994).

Uma característica importante desta meta-heurística é que permite fazer uma divisão entre uma parte independente do problema e uma parte dependente do problema. Com isso, uma parte do software pode ser reutilizada e permite que o designer do algoritmo se concentre na parte específica do problema que consiste na construção de um decodificador.

Da mesma forma que um algoritmo genético tradicional, o BRKGA evolui uma população, neste caso de vetores de chaves aleatórias, ao longo de várias gerações.

A população inicial é composta de um número p de vetores de chaves aleatórias. Cada alelo do cromossomo (vetor) é uma chave aleatória gerada independentemente no intervalo real $[0,1]$. Na Figura 15 mostra-se um exemplo de um vetor de chaves entre 0 e 1. Após a execução do decodificador e o cálculo do valor de *fitness* de cada indivíduo, a população é dividida em dois grupos de indivíduos:

- Um pequeno grupo de indivíduos de elite p_e , onde $p_e < p - p_e$
- Um grupo de indivíduos não-elite ($p - p_e$)

Figura 15 – Vetor de chaves com valores entre 0 e 1.

0.038	0.226	0.017	0.067	0.252	0.907	0.174
-------	-------	-------	-------	-------	-------	-------

Fonte: Próprio autor.

Para evoluir a população, uma nova geração de indivíduos deve ser gerada. Um BRKGA usa uma estratégia elitista uma vez que todos os indivíduos de elite da geração k são copiados para a geração $k + 1$. Em cada geração, um pequeno número p_m de mutantes é introduzido na população. Com os indivíduos de elite p_e e os mutantes p_m representados na população $k + 1$; precisam ser produzidos $p - p_e - p_m$ indivíduos adicionais para completar os p indivíduos que

compõem a população da geração $k + 1$. Isso é feito produzindo $p - p_e - p_m$ descendentes através do processo de recombinação.

4.2.2 Seleção

No algoritmo BRKGA, depois de ter feito a avaliação da população e ter feito a separação entre elementos elite e não elite, é realizada a seleção dos elementos que participarão do processo de recombinação. Este processo de seleção é sinalizado pelos criadores desta meta-heurística, como um processo "falho". É chamado de "defeituoso" porque busca a maior participação dos elementos de elite, considerando que um dos elementos que participaram do processo de recombinação deve pertencer necessariamente à população elite. Esta é uma das características que a diferenciam do Algoritmo Genético de Chaves Aleatórias (RKGA), na qual a seleção é completamente aleatória entre todos os indivíduos da população; a outra diferença com o algoritmo RKGA é a consideração de uma probabilidade ($0,5 < \rho_e$) de herdar as características do elemento elite.

Então, no algoritmo BRKGA, para o processo de recombinação, são escolhidos, de forma aleatória, um elemento da população de elite e outro elemento da população não elite. Embora seja também o caso, o segundo elemento pode ser escolhido da população total (elite e não elite).

4.2.3 Operadores genéticos no BRKGA

O algoritmo genético das chaves aleatórias viciadas também apresenta dois dos operadores genéticos presentes no algoritmo genético tradicional, que são a recombinação e a mutação. Mas, como será explicado nas seções a seguir, elas não são executadas da mesma forma que o algoritmo genético tradicional, e no caso do operador de mutação a diferença é mais marcante.

4.2.3.1 Recombinação em BRKGA

No BRKGA, cada elemento é gerado combinando um elemento selecionado aleatoriamente da partição denominada elite na população atual e uma da partição rotulada como não-elite ou de toda a população. A repetição na seleção de um elemento é permitida e, portanto, um indivíduo pode produzir mais de um descendente.

Uma vez que é exigido que $p_e < p - p_e$, isto é, a probabilidade de que um determinado indivíduo de elite seja selecionado para o processo de recombinação ($1/p_e$) seja maior do que a

de um determinado indivíduo não elite ($\frac{1}{p-p_e}$), o elemento de elite tem maior probabilidade de transmitir suas características às gerações futuras do que um indivíduo não elite.

Seja $\rho_e > 0.5$ um parâmetro escolhido pelo usuário. Este parâmetro é a probabilidade de que o descendente herde o alelo de seu pai de elite. Seja n o número de genes no cromossomo, para $i = 1, \dots, n$ o i -ésimo alelo $c(i)$ do filho c assume o valor do i -ésimo alelo $a(i)$ do pai elite com probabilidade ρ_e , e toma o valor do i -ésimo alelo $b(i)$ do pai não-elite com uma probabilidade de $1 - \rho_e$. Desta forma, é mais provável que o descendente herde as características do pai de elite do que as do pai não elite. Tendo em conta que qualquer vetor de chave aleatória pode ser decodificado em uma solução do problema de otimização combinatória.

Figura 16 – Processo de recombinação de dos cromossomos.



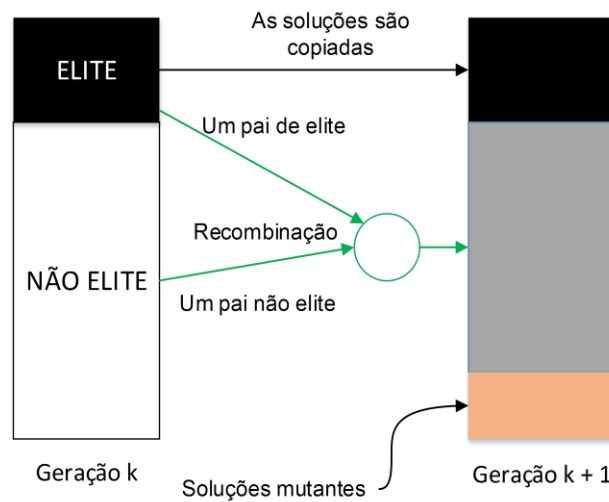
Fonte: Próprio autor.

A Figura 16 ilustra o processo de recombinação para dois vetores de chaves aleatórias com quatro genes cada. O cromossomo 1 refere-se ao indivíduo de elite e o cromossomo 2 ao não-elite. Neste exemplo, o valor de $\rho_e = 0,7$ o descendente herda o alelo do pai de elite com probabilidade 0,7 e do outro pai com probabilidade 0,3. Um número real gerado aleatoriamente no intervalo $[0,1]$ simula o lançamento de uma moeda viciada. Se o resultado for menor ou igual a 0,7, o descendente herda o alelo do pai elite. Caso contrário, ele herda o alelo do outro pai. Neste exemplo, o descendente herda o alelo do pai da elite em seus primeiro, terceiro e quarto genes, então ele tem mais semelhança com o pai de elite.

4.2.3.2 Mutação

Nos algoritmos genéticos, a mutação é o operador cujo objetivo é prevenir a convergência prematura da população. No BRKGA, o operador de mutação também é aplicado de maneira diferente em comparação com outros algoritmos genéticos. Neste caso, em vez de realizar mutação alelo por alelo, com uma probabilidade muito pequena em cada geração, são introduzidos novos indivíduos na próxima geração (Figura 17). Estes novos indivíduos são criados da mesma forma que a população inicial, de modo que nenhum material genético da população atual é trazido. Pode-se pensar sobre esses mutantes como sendo imigrantes. Uma vez que a população da geração $k+1$ está completa, os valores de aptidão de todos os vetores que foram recentemente incorporados são calculados e o processo de evolução da população é repetido até atingir o critério de parada.

Figura 17 – Mutação no BRKGA



Fonte: Próprio autor

As heurísticas BRKGA são baseadas em uma estrutura meta-heurística de propósito geral. Nesta estrutura, representada na Figura 18, há uma divisão clara entre a parte independente do problema do algoritmo e a parte dependente do problema. A parte independente do problema não tem conhecimento do problema que está sendo resolvido. É limitado a procurar o hipercubo. A única conexão ao problema de otimização combinatória que está sendo resolvida é a porção dependente do problema do algoritmo, onde o decodificador produz soluções dos vetores de

chaves aleatórias e calcula a adequação dessas soluções. Portanto, para especificar uma heurística BRKGA, é preciso apenas definir a representação cromossômica e o decodificador.

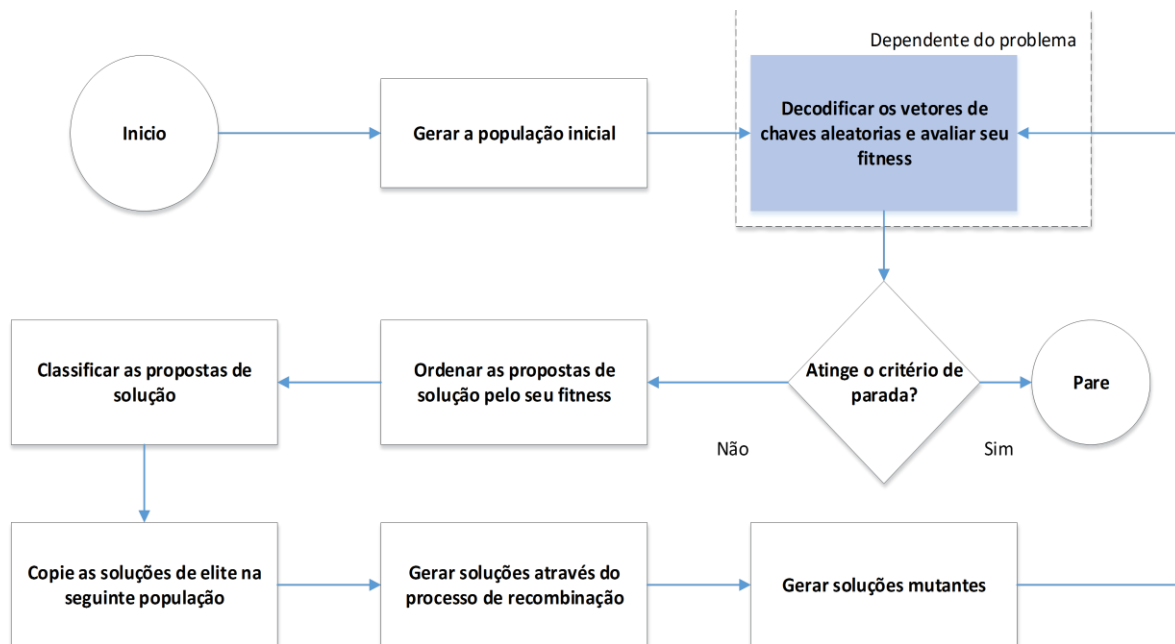
4.2.4 Componentes do BRKGA

Como mencionado anteriormente, o BRKGA tem dois módulos, um módulo independentemente do problema que é indiferente do problema que se deseja otimizar, e pode ser reutilizado para a otimização de outros problemas; o que permite focar no projeto e implementação do módulo dependente do problema, que é o decodificador.

4.2.4.1 Módulo independente do problema

Os componentes deste módulo dependem do número de genes no cromossomo de um indivíduo (n), o número de elementos na população (p), o número de elementos de elite na população (p_e), o número de mutantes (p_m) introduzido em cada geração e a probabilidade de herdar o alelo do pai elite (ρ_e).

Figura 18 – Fluxograma do BRKGA



Fonte: Próprio autor

Um dos componentes deste módulo é a matriz pop com dimensão $p \times n$ que armazena a população a ser evoluída.

Outro componente é o vetor *fitness*, que armazena os valores de aptidão obtidos depois de avaliar cada indivíduo na população.

Cada geração de BRKGA consiste nas seguintes etapas:

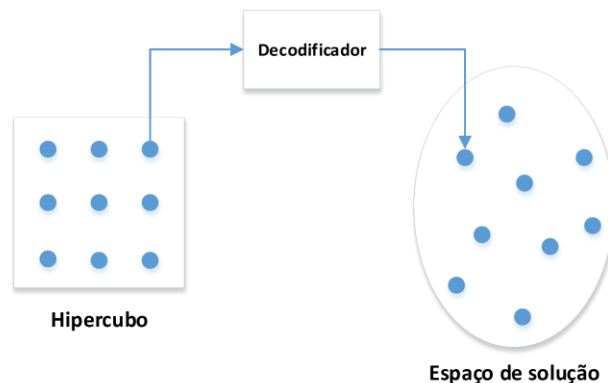
- (1) Ordenar o array *fitness* em ordem crescente.
- (2) O processo de recombinação é realizado para obter $p - p_e - p_m$ descendentes, esses descendentes serão temporariamente armazenados na matriz *tmppop*
- (3) Os p_m cromossomos mutantes são gerados e armazenados na matriz *pop*
- (4) Copiar os descendentes da recombinação na matriz *pop*
- (5) Avaliar o fitness dos cromossomos incorporados recentemente na matriz *pop*.

Esse processo é repetido até atingir o critério de parada. Cada iteração é chamada de geração. O critério de parada pode ser um número fixo de gerações, um número fixo de gerações após a última melhoria na solução, um limite de tempo ou depois de alcançar uma solução pelo menos tão boa como um valor determinado.

4.2.4.2 Módulo dependente do problema - Decodificador

O decodificador (Figura 19) é responsável por fazer a conexão entre as soluções no hipercubo e as funções objetivo de suas soluções correspondentes no espaço de solução do problema de otimização combinatória.

Figura 19 – Função realizada pelo decodificador.



Fonte: Próprio autor

Este módulo é deixado para a imaginação do pesquisador que procura otimizar o problema. A decodificação pode variar em sua complexidade, desde uma simples ordenação dos

valores de chaves aleatórias até heurísticas complexas com busca local. O mesmo problema pode ser otimizado usando diferentes descodificadores.

4.2.4.3 Ajuste de parâmetros

O BRKGA tem cinco parâmetros a serem ajustados. Estes parâmetros são: número de genes no cromossomo de um indivíduo (n), o número de elementos na população (p), o número de elementos de elite na população (p_e), o número de mutantes (p_m) introduzido em cada geração e a probabilidade de herdar o alelo do pai elite (ρ_e). Com base na experiência dos criadores desta meta-heurística, os parâmetros sugeridos são mostrados na Tabela 1.

Tabela 1 – Parâmetros sugeridos no BRKGA.

Parâmetro	Descrição	Valor atribuído
p	Tamanho da população	$p = a * n$, onde $1 \leq a \in \mathbb{R}$ é uma constante e n o comprimento do cromossomo
p_e	Tamanho da população de 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 herança do alelo elite	$0.5 < \rho_e \leq 0.8$

Fonte: Próprio autor.

4.2.4.4 População inicial

Para otimizar um problema a população inicial não deve necessariamente ser composta por chaves aleatórias; uma vez que existe a liberdade de introduzir propostas de soluções de melhor qualidade obtidas com alguma heurística na população inicial, para acelerar o processo de convergência.

5 O ALGORITMO GENÉTICO DE CHAVES ALEATÓRIAS VICIADAS APLICADO AO PROBLEMA DE RECONFIGURAÇÃO

Neste capítulo será explicado em detalhes a aplicação do algoritmo genético de chaves aleatórias viciadas ao problema da reconfiguração. Começa-se com a explicação da estrutura geral do algoritmo, depois o processo para gerar a população inicial, seguido da aplicação do decodificador aos vetores de chaves aleatórias, a avaliação das soluções propostas e a aplicação dos operadores genéticos.

5.1 ESTRUTURA GERAL DO ALGORITMO

O algoritmo proposto para otimizar o problema de reconfiguração, é descrito abaixo:

Iniciação: Definir os parâmetros: número de genes em um cromossomo (n), o tamanho da população (p), o tamanho da população de elite (p_e), o tamanho da população mutante (p_m), a probabilidade de herança do alelo de elite (ρ_e). Geração da população inicial, composta por vetores de chaves aleatórias entre 0 e 1, cada vetor tem um comprimento igual ao número total de ramos do sistema analisado (n).

Repetir: as etapas (1), (2), (3), (4), (5), (6) até que o critério de parada seja atingido:

- (1) Decodificação de cada vetor de chaves aleatórias através do algoritmo de Prim, obtendo apenas configurações radiais;
- (2) Avaliação da função objetivo de cada configuração obtida, isto é, cálculo das perdas do sistema, aplicando um fluxo de carga radial de varredura. Se atinge o critério de parada, pare, se não siga para a seguinte etapa;
- (3) Classificação de propostas de solução de acordo com a função objetivo (perdas ativas);
- (4) Copiar e mover as soluções de elite para a população da geração $k + 1$;
- (5) Aplicar o operador de recombinação para a geração de $(p - p_m)$ descendentes que serão parte da população da geração $k + 1$;
- (6) Completar a população da geração $k + 1$ com p_m indivíduos mutantes.

5.2 SELEÇÃO DE PARÂMETROS

O algoritmo genético das chaves aleatórias viciadas tem parâmetros que devem ser calibrados adequadamente, 3 dos quais (p , p_e , p_m) têm uma dependência do valor do número de genes no cromossomo (n).

5.3 GERAÇÃO DA POPULAÇÃO INICIAL

Para gerar a população inicial do algoritmo, são gerados vetores de chaves aleatórias entre 0 e 1. Esses valores são armazenados em uma matriz ($p \times m$), onde p representa o número de indivíduos da população e m representa o número de ramos totais (ligados e desligados).

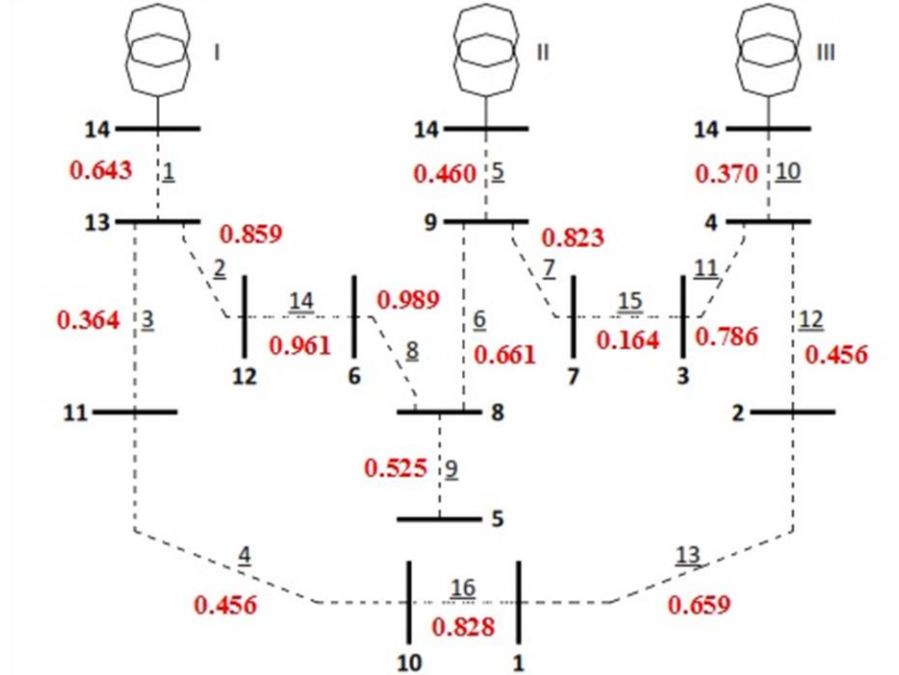
Cada uma dessas chaves geradas se tornará um peso para cada um dos ramos do sistema para reconfigurar. O primeiro valor do vetor da chave aleatória será o peso atribuído ao primeiro ramo do sistema, o segundo valor do vetor será o peso atribuído ao segundo e assim todos os pesos serão atribuídos a cada um dos ramos. Com esses pesos, o decodificador pode fazer a conexão entre as soluções no hipercubo unitário e o valor das funções objetivo de suas soluções correspondentes no espaço de soluções do problema de RSDEE.

Com o intuito de ilustrar a atribuição de pesos aos ramos e a representação do vetor de chaves aleatórias, é utilizado o sistema teste de 14 barras (CIVANLAR et al., 1988). Na Figura 20 são mostrados os pesos atribuídos a cada um dos ramos desse sistema, enquanto que na Figura 21 mostra-se o vetor de chaves aleatórias correspondente.

5.4 DECODIFICAÇÃO DAS PROPOSTAS DE SOLUÇÃO

O decodificador permite converter cada vetor de chaves aleatórias em uma representação radial que será avaliada. O decodificador que foi implementado é um algoritmo gerador de árvores, especificamente uma modificação do algoritmo de Prim para adaptá-lo ao problema de reconfiguração de sistemas de distribuição. Em vez de procurar a minimização dos pesos, propõe-se uma maximização. Os pesos atribuídos a cada nó serão os valores do vetor de chaves aleatórias. A estrutura dessa modificação foi usada no trabalho de Carreño, Romero e Padilha-Feltrin (CARRENO; ROMERO; PADILHA-FELTRIN, 2008) e sua operação é explicada abaixo.

Figura 20 – Sistema de 14 barras com pesos atribuídos a cada um dos seus ramos



Fonte: Próprio autor.

Figura 21 – Vetor de chaves aleatórias para o sistema de 14 barras.

0.64	0.85	0.36	0.45	0.46	0.66	0.82	0.98	0.52	0.37	0.78	0.45	0.65	0.96	0.16	0.82
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

Fonte: Próprio autor.

Seja D o conjunto de barras adicionado à configuração radial na fase de construção e N o conjunto de ramos do sistema não escolhido no processo de geração da topologia radial. O Algoritmo de Prim modificado executa os seguintes passos:

- (1) $D = \emptyset, N_1 = \emptyset, N_2 = \emptyset$. N é formado inicialmente por todos os ramos;
- (2) Atribuir o nó da subestação ao nó D ;
- (3) Identificar todos os ramos pertencentes a N com um dos seus vértices em D ;
- (4) Escolher um ramo identificado no passo (3). Neste trabalho será escolhido o ramo com maior peso;

- (5) Se o ramo selecionado em (4) gerar um loop, incluir o ramo em N_2 ; caso contrário, incluir o vértice oposto em D e o ramo selecionado em N_1 . Atualizar N removendo o ramo escolhido;
- (6) Se o número de elementos de N_1 for menor do que o número de barras menos 1, então voltar para o passo (3);
- (7) Se $N \neq \emptyset$, adicionar todos os elementos de N a N_2 .

No final da aplicação $[N_1, N_2]$ representa a topologia obtida. Ao aplicar este algoritmo, somente configurações radiais serão obtidas a partir dos vetores de chaves aleatórias.

Para o sistema teste de 14 barras o procedimento é o seguinte:

O processo de decodificação tem como ponto de partida a barra da subestação (barra 14) conforme a Figura 20. Em seguida, de acordo com os dados da Figura 21, será avaliado qual dos ramos conectados à subestação será adicionado à árvore; são comparados os pesos dos ramos candidatos e seleciona-se o ramo com o maior peso (no exemplo seria o ramo 1, que conecta a barra 14 com a barra 13). Após a escolha, são comparados os novos ramos candidatos (neste caso as linhas 2, 3, 5 e 10 seriam candidatas). Agora, comparam-se os pesos desses ramos e é anexado à árvore aquele com o maior peso (neste caso o ramo com o maior peso é o ramo 2, que conecta a barra 13 com a barra 12). Continua-se com este processo até formar uma árvore que conecta todas as barras do sistema.

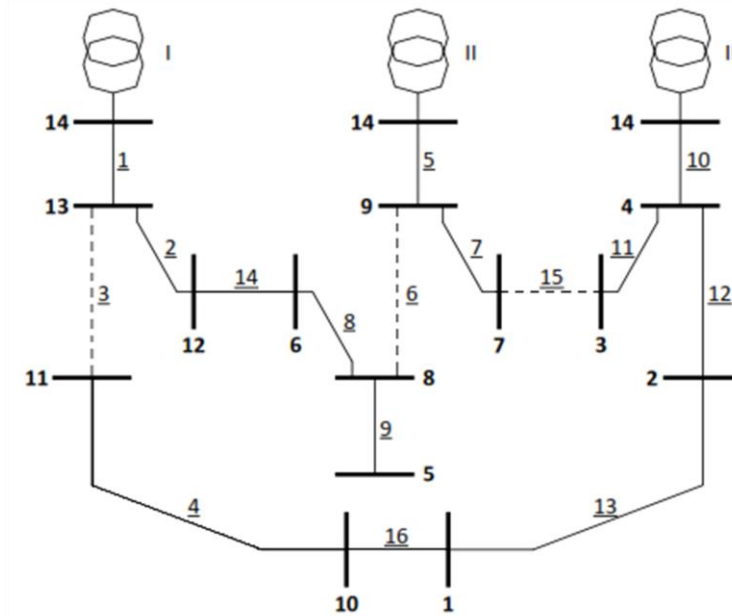
A saída do decodificador (algoritmo Prim) é uma configuração radial, representada por um vetor de zeros e uns, que indica quais ramos do sistema estão energizados (com um valor igual a 1) e quais estão desenergizados (com um valor igual a 0). A Figura 22 mostra o vetor obtido após a decodificação do vetor de chaves aleatórias. A Figura 23 mostra a configuração radial do sistema teste obtido ao considerar os ramos energizados (valor 1) e os ramos desenergizados (valor 0).

Figura 22 – Vetor binário que mostra os ramos ligados e desligados.

1	1	0	1	1	0	1	1	1		1	1	1	1	1	0	1
---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---

Fonte: Próprio autor.

Figura 23 – Configuração radial obtida a partir do vetor da Figura 20.



Fonte: Próprio autor.

5.5 AVALIAÇÃO DA FUNÇÃO OBJETIVO

Com o vetor de chaves aleatórias decodificado, procede-se ao cálculo da função objetivo, ou seja, as perdas de energia ativa nesta configuração. Para fazer esta avaliação, utiliza-se um programa de fluxo de carga de varredura na configuração radial obtida. Nesta fase do procedimento, também é avaliado se a configuração radial estiver em conformidade com as restrições operacionais, descartando as configurações que violam qualquer uma dessas restrições.

5.6 CLASSIFICAÇÃO DAS PROPOSTAS DE SOLUÇÃO

Nesta etapa, uma vez que a avaliação de todas as propostas de solução é realizada, uma classificação em ordem crescente de acordo com as perdas é feita. A primeira solução desse arranjo é a configuração com menos perdas e, portanto, o melhor indivíduo da população. Após esse ordenamento, é feita uma divisão da população em duas partes: uma parte composta das (p_e) melhores configurações (aquelas que apresentam perdas e são viáveis a partir do ponto operacional) e as configurações ($p - p_e - p_m$) de menor qualidade.

5.7 IMPLEMENTAÇÃO DE OPERADORES GENÉTICOS

Neste tópico existe uma diferença entre os algoritmos genéticos tradicionais e o algoritmo BKRGGA. Esta parte do algoritmo genético de chaves aleatórias pertence ao módulo da parte independente do problema, portanto, é indiferente ao problema que está sendo resolvido.

5.7.1 Recombinação

O processo de recombinação em algoritmos genéticos busca a troca de genes entre a população atual para encontrar descendentes mais adequados. Para o processo de recombinação em BRKGA, dois indivíduos da população atual são escolhidos aleatoriamente; um desses indivíduos pertence à população de elite e o outro da população total, podendo existir o caso em que ambos os indivíduos pertencem à população elite. Devido a este sistema de seleção, é necessário que pelo menos um dos indivíduos que entram no processo de recombinação seja um indivíduo de elite, então a probabilidade de que os genes pertencentes à população de elite sejam passados para o novo descendente é maior. Após selecionar os dois indivíduos, o parâmetro de ρ_e (probabilidade de herança do alelo de elite) é considerado para estabelecer qual dos alelos dos pais herdará cada descendente, através da aplicação da recombinação uniforme parametrizada.

Na Figura 24 é ilustrado o processo de recombinação para o sistema de 14 barras.

Figura 24 – Processo de recombinação no sistema de 14 barras.

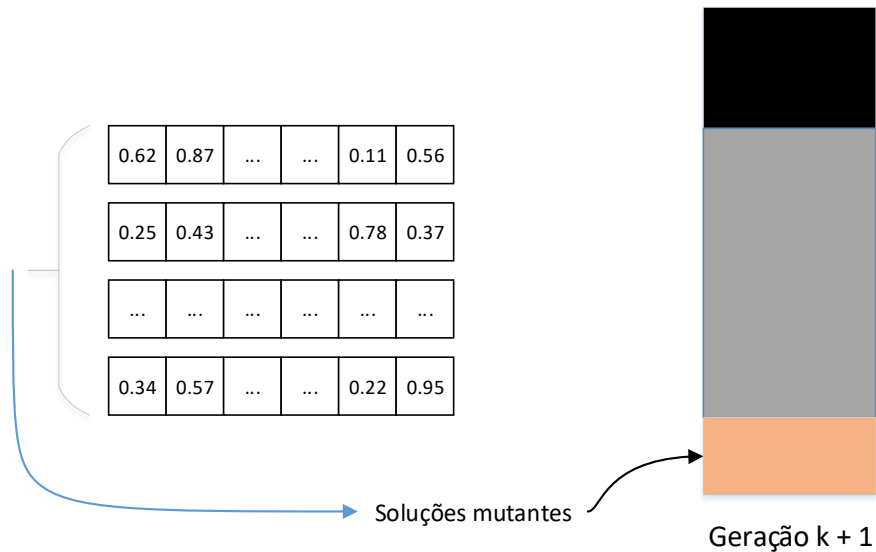
Pai 1															
0.64	0.85	0.36	0.45	0.46	0.66	0.82	0.98	0.52	0.37	0.78	0.45	0.65	0.96	0.16	0.82
Pai 2															
0.31	0.03	0.31	0.62	0.86	0.57	0.67	0.90	0.36	0.03	0.95	0.38	0.71	0.85	0.66	0.33
Número aleatório que simula uma probabilidade															
0.55	0.61	0.91	0.11	0.56	0.90	0.32	0.12	0.32	0.72	0.19	0.77	0.81	0.32	0.45	0.73
< 0.7	< 0.7	> 0.7	< 0.7	< 0.7	> 0.7	< 0.7	< 0.7	< 0.7	> 0.7	< 0.7	> 0.7	> 0.7	< 0.7	< 0.7	> 0.7
Descendente criado após a recombinação															
0.64	0.85	0.31	0.45	0.46	0.57	0.82	0.98	0.52	0.03	0.78	0.38	0.71	0.96	0.16	0.33

Fonte: Próprio autor.

5.7.2 Mutação

Este operador presente nos algoritmos genéticos, que visa prevenir a convergência prematura da população, como no processo de recombinação no BRKGA prossegue de maneira particular. Definido o número de elementos mutantes para cada população no início do processo, procede-se a criar novos indivíduos, seguindo a mesma metodologia que foi usada para gerar a população inicial, neste caso, formada por chaves aleatórias com um valor entre 0 e 1.

Figura 25 – Processo de mutação no sistema de 14 barras.



Fonte: Próprio autor.

Esses mutantes passam diretamente a fazer parte da nova geração. Com isso em cada nova geração, adicionamos novas configurações radiais que não compartilham material genético com os membros da população anterior. Uma vez que a nova população é gerada, o processo retorna à decodificação dos indivíduos desta nova população até atingir o critério de parada. Na Figura 25 são mostradas as soluções mutantes que são inseridas na nova população.

5.8 MODIFICAÇÃO DA POPULAÇÃO INICIAL

Como mencionado na seção 4.2.4.4, a população inicial pode ser composta de soluções de boa qualidade em vez de ser simplesmente composta de vetores-de chaves aleatórias. Assim, procuraremos formar uma população inicial com indivíduos de boa qualidade usando o algoritmo

Prim e executando um programa de fluxo de carga para sistemas com fracamente malhados (SHIRMOHAMMADI et al., 1988). A metodologia utilizada é descrita embaixo:

- Execute um programa de fluxo de carga para sistemas fracamente malhados, obtendo as potências aparentes em cada um dos ramos.
- Como é necessário que os valores de cada uma das chaves que formam um cromossomo tenham um valor entre 0 e 1, cada uma das potências aparentes é dividida pelo maior valor de potência aparente encontrado.
- Em seguida, fazer uma perturbação entre 0 e 10% do valor de cada uma das chaves.

Com isto é conseguido ter uma população inicial com cromossomos de boa qualidade.

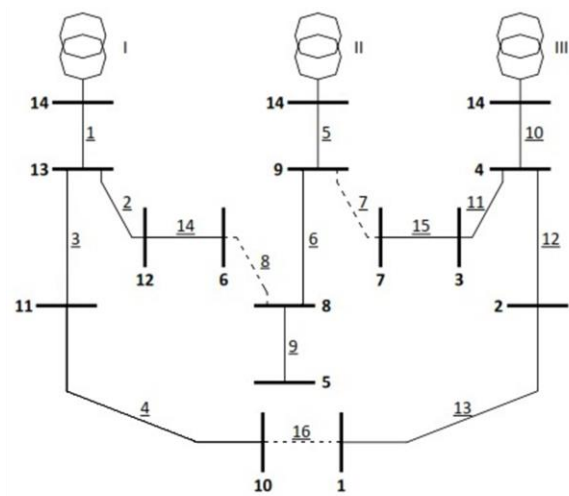
6 RESULTADOS

Nesta seção são apresentados os resultados obtidos a partir da aplicação do algoritmo genético de chaves aleatórias viciadas para a otimização do problema RSDEE com o objetivo de reduzir as perdas. A metodologia proposta foi escrita na linguagem FORTRAN e as simulações foram realizadas em um laptop com processador Intel Core i7 2,4 GHz, 8 GB de RAM. Para as simulações, foram utilizados os seguintes sistemas: 14 barras, 33 barras, 84 barras, 136 barras. Os dados correspondentes a cada sistema podem ser encontrados no Anexo desta dissertação.

Como procura-se reduzir as perdas, é esperado que as correntes nos circuitos mais carregados diminuam com a reconfiguração do sistema. Assim, tipicamente não são observadas violações na capacidade de corrente dos circuitos na solução final; na prática o processo de otimização pode ser executado sem considerar explicitamente os limites de corrente. Da mesma forma, as capacidades das subestações também foram desconsideradas nos sistemas testados; desta forma, todas as subestações podem ser interligadas, formando uma única barra e simplificando a representação do sistema (POSSAGNOLO, 2015). Assim também considerasse um limite inferior de tensão em as barras igual a 0.93 pu e um limite máximo igual a 1.05 pu.

6.1 SISTEMA DE 14 BARRAS

Figura 26 – Configuração ótima do sistema teste de 14 barras.



Fonte: Próprio autor.

O sistema teste de 14 barras foi proposto por Civanlar et al. (1998) e tem 1 barra de subestação, 13 barras de carga e 16 ramos que conectam as 14 barras, das quais 13 normalmente fechadas e 3 abertas.

A tensão na subestação é de 23 kV e a potência de base é de 100 MVA. A Figura 26 mostra a configuração ótima do sistema, os ramos abertos nesta configuração são os ramos 7, 8 e 16. A Tabela 2 mostra as perdas antes e depois da reconfiguração.

Tabela 2 – Resultados para o sistema teste de 14 barras.

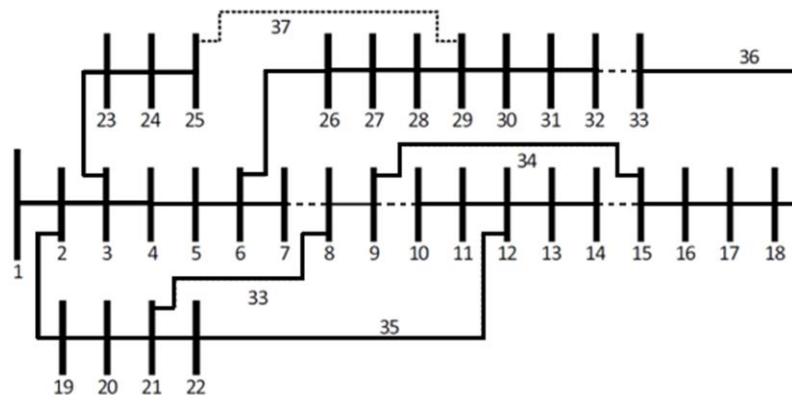
Configuração	Chaves abertas	Perdas [kW]
Inicial	14, 15, 16	511,430
BRKGA	7, 8, 16	466,127

Fonte: Próprio autor.

6.2 SISTEMA DE 33 BARRAS

Na Figura 27, a configuração ótima do sistema é mostrada. A Tabela 3 mostra as perdas antes, depois da aplicação do algoritmo BRKGA e uma comparação com o resultado de um trabalho da literatura especializada.

Figura 27 – Configuração ótima do sistema teste de 33 barras.



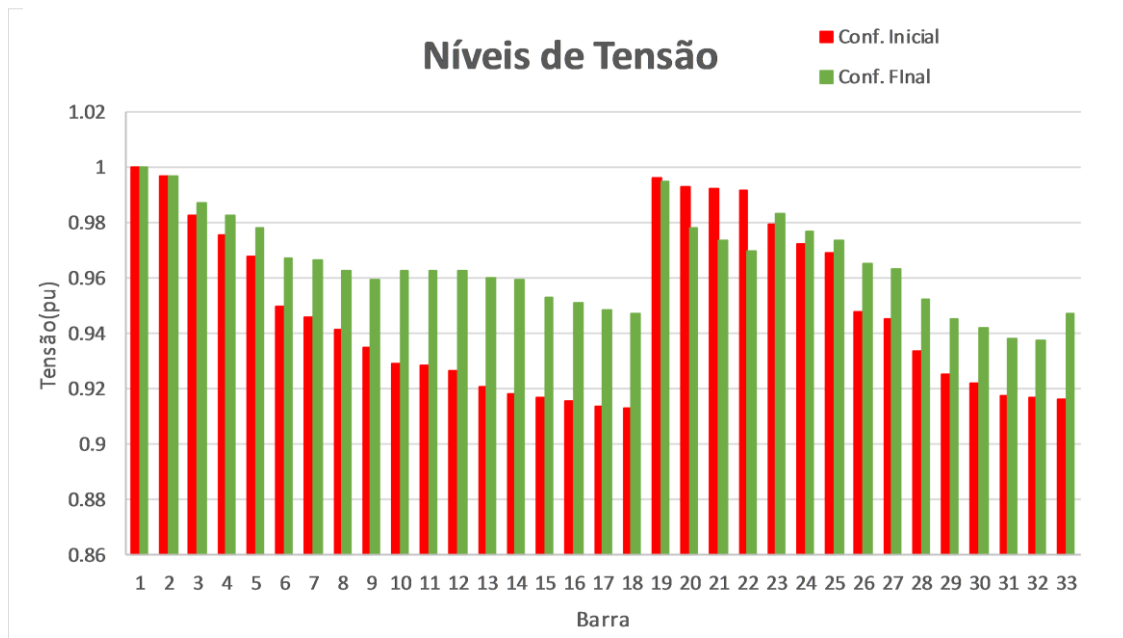
Fonte: Baran e Wu (1989).

Na figura 28 pode-se apreciar a evolução da melhor solução através das iterações. A Figura 29 mostra os níveis de tensão para o sistema de 33 barras antes e depois da reconfiguração.

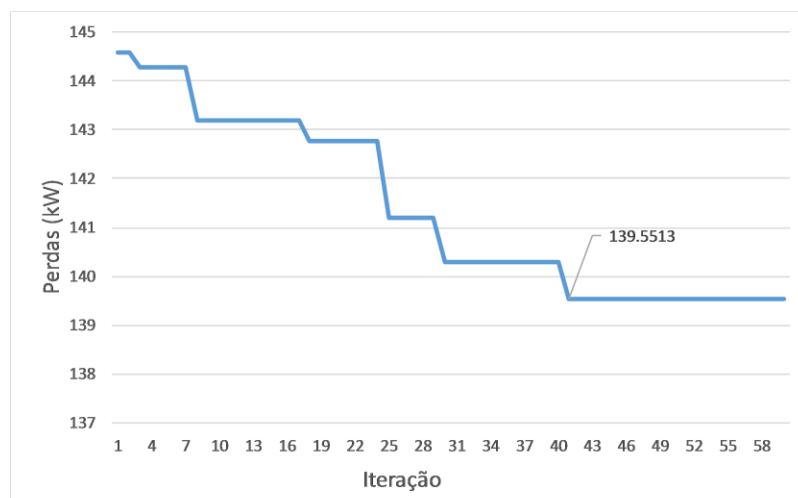
Tabela 3 – Resultados para o sistema teste de 33 barras.

Configuração	Chaves abertas	Perdas [kW]
Inicial	33, 34, 35, 36, 37	202, 67
BRKGA	7, 9, 14, 32, 37	139,55
FRANCO et al.,2012	7, 9, 14, 32, 37	139,55

Fonte: Próprio autor.

Figura 28 – Níveis de tensão antes e depois da reconfiguração no sistema de 33 barras.

Fonte: Próprio autor.

Figura 29 – Evolução do BRKGA para o sistema de 33 barras.

Fonte: Próprio autor.

6.3 SISTEMA DE 84 BARRAS

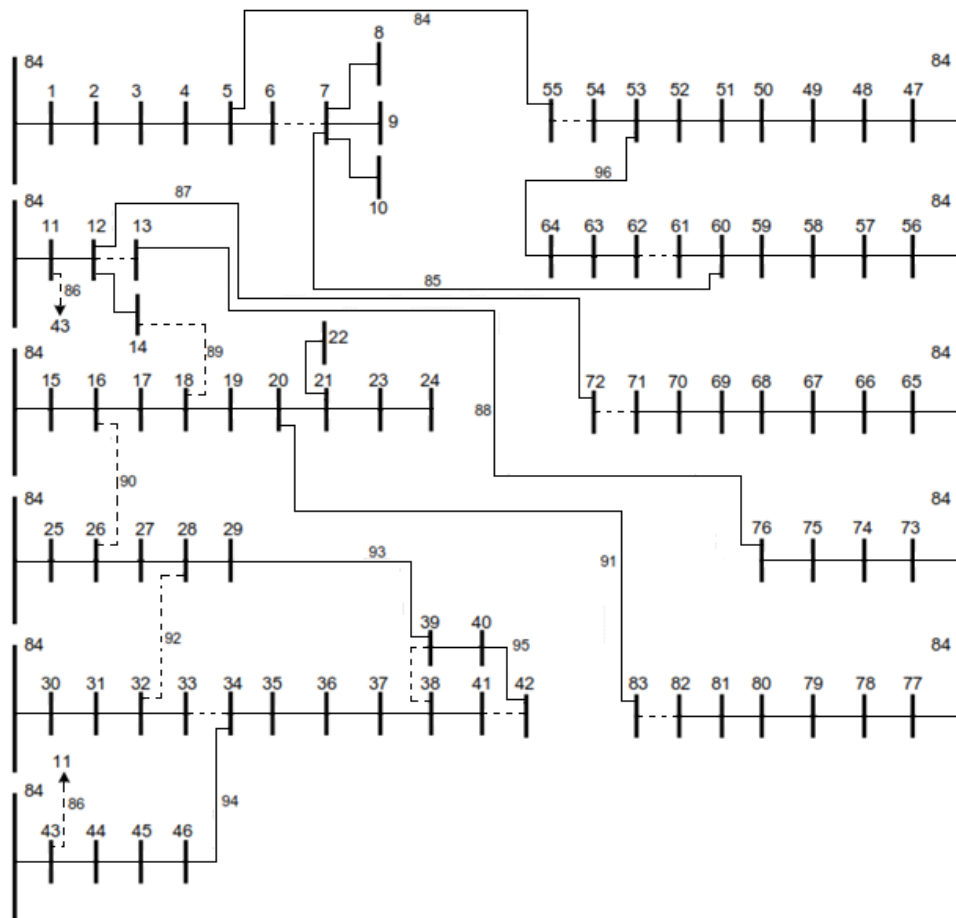
O sistema teste de 84 barras foi proposto por Chiou et al. (2005) e tem 1 barra de subestação, 83 barras de carga e 96 ramos que conectam as 84 barras, das quais 83 são normalmente fechadas e 13 abertas. A tensão na subestação é de 11,40 kV e a potência da base é de 10000 kVA.

Tabela 4 – Resultados para o sistema teste de 84 barras.

Configuração	Chaves abertas	Perdas [kW]
Inicial	84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	531,99
BRKGA	7, 13, 34, 39, 42, 55, 62, 72, 83, 86, 89, 90, 92	469,88
FRANCO et al., 2012	7, 13, 34, 39, 42, 55, 62, 72, 83, 86, 89, 90, 92	469,88

Fonte: Próprio autor.

Figura 30 – Configuração ótima do sistema teste de 84 barras.

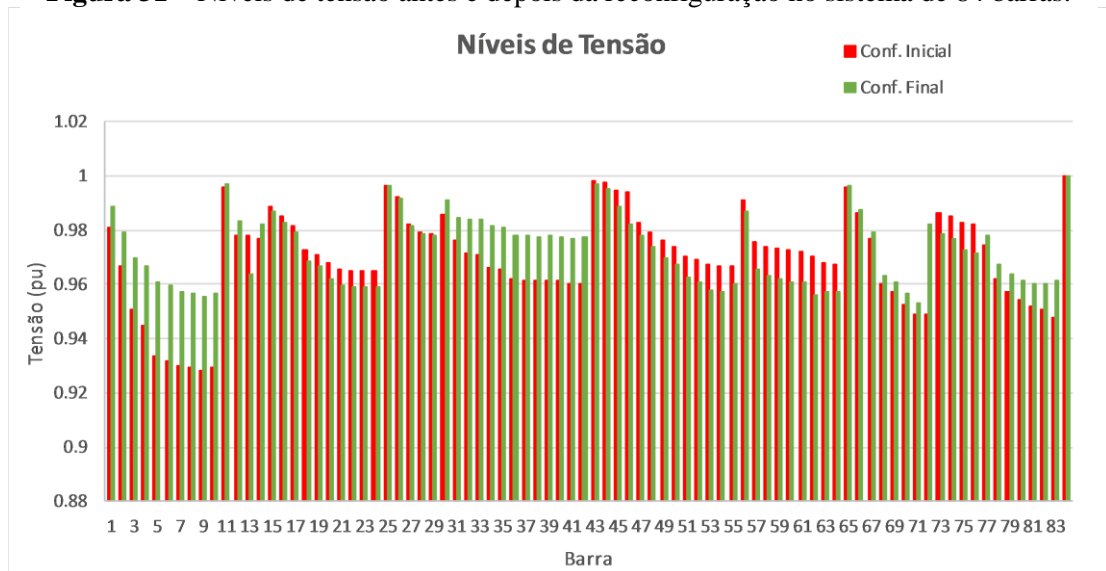


Fonte: Chiou et al. (2005).

A Tabela 4 mostra as perdas antes, depois da aplicação do algoritmo BRKGA e uma comparação com o resultado com um trabalho da literatura especializada. Na Figura 30 é mostrada a configuração ótima do sistema. Os ramos abertos nesta configuração são 7, 13, 34, 39, 42, 55, 62, 72, 83, 86, 89, 90 e 92.

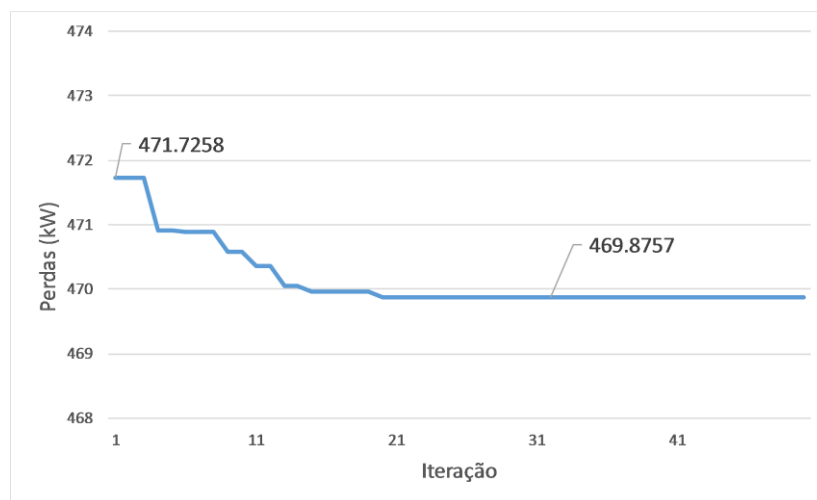
A Figura 31 mostra os níveis de tensão das barras do sistema de 84 barras enquanto que a Figura 32 apresenta evolução da melhor solução através das iterações.

Figura 31 – Níveis de tensão antes e depois da reconfiguração no sistema de 84 barras.



Fonte: Próprio autor.

Figura 32 – Evolução do BRKGA para o sistema de 84 barras.



Fonte: Próprio autor.

6.4 SISTEMA DE 136 BARRAS

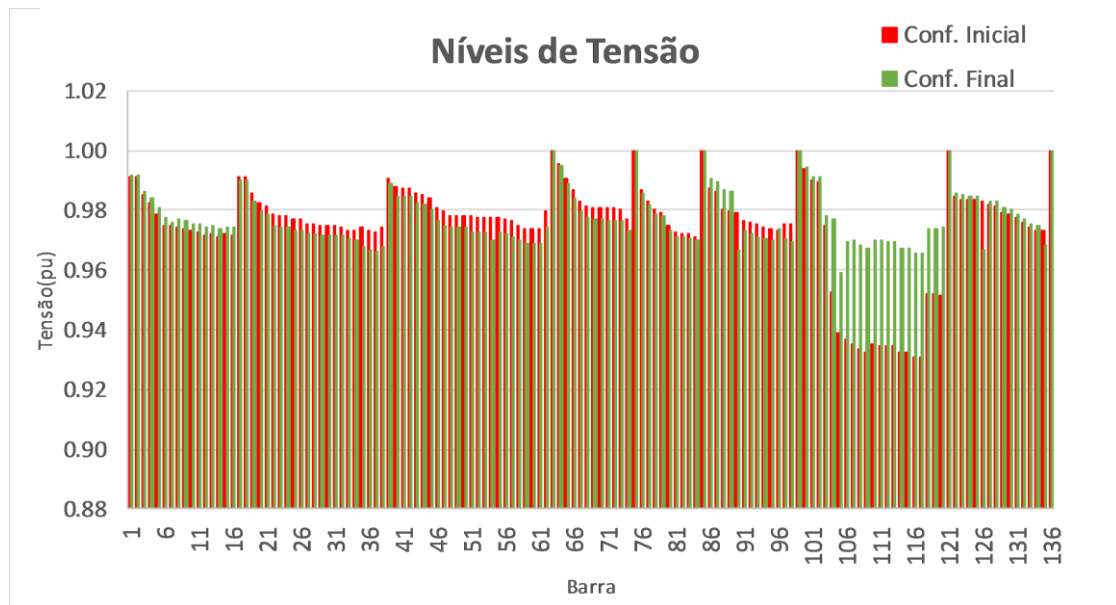
O sistema teste de 136 barras foi proposto por Mantovani et al. (2000) e tem 2 barras de subestação, 135 barras de carga e 156 ramos que conectam as 136 barras, das quais 135 são normalmente fechadas e 21 abertas. A tensão nas subestações é de 13,80 kV e a potência de base é de 100 000 kVA. Na configuração inicial do sistema os ramos abertos são os ramos 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156.

Tabela 5 – Resultados para o sistema teste de 136 barras.

Configuração	Chaves abertas	Perdas [kW]
Inicial	136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156	320,36
BRKGA	7, 35, 51, 90, 96, 106, 118, 126, 135, 137, 138, 141, 142, 144, 145, 146, 147, 148, 150, 151, 155	280,19
FRANCO et al.,2012	7, 35, 51, 90, 96, 106, 118, 126, 135, 137, 138, 141, 142, 144, 145, 146, 147, 148, 150, 151, 155	280,19

Fonte: Próprio autor.

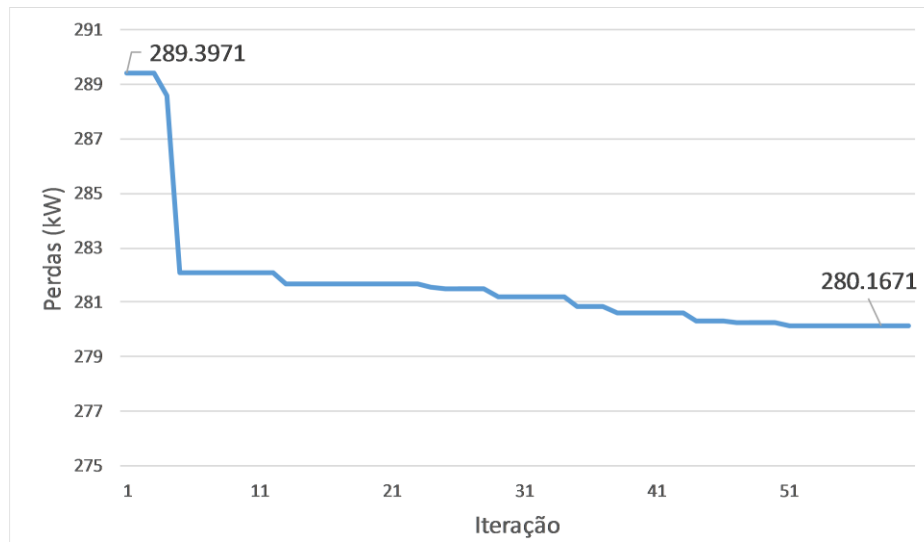
Figura 33 – Níveis de tensão antes e depois da reconfiguração no sistema de 136 barras.



Fonte: Próprio autor.

A Tabela 5 mostra as perdas antes, depois da aplicação do algoritmo BRKGA e uma comparação com o resultado com um trabalho da literatura especializada. Na Figura 33 é mostrado os níveis de tensão das barras do sistema e na Figura 34 mostra a evolução da melhor solução através das iterações.

Figura 34 – Evolução do BRKGA para o sistema de 136 barras.



Fonte: Próprio autor.

6.5 RESUMO DO RESULTADOS

Nesta secção se apresenta um resumo dos resultados obtidos da aplicação do BRKGA, apresentasse também o número de iterações feitas e o número de problemas de fluxo de carga resolvidos.

A Tabela 6 mostra os parâmetros escolhidos para os diferentes sistemas teste.

A tabela 7 mostra o número de iterações feitas e o número de problemas de fluxo de carga resolvidos até atingir o critério de parada. A Tabela 8 mostra as perdas e as chaves abertas pertencentes às configurações encontradas após a aplicação do algoritmo BRKGA.

Tabela 6 – Parâmetros escolhidos no algoritmo BRKGA para os sistemas teste.

Sistema	Tamanho da população (p)	Tamanho da população de elite (p_e)	Tamanho da população mutante (p_m)	Probabilidade de herança do alelo elite (ρ_e)
14 Barras	20	4	4	0,7
33 Barras	40	8	8	0,7
84 Barras	80	15	15	0,7
136 Barras	200	20	20	0,7

Fonte: Próprio autor.

Tabela 7 – Número de iterações feitas e número de problemas de fluxo de carga resolvidos.

Sistema	Número de Iterações	Número de FC resolvidos
14 Barras	10	20 + 10(16)
33 Barras	45	40 + 45(32)
84 Barras	50	80 + 50(65)
136 Barras	60	200 + 60(180)

Fonte: Próprio autor.

Tabela 8 – Resultados obtidos depois da aplicação do algoritmo BRKGA.

Sistema	Chaves abertas	Perdas [kW]	Tempo [s]
14 Barras	7, 8, 16	466,10	< 0,001
33 Barras	7, 9, 14, 32, 37	139,55	< 0,001
84 Barras	7, 13, 34, 39, 42, 55, 62, 72, 83, 86, 89, 90,92	469,87	1,5
136 Barras	7, 35, 51, 90, 96, 106, 118, 126, 135, 137, 138, 141, 142, 144, 145, 146, 147, 148, 150, 151, 155	280,19	16,5

Fonte: Próprio autor.

7 CONCLUSÕES

Neste trabalho, o algoritmo genético das chaves aleatórias viciadas foi proposto para resolver o problema da reconfiguração de sistemas de distribuição elétrica com o objetivo de minimizar as perdas em um sistema de distribuição.

Devido às características da meta-heurísticas, esta dissertação considera uma forma alternativa de codificação das propostas de solução para otimizar o problema da reconfiguração. Esta codificação consiste em um vetor com um comprimento igual ao número total de circuitos, em que cada elemento do vetor é uma chave aleatória entre 0 e 1, e cada chave representa um peso para cada um dos circuitos do sistema.

Dentro da estrutura do algoritmo genético utilizado, é proposto o uso do algoritmo de Prim como um nexos (decodificador) entre os vetores de chaves aleatórias e as soluções no espaço de busca do problema. O algoritmo de Prim é utilizado na geração de árvores e permitiu que todos os vetores de chaves aleatórias representassem uma configuração radial, com isso é possível reduzir o espaço de busca, pois só se trabalha com configurações exclusivamente radiais.

O algoritmo proposto permite encontrar soluções de boa qualidade em todos os sistemas testados. Essas soluções são comparáveis às soluções encontradas na literatura especializada. Além disso, essas soluções são avaliadas pelo algoritmo de fluxo de carga de varredura e mostram que elas cumprem as restrições de operação.

REFERÊNCIAS

- BÄCK, T.; FOGEL, D.; MICHALEWICZ, Z. **Evolutionary computation 2**. Boca Ratón: CRC, 2000. 308 p.
- BARAN, M. E.; WU, F. F. Network reconfiguration in distribution systems for loss reduction and load balancing. **IEEE Transactions on Power Delivery**, New York, v. 4, n. 2, p. 1401–1407, abr. 1989.
- BEAN, J. C. Genetic algorithms and random keys for sequencing and optimization. **ORSA Journal on Computing**, Baltimore, v. 6, n. 2, p. 154–160, maio 1994.
- BORGHETTI, A. A Mixed-integer linear programming approach for the computation of the minimum-losses radial configuration of electrical distribution networks. **IEEE Transactions on Power Systems**, New York, v. 27, n. 3, p. 1264–1273, ago. 2012.
- CARRENO, E. M.; ROMERO, R.; PADILHA-FELTRIN, A. An efficient codification to solve distribution network reconfiguration for loss reduction problem. **IEEE Transactions on Power Systems**, New York, v. 23, n. 4, p. 1542–1551, nov. 2008.
- CESPEDES, R. G. New method for the analysis of distribution networks. **IEEE Transactions on Power Delivery**, New York, v. 5, n. 1, p. 391–396, 1990.
- CIVANLAR, S. et al. Distribution feeder reconfiguration for loss reduction. **IEEE Transactions on Power Delivery**, New York, v. 3, n. 3, p. 1217–1223, jul. 1988.
- DE JONG, K. A. Genetic algorithms are NOT function optimizers. In: L. Darrell Whitley. **Foundations of genetic algorithms 2**. San Mateo: Morgan Kaufmann Publishers, Inc, 1993. v. 2, p. 5–17.
- FRANCO, J.; LAVORATO, M.; RIDER, M. J.; ROMERO, R. An efficient implementation of tabu search in feeder reconfiguration of distribution systems. In: IEEE POWER AND ENERGY SOCIETY GENERAL MEETING, 15., 2012, San Diego. **Proceedings...** San Diego: PESGM, 2012. p. 1-8.
- FRANCO, J. F.; RIDER, M. J.; LAVORATO, M.; ROMERO, R. A mixed-integer LP model for

the reconfiguration of radial electric distribution systems considering distributed generation. **Electric Power Systems Research**, Lausanne, v. 97, n. 1, p. 51–60, 2013.

GENDREAU, M.; POTVIN, J.-Y. **Handbook of metaheuristics**. Boston: Springer Publishing Company, 2010. 648 p.

GLAMOCANIN, V. Optimal loss reduction of distributed networks. **IEEE Transactions on Power Systems**, New York, v. 5, n. 3, p. 774–782, 1990.

GLOVER, F.; KOCHENBERGER, G. A. **Handbook of metaheuristics**. Boston: Kluwer Academic Publishers, 2003. 570 p.

GOMES, F. V.; CARNEIRO JÚNIOR, S.; PEREIRA, J. L. R.; VINAGRE, M. P.; GARCIA, P. A. N.; ARAUJO, L. R. A New distribution system reconfiguration approach using optimum power flow and sensitivity analysis for loss reduction. **IEEE Transactions on Power Systems**, New York, v. 21, n. 4, p. 1616–1623, nov. 2006.

GONÇALVES, J. F.; RESENDE, M. G. C. Biased random-key genetic algorithms for combinatorial optimization. **Journal of Heuristics**, New York, v. 17, n. 5, p. 487–525, 27 out. 2011.

GOSWAMI, S. K.; BASU, S. K. A new algorithm for the reconfiguration of distribution feeders for loss minimization. **IEEE Transactions on Power Delivery**, New York, v. 7, n. 3, p. 1484–1491, jul. 1992.

HOLLAND, J. H. **Adaptation in natural and artificial systems**. Ann Arbor: University of Michigan Press, 1975. 228 p.

JABR, R. A.; SINGH, R.; PAL, B. C. Minimum loss network reconfiguration using mixed-integer convex programming. **IEEE Transactions on Power Systems**, New York, v. 27, n. 2, p. 1106–1115, maio 2012.

LAVORATO, M.; FRANCO, J.F.; RIDER, M. J.; ROMERO, R. Imposing radiality constraints in distribution system optimization problems. **IEEE Transactions on Power Systems**, New York, v. 27, n. 1, p. 172–180, fev. 2012.

LEHMAN, E.; THOMSON, L.; MEYER, A. **Mathematics for computer science**. Massachusetts: MIT, 2017. 998 p

MENDOZA, J.; LÓPEZ, R.; MORALES, D.; LÓPEZ, E.; DESSANTE, P.; MORAGA, R. Minimal loss reconfiguration using genetic algorithms with restricted population and addressed operators: real application. **IEEE Transactions on Power Systems**, New York, v. 21, n. 2, p. 948–954, maio 2006.

MERLIN, A.; BACK, H. Search for minimum-loss operating spanning tree configuration for an urban power distribution system. In: POWER SYSTEM COMPUTATION CONFERENCE-PSCC, 5., 1975, Cambridge. **Proceedings...** Zurich: [s.n.], 1975. p. 1-18

MICHALEWICZ, Z. **Genetic algorithms + data structures: evolution programs**. Berlin: Springer Berlin Heidelberg, 1996. 387 p.

MONTICELLI, A. J. **Fluxo de carga em redes de energia elétrica**. São Paulo: Edgard Blücher, 1983. 164 p.

NARA, K.; SHIOSE, A.; KITAGAWA, M.; ISHIHARA, T. Implementation of genetic algorithm for distribution systems loss minimum re-configuration. **IEEE Transactions on Power Systems**, New York, v. 7, n. 3, p. 1044–1051, 1992.

POSSAGNOLO, L. H. F. M. **Reconfiguração de sistemas de distribuição operando em vários níveis de demanda através de uma meta-heurística de busca em vizinhança variável**. 2015. 184 f. Dissertação (Mestrado em Engenharia Elétrica)- Faculdade de Engenharia, Universidade Estadual Paulista Julio de Mesquita Filho- UNESP, Ilha Solteira, 2015.

SHIRMOHAMMADI, D. et al. A compensation-based power flow method for weakly meshed distribution and transmission networks. **IEEE Transactions on Power Systems**, New York, v. 3, n. 2, p. 753–762, maio 1988.

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

SOUZA, S. S. F.; ROMERO, R.; FRANCO, J. F. Artificial immune networks copt-aiNet and opt-aiNet applied to the reconfiguration problem of radial electrical distribution systems. **Electric**

Power Systems Research, Lausanne, v. 119, p. 304-312, 2015.

ZHANG, D.; FU, Z.; ZHANG, L. An improved TS algorithm for loss-minimum reconfiguration in large-scale distribution systems. **Electric Power Systems Research**, Lausanne, v. 77, n. 5-6, p. 685-694, abr. 2007.

ZHU, J. Optimal reconfiguration of electrical distribution network using the refined genetic algorithm. **Electric Power Systems Research**, Lausanne, v. 62, n. 1, p. 37-42, 2002.

ANEXO A - DADOS DOS SISTEMAS DE DISTRIBUIÇÃO

Este anexo apresenta os dados dos sistemas de distribuição utilizados para a verificação da funcionalidade do algoritmo BRKGA proposto nesta pesquisa.

A.1 Sistema de 14 barras

- Barra de subestação: 14
- Tensão base: 23 kV
- Potencia base: 100000 kVA

Tabela 9 – Dados do sistema de 14 barras

No. do ramos	Barra inicial	Barra inicial	Resistência [%]	Reatância [%]	No. da barra	Demanda ativa [kW]	Demanda reativa [kVAr]
1	14	13	7,50	10,00	1	2100,00	-800,00
2	13	12	8,00	11,00	2	1000,00	900,00
3	13	11	9,00	18,00	3	1000,00	-1100,00
4	11	10	4,00	4,00	4	1000,00	900,00
5	14	9	11,00	11,00	5	4500,00	-1700,00
6	9	8	8,00	11,00	6	600,00	-500,00
7	9	7	11,00	11,00	7	1000,00	900,00
8	8	6	11,00	11,00	8	5000,00	1800,00
9	8	5	8,00	11,00	9	4000,00	2700,00
10	14	4	11,00	11,00	10	1500,00	1200,00
11	4	3	9,00	12,00	11	2000,00	-400,00
12	4	2	8,00	11,00	12	3000,00	400,00
13	2	1	4,00	4,00	13	2000,00	1600,00
14	12	6	4,00	4,00	14	0,00	0,00
15	7	3	4,00	4,00			
16	10	1	9,00	12,00			

A.2 Sistema de 33 barras

- Barra de subestação: 1
- Tensão base: 12,66 kV
- Potencia base: 1000 kVA

Tabela 10– Dados do sistema de 33 barras

No. do ramos	Barra inicial	Barra inicial	Resistência [Ω]	Reatância [Ω]	No. da barra	Demanda ativa [kW]	Demanda reativa [kVAr]
1	1	2	0,0922	0,0470	1	0,00	0,00
2	2	3	0,4930	0,2511	2	100,00	60,00

3	3	4	0,3660	0,1864	3	90,00	40,00
4	4	5	0,3811	0,1941	4	120,00	80,00
5	5	6	0,8190	0,7070	5	60,00	30,00
6	6	7	0,1872	0,6188	6	60,00	20,00
7	7	8	0,7114	0,2351	7	200,00	100,00
8	8	9	1,0300	0,7400	8	200,00	100,00
9	9	10	1,0440	0,7400	9	60,00	20,00
10	10	11	0,1966	0,0650	10	60,00	20,00
11	11	12	0,3744	0,1238	11	45,00	30,00
12	12	13	1,4680	1,1550	12	60,00	35,00
13	13	14	0,5416	0,7129	13	60,00	35,00
14	14	15	0,5910	0,5260	14	120,00	80,00
15	15	16	0,7463	0,5450	15	60,00	10,00
16	16	17	1,2890	1,7210	16	60,00	20,00
17	17	18	0,7320	0,5740	17	60,00	20,00
18	2	19	0,1640	0,1565	18	90,00	40,00
19	19	20	1,5042	1,3554	19	90,00	40,00
20	20	21	0,4095	0,4784	20	90,00	40,00
21	21	22	0,7089	0,9373	21	90,00	40,00
22	3	23	0,4512	0,3083	22	90,00	40,00
23	23	24	0,8980	0,7091	23	90,00	50,00
24	24	25	0,8960	0,7011	24	420,00	200,00
25	6	26	0,2030	0,1034	25	420,00	200,00
26	26	27	0,2842	0,1447	26	60,00	25,00
27	27	28	1,0590	0,9337	27	60,00	25,00
28	28	29	0,8042	0,7006	28	60,00	20,00
29	29	30	0,5075	0,2585	29	120,00	70,00
30	30	31	0,9744	0,9630	30	200,00	600,00
31	31	32	0,3105	0,3619	31	150,00	70,00
32	32	33	0,3410	0,5302	32	210,00	100,00
33	8	21	2,0000	2,0000	33	60,00	40,00
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			

A.3 Sistema de 84 barras

- Barra de subestação: 84
- Tensão base: 11,4 kV
- Potencia base: 10000 kVA

Tabela 11 – Dados do sistema de 84 barras

No. do ramos	Barra inicial	Barra inicial	Resistência [Ω]	Reatância [Ω]	No. da barra	Demanda ativa [kW]	Demanda reativa [kVAr]
1	1	84	0,1944	0,6624	1	0,00	0,00
2	1	2	0,2096	0,4304	2	100,00	50,00
3	2	3	0,2358	0,4842	3	300,00	200,00
4	3	4	0,0917	0,1883	4	350,00	250,00
5	4	5	0,2096	0,4304	5	220,00	100,00
6	5	6	0,0393	0,0807	6	1100,00	800,00

7	6	7	0,0405	0,1380	7	400,00	320,00
8	7	8	0,1048	0,2152	8	300,00	200,00
9	7	9	0,2358	0,4842	9	300,00	230,00
10	7	10	0,1048	0,2152	10	300,00	260,00
11	11	84	0,0786	0,1614	11	0,90	0,00
12	11	12	0,3406	0,6944	12	1200,00	800,00
13	12	13	0,0262	0,0538	13	800,00	600,00
14	12	14	0,0786	0,1614	14	700,00	500,00
15	15	84	0,1134	0,3864	15	0,00	0,00
16	15	16	0,0524	0,1076	16	300,00	150,00
17	16	17	0,0524	0,1076	17	500,00	350,00
18	17	18	0,1572	0,3228	18	700,00	400,00
19	18	19	0,0393	0,0807	19	1200,00	1000,00
20	19	20	0,1703	0,3497	20	300,00	300,00
21	20	21	0,2358	0,4842	21	400,00	350,00
22	21	22	0,1572	0,3228	22	50,00	20,00
23	21	23	0,1965	0,4035	23	50,00	20,00
24	23	24	0,1310	0,2690	24	50,00	10,00
25	25	84	0,0567	0,1932	25	50,00	30,00
26	25	26	0,1048	0,2152	26	100,00	60,00
27	26	27	0,2489	0,5111	27	100,00	70,00
28	27	28	0,0486	0,1656	28	1800,00	1300,00
29	28	29	0,1310	0,2690	29	200,00	120,00
30	30	84	0,1965	0,3960	30	0,00	0,00
31	30	31	0,1310	0,2690	31	1800,00	1600,00
32	31	32	0,1310	0,2690	32	200,00	150,00
33	32	33	0,0262	0,0538	33	200,00	100,00
34	33	34	0,1703	0,3497	34	800,00	600,00
35	34	35	0,0524	0,1076	35	100,00	60,00
36	35	36	0,4978	1,0222	36	100,00	60,00
37	36	37	0,0393	0,0807	37	20,00	10,00
38	37	38	0,0393	0,0807	38	20,00	10,00
39	38	39	0,0786	0,1614	39	20,00	10,00
40	39	40	0,2096	0,4304	40	20,00	10,00
41	38	41	0,1965	0,4035	41	200,00	160,00
42	41	42	0,2096	0,4304	42	50,00	30,00
43	43	84	0,0486	0,1656	43	0,00	0,00
44	43	44	0,0393	0,0807	44	30,00	20,00
45	44	45	0,1310	0,2690	45	800,00	700,00
46	45	46	0,2358	0,4842	46	200,00	150,00
47	47	84	0,2430	0,8280	47	0,00	0,00
48	47	48	0,0655	0,1345	48	0,00	0,00
49	48	49	0,0655	0,1345	49	0,00	0,00
50	49	50	0,0393	0,0807	50	200,00	160,00
51	50	51	0,0786	0,1614	51	800,00	600,00
52	51	52	0,0393	0,0807	52	500,00	300,00
53	52	53	0,0786	0,1614	53	500,00	350,00
54	53	54	0,0524	0,1076	54	500,00	300,00
55	54	55	0,1310	0,2690	55	200,00	80,00
56	56	84	0,2268	0,7728	56	0,00	0,00
57	56	57	0,5371	1,1029	57	30,00	20,00
58	57	58	0,0524	0,1076	58	600,00	420,00
59	58	59	0,0405	0,1380	59	0,00	0,00
60	59	60	0,0393	0,0807	60	20,00	10,00
61	60	61	0,0262	0,0538	61	20,00	10,00
62	61	62	0,1048	0,2152	62	200,00	130,00

63	62	63	0,2358	0,4842	63	300,00	240,00
64	63	64	0,0243	0,0828	64	300,00	200,00
65	65	84	0,0486	0,1656	65	0,00	0,00
66	65	66	0,1703	0,3497	66	50,00	30,00
67	66	67	0,1215	0,4140	67	0,00	0,00
68	67	68	0,2187	0,7452	68	400,00	360,00
69	68	69	0,0486	0,1656	69	0,00	0,00
70	69	70	0,0729	0,2484	70	0,00	0,00
71	70	71	0,0567	0,1932	71	2000,00	1500,00
72	71	72	0,0262	0,0528	72	200,00	150,00
73	73	84	0,3240	1,1040	73	0,00	0,00
74	73	74	0,0324	0,1104	74	0,00	0,00
75	74	75	0,0567	0,1932	75	1200,00	950,00
76	75	76	0,0486	0,1656	76	300,00	180,00
77	77	84	0,2511	0,8556	77	0,00	0,00
78	77	78	0,1296	0,4416	78	400,00	360,00
79	78	79	0,0486	0,1656	79	2000,00	1300,00
80	79	80	0,1310	0,2640	80	200,00	140,00
81	80	81	0,1310	0,2640	81	500,00	360,00
82	81	82	0,0917	0,1883	82	100,00	30,00
83	82	83	0,3144	0,6456	83	400,00	360,00
84	5	55	0,1310	0,2690	84	0,00	0,00
85	7	60	0,1310	0,2690			
86	11	43	0,1310	0,2690			
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			

A.4 Sistema de 136 barras

- Barra de subestação: 136
- Tensão base: 13,8 kV
- Potencia base: 100000 kVA

Tabela 12 – Dados do sistema de 136 barras

No. do ramos	Barra inicial	Barra inicial	Resistência [Ω]	Reatância [Ω]	No. da barra	Demanda ativa [kW]	Demanda reativa [kVAr]
1	136	1	0,33205	0,76653	1	0,000	0,000
2	1	2	0,00188	0,00433	2	47,780	19,009
3	2	3	0,22324	0,51535	3	42,551	16,929
4	3	4	0,09943	0,22953	4	87,022	34,622
5	4	5	0,15571	0,35945	5	311,310	123,855
6	5	6	0,16321	0,37677	6	148,869	59,228

7	6	7	0,11444	0,26417	7	238,672	94,956
8	6	8	0,05675	0,05666	8	62,299	24,786
9	8	9	0,52124	0,27418	9	124,598	49,571
10	8	10	0,10877	0,10860	10	140,175	55,768
11	10	11	0,39803	0,20937	11	116,813	46,474
12	10	12	0,91744	0,31469	12	249,203	99,145
13	10	13	0,11823	0,11805	13	291,447	115,952
14	13	14	0,50228	0,26421	14	303,720	120,835
15	13	15	0,05675	0,05666	15	215,396	85,695
16	15	16	0,29379	0,15454	16	198,586	79,007
17	136	17	0,33205	0,76653	17	0,000	0,000
18	17	18	0,00188	0,00433	18	0,000	0,000
19	18	19	0,22324	0,51535	19	0,000	0,000
20	19	20	0,10881	0,25118	20	30,127	14,729
21	20	21	0,71078	0,37388	21	230,972	112,920
22	20	22	0,18197	0,42008	22	60,256	29,458
23	22	23	0,30326	0,15952	23	230,972	112,920
24	22	24	0,02439	0,05630	24	120,507	58,915
25	24	25	0,04502	0,10394	25	0,000	0,000
26	25	26	0,01876	0,04331	26	56,981	27,857
27	26	27	0,11823	0,11805	27	364,665	178,281
28	27	28	0,02365	0,02361	28	0,000	0,000
29	28	29	0,18954	0,09970	29	124,647	60,939
30	29	30	0,39803	0,20937	30	56,981	27,857
31	28	31	0,05675	0,05666	31	0,000	0,000
32	31	32	0,09477	0,04985	32	85,473	41,787
33	32	33	0,41699	0,21934	33	0,000	0,000
34	33	34	0,11372	0,05982	34	396,735	193,960
35	31	35	0,07566	0,07555	35	0,000	0,000
36	35	36	0,36960	0,19442	36	181,152	88,563
37	36	37	0,26536	0,13958	37	242,172	118,395
38	35	38	0,05675	0,05666	38	75,316	36,821
39	136	39	0,33205	0,76653	39	0,000	0,000
40	39	40	0,11819	0,27283	40	1,254	0,531
41	40	41	2,96288	1,01628	41	6,274	2,660
42	40	42	0,00188	0,00433	42	0,000	0,000
43	42	43	0,06941	0,16024	43	117,880	49,971
44	43	44	0,81502	0,42872	44	62,668	26,566
45	43	45	0,06378	0,14724	45	172,285	73,034
46	45	46	0,13132	0,30315	46	458,556	194,388
47	46	47	0,06191	0,14291	47	262,962	111,473
48	47	48	0,11444	0,26417	48	235,761	99,942
49	48	49	0,28374	0,28331	49	0,000	0,000
50	49	50	0,28374	0,28331	50	109,215	46,298
51	48	51	0,04502	0,10394	51	0,000	0,000
52	51	52	0,02626	0,06063	52	72,809	30,865
53	52	53	0,06003	0,13858	53	258,473	109,570
54	53	54	0,03002	0,06929	54	69,169	29,322
55	54	55	0,02064	0,04764	55	21,843	9,260
56	52	56	0,10881	0,25118	56	0,000	0,000
57	56	57	0,25588	0,13460	57	20,527	8,702
58	57	58	0,41699	0,21934	58	150,548	63,819
59	58	59	0,50228	0,26421	59	220,687	93,552
60	59	60	0,33170	0,17448	60	92,384	39,163
61	60	61	0,20849	0,10967	61	0,000	0,000
62	47	62	0,13882	0,32047	62	226,693	96,098

63	136	63	0,00750	0,01732	63	0,000	0,000
64	63	64	0,27014	0,62362	64	294,016	116,974
65	64	65	0,38270	0,88346	65	83,015	33,028
66	65	66	0,33018	0,76220	66	83,015	33,028
67	66	67	0,32830	0,75787	67	103,770	41,285
68	67	68	0,17072	0,39409	68	176,408	70,184
69	68	69	0,55914	0,29412	69	83,015	33,028
70	68	70	0,05816	0,13425	70	217,917	86,698
71	70	71	0,70130	0,36890	71	23,294	9,267
72	71	72	1,02352	0,53839	72	5,075	2,019
73	70	73	0,06754	0,15591	73	72,638	28,899
74	73	74	1,32352	0,45397	74	405,990	161,524
75	136	75	0,01126	0,02598	75	0,000	0,000
76	75	76	0,72976	1,68464	76	100,182	42,468
77	76	77	0,22512	0,51968	77	142,523	60,417
78	77	78	0,20824	0,48071	78	96,042	40,713
79	78	79	0,04690	0,10827	79	300,454	127,366
80	79	80	0,61950	0,61857	80	141,238	59,873
81	80	81	0,34049	0,33998	81	279,847	118,631
82	81	82	0,56862	0,29911	82	87,312	37,013
83	81	83	0,10877	0,10860	83	243,849	103,371
84	83	84	0,56862	0,29911	84	247,750	105,025
85	136	85	0,01126	0,02598	85	0,000	0,000
86	85	86	0,41835	0,96575	86	89,878	38,101
87	86	87	0,10499	0,13641	87	1137,280	482,108
88	86	88	0,43898	1,01338	88	458,339	194,296
89	88	89	0,07520	0,02579	89	385,197	163,290
90	89	90	0,07692	0,17756	90	0,000	0,000
91	90	91	0,33205	0,76653	91	79,608	33,747
92	91	92	0,08442	0,19488	92	87,312	37,013
93	92	93	0,13320	0,30748	93	0,000	0,000
94	93	94	0,29320	0,29276	94	74,001	31,370
95	94	95	0,21753	0,21721	95	232,050	98,369
96	95	96	0,26482	0,26443	96	141,819	60,119
97	93	97	0,10318	0,23819	97	0,000	0,000
98	97	98	0,13507	0,31181	98	76,449	32,408
99	136	99	0,00938	0,02165	99	0,000	0,000
100	99	100	0,16884	0,38976	100	51,322	21,756
101	100	101	0,11819	0,27283	101	59,874	25,381
102	101	102	2,28608	0,78414	102	9,065	3,843
103	101	103	0,45587	1,05236	103	2,092	0,887
104	103	104	0,69600	1,60669	104	16,735	7,094
105	104	105	0,45774	1,05669	105	1506,522	638,634
106	105	106	0,20298	0,26373	106	313,023	132,694
107	106	107	0,21348	0,27737	107	79,831	33,842
108	107	108	0,54967	0,28914	108	51,322	21,756
109	108	109	0,54019	0,28415	109	0,000	0,000
110	107	110	0,04550	0,05911	110	202,435	85,815
111	110	111	0,47385	0,24926	111	60,823	25,784
112	111	112	0,86241	0,45364	112	45,618	19,338
113	112	113	0,56862	0,29911	113	0,000	0,000
114	108	114	0,77711	0,40878	114	157,070	66,584
115	114	115	1,08038	0,56830	115	0,000	0,000
116	109	116	1,09933	0,57827	116	250,148	106,041
117	116	117	0,47385	0,24926	117	0,000	0,000
118	104	118	0,32267	0,74488	118	69,809	29,593

119	118	119	0,14633	0,33779	119	32,072	13,596
120	119	120	0,12382	0,28583	120	61,084	25,894
121	136	121	0,01126	0,02598	121	0,000	0,000
122	121	122	0,64910	1,49842	122	94,622	46,260
123	122	123	0,04502	0,10394	123	49,858	24,375
124	123	124	0,52640	0,18056	124	123,164	60,214
125	123	125	0,02064	0,04764	125	78,350	38,304
126	125	126	0,53071	0,27917	126	145,475	71,121
127	125	127	0,09755	0,22520	127	21,369	10,447
128	127	128	0,11819	0,27283	128	74,789	36,564
129	127	129	0,13882	0,32047	129	227,926	111,431
130	129	130	0,04315	0,09961	130	35,614	17,411
131	130	131	0,09192	0,21220	131	249,295	121,877
132	131	132	0,16134	0,37244	132	316,722	154,842
133	132	133	0,37832	0,37775	133	333,817	163,199
134	133	134	0,39724	0,39664	134	249,295	121,877
135	134	135	0,29320	0,29276	135	0,000	0,000
136	7	73	0,13132	0,30315	136	0,000	0,000
137	9	24	0,26536	0,13958			
138	15	83	0,14187	0,14166			
139	38	135	0,08512	0,08499			
140	25	51	0,04502	0,10394			
141	50	96	0,14187	0,14166			
142	55	98	0,14187	0,14166			
143	62	120	0,03940	0,09094			
144	66	79	0,12944	0,29882			
145	79	131	0,01688	0,03898			
146	84	135	0,33170	0,17448			
147	91	104	0,14187	0,14166			
148	90	129	0,07692	0,17756			
149	90	103	0,07692	0,17756			
150	92	104	0,07692	0,17756			
151	92	132	0,07692	0,17756			
152	96	120	0,26482	0,26443			
153	110	47	0,49696	0,64567			
154	126	76	0,17059	0,08973			
155	128	77	0,05253	0,12126			
156	135	98	0,29320	0,29276			