



UNIVERSIDADE ESTADUAL PAULISTA

Pós-Graduação em Engenharia Elétrica

**ALGORITMO GENÉTICO USADO NO PROBLEMA DE ALOCAÇÃO
DE CAPACITORES EM SISTEMAS DE DISTRIBUIÇÃO RADIAIS**

Eliedete Pinheiro dos Santos

Ilha Solteira - SP

1210001332



UNIVERSIDADE ESTADUAL PAULISTA "JULIO DE MESQUITA FILHO"
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
STRICTU SENSU

ALGORITMO GENÉTICO USADO NO PROBLEMA DE ALOCAÇÃO DE
CAPACITORES EM SISTEMAS DE DISTRIBUIÇÃO RADIAL

Proc. 071/2002 - NAD 155/02

UNESP - "CAMPUS DE ILHA SOLTEIRA"	
SERVIÇO TÉCNICO DE BIBLIOTECA E DOCUMENTAÇÃO	
DATA DE CHEGADA	DATA DE TOMBO
<i>02.12.02</i>	<i>31.12.02</i>
RECEBIDO POR	TOMBO
<i>Ailza</i>	<i>Te. 1332</i>
AQUISIÇÃO	CLASSIFICAÇÃO
<i>Doações Anten R\$ 10,00</i>	<i>S237a</i>

1210001332



"Dissertação apresentada à Faculdade de Engenharia de Ilha Solteira (UNESP), como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica"

sup - 183668
sup - 54677

Eliedete Pinheiro dos Santos
Orientador: Rubén Augusto Romero Lázaro

Ilha Solteira, Abril de 2000

BCpIS - FEIS - UNESP

FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação/Serviço Técnico de Biblioteca e Documentação da FEIS/UNESP

S237a Santos, Eliedete Pinheiro dos
Algoritmo genético usado no problema de alocação de capacitores em sistemas de distribuição radial / Eliedete Pinheiro dos Santos. – Ilha Solteira : [s.n.], 2000
109 p.

Dissertação (mestrado) – Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2000.

Orientador: Rubén Augusto Romero Lázaro

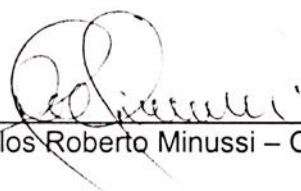
Bibliografia: p. 93-95

1. Algoritmos genéticos. 2. Capacitores. 3. Operadores genéticos.

Algoritmo Genético Usado no Problema de Alocação de Capacitores em Sistemas de Distribuição Radial

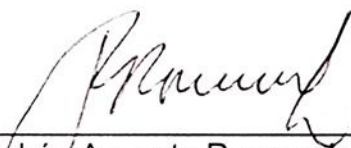
Eliedete Pinheiro dos Santos

DISSERTAÇÃO SUBMETIDA À FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA – UNESP – COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA ELÉTRICA (ME).

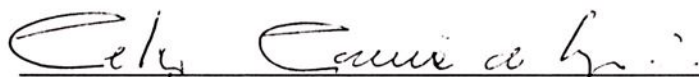


Prof. Dr. Carlos Roberto Minussi – Coordenador

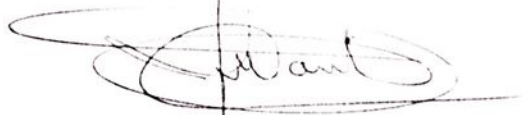
COMISSÃO EXAMINADORA:



Prof. Dr. Rubén Augusto Romero Lázaro - orientador



Prof. Dr. Celso Correia de Souza



Prof. Dr. José Roberto Sanches Mantovani

Ilha Solteira – SP, abril de 2000

SUMÁRIO

Introdução.....	08
Modelagem Matemática e Técnicas de Solução Usadas no Problema de Alocação de Capacitores.....	11
Algoritmo de Fluxo de Carga Radial.....	34
O Algoritmo Genético.....	37
O Algoritmo Genético Proposto.....	59
Testes e Resultados Obtidos.....	73
Conclusões.....	91
Referências Bibliográficas.....	93
Apêndice A.....	99

DEDICATÓRIA

Aos meus filhos, Renata, Claudinir e Douglas, pela compreensão da ausência e pelo estímulo de seguir em frente.

Aos meus pais, que fizeram de mim uma pessoa melhor.

A Deus, com sua infinita generosidade, persisti em me apresentar oportunidades.



AGRADECIMENTOS ESPECIAIS

Ao Prof. Rubén Augusto Romero Lázaro, pela orientação segura e competente e pela especial compreensão.

Aos professores da Pós Graduação da UNESP de Ilha Solteira, em especial ao professor Minussi, pela contribuição.

Aos servidores do Departamento de Engenharia Elétrica, pela atenção e pelos serviços prestados.

Aos amigos especiais: Wendel Cleber, Marcos Duarte, Sonia Angelina, Jair Longo, Alessandro Basso, José Viana e à todos que contribuíram para a realização deste trabalho.

À UFMS, pelo apoio.



ÍNDICE

1. INTRODUÇÃO.....	8
2. MODELAGEM MATEMÁTICA E TÉCNICAS DE SOLUÇÃO USADAS NO PROBLEMA DE ALOCAÇÃO DE CAPACITORES.....	11
2.1. Formulação do Problema de Reativos.....	11
2.2. Métodos de Otimização Clássica.....	12
2.2.1. Formulação do Problema.....	12
2.2.2. Curva de Duração de Carga.....	13
2.2.3. Fluxo de Carga em Sistemas de Distribuição Radial.....	13
2.2.4. Fluxo de Carga em Alocação de Capacitores.....	17
2.2.5. Tipos de Bancos de Capacitores Usados.....	17
2.2.6. Função Objetivo no Problema de Alocação de Reativos.....	18
2.2.7. Formulação Matemática Geral.....	19
2.3. Algoritmos Combinatoriais.....	20
2.3.1 Simulated Annealing Usado no Problema Geral de Alocação de Capacitores.....	20
2.3.1.1. Algoritmo SA para o Problema de Capacitores Fixos.....	21
2.3.1.2. Algoritmo SA para o Problema Geral de Alocação de Capacitores.....	23
2.3.1.3. Resultados Numéricos Obtidos com o Algoritmo SA	25
2.3.2. Algoritmo Genético Usado no Problema de Alocação de Capacitores.....	25
2.3.2.1 Função Objetivo.....	25
2.3.2.2 Análise de Sensibilidade.....	26
2.3.2.3. Algoritmo Proposto em [3].....	27



2.3.2.4.	Resultados Obtidos.....	29
2.3.3.	Algoritmo de Busca Tabu (TS) para o Problema de Alocação de Capacitores.....	30
2.3.3.1.	Análise de Sensibilidade.....	30
2.3.3.2.	Algoritmo de Busca Tabu Proposto em [4]	30
2.3.3.3.	Algoritmo Busca Tabu Implementado em [4]	32
3.3.3.4.	Resultados Obtidos.....	33

3. ALGORITMO DE FLUXO DE CARGA RADIAL.....34

3.1. Algoritmo de Fluxo de Carga para Redes de Distribuição Radial:

Métodos de Soma de Potências.....	34
3.1.1. Introdução.....	34
3.1.2. Justificativa Matemática.....	35
3.1.3. Algoritmo Proposto.....	38
3.1.4. Comentários Adicionais sobre o Algoritmo de Fluxo de Carga.....	39

4. O ALGORITMO GENÉTICO.....40

4.1. Introdução.....	40
4.1.1. Técnicas de Buscas para Soluções de Problemas de Otimização.....	40
4.1.2. Técnicas Tradicionais de Otimização.....	40
4.1.3. Técnicas Enumerativas.....	42
4.1.4. Técnicas de Busca Aleatoriamente Direcionada.....	42
4.2. Algoritmo Genético: Teoria.....	43
4.2.1. Algoritmo Genético e Evolução Natural.....	44
4.2.2. Funcionamento.....	45
4.3. Algoritmo Genético Simples (AGS).....	46
4.3.1. Componentes e Estruturas de um AGS.....	47
4.4. Codificação e Avaliação da Função Objetivo.....	49
4.5. Discussão das Funções Usadas pelo Algoritmo Genético.....	49
4.5.1. Seleção.....	49

4.5.2.	Outros Operadores Genéticos.....	50
4.5.2.1.	Recombinação.....	51
5.5.2.2.	Mutação.....	51
4.6.	Ciclo Geracional.....	53
4.7.	Critério de Parada.....	53
4.8.	Funções Avançadas do Algoritmo Genético.....	54
4.8.1.	Teorema do "Schema".....	54
4.8.2.	Conceito de Building Blocks.....	56
4.8.3.	Modificações no Algoritmo Genético.....	56
4.8.3.1.	Formas Alternativas de Implementar a Seleção.....	56
4.8.3.1.1.	Mecanismo de Escalonamento (Scaling Mechanism).....	57
4.8.3.1.2.	Seleção Baseada em Ordenamento (Rank-Based Selection).....	57
4.8.3.1.3.	Seleção Baseada em Torneio (Tournament Selection).....	58
4.8.3.1.4.	Técnica de Seleção Estocástica do Resíduo (Stochastic Remainder Technique).....	58
5.	ALGORITMO GENÉTICO PROPOSTO.....	59
5.1.	Introdução.....	59
5.2.	Características Básicas do Algoritmo Genético para o Problema de Alocação de Capacitores Fixos.....	60
5.2.1.	Representação e Codificação do Problema.....	61
5.2.2.	Funções do Algoritmo Genético.....	62
5.2.2.1.	População.....	62
5.2.2.2.	Seleção.....	63
5.2.2.3.	Recombinação.....	65
5.2.2.4.	Mutação.....	66
5.2.3.	Programa de Controle do Algoritmo Genético.....	67
5.2.4.	Outras Funções ou Estratégias Implementadas.....	67
5.2.5.	Técnica de Redução de Barras Candidatas.....	68



5.2.6.	Determinação da Configuração Inicial.....	69
5.2.7.	Uso de Indicadores de Sensibilidade.....	70
5.3.	Algoritmo Genético Proposto.....	71
5.3.1.	Fase I: Estratégia Heurística.....	71
5.3.2.	Fase II: Algoritmo Genético.....	72
5.3.3.	Algoritmo Implementado.....	73
5.4.	O Problema de Alocação de Capacitores Variáveis.....	74

6. TESTES E RESULTADOS OBTIDOS.....76

6.1.	Sistema de 10 barras.....	77
6.2.	Sistema de 34 barras.....	84
6.3.	Sistema de 70 barras.....	92

7. CONCLUSÕES.....94

REFERÊNCIAS BIBLIOGRÁFICAS.....96

APÊNDICE A.99

A.1.	Dados completos do sistema de 10 barras.....	100
A.2.	Dados completos do sistema de 34 barras	102
A.3.	Dados completos do sistema de 70 barras.....	105

CAPÍTULO 1

INTRODUÇÃO

Em sistemas de distribuição radiais devem ser instalados capacitores para compensar a potência reativa a fim de reduzir a potência e as perdas de energia, melhorar o perfil de tensão e aliviar a capacidade do sistema. Assim, aparece o problema de como alocar capacitores no sistema a fim de atingir da melhor maneira os benefícios antes mencionados. Este problema é chamado de **problema geral de alocação de capacitores** ou **alocação ótima de capacitores** em sistemas de distribuição.

O problema geral de alocação de capacitores consiste em determinar a localização, os tipos e tamanhos dos capacitores que devem ser instalados e o esquema de controle para os capacitores variáveis de tal maneira que seja minimizada uma função objetivo sujeita a restrições de operação e de carga (tipicamente restrições de módulos de tensão) para os diferentes níveis de carga.

Este problema é interessante porque sua formulação matemática leva a um modelo matemático conhecido como sendo um problema de programação não linear inteiro misto (PNLIM) não diferenciável. A função objetivo do problema é não diferenciável onde os custos dos capacitores varia de forma discreta devido aos bancos de capacitores usados na prática, e a existência de uma função objetivo não diferenciável torna muito difícil a utilização da maioria dos métodos de



programação não linear. Como a carga de um sistema de distribuição é variável durante o dia, deve-se discretizar essa variação num número determinado de intervalos de operação que são chamados níveis de carga. Para cada nível de carga, deve-se determinar o esquema de operação (control settings) dos capacitores variáveis. Assim o tamanho e o esquema de operação dos capacitores são variáveis discretas. Portanto, a natureza do problema sugere a utilização de algoritmos de uso geral como os algoritmos combinatoriais tais como Simulated Annealing (AS), Algoritmos Genéticos (GA), Busca Tabu (TS), etc.

A solução de um problema de alocação de capacitores deve fornecer a informação das localizações (barras do sistema) onde devem ser instalados os capacitores, os tipos e tamanhos de capacitores que devem ser instalados em cada lugar (barra do sistema) e um esquema de controle (a posição de operação de cada um dos capacitores variáveis) para cada um dos diferentes níveis de carga considerados.

Na presente pesquisa foi desenvolvido um algoritmo genético para o problema de alocação de bancos de capacitores em sistemas de distribuição radial para o caso de alocação de capacitores fixos e também para a alocação de capacitores variáveis.

No capítulo 2 é apresentada a modelagem matemática mais usada para o problema de alocação de bancos de capacitores em sistemas de distribuição radiais. São analisadas, em forma resumida, as principais propostas para resolver o problema antes mencionado.

No capítulo 3 é apresentado um dos métodos de fluxo de carga especializados para resolver sistemas de distribuição radiais. Todos os algoritmos usados para resolver o problema de alocação de bancos de capacitores resolvem muitos subproblemas de fluxo de carga de sistemas radiais. Assim, um dos algoritmos de fluxo de carga radial mais usados é o método da soma de potências de Céspedes[17].

No capítulo 4 é apresentada a teoria sobre os algoritmos genéticos para problemas de tipo geral. É analisado em detalhe as formas de codificação, a avaliação da função objetivo, a implementação dos operadores genéticos (seleção, recombinação e mutação) e algumas funções avançadas apresentadas recentemente na teoria sobre algoritmos genéticos. É apresentado o



algoritmo genético proposto para o problema de alocação de bancos de capacitores em sistemas de distribuição radiais. Nesse capítulo também são analisadas propostas alternativas apresentadas na literatura especializada, assim como as novas propostas apresentadas na pesquisa. No capítulo 5, são apresentados testes realizados em 3 sistemas elétricos que são amplamente usados na literatura especializada para analisar o desempenho de algoritmos de alocação de capacitores em sistemas de distribuição radiais. São mostrados resultados para o caso de alocação de bancos de capacitores fixos e variáveis. Adicionalmente, são apresentados resultados parciais e detalhados do desempenho do algoritmo genético para um dos sistemas.

Finalmente, no capítulo 6 são apresentadas as principais conclusões da pesquisa e, no apêndice, os dados dos sistemas testados.



CAPÍTULO 2

Modelagem Matemática e Técnicas de Solução Usadas no Problema de Alocação de Capacitores

Neste capítulo, apresenta-se a modelagem matemática mais usada, na literatura especializada, para o problema de alocação de reativos em sistemas de distribuição radiais.

2.1 Formulação do Problema de Alocação de Reativos.

A formulação matemática do problema de alocação de capacitores em sistemas de distribuição radiais assume a seguinte forma:

$$\text{Minv} = \sum_{i=1}^{i=L} C_i(q_i^0) + \sum_{j=1}^{j=L} k_{ej} T_j P_{perdas,j}(x^j, q^j) \quad (2.1)$$

$$\text{s.a.} \quad P_{fluxo}(z^i, q^i) = 0$$

$$V_i^{\min} \leq V_i^j \leq V_i^{\max}$$

q_i^j discreto:

Para capacitores fixos: $q_i^l = q_i^j = q_i^0; l, j \in \{1, 2, 3, \dots, L\}$

Para capacitores variáveis: $0 \leq q_i^j \leq q_i^0; i \in \{1, 2, 3, \dots, I\}$

Na formulação acima, a relação $[P_{\text{fluxo}}(z^i, q^i) = 0]$ representa as restrições de fluxo de carga; $[V_i^{\min} \leq V_i^j \leq V_i^{\max}]$ representa restrições de limites de operação da tensão para cada barra e para cada nível de carga analisado; q^0 é o vetor que determina o tamanho do banco de capacitores e cada elemento de q^0 tem um valor que é múltiplo do tamanho padrão de banco de capacitores; q^j é o vetor que caracteriza o esquema de controle dos bancos de capacitores em cada nível de carga j e cujos componentes são variáveis discretas; $C_i(q_i^0)$ é uma função não diferenciável e representa o investimento pela instalação do capacitor na barra i ; $P_{\text{perdas},j}$ é a perda de potência quando o sistema opera no nível de carga j durante o tempo T_j ; k_{ej} representa os custos de perdas de energia para cada nível de carga $x_k = [P_k, Q_k, |V_k|^2]$ e $z_k = [P_k, Q_k]$ representam vetores de estado das variáveis P_k (Potência Ativa), Q_k (Potência Reativa) e $|V_k|^2$ magnitude da tensão na barra k ; L é o número de níveis de carga e I é o número de barras candidatas à instalação de capacitores.

2.2. Métodos de Otimização Clássica

2.2.1. Formulação do Problema

Nesta seção é discutida separadamente cada etapa importante do problema de alocação de capacitores, obtendo-se finalmente a formulação matemática do conjunto de equações apresentada no capítulo anterior.



2.2.2. Curva de Duração de Carga.

A perda de energia é um dos critérios que deve ser usado na análise e, portanto, deve ser considerado um período de tempo T . Para o período de tempo T , assume-se que as variações de carga podem ser representadas de maneira discretizada como é mostrado na Figura 2.2. Além disso, assume-se que o perfil de carga varia de maneira uniforme. Portanto, assumindo que $S(\tau)$ é a curva de variação de carga comum, então uma carga qualquer como Q_L pode ser representada da seguinte maneira:

$$Q_L(\tau) = Q_L^0(\tau)s(\tau) \quad (2.2)$$

em que Q_L^0 representa o valor de pico de carga. Assim, o período T de análise pode ser dividido como um conjunto de nt períodos (intervalos) com perfil de carga constante onde nt é o número de níveis de carga em que foi discretizada a curva de carga.

Para cada nível de carga, deve-se satisfazer as seguintes exigências:

- As equações de fluxo de potência.
- Os limites de operação dos módulos de tensão.
- A capacidade e as restrições de operação das variáveis de controle que neste caso são os capacitores.

Portanto, é evidente que para conhecer a operação em cada nível de carga, deve-se resolver um problema de fluxo de carga de sistemas de distribuição radial.

2.2.3. Fluxo de Carga em Sistemas de Distribuição Radial

Existem muitos métodos para resolver de maneira eficiente o fluxo de carga de sistema de distribuição radial disponíveis na literatura especializada. Para fins de ilustração,



apresenta-se a metodologia de fluxo de carga radial conhecida como **DistFlow (distribuição de fluxo – distribution flow)** apresentada em [8].

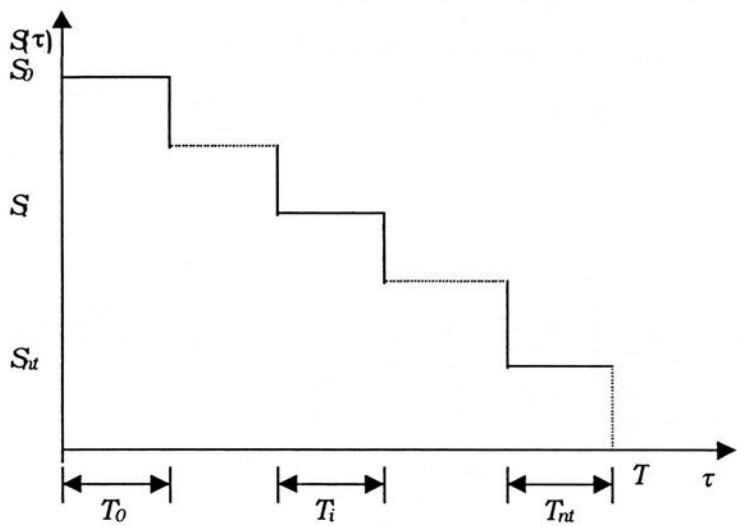


Figura 2.1: Curva de Duração de Carga

A metodologia DistFlow é usada em fluxo de carga radial em substituição aos métodos convencionais de fluxo de carga AC como o método de Newton e suas versões desacopladas. Quando são usados os algoritmos combinatoriais para resolver o problema de alocação ótima de capacitores, deve-se resolver em torno de milhares de problemas de fluxo de carga de sistemas radiais e, portanto, é evidente a necessidade de dispor de um algoritmo de fluxo de carga para sistemas radiais extremamente rápido e robusto.

Na figura 2.2, apresenta-se um diagrama unifilar de um sistema típico de distribuição radial. Pode ser mostrado que o fluxo de potência através de um ramo pode ser representado através das seguintes equações recursivas:

$$P_{k+1} = P_k - r_{k+1}P_k^2 + (\frac{Q_k}{V_k})^2 - P_{LK+1} \tag{2.3}$$

$$Q_{k+1} = Q_k - x_{k+1}P_k^2 + (\frac{Q_k}{V_k})^2 - Q_{LK+1} + Q_{ck+1} \tag{2.4}$$

$$V_{k+1}^2 = V_k^2 - 2(r_{k+1}P_k + x_{k+1}Q_k) + (r_{k+1}^2 + x_{k+1}^2)P_k^2 + \left(\frac{Q_k}{V_k}\right)^2 \quad (2.5)$$

em que P_k e Q_k são os fluxos de potência ativa e reativa chegando à barra $k+1$ através do ramo que conecta as barras k e $k+1$; V_k é o módulo da tensão na barra k ; Q_{ck} é a injeção de potência reativa devido ao capacitor alocado na barra k e, r_{k+1} e x_{k+1} são a resistência e a reatância do ramo que conecta as barras k e $k+1$.

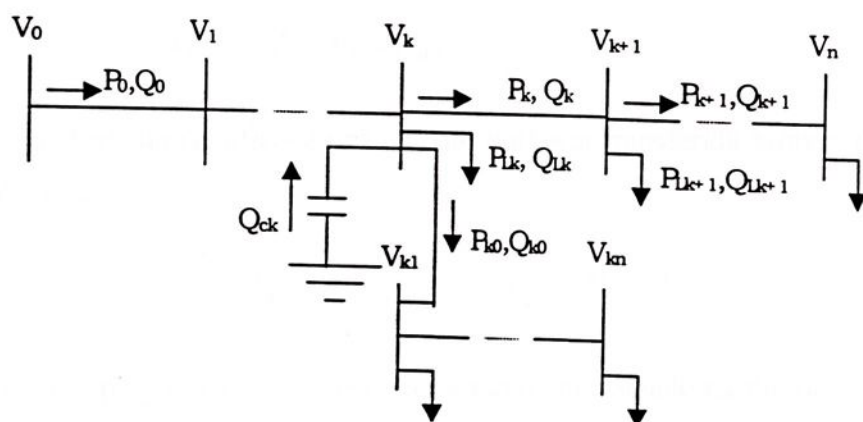


Figura 2.2: Diagrama Unifilar de um Alimentador de Distribuição

As equações (2.3), (2.4) e (2.5), chamadas de equações de fluxo de ramos, podem ser representadas em forma sintetizada da seguinte forma:

$$x_{k+1} = f_{k+1}(x_k, u_{k+1}) \quad (2.6)$$

em que $x_k = [P_k, Q_k, V_k^2]^T$ e $u_{k+1} = Q_{ck+1}$

Evidentemente se não existe capacitor na barra k então para uma equação do tipo (2.6) temos que $u_k=0$. Agora, pode-se usar um vetor u para representar todos os capacitores colocados em nc barras do sistema:

$$u^T = [u_1 \dots u_{nc}] = [Q_{cn_1} \dots Q_{cn_{nc}}]$$

Adicionalmente às equações (2.6), o sistema deve satisfazer condições de fronteira para cada ramal (ou ramificação) e considerando a ramificação principal como ramificação 0. Por exemplo, para a ramificação k mostrada na figura 2.2, deve-se satisfazer as seguintes condições de fronteira:

1. Na barra de separação (branching) k onde a ramificação é conectada ao alimentador principal, define-se uma variável artificial V_{k0} e seja:

$$x_{k03} = V_{k0}^2 = V_k = x_{ok3} \quad (2.7)$$

2. No final da ramificação não existe potência transferida através dos ramos, assim temos que:

$$x_{kn_1} = P_{kn} = 0; \quad x_{kn_2} = Q_{kn} = 0 \quad (2.8)$$

Portanto, para um sistema genérico como o apresentado na figura 2.2 (com n barras e l ramificações além da ramificação principal), existem $3(n + l + 1)$ equações DistFlow correspondentes às equações (2.6), (2.7) e (2.8). Este conjunto de equações podem ser representadas na seguinte forma compacta:

$$G(x,u) = 0 \quad (2.9)$$

em que $x = [x_1^T \dots x_1^T x_0^T]^T$ e $x_k = [x_{k0}^T \dots x_{kn}^T]^T$.

As equações DistFlow mostradas em (2.9) simplesmente representam uma forma equivalente de encontrar o ponto de operação de um sistema de distribuição radial, isto é, representam **equações de fluxo de carga** para sistemas de distribuição radial. No sistema de equações (2.9), são dados do problema, os parâmetros nos ramos, as demandas nas barras e o conjunto de capacitores alocados (conhecendo a localização, tamanho, tipo e ponto de operação). Resolver (2.9) consiste em encontrar os fluxos de potência ativa e reativa em todos os ramos do sistema e o módulo de tensão em todas as barras do sistema. Em outras palavras, (2.9) consiste em encontrar o ponto de operação x do sistema para um determinado perfil de carga $P_{Li}, Q_{Li} \ i = 1, \dots, n$ e



uma determinada proposta de alocação de capacitores u . É evidente que nos algoritmos combinatoriais, deve-se resolver muitos problemas do tipo (2.9).

2.2.4. Fluxo de Carga Para Alocação de Capacitores

Para o problema de alocação de capacitores com nt diferentes níveis de carga, além do nível zero ou referencial que geralmente é a carga pico, existem nt problemas de fluxo de carga do tipo mostrado em (2.9), assim temos o seguinte sistema:

$$G^i(x^i, u^i) = 0 \quad i = 0, 1, \dots, nt \quad (2.10)$$

em que x^i e u^i representam o estado e as variáveis de controle para o perfil de carga i .

As restrições de tensão basicamente se restringem a manter valores mínimo e máximo para o módulo da tensão das barras do sistema para cada tipo de perfil de carga, assim deve-se satisfazer as seguintes restrições:

$$(V^{\min})^2 \leq (V_j^i)^2 = v_j^i(x^i) \leq (V^{\max})^2 \quad j = 1, \dots, n; \quad i = 0, 1, 2, \dots, nt \quad (3.11)$$

As restrições anteriores representam restrições de desigualdade funcionais e podem ser representadas na seguinte forma:

$$H^i(x^i) \leq 0 \quad i = 0, 1, \dots, nt \quad (3.12)$$

2.2.5. Tipos de Bancos de Capacitores Usados

No problema de alocação de reativos são usados dois tipos de bancos de capacitores:



1. Capacitores Fixos:

Os capacitores fixos são tratados simplesmente como fontes de potência reativa de módulo constante para todos os níveis de carga então é válida a seguinte relação:

$$u^0 = u^I = \dots = u^{nt} \quad (2.13)$$

2. Capacitores Variáveis (Switched):

Neste caso se assume que podem ser alterados os taps dos capacitores para operar de maneira diferente para cada tipo de perfil de carga, portanto, para cada capacitor k existem $nt + 1$ formas de operação, u_k^i ; $i = 0, 1, \dots, nt$ que devem ser determinadas, uma para cada tipo de operação ou perfil de carga. É lógico supor que a capacidade instalada para operar na carga pico, u_k^0 é maior ou igual que a capacidade em operação para outro nível de carga u_k^i . Portanto, os tamanhos dos capacitores (capacidade instalada) que devem ser determinados u^0 e as capacidades operando (ponto de operação do capacitor variável) para cada perfil de carga devem satisfazer a seguinte restrição:

$$0 \leq u_k^i \leq u_k^0 \quad (2.14)$$

2.2.6. Função Objetivo no Problema de Alocação de Reativos

A função objetivo apresenta duas parcelas claramente diferenciadas: os custos devido as perdas de energia e o investimento pela instalação dos capacitores.

Para cada nível (perfil) de carga i as perdas de potência no sistema é $p_i(x^i)$ e, portanto, o custo pelas perdas de energia pode ser representado da seguinte forma:

$$k_e \sum_{i=0}^{nt} T_i p_i(x^i) \quad (2.15)$$



em que T_i é a duração em horas da carga no nível i e a constante k_e é o custo por unidade de energia (dólares/KWh). Pode-se adicionar na função objetivo o custo das perdas de potência ativa para o nível de carga pico modificando adequadamente T_0 .

A função de custo do capacitor apresenta a forma escalonada (step like) mostrada na Figura 2.3 onde, na prática, são instalados bancos de capacitores discretos. Assim, existe um custo fixo pela infra-estrutura de instalação que deve ser considerado toda vez que é decidida a instalação de capacitores e outra parcela que corresponde à compra dos bancos de capacitores instalados.

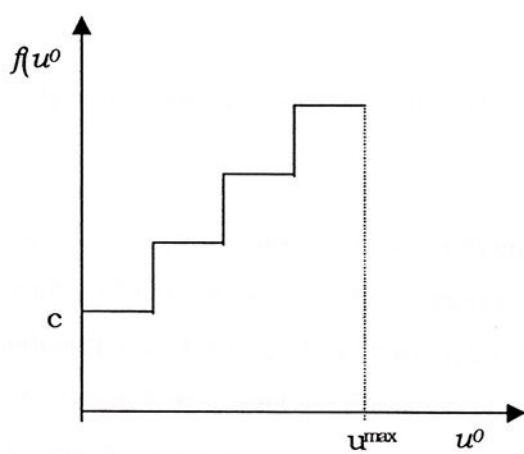


Figura 2.3: Função de Custo de Capacitores

2.2.7. Formulação Matemática Geral

Seja nc o número de barras candidatas a alocação de capacitores, C_1 o conjunto de capacitores fixos e C_2 o conjunto de capacitores variáveis então o problema geral de alocação de capacitores pode ser formulado como o seguinte PNLIM:

$$\begin{aligned} \min v &= k_e \sum_{i=0}^{nt} T_i p_i(x^i) + \sum_{k=1}^{nc} f(u_k^0) & (2.16) \\ \text{s.a.} \quad & G^i(x^i, u^i) = 0 \quad i = 0, 1, \dots, nt \\ & H^i(x^i) \leq 0; \quad i = 0, 1, \dots, nt \end{aligned}$$

$$0 \leq u_k^i = u_k^0; k \in C_1$$

$$0 \leq u_k^i \leq u_k^0; k \in C_2$$

2.3. Algoritmos Combinatoriais.

2.3.1. Simulated Annealing Usado no Problema Geral de Alocação de Capacitores.

Em [1,2] foi apresentado um algoritmo de simulated annealing (SA) para o problema de alocação de capacitores.

Em [1] são apresentadas claramente as principais etapas que devem ser definidas ou caracterizadas para usar um algoritmo SA num problema específico. Essas etapas são separadas em 4 partes: 1) identificação do espaço de configurações factíveis; 2) identificação de um movimento factível ou mecanismo de transição que permite encontrar uma configuração candidata a partir da configuração corrente; 3) identificação da função objetivo e, 4) formulação do programa de esfriamento (cooling schedule).

A estratégia de busca escolhida determina as características das 3 primeiras etapas do parágrafo anterior. O Algoritmo SA apresentado tem como estratégia fundamental iniciar o processo a partir de uma configuração factível e as novas configurações aceitas devem ser necessariamente factíveis. Assim, o espaço de configurações factíveis é claramente especificado. Em [1], para o problema de alocação de capacitores fixos, uma configuração é definida como uma proposta de alocação $\Omega = \{u_1, u_2, \dots, u_c\}$ em que u^i representa a capacidade de geração reativa do capacitor alocado na barra i e permanece fixo para todos os níveis de carga. Para o caso de alocação de capacitores variáveis uma configuração é representada por $\Omega = \{u_1, u_2, \dots, u_c\}$ em que $u_i = \{u_i^1, u_i^2, \dots, u_i^{n_i}\}$ é um vetor de dimensão n_i e u_i^j determina a capacidade operativa (ponto de operação - posição de operação) do capacitor alocado na barra i operando no nível de carga j . Assim, uma configuração e o

espaço de configurações está adequadamente caracterizado. O mecanismo de transição é modificado para rejeitar sistematicamente configurações candidatas inactíveis e aceitar as configurações factíveis de acordo com os critérios tradicionais de SA. A estratégia apresentada, de aceitar somente configurações factíveis, representa um tipo de filosofia de trabalho para implementar problemas restritos usando os algoritmos combinatoriais; outra estratégia significativamente diferente é aceitar configurações factíveis e inactíveis, penalizando adequadamente a função objetivo das configurações inactíveis. Escolher uma das alternativas representa tomar uma decisão que deve influir significativamente no desempenho do algoritmo desenvolvido e essa decisão depende de vários fatores especialmente do tipo de problema que está sendo pesquisado.

No Programa de Esfriamento devem ser especificadas 4 grandezas: 1) a temperatura inicial; 2) a taxa de diminuição de temperatura; 3) o número de tentativas em cada nível de temperatura e, 4) o critério de parada.

Em [1] a temperatura é determinada empiricamente de maneira que no primeiro nível de temperatura sejam aceitas 85% das configurações candidatas para o problema de alocação de capacitores fixos e 55% das configurações candidatas para o problema de alocação de capacitores variáveis. Nestes testes, uma configuração candidata pode ser rejeitada porque ela é inactível ou, sendo factível, também pelos critérios usuais de SA.

Em [1] é usada uma taxa de diminuição de temperatura β que pode variar no intervalo $\beta \in [0,85-0,98]$ e o valor usado em cada iteração é dependente de dois fatores: 1) taxa de aceitação e, 2) valores médios e a variância da função objetivo para o nível de temperatura considerado. Em cada nível de temperatura é realizado um conjunto de n_k tentativas sem especificar o valor de n_k utilizado. Também é usado um critério de parada misto: se durante 5 níveis consecutivos de temperatura o valor médio da função objetivo não variar significativamente (por exemplo menos de 1%) então o processo é considerado congelado e, deve-se parar.

2.3.1.1. Algoritmo SA para o Problema de Capacitores Fixos.

Em [2] é apresentado o seguinte algoritmo SA para o problema de alocação de reativos com capacitores fixos:

1 - Leitura de dados e dos parâmetros de controle do algoritmo SA → feita a leitura dos dados, escolher ou determinar os parâmetros de controle.

2 - Obtenção da configuração factível inicial: → implementar os seguintes passos para obter uma configuração inicial:

(a) Escolher uma configuração inicial do espaço de configurações, usando um gerador de número aleatório.

(b) Para cada nível de carga, processar um **fluxo de carga** de distribuição radial para checar a factibilidade da configuração para todos os níveis de carga. Se alguma restrição para algum nível de carga é violada então voltar a (a); em caso contrário ir a (c).

(c) Calcular o valor da função objetivo (custo total).

3 - Programa de esfriamento - cooling schedule: → para cada nível de temperatura T_k realizar n_k tentativas de obtenção de novas configurações. Para $p=1,2,\dots,n_k$, implementar os passos 4 e 5; em caso contrário, se já foram realizadas as n_k tentativas, ir ao passo 6.

4 - Geração de uma configuração candidata factível:

(a) Encontrar uma configuração candidata usando um dos mecanismos de transição propostos.

(b) Para cada nível de carga, processar um **fluxo de carga** de distribuição radial para checar a factibilidade da configuração para todos os níveis de carga. Se alguma



restrição para algum nível de carga é violada então voltar a (a); em caso contrário ir a (c).

(c) Calcular o valor da função objetivo (custo total).

5 - Teste de aceitação da configuração candidata: → aceitar ou rejeitar a configuração candidata de acordo com os critérios de aceitação de SA.

6 - Checar o critério de parada: → se o critério de parada não é satisfeito voltar ao passo 3; em caso contrário ir ao passo 7.

7 - Imprimir os resultados finais (configuração ótima ou quase ótima).

A resposta que fornece o algoritmo anterior é um vetor $u = (u_1, u_2, \dots, u_{nc}) = u^p$ de onde podem ser obtidos os tamanhos adequados do vetor u^0 usando a relação $u_i \leq u_i^0, i=1, 2, \dots, nc$ e u_i^0 seria o menor valor que é um múltiplo inteiro de um banco de capacitores. Assim, por exemplo, se foi usado um tamanho de passo de 30 kVar na aplicação do mecanismo de perturbação para obter uma configuração candidata o processo pode parar indicando para uma barra i uma capacidade de $u_i = 270$ kVar e se existem bancos de 300 kVar então a capacidade que deve ser alocada é $u_i^0 = 300 \text{ kVar}$

2.3.1.2. Algoritmo SA para o Problema Geral de Alocação de Capacitores.

Em [2] é apresentado o seguinte algoritmo SA para o problema geral de alocação de reativos usando capacitores variáveis. O algoritmo é parecido ao caso anterior mas como o espaço de configurações foi acrescido significativamente, então existe a proposta de usar dois laços de controle para o algoritmo: um laço externo que controla a posição de controle dos capacitores para a carga pico e um laço interno que controla as posições de controle dos capacitores para os outros níveis de carga. Esta estratégia precisa da utilização de um programa de esfriamento local para cada nível de carga no laço interno. O algoritmo apresentado em [2] é o seguinte:

- 1 - Leitura de dados e dos parâmetros de controle do algoritmo SA:
- 2 - Obtenção da configuração factível inicial:
- 3 - Programa de Esfriamento Global - cooling schedule $\rightarrow$ para cada nível de temperatura T_k realizar n_k tentativas de obtenção de novas configurações. Para $p=1,2,\dots,n_k$, implementar os passos 4 a 7; em caso contrário, se já foram realizadas as n_k tentativas, ir ao passo 8.
- 4 - Geração de uma configuração candidata factível:
- 5 - SA com programa de esfriamento local:
- 6 - Checar a factibilidade global da configuração candidata e calcular a função objetivo global:
- 7 - Teste de aceitação da configuração candidata global:
- 8 - Checar o critério de parada: $\rightarrow$ Se o critério de parada não é satisfeito voltar ao passo 3; em caso contrário ir ao passo 9.
- 9 - Imprimir os resultados finais (configuração ótima).

A resposta que fornece o algoritmo anterior é um vetor $u = (u_1, u_2, \dots, u_{nc}) = u^p$ onde $u_i = (u_i^1, u_i^2, \dots, u_i^m)$ e u_i^j representa o esquema de controle na barra i para o perfil de carga j . Portanto, o tamanho adequado do vetor u^0 pode ser obtido usando a relação $u_i^p \leq u_i^0; i=1,2,\dots,nc$ e u_i^0 seria o menor valor que é um múltiplo inteiro de um banco de capacitores.

O algoritmo apresentado, aparentemente seria demorado devido aos laços internos de SA, isto é, para estar em condições de tentar uma transição global simples tentando passar da



configuração corrente para uma configuração candidata aceita, deve-se implementar m algoritmos SA internos.

2.3.1.3. Resultados Numéricos Obtidos com o Algoritmo SA.

Em [2] é testado um sistema de 69 barras cujos dados são apresentados em [10]. Aparentemente os resultados encontrados são interessantes mas existem alguns testes que devem ser confirmados. Por exemplo, as perdas encontradas para o sistema com carga média e sem alocação de capacitores não corresponde com os valores obtidos em [8] para o mesmo sistema o que indica que um dos testes está errado pois, trata-se de um mesmo sistema com condições de carga iguais.

Também não é possível avaliar o esforço computacional porque não é mencionado o tempo de computação ou, mais importante ainda, não é mencionado o número de fluxos de carga que foram resolvidos. Para este tipo de problema o melhor indicador de esforço computacional seria o número de vezes que é acionada a subrotina de fluxo de carga radial pois esse passo requer o maior esforço computacional.

2.3.2. Algoritmo Genético Usado no Problema Geral de Alocação de Capacitores.

Em [3] é apresentado um algoritmo genético aplicado ao problema de alocação de capacitores.

2.3.2.1. Função Objetivo.

A função de adaptação usada (fitness value) em [3] é a própria função objetivo que neste caso tem 3 parcelas: custos relacionados com as perdas de energia, custos relacionados com a diminuição de perdas para a carga pico e os custos devido à instalação dos bancos de capacitores.



Todas as outras suposições anteriores persistem tais como: análise para um determinado período T , discretização da variação da carga em nt níveis de operação, variação uniforme do perfil de carga e escolhendo nc barras candidatas a alocação de capacitores.

A função objetivo assume a seguinte forma:

$$\min v = k_e \sum_{i=1}^{nt} T_i p_i + k_p p_0 + k_c \sum_{j=1}^{nc} C_j \quad (2.17)$$

em que k_e , k_p e k_c são constantes de custo de energia, custo de potência pico e custos de capacitores respectivamente. p_i representa as perdas para o nível de carga i operando durante um tempo T_i ; p_0 representa as perdas na carga pico e C_j o tamanho do banco de capacitores alocado na barra j e dado em kVar. A função objetivo pode ser modificada para diminuir o número de localizações, isto é, para tentar alocar capacitores num número mínimo de barras. Este objetivo pode ser atingido penalizando o número de localizações na função objetivo da seguinte forma:

$$\min v = k_e \sum_{i=1}^{nt} T_i p_i + k_p p_0 + k_c \sum_{j=1}^{nc} C_j + p_f nc^2 \quad (2.18)$$

em que p_f é um fator de penalidade.

2.3.2.2. Análise de Sensibilidade.

Na configuração ótima de um problema de alocação de capacitores geralmente são alocados capacitores num número muito reduzido de barras, por exemplo, no sistema teste de 70 barras muito usado na literatura especializada são alocados capacitores no máximo em 3 ou 4 barras. Portanto, geralmente não é uma boa estratégia considerar todas as barras como candidatas à alocação de capacitores pois aumentaria desnecessariamente o esforço computacional. O ideal seria escolher um número reduzido de barras candidatas desde que nesse número reduzido se encontrem todas as barras da configuração ótima.



Existem várias estratégias para escolher um número reduzido de barras candidatas à alocação de capacitores e a maioria dessas estratégias está baseado em análise de sensibilidade. Em [3] é usado um indicador de sensibilidade baseada na máxima variação das perdas ativas no sistema devido à variação das potências reativas nas barras.

A seguir são discutidos os aspectos principais do algoritmo genético implementado em [3]. Infelizmente, os autores não explicam como foi implementado o algoritmo específico do problema de alocação de capacitores.

Assim, não são apresentados adequadamente os seguintes aspectos básicos do algoritmo genético:

- Representação e Codificação do Problema.
- Determinação da População Inicial.
- Avaliação da Função Objetivo.

O algoritmo genético foi originalmente desenvolvido para um problema de maximização. Embora a transformação para minimização não seja trivial, na pesquisa não é mencionada como foi contornado este problema.

2.3.2.3. Algoritmo Proposto em [3]

Parte I - Determinação das barras candidatas à alocação:

1. Entrar com os dados do sistema (parâmetros da rede, cargas, etc.).
2. Montar a matriz de impedâncias.



3. Encontrar os fatores de sensibilidade.
4. Ordenar as barras baseado nos fatores de sensibilidade.
5. Selecionar as barras com maiores fatores de sensibilidade.

Parte II - Determinação do tamanho ótimo, localização, número e tipo de capacitores:

Uma configuração típica está formada por um vetor de $nc \times nt$ variáveis.

1. Encontrar a população inicial formada por k configurações.
2. Calcular a função objetivo de cada configuração. Os cálculos são obtidos usando o seguinte esquema (onde o nível 0 corresponde à carga de pico e nesse nível, deve-se encontrar as perdas P_0):

Procedimento de avaliação:

inicia

nível de carga = 0

enquanto nível de carga < nt fazer:

iniciar

determinar o tamanho dos capacitores

processar fluxo de carga

encontrar as perdas de potência

incrementar o nível de carga

fim

encontrar perdas de energia (P_{en})



encontrar os custos dos capacitores (P_c)

calcular a função objetivo

retornar (fitness)

fim

3. Encontrar a próxima geração usando seleção, recombinação e mutação e preservando sempre a melhor configuração encontrada (estratégia elitista - a incumbente participa de recombinação e mutação mas também é preservada).
4. Repetir os passos 2 e 3 até que seja gerado um número máximo de gerações, previamente especificado.
5. Em qualquer caso, o capacitor de menor tamanho que é requerido por todos os níveis de carga pode ser considerado como o tamanho do capacitor fixo que deve ser alocado na barra.

2.3.2.4. Resultados Obtidos em [3]

São testados 2 sistemas conhecidos pela literatura especializada, um sistema de 9 barras e um sistema de 30 barras. Os melhores parâmetros de controle usados, aparentemente, foram: tamanho da população de 50 (para ambos sistemas), taxa de crossover de 0,8 e taxa de mutação de 0,003. Do gráfico apresentado, pode-se concluir que o algoritmo converge após avaliar 100 gerações. Assim, para resolver o sistema de 9 barras foram resolvidos $50(3)(100) = 15000$ problemas de fluxo de carga.



2.3.3. Algoritmo de Busca Tabu (TS) para o Problema de Alocação de Capacitores

Busca Tabu foi desenvolvido por F. Glover [18] que inventou um conjunto de princípios (funções) que permite resolver um problema de maneira mais inteligente, de forma integrada e eficiente. Para resolver cada problema específico leva-se em conta que cada uma das funções podem ser implementadas de várias formas diferentes dependendo das características do problema e do nível de sofisticação da implementação do algoritmo TS. Em [4] é apresentado um algoritmo TS para o problema de alocação de capacitores. Portanto, apresenta-se uma análise crítica do algoritmo TS apresentado em [4]

2.3.3.1. Análise de Sensibilidade

Em [4], antes de iniciar o processo TS é realizado um processo de redução de barras candidatas, assim uma análise de sensibilidade elimina a maioria das barras candidatas, obtendo-se um número reduzido de barras candidatas à alocação de capacitores. São escolhidas como barras candidatas aquelas que apresentam um máximo impacto das perdas ativas com respeito à variação de potência reativa usando a seguinte relação:

$$\frac{\partial P_{loss}}{\partial Q_{ij}} = \sum_{i=0}^b \sum_{j=0}^n r_{i,j+1} \left\{ (2P_{ij} \frac{\partial P_{ij}}{\partial Q_{ij}} + 2Q_{ij}) / V_{ij}^2 - 2 \frac{V_{ij}}{\partial Q_{ij}} (P_{ij}^2 + Q_{ij}^2) / V_{ij}^3 \right\} \quad (2.19)$$

2.3.3.2. Algoritmo de Busca Tabu Proposto em [4]

Em [4] é apresentado um algoritmo de busca tabu aplicado ao problema de alocação de capacitores. Entretanto, a implementação do algoritmo de busca tabu para resolver o problema de alocação de capacitores deve levar em conta as seguintes implementações:



1. Algoritmo Básico de Busca Tabu :

O algoritmo básico de busca tabu apresenta as seguintes partes:

- Dada a configuração corrente, deve-se definir e caracterizar a **vizinhança da configuração corrente** e, de acordo com a lógica de TS, deve-se passar para a melhor configuração vizinha (ou para a menos pior).
- A formação da **Lista Tabu** que armazena os atributos das configurações vizinhas que estão proibidas. Este mecanismo evita retornar a configurações já visitadas (ciclagem).
- A definição dos **critérios de aspiração** que tenta contornar as proibições impostas pela lista tabu.
- Implementação de **estratégias de intensificação e diversificação**. Em [4] estas estratégias são basicamente implementadas usando as **memórias baseadas em frequência**.

2. Algoritmo Avançado de Busca Tabu :

Um algoritmo avançado de busca tabu deve levar em conta os seguintes aspectos:

- Montar uma estratégia adequada para gerar uma **configuração inicial de boa qualidade** ou, ainda melhor, um conjunto de configurações de boa qualidade e de composição diversificada.
- Armazenamento das chamadas **configurações de elite** que podem ser posteriormente usadas para implementar a intensificação e a diversificação, e também na implementação do chamado "Path Relinking" que é usado para gerar novas configurações de qualidade, "misturando" adequadamente atributos das configurações de elite.



- Montar uma estratégia destinada a reduzir o número de vizinhos candidatos. Em TS, dada uma configuração, existe um número de configurações vizinhas que devem ser analisadas e TS passa para a melhor configuração vizinha. Se o número de vizinhos é elevado, TS pode precisar de um elevado esforço computacional para realizar um simples movimento. Existem várias propostas para reduzir o número de vizinhos que devem ser analisadas no algoritmo TS. As principais propostas são o chamado critério de **aspiração plus** e a formação de uma **lista de vizinhos considerados de elite**. Para problemas específicos, pode-se ainda montar uma lista reduzida, aproveitando as características do problema.

Em [4] é especificada a chamada **estrutura de solução**, isto é, uma configuração típica e as outras grandezas típicas como valor das perdas, função objetivo, etc. Assim, uma estrutura de solução assume a seguinte forma:

Estrutura de solução: { localizações candidatas, posições (setting) de operação dos capacitores, capacidade instalada, perdas de potência ativa, magnitude de tensão, valor da função objetivo, contador de frequência }

Em [4] também são mencionadas as funções de TS como a lista tabu, o critério de aspiração, intensificação e diversificação. Os autores mencionam que foram usados lista tabu com uma duração de 15 movimentos, um critério de aspiração que elimina a proibição se a configuração candidata é melhor das últimas k configurações aceitas e o uso de intensificação e diversificação baseado em frequência.

2.3.3.3. Algoritmo Busca Tabu Implementado em [4]

É implementado o seguinte algoritmo TS:

1. Inicialização
2. Análise de Sensibilidade



Selecionar as barras candidatas a alocação de capacitores usando a equação (3.18)

3. Geração de uma configuração inicial factível
4. Avaliação de um passo TS
5. Atualizar a configuração corrente
6. Atualizar a lista tabu
7. Verificar o critério de parada
8. Imprimir resultados finais

No algoritmo apresentado em [4] não aparece a utilização de intensificação nem de diversificação. Entretanto, o problema mais sério é que não aparece a estrutura de vizinhança nem a técnica de obtenção da configuração de elite. Em outras palavras, não se especifica quais são os tipos de movimentos que permitem identificar configurações vizinhas da configuração corrente nem quantos desses vizinhos devem ser analisados.

2.3.3.4. Resultados Obtidos

Os resultados realizados são apresentados para o sistema de 70 barras, depois de efetuados vários testes para alocação de capacitores



CAPÍTULO 3

Algoritmo de Fluxo de Carga Radial.

3.1. Algoritmo de Fluxo de Carga Para Redes de Distribuição Radiais: Método de Soma de Potências.

3.1.1. Introdução.

Neste capítulo é analisado de forma resumida o algoritmo de fluxo de carga para redes de distribuição radiais desenvolvido por Cespedes e apresentado em [17]. A metodologia proposta é relativamente simples de implementar e foi usada na forma original ou com algumas modificações em muitos trabalhos de pesquisa que usam algoritmos de fluxo de carga para sistemas de distribuição radial.

Nos últimos anos foram desenvolvidos vários algoritmos especializados de fluxo de carga para sistemas de distribuição radial. Em sistemas de transmissão os algoritmos mais usados são os métodos de Newton e suas versões desacopladas. Entretanto, para sistemas de distribuição que normalmente operam em configuração radial foram desenvolvidos algoritmos rápidos aproveitando a estrutura radial, isto é, ausência de laços ou malhas no sistema elétrico.



3.1.2. Justificativa Matemática.

A metodologia consiste basicamente em eliminar as variáveis θ da formulação convencional. A idéia básica consiste em resolver para cada ramo do sistema elétrico a seguinte equação básica que está relacionada com a figura 3.1:

$$V_r^4 + [2(RP + XQ) - V_s^2] V_r^2 + (P^2 + Q^2)(R^2 + X^2) = 0 \quad (3.1)$$

onde:

s representa o nó do lado da fonte

r representa o nó do lado da carga

V_s é o módulo da tensão do nó s.

V_r é o módulo da tensão do nó r.

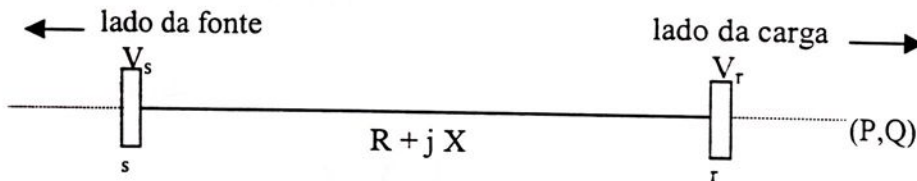


Figura 3.1. : Modelo Básico.

P representa a carga ativa

Q representa a carga reativa

Em outras palavras, conhecidos os valores de R, X, V_s , P e Q então é possível encontrar o valor de V_r usando (3.1), pois neste caso, deve-se resolver uma equação algébrica de quarta ordem redutível para uma equação algébrica de segunda ordem.

Demonstração de (3.1) :

A equação (3.1) pode ser facilmente demonstrada usando a figura 4.2.

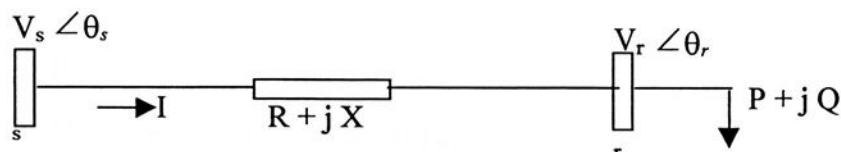


Figura 3.2 : Formulação Matemática.

A corrente elétrica no ramo pode ser representada pela seguinte relação:

$$I = \frac{V_s \angle \theta_s - V_r \angle \theta_r}{R + jX} \quad (3.2)$$

A potência da carga pode ser representada da seguinte forma:

$$S = V_r I^*$$

e assim

$$\begin{aligned} I^* &= \frac{S}{V_r} = \frac{P + jQ}{V_r \angle \theta_r} \\ I &= \frac{P - jQ}{V_r \angle -\theta_r} \end{aligned} \quad (3.3)$$

Igualando (3.2) e (3.3) temos o seguinte:

$$\frac{V_s \angle \theta_s - V_r \angle \theta_r}{R + jX} = \frac{P - jQ}{V_r \angle -\theta_r}$$

$$V_s V_r \angle \theta_s - \theta_r - V_r^2 = (P - jQ)(R + jX)$$

$$V_r V_s \cos(\theta_s - \theta_r) - V_r^2 + j V_r V_s \sin(\theta_s - \theta_r) = (RP + XQ) + j(XP - RQ)$$

temos as seguintes relações (igualando partes real e imaginária):

$$V_r V_s \cos(\theta_s - \theta_r) = RP + XQ + V_r^2 \quad (3.4)$$

$$V_r V_s \sin(\theta_s - \theta_r) = XP - RQ \quad (3.5)$$

Elevando ao quadrado (3.4) e (3.5) e somando é possível eliminar as funções trigonométricas, e reagrupando os termos temos o seguinte:

$$V_r^4 + [2(RP + XQ) - V_s^2] V_r^2 + (P^2 + Q^2)(R^2 + X^2) = 0$$

que é a equação (3.1).

Na formulação proposta P e Q representam na verdade a carga total na barra r, que geralmente estão constituídas pela carga da própria barra r, as cargas de todas as barras que estão "na frente" da barra r e as perdas de todos os ramos que estão na direção da carga a partir da barra r. Em outras palavras, P e Q é um equivalente exato da rede conectada à barra r.

Na formulação do algoritmo também é necessário calcular as perdas ativas e reativas em cada ramo. As perdas ativa e reativa em cada ramo são facilmente deduzidas na seguinte forma:

Perdas Ativas:

$$P_e = I^2 R$$

mas como $I^* = \frac{P + jQ}{V_r \angle \theta_r}$, então $I^2 = \frac{(P^2 + Q^2)}{V_r^2}$

e assim temos:

$$P_e = R \frac{(P^2 + Q^2)}{V_r^2} \quad (3.6)$$

Perdas Reativas:

De maneira similar, pode-se demonstrar que:

$$Q_e = X \frac{(P^2 + Q^2)}{V_r^2} \quad (3.7)$$

3.1.3. Algoritmo Proposto

O algoritmo proposto em [17] é o seguinte:

- 1 - Ler os dados da rede incluindo os parâmetros, descrição topológica, módulo de tensão nas barras fonte e as cargas para tensão nominal.
- 2 - Assumir um módulo de tensão para cada barra que será a primeira estimativa dos módulos de tensão de barra que se deseja calcular.
- 3 - Calcular o equivalente para cada barra somando todas as cargas da parte da rede que são alimentados através da barra que está sendo analisado incluindo as perdas. Assim está sendo implementada uma iteração "para cima" a partir das barras extremas para as barras fontes.
- 4 - A partir da barra fonte (ou barras fontes) e usando a equação (3.1) calcular a tensão de carga V_r para todas as barras. Esta é uma iteração "para baixo" iniciando da barra fonte (ou barras fontes) para as barras extremas.
- 5 - Com as novas tensões recalculadas, deve-se recalculas as perdas. Se a variação das perdas totais em relação à iteração prévia é maior que uma tolerância especificada então voltar ao passo (3). Em caso contrário, calcular todos os outros resultados necessários e parar.

3.1.4. Comentários Adicionais sobre o Algoritmo de Fluxo de Carga.

Pode-se verificar que o algoritmo analisado é relativamente simples na sua formulação matemática. Entretanto, a implementação computacional exige uma enumeração adequada das barras e dos ramos como é proposto por Cespedes em [17].

Este algoritmo, conhecido também como método de soma de potência, foi implementado computacionalmente e está sendo usado como uma subrotina na resolução do problema de alocação ótima de capacitores.



CAPÍTULO 4

O Algoritmo Genético

4.1. Introdução

4.1.1. Técnicas de Buscas para Soluções de Problemas de Otimização

Diferentes técnicas de buscas para solução de problemas de Otimização, foram propostas por diversos pesquisadores. Elas podem ser agrupadas em três grandes classes: Técnicas Tradicionais de Otimização (Gradiente, Newton, Fibonacci), Busca Aleatória Direcionada e Técnicas Enumerativas. As ramificações destas três importantes técnicas são apresentadas no diagrama de blocos, mostrado na figura 4.1

4.1.2. Técnicas Tradicionais de Otimização

As técnicas tradicionais de otimização (técnicas de cálculo-fixo) usam um conjunto de condições necessárias e suficientes para serem satisfeitas pelas soluções de um problema de otimização. Estas técnicas subdividem-se em métodos diretos e indiretos.

Métodos indiretos buscam por extremos locais resolvendo conjuntos geralmente não lineares de equações resultantes da fixação do gradiente da função objetivo igual a zero.



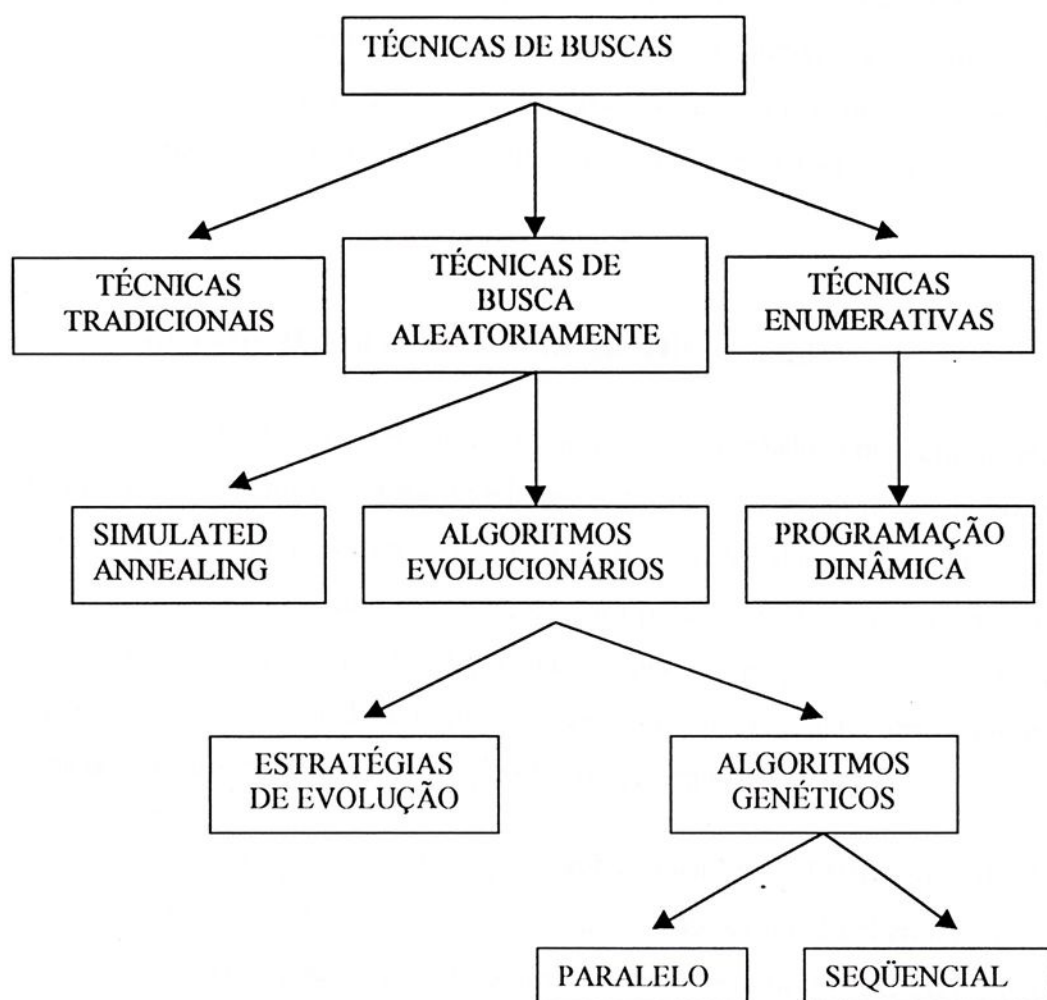


Figura 4.1 : Classificação das Técnicas de Buscas

Métodos diretos, têm como exemplo os métodos de Newton e Fibonacci, procuram o extremo movimentando-se no espaço de busca e assegurando o gradiente do novo ponto, que direciona a busca, satisfazendo as condições de factibilidade e de melhoria da função objetivo. Estas técnicas podem ser usadas apenas em um conjunto restrito de problemas bem comportados.

4.1.3. Técnicas Enumerativas

As técnicas enumerativas buscam um ponto por vez, considerando-se um conjunto, normalmente definido em um período de tempo, de uma função objetivo e um conjunto de restrições amplamente utilizadas em diversas aplicações, são simples para implementar, mas podem exigir um elevado tempo computacional. A programação dinâmica é um bom exemplo de técnica enumerativa.

4.1.4 Técnicas de Busca Aleatoriamente Direcionada

As técnicas de busca aleatória direcionada são baseadas em técnicas enumerativas, mas usando informações adicionais para direcionar a busca. São geralmente muito competentes podendo resolver vários problemas complexos. As duas maiores subclasses são *Simulated Annealing* e *Algoritmo Evolucionário*. *Simulated Annealing* usa um processo de evolução termodinâmico para buscas de estado de energia mínimos. Os *Algoritmos Evolucionários*, por outro lado, são baseados em princípios de seleção natural. Esta forma de busca, através das gerações, melhora traços de soluções potenciais por meio de operações inspiradas biologicamente.

Os Algoritmos Evolucionários se subdividem em: Estratégias de Evolução e Algoritmo Genético. As estratégias de evolução foram propostas por Rechenberg e Schwefel no começo da década de 70 e podem adaptar o processo de “evolução artificial”, com base em informações referentes ao comportamento ótimo da função objetivo, possibilitando a busca de soluções ótimas.

O *Algoritmo Genético* foi originalmente proposto por Holland e a partir deste algoritmo básico muitas variações têm sido introduzidas. John Holland, em meados dos anos 70, acreditou que as peculiaridades da Evolução Natural poderiam ser implementadas de forma algorítmica a fim de alcançar uma versão computacional dos processos de evolução. Este algoritmo poderia solucionar qualquer problema que apresentasse as mesmas características da evolução. O sistema trabalhava com uma população de algumas cadeias de bits (0's e 1's) denominadas



cromossomos. Os Algoritmos Genéticos são bastante adequados para implementação em máquinas de memória distribuída, em decorrência da própria estrutura deles.

4.2. Algoritmo Genético: Teoria

Os *Algoritmos Genéticos* têm se apresentado como técnicas de otimização robusta, um modelo computacional que imita a evolução natural das espécies, combinando os conceitos de adaptação seletiva e sobrevivência dos indivíduos mais capazes. Os algoritmos genéticos utilizam mecanismos para buscar melhores soluções para os problemas baseados na teoria da genética e evolução natural. Na natureza os indivíduos melhor dotados, isto é, indivíduos que possuem um conteúdo genético de excelente qualidade, têm maior capacidade de sobrevivência quando os recursos se tornam escassos ou quando muda o meio ambiente.

Proposto inicialmente por John Holland no começo da década de 70, os algoritmos genéticos são programas de computador que imitando a evolução natural resolvem problemas em uma extensa variedade de domínios. Holland visava expandir o processo adaptativo de sistemas naturais e desenvolver sistemas computacionais envolvendo esses importantes mecanismos. O espetacular crescimento do Algoritmo Genético se deve inicialmente a Holland ao lado de Goldberg, Dejong, Grefenstette, Davis, Mühlern e outros. O interesse em algoritmos de pesquisa heurística com embasamento em processos naturais e físicos reacendeu através de Kirkpatrick, Gellat e com a técnica Simulated Annealing de Vicck's em 1983. Essa técnica Simulated Annealing é baseada em considerações termodinâmicas e surgiu em analogia algorítmica com o Processo de Annealing em sólidos, enquanto Estratégias de Evolução e Algoritmos Genéticos vêm da inspiração natural das espécies e no processo de seleção.

Simulated Annealing, Algoritmo Genético e Estratégia de Evolução são similares em seus usos pelo mecanismo de busca probabilística atuando diretamente no aumento ou diminuição do custo, com grande probabilidade de alcançar o ótimo global. Cada método têm entretanto diferente significado na forma de operação.



Simulated Annealing - gera probabilisticamente uma seqüência de transições controlada por um programa de resfriamento para finalmente convergir para o ótimo global.

Estratégia de Evolução - usa mutação como mecanismo de pesquisa e seleção para direcionar a busca a regiões interessantes no espaço de busca.

Algoritmos Genéticos - são técnicas de busca estatística baseadas nos mecanismos de evolução das espécies na natureza, geram uma seqüência de populações usando mecanismos de seleção, e utilizam a recombinação e mutação como mecanismos de busca. Algoritmo genético imita as teorias de evolução biológica para resolver problemas de otimização. Abrange um conjunto de indivíduos (a população) e um conjunto de operadores inspirados biologicamente definido sobre a própria população, consistindo em seleção natural, recombinação e mutação e consta de um programa de controle formado pelas taxas de recombinação (**tr**) , mutação (**tm**) e o tamanho da população (**tp**).

Os Algoritmos Genéticos trabalham com representações codificadas das soluções, equivalente ao material genético dos indivíduos na natureza. Para que o Algoritmo Genético represente adequadamente uma configuração do problema, deve-se optar por uma codificação adequada às características do problema. Originalmente, foi muito utilizada a codificação binária, pois nesta codificação é fácil de implementar os operadores recombinação e mutação e também porque nesta codificação existem provas de convergência que justificam matematicamente as propriedades de convergência do Algoritmo Genético.

4.2.1. Algoritmo Genético e Evolução Natural

A capacidade de se reproduzir é o principal fator de evolução dos seres vivos. Por evolução entende-se a adaptação de um indivíduo ao meio ambiente. A adaptação por reprodução implica-se no fato dos seres vivos modificarem-se a medida que vão se reproduzindo (Darwinismo). A adaptação não ocorre de forma direta, ou seja, os indivíduos se modificam de acordo com as necessidades e essa modificação passa diretamente para os descendentes (Lamarckismo). O princípio



adotado pelos AG's é o processo de evolução por seleção natural, ou seja, os indivíduos mais adaptados ao meio ambiente têm mais chances de sobreviver. Evolução biológica sabe como produzir seres vivos que se adaptam melhor ao meio ambiente. Este conhecimento está armazenado no código genético do indivíduo, responsáveis pelas características do indivíduo. As mudanças no material genético constitui a essência da evolução, portanto a evolução genética é consequência da ação conjunta de seleção natural e recombinação genética que acontece como parte da reprodução. A evolução é iniciada com a recombinação genética que acontece na formação dos gametas masculinos e femininos. A troca de material genético entre cromossomos é chamado recombinação (crossover ou crossing over), produzindo uma recombinação de genes.

4.2.2. Funcionamento

O Algoritmo Genético utiliza uma população de indivíduos (configurações) identificadas pelo cromossomo para resolver um problema complexo. As soluções de qualquer problema precisam estar codificadas nos cromossomos. O grau máximo de evolução é encontrar cromossomos que efetivamente resolvessem o problema. Para representar adequadamente uma configuração do problema é usado codificação binária onde podem ser facilmente simulados os operadores de recombinação e mutação. A função de adaptação deve testar cada cromossomo, verificando se ele resolve adequadamente o problema. Deve-se encontrar uma forma adequada para avaliar a função objetivo ou seu equivalente. Assim é possível identificar as configurações de melhor qualidade como sendo aquelas que apresentam uma função objetivo de melhor qualidade (fitness value). A função de avaliação é considerada como elo entre o AG e o problema a ser resolvido. Esta função retorna um valor ou uma lista de valores que apresentarão o desempenho do indivíduo frente ao problema. Deve-se ter uma estratégia de seleção das configurações com direito a participar na conformação dos elementos da nova população. Deve existir um mecanismo para realizar a recombinação e um mecanismo para realizar a mutação que é um elemento secundário na evolução natural e produz a recuperação do material genético perdido. Também é preciso definir o tamanho da população (número de configurações).



4.3. Algoritmo Genético Simples (AGS)

O Algoritmo Genético de Holland é comumente chamado de *Algoritmo Genético Simples* ou *AGS*. Trabalha com um conjunto de indivíduos, “strings” de bits que constitui a população, onde cada um dos “bits” 0 ou 1 é a versão codificada de uma solução para problemas de otimização. Usando os operadores genéticos - **recombinação** e **mutação**- o Algoritmo Genético cria a geração seguinte através de “strings” a partir da população atual.

A estrutura de um algoritmo genético simples é igual à estrutura de qualquer programa de evolução, como visto na figura 4.2.

Procedimento do programa de evolução

Início

$t \leftarrow 0$

iniciar $P(t)$

avaliar $P(t)$

enquanto (não termina condição) **faça**

início

$t \leftarrow t+1$

selecionar $P(t)$ de $P(t-1)$

alterar $P(t)$

avaliar $P(t)$

fim

Fig. 4.2 Estrutura de um programa de evolução

Durante a repetição (t), um algoritmo genético mantém uma população de soluções potenciais (cromossomos, vetores) $P(t) = \{x'_1, \dots, x'_n\}$. Cada solução x'_i é avaliada para dar alguma medida de sua adaptação. Então, uma nova população (repetição $t + 1$) é formada selecionando indivíduos melhores ajustados. Alguns elementos desta nova população sofrem alterações por meio de recombinação e mutação para formar novas soluções. Recombinação – combina as características de dois cromossomos de para formar dois descendentes semelhantes trocando segmentos correspondentes dos cromossomos. A aplicabilidade do operador recombinação é a troca de informação entre diferentes soluções potenciais. Mutação – altera um ou mais genes de um cromossomo selecionado arbitrariamente por uma mudança fortuita com uma probabilidade igual à taxa de mutação. A mutação é a introdução de uma variabilidade extra na população.

Pode-se fazer um paralelo entre algoritmo genético e a teoria da evolução onde somente os indivíduos mais capazes de uma população sobrevivem, transmitindo desta forma o material genético destes indivíduos sobreviventes para seus descendentes. Analisam um conjunto de soluções potenciais devidamente codificadas (codificação binária), que constitui a população. Manipulam os indivíduos mais aptos, utilizando-se dos operadores genéticos, para a obtenção de uma população otimizada. Esse ciclo geracional é repetido até que um critério de parada seja atingido.

4.3.1. Componentes e Estruturas de um AGS

A estrutura de um Algoritmo Genético Simples deverá ter definida a população inicial, seu mecanismo de codificação, os parâmetros de controle a serem utilizados, a avaliação da função objetivo, o mecanismo de seleção, os operadores genéticos (recombinação e mutação) e o critério de parada. Os Algoritmos Genéticos são muito sensíveis a estes parâmetros. Um Algoritmo Genético Simples funciona como segue:

1. Gerar uma população inicial após escolher a forma de codificação;
2. Calcular a função objetivo de cada configuração e armazenar a incumbente;



3. Implementar a seleção;
4. Implementar a recombinação;
5. Implementar a mutação e terminar de gerar a nova geração;
6. Se o critério de parada for satisfeito pára, caso contrário voltar para 2.

Os passos 2 a 5 são chamados de Ciclo Geracional do Algoritmo Genético, representado na Figura 4.3



Figura 4.3: Ciclo Geracional do Algoritmo Genético

4.4. Codificação e Avaliação da Função Objetivo

O mecanismo de codificação, fundamental para a estrutura do Algoritmo Genético, depende da natureza das variáveis do problema de otimização. Existem problemas com variáveis binárias (os mais simples de implementar), outros têm variáveis reais, discretas (inteiras). Um método comum para codificá-las é usar sua representatividade inteira. Os primeiros Algoritmos Genéticos usaram a codificação binária, isto é, as variáveis inteiras e reais eram transformadas a uma forma de representação binária.

Definida a codificação que será utilizada, deve-se ter disponível uma estratégia para calcular a função objetivo para cada configuração. A função objetivo, função a ser otimizada, fornece o mecanismo para avaliar cada "string", contudo seu intervalo de valores varia de problema para problema. Para manter uniformidade sobre o domínio de vários problemas, geralmente torna-se necessário normalizar a função objetivo dentro de determinados parâmetros. Usamos a função de adaptação "*fitness function*" mecanismo que permite avaliar cada indivíduo da população, normalizando a função objetivo a um intervalo conveniente 0 – 1. O valor normalizado da F.O. é a "*adaptação do string*", que o mecanismo de seleção usa para avaliar cada string da população. Baseado na adaptação de cada um, o mecanismo de seleção escolhe os indivíduos que farão pares no processo de manipulação genética.

4.5. Discussão das Funções Usadas pelo Algoritmo Genético.

4.5.1. Seleção

A seleção na natureza, indica a sobrevivência dos indivíduos mais adaptados. Análogo ao mecanismo biológico, as soluções mais adaptadas sobrevivem enquanto as mais fracas perecem. Nos AGS um "string" bem adaptado possui um grande número de descendentes. Tipicamente, para selecionar os indivíduos é usado o chamado *esquema de seleção proporcional*. Nesta estratégia, determina-se o número de descendentes para cada configuração usando a seguinte relação:



$$Nd_i = n \frac{Z_i(x)}{\sum_{i=1}^n Z_i(x)}$$

onde $Z_i(x)$ é o valor da função objetivo da configuração i na população de tamanho n . Assim, Nd_i é proporcional ao valor da função objetivo. Como os valores dos Nd_i são não inteiros então, deve-se encontrar um número de descendentes inteiros usando a roleta. Com essa estratégia a seleção é realizada de forma aleatória designando a cada configuração um setor circular proporcional à sua função de adaptação.

Assim utiliza-se a seguinte relação:

$$2\pi(1/n)Nd \Leftrightarrow 360(1/n)Nd$$

n : tamanho da população

Nd : número de descendente de cada configuração

onde cada "string" é alocado em um setor ("slot") de uma roleta metálica com o ângulo descrito pelo setor do centro do disco igual a $2\pi(1/n)Nd$. A um "string" é alocado um descendente, se um número aleatoriamente gerado no intervalo 0 a 2π cai no setor correspondente ao "string". O algoritmo seleciona os "strings" desta maneira até que seja gerada toda a população para a próxima geração.

4.5.2. Outros Operadores Genéticos

O algoritmo genético usa operadores genéticos para produzir uma nova população de indivíduos (descendentes) através da manipulação da "informação genética" denominada como genes atribuídos aos pais (originais) da população atual. Geralmente são utilizados os operadores de recombinação em um ponto e mutação, devido às facilidades de implementação e da alta confiabilidade, que serão discutidos a seguir.

4.5.2.1 Recombinação

Uma vez selecionadas as configurações, deve-se escolher duas configurações aleatoriamente para realizar a recombinação. Na recombinação um par de configurações “strings” são “trocados” aleatoriamente dentre os elementos da população que estão sujeitos a recombinação, misturando seus conteúdos genéticos e formando dois descendentes. Existem três técnicas básicas de recombinação: recombinação de um simples ponto, recombinação de dois pontos e recombinação uniforme. O AGS utiliza a técnica de recombinação de um único ponto (*single point crossover*). Escolhe-se um número aleatório no intervalo 1 e (k-1) para uma configuração de k elementos. Uma vez escolhido o ponto de recombinação, assinala-se nas duas configurações candidatas, fazendo-se então a recombinação, que é a permutação da parcela à direita deste ponto de uma configuração, pela parcela à direita da outra, gerando as duas novas configurações. Observa-se que nem sempre todas as configurações são recombinadas. Uma parcela pequena pode passar para a próxima etapa sem recombinação. Somente são recombinadas configurações de acordo com uma taxa de recombinação ρ_c . Assim define-se a taxa de recombinação (ρ_c), que especifica de forma estatística, a porcentagem de configurações que devem ser recombinadas. Assim, o operador recombinação é semelhante ao realizado na natureza, onde os indivíduos possuem cromossomos e trocam parte das informações genéticas de dois deles, para produzirem dois novos indivíduos. Desta forma, para implementar recombinação o algoritmo a ser seguido é o seguinte:

- a. escolher as duas configurações candidatas a recombinação;
- b. gerar um número aleatório p no intervalo [0-1];
- c. se $p < \rho_c$, efetua-se a recombinação. Caso contrário, passar as duas configurações para a outra fase.

A Figura 4.4 mostra que, após o ponto de recombinação ter sido aleatoriamente escolhido, partes dos “strings” 1 e 2 são permutadas para produzir novos “strings”.



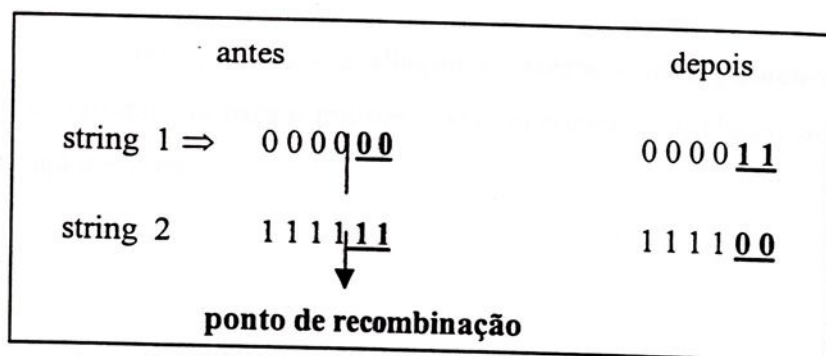


Figura 4.4: Recombinação

4.5.2.2. Mutação

Depois da recombinação os “strings” estão sujeitos à mutação. A mutação exerce uma influência especial para o Algoritmo Genético, pois permite sair de um mínimo local (convergência prematura). A necessidade da mutação diminui na fase final do procedimento quando as populações tornam-se mais homogêneas e dominadas pelos genes mais eficientes. Na maioria das pesquisas sobre Algoritmo Genético, a mutação é considerada um operador secundário. A taxa de mutação ρ_m indica a probabilidade de que uma posição (variável binária) deve ser mutada. Na formulação original a mutação é implementada bit por bit e a mutação de uma posição é independente da mutação das outras posições de uma configuração. Na codificação binária, a mutação de um “bit” consiste em transformar o valor de uma posição de 0 para 1, ou vice-versa. A Figura 4.5 mostra o operador mutação aplicado para o 4º elemento do “string”.

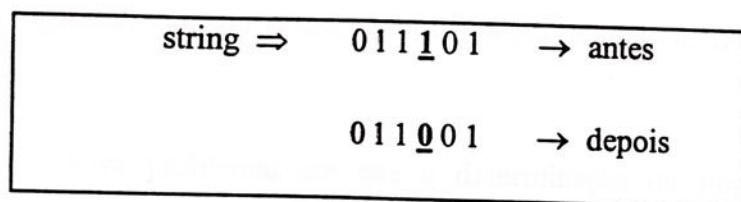


Figura 4.5: Mutação

O ciclo **criação – avaliação – seleção – manipulação genética** se repete até que uma solução satisfatória para o problema seja encontrada ou algum outro critério de parada para o algoritmo seja atendido.

4.6. Ciclo Geracional

O processo seleção – recombinação – mutação é que permite encontrar as configurações da nova geração a partir das configurações (população) da geração corrente. O ciclo geracional é controlado pelo Programa de Controle do Algoritmo Genético, um conjunto de parâmetros que define o tamanho da população, a taxa de recombinação e a taxa de mutação. Variando de problema para problema e em grande medida, define a qualidade do Algoritmo. Valores típicos usados são:

Número de elementos das populações:	η_e [30 200]
Taxa de cruzamento:	ρ_c [0,5 1,0]
Taxa de mutação	ρ_m [0,001 0,05]

4.7. Critério de Parada

Existem vários critérios de parada do algoritmo genético. Pode-se adotar um critério de parada quando foi realizado um número determinado de gerações, ou quando a solução incumbente encontrou um valor especificado, ou ainda quando as configurações da população ficam muito homogêneas. Pode-se utilizar, também, a combinação de dois destes critérios, ou mesmo dos três.

Para problemas em que a determinação da função objetivo exige um esforço computacional elevado e geralmente muito maior que o restante do algoritmo genético é mais

adequado escolher um critério de parada que depende do número de avaliações de função objetivo. Assim, por exemplo, no algoritmo para o problema de alocação de capacitores em sistemas de distribuição radial, encontrar o valor da função objetivo de uma configuração implica resolver 3 problemas de fluxo de carga radial. Portanto, um critério de parada coerente é parar o processo após processar um número máximo de chamadas do algoritmo de fluxo de carga. Este critério pode ser combinado com algum dos critérios mencionados anteriormente.

4.8 Funções Avançadas do Algoritmo Genético

4.8.1. Teorema do "Schema"

As bases teóricas dos AGs foram desenvolvidas para uma representação binária de strings e na noção de esquema [19].

Um esquema representa um subconjunto de todas as configurações possíveis de uma codificação binária de configurações e que tem parte de suas configurações fixadas. Tem-se como exemplo, uma codificação binária com cinco posições que permite representar 32 configurações diferentes. Portanto o esquema $* * 0 0 0$ é um subconjunto dessas 32 configurações e que tem posições fixas. Assim, o esquema representa as seguintes configurações:

$$* * 0 0 0 \left\{ \begin{array}{l} 00\ 000 \\ 01\ 000 \\ 10\ 000 \\ 11\ 000 \end{array} \right.$$

Cada configuração representada por um esquema é uma representação (instância) do schema. Assim, um esquema representa um conjunto de configurações que possuem entre elas algumas características comuns que são as posições fixas. Essas características comuns são denominadas *Similarite Template* (conjunto de características ou propriedades similares). Portanto

um esquema representa um subespaço do espaço de configurações e quanto maior for o número de * maior o subespaço representado pelo esquema. Logo, o esquema (* * * *) representa o espaço completo de configurações. Um esquema representa, portanto, um subconjunto de todos os strings possíveis que têm os mesmos “bits” em certas posições dos strings.

Um esquema * * 0 0 0 representa “strings” com zeros nas três últimas posições. Cada string representado por um esquema é chamado *elemento do esquema*. Define-se como a *ordem de um esquema*, o número de posições fixas do esquema, isto é, as posições dos strings que têm 0 ou 1. Tem-se por exemplo 1 * * * 1 0 a representação de um esquema de ordem três.

Comprimento definido (schema defining length) de um esquema é a distância entre as posições fixas mais extremas. O esquema, * * 0 1 0 possui defining length de 2.

Função de adaptação média de um esquema (average fitness of the schema) é o valor médio das funções objetivos ou o seu equivalente de todas as configurações representadas pelo esquema.

Esquema de competição (schema de competition); considerando um esquema com r posições fixadas como sendo * 0 1 em que $r = 2$ (duas posições fixadas), existem portanto, $2^r - 1 = 2^2 - 1 = 3$ outros esquemas com as mesmas posições fixadas que podem ser obtidas permutando os valores 0 e 1 das posições fixadas. Para uma configuração de k configurações binárias é possível construir 2^k esquemas de competição. Assim, para $k = 3$, configurações com três posições, existem $2^3 = 8$ esquemas de competição.

O Algoritmo Genético gera esquemas de competição e tenta analisar os 2^k esquemas de competição e encontrar o melhor esquema para cada conjunto de posições fixadas. Assim um algoritmo genético produz uma competição entre esquemas e a configuração ótima deve aparecer como consequência da sobreposição de esquemas com valor da função objetivo de grande qualidade e de pequeno comprimento de definição. Esse tipo de esquema é chamado de bloco construtivo (building blocks).

4.8.2. Conceito de Building Blocks

Bloco construtivo é o esquema que possui um valor médio da função objetivo de excelente qualidade e também pequeno comprimento de definição. Portanto, um bloco construtivo é um esquema de excelente qualidade e ainda mais, essa informação está concentrada.

Os operadores de recombinação e mutação geram, selecionam, e fazem sobreposições de blocos construtivos para encontrar configurações de excelentes qualidades ou as ótimas. A recombinação tende a preservar a informação genética presente na configuração a ser cruzada. Assim, a recombinação de duas configurações parecidas perde capacidade de gerar blocos construtivos.

Deve-se considerar o efeito da seleção, recombinação e mutação para um esquema:

Seleção –favorece a formação de blocos construtivos ao escolher configurações de qualidade para rcombinação.

Recombinação – faz a sobreposição de blocos construtivos e esse operador é considerado como principal gerador de configurações com blocos construtivos de alta qualidade.

Mutação – gera novos blocos construtivos.

4.8.3 Modificações no Algoritmo Genético

4.8.3.1. Formas Alternativas de Implementar a Seleção

Uma forma de conseguir algoritmos genéticos avançados é desenvolvendo técnicas de seleção sofisticadas cuja característica fundamental é preservar a propriedade de seletividade durante todo o processo.

O esquema de seleção proporcional usado no algoritmo genético básico pode apresentar dois problemas: na fase inicial do processo, quando a maioria das configurações possui



valores da função objetivo de baixa qualidade, proporcionando o aparecimento de algumas superconfigurações com funções objetivo de excelente qualidade, eliminando desta forma, prematuramente, esquemas de outras configurações, eliminando também a diversidade, levando a uma convergência prematura do algoritmo. Na fase final do processo a maioria das configurações possui valores da função objetivo muito parecidos, e nestas condições o esquema de seleção proporcional perde a característica de seletividade

4.8.3.1.1. Mecanismo de Escalonamento (scaling mechanism)

A função objetivo de cada configuração é corrigida nesta estratégia, realizando um escalonamento da função objetivo usando a relação linear

$$f'' = af + b$$

f → valor da função objetivo de cada configuração

f'' → valor da função objetivo corrigida de cada configuração

$a - b$ → constantes calculadas de tal forma que o maior valor de f'' seja igual a um número de vezes do valor médio das funções objetivos da população (geralmente de 1.2 a 1.5 vezes). Nesta estratégia a seleção é implementada usando os valores de f'' .

4.8.3.1.2. Seleção Baseada em Ordenamento (rank – based selection)

Nesta estratégia o número de descendentes para cada configuração depende de uma seleção ordenada (ranking) das configurações em ordem decrescente da função objetivo. O número de descendentes de cada configuração varia linearmente com a localização no ranking. Esta estratégia, além de eficiente é muito rápida.

4.8.3.1.3. Seleção Baseada em Torneio (tournament selection)

Em um conjunto de configurações aleatoriamente selecionada, para que uma configuração seja escolhida para *seleção* deve ganhar uma competição (ser melhor). Assim, deve-se realizar k torneios para escolher as k configurações que devem participar da próxima geração

A vantagem desta proposta é não apresentar o problema de designar um número fracionário de descendentes pois a cada torneio é escolhida uma configuração.

4.8.3.1.4. Técnica de Seleção Estocástica do Resíduo (stochastic remainder technique)

Nesta estratégia, a cada configuração é designada, em forma determinística, a parte inteira do número de descendentes e pode ser usado o esquema de seleção da roleta somente para designar a parte fracionária.



CAPÍTULO 5

Algoritmo Genético Proposto

5.1 Introdução

Neste capítulo são apresentadas as principais características de um algoritmo genético aplicado ao problema de alocação de capacitores fixos e variáveis. Inicialmente são discutidas as características básicas do algoritmo genético proposto e suas diferenças em relação a outras propostas que usam algoritmos combinatoriais (simulated annealing, algoritmo genético, tabu search) para este tipo de problema. Também são apresentadas propostas diferentes em outras partes importantes do algoritmo tais como na técnica de redução de barras candidatas, na determinação da configuração inicial e no uso dos chamados indicadores de sensibilidade. Posteriormente, é apresentada a proposta do algoritmo genético com suas principais características e o algoritmo em forma simplificada.

O algoritmo foi integralmente implementado usando FORTRAN e foram realizados testes experimentais usando sistemas conhecidos na literatura especializada.



O modelo matemático do problema da alocação ótima de capacitores é um problema de programação não linear inteiro misto (PNLIM) com função objetivo não diferenciável e adicionalmente apresenta o fenômeno da explosão combinatorial. A resolução desse problema é muito difícil usando as técnicas tradicionais de otimização matemática. Entretanto, se existe uma proposta de solução (uma topologia proposta, isto é, uma proposta de alocação de capacitores em barras especificadas e com número de bancos especificados) então simplesmente através de resoluções de problemas de fluxo de carga radial é possível conhecer se essa proposta de solução é factível (não apresenta violações das restrições de operação) e a **qualidade** dessa proposta de solução pois a função objetivo pode ser facilmente calculada conhecendo as perdas em cada nível de carga. Essa é a filosofia básica dos algoritmos combinatoriais antes citados. Assim, o algoritmo genético inicia o processo de otimização a partir de um conjunto de configurações (população inicial) que pode ser obtida aleatoriamente ou usando algoritmos heurísticos construtivos simples e rápidos. Em cada iteração é obtido um novo conjunto de configurações (nova população) a partir da configuração corrente usando os operadores de seleção, recombinação e mutação. Em cada nova iteração são encontradas configurações de melhor qualidade e, eventualmente, nesse processo iterativo pode ser encontrada a solução (configuração) ótima global. Em outras palavras, um algoritmo genético realiza uma busca usando um conjunto de soluções (configurações) e através de um processo iterativo são encontradas novas configurações candidatas. Logicamente, o número de configurações visitadas nesse processo de busca deve ser um número muito reduzido de configurações do espaço de configurações e deve existir uma estratégia adequada para visitar as configurações mais atrativas. Neste capítulo é apresentada uma proposta de um algoritmo genético aplicado ao problema de alocação de capacitores em sistemas de distribuição radial.

5.2. Características Básicas do Algoritmo Genético para o Problema de Alocação de Capacitores Fixos

Apresenta-se, em forma separada, as principais características do algoritmo genético proposto e em cada caso é realizada uma análise comparativa com outras pesquisas que usam algoritmos combinatoriais para o problema de alocação de capacitores.



5.2.1. Representação e Codificação do Problema

No capítulo 2 é apresentada a forma matemática do problema de alocação de capacitores como sendo um problema de programação não linear inteiro misto. Entretanto, quando é especificada uma proposta de solução ou configuração então o problema se reduz apenas a verificar a factibilidade e a qualidade da solução proposta. A factibilidade é verificada através da resolução de um problema de fluxo de carga radial que, adicionalmente, fornece as perdas do sistema elétrico. Tipicamente o sistema elétrico é discretizado em 3 níveis de operação, portanto, analisar uma proposta de solução consiste em resolver 3 problemas de fluxo de carga radial.

A função objetivo, especificada no capítulo 2, pode ser facilmente encontrada conhecendo as perdas dos 3 níveis de carga obtidos na resolução de fluxo de carga radial e a proposta de solução ou configuração que especifica o número de barras com adição de bancos de capacitores e o número de bancos em cada barra. Entretanto, ainda deve ser especificada a codificação usada para representar uma proposta de solução ou configuração.

Uma proposta de solução para o problema de alocação de capacitores deve especificar as barras escolhidas para alocação de capacitores, o número de bancos de capacitores que deve ser alocado em cada barra selecionada e o número de bancos que devem estar operando em cada nível de carga. A literatura especializada analisa somente 3 níveis de carga. Assim, seja n_n o número de barras candidatas a alocação de capacitores, então uma configuração é representada por um vetor de tamanho $3 \times n_n$. Um elemento de uma configuração indica o número de bancos de capacitores que devem estar operando numa barra num determinado nível de operação.

As características do problema mostram que a codificação mais adequada para representar uma configuração do problema é a codificação decimal. Portanto, cada elemento de uma configuração é um número inteiro representando o número de bancos de capacitores que está operando numa barra para um determinado nível de carga.

Com a codificação decimal proposta, resulta simplesmente determinar a função objetivo de cada configuração proposta assim como implementar os operadores de recombinação e mutação.



5.2.2. Funções do Algoritmo Genético

5.2.2.1. População:

Em relação à população existem dois aspectos que devem ser especificados. O tamanho da população n_p e a forma em que é determinada a população inicial.

O tamanho da população deve ser escolhido levando em conta o tamanho do problema e os outros parâmetros genéticos como a taxa de recombinação e a taxa de mutação. No caso do problema de alocação de capacitores o tamanho do problema está relacionado com o número de barras do sistema ou o número de barras candidatas à alocação de capacitores.

A população inicial pode ser encontrada de várias formas diferentes. Aleatoriamente, controlada ou usando algoritmos heurísticos construtivos rápidos. Aleatoriamente significa escolher para cada barra candidata um número de bancos de capacitores entre zero e o número máximo de bancos permitidos. Uma estratégia deste tipo colocaria muitos bancos de capacitores em muitas barras candidatas. Tipicamente, em problemas de alocação de capacitores, as melhores soluções são obtidas com alocação de capacitores num número reduzido de barras. Portanto, uma estratégia aleatoriamente controlada é potencialmente mais atrativa. Nesta estratégia, para cada configuração, deve-se escolher primeiro um número reduzido de barras e somente nessas barras são alocados bancos de capacitores.

A proposta mais interessante consiste em encontrar a população inicial usando um algoritmo heurístico construtivo rápido. As pesquisas sobre a utilização de algoritmos combinatoriais aplicados a problemas de engenharia geralmente geram a população inicial com estratégias diferentes ao esquema aleatório. Assim, uma das propostas mais interessantes é gerar a população inicial usando um algoritmo construtivo rápido que permite encontrar configurações com as seguintes características: (1) as configurações obtidas são de boa qualidade e (2) as configurações obtidas podem ser diferentes representando configurações interessantes de regiões diferentes do espaço de configurações.



Na presente pesquisa é proposta a utilização de um algoritmo heurístico construtivo que é detalhado posteriormente e esta parte constitui a Fase I ou Estratégia de busca heurística do algoritmo genético proposto.

5.2.2.2. Seleção:

O processo de seleção consiste em determinar o número de descendentes que deve gerar cada uma das configurações candidatas da população corrente. A seleção é realizada levando em conta o valor da função objetivo, ou seu equivalente, das configurações da população corrente. Entretanto a seleção pode ser realizada de várias formas diferentes e constitui um dos operadores ou funções mais importantes do algoritmo genético.

Algumas das propostas mais importantes para implementar a seleção, apresentadas na bibliografia especializada, são as seguintes:

- Seleção proporcional.
- Seleção usando escalonamento.
- Seleção baseada em ordenamento.
- Seleção baseada em jogo.

A seleção proporcional determina para cada configuração um número de descendentes proporcional ao valor da função objetivo ou seu equivalente. A principal desvantagem da seleção proporcional é a perda de seletividade que acontece nos seguintes casos: (1) no início do processo podem aparecer as chamadas superconfigurações com excelente valor de função objetivo que produzem a eliminação prematura da maioria das outras configurações e, portanto, uma convergência prematura do processo, e (2) nas fases finais do processo os valores das funções objetivo se tornam muito parecidas e nesse contexto a seleção proporcional perde eficiência, isto é, se os valores das funções objetivos são muito parecidos então praticamente todas as configurações devem gerar um descendente e a seleção proporcional se torna pouco eficiente. A seleção

proporcional ainda apresenta uma outra desvantagem: o número de descendentes que designa a cada configuração não é inteiro e, portanto, deve-se terminar o processo de seleção com a ajuda de um outro mecanismo que pode ser, por exemplo, a seleção pela roleta ou equivalente.

A seleção usando escalonamento consiste em corrigir a função objetivo das configurações ajustando esses valores a uma função linear penalizando (diminuindo) os valores das funções objetivo com valores acima da média e incentivando (aumentando) os valores das funções objetivo que estão abaixo da média. Assim, são eliminadas as duas principais desvantagens existentes no esquema da seleção proporcional. Mas também neste caso o número de descendentes que corresponde a cada configuração não é inteiro e o processo de seleção pode ser terminado usando a roleta como no caso da seleção proporcional.

A seleção baseada em ordenamento é conceitualmente diferente aos anteriores pois nesta estratégia os valores das funções objetivo das configurações são usadas apenas para realizar um ordenamento. Depois a seleção é realizada levando em conta apenas a posição da configuração no ordenamento. Assim, a melhor configuração pode receber um número máximo de descendentes e as outras um número menor proporcional a sua posição no ordenamento, isto é, é realizada uma seleção usando uma função linear levando em conta simplesmente o ordenamento. Esta estratégia também elimina os problemas da seleção proporcional mas também o número de descendentes de cada configuração não é um número inteiro e, portanto, o processo de seleção deve ser terminado usando a roleta ou equivalente.

A seleção baseada em jogo também não leva em conta o módulo da função objetivo das configurações sendo importante apenas os valores relativos das configurações. A seleção é realizada em n_p jogos para uma população de tamanho n_p . Em cada jogo são escolhidas aleatoriamente n_k configurações que devem participar do jogo sendo ganhadora a configuração que possui a função objetivo de melhor qualidade. Assim depois dos n_p jogos o processo de seleção foi concluído. Logicamente, as melhores configurações devem receber um maior número de descendentes e devem ser as ganhadoras dos jogos em que são escolhidas aleatoriamente para participar. Esta estratégia elimina as desvantagens da seleção proporcional pelo aparecimento de super configurações (início do processo) ou configurações uniformes (final do processo) e também



o problema de todas as propostas anteriores devido a um número de descendentes não inteiro. A seleção baseada em jogo encontra um número inteiro de descendentes para cada configuração da população corrente.

5.2.2.3. Recombinação:

Uma vez terminada a seleção, deve-se proceder à recombinação ou crossover. A recombinação consiste em escolher duas configurações com direito a gerar descendentes e proceder a uma recombinação entre essas configurações para produzir duas novas configurações candidatas. A recombinação pode ser realizado de várias formas diferentes:

- Recombinação simples ou num ponto.
- Recombinação em dois pontos.
- Recombinação multiponto.
- Recombinação uniforme.

A diferença entre todas as propostas de recombinação é apenas no número de pontos escolhidos para implementar a recombinação. Num extremo está a proposta conservativa de realizar recombinação escolhendo apenas um ponto de recombinação. Esta estratégia é conservativa pois apenas existe uma troca de esquemas ou blocos construtivos entre as configurações. Assim, todo o trabalho de geração, destruição ou melhoria de blocos construtivos, que são os esquemas de excelente qualidade, devem ser realizados pelo operador de mutação. No outro extremo está a recombinação uniforme em que cada elemento das configurações devem ser testados para implementar a recombinação. Esta estratégia é altamente destrutiva para os blocos construtivos que já foram geradas e torna o operador de mutação num operador secundário. Assim, a melhor estratégia deve ser uma recombinação multiponto que deve escolher um número de pontos proporcional ao número de elementos de uma configuração, tipicamente entre 5 a 10% de pontos de recombinação em relação ao número de elementos de uma configuração.



Associado com a recombinação existe a chamada taxa de recombinação de $\rho_c = 0,9$ significa que em torno de 90% das configurações devem ser submetidas a recombinação e as restantes devem passar diretamente a mutação, sem recombinação. A literatura especializada especifica valores de $\rho_c \in \{0,8 - 1,0\}$ como sendo o intervalo de valores mais adequado.

5.2.2.4. Mutação:

Uma vez implementada a recombinação, deve-se realizar a mutação nas configurações candidatas a constituir a nova população. A mutação, na codificação binária, consiste em escolher um elemento da configuração e trocar o valor corrente de 0 para 1 ou vice versa.

Para outros tipos de codificação geralmente é possível definir outras alternativas de mutação. Assim, por exemplo, na codificação decimal, pode-se definir uma mutação simples ou múltipla. Por exemplo, se uma configuração está na forma de codificação decimal e foi escolhida para mutação uma casa decimal com valor corrente igual a 4 então uma mutação simples produz uma mutação do valor 4 para 3 ou 5. De maneira similar, pode-se definir uma mutação dupla ou tripla que deve produzir uma variação maior do valor corrente de 4 para 6 ou 7 no caso de incremento ou para 2 ou 1 para o caso de decremento.

No problema de alocação de capacitores, na presente pesquisa é proposta a codificação decimal e podem ser implementadas mutações simples, duplas ou triplas do banco de capacitores existentes na barra escolhida para mutação. A melhor estratégia de mutação, verificada experimentalmente, deve ser finalmente escolhida para o problema de alocação de capacitores.

Associado com a mutação existe a chamada taxa de mutação ρ_m que determina o número de elementos de uma configuração que deve ser submetida a mutação. A literatura especializada recomenda valores de ρ_m no intervalo de $[0,001 - 0,05]$ mutações por bit em cada configuração. Para o problema de alocação de capacitores deve ser encontrado o melhor valor de



ρ_m obtido experimentalmente em conjunto com os outros parâmetros de controle do algoritmo genético.

5.2.3. Programa de Controle do Algoritmo Genético

O programa de controle do Algoritmo genético é especificado pela escolha ou determinação do tamanho da população, da taxa de recombinação, da taxa de mutação e o critério de parada.

Na pesquisa, inicialmente, escolhe-se parâmetros recomendados pela bibliografia especializada. Entretanto, para o problema de alocação de capacitores, o critério de parada deve ser especificado levando em conta outros critérios. No problema de alocação de capacitores, a parte do algoritmo que precisa de maior esforço computacional é a determinação da função objetivo que exige a resolução de 3 problemas de fluxo de carga radial para avaliar a função objetivo de uma configuração da população. Assim, o número máximo de problemas de fluxo de carga resolvidos deve fazer parte do critério de parada. Também deve ser armazenada a incumbente do processo assim como a geração em que foi encontrada para que seja possível comparar as diferentes versões do algoritmo genético proposto.

5.2.4. Outras Funções ou Estratégias Implementadas

Outras estratégias adicionais devem ser implementadas no algoritmo genético. A principal estratégia proposta consiste na geração da população inicial usando algoritmos heurísticos construtivos. A bibliografia básica de algoritmos genéticos sugere gerar a população inicial de maneira aleatória. Entretanto, pesquisas recentes de aplicação de algoritmos genéticos a problemas de engenharia sugerem e mostram resultados experimentais em que uma população inicial gerada usando estratégias heurísticas produzem resultados de melhor qualidade e com esforços computacionais menores. Assim, por exemplo, no problema de alocação de capacitores as



configurações mais atrativas apresentam adição de capacitores num número muito reduzido de barras. Uma geração aleatória de configurações produziria configurações com capacitores em muitas barras e geralmente em barras inadequadas.

Na presente pesquisa a população inicial é gerada usando algoritmos heurísticos construtivos simples baseados em sensibilidade. O processo de geração da população inicial usando um algoritmo heurístico construtivo também é usado para encontrar um número reduzido de barras candidatas a adição de capacitores. Assim, são apresentadas as duas estratégias separadamente.

5.2.5. Técnica de Redução de Barras Candidatas

A maioria dos algoritmos apresentados para resolver o problema de alocação de capacitores, especialmente aqueles baseados em algoritmos combinatoriais, apresenta uma proposta para reduzir o número de barras candidatas à adição de bancos de capacitores. A idéia de reduzir o número de barras candidatas é logicamente atenuar a explosão combinatorial do problema. Assim, reduzindo o número de barras candidatas, é reduzido o espaço de configurações facilitando a resolução do problema. Entretanto, este processo de redução do espaço de configurações (soluções) pode eliminar a solução ótima global do problema. Portanto, uma técnica eficiente de redução de barras candidatas à alocação de banco de capacitores deve reduzir adequadamente o espaço de configurações candidatas sem eliminar as configurações mais atrativas.

Em [3,4] são usados indicadores de sensibilidade para eliminar a maioria das barras do sistema como candidatas à adição de bancos de capacitores restando apenas um número reduzido de barras candidatas à adição de bancos de capacitores. Na verdade, praticamente todos os trabalhos sobre alocação de capacitores utilizam indicadores de sensibilidade para reduzir o número de barras candidatas. O indicador de sensibilidade é tipicamente uma relação matemática, exata ou aproximada, da variação das perdas do sistema com a variação de potência reativa em cada barra do sistema. Assim, para a configuração inicial, isto é, para a configuração sem alocação de capacitores é processado um fluxo de carga radial (geralmente para um nível de carga) e os resultados obtidos são usados para determinar um ranking das barras mais atrativas como sendo aquelas barras que



produzem uma maior diminuição de perdas do sistema quando são colocados reativos (bancos de capacitores) nessas barras. Essas barras atrativas são identificadas pelos indicadores de sensibilidade que são relações matemáticas do tipo $\partial P_L / \partial Q_i$.

Uma das limitações da estratégia anterior de seleção de um número reduzido de barras candidatas é que a seleção é realizada levando em conta apenas a configuração inicial, sem nenhuma alocação de bancos de capacitores e geralmente avaliando somente um nível de carga. Além disso, os indicadores de sensibilidade são relações matemáticas que fornecem apenas tendências aproximadas. Existem evidências de que essa estratégia pode eliminar barras potencialmente atrativas da lista de barras candidatas à adição de bancos de capacitores e, como consequência, elimina a configuração ótima do espaço reduzido de configurações candidatas.

Na presente pesquisa é implementada uma estratégia um pouco diferente e incorporando um número maior de barras como candidatas à adição de bancos de capacitores e essas barras são obtidas num processo dinâmico usando um algoritmo heurístico construtivo chamado de fase I, cujo objetivo é determinar as barras potencialmente atrativas e um número de configurações candidatas potencialmente atrativas e diferentes que depois são usados como parte da população inicial na fase II do Algoritmo. Este processo é apresentado depois como parte da fase I do algoritmo implementado.

5.2.6. Determinação da Configuração Inicial

Os algoritmos combinatoriais (genético e tabu search) apresentados em [3,4] geram a configuração inicial (ou o conjunto de configurações iniciais) de maneira aleatória. Esta estratégia, de gerar configurações iniciais aleatoriamente, está baseada em análises teóricas desses algoritmos que estipulam que “a qualidade da configuração final ou a melhor configuração encontrada não depende da configuração inicial de partida”. Na presente pesquisa é determinado um conjunto de configurações usando um algoritmo heurístico construtivo muito rápido. Essas configurações são geralmente atrativas e relativamente diferentes entre elas determinadas pela forma em que elas são geradas. As configurações assim obtidas fazem parte da população inicial e, pode-se implementar



recombinação, mutação ou distúrbios aleatórios em algumas dessas configurações para gerar outras configurações para completar a população inicial caso seja necessário. Esta estratégia faz parte do processo de fase I do algoritmo genético apresentado depois.

5.2.7. Uso de Indicadores de Sensibilidade

Os indicadores de sensibilidade, isto é, a variação das perdas ativas do sistema com respeito à variação da potência reativa numa barra, $\frac{\partial P_L}{\partial Q_i}$, é muito usado em problemas de sistemas de distribuição. Existem várias formas de encontrar esses indicadores de sensibilidade e no problema de alocação de capacitores são usados basicamente para reduzir o número de barras candidatas a alocação de capacitores diminuindo portanto o número de variáveis do problema. Assim, por exemplo, em [3] e [4] são usados diferentes indicadores de sensibilidade para reduzir o número de barras candidatas à alocação de capacitores, usando como base a configuração sem alocação de capacitores e o nível de carga elevado. A estratégia tem a vantagem de reduzir o número de barras candidatas mas apresenta a limitação de que freqüentemente pode excluir barras que fazem parte da configuração ótima especialmente quando esses indicadores são obtidos num único passo usando a configuração base e um único nível de carga.

Na presente pesquisa são usados indicadores de sensibilidade mas com estratégias diferentes. Assim, os indicadores de sensibilidade são usados para guiar o processo de alocação de capacitores num processo heurístico de fase I que consiste em obter um conjunto de configurações factíveis atrativas e significativamente diferentes e depois, na fase II, é implementado o algoritmo genético. Além dos indicadores de sensibilidade usados em [3,4] podem ser usados outros tais como o seguinte:

$$\frac{\partial P_L}{\partial Q_i} = 2\alpha_{ii}Q_i + 2\sum_{j \neq i}^n (\alpha_{ij}Q_j - \beta_{ij}P_j) \quad (5.1)$$

em que

$$\alpha_{ij} = r_{ij} \cos(\theta_i - \theta_j) / V_i V_j$$

$$\beta_{ij} = r_{ij} \sin(\theta_i - \theta_j) / V_i V_j$$

Na presente pesquisa, deve-se usar os indicadores de sensibilidade mostrados acima assim como os indicadores de sensibilidade usados em [3,4] e outros que eventualmente sejam identificados usando-os em forma integrada.

5.3. Algoritmo Genético Proposto

É proposto um algoritmo de duas fases em que a primeira fase corresponde a uma estratégia heurística e a segunda fase corresponde ao algoritmo genético. A seguir são apresentadas as duas estratégias separadamente e depois o algoritmo integrado proposto.

5.3.1. Fase I: Estratégia Heurística

A idéia básica é encontrar um conjunto de configurações factíveis, atrativas e significativamente diferentes que permita gerar uma boa população inicial antes que configurações geradas aleatoriamente. Esta idéia apresentou excelentes resultados num outro problema complexo de sistemas de potência [7].

Usando vários indicadores de sensibilidade, pode-se gerar as n_p configurações iniciais num processo heurístico construtivo de adicionar um banco de capacitores por vez até encontrar uma configuração factível e que apresente o menor valor da função objetivo. Em outras palavras, pode-se obter uma configuração atrativa adicionando em cada passo um banco de 300 kVar na barra considerada mais atrativa pelo indicador de sensibilidade e calculando em cada passo a função objetivo correspondente à adição dos capacitores e às perdas para todos os níveis de carga e terminando o processo quando já não é mais possível melhorar a função objetivo o que significa que foi encontrado um ótimo local. Pode-se repetir o processo anterior agora proibindo em cada caso a

inclusão de uma das barras usadas no processo anterior obtendo um conjunto diversificado de configurações iniciais atrativas e diferentes. Outra alternativa pode ser usar vários indicadores de sensibilidade e / ou escolher várias propostas de cada indicador de sensibilidade para encontrar um conjunto de configurações diferentes. Eventualmente, pode-se encontrar outras configurações iniciais usando uma espécie de “path relinking” (usado em tabu search) ou recombinação (usado no algoritmo genético) entre as configurações geradas.

A fase I termina com a geração da população inicial constituída de configurações atrativas e com uma lista de barras candidatas à alocação de capacitores. Ao contrário da maioria das estratégias de redução de barras candidatas apresentadas na literatura especializada que usam indicadores de sensibilidade usando a configuração base para encontrar um número reduzido de barras candidatas, na presente proposta as barras candidatas à alocação são todas aquelas barras que foram acionadas (usadas) na fase I e que aparecem em alguma das configurações iniciais. Deve-se observar que as duas estratégias não são equivalentes pois na presente proposta as barras escolhidas foram obtidas dentro de um processo dinâmico de geração de configurações.

5.3.2. Fase II: Algoritmo Genético

Na Fase II é implementado o algoritmo genético usando a população inicial encontrada na Fase I. Nesta fase são implementadas todas as funções e estratégias do algoritmo genético apresentado anteriormente.

5.3.3 Algoritmo Implementado

Fase I: Estratégia Heurística:



1. Realizar uma busca heurística usando indicadores de sensibilidade e o nível de carga mais elevado. Em cada passo, deve-se adicionar um banco de capacitores de 300 kVar até que a função objetivo não melhore (ótimo local). Supor que a configuração obtida possui k_n barras com alocação de capacitores.
2. Repetir o passo anterior realizando k_n buscas heurísticas e proibindo em cada busca a alocação de capacitores numa das barras usadas no passo (1).
3. Se o número de configurações candidatas n_p já foi encontrada então termina a Fase I. Em caso contrário encontrar as configurações restantes usando "path relinking".
4. Armazenar as n_e configurações de elite como sendo um subconjunto da população inicial, $n_e < n_p$.

Fase II: Algoritmo Genético:

1. Cálculo da função objetivo (ou equivalente):

Para a população corrente, encontrar a função objetivo de cada uma das configurações da população. Neste passo, para cada configuração da população, deve-se resolver 3 problemas de fluxo de carga radial. Atualizar a incubente se possível.

2. Seleção:

Usando um dos métodos de seleção propostos, implementar a seleção, isto é, encontrar o número de descendentes de cada configuração.

3. Recombinação:

Usando um dos tipos de recombinação propostos implementar a recombinação.

4. Mutação:

Usando um dos tipos de mutação propostos implementar a mutação. Assim foi obtida a nova população corrente.



4. Testar critérios de parada:

Se um dos critérios de parada é satisfeita então parar o processo. Em caso contrário voltar ao passo 1.

5.4. O Problema de Alocação de Capacitores Variáveis

O caso de alocação de capacitores variáveis é uma generalização do problema de alocação de capacitores fixos e representa a resolução de um problema mais complexo.

Definida uma configuração para o nível de carga elevado, deve-se encontrar a configuração (posição dos taps de capacitores variáveis) dos outros níveis de carga. Existem duas maneiras de implementar o problema de alocação de capacitores variáveis:

- Usando a mesma estratégia usada no caso de alocação de capacitores fixos. Neste caso, deve-se fazer duas modificações: (1) o tamanho de uma configuração neste caso deve ser de um tamanho 3 vezes maior que no caso de capacitores fixos, e (2) o processo deve levar em conta que os capacitores alocados nos níveis de carga inferiores devem ser menores ou iguais que os capacitores alocados no nível de carga elevado.
- Implementar o algoritmo genético somente para o nível de carga mais elevado e, para cada configuração, os capacitores alocados para os níveis de carga menores devem ser obtidos por uma estratégia heurística levando em conta a alocação de capacitores no nível de carga mais elevado. Este processo consiste em realizar basicamente adição, troca ou retirada de bancos a partir da configuração corrente e levando em conta o tipo de perturbação realizado no nível de carga elevada e os limites permitidos em carga elevada. Entretanto, é importante mencionar que é possível resolver esta parte do problema por métodos de otimização exata como foi realizado em [8] ou usando enumeração explícita para o caso em que a configuração analisada apresente poucas barras com alocação. Este problema é conhecido como “sizing problem” na literatura especializada.



Na presente pesquisa foi implementada a primeira alternativa levando em conta que a recombinação deve ser implementada de maneira adequada considerando as adições para os diferentes níveis de carga, assim como a mutação.



CAPÍTULO 6

TESTES E RESULTADOS OBTIDOS

Usando o algoritmo genético proposto no capítulo anterior e um algoritmo de fluxo de carga, foram realizados testes com três sistemas conhecidos na literatura especializada: um sistemas de 10 barras, um sistema de 34 barras e um sistema de 70 barras.



6.1. Sistema de 10 Barras

Este sistema apresenta 10 barras, tensão de 23KV e potência base de 100 MVA. Os dados completos deste sistema aparecem em [3] e encontram-se no apêndice A. São realizados testes usando os mesmos critérios usados no algoritmo genético proposto em [3]. Portanto, a função objetivo assume a seguinte forma:

$$V = K_e \sum_{i=0}^2 T_i P_i + K_p P_o + K_c \sum_{j=1}^{nc} C_j$$

para o caso de 3 níveis de carga e onde o nível de carga elevado é levado em conta duas vezes. Os parâmetros usados em [3] são os seguintes: $K_e = 0,06 \text{ \$/KWh}$, $K_p = 168 \text{ \$/KW/ano}$, $K_c = 4,9 \text{ \$/kVar}$, $S_0 = 1,1$, $S_1 = 0,6$, $S_2 = 0,3$, $T_0 = 1000$, $T_1 = 6760$, $T_2 = 1000$ horas e $\sum C_j$ é a capacidade total instalada (kVar) no sistema

Podem ser alocados até 5 bancos de 300 kVar em cada barra candidata e um horizonte de operação de um ano. Para o caso de operação sem compensação (sem alocação de bancos de capacitores) a função objetivo apresenta um custo de operação de $\text{US\$ } v = 329039$.

Para o caso de alocação de capacitores fixos o algoritmo genético encontra a configuração mostrada na tabela 6.1 com função objetivo $v = 308907,80 \text{ US\$}$.

Barra	Nº de bancos de 300 kVar
1	1
4	1
5	2
6	4

Tabela 6.1: Alocação de Bancos de Capacitores



Portanto, a alocação de capacitores produz um ganho de US\$ 20.131,2 em relação à operação sem alocação de capacitores, isto é, sem compensação.

A tabela 6.2 mostra os resultados obtidos após a aplicação de 25 testes para capacitores fixos considerando-se um tamanho de população de 100, taxa de recombinação de 0,9 e taxa de mutação de 0,050. Nos resultados apresentados na tabela 6.2 o algoritmo pára após executar 15 gerações, isto é, após processar 4500 problemas de fluxo de carga radial. N° ger indica a geração na qual foi encontrada a incumbente (melhor solução encontrada durante o processo)

N°	N° alea.	N° ger	Objetivo
1	2345	10	308907,80
2	3456	11	309907,80
3	4567	06	308907,80
4	5678	08	308907,80
5	1121	14	308907,80
6	1222	15	308963,80
7	1323	08	308907,80
8	1424	09	308907,80
9	1525	05	308907,80
10	2133	11	308907,80
11	2234	10	308907,80
12	2335	14	308963,80
13	2436	10	308907,80
14	2537	04	308907,80
15	3456	11	308907,80
16	3557	09	308907,80
17	3658	10	308907,80



18	4061	10	308907,80
19	4162	12	308907,80
20	4263	08	308907,80
21	4465	14	308907,80
22	6090	11	308907,80
23	6292	07	308907,80
24	6393	09	308907,80
25	6494	30	308907,20

Tabela 6.2: Capacitores Fixos - 10 barras

Os melhores resultados encontrados após a aplicação de vários testes para alocação de banco de capacitores fixos são apresentados na Tabela 6.3

1:	v = 308907,80	$n_1 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$
2:	v = 308963,80	$n_1 = 1$	$n_3 = 1$	$n_5 = 2$	$n_6 = 4$
3:	v = 309034,80	$n_1 = 1$	$n_4 = 2$	$n_5 = 1$	$n_6 = 4$
4:	v = 309041,20	$n_1 = 1$	$n_4 = 0$	$n_5 = 3$	$n_6 = 4$
5:	v = 309049,00	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$
6:	v = 309079,40	$n_1 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 3$
7:	v = 309091,00	$n_1 = 1$	$n_3 = 1$	$n_5 = 1$	$n_6 = 4$
8:	v = 309134,20	$n_1 = 1$	$n_3 = 1$	$n_5 = 2$	$n_6 = 3$
9:	v = 309154,30	$n_1 = 1$	$n_4 = 1$	$n_5 = 1$	$n_6 = 5$
10:	v = 309202,40	$n_1 = 1$	$n_4 = 1$	$n_5 = 3$	$n_6 = 3$

Tabela 6.3: As Melhores Configurações Encontradas



Os melhores resultados foram obtidos com tamanho da população em torno de 100, taxa de recombinação variando entre 0,5-0,9 e taxa de mutação em torno de 0,050.

Para o caso de alocação de capacitores variáveis, o algoritmo genético encontra a configuração mostrada na Tabela 6.4 com função objetivo $v = 307585,40$ US\$.

Barra	Nº de bancos para carga:		
	Elevada	Média	Baixa
1	1	0	0
2	1	1	0
4	1	0	1
5	2	2	0
6	4	4	3

Tabela 6.4: Alocação de Bancos de Capacitores Variáveis

Portanto, a alocação de capacitores produz um ganho de US\$21.453,60 em relação à operação sem alocação de capacitores, isto é, sem compensação.

A tabela 6.5 apresenta os resultados obtidos para alocação de capacitores variáveis - 10 barras, após a realização de 25 testes usando um tamanho de população de 100, taxa de recombinação de 0,90 e taxa de mutação de 0,020. Nos testes apresentados na tabela 6.5 o algoritmo para após executar 30 gerações, isto é, após processar 9000 problemas de fluxo de carga radial.

Nº	Nº alea.	Nº ger	Objetivo
1	1234	30	307585,40
2	2345	29	307633,40
3	3456	25	307587,00
4	1121	18	307587,00
5	1222	29	307631,10
6	1323	29	307624,80
7	1424	26	307608,20
8	1525	17	307608,20
9	2133	25	307587,00
10	2234	27	307652,50
11	2335	17	307652,60
12	2436	25	307652,60
13	3254	22	307587,00
14	3355	20	307609,70
15	3456	25	307587,00
16	3557	22	307585,40
17	3658	27	307663,00
18	4061	20	307587,00
19	4162	29	307591,90
20	4263	23	307587,00



21	4364	26	307601,80
22	4465	26	307587,00
23	6191	26	307587,00
24	6292	18	307587,00
25	6494	30	307608,20

Tabela 6.5: Capacitores Variáveis - 10 barras

As melhores configurações encontradas para capacitores variáveis após vários testes aplicados para diferentes valores do parâmetro de controle encontram-se na tabela 6.6

1:	v = 307585,40					
carga elevada:	$n_1 = 1$	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$	
carga média:	$n_1 = 0$	$n_2 = 1$	$n_4 = 0$	$n_5 = 2$	$n_6 = 4$	
carga baixa:	$n_1 = 0$	$n_2 = 0$	$n_4 = 1$	$n_5 = 0$	$n_6 = 3$	
2:	v = 307587,00					
carga elevada:	$n_1 = 1$	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$	
carga média:	$n_1 = 0$	$n_2 = 1$	$n_4 = 0$	$n_5 = 2$	$n_6 = 4$	
carga baixa:	$n_1 = 0$	$n_2 = 0$	$n_4 = 0$	$n_5 = 1$	$n_6 = 3$	
3:	v = 307590,30					
carga elevada:	$n_1 = 1$	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$	
carga média:	$n_1 = 0$	$n_2 = 1$	$n_4 = 0$	$n_5 = 2$	$n_6 = 4$	
carga baixa:	$n_1 = 0$	$n_2 = 0$	$n_4 = 1$	$n_5 = 0$	$n_6 = 2$	
4:	v = 307591,90					
carga elevada:	$n_1 = 1$	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$	
carga média:	$n_1 = 0$	$n_2 = 1$	$n_4 = 0$	$n_5 = 2$	$n_6 = 4$	
carga baixa:	$n_1 = 0$	$n_2 = 0$	$n_4 = 0$	$n_5 = 1$	$n_6 = 2$	

5:	v = 307596,90					
carga elevada:	$n_1 = 1$	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$	
carga média:	$n_1 = 0$	$n_2 = 1$	$n_4 = 0$	$n_5 = 2$	$n_6 = 4$	
carga baixa:	$n_1 = 0$	$n_2 = 0$	$n_4 = 1$	$n_5 = 1$	$n_6 = 2$	
6:	v = 307598,50					
carga elevada:	$n_1 = 1$	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$	
carga média:	$n_1 = 0$	$n_2 = 1$	$n_4 = 0$	$n_5 = 2$	$n_6 = 4$	
carga baixa:	$n_1 = 0$	$n_2 = 0$	$n_4 = 0$	$n_5 = 2$	$n_6 = 2$	
7:	v = 307599,00					
carga elevada:	$n_1 = 1$	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 5$	
carga média:	$n_1 = 0$	$n_2 = 1$	$n_4 = 0$	$n_5 = 2$	$n_6 = 5$	
carga baixa:	$n_1 = 0$	$n_2 = 0$	$n_4 = 1$	$n_5 = 0$	$n_6 = 3$	
8:	v = 307601,80					
carga elevada:	$n_1 = 1$	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$	
carga média:	$n_1 = 0$	$n_2 = 1$	$n_4 = 0$	$n_5 = 2$	$n_6 = 4$	
carga baixa:	$n_1 = 0$	$n_2 = 0$	$n_4 = 1$	$n_5 = 1$	$n_6 = 1$	
9:	v = 307608,20					
carga elevada:	$n_1 = 1$	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$	
carga média:	$n_1 = 0$	$n_2 = 1$	$n_4 = 1$	$n_5 = 1$	$n_6 = 4$	
carga baixa:	$n_1 = 0$	$n_2 = 0$	$n_4 = 1$	$n_5 = 0$	$n_6 = 3$	
10:	v = 307609,70					
carga elevada:	$n_1 = 1$	$n_2 = 1$	$n_4 = 1$	$n_5 = 2$	$n_6 = 4$	
carga média:	$n_1 = 0$	$n_2 = 1$	$n_4 = 1$	$n_5 = 1$	$n_6 = 4$	
carga baixa:	$n_1 = 0$	$n_2 = 0$	$n_4 = 0$	$n_5 = 1$	$n_6 = 3$	

Tabela 6.6: As Melhores Configurações Encontradas – Sistema de 10 barras.

6.2. Sistema de 34 Barras

Este sistema apresenta 34 barras, tensão de 11KV potência base de 100 MVA. Os dados deste sistema aparecem em [15] e são apresentados no apêndice A. Os parâmetros usados foram os mesmos usados em [2], isto é, um custo fixo de instalação de bancos de capacitores de US\$ 1000 dólares em cada barra usada e um custo de US\$900 dólares por cada banco de 300 kVar instalado. Os outros dados são: $K_e = 0,06\text{US\$} / \text{KWh}$, $S_0 = 1,8$, $S_1 = 1,0$, $S_2 = 0,5$, $T_0 = 1000$, $T_1 = 6760$, $T_2 = 1000$ e um horizonte de operação de um ano.

Para o caso de operação sem compensação, isto é, sem alocação de bancos de capacitores, a função objetivo apresenta um custo de US\$ 140.376,80.

Para o caso de alocação de capacitores fixos o algoritmo genético encontra a configuração mostrada na tabela 6.7 e cuja função objetivo é $v = 114.525,80 \text{ US\$}$.

Barra	Nºde bancos de 300 kVar
10	2
20	3
25	2

Tabela 6.7: Alocação de Banco de Capacitores Fixos.



A tabela 6.8 apresenta os melhores resultados encontrados após a aplicação de vários testes para a alocação de banco de capacitores fixos onde em média, em cada teste foram processados 9000 problemas de fluxo de carga radial.

1:	v = 114525,80	$n_{10} = 2$	$n_{20} = 3$	$n_{25} = 2$
2:	v = 114615,10	$n_{10} = 2$	$n_{21} = 3$	$n_{25} = 2$
3:	v = 114660,80	$n_{20} = 3$	$n_{25} = 2$	$n_{31} = 2$
4:	v = 114670,70	$n_{11} = 2$	$n_{19} = 2$	$n_{24} = 3$
5:	v = 114678,80	$n_{10} = 2$	$n_{20} = 2$	$n_{24} = 3$
6:	v = 114709,60	$n_9 = 3$	$n_{21} = 2$	$n_{25} = 2$
7:	v = 114718,50	$n_9 = 3$	$n_{20} = 3$	$n_{25} = 2$
8:	v = 114727,50	$n_{11} = 2$	$n_{20} = 2$	$n_{24} = 3$
9:	v = 114750,10	$n_{11} = 2$	$n_{19} = 3$	$n_{25} = 2$
10:	v = 114751,10	$n_{10} = 2$	$n_{18} = 2$	$n_{24} = 3$

Tabela 6.8: As Melhores Configurações Encontradas

Portanto, a alocação de capacitores produz um ganho de US\$ 25.851,00 em relação à operação sem alocação de capacitores, isto é, sem compensação.

Para o caso de alocação de capacitores variáveis, o algoritmo genético encontra a configuração mostrada na Tabela 6.9 com função objetivo $v = 113.845,10$ US\$.

Barra	Nº de bancos para carga		
	Elevada	Média	Baixa
9	3	3	2
21	3	3	1
25	2	2	1

Tabela 6.9: Alocação de Bancos de Capacitores Variáveis



A tabela 6.10 mostra os resultados obtidos após a aplicação de vários testes para capacitores variáveis onde em cada teste foram processados em torno de 18000 problemas de fluxo de carga radial.

1:	v = 113845,10	Barra	Nível de carga:		
			alta	média	baixa
		9	3	3	2
		21	3	3	1
2:	v = 113933,90	Barra	Nível de carga:		
			alta	média	baixa
		9	3	3	2
		20	2	2	0
3:	v = 113967,40	Barra	Nível de carga:		
			alta	média	baixa
		9	3	3	1
		19	2	2	2
4:	v = 113985,00	Barra	Nível de carga:		
			alta	média	baixa
		10	2	2	1
		19	3	3	3
5:	v = 113989,00	Barra	Nível de carga:		
			alta	média	baixa
		9	3	3	2
		21	3	3	1
		Barra	Nível de carga:		
			alta	média	baixa
		24	Nível de carga:		
			alta	média	baixa
		24	Nível de carga:		
			alta	média	baixa

6:	$v = 113997,10$	Barra	Nível de carga:		
			alta	média	baixa
		20	3	3	1
		25	3	2	2
7:	$v = 113998,20$	Barra	Nível de carga:		
			alta	média	baixa
		9	3	3	2
		21	3	3	1
8:	$v = 114001,60$	Barra	Nível de carga:		
			alta	média	baixa
		9	3	3	2
		21	3	3	1
9:	$v = 114003,10$	Barra	Nível de carga:		
			alta	média	baixa
		11	2	2	1
		18	3	3	2
10:	$v = 114008,90$	Barra	Nível de carga:		
			alta	média	baixa
		9	3	3	1
		19	2	2	1
		Barra	Nível de carga:		
			alta	média	baixa
		25	3	3	1

Tabela 6.10: Melhores Resultados para Capacitores Variáveis

Portanto, a alocação de capacitores produz um ganho de US\$ 26.531,7 em relação à operação sem alocação de capacitores, isto é, sem compensação.

6.3. Sistema de 70 Barras

Este sistema apresenta 70 barras, tensão de 12,66 KV e potência base de 100 MVA. Os dados completos deste sistema aparecem em [10] e são apresentados no apêndice A. Os parâmetros utilizados foram os mesmos usados em [2], isto é, um custo fixo de instalação de bancos de capacitores de US\$1000 em cada barra usada e um custo de US\$900 para cada banco de 300 kVar instalado. Os outros dados são: $K_c = 0,06$ US\$ /KWh, $S_0 = 1,8$, $S_1 = 1,0$, $S_2 = 0,5$, $T_0 = 1000$, $T_1 = 6760$, $T_2 = 1000$ e um horizonte de operação de um ano.

Para o caso de operação sem compensação, isto é, sem alocação de bancos de capacitores, a função objetivo apresenta um custo de US\$146.407.

Para o caso de alocação de capacitores fixos o algoritmo genético encontra a configuração mostrada na tabela 6.11 e cuja função objetivo é $v = 105.693,00$ US\$.

Barra original	Barra modificada	Nº de bancos de 300 kVar
17	19	1
50	62	4

Tabela 6.11: Alocação de Banco de Capacitores Fixos

Portanto, a alocação de capacitores produz um ganho de US\$ 40.714 em relação à operação sem alocação de banco de capacitores.



Os melhores resultados encontrados após a aplicação de vários testes para alocação de bancos de capacitores fixos são apresentados na tabela 6.12. Nos testes da tabela 8.12 o algoritmo para após executar 30 gerações, isto é, após executar, em média, 9000 problemas de fluxo de carga radial.

1:	v = 105693,00	barra original. 17 50	barra modificada 19 62	Nº de Bancos 1 4
2:	v = 105693,00	barra original. 16 50	barra modificada 18 62	Nº de Bancos 1 4
3:	v = 105706,40	barra original. 18 50	barra modificada 20 62	Nº de Bancos 1 4
4:	v = 105715,00	barra original. 19 50	barra modificada 21 62	Nº de Bancos 1 4
5:	v = 105729,50	barra original. 20 50	barra modificada 22 62	Nº de Bancos 1 4
6:	v = 105732,90	barra original. 21 50	barra modificada 23 62	Nº de Bancos 1 4
7:	v = 105740,80	barra original. 15 50	barra modificada 17 62	Nº de Bancos 1 4
8:	v = 105780,00	barra original. 14 50	barra modificada 16 62	Nº de Bancos 1 4
9:	v = 105857,30	barra original. 23 50	barra modificada 25 62	Nº de Bancos 1 4

10:	$v = 105990,40$	barra original.	barra modificada	Nº de Bancos
		13	15	1
		50	62	4

Tabela 6.12: As Melhores Configurações Encontradas para Capacitores Fixos

Para o caso de alocação de capacitores variáveis, o algoritmo genético encontra a configuração mostrada na tabela 6.13 com função objetivo $v = 103.516,20\text{US\$}$.

Barra	Barra	Nº de bancos para carga		
original	modificada	Elevada	Média	Baixa
17	19	1	1	1
50	62	6	4	2

Tabela 6.13: Alocação de Capacitores Variáveis

As melhores configurações encontradas para capacitores variáveis após vários testes aplicados para diferentes métodos de controle encontram-se na tabela 6.14. Nos testes da tabela 6.14 o algoritmo para após executar, em média, 18000 problemas de fluxo de carga.

1:	$v = 103516,20$	Nº de bancos em carga:			
	Barra original	Barra modificada	alta	média	baixa
	17	19	1	1	1
	50	62	6	4	2
2:	$v = 103564,60$				
	Barra original	Barra modificada	alta	média	baixa
	16	18	1	1	0
	50	62	6	4	2

3:	$v = 103582,80$				
	Barra original	Barra modificada	alta	média	baixa
	20	22	1	1	0
	50	62	6	4	2
4:	$v = 103686,30$				
	Barra original	Barra modificada	alta	média	baixa
	17	19	1	1	0
	50	62	6	4	3
5:	$v = 103693,00$				
	Barra original	Barra modificada	alta	média	baixa
	18	20	1	1	0
	50	62	6	4	3
6:	$v = 103783,60$				
	Barra original	Barra modificada	alta	média	baixa
	15	17	1	1	0
	50	62	6	4	3
7:	$v = 103742,00$				
	Barra original	Barra modificada	alta	média	baixa
	22	24	1	1	0
	50	62	6	4	3
8:	$v = 103983,30$				
	Barra original	Barra modificada	alta	média	baixa
	15	17	1	1	1
	51	63	6	4	2
9:	$v = 103984,10$				
	Barra original	Barra modificada	alta	média	baixa
	16	18	1	1	0
	51	63	6	4	2

10: $v = 103990,50$

Barra original	Barra modificada	alta	média	baixa
18	20	1	1	0
51	63	6	4	2

Tabela.6.14: As Melhores Configurações Encontradas

Portanto, a alocação de capacitores produz um ganho de US\$42.890,8 em relação à operação sem alocação de bancos de capacitores.

O algoritmo genético encontra configurações de melhor qualidade que as configurações encontradas pelo algoritmo genético apresentado em [3] para o sistema de 10 barras.

Quando comparado com o algoritmo TS apresentado em [16] o algoritmo genético proposto encontra as melhores configurações encontradas pelo algoritmo TS para o sistema de 10 barras (para alocação de capacitores fixos e variáveis) e de 70 barras (alocação de capacitores fixos e variáveis), assim como a melhor configuração para o sistema de 34 barras para o caso de capacitores fixos. Para o caso de capacitores variáveis, a melhor configuração encontrada pelo algoritmo genético apresenta um movimento de 44 US\$ superior (pior qualidade) que o algoritmo TS. Entretanto, para todos os testes, o esforço computacional do algoritmo genético é muito menor.



CAPÍTULO 7

CONCLUSÕES

Nesse trabalho foi apresentado um algoritmo genético para o problema de alocação de capacitores em sistemas de distribuição radiais a fim de reduzir as perdas de operação do sistema elétrico.

O algoritmo genético desenvolvido trabalha adequadamente e encontra excelentes configurações para os sistemas testados. Particularmente, o algoritmo encontra topologias de melhor qualidade que os encontrados pelo algoritmo genético apresentado em [3], para o sistema de 10 barras e realizando os testes para as mesmas condições de operação. O melhor desempenho do algoritmo apresentado, em relação ao algoritmo genético proposto em [3], deve-se provavelmente a dois motivos: (1) o Algoritmo genético desenvolvido na presente pesquisa usa operadores e funções de melhor qualidade assim como uma eficiente implementação da geração da população inicial, e (2) o algoritmo genético proposto em [3] usa a chamada técnica de redução de bancas candidatas que exclui algumas barras interessantes para alocação de bancos de capacitores. Assim, por exemplo, a barra 6 é excluída da lista de barras candidatas. Entretanto, na topologia encontrada por nosso algoritmo genético, a barra 6 é justamente a barra que recebe o maior suporte de reativos (bancos de capacitores).



Para os outros sistemas testados, o algoritmo genético proposto encontra configurações de boa qualidade. Entretanto, essas topologias são ligeiramente inferiores aos encontrados usando um algoritmo tabu search apresentado em [16] para o caso do sistema de 34 barras com capacitores variáveis, embora o esforço computacional do algoritmo genético seja significativamente menor que o algoritmo genético para todos os sistemas testados.

Nos sistemas de 34 e 70 barras não é possível fazer comparações com outras propostas tais como as apresentadas em [1] e [4] porque as condições de operação não são exatamente iguais.



REFERÊNCIAS BIBLIOGRÁFICAS

- [1] *H. D. Chiang, J. C. Wang, O. Cockings, H. D. Shin*: "Optimal Capacitor Placement in Distribution Systems Part I: A New Formulation and the Overall Problem", IEEE Transactions on Power Delivery, Vol. 5, No 2, pp 634-642, April 1990.
- [2] *H. D. Chiang, J. C. Wang, O. Cockings, H. D. Shin*: "Optimal Capacitor Placement in Distribution Systems Part II: Solution Algorithms and Numerical Results", IEEE Transactions on Power Delivery, Vol. 5, No 2, pp 643-649, April 1990.
- [3] *S. Sundhararajan, A. Pahwa*: "Optimal Selection of Capacitors For Radial Distribution Systems Using a Genetic Algorithm", IEEE Transactions on Power Systems, Vol. 9, No 3, pp 1499-1505, August 1994.



- [4] *Y. C. Huang, H. T. Yang, C. L. Huang*: "Solving the Capacitor Placement Problem in a Radial Distribution System Using Tabu Search Approach", IEEE Transactions on Power Systems, vol. 11, No 4, pp 1868 – 1873, November 1996.
- [5] *F. Glover*: "Tabu Search Fundamental and Uses", Graduate School of Business, University of Colorado, 1995.
- [6] *C. R. Reeves*: "Modern Heuristic Techniques for Combinatorial Problems", McGraw-Hill Book Company, 1995.
- [7] *R.A. Gallego, A. Monticelli, R. Romero*: "Comparative Studies of Non-Convex Optimization Methods for Transmission Network Expansion Planning", Annals of IEEE Power Industry Computer Application Conference, PICA, Columbus, Ohio, May 1997.
- [8] *M. E. Baran, F. F. Wu*: "Optimal Sizing of Capacitors Placed on a Radial Distribution System", IEEE Transactions on Power Delivery, Vol. 4, No 1, pp 735-743, January 1989.
- [9] *M. E. Baran, F. F. Wu*: "Network Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing", IEEE Transactions on Power Delivery, Vol. 4, No 2, pp 1401-1407, April 1989.
- [10] *M. E. Baran, F. F. Wu*: "Optimal Capacitors Placement on Radial Distribution System", IEEE Transactions on Power Delivery, Vol. 4, No 1, pp 725-733, January 1989.
- [11] *M. Kaplan* : "Optimization of Number, Location, Size, Control Type, and Control Setting of Shunt Capacitors on Radial Distribution Feeders", IEEE Transactions on Power Apparatus and Systems, Vol. PAS 103, No. 9, pp 2659-2663, September 1984.



- [12] *J. J. Grainger, S. Civanlar*: "Volt / Var Control on Distribution Systems with Lateral Branches Using Shunt Capacitors and Voltage Regulators Part I, II and III", IEEE Transactions on Power Apparatus and Systems, Vol. PAS 104, No. 11, pp 3278-3297, November 1985.
- [13] *J. J. Grainger, S. H. Lee*: "Capacite Release by Shunt Capacitor Placement on Distribution Feeders: A New Voltage-Dependent Model", IEEE Transactions on Power Apparatus and Systems, Vol. PAS 101, No. 15, pp 1236-1244, May 1982.
- [14] *J. L. Bala, P. A. Kuntz, M. J. Pebles*: "Optimal Capacitor Allocation Using a Distribution Recorder", IEEE Transactions on Power Delivery, Vol. 12, No 1, pp 464-469, January 1997.
- [15] *M. Chis, M. M. A. Salama, S. Jayaram*: "Capacitor Placement in Distribution System Using Heuristic Search Strategies", IEEE Proc. Gener. Transm. Distrib., Vol. 144, N° 3, pp. 225-230, 1997.
- [16] *W.C. Soares*: "Algoritmo de Busca Tabu usado no Problema de Alocação de Capacitores em Sistemas de Distribuição Radial", dissertação de Mestrado, DEE-FEIS-UNESP, outubro de 1999.
- [17] *Renato Cespedes G.*: "New Method for the Analysis of Distribution Networks", IEEE Transactions on Power Delivery, Vol.5, N°1, pp. 391-396, January 1990.
- [18] *F. Glover*: "Tabu Search Fundamental and Uses", University of Colorado, Boulder, Colorado, April 1995.
- [19] *Holland, J. H.*: "Adaptations in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, 1975



APÊNDICE A

SISTEMAS TESTADOS

Este apêndice apresenta os dados dos três sistemas conhecidos na literatura especializada e utilizados nos testes realizados para a resolução do algoritmo genético usado no problema de alocação de capacitores em sistemas de distribuição radial. Os dados das barras e das linhas estão representados em Tabelas. Em A.1. estão os dados completos do sistema de 10 barras [3], em A.2. estão os dados completos do sistema de 34 barras [16] e em A.3. estão os dados do sistema de 70 barras [7].



A.1. Dados Completos do Sistema de 10 Barras [3]

O sistema hipotético de 10 barras apresenta uma tensão base de 23 KV, potência base de 100 MVA. Seus dados foram obtidos da referência [3]. Abaixo está a Tabela 1 com os dados das barras e a Tabela 2 com os dados das linhas.

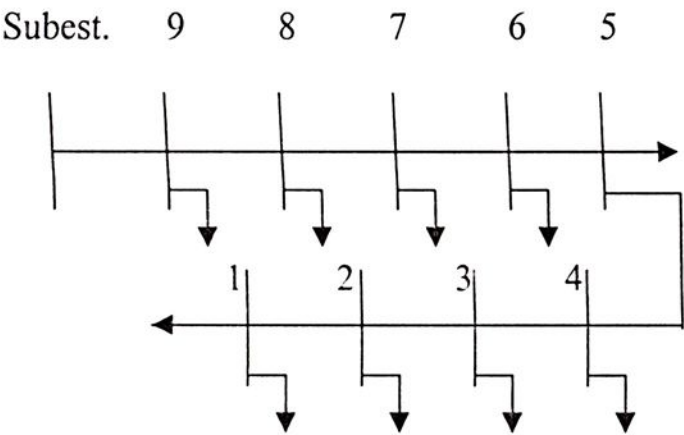


Fig. A.1.: Diagrama Unifilar do Sistema de Distribuição de 10 barras

Tabela A1: Dados da Referência [3]

DADOS DAS BARRAS

No. das Barras	Potência Ativa(kW)	Potência Reativa(kVAr)
1	1640	200
2	980	130
3	1150	60
4	780	110
5	1610	600
6	1598	1840
7	1790	446
8	980	340
9	1840	460
10	0,0	0,0

Tabela A2: Dados da Referência [3]

DADOS DAS LINHAS

Barra Inicial	Barra Final	Resistência da Seção em (ohms)	Reactância da Seção em (ohms)
1	2	5,3434	3,0264
2	3	4,7953	2,7160
3	4	2,0552	1,1640
4	5	0,9053	0,7886
5	6	1,9831	1,7276
6	7	0,6984	0,6084
7	8	0,7463	1,2050
8	9	0,0140	0,6051
9	Subestação	0,1233	0,4127



A.2. **Dados Completos do Sistema de 34 Barras [16]**

O sistema hipotético de 34 barras apresenta uma tensão base de 11 KV, potência base de 100 MVA. Seus dados foram obtidos da referência [16]. Abaixo está a Tabela 3 com os dados das barras e a Tabela 4 com os dados das linhas.

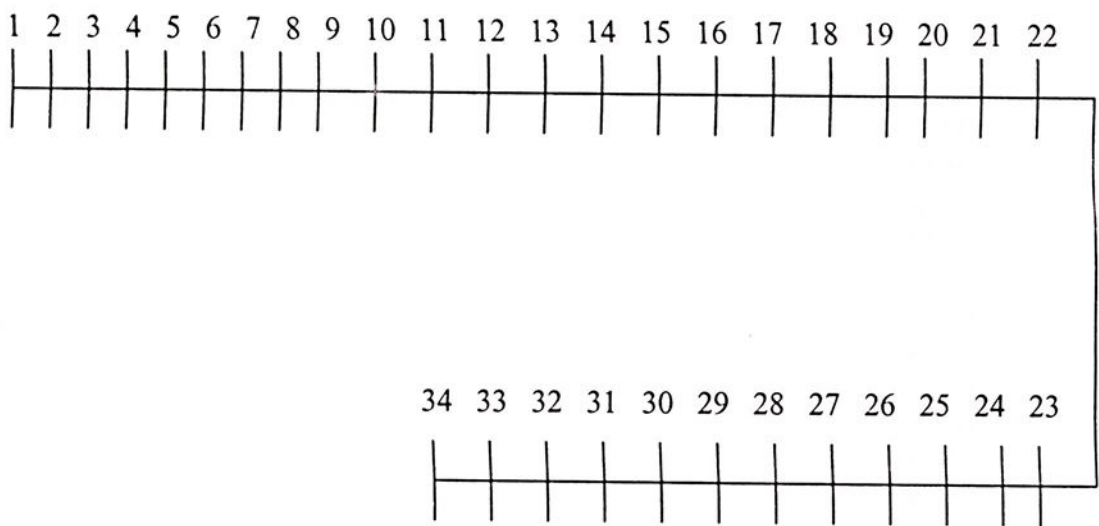


Fig. A.2.: Diagrama Unifilar do Sistema de Distribuição de 34 Barras

Tabela A3: Dados da Referência [16].

Dados das Barras

N'. das Barras	Potência Ativa (W)	Potência Reativa (kVAr)
1	0,00	0,00
2	230,00	142,50
3	0,00	0,00
4	230,00	142,50
5	230,00	142,50
6	0,00	0,00
7	0,00	0,00
8	230,00	142,50
9	230,00	142,50
10	0,00	0,00
11	230,00	142,50
12	137,00	84,00
13	72,00	45,00
14	72,00	45,00
15	72,00	45,00
16	13,50	7,50
17	230,00	142,50
18	230,00	142,50
19	230,00	142,50
20	230,00	142,50
21	230,00	142,50
22	230,00	142,50
23	230,00	142,50
24	230,00	142,50
25	230,00	142,50
26	230,00	142,50
27	137,00	85,00
28	75,00	48,00
29	75,00	48,00
30	75,00	48,00
31	57,00	34,50
32	57,00	34,50
33	57,00	34,50
34	57,00	34,50



Tabela A4: Dados da Referência [16].

Dados das Linhas

Barra Inicial	Barra Final	Resistência da Seção em (ohms)	Reactância da Seção em (ohms)
1	2	0,1170	0,0480
2	3	0,1073	0,0440
3	4	0,1645	0,0457
4	5	0,1495	0,0415
5	6	0,1495	0,0415
6	7	0,3144	0,0540
7	8	0,2096	0,0360
8	9	0,3144	0,0540
9	10	0,2096	0,0360
10	11	0,1310	0,0225
11	12	0,1048	0,0180
12	13	0,1572	0,0270
13	14	0,2096	0,0360
14	15	0,1048	0,0180
15	16	0,0524	0,0090
16	17	0,1794	0,0498
17	18	0,1645	0,0457
18	19	0,2079	0,0473
19	20	0,1890	0,0430
20	21	0,1890	0,0430
21	22	0,2620	0,0450
22	23	0,2620	0,0450
23	24	0,3144	0,0540
24	25	0,2096	0,0360
25	26	0,1310	0,0225
26	27	0,1048	0,0180
27	28	0,1572	0,0270
28	29	0,1572	0,0270
29	30	0,1572	0,0270
30	31	0,1572	0,0270
31	32	0,2096	0,0360
32	33	0,1572	0,0270
33	34	0,1048	0,0180



A.3.

O sistema hipotético de 70 barras apresenta uma tensão base de 12,66 KV, potência base de 100 MVA. Seus dados foram obtidos da referência [7]. Abaixo está a Tabela 5 com os dados das barras e a Tabela 6 com os dados das linhas.

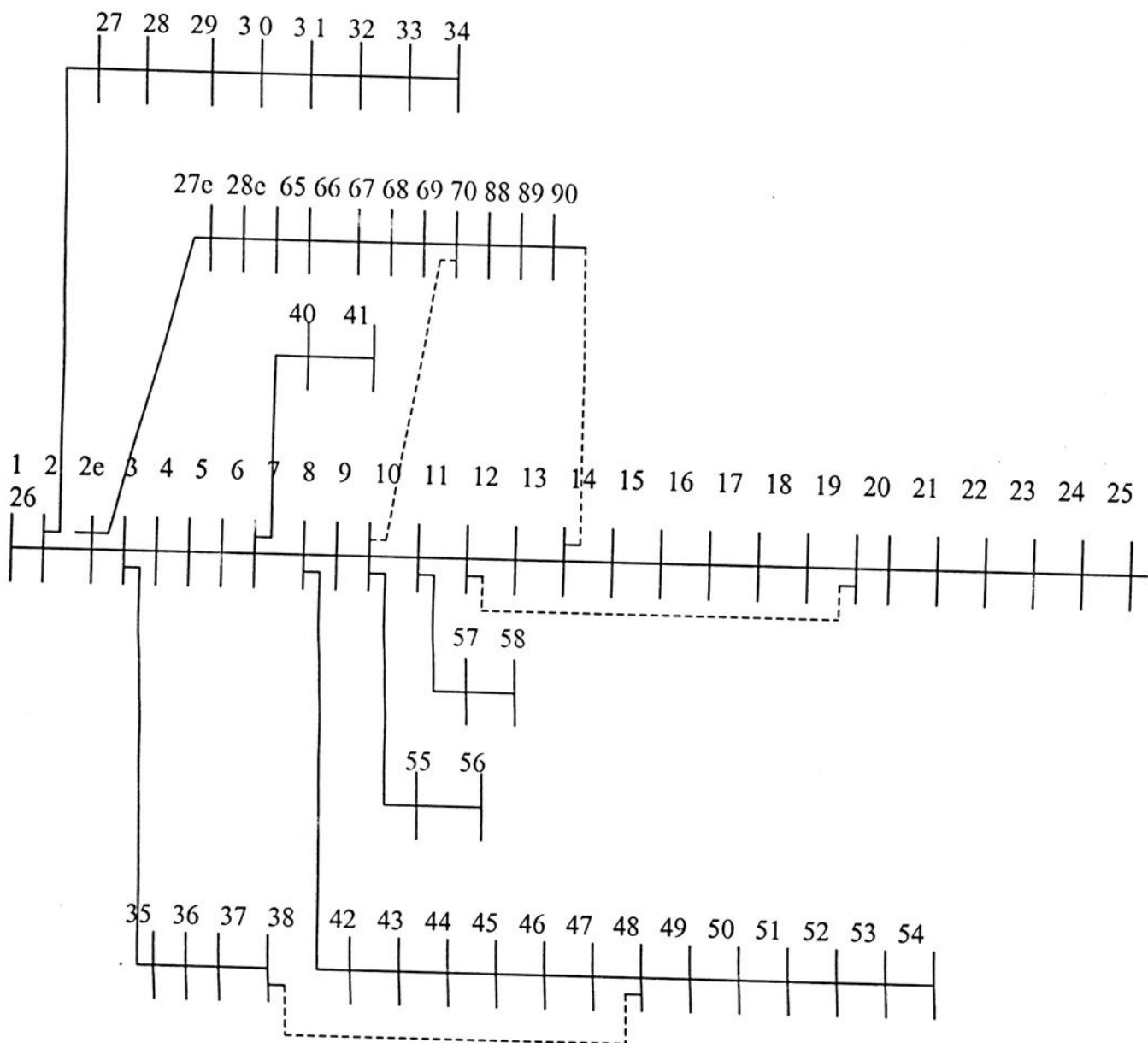


Fig. A.3.: Diagrama Unifilar do Sistema de Distribuição de 70 Barra

Tabela A5: Dados da Referência [7].

Dados das Barras

Nº. das Barras	Potência Ativa (M)	Potência Reativa (kVAr)
1	0,00	0,00
2	0,00	0,00
3	0,00	0,00
4	0,00	0,00
5	0,00	0,00
6	0,00	0,00
7	2,60	2,20
8	40,40	30,00
9	75,00	54,00
10	30,00	22,00
11	28,00	19,00
12	145,00	104,00
13	145,00	104,00
14	8,00	5,50
15	8,00	5,50
16	0,00	0,00
17	45,50	30,00
18	60,00	35,00
19	60,00	35,00
20	0,00	0,00
21	1,00	0,60
22	114,00	81,00
23	5,30	3,50
24	0,00	0,00
25	28,00	20,00
26	0,00	0,00
27	14,00	10,00
28	14,00	10,00
29	26,00	18,60
30	26,00	18,60
31	0,00	0,00
32	0,00	0,00
33	0,00	0,00
34	14,001	0,00
35	19,50	14,00
36	6,00	4,00
37	26,00	18,55
38	26,00	18,55
39	0,00	0,00
40	24,00	17,00



Nº. das Barras	Potência Ativa (kW)	Potência Reativa (kVAr)
41	24,00	17,00
42	1,20	1,00
43	0,00	0,00
44	6,00	4,30
45	0,00	0,00
46	39,22	26,30
47	39,22	26,30
48	0,00	0,00
49	79,00	56,40
50	384,70	274,50
51	384,70	274,50
52	40,50	28,30
53	3,60	2,70
54	4,35	3,50
55	26,40	19,00
56	24,00	17,20
57	0,00	0,00
58	0,00	0,00
59	0,00	0,00
60	100,00	72,00
61	0,00	0,00
62	1244,00	888,00
63	32,00	23,00
64	0,00	0,00
65	227,00	162,00
66	59,00	42,00
67	18,00	13,00
68	18,00	13,00
69	28,00	20,00
70	28,00	20,00



Tabela A6: Dados da Referência [7]

Dados das Linhas

Barra Inicial	Barra Final	Resistência da Seção em (ohms)	Reactância da Seção em (ohms)
1	2	0,0005	0,0012
2	3	0,0005	0,0012
3	4	0,0001	0,0001
4	5	0,0015	0,0036
5	6	0,0251	0,0294
6	7	0,3660	0,1864
7	8	0,3811	0,1941
8	9	0,0922	0,0470
9	10	0,0493	0,0251
10	11	0,8190	0,2707
11	12	0,1872	0,0619
12	13	0,7114	0,2351
13	14	1,0300	0,3400
14	15	1,0440	0,3450
15	16	1,0580	0,3496
16	17	0,1966	0,0650
17	18	0,3744	0,1238
18	19	0,0047	0,0016
19	20	0,3276	0,1083
20	21	0,2106	0,0696
21	22	0,3416	0,1129
22	23	0,0140	0,0046
23	24	0,1591	0,0526
24	25	0,3463	0,1145
25	26	0,7488	0,2475
26	27	0,3089	0,1021
27	28	0,1732	0,0572
3	29	0,0044	0,0108
29	30	0,0640	0,1565
30	31	0,3978	0,1315
31	32	0,0702	0,0232
32	33	0,3510	0,1160
33	34	0,8390	0,2816
34	35	1,7080	0,5646
35	36	1,4740	0,4873
4	37	0,0044	0,0108
37	38	0,0640	0,1565
38	39	0,1053	0,1230





UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

Câmpus de Ilha Solteira
Programa de Pós-Graduação em Engenharia Elétrica
Av. Brasil Centro, 56
15385-000 Ilha Solteira - SP
www.dee.feis.unesp.br

