

unesp

UNIVERSIDADE ESTADUAL PAULISTA
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DISSERTAÇÃO DE MESTRADO
EM
ENGENHARIA ELÉTRICA

1210001145



Nº 44

ANÁLISE EXPERIMENTAL DE ALGORITMOS
GENÉTICOS NO PLANEJAMENTO A LONGO
PRAZO DE SISTEMAS DE TRANSMISSÃO
ALUNA: MARILENE ANDREIA MANTOVANI SILVA
ORIENTADOR: RUBÉN AUGUSTO ROMERO LÁZARO

Te.1145

**UNIVERSIDADE ESTADUAL PAULISTA – UNESP
FACULDADE DE ENGENHARIA – CAMPUS DE ILHA
SOLTEIRA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA**

MARILENE ANDRÉIA MANTOVANI SILVA

**“ANÁLISE EXPERIMENTAL DE ALGORITMOS
GENÉTICOS NO PLANEJAMENTO A LONGO PRAZO
DE SISTEMAS DE TRANSMISSÃO”**



MARILENE ANDRÉIA MANTOVANI SILVA

**“ANÁLISE EXPERIMENTAL DE ALGORITMOS
GENÉTICOS NO PLANEJAMENTO A LONGO
PRAZO DE SISTEMAS DE TRANSMISSÃO”**

Dissertação submetida à Faculdade de
Engenharia de Ilha Solteira – UNESP como
parte dos requisitos necessários para obtenção do
título de mestre em Engenharia Elétrica (ME).

Orientador: Prof. Dr. Rubén Augusto Romero Lázaro

Proc. 17/2000 NRD 024

UNESP - "CAMPUS DE ILHA SOLTEIRA"	
SERVIÇO TÉCNICO DE BIBLIOTECA E DOCUMENTAÇÃO	
DATA DE CHEGADA <i>27/3/2000</i>	DATA DE TOMBO <i>28/07/2000</i>
REGISTRADO POR <i>Kélia</i>	TOMBO <i>1145</i>
AQUISIÇÃO <i>Doação - autor</i> <i>R\$ 10,00</i>	CLASSIFICAÇÃO <i>S 586a</i>

8040400



Ilha Solteira – SP
Fevereiro - 2000

“Análise Experimental de Algoritmos Genéticos no Planejamento a Longo Prazo de Sistemas de Transmissão”

Marilene Andréia Mantovani Silva

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).

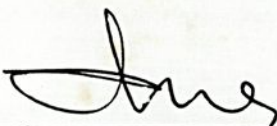


Prof. Dr. Carlos Roberto Minussi - Coordenador

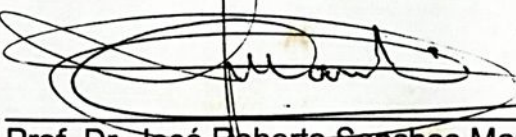
COMISSÃO EXAMINADORA:



Prof. Dr. Rubén Augusto Romero Lázaro - orientador



Prof. Dr. Francisco José Gomes



Prof. Dr. José Roberto Sanches Mantovani

Ilha Solteira – SP, dezembro de 1999

Ao meu marido Cláudio que sempre
me apoiou e incentivou.



AGRADECIMENTOS

- ♦ Ao professor Rubén Augusto Romero Lázaro pelo estímulo, orientação e pelo profissionalismo;
- ♦ Aos Professores José Roberto Sanches Mantovani e Francisco José Gomes pela grande ajuda na correção do texto da tese, assim como a boa vontade dispensada;
- ♦ Ao professor Ramon Alfonso Gallego pela grande ajuda e valiosas sugestões;
- ♦ Aos professores e funcionários do Departamento de Engenharia da FEIS;
- ♦ À secretaria de Pós – Graduação;
- ♦ Um agradecimento a FAPESP, que apoiou esta pesquisa através do processo 97 / 10494-6

Sumário

Capítulo 1	01
1. Introdução	01
Capítulo 2	04
2. Modelagem Matemática	04
Capítulo 3	11
3. Técnicas de Soluções.....	11
3.1. Algoritmos Heurísticos Usados no Problema de Planejamento de Sistemas de Transmissão.....	11
3.1.1 Heurísticas Construtivas Usando o Modelo de Transportes	13
3.1.2 Heurísticas Construtivas Usando o Modelo DC.....	15
3.1.3 Heurísticas Construtivas Para Modelos Mistos.....	18
3.2. Outros Algoritmos Usados no Problema de Planejamento.....	21
3.2.1. Algoritmos de Otimização Clássica	21
3.2.2 Algoritmos Combinatoriais	22
3.3 Conclusões.....	23
Capítulo 4	24
4. O Algoritmo Genético: Teoria.....	24
4.1. Introdução	24
4.2. Aspectos Gerais do Algoritmo Genético	26
4.2.1. População Inicial	27
4.2.2. Mecanismo de Codificação do Problema.....	27
4.2.3. Avaliação da Função Objetivo ou Equivalente (Fitness Function).....	29
4.2.4. Seleção e Operadores Genéticos do Ciclo Geracional.....	30
4.2.5. Parâmetros de Controle do AG	34
4.2.6. Critério de Parada.....	35

4.2.7 Teoria do Esquema.....	36
4.2.8. O Conceito de Building Blocks	38
4.3. Conclusões.....	39
Capítulo 5.....	40
5. O Operador Genético de Seleção.....	40
5.1. Introdução	40
5.2 Análise de técnicas de Seleção.....	41
5.2.1. Propostas Baseadas em Seleção Proporcional.....	41
5.2.1.1. Seleção Proporcional Básica.....	41
5.2.1.2. Seleção Estocástica do Resíduo.....	46
5.2.1.3. Seleção Proporcional com Número Máximo de Descendentes	47
5.2.1.4. Seleção Determinística.....	48
5.2.2. Seleção Usando Escalonamento Linear	49
5.2.3. Seleção Baseada em Ordenamento.....	52
5.2.4. Seleção Baseada em Torneio	56
5.2.5. Algoritmo Genitor	57
5.2.6. Janelamento – “Windowing”	58
5.2.7. Escalonamento Sigma	59
5.3. Critérios Teóricos de Desempenho das Técnicas de Seleção.....	60
5.3.1. Pressão Seletiva.....	60
5.3.2. Tempo de “Take Over”	61
5.3.3. Complexidade Computacional.....	65
5.3.4. Taxa de Crescimento da Melhor Configuração.....	67
5.3.5. Desempenho na Presença de Ruído	68
5.4. Conclusões	69
Capítulo 6	70
6.O Algoritmo Genético Básico Aplicado ao Problema de Planejamento.....	70
6.1. Representação do Problema.....	71
6.2. Codificação do Problema.....	72
6.3. O Mecanismo Geracional: Seleção-Recombinação-Mutação	75
6.4. Geração da População Inicial.....	77
6.5. Parâmetros de Controle e Critérios de Parada	78
6.6.Implementações Especializadas	78
6.6.1.Implementação de uma Taxa de Mutação Variável	78
6.6.2 Busca Eficiente de Populações Iniciais	79
6.6.3 Parâmetro de Penalidade α	80
6.6.4. Identificação de “Building Blocks” Especiais.....	81
6.6.5. Algoritmo Genético Proposto	84
6.7. Conclusões.....	85

Capítulo 7	87
7. Resultados Obtidos.....	87
7.1. Testes Realizados com o Sistema de 06 Barras de Garver Sem Redespacho.....	87
7.2. Testes Realizados com o Sistema Sul Brasileiro Sem Redespacho	91
7.3. Testes Realizados com o Sistema Norte-Nordeste (2008).....	94
7.4. Comentário dos Resultados	98
7.5. Resultados Encontrados em Relação ao Algoritmo Genético Apresentado em [1].....	99
Capítulo 8	101
8. Conclusões.....	101
Referências Bibliográficas	103
Apêndice A	106
Apêndice B	112
Apêndice C	123

RESUMO

Este trabalho apresenta o algoritmo genético aplicado ao planejamento da expansão de sistemas de transmissão a longo prazo. Diferentes técnicas de seleção foram implementadas, testadas e seus resultados foram comparados. A seleção de torneio obteve melhor desempenho. O algoritmo proposto foi capaz de encontrar a solução ótima para sistemas de pequeno e médio porte. Para sistemas de grande porte foi encontrado excelentes soluções, relativos ao custo e as estruturas das configurações. Os sistemas utilizados nas simulações foram: sistema de 06 barras de Garver, sistema Sul brasileiro e Norte – Nordeste brasileiro.

ABSTRACT

This work presents the genetic algorithm applied in the planning of the long term expansion of transmission systems. Different selection techniques were implemented, tested and results comparasion. The tournament selection obtain best performance. The proposed algorithm was able to find optimal solutions for small systems and mediun size. For large size systems excellents sets of solutions were found, both regarding costs and the structure of the proposed configurations. The systems used in the simulation were: the 6-bus Garver, the brazilian South system and brazilian North–Northeastern system.

CAPÍTULO 1

Introdução

O planejamento a longo prazo de sistemas de transmissão, é um problema clássico de sistemas de energia elétrica e cuja modelagem matemática ideal corresponde a um problema de programação não linear inteira mista (PNLIM), apresentando ainda, o fenômeno de explosão combinatorial, o qual mereceu grande atenção de muitos pesquisadores que utilizam técnicas diferenciadas para solucionar um problema complexo desse tipo. As principais dificuldades na solução deste problema estão relacionadas com a natureza combinatorial do processo, que normalmente leva a um número explosivo de alternativas, mesmo no caso de sistemas de médio porte. Além disso, o problema apresenta uma estrutura multimodal com um número elevado de ótimos locais levando a maioria dos métodos aproximados a pararem numa solução ótima local e às vezes de pobre qualidade. Para contornar estes problemas foram propostas muitas alternativas na literatura especializada.

Neste trabalho apresenta-se o problema de planejamento estático da expansão de sistemas de energia elétrica. Neste tipo de planejamento, dada uma configuração inicial e os dados de geração e demanda do horizonte de planejamento (além de outros dados como

limites de operação, custos e restrições de investimento), objetiva-se determinar o plano de expansão com um custo mínimo, isto é, deve-se determinar *onde e que tipos* de novos equipamentos devem ser instalados. Naturalmente, este é um subproblema de um caso mais geral, chamado planejamento dinâmico da expansão onde, adicionalmente, deseja-se saber *quando* instalar estes novos equipamentos.

Como todo problema de otimização matemática, o problema de planejamento de sistemas de transmissão pode ser separado em duas partes claramente definidas: modelagem matemática e a técnica de solução escolhida para resolver esse modelo matemático. Na literatura especializada existem três modelos ou formulações matemáticas para realizar a modelagem do problema de planejamento de sistemas de transmissão: modelo DC, modelo de transportes e o modelo híbrido. Tradicionalmente, o modelo de fluxo de carga DC (Direct Current) é considerado ideal, sendo os outros versões relaxadas ou simplificadas do modelo DC. Na resolução destes modelos são utilizadas várias técnicas as quais podem ser agrupadas em três grupos: *métodos heurísticos ou aproximados, métodos de otimização clássica e métodos de otimização combinatorial*. A modelagem matemática escolhida neste trabalho é o *modelo DC modificado*. No capítulo dois é analisado em detalhes a modelagem matemática do problema de planejamento de sistemas de transmissão.

Os *métodos aproximados* apresentam a vantagem de fornecer soluções de boa qualidade com pouco esforço computacional, mas os mesmos raramente encontram a solução ótima de sistemas reais e não fornecem informação da qualidade da solução obtida, isto é, o quão perto da solução ótima chegamos. Entretanto, esses métodos ainda são os mais usados pelas concessionárias de energia elétrica nos trabalhos de planejamento (são os mais utilizados pelas empresas elétricas no planejamento de sistemas reais).

Os *métodos de aproximação clássica*, geralmente usando técnicas de decomposição matemática, apresentam a característica de encontrar a solução ótima do problema de planejamento e são muito eficientes em sistemas de pequeno e médio porte, mas para sistemas de grande porte ainda apresentam problemas de elevado esforço computacional e de convergência prematura.



Os *Métodos de otimização combinatorial* como “Simulated Annealing” (SA), “Genetic Algorithms” (GA), “Tabu Search” (SA), etc, foram aplicados com sucesso na solução de muitos problemas de pesquisa operacional e de engenharia elétrica. Estes métodos apresentam a característica geral de convergirem em soluções ótimas ou quase ótimas, porém com esforço computacional elevado. No capítulo três são apresentadas, resumidamente, as principais técnicas de solução usadas na solução do problema de planejamento de sistemas de transmissão.

No capítulo quatro é descrita a teoria básica dos algoritmos genéticos, analisando as características básicas do algoritmo genético e os principais operadores. Dentre todos os operadores genéticos, no presente trabalho foi analisado com mais detalhe o operador de seleção nos seus aspectos teóricos e práticos. Essa análise detalhada é discutida no capítulo cinco.

No capítulo seis é apresentado o algoritmo genético básico aplicado ao problema de planejamento de sistemas de transmissão e, adicionalmente, um conjunto de características específicas do problema.

No capítulo sete são apresentados os resultados obtidos nos testes de vários sistemas elétricos conhecidos na literatura especializada, sendo analisado em detalhes o comportamento prático de vários operadores de seleção.

Finalmente, no capítulo oito são discutidas as conclusões mais importantes do trabalho desenvolvido.



CAPÍTULO 2

Modelagem Matemática

A modelagem matemática do problema de planejamento de sistemas de transmissão a longo prazo é apresentada neste capítulo, sendo utilizado nesta representação o modelo DC.

A modelagem matemática consiste na representação de um problema da vida real através de um modelo matemático, relacionando um conjunto de variáveis de decisão as quais podem assumir formas e tipos variados. Em geral, quanto mais exata é a modelagem matemática, mais complexa é sua resolução. Assim, deve existir um compromisso entre a modelagem matemática e a técnica de solução: a modelagem matemática deve representar de maneira adequada o problema da vida real e, além disso, permitir sua resolução por técnicas disponíveis e com esforços computacionais aceitáveis. Neste contexto, o conceito de modelagem adequada varia com o tempo porque modelos muito complexos podem se tornar adequados no futuro com o aparecimento de novas técnicas de solução e/ou a fabricação de computadores muito mais velozes que os disponíveis atualmente.

No planejamento de sistemas de transmissão, o problema da vida real, existe um sistema elétrico com uma topologia corrente e se deseja encontrar o plano de expansão

ótima (construção de novos circuitos) para um horizonte de planejamento definido, isto é, onde e que tipos de circuitos devem ser construídos para que o sistema opere adequadamente num horizonte de planejamento, para um crescimento especificado da demanda. Nesse contexto, a modelagem matemática ideal para indicar a operação adequada seria a representação do problema através das relações matemáticas de fluxo de carga AC (Alternative Current). Entretanto, em planejamento de sistemas de transmissão esta representação não é utilizada por vários motivos, sendo os mais importantes os seguintes:

- A topologia inicial do sistema elétrico usado em planejamento é um sistema não conexo, isto é, apresenta um conjunto de barras isoladas ou ilhadas da parte principal do sistema e, pelo menos no contexto atual, não é possível resolver sistemas deste tipo usando a modelagem matemática de fluxo de carga AC e as técnicas de solução conhecidas para resolver este tipo de problema.
- O planejamento de sistemas de transmissão soluciona somente o fornecimento de potência ativa no sistema elétrico, sendo que o problema de fornecimento de reativos (planejamento de reativos) é resolvido em uma fase posterior. Neste último caso, mesmo que o sistema elétrico seja conexo, a convergência do modelo AC seria difícil ou às vezes impossível, pois o problema frequentemente não teria solução. Simultaneamente ainda não existem técnicas de solução para resolver o problema da expansão dos sistemas de transmissão (construção de linhas e/ou transformadores ou planejamento de sistemas de transmissão) e de alocação de reativos no sistema elétrico.

Levando em conta as observações anteriores, atualmente se considera a modelagem matemática conhecida como *fluxo de carga DC*, doravante denominada modelo DC, como sendo a ideal a ser usada no planejamento de sistemas de transmissão. Os principais motivos para esse consenso são os seguintes:



- Testes experimentais exaustivos mostraram que os resultados obtidos usando o modelo DC apresentam-se muito próximos aos obtidos usando o fluxo de carga AC em relação à distribuição dos fluxos de potência ativa.
- Existem várias técnicas de solução (algoritmos) que solucionam de maneira adequada os problemas de planejamento usando o modelo DC.

Entretanto, nas três décadas de pesquisas realizadas em planejamento de sistemas de transmissão foram propostos vários modelos matemáticos para representá-lo. Três desses modelos ainda são usados nos trabalhos de planejamento de sistemas de transmissão: *modelo de transportes, modelo híbrido e modelo DC*.

Usando os três modelos matemáticos mencionados anteriormente, o problema resultante é um problema de otimização matemática que envolve equações algébricas lineares e/ou não lineares e com variáveis de decisão inteiras e reais. Assim, os problemas resultantes correspondem a um campo da pesquisa operacional conhecida como programação não linear inteira mista e, em princípio, para resolver estes problemas deve-se usar técnicas de programação não linear inteira mista (PNLIM).

Os modelos de transportes e híbridos são versões relaxadas do modelo DC. O modelo DC usa as duas leis de Kirchhoff na modelagem matemática do problema de planejamento, isto é, todas as barras do sistema devem satisfazer a primeira lei de Kirchhoff e todos os laços devem satisfazer a segunda lei de Kirchhoff.

A formulação matemática assume a seguinte forma:



$$\min v = \sum_{(i,j)} c_{ij} n_{ij} \quad (2.1)$$

s.a.

$$S f + g = d \quad (2.1.a)$$

$$f_{ij} - (\gamma_{ij}^0 + x_{ij})(\theta_i - \theta_j) = 0 \quad (2.1.b)$$

$$|f_{ij}| \leq (x_{ij} + \gamma_{ij}^0) \bar{\phi}_{ij} \quad (2.1.c)$$

$$0 \leq g \leq \bar{g} \quad (2.1.d)$$

$$0 \leq n_{ij} \leq \bar{n}_{ij} \quad (2.1.e)$$

$$n_{ij} \text{ inteiro} \quad (2.1.f)$$

$$x_{ij} \text{ discreto} \quad (2.1.g)$$

$$f_{ij} \text{ irrestrito}$$

$$\theta_j \text{ irrestrito}$$

Nesta formulação são utilizadas as seguintes variáveis:

$c_{ij} \rightarrow$ Custo de uma linha entre as barras i-j.

$S \rightarrow$ Matriz de incidência nó-ramo.

$g \rightarrow$ Vetor de geração.

$\bar{g} \rightarrow$ Vetor de gerações máximas permitidas.

$d \rightarrow$ Vetor de cargas.

$f \rightarrow$ Vetor de fluxo de potência nos ramos.

$f_{ij} \rightarrow$ Fluxo de potência elétrica na linha entre i-j.

$\gamma_{ij}^0 \rightarrow$ Susceptância equivalente dos circuitos existentes no ramo i-j na configuração base.

$x_{ij} \rightarrow$ Susceptância equivalente dos circuitos adicionados no ramo i-j.

$\theta \rightarrow$ Vetor de ângulos das tensões nodais.

θ_i e $\theta_j \rightarrow$ Ângulos de tensão nas barras terminais do ramo i-j.

$n_{ij} \rightarrow$ Número de linhas adicionadas no ramo i-j.

$\bar{n}_{ij} \rightarrow$ Número máximo de adições de linhas permitidas no ramo i-j.

Relações auxiliares:

$$n_{ij} = \frac{x_{ij}}{\gamma_{ij}}; \quad \bar{\phi}_{ij} = \frac{\bar{f}_{ij}}{\gamma_{ij}}$$

No modelo DC apresentado na formulação (2.1), o sistema elétrico obedece as duas leis de Kirchhoff. A relação (2.1.a) apresenta um conjunto de equações que satisfaz a primeira Lei de Kirchhoff, a relação (2.1.b) apresenta um outro conjunto de equações que satisfaz a segunda Lei de Kirchhoff, a relação (2.1.c) impõe limites máximos nos fluxos através dos circuitos, a equação (2.1.d) impõe limites à geração nas barras de geração, a equação (2.1.e) impõe limites no número de adições de linhas por circuito.

Após algumas manipulações algébricas, chega-se a uma formulação equivalente que assume a seguinte forma:

$$\min v = \sum_{(i,j)} c_{ij} n_{ij} \quad (2.2)$$

s.a.

$$B\theta + g = d$$

$$|\theta_i - \theta_j| \leq \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_1$$

$$x_{ij} |\theta_i - \theta_j| \leq x_{ij} \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_2$$

$$0 \leq g \leq \bar{g}$$

$$0 \leq n_{ij} \leq \bar{n}_{ij}$$

$$n_{ij} \text{ inteiro}$$

$$x_{ij} \text{ discreto}$$

$$\theta_j \text{ irrestrito}$$

$B \rightarrow$ Matriz de susceptância do sistema elétrico.

$\theta \rightarrow$ Vetor de ângulos de tensão de barras do sistema elétrico.

$\Omega_1 \rightarrow$ Conjunto de todos os ramos ij em que existem circuitos na configuração base.

$\Omega_2 \rightarrow$ Conjunto dos ramos novos onde é possível adicionar circuitos.

É recomendado, por vezes, realizar modificações nas equações básicas dos três modelos utilizados. A modificação mais utilizada consiste no emprego de geradores artificiais em todas as barras de carga, com o objetivo fundamental de facilitar o uso de determinados algoritmos. Do ponto de vista matemático os geradores artificiais constituem apenas novas variáveis (incógnitas) no problema original, que fica com o número de variáveis acrescido, mas com solução facilitada. Logicamente, para que a solução de ambos problemas (original e modificado) sejam equivalentes, na solução final todas as variáveis correspondentes aos geradores artificiais devem ser iguais a zero.

Usando as novas variáveis r_i correspondentes aos geradores artificiais tem-se o modelo DC modificado com a seguinte forma:

$$\min v = \sum_{(i,j)} c_{ij} n_{ij} + \alpha \sum_{k \in \Gamma} r_k \quad (2.3)$$

s.a.

$$B\theta + g + r = d$$

$$|\theta_i - \theta_j| \leq \bar{\phi}_{ij} \quad \forall (ij) \in \Omega_1$$

$$x_{ij} |\theta_i - \theta_j| \leq x_{ij} \phi_{ij} \quad \forall (ij) \in \Omega_2$$

$$0 \leq r \leq d$$

$$0 \leq g \leq \bar{g}$$

$$0 \leq n_{ij} \leq \bar{n}_{ij}$$

$$n_{ij} \text{ inteiro}$$

$$x_{ij} \text{ discreto}$$

$$f_{ij} \text{ irrestrito}$$

$$\theta_j \text{ irrestrito}$$

$\Gamma \rightarrow$ Conjunto de todas as barras k que existe demanda.

$r_k \rightarrow$ Geração artificial (ou fictícia) na barra k .

$\alpha \rightarrow$ Parâmetro de penalidade devidamente ponderado para tornar pouco atrativa alternativas de investimento com valores de r_k diferentes de zero.

Esta formulação será usada na resolução do problema de planejamento de sistemas de transmissão usando algoritmos genéticos. Lembrar que o modelo modificado sempre possui uma solução factível trivial, não necessariamente ótima, fazendo cada $r_k = d_k$ e todas as outras variáveis iguais a zero.



CAPÍTULO 3

Soluções Utilizadas

O conteúdo apresentado neste capítulo refere-se a um resumo das diferentes metodologias encontradas na literatura para solucionar o problema de planejamento a longo prazo de transmissão de sistemas elétricos.

3.1. Algoritmos Heurísticos Utilizados no Planejamento de Sistemas de Transmissão

Um algoritmo heurístico construtivo é um procedimento passo-a-passo em que de maneira sistemática se encontra uma boa proposta de expansão do sistema elétrico, isto é, a partir da configuração base, em cada passo são adicionados um ou vários circuitos até que o conjunto de adições permita uma operação adequada do sistema elétrico. Portanto, em cada passo a configuração do sistema é modificada pela adição de um ou vários circuitos e esta configuração obtida é denominada configuração corrente. O circuito a ser adicionado em cada passo, é o circuito correspondente ao caminho mais atrativo identificado pelo chamado critério de sensibilidade, indicador de sensibilidade ou índice de desempenho. Assim, a diferença fundamental entre os diferentes algoritmos heurísticos construtivos

reside no indicador de sensibilidade escolhido e, obviamente, do modelo escolhido (transportes, híbrido, DC).

Um indicador de sensibilidade é basicamente um parâmetro que de alguma maneira está relacionado com a variação da função objetivo, devido a alguma variação dos parâmetros do sistema considerando como sistema a configuração corrente. Assim, o indicador de sensibilidade possui as seguintes características:

- Identifica os caminhos mais atrativos para realizar adição de circuitos.
- É um indicador de caráter local, isto é, identifica a melhor estratégia para a configuração corrente em contraposição a um indicador de caráter global que identificaria a melhor estratégia para encontrar a melhor configuração do sistema.
- Como os indicadores locais nem sempre coincidem com os indicadores globais então os algoritmos heurísticos construtivos, frequentemente, não têm capacidade de encontrar as configurações ótimas globais de sistemas reais.

Sua importância deriva dos seguintes fatos:

- Na primeira fase de pesquisas (década de 60 e 70) constituíam a única maneira existente para solucionar problemas de planejamento de sistemas elétricos de grande porte.
- A maioria destes algoritmos são robustos e simples de entender, programar e implementar.
- Requerem pouco esforço computacional.



- Muitas características e propriedades destes algoritmos podem ser utilizadas no desenvolvimento de algoritmos mais complexos como os combinatoriais (“simulated annealing”, algoritmo genético, busca tabu, GRASP, etc.)
- Ainda hoje, constituem as técnicas de solução mais empregadas pelas empresas elétricas.

3.1.1. Heurísticas Construtivas Utilizando o Modelo de Transportes

Garver [2] desenvolveu o primeiro algoritmo de larga difusão para o planejamento de sistemas de transmissão utilizando o modelo de transportes como o mais adequado para solucionar problemas de planejamento. O modelo de transportes assume a seguinte forma:

$$\min v = \sum_{(i,j)} c_{ij} n_{ij} \quad (3.1)$$

s.a.

$$S f + g = d$$

$$|f_{ij}| \leq (n_{ij} + n_{ij}^o) \bar{f}_{ij}$$

$$0 \leq g \leq \bar{g}$$

$$0 \leq n_{ij} \leq \bar{n}_{ij}$$

$$n_{ij} \text{ inteiro}$$

$$f_{ij} \text{ irrestrito}$$

O algoritmo de Garver [2] é um processo chamado construtivo ou passo-a-passo onde em cada iteração do algoritmo se toma uma decisão de adicionar um circuito na configuração corrente. O circuito adicionado é aquele que aparece mais sobrecarregado quando o modelo de transportes é solucionado.

No modelo de transporte é levado em conta somente a primeira lei de Kirchhoff e os limites de fluxos nos circuitos (não é levado em conta a segunda lei de Kirchhoff)



representando um modelo relaxado em relação aos modelos híbridos e DC. Portanto, a solução encontrada para o modelo de transportes não é necessariamente uma solução factível para os modelos híbrido e DC. Além disso, é necessário mencionar a forma matemática do modelo de transportes apresentado por Garver em [2], a qual é significativamente diferente do sistema (3.1) apresentado aqui, embora sejam conceitualmente equivalentes.

O problema (3.1) é um PLIM (Programação Linear Inteira Mista), mas se aceitar como solução valores de n_{ij} contínuos então, (3.1) torna-se um simples problema de PL (Programação Linear). Em outras palavras, se relaxar a integralidade das variáveis inteiras n_{ij} aceitando que assumam valores fracionários então o problema (3.1), muito complexo, transforma-se num problema simples.

A proposta de Garver consiste em resolver o sistema (3.1) relaxado e encontrar a solução ótima contínua para a configuração corrente n^0_{ij} . Conhecidas as incógnitas n_{ij} , encontradas usando um algoritmo de PL, pode-se encontrar os fluxos de potência em todos os circuitos antigos (n^0_{ij}) e novos (n_{ij}). Portanto, aquele caminho novo ij identificado pelo n_{ij} que leva o maior fluxo de potência representa o caminho mais atrativo de acordo com a proposta de Garver. Neste caso, a proposta de Garver consiste em adicionar uma linha na configuração corrente no caminho mais atrativo e atualizar a configuração corrente de acordo com a adição escolhida.

O processo termina quando a solução do PL correspondente à configuração corrente apresentar uma solução com todos os $n_{ij} = 0$, o que significa dizer, que não é necessário realizar mais adições, sendo o conjunto de adições realizadas a proposta de solução do algoritmo de Garver.

O algoritmo pode ser resumido nos seguintes passos:

1. Assumir a configuração base n^0_{ij} como configuração corrente.



2. Resolver o PL correspondente do sistema (3.1) para a configuração corrente. Se todos os $n_{ij} = 0$ então pare, pois foi encontrada uma boa configuração factível. Caso contrário, ir ao passo 3.
3. Calcular os fluxos através de todos os novos circuitos adicionados pela PL, ($n_{ij} \neq 0$), usando a relação $f_{ij}^v = n_{ij} \bar{f}_{ij}$. Identificar o caminho novo ij com o maior valor de f_{ij}^v e atualizar a configuração corrente adicionando um circuito naquele caminho ij . Voltar ao passo 2.

O algoritmo apresentado anteriormente, não tem a mesma estrutura nem as simbologias usadas em [2], mas são conceitualmente equivalentes.

A vantagem da metodologia de Garver é a simplicidade na implementação do algoritmo, pois exige somente soluções sucessivas de programação linear. A maior limitação da metodologia é não garantir a obtenção da solução ótima do sistema planejado. A metodologia de Garver, em essência, é de natureza heurística.

3.1.2. Heurísticas Construtivas Utilizando o Modelo DC

Dois algoritmos heurísticos construtivos que utilizam o modelo DC e muito usados em sistemas brasileiros são o *algoritmo de mínimo esforço* e o *algoritmo de mínimo corte de carga*. Ambos algoritmos são relativamente parecidos e a diferença fundamental está no tipo e na determinação do indicador de sensibilidade usado.

O algoritmo de mínimo esforço [3] é um algoritmo heurístico que faz também um plano de expansão passo-a-passo, isto é, para uma configuração da rede os circuitos são adicionados um a um ou em pequenos grupos. O critério para a adição do próximo circuito é determinado por uma análise de sensibilidade chamada de mínimo esforço.

Nesta metodologia, a dificuldade de desconexões na rede inicial é contornada adotando-se, superposta à configuração do sistema, uma “rede fictícia” constituída por ligações com susceptâncias iguais a, por exemplo, 10^{-4} dos valores nominais, colocados em todos os ramos onde são permitidas a construção de novas linhas. A baixa capacidade de



transmissão da rede fictícia faz com que estas só sejam utilizadas quando não houver possibilidade de transporte de potência pela rede real.

Nesta, emprega-se a seguinte relação, a qual é utilizada como um critério de desempenho na adição de novos circuitos ao sistema.

$$\Delta Z_{ij}^* = -\frac{1}{2} \times (\theta_i^* - \theta_j^*)^2 \times \Delta \gamma_{ij} \quad (3.2)$$

em que $\Delta \gamma_{ij}$ é a variação da susceptância de um circuito no ramo ij , isto é, a susceptância nominal de um circuito onde, em cada passo é resolvida uma versão modificada do modelo DC:

$$B\theta = P \quad (3.3)$$

onde P é a potência líquida do sistema.

Em cada passo do processo de planejamento é adicionado ao sistema aquele circuito que produz o maior impacto na distribuição de fluxos na rede, isto é, aquele que apresenta o maior valor de $|\Delta Z_{ij}|$.

Esta metodologia tem como vantagem ser muito rápida e geralmente apresenta soluções de boa qualidade. A desvantagem é não garantir a otimalidade da solução e, às vezes, apresentar soluções distantes do ótimo global, pois não existem formas de determinar o quão perto da solução ótima está a solução encontrada.

A seguir é mostrado o aspecto básico do algoritmo construtivo utilizando o critério de Mínimo Esforço:

Fase 1: Adição de Circuitos

- a. Definir uma rede base a ser expandida e resolve-se (3.3).
- b. Calcular o índice de desempenho usando (3.2) e a análise de sensibilidade para ordenar as adições mais atraentes.



- c. Adicionar o primeiro elemento da lista à rede e resolver (3.3); se as sobrecargas foram eliminadas, vá para a fase 2, caso contrário volte para (b).

Fase 2: Retirada de Circuitos

- a. Durante a expansão é possível ocorrer a adição de circuitos que depois se revelem supérfluos, então, ordenar os circuitos adicionados pelo custo e tentar retirá-los um a um; se a retirada do circuito não levar à sobrecarga, elimina-lo da expansão e voltar a percorrer a lista ordenada.

O algoritmo do mínimo corte de carga (MCC) [4] utilizado no planejamento de sistemas de transmissão é um algoritmo heurístico do tipo construtivo onde a cada passo se produz a adição de um circuito na configuração base.

Em cada iteração do MCC é realizada a adição de um circuito, o qual é selecionado de acordo com um índice de desempenho ou índice de sensibilidade.

O índice de sensibilidade permite encontrar o circuito mais atrativo a ser adicionado, determinado pela seguinte relação:

$$IS_{MCC} = (\pi_i - \pi_j) \times (\theta_j - \theta_i) \quad (3.4)$$

π_i é o multiplicador de Lagrange da j -ésima restrição do sistema $B \times \theta + g + r = d$ em (2.3) e θ_j são os ângulos de tensão de barra obtidos na solução de (2.3) para a configuração corrente usando um algoritmo de PL.

O índice de sensibilidade é um indicador do impacto que produziria a adição de um circuito no corte de carga de um sistema, se o circuito fosse adicionado ao sistema elétrico. Assim, o circuito possuidor do maior valor do índice de sensibilidade deve ser adicionado à configuração base, pois é o melhor candidato para produzir uma maior diminuição no corte de carga do sistema.

O algoritmo construtivo apresentado em [3], possui a mesma estrutura do algoritmo de mínimo esforço, sendo mostrado a seguir sua estrutura básica:



Fase 1: Adição de Circuitos

- a. Definir a configuração base do sistema elétrico e os circuitos candidatos a adição.
- b. Calcular o corte de carga do sistema para a configuração base corrente resolvendo (2.3) e usando um algoritmo PL. Se não existir corte de carga, prosseguir para a fase 2. Caso contrário, calcular o índice de sensibilidade usando (3.4), ordenar em uma lista os circuitos em ordem decrescente do valor de IS_{MCC} .
- c. Adicionar o primeiro circuito da lista ordenada à configuração corrente, obtendo uma nova configuração base. Voltar ao passo b.

Fase 2: Retirada de Circuitos

- a. Nesta fase tenta-se retirar circuitos adicionados que se revelarem supérfluos. Ordenar numa lista os circuitos adicionados pelo custo (ou pelo nível de carregamento). Tentar retirar cada circuito um a um verificando se sua retirada não produz sobrecarga. Se o circuito foi retirado então, deve-se percorrer novamente a lista até que não seja mais possível retirar nenhum circuito adicionado.

3.1.3. Heurísticas Construtivas Para Modelos Mistos

Existem várias pesquisas usando modelagem matemática mista [5, 7], nas quais, a solução encontrada satisfaz às exigências do modelo DC, com a utilização de algumas modificações na modelagem que substituem os indicadores de sensibilidade dos outros algoritmos heurísticos construtivos. Pode-se destacar o *algoritmo da Rede Marginal de Levi-Calovic* e o *algoritmo de Villasana-Garver*.

Na proposta de Villasana-Garver [7] é utilizada uma modelagem matemática representando duas redes elétricas superpostas, uma rede elétrica correspondente aos



circuitos existentes na chamada configuração corrente e uma rede artificial ou fictícia existente em todos os caminhos onde podem ser adicionados circuitos. A idéia fundamental do algoritmo consiste na hipótese que o sistema elétrico deve tentar solucionar o problema de operação usando somente os circuitos existentes na configuração corrente e usar os circuitos artificiais somente quando os circuitos existentes não forem suficientes para resolver o problema de operação.

Em cada barra do sistema elétrico deve-se satisfazer a primeira lei de Kirchhoff, porém a segunda lei de Kirchhoff deve ser satisfeita somente nos laços formados exclusivamente por circuitos existentes na configuração corrente. Esta formulação pode ser chamada de híbrida porque a rede formada pelos circuitos existentes apresenta a estrutura do modelo DC e a rede artificial possui a estrutura do modelo de transportes. Entretanto, os dois modelos não estão separados, pois ambas as redes, superpostas, devem resolver, em conjunto, os problemas de operação.

Em cada passo do algoritmo apresentado em [7], resolve-se o seguinte problema:

$$\min v = \sum c_{ij} n_{ij} \quad (3.5)$$

s.a.

$$Sf + B_1\theta_1 + g = 0$$

$$|\theta_i - \theta_j| \leq \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_1$$

$$|f_{ij}| \leq \bar{f}_{ij} n_{ij} \quad \forall (i, j)$$

$$0 \leq g_1 \leq \bar{g}$$

$$n_{ij} \geq 0$$

$$f_{ij} \text{ irrestrito}$$

$$\theta_j \text{ irrestrito}$$

$S \rightarrow$ Matriz de incidência nó-ramo da rede artificial, isto é, S é a matriz de incidência nó-ramo do sistema completo.

$f = \{f_{ij}\} \rightarrow$ Fluxo de potência através do circuito artificial no caminho $i - j$.

$B_1 \rightarrow$ Matriz de susceptâncias dos circuitos existentes na configuração corrente.



$\theta_1 \rightarrow$ Vetor de ângulos de tensões das barras que estão ligados ao sistema elétrico através de circuitos normais na configuração corrente.

$n_{ij} \rightarrow$ Número de circuitos artificiais que devem ser adicionados para que o sistema “opere adequadamente”.

A justificativa usada para escolher o circuito à ser adicionado ao sistema é empírica e segue basicamente os mesmos critérios que justificam o critério de adição usado no algoritmo de Garver para o modelo de transportes. Obviamente, os problemas de rede não conexa são contornados pela rede fictícia. No final do processo, quando as adições realizadas operam adequadamente o sistema, a rede fictícia é eliminada ficando disponível a configuração final do problema constituído pelos circuitos existentes na configuração base e as adições de circuitos realizados no processo iterativo. Esta configuração final é uma solução factível para o modelo DC, mas não necessariamente a solução ótima porque foi obtida usando um algoritmo heurístico construtivo.

O algoritmo de Villasana-Garver pode ser resumido da seguinte forma:

Fase 1: Adição de Circuitos

- a. Assumir a configuração base, sem nenhuma adição, como configuração corrente e montar o sistema (3.5) para essa configuração.
- b. Resolver o PL correspondente ao sistema (3.5) para a configuração corrente. Se a resolução do PL não apresenta investimento, isto é, se todos os $n_{ij} = 0$ então pare, pois foi encontrada uma boa configuração factível. Caso contrário, calcular o investimento corrente, ou seja, a soma do investimento das linhas adicionadas mais o investimento contínuo fornecido pelo PL e ir ao passo c.
- c. Adicionar na configuração corrente um circuito naquele caminho fictício, o qual leva o maior fluxo de potência entre todos os caminhos fictícios para obter a nova configuração corrente. Montar o sistema (3.5) para a configuração



corrente, atualizar o investimento correspondente aos circuitos adicionados e voltar ao passo b.

Fase2: Retirada de Circuitos

- a. É possível a existência de alguns circuitos irrelevantes adicionados durante a fase 1 devido a outras adições mais importantes realizadas posteriormente. Ordenar os circuitos adicionados em ordem decrescente de seus custos e eliminar aqueles que, uma vez simulada sua saída, não produzem corte de carga no sistema.

O algoritmo de rede marginal de Levi-Calovic é muito parecido ao algoritmo de Garver-Villasana mas, apresenta algumas contribuições adicionais interessantes. Em vez de usar duas redes superpostas como em [7], em [5] se realiza a montagem e resolvem-se dois modelos matemáticos correspondentes às duas redes diferentes em cada passo do algoritmo heurístico construtivo. O primeiro modelo matemático corresponde ao modelo DC para a chamada configuração corrente, isto é, verifica-se se o modelo opera corretamente para a configuração corrente. Caso o sistema não opere adequadamente para a configuração corrente, monta-se a chamada rede marginal com os resultados obtidos na resolução do primeiro modelo matemático.

3.2. Outros Algoritmos Utilizados em Planejamento

3.2.1. Algoritmos de Otimização Clássica

Na década de 80 iniciou-se uma nova fase na tentativa de resolver a formulação (2.1) de maneira ótima e a principal ferramenta matemática encontrada foram as técnicas de decomposição matemática. O objetivo principal era encontrar a solução ótima do problema



de planejamento de sistemas de transmissão, usando o modelo DC, isto é, resolver a formulação (2.1), que significa resolver um problema PNLIM.

Nesta perspectiva, a metodologia mais usada foi a técnica de decomposição de Benders a qual explora a decomposição natural do problema de planejamento da expansão de sistemas de transmissão em duas partes:

- *Um subproblema de investimento* em que se escolhe um plano de expansão candidato e calculam-se os custos de investimento associados ao mesmo;
- *Um subproblema de operação* onde é testado o plano de expansão candidato em termos do adequado fornecimento de carga.

A otimização global é atingida através de uma resolução iterativa das resoluções separadas dos subproblemas de operação e investimento.

Esta metodologia de decomposição encontrou as configurações ótimas para sistemas de pequeno e de médio porte, porém foi incapaz de resolver sistemas de grande porte como o sistema Norte-Nordeste brasileiro. Os detalhes sobre a implementação de algoritmos de decomposição matemática usados no problema de planejamento de sistemas de transmissão está fora do escopo do presente trabalho. A análise detalhada destes tópicos podem ser encontradas em [18,19].

3.2.2. Algoritmos Combinatoriais

Pode-se mostrar que o problema de planejamento de sistemas de transmissão é um problema NP(Não Polinomial) - completo. Problemas complexos deste tipo podem ser adequadamente resolvidos usando os algoritmos combinatoriais tais como SA, GA, TS, GRASP, etc.

Os algoritmos acima estão sendo empregados na solução de sistemas de grande porte; nestes, reportam-se resultados muito satisfatórios na maioria das aplicações apresentadas. O SA faz uso de conceitos descritos na Mecânica Estatística. No processo físico o material é esquentado a uma temperatura elevada para logo ser resfriado de maneira lenta de tal forma que a estrutura do material ao final do processo fique perfeita. O



algoritmo genético, é uma técnica de busca baseada na mecânica da seleção natural das espécies. Busca tabu, nasce de regras geradas na inteligência artificial: neste, se conta com um esquema de busca local para explorar o espaço de soluções além do ótimo local.

Algoritmos combinatoriais como simulated annealing, genéticos e busca tabu foram usados com sucesso no problema de planejamento de sistemas de transmissão. Assim, por exemplo, para o problema de planejamento de sistemas de transmissão apresentados em [1], [4], [5] onde, foram encontradas soluções ótimas para sistemas de pequeno e médio porte e soluções quase-ótimas para sistemas de grande porte como o sistema Norte-Nordeste brasileiro.

No trabalho propõe-se o desenvolvimento de um algoritmo genético especializado aplicado ao problema de planejamento de sistemas de transmissão. A pesquisa é uma continuação natural da apresentada em [1]. Os aspectos mais importantes a serem explorados são: a incorporação de diferentes formas de implementar a seleção, melhorar os critérios para gerar a população inicial, a codificação das configurações, incorporação de funções próprias de outros algoritmos combinatoriais e testes exaustivos para otimizar de maneira empírica os parâmetros de controle do algoritmo genético.

Nesta pesquisa foi analisada a metodologia de otimização combinatorial do algoritmo genético (GA) e aplicada na resolução do problema da expansão de redes de transmissão a longo prazo.

3.3. Conclusões

Neste capítulo foram apresentados, em forma resumida, as principais técnicas de solução propostas para o problema de planejamento a longo prazo de sistemas de transmissão. Foram analisados com mais detalhe os algoritmos heurísticos construtivos porque eles serão utilizados na estrutura do algoritmo genético proposto no presente trabalho.



CAPÍTULO 4

O Algoritmo Genético: Teoria

Serão discutidos os passos de um algoritmo genético básico, desde a geração da população inicial, tipos de codificação, esquemas de seleções, recombinação, mutação e os principais parâmetros utilizados em seu desenvolvimento.

4.1. Introdução

Os algoritmos genéticos tem-se apresentado como uma técnica de otimização robusta e sua metodologia vem sendo empregada em diversas áreas, tais como: otimização de problemas numéricos e combinatoriais como o escalonamento de trabalho em empresas e o desenvolvimento de layouts para circuitos, em programação automática, aprendizagem de máquinas, economia, ecologia, genética, etc. Os algoritmos genéticos foram inicialmente formulados por John Holland nos anos 60 e desenvolvidos por Holland e seus alunos na Universidade de Michigan em meados de 1970 [17]. Seu objetivo principal não foi desenvolver algoritmos para solucionar problemas específicos, mas dedicar-se ao estudo formal do fenômeno da evolução, como ocorre na natureza, e desenvolver maneiras de incorporá-las aos sistemas computacionais. Os algoritmos genéticos empregam um

conjunto de configurações (populações de indivíduos) para resolver um problema complexo, simulando o que acontece no meio ambiente. Estão baseados no princípio da seleção natural, ou seja, indivíduos melhores dotados tem maiores chances de sobrevivência em um ambiente competitivo e a capacidade de adaptação a um meio mutante é essencial para a sobrevivência de indivíduos e espécies. As características específicas de um indivíduo determinam sua capacidade de sobrevivência e, em última instância, essa capacidade específica é determinada pelo seu conteúdo genético, isto é, pela unidade elementar chamada na biologia *gene*. A seleção natural leva à sobrevivência dos indivíduos melhores dotados e no processo de reprodução estes indivíduos transmitem aos descendentes os melhores genes; por outro lado, os indivíduos menos dotados morrem no processo de competição por espaço, alimentos, etc. Assim, o princípio de seleção natural permite gerar descendentes somente dos indivíduos melhores dotados. A evolução acontece quando as metades dos cromossomos de dois indivíduos bem dotados se juntam durante a reprodução. Neste processo ocorre uma troca de material genético o qual pode levar à geração de um indivíduo muito bem dotado. No algoritmo genético o processo de otimização se inicia com a seleção de uma população, isto é, um conjunto de soluções (configurações) candidatas. Cada configuração é qualificada pelo valor da função objetivo que apresenta. Geralmente a codificação de uma configuração é realizada em sistema binário. Todos os elementos da população são classificados pela qualidade de sua correspondente função objetivo. Assim, cada elemento recebe uma determinada probabilidade para gerar os elementos (configurações) na geração seguinte. Os elementos melhores qualificados ("higher fitness") neste processo tem, no sentido probabilístico, maior probabilidade de participar na geração dos elementos da nova população. Essa nova geração é obtida com as operações de recombinação e mutação. A recombinação leva a uma transferência de material genético entre as melhores configurações. Eventualmente devem aparecer elementos da nova geração com o melhor material genético das configurações melhores qualificadas, produzindo novas configurações de excelente qualidade. Na natureza, a mutação produz a regeneração da perda de material genético. No algoritmo genético, a mutação produz uma modificação esporádica e aleatória de algum gene de um elemento (configuração) da população. Resumindo, pode-se dizer que os algoritmos genéticos são uma técnica de otimização combinatorial que inicia o processo



com uma população inicial, faz um ordenamento seletivo dos elementos da população avaliando a qualidade da função objetivo e gera uma nova população num processo de três etapas: na primeira é realizada a seleção (onde à cada configuração é designada um número determinado de descendentes), na segunda etapa é realizada a recombinação entre os elementos melhores qualificados e com direito a gerar descendentes e na etapa final realiza-se uma mutação aleatória para obter-se a nova população.

4.2. Aspectos Gerais do Algoritmo Genético

O algoritmo genético em sua forma mais simples deve ter definida uma população inicial (nos problemas combinatoriais significa um conjunto de configurações) para resolver um problema de otimização complexo; além disso, é necessário ter disponível um mecanismo de codificação, parâmetros de controle a serem utilizados, avaliação da função objetivo, um mecanismo de seleção, operadores genéticos (recombinação e mutação) e um critério de parada. O algoritmo genético procede da seguinte maneira:

1. Representar adequadamente uma configuração do problema. A mais popular é a representação em codificação binária, onde são facilmente simulados os operadores genéticos de recombinação e mutação;
2. Encontrar uma forma adequada para avaliar a função objetivo ou o seu equivalente ("fitness"). Podendo identificar as configurações de melhor qualidade como sendo aquelas com função objetivo de melhor qualidade;
3. Deve existir uma estratégia de seleção das configurações com direito a participar da formação das configurações da nova população (nova geração);
4. Criar um mecanismo que permita implementar o operador genético de recombinação;
5. Implementar o operador genético de mutação e terminar de gerar a nova população;



6. Se o critério de parada for satisfatório pare, caso contrário, voltar ao passo 2.

4.2.1. População Inicial

As configurações da população inicial podem ser geradas aleatoriamente, pois segundo a literatura, a melhor solução encontrada deve ser independente da população inicial. Fazer evoluir uma população gerada inicialmente de maneira aleatória até chegar a ter configurações perto da solução ótima é um bom teste para se conhecer como está funcionando a implementação do algoritmo, já que teoricamente as características essenciais e críticas da solução final deverão ser consequências do processo evolutivo e não das características dos métodos usados para gerar a população inicial, como é o caso de sistemas de pequeno e médio porte.

Entretanto, novas experiências indicam a existência de uma grande diferença em relação ao esforço computacional quando a população inicial é gerada de maneira aleatória e adequadamente, geralmente usando algoritmos heurísticos muito rápidos (eficientes). Dependendo da complexidade do problema o esforço computacional torna-se muito elevado; dessa forma, na maioria dos problemas complexos de engenharia elétrica foram usadas estratégias diferentes para gerar as configurações iniciais. Em sistemas de grande porte, o processo necessita de uma boa estratégia para gerar a população inicial e essas razões serão comentadas nos próximos capítulos.

4.2.2. Mecanismo de Codificação do Problema

Os algoritmos genéticos trabalham com representações codificadas das soluções, equivalente ao material genético dos indivíduos na natureza. Para que o algoritmo genético represente adequadamente um problema, deve-se optar por uma codificação apropriada às características desse problema. Suponha-se um problema com 3 variáveis binárias: x_1 , x_2 , $x_3 \in \{0,1\}$.

Uma configuração assume a seguinte forma:



$$X = \begin{bmatrix} X_1 & X_2 & X_3 \end{bmatrix}$$

Pode-se representá-la utilizando a codificação binária, fornecendo:

$$X = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$$

Mesmo sendo a codificação binária a mais usada na literatura, existe uma grande variedade de aplicações onde essa representação não é efetiva; por esse motivo, alguns pesquisadores começaram explorar outras formas de representação; em geral aquelas relacionadas com aplicações práticas dos algoritmos genéticos.

Alguns tipos de problemas são mais facilmente representáveis em codificação binária, portanto melhor condicionados à serem solucionados por um algoritmo genético. Também existem problemas onde as variáveis de decisão são inteiras com limitante superior. Neste caso, deve-se fazer uma representação binária destas variáveis, o que aumenta o tamanho do problema. Supõe $x_1, x_2, x_3 \in \{0,1,2,3,4,5,6,7\}$. Cada variável inteira pode ser representada por um equivalente binário ocupando até 3 casas binárias, com uma configuração típica assumindo a seguinte forma:

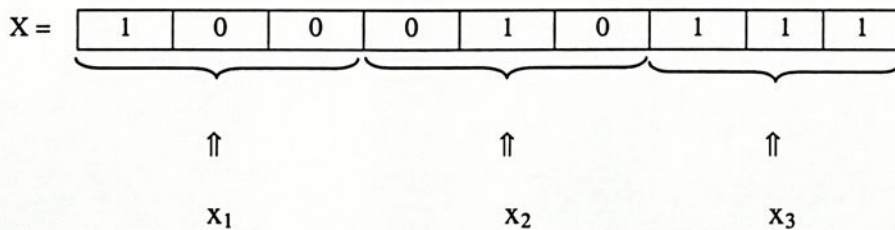


Figura 4.1: Codificação Binária

em que $x_1 = 4$, $x_2 = 2$ e $x_3 = 7$.

O problema mais complexo a ser resolvido em termos de representação das variáveis ocorre quando as variáveis de decisão são contínuas. Na formulação tradicional deve-se fazer a representação binária de uma variável contínua variando em um determinado intervalo. Suponha-se que as variáveis possam assumir valores reais em um intervalo especificado, por exemplo, $x_1, x_2, x_3 \in [0 - 0,31]$. Escolhe-se novas variáveis x'_j

tendo a seguinte equivalência: $x'_j = 100 x_j$. Depois trabalha-se com as novas variáveis x_j , as quais podem assumir novos valores inteiros para o intervalo $[0 - 0,31]$, ou seja, x'_1, x'_2 e $x'_3 \in [0,31]$. Uma configuração típica assume a seguinte forma:

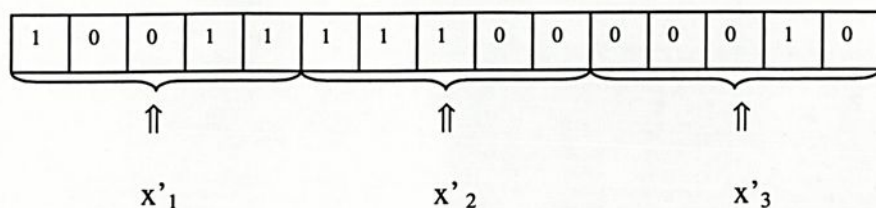


Figura 4.2: Representação Binária de Uma Variável Contínua

Neste caso particular $x'_1 = 19$, $x'_2 = 28$, $x'_3 = 2$ sendo as variáveis reais $x_1 = 0,19$, $x_2 = 0,28$ e $x_3 = 0,02$.

No planejamento de transmissão as variáveis de decisão são as linhas adicionadas em cada caminho candidato e portanto variáveis inteiras e, no caso mais geral, cada variável de decisão pode ser representada por 4 dígitos binários equivalentes a adicionar mais de 8 linhas em cada caminho candidato. Resumindo, em problemas onde as variáveis são binárias a codificação natural é a binária, mas em problemas onde as variáveis são inteiras ou contínuas a transformação para codificação binária leva a um trabalho significativo e as casas binárias de uma configuração podem ser muito grandes. Em [16] apresenta-se uma estratégia muito interessante de codificação binária de variáveis reais.

No presente trabalho, as variáveis de investimento são codificadas em base decimal e as variáveis contínuas são calculadas através de um problema subsidiário, um problema de PL. Logo, uma configuração é representada por um vetor constituído por números decimais e esta proposta exige uma redefinição do operador de mutação.

A implementação adequada da codificação é um problema crucial no desempenho do algoritmo genético.

4.2.3. Avaliação da Função Objetivo ou Equivalente ("Fitness Function")

A função objetivo, a ser otimizada, fornece o mecanismo para avaliar cada configuração. Para manter a uniformidade sobre o domínio de vários problemas,

geralmente, torna-se necessário normalizar a função objetivo dentro de determinados parâmetros. O parâmetro identifica a qualidade de uma configuração, podendo ser a própria função objetivo ou uma equivalente (que poderá ser mais adequada para o algoritmo genético), é chamada de função de adaptação, "*fitness function*", que o mecanismo de seleção utiliza para avaliar cada indivíduo da população.

A função empregada para atribuir os valores de "fitness" aos membros da população deverá possuir boa sensibilidade, tal que consiga fazer uma boa discriminação destes últimos. Assim, por exemplo, os resultados da avaliação de uma função objetivo pode atribuir um valor de 1,09 descendentes a um indivíduo muito bom; por outro lado, pode dar um valor de 0,96 a um indivíduo muito ruim. Se estes valores forem empregados para selecionar os indivíduos na fase de recombinação, seria necessário muito tempo de computação para que os descendentes do primeiro influam mais do que o pior indivíduo.

Uma transformação da função de adaptação será efetuada trocando a referência dos valores das funções, isto é, subtraindo um valor preestabelecido de cada valor das funções. Desta maneira os indivíduos bons participarão em maior porcentagem na formação da nova geração.

Esta transformação deverá privilegiar as melhores configurações, porém sem descartar o material genético presente na população, já que se a transformação estiver influenciada fortemente pelo melhor indivíduo, a busca ficará em uma região do espaço de configurações perto daquela que influenciou na escolha. Caso contrário, se não existir influência das melhores soluções, pode-se ter uma busca demasiada lenta.

4.2.4. Seleção e Operadores Genéticos do Ciclo Geracional

A mais simples implementação de seleção é o chamado *esquema de seleção proporcional*. Neste esquema, cada configuração tem o direito de gerar um número de descendentes proporcional à qualidade da função objetivo. As configurações com maiores valores da função objetivo participarão na geração com muitos descendentes e aquelas com menor valor na função objetivo, praticamente não participarão na geração de novas



configurações. A seleção proporcional encontra um número não inteiro de descendentes para cada configuração devendo-se então utilizar uma estratégia para “integralizar” esse número não inteiro de descendentes. A estratégia mais popular é usar o esquema da roleta.

O esquema de seleção proporcional é implementado usando o chamado *esquema de seleção da roleta*. No esquema da roleta cada configuração (“string”) é alocada num setor circular da roleta, proporcional ao valor da correspondente função objetivo. Um processo aleatório determina o número de descendentes a que tem direito cada configuração. Logicamente, o esquema da roleta pode ser implementado no computador usando um simples vetor de trabalho. Entretanto, é possível substituir o esquema da roleta por mecanismos mais eficientes e rápidos designando, a cada configuração um número de descendentes proporcional à porcentagem determinada para cada configuração e fazendo um ajuste no final do processo para determinar um número de descendentes igual ao tamanho da população corrente. Uma vez terminado o processo de seleção deve-se implementar a recombinação. Na formulação original do algoritmo genético a *recombinação* é a operação fundamental e crucial e a *mutação* é designada uma função secundária. Uma vez definida a seleção, escolhe-se um par de configurações com direito a gerar descendentes e este par é submetido a uma recombinação para gerar dois novos descendentes. Em relação à operação de recombinação, os algoritmos genéticos definem a chamada *taxa de recombinação* (ρ_c). Um algoritmo genético normal não procede a recombinação de todas as configurações escolhidas. A taxa de recombinação determina probabilisticamente a porcentagem de configurações submetidas efetivamente à recombinação. Desta maneira, uma parcela das configurações da população corrente passa integralmente a constituir a lista de novas configurações candidatas para a nova população sem recombinação. Escolhida as duas configurações candidatas, gera-se um número aleatório entre 0 e 1. Se o número aleatório gerado for maior que (ρ_c - taxa de recombinação) as duas configurações passam diretamente para a lista de configurações candidatas sem recombinação e, caso contrário, procede-se a recombinação. Tipicamente o valor data taxa de recombinação (ρ_c) varia entre 0,9 e 1,0.

A escolha das duas configurações para recombinação pode ser feita de forma aleatória ou de forma sistemática ou ordenada, iniciando a recombinação com aquelas



configurações com direito a um número maior de descendentes. Existem quatro tipos de recombinação: simples (“single-point”), dois pontos (“two-point”), multipontos (“multi-point”) e uniforme.

Na recombinação simples, escolhe-se aleatoriamente um ponto de recombinação e o material genético além desse ponto é trocado. Seja L o tamanho ou número de elementos de uma configuração, escolhe-se aleatoriamente um número entre 1 e $L-1$. Seja L_p o número aleatório selecionado então, as duas configurações escolhidas para recombinação trocam a parte correspondente às primeiras L_p posições. Essas duas novas configurações, assim geradas, são as candidatas a configurações na nova população. Pode-se sintetizar o mecanismo de recombinação de um simples ponto da seguinte maneira:

- (i) escolher duas configurações candidatas a recombinação;
- (ii) gerar um número aleatório p no intervalo $[0-1]$;
- (iii) se $p < \text{taxa de recombinação}$, efetua-se a recombinação. Senão, passar as duas configurações para a fase de mutação.

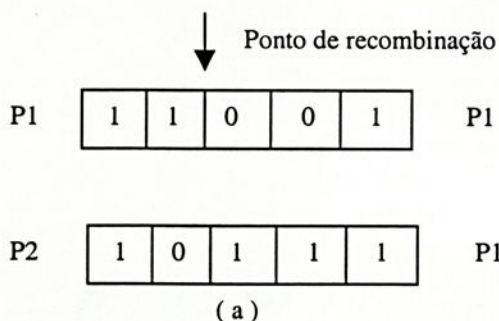


Figura 4.3(a): Configuração Candidata
a Recombinação

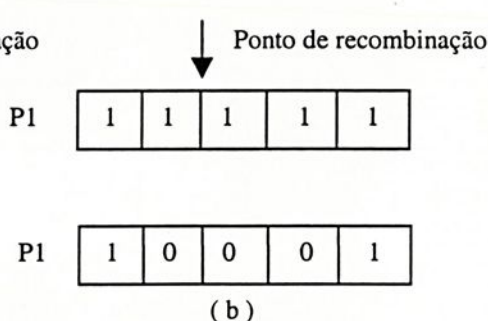


Figura 4.3(b): Configuração Recombinada

A figura 4.3(a) representa duas configurações (“string”) candidatas a recombinação, sendo escolhido como ponto de recombinação o segundo elemento (bit) da configuração. A figura 4.3(b) representa duas novas configurações já permutadas com troca de material genético entre elas.

Na recombinação de dois pontos são selecionados de maneira aleatória dois pontos, sendo trocado o material genético entre eles. No processo de generalização da recombinação pode-se implementar a chamada recombinação multipontos e uniforme. No

caso da recombinação multipontos são escolhidos vários pontos de recombinação trocando parcelas alternadas das duas configurações; finalmente, na recombinação uniforme, cada elemento (bit) é selecionado de maneira aleatória; começando pelo primeiro, é escolhida a primeira configuração que contribui com o primeiro gene (bit) ao primeiro descendente. Na sequência, o segundo descendente recebe o bit da segunda configuração.

Devido ao caráter aleatório que são escolhidas as configurações participantes da recombinação e até mesmo o ponto da recombinação podem ocorrer os seguintes fatos:

- O mesmo par de configurações pode ser escolhido várias vezes, especialmente quando existem configurações com direito a gerar muitos descendentes. Neste caso, se diferentes pontos da recombinação são escolhidos o problema é atenuado, mas se isso não ocorre pode-se gerar configurações idênticas. É necessário existir um mecanismo que evite tal problema.
- Uma mesma configuração pode ser escolhida para recombinação com ela mesma. Uma estratégia simples elimina esse problema, escolhendo outra configuração cada vez que isso acontecer. É possível que para gerar o último par de configurações candidatas exista apenas uma única configuração com direito a gerar descendentes. Neste caso, simplesmente não seria possível realizar recombinação.
- Para eliminar os dois problemas anteriores pode-se escolher-se uma estratégia mais determinística. Por exemplo, iniciar a recombinação escolhendo aquelas configurações com direito a gerar um número maior de descendentes e esgotar todas as recombinações, depois repetir o processo com a próxima configuração com direito a gerar mais descendentes, até esgotar todas sujeitas a gerar descendentes. Este processo deixa de lado partes do algoritmo genético onde as decisões são aleatórias.

Depois da recombinação, as novas configurações candidatas estão sujeitas à mutação. A mutação exerce uma influência especial no algoritmo genético, pois ajuda evitar o problema de mínimo local (convergência prematura). A mutação introduz mudança em torno das variáveis, explorando novas zonas no campo da otimização. A operação de mutação termina com o processo de obtenção da nova geração. O processo de mutação



implica modificar de 0 para 1 ou vice-versa um elemento (gene) de uma configuração. Esse processo é controlado por um parâmetro denominado *taxa de mutação* (ρ_m). No algoritmo genético simples, a mutação é considerada um operador secundário comparado a recombinação e tem a finalidade de restaurar a perda de material genético. Por exemplo, em uma posição k todas as configurações (“strings”) tem valor 0 na representação binária, sendo que a configuração ótima o valor correto nessa posição é 1. Sabe-se que a recombinação não pode regenerar o valor 1 nessa posição, entretanto, com a operação de mutação consegue-se resolver essa dificuldade. O exemplo abaixo mostra o operador mutação, o quarto elemento da “string” foi escolhido como ponto de mutação:

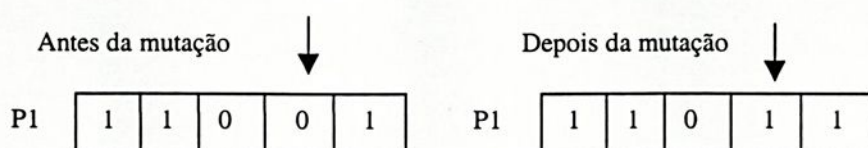


Figura 4.4: Exemplo de Mutação

Após efetuar mutação (o último operador genético), as configurações estão sujeitas a se tornarem inactíveis. Para tal situação é necessário formular alguma estratégia permitindo transformar em factíveis aquelas configurações inactíveis. Uma outra estratégia a ser empregada é considerar todas essas configurações como sendo “factíveis” e penalizar as inactibilidades na função objetivo para que sejam eliminadas pelo operador de seleção.

4.2.5. Parâmetros de Controle do AG

O ciclo geracional compreende os processos de seleção, recombinação e mutação (correção da inactibilidade), permitindo encontrar as configurações da nova geração a partir da população corrente. O ciclo geracional é realizado com base no programa de controle do algoritmo genético, um conjunto de parâmetros, os quais definem o tamanho da população, a taxa de recombinação e a taxa de mutação. Estes parâmetros variam de problema para problema e, em grande medida, definem a qualidade do algoritmo.

O tamanho da população afeta o desempenho global e a eficiência dos algoritmos. Com uma população pequena o desempenho tende a cair, pois deste modo a população fornece uma pequena cobertura do espaço de busca do problema. Uma população grande geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais.

Quanto maior for a taxa de recombinação mais rápidas novas estruturas serão introduzidas na população. Mas se esta for muito alta, estruturas com boa qualidade poderão ser perdidas durante o processo. Com um valor baixo, o algoritmo tornar-se muito lento. Uma alta taxa de mutação previne que uma dada posição fique estagnada em um valor, além de abranger outro espaço de busca.

Valores típicos usados na literatura especializada para os parâmetros de controle são:

- Tamanho da população entre 30 e 200.
- Taxa de recombinação entre 0,5 e 1,0 .
- Taxa de mutação entre 0,001 e 0,050.

4.2.6. Critério de Parada

Existem vários critérios de parada. Assim, deve-se interromper o processo do algoritmo quando:

- Foi executado um número determinado de gerações.
- A incumbente, isto é, a melhor solução encontrada, atingiu um valor especificado.
- As configurações da população ficaram muito homogêneas.

Em problemas reais, são relatados vários critérios de parada. No problema de planejamento de sistemas de transmissão um critério de parada importante é o número máximo de PL's executados.



4.2.7. Teoria do Esquema (“Schema Theory”)

Um esquema representa um subconjunto de todas as possibilidades de uma codificação binária de configurações com algumas posições fixas. Por exemplo, um esquema seria o seguinte arranjo:

*	*	0	0	1
---	---	---	---	---

Figura 4.5: Representação de um Esquema

Acima, tem-se a representação de uma configuração com cinco casas binárias, sendo três casas fixas e duas casas indefinidas podendo assumir valores 0 ou 1.

Uma configuração binária com cinco posições permite representar 32 configurações diferentes (2^5). Contudo, a figura 4.5 apresenta um subconjunto destas 32 configurações, tendo 3 posições fixas. Este esquema apresenta as seguintes configurações:

0	0	0	0	1
1	0	0	0	1
1	1	0	0	1
0	1	0	0	1

Figura 4.6: Configurações de um Esquema

O esquema representa um conjunto de configurações com algumas características em comuns (no exemplo anterior foram representadas 4 configurações usando um mesmo esquema). Essas características comuns são denominadas *Similarite Template* (conjunto de características ou propriedades análogas).

*	*	*	*	*
---	---	---	---	---

Figura 4.7: Representação do Espaço Completo de Configurações.

O esquema da figura 4.7 representa o espaço completo de configurações.

Define-se a *ordem de um esquema* como sendo o número de posições fixas do mesmo. A figura 4.8 representa um esquema de ordem três.

1	*	*	*	0	1
---	---	---	---	---	---

Figura 4.8: Representação de um Esquema de Ordem 3.

O *comprimento de definição de um esquema* ("schema defining length"), isto é, a distância das posições fixas mais distantes. O esquema da figura 4.9 possui comprimento 5.

1	*	*	*	1	0
---	---	---	---	---	---

Figura 4.9: Comprimento de um Esquema

A *função de adaptação média de um esquema* ("average fitness of the schema") é o valor médio das funções objetivo ou o seu equivalente de todas as configurações representadas por ele.

Esquema de competição ("schema of competition"). Considerando que exista um esquema de r posições fixas como o seguinte (* 0 1), neste caso $r = 2$. Então, existem $2^r - 1 = 2^2 - 1 = 3$ outros esquemas com as mesmas posições fixas, os quais podem ser obtidos trocando os valores 0 - 1 fixos. Para uma configuração de k casas binárias é possível construir 2^k *esquemas de competição*. Dessa maneira, para $k = 3$, as configurações com três posições existem $2^3 = 8$ *esquemas de competição*. O algoritmo genético gera esquemas de competição e tenta analisar de alguma forma os 2^k esquemas de competição e encontra o melhor para cada conjunto de posições fixas.

Um algoritmo genético produz uma competição entre esquemas, 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").

4.2.8. O Conceito de Blocos Construtivos (“Building Blocks”)

É chamado de bloco construtivo aquele *esquema* com valor médio da função objetivo de excelente qualidade e também um comprimento de definição pequeno (informação concentrada).

Os operadores de recombinação e mutação geram, selecionam, e fazem sobreposições de blocos construtivos para encontrar configurações de excelentes qualidades (freqüentemente o ótimo). A recombinação tende a preservar a informação genética presente nas configurações cruzadas (exceto no ponto de recombinação). A recombinação de duas configurações parecidas perde capacidade de gerar blocos construtivos. Mutação pode gerar (ou destruir) novos blocos construtivos.

Pode-se resumir o processo de geração de blocos construtivos da seguinte forma:

- Seleção: escolhe as configurações com maior número de blocos construtivos ao selecionar as configurações de melhor qualidade.
- Recombinação: realiza a superposição de blocos construtivos e, por esta razão, a teoria básica de algoritmos genéticos considera este operador como o principal gerador de configurações com blocos construtivos de alta qualidade.
- Mutação: gera novos blocos construtivos.

Observações:

Nem sempre o operador de recombinação das configurações de excelente qualidade geram outras configurações de melhor qualidade. Pode acontecer o contrário, pois, nem sempre a superposição de blocos construtivos de alta qualidade gera novos blocos construtivos de melhor qualidade. Problemas cujas funções de adaptação (função objetivo) apresentam este tipo de comportamento são chamadas de funções objetivo enganosas (“deceptive function”).

Em um simples ciclo geracional o AG processa somente um número de P configurações (P é o tamanho da população), mas avalia implicitamente um número



elevado de esquemas (aproximadamente P^3). Esta capacidade de processar um número elevado de esquemas, é conhecido como *paralelismo implícito*.

4.3. Conclusões

Apresentou-se a teoria básica sobre os algoritmos genéticos, analisando-se os aspectos mais importantes como os operadores genéticos e o conceito de “building blocks”. Destaca-se as diferentes formas de codificação encontradas na literatura. Para uma informação mais detalhada ver as referências [12-16]. O operador seleção é discutido no próximo capítulo.



CAPÍTULO 5

O Operador Genético de Seleção

Neste capítulo será analisado os diferentes tipos de seleção apresentados na literatura especializada, bem como alguns conceitos importantes, tais como: pressão seletiva, tempo de “take over”, ruído, etc, os quais são usados para comparar e determinar o desempenho dos diferentes tipos de seleção.

5.1. Introdução

Nos primeiros anos de pesquisa dos algoritmos genéticos o operador seleção foi considerado um operador secundário, sendo implementado somente a seleção proporcional. Pesquisas mais recentes mostram que esse contexto vem sendo modificado, em que especialistas passaram a questionar o papel do operador seleção, surgindo outras propostas de seleção.

Na literatura especializada encontra-se diferentes tipos de seleção: alguns ordenam as configurações, outros realizam torneios, já os mais antigos utilizam esquemas como a seleção da roleta. Mas não basta implementar determinado tipo de seleção, é necessário analisar o tipo de problema a ser otimizado e o desempenho de cada tipo de seleção aplicado. Em consequência disso, pesquisadores como Goldberg e

Deb [15], entre outros, desenvolveram critérios para avaliar o desempenho das diferentes propostas de seleção mostrando que a qualidade de um tipo de seleção é determinada analisando-se 4 critérios de desempenho: tempo de "take over", complexidade computacional para implementar a seleção, taxa de crescimento da melhor configuração e efeito do ruído na determinação da função objetivo. Estes critérios serão apresentados separadamente, sendo discutido a importância de cada um para a solução do planejamento de sistemas de transmissão, como também as diferentes técnicas de seleção encontradas na literatura.

5.2. Análise de Técnicas de Seleção

O operador seleção é um dos passos mais importantes do algoritmo genético por ser capaz de selecionar os melhores indivíduos (configurações) e repassá-los às novas configurações garantindo assim a qualidade e o bom desempenho do algoritmo. Nos últimos anos o operador de seleção foi intensamente estudado surgindo formas alternativas de implementar seleção. São analisados, a seguir diferentes tipos de seleção encontrados na literatura especializada.

5.2.1. Propostas Baseadas em Seleção Proporcional

O algoritmo genético básico utiliza como operador de seleção o mecanismo de seleção proporcional, considerado como um dos pioneiros nas pesquisas sobre algoritmo genético. Sua metodologia está baseada no princípio de seleção natural (a sobrevivência dos mais aptos ao meio competitivo) e alicerçando-se nesses princípios novas alternativas de seleção foram criadas, como visto a seguir.

5.2.1.1. Seleção Proporcional Básica

A forma mais simples de se implementar seleção é a chamada seleção proporcional. Neste esquema de seleção cada configuração tem direito a gerar um



número de descendentes proporcional a qualidade de sua função objetivo. Configurações com excelente função objetivo terão a chance de gerar mais descendentes comparadas àquelas de pior qualidade. Calcula-se o número de descendentes Nd_i atribuído a cada configuração dividindo o valor da função objetivo $Z_i(x)$ de cada configuração pela média total da população $Z_m(x)$. Tem-se a seguinte relação:

$$N^{\circ} \text{ de descendentes} = \frac{f.o.}{\text{média da } f.o.} \Rightarrow Nd_i = \frac{Z_i(x)}{Z_m(x)}$$

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

e por tanto:

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

N_i : número de descendentes

$Z_i(x)$: valor da função objetivo de cada configuração

N : número de configurações

$Z_m(x)$: média da população

Exemplo: Dada uma população com 4 configurações de um problema de maximização, calcula-se o número de descendentes atribuído a cada configuração.

Configurações	$f(x)$	Nd (Seleção Proporcional)
1	13	1.3
2	09	0.9
3	03	0.3
4	15	1.5

Tabela 5.1: Seleção Proporcional

$$Z_m(x) = 40$$

Obviamente, espera-se que o número de descendentes atribuído à cada configuração seja um número inteiro. Encontrar um número de descendentes não

inteiro é uma característica da seleção proporcional e da maioria das técnicas de seleção, por esse motivo, divide-se o esquema de seleção proporcional em dois passos:

- 1º Passo: Obter o número de descendentes Nd_i (não inteiro) gerados por cada configuração.
- 2º Passo: Transformar esse número em inteiro.

Uma proposta inicial para resolver o problema do número de descendentes não inteiro é utilizar o mecanismo de seleção da roleta.

No esquema de seleção da roleta é designada a cada configuração um setor circular proporcional a sua função objetivo e são realizados n giros até gerar a nova população. Faz-se uso da seguinte relação:

$$2\pi\left(\frac{1}{n}\right)Nd_i \Leftrightarrow 360\left(\frac{1}{n}\right)Nd_i$$

dividindo dessa maneira, a roleta em graus ou radianos.

Exemplo: Implementar o esquema da roleta para a função apresentada no exemplo anterior.

Usando os resultados apresentados no exemplo anterior calcula-se a zona de influência de cada configuração da roleta:

$$P1 \Rightarrow \frac{1.3}{4}360 = 117^\circ$$

$$P2 \Rightarrow \frac{0.9}{4}360 = 81^\circ$$

$$P3 \Rightarrow \frac{0.3}{4}360 = 27^\circ$$

$$P4 \Rightarrow \frac{1.5}{4}360 = 135^\circ$$

Melhora-se a apresentação da roda da roleta, dividindo os valores encontrados em $P1, P2, P3, P4$ por 360° , o que levaria a eliminar a roleta (convencional). Para implementar o esquema da roleta em um computador pode-se usar um único vetor além de um gerador de números aleatórios no intervalo de $[0,1]$.



Uma vez montada a roleta deve-se girar, aleatoriamente, um número entre 0° - 360° . Esse número permite identificar uma região da roleta e a configuração que corresponde a essa região, conhecendo assim o número de descendentes dessa configuração. Esse processo é executado n vezes com a finalidade de gerar a nova população corretamente.

Exemplo: Terminar o processo da seleção do exemplo anterior usando a roleta padronizada.

Usando um gerador de números aleatórios no intervalo $[0, 1]$ para encontrar o número de descendentes de cada configuração:

Nº aleatório gerado	Configuração escolhida
0,42	2
0,18	1
0,74	4
0,96	4

Tabela 5.2: Início do Processo de Seleção Utilizando a Roleta Padronizada.

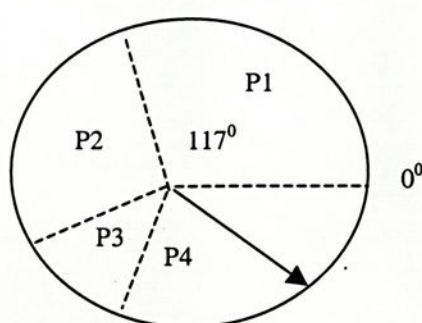


Figura 5.1: Divisão da Roleta

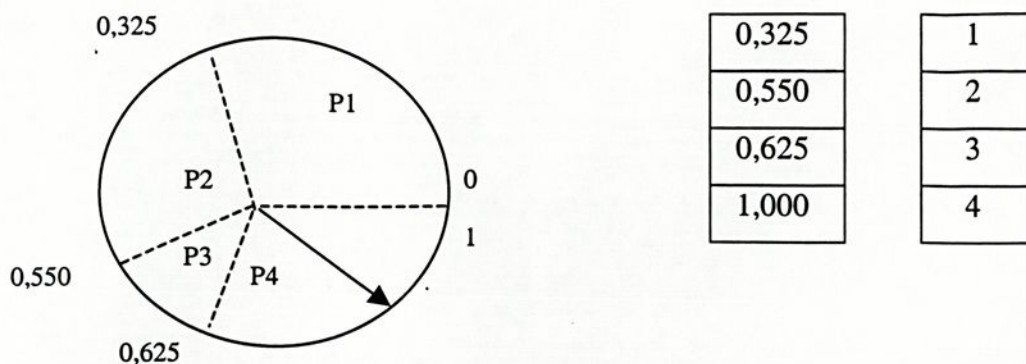


Figura 5.2: Divisão da Roleta Padronizada

O processo de seleção termina com a seguinte distribuição de descendentes:

Configuração escolhida	Nº e descendentes
P1	1
P2	1
P3	0
P4	2

Tabela 5.3: Distribuição do Número de Descendentes para Cada Configuração

Embora muito utilizada em diversos problemas, a seleção proporcional possui alguns pontos críticos os quais devem ser analisados:

- Em determinados problemas podem existir valores de função de adaptação negativas. Por exemplo, suponha-se que as funções de adaptação de uma população de quatro configurações sejam as seguintes: -30,-10,15 e 25. O valor médio das 4 configurações de adaptação é igual a zero e não seria possível usar a mesma estratégia do exemplo anterior. Na realidade, deve-se buscar outro critério de seleção ou ter em mãos um mecanismo para eliminar esse problema. Uma saída seria fazer um deslocamento dos valores de função objetivo somando-se a todas as funções de adaptação uma constante, por exemplo, $K=30$ para transformar as funções de adaptação em números positivos.
- Nas fases iniciais do processo a maioria das configurações possui função objetivo de baixa qualidade. Nestas condições, o aparecimento de algumas

super configurações (f.o. de excelente qualidade) pode eliminar prematuramente a diversidade genética ocasionando uma convergência prematura.

- Em contra posição, nas fases finais do processo as configurações tendem ser muito homogêneas, ou seja, possuem funções objetivo de qualidades semelhantes, perdendo assim, seletividade (quase todas as configurações tem direito a gerar descendentes).

Para implementar seleção proporcional básica não é necessário escolher ou calcular nenhum parâmetro de controle adicional.

Algumas novas propostas de seleção são consideradas variantes da seleção proporcional porque modificam apenas um dos passos da mesma, ou seja, o primeiro passo em que se calcula Nd_i (número de descendentes não inteiros) é executado em todas as alternativas da seleção proporcional; o segundo passo onde é transformado o número de descendentes não inteiros em números inteiros é obtido de acordo com cada estratégia proposta abaixo:

5.2.1.2. Seleção Estocástica do Resíduo

Cada configuração recebe de forma determinística, a parte inteira do número de descendentes encontrada na seleção proporcional. A parte residual (não inteira) é completada usando o mecanismo da roleta com a finalidade de gerar o número correto de descendentes da nova população (destaca-se mais uma vez, o caráter aleatório do algoritmo genético).

Esta proposta acelera consideravelmente o processo de seleção, pois a roleta é utilizada apenas para terminar o processo de seleção. Observa-se nesta proposta a equivalência entre a seleção proporcional no aspecto relacionado a capacidade de seleção, especialmente quando o número de configurações da população é grande. A diferença fundamental está relacionada com a rapidez do processo de implementação.

Exemplo: Implementar a seleção estocástica do resíduo para um problema com uma população de 4 configurações cujas funções de adaptação são as seguintes: 43, 39, 33 e 45.



Configurações	F(x)	N _d	Seleção determinística	Seleção aleatória usando a roleta
P1	43	1.3	1	0.3
P2	39	0.9	0	0.9
P3	33	0.3	0	0.3
P4	45	1.5	1	0.5

Tabela 5.4: Seleção Estocástica do Resíduo

Seleciona-se somente duas configurações usando a roleta (duas configurações foram selecionadas de forma determinísticas - P1 e P4). A roleta selecionará provavelmente as configurações P2 e P4 (são configurações com maior valor fracionário) para completar o processo de seleção. Para implementar a seleção estocástica do resíduo não é necessário escolher ou calcular nenhum parâmetro de controle adicional.

Observação: Se desejável, pode-se especificar o número máximo de descendentes para a melhor configuração. Caso isso ocorra, é necessário executar a roleta um número maior de vezes, pois, se a melhor configuração gerou todos os seus descendentes e a mesma foi novamente selecionada, precisa-se descartar esse giro e executar o procedimento novamente para obter uma configuração que não tenha completado (gerado) o número de descendentes. Esse processo ocorre com todas as estratégias utilizando limitante superior.

5.2.1.3. Seleção Proporcional com Número Máximo de Descendentes

Esta estratégia tem por objetivo fixar o número máximo de descendentes gerados pela configuração. Inicialmente sabe-se o número máximo de descendentes atribuído a melhor configuração, bem como, sua participação na formação da nova população. Essa estratégia elimina, em parte, o problema de super configurações nas fases iniciais do processo, e portanto, constitui uma proposta um pouco diferente da seleção proporcional.

A implementação desse mecanismo de seleção, por outro lado, é apenas uma variante do esquema de seleção proporcional, logo seria suficiente usar a roleta com



uma barreira superior para o número de descendentes. Pode-se também implementar a seleção estocástica do resíduo com limitante superior, surgindo uma outra alternativa de seleção que seria a seleção estocástica do resíduo com limitante superior ao número de descendentes.

Para implementar a seleção proporcional com número máximo de descendentes é necessário escolher um parâmetro de controle N_{km} - número máximo de descendentes atribuído às melhores configurações .

5.2.1.4. Seleção Determinística

Cada configuração recebe de maneira determinística a parte inteira do número de descendentes calculado pela seleção proporcional (referindo-se ao primeiro passo da seleção proporcional). A parte fracionária é atribuída novamente de forma determinística, iniciando o processo com aquelas que possuem maior valor fracionário, com intuito de gerar o número correto de descendentes da nova população. Como essa estratégia não requer o uso da roleta e nem a geração de números aleatórios, sua execução torna-se mais rápida em relação ao tempo computacional gasto pelas outras estratégias de seleção (esta proposta foge da base teórica do algoritmo genético).

É permitido adicionar uma limitante superior ao número máximo de descendentes para contornar o problema de super configurações nas fases iniciais do processo, essa estratégia garante bons resultados principalmente em sistemas de grande porte.

Para implementar a seleção determinística não é necessário escolher um parâmetro de controle adicional. Entretanto, pode-se fixar um número máximo de descendentes à melhor configuração e nesse caso deve ser escolhido o parâmetro N_{km} .

Exemplo: Implementar seleção determinística para uma população com quatro configurações e cujas funções de adaptação são as seguintes: 43,39,33 e 45.



Configurações	F(x)	N _d	Parte inteira	Parte fracionária	Nd'	Total de descendentes
P1	43	1.3	1	0.3	0	1
P2	39	0.9	-	0.9	1	1
P3	33	0.3	-	0.3	0	0
P4	45	1.5	1	0.5	1	2

Tabela 5.5: Implementação da Seleção Determinística

5.2.2. Seleção Usando Escalonamento Linear

O escalonamento linear tenta atenuar problemas bem conhecidos na seleção proporcional: o surgimento de super configurações, a perda de seletividade e homogeneidade. A idéia consiste fundamentalmente em modificar os valores das funções objetivo encontrando novas funções de adaptação. Quando os valores das funções objetivo são muito distantes pode surgir uma super configuração (ocasionando uma possível convergência a um ótimo local), tenta-se diminuir essas diferenças de valores. Quando os valores das funções objetivo são muito semelhantes, tenta-se modificá-los afim de se tornarem diferentes e garantir a seletividade.

Uma proposta simples a ser implementada consiste em realizar um escalonamento linear encontrando novas funções de adaptação usando a seguinte relação:

$$f' = af + b$$

em que:

f : valor da função objetivo original

f' : nova função objetivo

a e b : constantes a serem determinadas

Uma forma simples de calcular os valores de a e b é escolher dois pontos característicos, tais como:

- O maior valor de f' , correspondente a função de adaptação de maior valor que é igual a um número de vezes o valor médio da população, tipicamente 1,2 a 2,0 vezes.

- O valor médio de ambas populações permanecem constantes.

Exemplo: Implementar a seleção usando escalonamento para um problema com 4 configurações: 40, 60, 80 e 360.

Observa-se a presença de uma super configuração (P4):

Configurações	f(x)
P1	40
P2	60
P3	80
P4	360
Total	540

Tabela 5.6: Configurações da População

Faz-se um escalonamento com a condição de que $f'_{\text{melhor}} = 2\bar{f}_m$, sendo $\bar{f}_m$ a média dos valores das funções objetivo.

$$\bar{f}_m = \frac{1}{4}(40 + 60 + 80 + 360)$$

$$\bar{f}_m = 135$$

$$f'_{\text{melhor}} = 270$$

Para calcular **a** e **b** são necessários dois pontos:

$$f = 360 \Rightarrow f' = 270$$

$f = 135 \Rightarrow f' = 135$ (o valor médio da função objetivo da população não deve ser corrigido), desse modo pode-se montar o seguinte sistema:

$$\begin{cases} 270 = 360a + b \\ 135 = 135a + b \end{cases}$$

$$\text{obtendo: } a = \frac{3}{5} \text{ e } b = 54, \text{ fornecendo: } f' = \frac{3}{5}f + 54$$

Configurações	F(x)	Nd	F'(x)	N'd
P1	40	0.30	78	0.57
P2	60	0.44	90	0.66
P3	80	0.60	102	0.77
P4	360	2.66	270	2.00

Tabela 5.7: Número de Descendentes

Nos resultados apresentados pela seleção usando escalonamento linear observa-se que as funções objetivo com valores acima da média perderam valores e as funções objetivo abaixo da média tiveram um acréscimo em seus valores.

Observações:

- O processo de seleção ainda não está terminado, apenas foi modificado a função objetivo. A segunda parte, consiste em selecionar um número inteiro de descendentes a cada configuração (pode ser implementada usando-se a roleta ou alguma das formas alternativas antes mencionadas).
- A seleção por escalonamento linear contorna os problemas tipicamente encontrados na seleção proporcional, diminuindo as vantagens das super configurações nas fases iniciais do processo e aumentando a dispersão das funções objetivo nas fases finais do processo.
- O escalonamento linear pode produzir, em casos especiais, funções objetivo com valores negativos. Assim, é necessário realizar uma correção, por exemplo, adicionando-se um valor a todas as funções objetivo para que não se tornem negativas.

Para implementar a seleção usando escalonamento linear, deve-se escolher um parâmetro de controle N_{ke} - número de vezes que a melhor função objetivo escalonada deve ser em relação ao valor médio das funções objetivo. Pode-se especificar o número máximo de descendentes.



5.2.3. Seleção Baseada em Ordenamento (“Rank Based Selection”)

O número de descendentes para cada configuração depende de uma seleção ordenada (“ranking”) das configurações em ordem crescente dos valores das funções objetivo da população (para problemas de minimização). O número de descendentes de cada configuração varia linearmente de forma crescente com sua localização no ranking. O valor da função objetivo é usado apenas para encontrar a posição da configuração dentro do ordenamento. Nota-se que essa estratégia elimina os problemas de seleção proporcional porque neste esquema a função objetivo não tem papel direto no número de descendentes (apenas para saber a posição relativa da função objetivo em relação as outras).

A função de designação linear $\alpha(x)$ para realizar escalonamento linear deve cumprir algumas propriedades:

1. $\alpha(x) \in R$ para $x \in [0,1]$
2. $\alpha(x) \geq 0$
3. $\int_0^1 \alpha(x) dx = 1$

Seja a forma geral de $\alpha(x) = c_0 - c_1 x$. Tem-se:

$$\begin{aligned} \int_0^1 \alpha(x) dx = 1 &\Rightarrow \int_0^1 (c_0 - c_1 x) dx = \left[c_0 x - \frac{c_1}{2} x^2 \right]_0^1 = 1 \\ \Rightarrow c_0 - \frac{c_1}{2} &= 1 \Rightarrow c_1 = 2(c_0 - 1) \Rightarrow \alpha(x) = c_0 - 2(c_0 - 1)x \end{aligned}$$

Exemplo: Implementar o mecanismo de seleção usando um ordenamento linear para um problema de minimização com uma população de 4 configurações cujas funções objetivo são as seguintes: 43, 39, 33 e 45.

Primeiramente ordenam-se as configurações em ordem crescente em relação aos valores de suas funções objetivo. Assim, temos:

$$P1 = 33$$

$$P2 = 39$$

$$P3 = 43$$

$$P4 = 45$$

Escolhe-se a função de designação com $c_0 = 2$ (valor que deve ser escolhido pelo pesquisador) e separa-se o intervalo de variação de x em 4 partes (4 configurações).

Seja α_1 a fração correspondente a melhor configuração (para o problema de minimização, a melhor configuração é P1):

$$\alpha_1 = \int_0^{0.25} \alpha(x) dx = \int_0^{0.25} (2-2x) dx = [2x - x^2]_0^{0.25} = \frac{7}{16}$$

De maneira similar tem-se:

$$\alpha_2 = \int_{0.25}^{0.5} \alpha(x) dx = \int_{0.25}^{0.5} (2-2x) dx = [2x - x^2]_{0.25}^{0.5} = \frac{5}{16}$$

$$\alpha_3 = \int_{0.5}^{0.75} (2-2x) dx = [2x - x^2]_{0.5}^{0.75} = \frac{3}{16}$$

$$\alpha_4 = \int_{0.75}^1 (2-2x) dx = [2x - x^2]_{0.75}^1 = \frac{1}{16}$$

O número de descendentes de cada configuração deve ser igual a $\alpha_k n$ com $k = 1,2,3,4$. Pode-se montar o seguinte quadro dos resultados obtidos:

Configuração	$f(x)$	α_i	$N_d = 4 \alpha_i$
P1	33	7/16	1,75
P2	39	5/16	1,25
P3	43	3/16	0,75
P4	45	1/16	1,25
Total			4,00

Tabela 5.8: Implementação do Mecanismo de Seleção Usando Ordenamento Linear

Logicamente, o número de descendentes não depende do valor absoluto das funções de adaptação, mas do ordenamento e do valor escolhido para a constante c_0 . Para terminar o processo de seleção, utiliza-se a roleta ou alguma estratégia apresentada anteriormente.

Em implementações práticas pode-se estabelecer facilmente uma regra determinando o número de descendentes de cada configuração ordenada de forma crescente à qualidade de sua função de adaptação ajustada a uma função linear decrescente. Assim, existe uma alternativa de implementação prática de $\alpha(x)$. Geralmente, encontra-se essa relação linear escolhendo alguns parâmetros, por exemplo, o número de descendentes atribuído a melhor configuração p_m , o número de configurações participantes na geração dos descendentes n_{pr} . Com esses parâmetros deve-se conhecer a relação linear decrescente permitindo calcular o número de descendentes de cada configuração considerada para seleção, ρ_m .

Os parâmetros p_m , n_p podem ser previamente selecionados antes de iniciar o processo de otimização. São vantagens desta proposta:

- Os parâmetros anteriores podem ser adaptados e modificados durante o processo.
- A proposta de seleção contorna adequadamente os problemas mencionados anteriormente, persistindo a seletividade, podendo ser modificada e adaptada por não depender dos valores absolutos das funções objetivo e sim da posição no "ranking".
- A proposta de seleção elimina a discussão entre problemas de minimização e maximização (o problema é contornado invertendo-se o ordenamento).

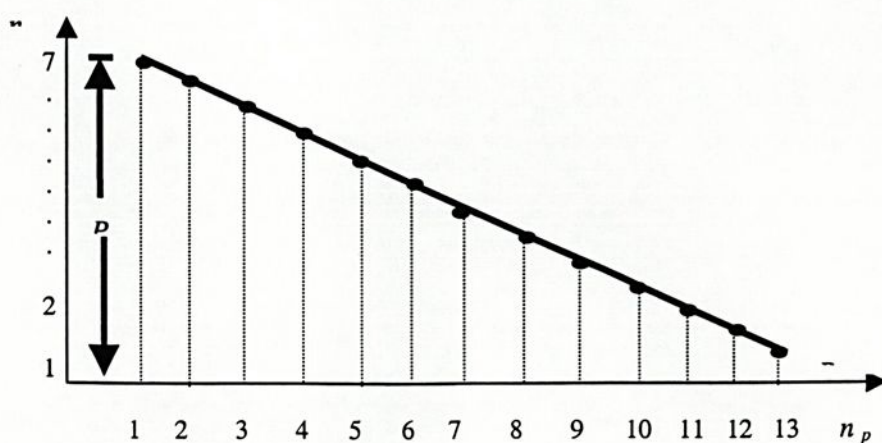


Figura 5.3: Seleção Baseada em Ordenamento.

As vantagens propostas anteriormente, tornam a seleção baseada em ordenamento muito atrativa, porém ainda persiste um problema: o número de descendentes nem sempre é inteiro. Este problema será contornado usando a roleta ou outra estratégia equivalente.

Exemplo: Implementar a seleção alternativa usando ordenamento linear para um problema de maximização com uma população de 20 configurações e onde a melhor configuração deve receber 4 descendentes e somente as 10 melhores configurações devem participar da seleção.

Somente as 10 configurações terão direito a gerar descendentes e as demais são eliminadas (tanto o número de configurações participantes, como o número máximo de configuração ganhadoras são parâmetros escolhidos pelo pesquisador).

Seja δ_i a variável que identifica cada uma das 10 configurações, $\delta_i, i=1,2,3,\dots,10$. Seja n_i o número de descendentes de cada configuração participantes da seleção e ρ o número de configurações da última configuração considerada para seleção (a décima neste caso), a seguinte relação pode ser deduzida:

$$n_i = 4 - \frac{\delta_i - 1}{9}(4 - \rho) \quad \delta_i = 1, 2, \dots, 10$$

como $\sum_{i=1}^{n=10} n_i = 40$ é o tamanho da população, encontra-se $\rho = 0$. O número de descendentes correspondente a cada uma das 10 primeiras configurações com direito a participar da seleção pode ser encontrado usando a seguinte relação:

$$n_i = 4 - \frac{4}{9}(\delta_i - 1) \quad \delta_i = 1, 2, \dots, 10$$

cujos resultados são resumidos no seguinte quadro.



Configurações	Nd = n _i
P1	4.0
P2	3.5
P3	3.1
P4	2.7
P5	2.2
P6	1.8
P7	1.3
P8	0.9
P9	0.5
P10	0.0

Tabela 5.9: Atribuição do Número de Descendentes para Cada Configuração

Não foram mencionados os valores específicos das funções de adaptação (função objetivo). Precisa-se terminar o processo de seleção, uma vez que ainda existem números de descendentes não inteiros; como sugestão aconselha-se alguns dos mecanismos propostos anteriormente.

Para implementar a seleção alternativa baseada em ranking, é necessário escolher dois parâmetros: ρ_M - número máximo de descendentes correspondentes a melhor configuração e n_{pt} - número de configurações participantes na seleção (se assumimos que $n=n_{pt}$ então, somente precisa escolher ρ_M ou o parâmetro c_0). Logicamente, existe a proposta alternativa que consiste em escolher o parâmetro c_0 .

5.2.4. Seleção Baseada em Torneio

As configurações são escolhidas mediante jogos (torneios) e a quantidade de torneios é equivalente ao tamanho da população, tornando essa proposta significativamente diferente da seleção proporcional. A cada jogo são escolhidos

aleatoriamente um conjunto de k configurações e a configuração ganhadora é aquela com melhor função objetivo. O valor de k geralmente é pequeno, tipicamente $k \in \{2,3,4,5\}$. Realiza-se n jogos.

Uma vantagem oferecida pelo mecanismo de torneio é o pequeno esforço computacional (a estratégia é a mesma tanto para problemas de minimização como para problemas de maximização), apenas o critério de melhor valor da função objetivo é diferente. Este esquema contorna adequadamente os problemas existentes na seleção proporcional, uma vez, que a seleção de torneio não necessita dos valores das funções objetivo de cada configuração e além disso, não existe o problema de números de descendentes não inteiros típico da seleção proporcional.

3.2.5. Algoritmo Genitor

Os diferentes tipos de seleção mencionados anteriormente são chamados de geracionais pois de uma geração para outra substituem integralmente todas as configurações. Algoritmos genéticos que substituem somente uma parte das configurações em cada geração são chamados algoritmos genéticos de estado fixo ou simplesmente algoritmos genitores.

O algoritmo genitor procura preservar as melhores configurações modificando apenas as piores e substituindo-as por configurações geradas da recombinação ou mutação. Quando se conservam as melhores configurações aparece o conceito de elitismo, o qual significa dizer que as melhores configurações são mantidas. Esse comportamento compromete o trabalho da recombinação e mutação por isso o algoritmo genitor usa grandes populações e altas taxas de recombinação e mutação comparados com outras estratégias de seleção (o elitismo realiza um balanço adequado do poder destrutivo (e construtivo) das elevadas taxas de recombinação e mutação).



5.2.6. Janelamento – “Windowing”

Nos diferentes tipos de seleção analisados neste capítulo, frequentemente depara-se com o problema da estagnação, principalmente naqueles tipos de seleção em que o valor da função objetivo é diretamente empregado no cálculo do número de descendentes, como ocorre, por exemplo, na seleção proporcional. Uma estratégia encontrada para contornar esse problema está apresentada em [13], onde sugere-se utilizar o pior valor da função objetivo observado nas configurações da população e subtrair esse valor de todas as outras, calculando novos valores para as funções objetivo. A configuração pertencente a esse valor será chamada de linha base.

Exemplo: Implementar janelamento na população composta por 5 configurações cujas funções objetivo são as seguintes: 20, 39, 43, 60 e 120.

Configurações	$F(x)$	$F'(x)$	Nd'	Nd
P1	20	0	0	0
P2	39	19	0,52	0
P3	43	23	0,63	1
P4	60	40	1,09	1
P5	120	100	2,77	3

Tabela 5.10: Implementação da Técnica de Janelamento

P1 será usado como linha base (o pior valor encontrado na população), obtendo novos valores para as funções objetivo ($F'(x)$). Pode-se calcular o número de descendentes utilizando algum método de seleção apresentado neste capítulo por exemplo, a seleção determinística.

Observações: Esta estratégia contorna o problema da estagnação, todavia de início atribui a pior configuração nenhum descendente.

5.2.7. Escalonamento Sigma

No processo de seleção[13] a pressão seletiva está diretamente relacionada ao valor atribuído a cada configuração, o escalonamento sigma explora essa observação, mantendo a pressão seletiva ligeiramente constante e o desvio padrão utilizado na obtenção do número de descendentes abaixo da média. Para configurações com função objetivo abaixo do valor do desvio padrão serão atribuídos uma adaptação zero, com uma conseqüente perda de seletividade. O cálculo do número de descendentes atribuído a cada configuração será gerado da seguinte maneira:

$$Nd_i = \begin{cases} 1 + \frac{f(i) - \bar{f}(i)}{2\sigma(i)} & \text{para } \sigma(i) \neq 0 \\ 1 & \text{para } \sigma(i) = 0 \end{cases}$$

$f(i)$ - Função objetivo de cada configuração

$\bar{f}(i)$ - Média da população

$\sigma(i)$ - Desvio padrão, em que $\sigma(i) = \sqrt{\text{var}(i)}$

$$\text{var}(i) = \sum_{i=1}^n (x_i - \bar{x})^2 / n$$

x_i - Função objetivo de cada configuração; $\bar{x}$ - média da população; n - número de configurações, $\bar{f}(i)=75$, $\sigma=29,58$.

Configurações	F(x)	Nd _i	Nd
P1	40	0,408	0
P2	60	0,746	1
P3	80	1,084	1
P4	120	1,760	2

Tabela 5.11: Escalonamento Sigma

5.3. Critérios Teóricos de Desempenho das Técnicas de Seleção

É comum nos métodos de seleção tradicionais, como o esquema de seleção proporcional, surgir problemas bem conhecidos na literatura, como a perda de seletividade e convergência prematura.

A perda de seletividade acontece quando as configurações da população possuem funções objetivo de qualidade semelhante. Possivelmente todas as configurações terão a chance de gerar descendentes e conseqüentemente a nova população será gerada pelas mesmas configurações (tanto as configurações de boa como de má qualidade); quando estas configurações são muito parecidas, a seleção se torna praticamente aleatória aparecendo o problema da perda de seletividade.

A convergência prematura ocorre quando existe na população uma configuração de excelente qualidade. A atuação dessa super configuração pode ser tão grande a ponto de eliminar prematuramente a informação genética do restante da população, ocasionando uma convergência prematura (um ótimo local). O desempenho de um operador de seleção pode ser analisado usando quatro critérios conhecidos como: pressão seletiva, tempo de "take over", complexidade computacional e presença de ruído na função objetivo. Discutiremos estes critérios separadamente.

5.3.1. Pressão Seletiva

A pressão seletiva está relacionada com a importância destinada às melhores configurações pelo operador de seleção. Se um tipo de seleção A fornece um maior número de descendentes com melhores configurações quando comparadas a outro tipo de seleção B, diz-se que essa configuração realiza uma maior pressão seletiva.

Para ilustrar a idéia de pressão seletiva pode-se usar a seguinte população:



Configurações	f(x)	Seleção Proporcional	Seleção por Torneio
P1	20	0.22	0
P2	40	0.44	1
P3	60	0.66	1
P4	240	2.68	2

Tabela 5.12: Esquema de Seleção: Proporcional e Torneio

Na tabela 5.12, a seleção proporcional apresenta maior pressão seletiva. Em geral, pode-se afirmar que quando existem super configurações a seleção proporcional possui elevada pressão seletiva e quando as configurações são muito parecidas, a mesma suporta baixa pressão seletiva. Por outro lado, na seleção baseada em torneio a pressão seletiva não depende dos valores específicos das funções objetivo, pressão permanecendo uniforme durante todo o processo de otimização. Esquematiza-se os seguintes fatos:

Baixa pressão seletiva $\Leftrightarrow$ perda de seletividade

Alta pressão seletiva $\Leftrightarrow$ convergência prematura

Resumindo, sabe-se que nem alta pressão seletiva, nem baixa pressão seletiva são ideais ao desempenho de um mecanismo de seleção. Na seleção proporcional é preciso criar uma estratégia para contornar este tipo de problema.

5.3.2. Tempo de “Take Over”

O conceito de *tempo de “take over”* foi introduzido por Goldberg e Deb [15] com a finalidade de avaliar o desempenho do operador de seleção na ausência de outros operadores, tais como recombinação ou mutação. Supor a existência de uma população com n configurações. O tempo de “take over” é o tempo necessário para gerar uma nova população na qual a melhor configuração substitua todas as outras ($n-1$) de pior qualidade, usando unicamente o operador de seleção. Logicamente, se a

pressão seletiva do operador de seleção usado for elevada o tempo de “take over” deve ser menor se comparado com outro operador de seleção que apresenta uma pressão seletiva menor.

Será ilustrado abaixo um exemplo simples, de uma população constituída por 5 configurações e seus respectivos valores:

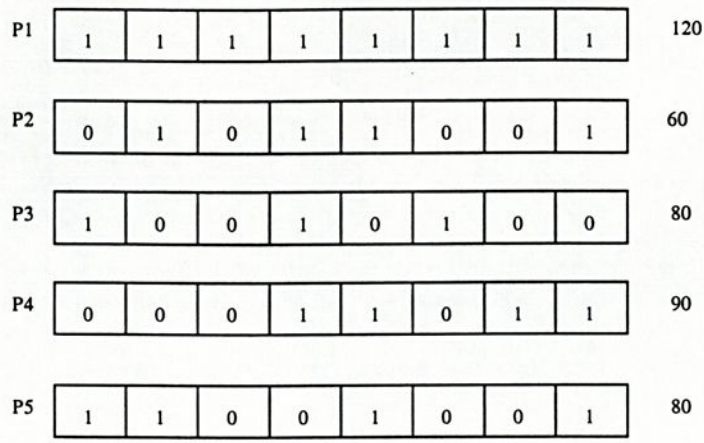


Figura 5.4: Exemplo de População

Na figura 5.4, P1 corresponde a configuração de melhor qualidade presente na população. Para gerar populações de melhor qualidade é necessário repetir de alguma maneira P1, como mostrado a seguir:

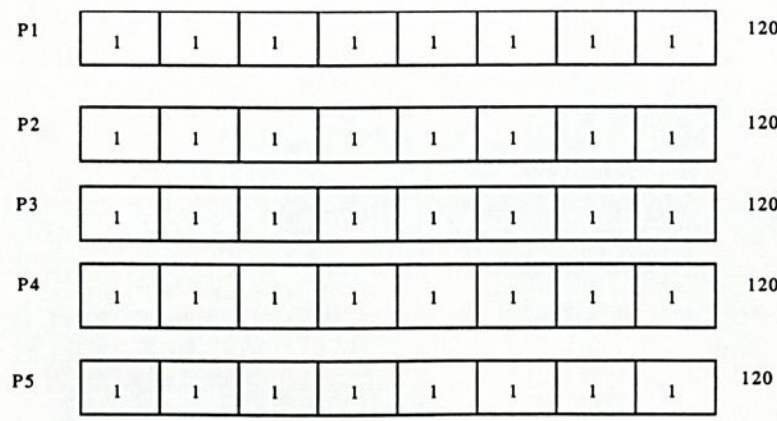


Figura 5.5: População Formada pela Melhor Configuração

Para cada tipo de seleção empregada é especificado um tempo de “take over” segundo suas características. Em algumas propostas (Goldberg e Deb [15]) o tempo de

“take over” é o tempo necessário para que uma população seja formada por $(n-1)$ elementos de n_{best} . As duas definições são usadas indistintamente.

- **Tempo de “Take Over” para Seleção Proporcional:**

Existem dois casos de funções a serem analisadas, as funções do tipo polinomial $f(x)=x^c$, e as funções do tipo exponencial $f(x)=e^{cx}$.

Para as funções polinomiais, o seguinte cálculo para o tempo de “take over” é empregado [13]:

$$ct+1=\frac{1}{1-\frac{\log(n-1)}{\log n}}$$

em que $c \neq 0$ é uma constante.

A expressão $\frac{1}{1-\frac{\log(n-1)}{\log n}}$ pode ser simplificada, expandindo-se $\log(n-1)$ na série de

Taylor em relação ao valor de n , mantendo os dois primeiros termos e substituindo na expressão $\frac{1}{1-\frac{\log(n-1)}{\log n}} \approx n \ln n$, (a aproximação melhora com o aumento de n).

Usando-se esta aproximação, obtem-se o seguinte tempo de “take over” para funções polinomiais:

$$\tau_t = \frac{1}{c}(n \ln n - 1)$$

O tempo de “take over” para funções polinomiais é $O(n \log n)$; para funções exponenciais do tipo $f(x)=e^{cx}$, tem-se o seguinte tempo de “take over” [13]:

$$\tau_t = \frac{1}{c} n \ln n.$$



- Tempo de “Take Over” para Seleção de Ordenamento (“Ranking Linear”):

$$\tau = \frac{2}{c} \ln(n-1)$$

- Tempo de “Take Over” para Seleção de Ordenamento (“Ranking Exponencial”):

$$\tau = \frac{2}{c} \log(n-1)$$

- Tempo de “Take Over” para Seleção Baseada em Torneio:

$$\tau = \frac{1}{\ln q} [\ln n + \ln(\ln n)]$$

onde q identifica o número de torneios executados.

A tabela 5.13 ilustra os diferentes tipos de seleção apresentados nesse trabalho e seus respectivos tempos de “take over” [13, 15]:

Nome	Probabilidade de Seleção	Tempo de “take over”
Proporcional	$\frac{f_i}{\bar{f}}$	$t = \frac{1}{c} (n \ln n - 1)$ para $f(x) = x^c$ $t = \frac{1}{c} (n \ln n)$ para $f(x) = e^{c \cdot x}$
“Ranking” Linear	$N(i) = s - \frac{(2i(s-1))}{n-1}$	$t = \frac{2}{c} \ln(n-1)$
“Ranking” Exponencial	$N(1)=1, N(2)=s, N(3)=s^2 \dots$ $\dots N(n) = s^{n-1}$	$t = \frac{2}{c} \log(n-1)$
Torneio		$t = \frac{1}{\ln q} (\ln n + (\ln n))$

Tabela 5.13: Tempo de “Take Over” Associado a Cada Tipo de Seleção

A interpretação feita a respeito do tempo de “take over” associado à pressão seletiva é portanto: pequenos tempos de “take over” caracterizam forte pressão

seletiva, grandes tempos de “take over” implicam pequena pressão seletiva. O tempo de “take over” é uma consequência da pressão seletiva, isto é, uma forma indireta de qualificar a pressão seletiva que aparece na escolha de cada tipo de seleção.

A seleção proporcional é caracterizada por um grande tempo de “take over” da ordem de $O(n \ln n)$ em contraste com outros métodos de seleção. A seleção proporcional possui pequena pressão seletiva, mas nenhum controle da mesma.

Os tempos de “take over” para as demais estratégias seguem aumentando, respectivamente, da seleção de torneio para seleção de ranking e finalizando com a seleção de escalonamento. A seleção de “ranking” apresenta alta pressão seletiva controlando-a satisfatoriamente durante o processo de seleção. Essas justificativas são enfatizadas por alguns pesquisadores como fortes argumentos para adotar o método de seleção baseada em “ranking” [13] como sendo o mais eficiente.

5.3.3. Complexidade Computacional

Complexidade computacional é o esforço computacional necessário para terminar de implementar um processo de seleção, isto é, fornece uma medida do tempo necessário para escolher o número de descendentes correspondente a cada uma das configurações.

Existem diferentes tipos de seleção, sendo que alguns são executados em 2 passos, como o caso da seleção baseada em “ranking”, enquanto outros necessitam de vários passos como o mecanismo de seleção proporcional. Em seus trabalhos, Goldberg e Deb [15] analisaram esta questão, discutindo o esforço computacional necessário para a execução dos vários mecanismos de seleção, com o intuito de estabelecer qual a estratégia mais rápida para implementar a seleção, dessa forma tem-se:

- **Complexidade Computacional da Seleção Proporcional:**

O modelo de seleção proporcional por geração pode ser implementado de diversos modos. Uma implementação simples é simulada usando a roleta. Se a procura pelo setor circular proporcional a sua função de adaptação for executada via procura



linear, cada seleção requer $O(n)$ passos. De uma maneira geral, a roleta executa este método em $O(n^2)$ passos, pois, são necessários n giros para gerar o número total de descendentes.

A seleção proporcional pode também ser executada pela seleção estocástica do resíduo. Aqui o número esperado de descendentes é calculado pela divisão $\frac{f_i}{\bar{f}}$, sendo que a parte inteira do número de descendentes é determinada de maneira determinística e a parte residual de maneira probabilística afim de completar o número correto de descendentes.

- **Complexidade Computacional da Seleção por Ordenamento (“Ranking”):**

O mecanismo de seleção é executado em 2 passos e o cálculo da complexidade computacional do “ranking” requer a consideração desses passos separadamente. Primeiramente uma lista de indivíduos é sorteada, sendo que esta escolha dos indivíduos deve ser feita de alguma forma a satisfazer as condições específicas do problema. Essa escolha pode ser realizada em $O(n \log n)$ passos. O próximo passo necessita de uma estratégia para atribuir o número de descendentes a cada configuração, utilizando-se para isso, podemos utilizar o esquema de seleção proporcional.

- **Complexidade Computacional da Seleção de Torneio:**

Cada seleção do torneio requer uma seleção aleatória de um número constante de indivíduos da população (estabelecido pelo tamanho do problema). A comparação entre esses indivíduos pode ser executada em tempo constante, e tais n competições são realizadas para gerar o número correto de descendentes. A seleção de torneio leva $O(n)$ passos.



Nome	Complexidade Computacional
Seleção Proporcional	$O(n^2)$
"Ranking" Linear	$O(n \ln n) + \text{tempo da seleção}$
Seleção estocástica do Resíduo	$O(n)$
Seleção Proporcional com Número Máximo de Descendentes	$O(n)$
Seleção Proporcional Determinística	$O(n)$
Torneio	$O(n)$

Tabela 5.14: Complexidade Computacional Associada a Cada Tipo de Seleção

A tabela 5.14 ilustra as diferentes complexidades computacionais apresentadas nos diversos esquemas de seleção em [15]. Os melhores métodos são aqueles executados em $O(n)$ passos, devendo-se contudo destacar em particular o desempenho da seleção de torneio que mesmo utilizando vários passos, pode ser executado em máquinas paralelas reduzindo assim seu tempo computacional.

Goldberg e Deb em [15] fizeram várias considerações a respeito da complexidade computacional necessária às diferentes formas de seleção. Neste trabalho não será adotada esta análise levando-se em conta que o tempo gasto com a execução do tipo de seleção é quase insignificante comparado ao esforço computacional necessário para implementar uma iteração de outras partes do algoritmo, especialmente na resolução do problema de PL.

5.3.4. Taxa de Crescimento da Melhor Configuração

Um outro critério para comparar diferentes tipos de seleção é a taxa de crescimento da melhor configuração quando, além do operador de seleção, é adicionado aos testes o operador de mutação. Este critério de desempenho foi proposto por Goldberg e Deb [15] tendo em vista de que vários tipos de seleção apresentam



tempos de “take over” muito próximos e também para efetuar testes mais próximos de aplicações reais incorporando o operador de seleção.

Em [13] foram efetuados testes para analisar a taxa de crescimento da melhor configuração para vários tipos de seleção. Nestes testes, escolhe-se uma população $n=100$ indivíduos e cada configuração é mutada adicionando-se uma variável Gaussiana aleatória com desvio padrão de 0.02 (os elementos de cada configuração tendem a unicidade na medida que evolui o processo).

No planejamento de sistemas de transmissão este critério de desempenho pode ser muito valioso devido ao papel importante do operador de mutação.

5.3.5. Desempenho na Presença de Ruído

Outro critério importante apresentado pelos teóricos em algoritmos genéticos é o desempenho dos tipos de seleção na presença de ruído. Ruído significa que, por diferentes motivos, os valores da função objetivo das configurações não são exatas, isto é, as funções objetivo encontradas são aproximadas. Como alguns tipos de seleção usam o valor absoluto da função objetivo para a seleção então, uma distorção (ruído) nos valores da função objetivo pode modificar o desempenho de alguns tipos de seleção.

Um caso típico de ruído ocorre quando são usados métodos aproximados para calcular a função objetivo. Por exemplo, uso de métodos simples, mas rápido para calcular um problema de fluxo de carga radial apresenta a vantagem de rapidez computacional em relação aos métodos mais exatos, porém os valores de perdas do sistema, por exemplo, serão aproximados, isto é, estão sujeitos a ruído e esta estratégia de trabalho pode modificar o desempenho do operador seleção.

Em [13] são feitos testes para avaliar o efeito do ruído nos tipos de seleção. Neste caso, os valores exatos da função objetivo são distorcidos adicionando-se uma variável aleatória Gaussiana.

Para o planejamento de sistemas de transmissão, o efeito do ruído é considerado irrelevante pois a função objetivo é calculada de maneira exata usando um software de PL chamado Minos.



A maioria dos métodos de seleção apresentados neste capítulo foram implementados, exceto os métodos do algoritmo genitor, escalonamento sigma e janelamento. O método de janelamento poderia ter sido implementado e incorporado como uma alternativa da seleção proporcional. Os métodos do algoritmo genitor e escalonamento sigma não foram implementados devido a pouca confiabilidade encontrada na literatura especializada [13,15].

5.4. Conclusões

Neste capítulo apresentou-se o operador seleção, analisando-se as principais propostas de implementação, vantagens e desvantagens de cada método proposto. Os exemplos ilustrativos ajudam a entender o funcionamento de cada tipo de seleção. É realizada uma revisão dos conceitos teóricos do algoritmo genético, sugeridas na literatura especializada, para comparar os diferentes tipos de seleção.



CAPÍTULO 6

O Algoritmo Genético Básico Aplicado ao Problema de Planejamento

Muitos problemas de otimização combinatorial são resolvidos utilizando métodos aproximados devido à grande dificuldade em encontrar a solução ótima, desfocando-se entre esses métodos, o modelo de transportes proposto por Garver. Os métodos combinatoriais, como é o caso dos algoritmos genéticos, foram projetados para lidar com este tipo de problema. Matematicamente, pode-se considerar o algoritmo genético como uma técnica de otimização combinatorial com alta probabilidade de encontrar a solução ótima para problemas grandes e complexos os quais, apresentam muitas soluções ótimas locais. Este é o caso de redes de transmissão de grande porte, que devido às restrições de integralidade (isto é, as linhas só assumem valores inteiros) se transformam em um problema com explosão combinatorial de difícil resolução. Por exemplo, o sistema Norte-Nordeste brasileiro, com 89 nós e 183 linhas chega a ter $(11+1)^{183}$ possíveis soluções supondo um máximo de 11 linhas candidatas por ramo.

No algoritmo genético surge um conjunto de componentes definindo o tipo e a qualidade do algoritmo. Pode-se dividir esses componentes em quatro partes:

- 1- Representação e codificação do problema de planejamento;
- 2- Mecanismo geracional: seleção - recombinação - mutação;

- 3- Geração da população inicial;
- 4- Parâmetros de controle e o critério de parada.

6.1. Representação do Problema

A função objetivo em (2.3) representa o *custo das linhas adicionadas* mais o parâmetro de *penalidade* α correspondente ao corte de carga produzido em uma determinada configuração. Assim, na representação do planejamento de sistemas de transmissão uma configuração ("string") está constituída pelas variáveis de investimento n_{ij} (número de linhas adicionadas em cada caminho candidato) e o efeito das outras variáveis contínuas de operação são consideradas implicitamente em conjunto como corte de carga no sistema. O valor da função objetivo ou equivalente "fitness" determina a qualidade da configuração. Na *formulação tradicional* dos algoritmos genéticos o problema é formulado na forma de maximização. No planejamento de sistema de transmissão a transformação de um problema de minimização em um equivalente de maximização não é muito trivial, podendo comprometer a seletividade do processo, pode-se implementá-lo da seguinte forma:

$$\min v \iff \max [K - v] \quad (6.a)$$

K é uma constante mantida fixa ou variável durante o processo de otimização. A relação (6.a) equivale à transformação típica usada em otimização onde $\min v \iff -[\max -v]$ e a função objetivo transformada do problema de planejamento de sistemas de transmissão assume valores negativos. Alguns métodos de seleção, não trabalham com valores negativos sendo preciso utilizar uma constante K para transformar todos os valores da função objetivo em positivos.

Quando o algoritmo inicia o processo com configurações obtidas por outros métodos, K pode ser fixo e assumir um valor 20% acima da função objetivo de uma configuração típica ou da melhor configuração sem corte de carga encontrada (incumbente). Quando a população inicial é gerada aleatoriamente K assume um valor



variável em cada geração em torno de 5% acima da pior configuração. Uma segunda proposta de transformação equivale a fazer a seguinte alteração:

$$\min v \iff \max [1 / v] \quad (6.b)$$

6.2. Codificação do Problema

No algoritmo genético tradicional as variáveis do problema são codificadas em sistema *binário*, permitindo uma melhor implementação dos operadores de recombinação e mutação. No problema de planejamento de sistemas de transmissão as variáveis que formam uma configuração são inteiras e a codificação mais óbvia consistiria em representá-las em sistema binário. Entretanto, analisando-se o problema específico, pode-se verificar a existência de três formas de codificar uma configuração:

- Codificação em sistema binário,
- Codificação em bits independentes e
- A própria codificação decimal.

A figura 6.1 ajuda a analisar estes 3 tipos de codificação.

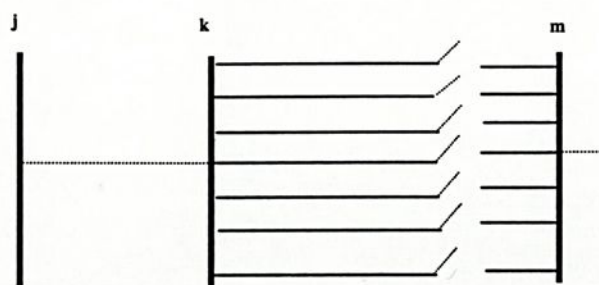


Figura 6.1: O Problema da Codificação

Na figura 6.1, existe a possibilidade de adicionar 7 linhas entre as barras *k* e *m*. Supõe-se a seguir a adição de três linhas na configuração corrente. A figura 6.2, mostra a codificação da configuração para o caminho *k* – *m*. As três formas de codificação apresentam vantagens e desvantagens quando são considerados os aspectos mais importantes do processo.

Um detalhe importante é o *espaço necessário* para o armazenamento das configurações. No exemplo, a codificação tradicional em *sistema binário* dispõe de três casas binárias, a representação por *bits independentes* utiliza sete casas binárias e a representação decimal usa uma casa decimal. Um problema de grande porte como o Norte-Nordeste brasileiro, tipicamente permite adicionar mais de 7 linhas por caminho candidato, sendo um total de 190 caminhos candidatos. Desse modo, entre as três alternativas existe uma diferença enorme em requerimento de memória necessária ao armazenamento das configurações, mostrando a grande vantagem da codificação decimal. Adicionalmente, as duas primeiras formas de codificação requer um esforço computacional maior, pois cada configuração deve ser transformada em seu equivalente decimal para usar a subrotina de **PL** que determina o corte de carga correspondente a cada configuração.

Codificação em bits independentes

...	0	1	0	1	1	0	0	...
-----	---	---	---	---	---	---	---	-----

Codificação em sistema binário

...	0	1	1	...
-----	---	---	---	-----

Codificação em sistema decimal

...	3	...
-----	---	-----

Figura 6.2: Diferentes Formas de Codificação

Outro aspecto importante presente na codificação binária é o surgimento de um problema conhecido como *penhasco de Hamming* ("Hamming cliffs"), significando dizer, que existe uma grande distância entre dois inteiros consecutivos. Por exemplo os números inteiros 3 e 4 são representados em sistema binário respectivamente, como 011 e 100. Um algoritmo genético deve transformar, usando recombinação e mutação, todos os seus dígitos para passar de 3 a 4 circuitos que realmente são configurações vizinhas. Por outro

lado, a codificação usando bits independentes e a codificação decimal não apresentam este problema.

Uma questão a ser discutida relacionada com a codificação escolhida acontece na implementação da recombinação e da mutação. A figura 6.3 apresenta, em codificação binária tradicional, as adições no caminho $k - m$ mostrada na figura 6.1 de duas configurações. Antes da recombinação as configurações indicam a alocação de 3 e 4 linhas respectivamente. Após a recombinação, as novas configurações indicam a alocação de 7 e 0 linhas adicionadas respectivamente; portanto, *no ponto de recombinação* não acontece uma troca de informação adequada. O que realmente acontece no ponto de recombinação é uma mutação violenta (de 3 para 7 linhas e de 4 para 0 linhas). Igualmente, no caso da mutação, ocorreu uma transformação na posição central da segunda configuração mas isto implica passar de 0 a 2 linhas indicando uma mudança brusca. Este problema não acontece com as outras duas formas de codificação.

Um dado final em relação a codificação diz respeito a codificação binária e por bits independentes, onde as operações de recombinação e mutação são implementadas, do ponto de vista computacional, de uma maneira trivial, enquanto que na representação decimal, deve-se redefinir a operação de mutação. Também é possível realizar uma codificação binária de forma implícita, isto é, manter uma codificação decimal, porém realizar as operações de recombinação e mutação simulando a codificação binária.

As observações levam a pensar que uma codificação decimal com um aumento de taxa de mutação seria equivalente e até mais eficiente se comparada a uma codificação binária em termos de qualidade de soluções encontradas e com um menor esforço computacional. Mantendo uma codificação decimal, na recombinação realmente aconteceria uma transferência adequada de informações e a mutação simplesmente consistiria em aumentar ou diminuir em uma unidade o número de linhas para um caminho candidato. Esta codificação ainda facilita o trabalho computacional na hora de testar as configurações através de um PL. Pode-se incorporar mutações maiores e definir mutações simples, duplas, etc.

Pesquisas recentes sugerem a melhor codificação como aquela que mantém a natureza das variáveis, isto é, as variáveis binárias devem ser codificadas em sistema



binário, as variáveis inteiras devem ser codificadas em sistema decimal e as variáveis reais como números reais. Portanto, o vetor que representa uma configuração teria o tamanho do número de variáveis do problema.

6.3. O Mecanismo Geracional: Seleção - Recombinação - Mutação

A partir da configuração corrente é determinada uma nova população através do processo geracional implicando realizar sucessivamente as operações de seleção, recombinação e mutação.

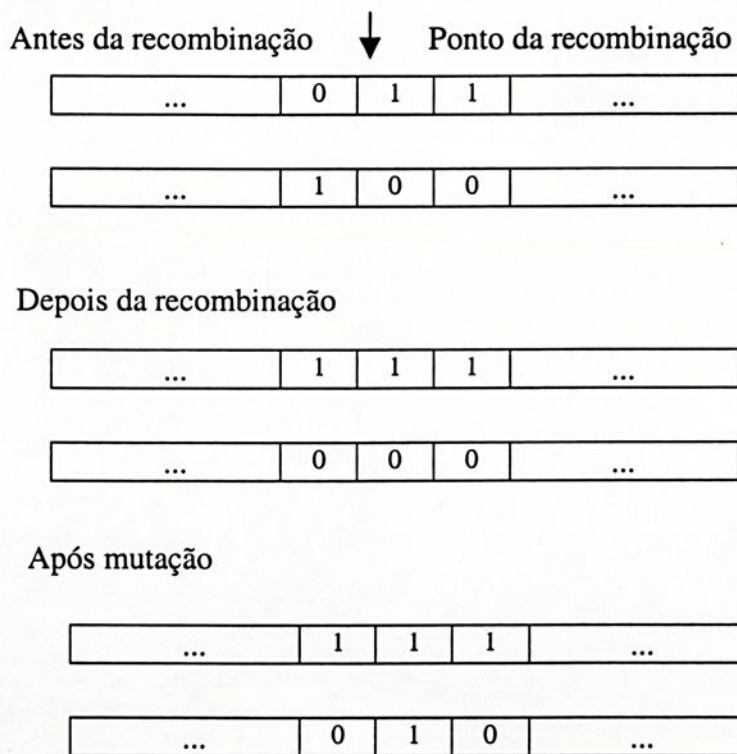


Figura 6.3: Recombinação e Mutação

Nos algoritmos genéticos o mecanismo de seleção é a primeira etapa no processo de obtenção da nova população. No processo de seleção determina-se o número de descendentes (“offspring”) a que tem direito cada configuração (“string”). Nas etapas seguintes são implementadas recombinação e mutação para terminar de gerar os elementos (configurações) da nova população.

A figura 6.4 ilustra as operações de recombinação e mutação para duas configurações de uma rede simples usando a codificação binária. Observa-se em primeira instância o procedimento de recombinação entre configurações, para isto é selecionado o ponto de recombinação correspondente ao quarto bit, o bloco identificado entre a posição seguinte e o último bit será trocado entre as duas configurações e a troca dos blocos originais só inclui a linha 2-3. No passo seguinte realiza-se o procedimento de mutação, o qual é determinado pela taxa de mutação. Na primeira configuração foi selecionada a terceira posição correspondente a linha 1-3, adicionando-se duas novas linhas; na segunda configuração foi selecionada a segunda posição, correspondente a linha 1-2 e retirando-se uma nova linha.

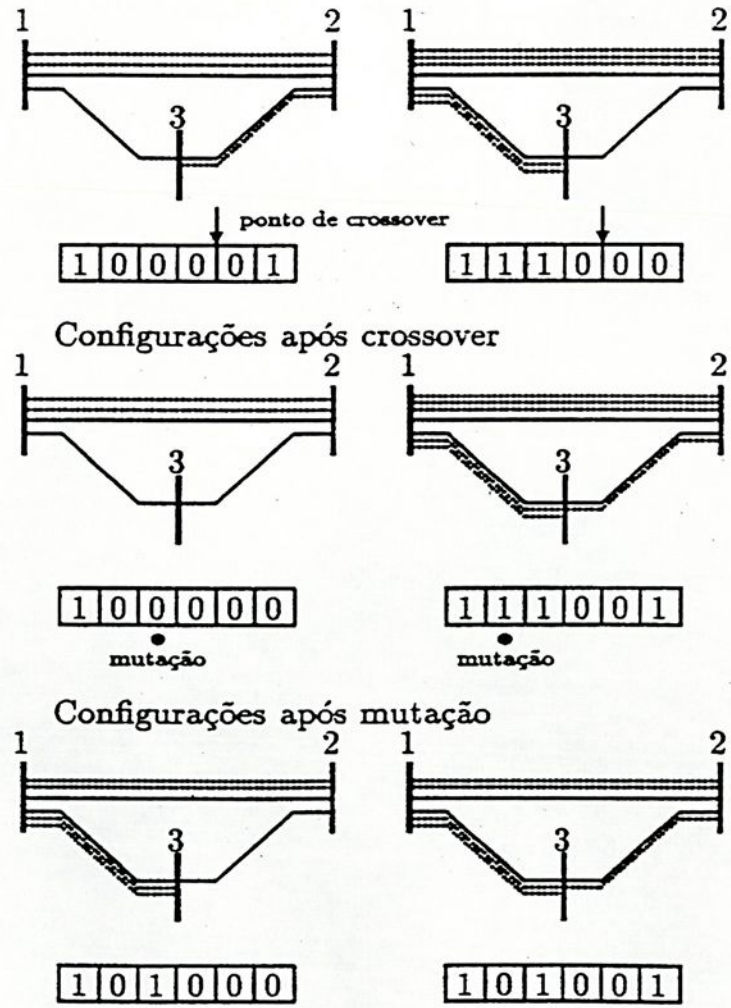


Figura 6.4: Recombinação e Mutação no Sistema Binário

6.4 Geração da População Inicial

No planejamento de redes de transmissão, gera-se a população inicial de duas maneiras. A primeira corresponde a uma forma aleatória na qual se controla o número de posições diferentes de zero. Tipicamente o número de posições diferentes de zero na configuração final está na faixa de [15 - 20] % do total de posições (para a geração aleatória ocupa-se entre [30 - 40] % das posições). Este tipo de inicialização será empregado em sistemas pequenos (sistema de Garver) e de médio porte (sistema Sul brasileiro). Chega-se, com alguma frequência, à solução ótima, mas com esforço computacional elevado.

Para sistemas de grande porte (Norte-Nordeste brasileiro), a escolha aleatória da população mostra-se pouco efetiva devido ao grande número de regiões no espaço de soluções que contém esta rede. Iniciando o processo com configurações geradas de maneira aleatória, pode-se ficar preso em alguma região de pobre qualidade dificultando sua resolução.

É conveniente gerar a população inicial usando algoritmos heurísticos construtivos. Três algoritmos heurísticos construtivos podem ser empregados no estudo do planejamento das redes de transmissão:

- mínimo esforço [3],
- mínimo corte de carga [4]
- Garver, baseado no modelo de transportes da rede de transmissão [2].

Os algoritmos mencionados foram apresentados no capítulo 3. Em [1] concluiu-se que o algoritmo de Garver é o mais adequado para gerar configurações. Os principais motivos são:

- A metodologia de Garver apresenta configurações de melhor qualidade especialmente na alocação adequada de linhas de grande capacidade de transmissão nas primeiras fases do processo de adição.
- A metodologia de Garver leva em conta apenas a primeira lei de Kirchhoff, as configurações obtidas não adicionam linhas pequenas permitindo que esta parte da alocação seja adequadamente executada pelo algoritmo genético.



6.5 Parâmetros de Controle e Critérios de Parada

A escolha dos parâmetros de controle de um algoritmo genético é um tópico polêmico entre os especialistas. Estes parâmetros não podem ser selecionados de maneira independente, pois estão estritamente relacionados.

O bom desempenho de um algoritmo depende em grande parte de uma boa escolha dos parâmetros de controle, taxa de recombinação, taxa de mutação e o tamanho da população; além dos anteriores, é importante uma adequada seleção do tipo de recombinação (simples, duplo, etc.) e a maneira como será efetuada a mutação. Com relação ao tamanho da população, esta dependerá do sistema analisado. A literatura recomenda uma população entre [30 - 200]. Na prática, para algoritmos genéticos de aplicação específica os parâmetros de controle devem ser calibrados através de testes exaustivos usando sistemas com soluções ótimas conhecidas.

6.6. Implementações Especializadas

Apresentam-se propostas adicionais as quais devem ser incorporadas no algoritmo genético para melhorar seu desempenho. As propostas são de caráter especializado, isto é, aproveitam as características específicas do problema de planejamento de sistemas de transmissão.

6.6.1. Implementação de uma Taxa de Mutação Variável

A mutação permite encontrar novas alternativas ou regiões promissoras ao alterar o estado em uma posição da configuração. Este aspecto é importante nas fases finais do processo quando a maioria das configurações apresentam funções objetivo muito próximas, onde uma taxa de mutação relativamente elevada pode produzir modificações significativas dando mais dinamismo ao processo.



Os algoritmos genéticos básicos utilizam taxas de mutação pequenas atribuindo-lhe um papel secundário. Entretanto, novas pesquisas realizadas em problemas reais assinalam a importância de se usar uma taxa de mutação alta e variável. Por exemplo, para uma dada configuração pode ser gerado um número aleatório entre 0 e 5, que define o número de pontos para mutação. Logo, para cada ponto de mutação escolhido aleatoriamente, implementa-se uma mutação simples ou dupla na representação decimal, isto é, procede-se à adição ou retira-se uma ou duas linhas na configuração do sistema.

Todas as implementações em relação a mutação, tem dois objetivos:

1. Produzir mudanças bruscas na configuração afim de romper a homogeneidade das configurações da população e
2. Recuperar a perda do material genético fazendo pequenas alterações nas configurações.

6.6.2. Busca Eficiente de Populações Iniciais

O algoritmo de Garver [2] foi modificado com a finalidade de se gerar configurações radicalmente diferentes introduzindo diversidade no material genético. O algoritmo modificado proposto em [1] e usado para gerar a população inicial, é resumido como segue:

- 1-Determinar um plano de expansão ótimo usando variáveis de decisão contínuas,
- 2-Adicionar a parte inteira de todas as linhas/transformadores do passo 1 e formar a nova configuração base,
- 3-Usando a configuração anterior, preparar uma lista de NL linhas/transformadores com investimento fracionário e que nesta configuração transportam os maiores fluxos de potência (com cada elemento da lista será obtido uma nova configuração),



4-Para cada elemento da lista fazer:

- (a) Adicionar um circuito (linha/transformador) que identifique a lista para configuração base e formar a configuração,
- (b) Usando um algoritmo de PL testar a factibilidade da configuração corrente. Se não existe a necessidade da adição de novas linhas/transformadores, foi encontrada uma configuração a qual deve ser incorporada à população inicial e voltar em (a) para analisar outra configuração da lista. Caso contrário, ir para (c),
- (c) Obter uma nova configuração adicionando à configuração corrente a linha/transformadores, que transporta maior fluxo de potência e voltar a (b).

6.6.3. Parâmetro de Penalidade α

O parâmetro α penaliza a função objetivo com corte de carga tornando as configurações pouco atrativas. No processo de seleção α deve assumir um valor relativamente elevado para garantir o término do processo com uma configuração sem corte de carga. Nas fases iniciais do processo, um α grande produz uma eliminação prematura de configurações com corte de carga e que eventualmente poderiam levar o processo à regiões promissoras. Logo, é desejável nas fases iniciais do processo que as configurações com corte de carga razoáveis tenham chance de sobrevivência por algumas gerações. Isto pode ser obtido com um α variável, iniciando com um α menor e ir incrementando seu valor após cada geração.



6.6.4. Identificação de "Building Blocks" Especiais

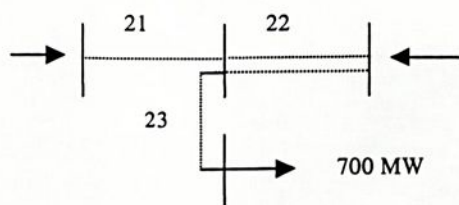
A grande capacidade do algoritmo genético de encontrar configurações ótimas ou quase-ótimas é explicada pela geração sistemática dos chamados blocos construtivos ("building blocks").

A hipótese de "building blocks" supõe que, os operadores de recombinação e mutação, fazem a geração, promoção e justaposição de "blocos construtivos" para formar configurações ótimas. No problema de planejamento de sistemas de transmissão existem pequenos grupos de circuitos que, em conjunto, resolvem adequadamente problemas de geração ou demanda em barras específicas. Aqueles circuitos correspondentes aos novos caminhos são especialmente interessantes por dois motivos:

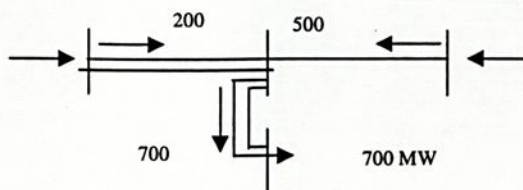
1. A adição nesses novos caminhos deve ser feita de maneira a fechar o novo caminho implicando a adição de dois ou mais circuitos simultaneamente.
2. A combinação de adições propostas num grupo de caminhos identificados definem a qualidade do investimento e não as adições consideradas isoladamente.

A figura 6.5, mostra uma parcela de um sistema elétrico, ilustrando o caso especial de barras isoladas.

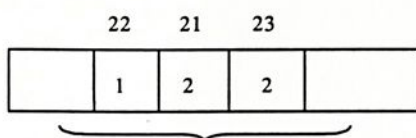




(a) Adição de uma Linha no Sistema



(b) Adições de Linhas no Sistema



(c) Adição de Linhas em Codificação Decimal.

Figura 6.5: "Building Blocks"

Na figura 6.5(a) aparece uma carga de 700 MW, dois pontos prováveis de fornecimento de potência e três caminhos candidatos. Na figura 6.5(b) surge a combinação de adições resolvendo de maneira adequada o fornecimento de carga de 700 MW, e em 6.5(c) são apresentadas estas adições em codificação decimal. Nestas condições são importante as seguintes observações:

- 1- Para chegar da figura 6.5(a) para 6.5(b) foi gerado uma espécie de "building blocks", isto é, adições adequadas em vários caminhos que em conjunto resolvem o problema da demanda de 700 MW;
- 2- A combinação de adições nos três caminhos constitui uma informação de excelente qualidade e não a informação de adições isoladas;
- 3- Tanto recombinação como mutação podem destruir os "building blocks" especiais da figura 6.5(b);

4- Na figura 6.5(a) na implementação de adições, não faz sentido adicionar linhas no caminho 23 sem fazer adições simultaneamente no caminho 21 e/ou 22, pois uma adição isolada em 23 é completamente inútil;

5- Na codificação de uma configuração normalmente os caminhos 21, 22 e 23 não ocupam casas consecutivas e no sistema elétrico existem muitos subsistemas como os mostrados na figura 6.5(c). Dessa forma, com os caminhos 21, 22 e 23 ocupando casas distantes na configuração, podem ser facilmente destruídos pelo operador recombinação.

Levando em conta as observações anteriores, no presente trabalho foi realizada uma implementação computacional identificando os "blocos construtivos" e utilizando-os na alocação de novos caminhos. Assim, um "building block" está constituído por linhas adicionadas em novos caminhos, sendo consecutivas ou adjacentes, surgindo geralmente em torno de grandes barras de carga ou geração.

A operação de recombinação não destrói estes "building blocks", pois todos os elementos entram numa nova configuração. Esta estratégia também elimina a possibilidade de que a recombinação forneça configurações com adições inadequadas (linhas que não transmitem potência porque não fecham um caminho). É interessante observar que estes blocos apresentam comprimento de definição ideal, não sendo destruídos por recombinação, mesmo que seus elementos não sejam codificados em casas vizinhas ou consecutivas.

A operação de mutação pode aumentar ou diminuir o número de "blocos construtivos" presentes no sistema. Por exemplo, na figura 6.5(b), uma mutação em 21 e duas mutações sucessivas em 23 levaria à eliminação desse "building block" com somente 2 elementos. Esta implementação dá um papel especial a mutação, sendo necessário uma taxa de mutação maior.

Na implementação da mutação deve-se implementar uma estratégia para evitar configurações indesejáveis, contornando o problema da adição inadequada de linhas em novos caminhos, e aproveitando características específicas do problema de planejamento formando "blocos construtivos" com linhas vizinhas. Embora seja possível generalizar essa idéia, considerando também os caminhos correspondentes a reforços em linhas já



existentes, a implementação computacional cresce em complexidade e se torna difícil identificar os blocos independentes. Assim, estes outros caminhos funcionam como "building blocks" de um único elemento formando "blocos construtivos" tradicionais com operações de recombinação e mutação convencionais.

Na genética natural também existe um fenômeno que apresenta características parecidas à estratégia de identificação de caminhos podendo ser agrupados em blocos construtivos especiais da forma apresentada anteriormente. Na genética a característica de um indivíduo está definida pelo fenotipo que por sua vez, é definido pelo genotipo. Normalmente, cada gene de um cromossomo determina um tipo de fenotipo (por exemplo cor dos olhos). Mas existem casos em que um conjunto de genes determinam um único fenotipo. A este fenômeno dá-se o nome de poligenia, onde um conjunto de genes (vários genes) determinam, interagindo em conjunto um único fenotipo. Portanto, determinados fenotipos são estabelecidos pela ação conjunta de vários genes. No problema de planejamento de sistemas de transmissão pode-se dizer que é possível identificar vários circuitos, interagindo em conjunto resolvendo o problema de geração ou demanda numa barra. Assim, a estratégia chamada blocos construtivos especiais apresentados anteriormente, também poderia ser chamada de "identificação de poligenia" no sistema elétrico.

6.6.5. Algoritmo Genético Proposto

Levando em conta os critérios anteriores, implementou-se o seguinte algoritmo genético:

Passo 1: Definir T_p (tamanho da população) e determinar a população inicial de forma aleatória, usando configurações conhecidas ou o algoritmo de Garver modificado. Definir $K_{máx}$ (critério de parada) e K_{stop} (número máximo de gerações permitidas).



Passo2: Calcular a função objetivo ("fitness") ou equivalente, usando o algoritmo de PL. Atualizar se possível a incumbente. K1 conta o número de gerações em que a incumbente permanece sem variar, e K2 conta o número de gerações.

Passo 3: Se $K1 > K_{\text{máx}}$ pare, pois o processo convergiu. Se $K2 > K_{\text{stop}}$ pare, pois foi excedido o número máximo de gerações permitidas. Caso contrário, vá para o passo 4.

Passo 4: Implementar o operador seleção e determinar o número de descendentes correspondente a cada configuração.

Passo 5: Usando um dos tipos de recombinação propostos, implementar a recombinação de maneira sistemática entre cada par de configurações que devem participar na formação da nova geração, até gerar um número de descendentes igual ao tamanho da população. A taxa de recombinação controla estatisticamente o número de configurações que serão recombinadas.

Passo 6: Implementar a mutação de maneira sistemática. Cada configuração candidata realiza até n_m mutações (simples ou duplas), sendo proporcional à taxa de mutação ρ_m . Este processo também é controlado estatisticamente pela taxa de mutação ρ_m . Voltar ao passo 2.

6.7. Conclusões

Apresentou-se a metodologia do algoritmo genético para o problema de planejamento da expansão de sistemas de transmissão. O algoritmo proposto é uma modificação do apresentado em [1]. Porém, foram realizadas modificações significativas especialmente na implementação de várias técnicas de seleção, ajustes dos parâmetros de



controle, melhoria dos “building blocks”, assim como da implementação computacional, entre outros.



CAPÍTULO 7

Resultados Obtidos

Os testes foram executados com o objetivo de comparar o desempenho dos diferentes métodos de seleção implementados no algoritmo, realizou-se os testes com sistemas de 06 Barras de Garver, Sistema Sul Brasileiro e o Sistema Norte-Nordeste Brasileiro. Os métodos de seleção implementados foram:

- Método 1: Seleção Proporcional
- Método 2: Seleção Baseada em Escalonamento Linear
- Método3: Seleção de Ordenamento (“ranking”)
- Método 4: Seleção de Torneio

7.1. Testes Realizados Com o Sistema de 06 Barras de Garver Sem Redespacho

Empregou-se os seguintes parâmetros nos testes realizados com o Sistema de 06 Barras de Garver:

- Custo de racionamento (nafa): $\alpha = 10$
- Tamanho da população: 20
- Valor máximo da incumbente (kincumb): 300000

- Número máximo de gerações (njera): 40
- Taxa de recombinação (ycross): 0.90
- Taxa de variação das mutações (bconst): 0.97
- Número máximo de mutações por configuração (nmutar): 2
- Número máximo de descendentes gerados por configuração (ndes em %): 6
- Número máximo de configurações que participam da seleção (nmp): 26
- Número de configurações que entram no jogo (ncj): 6

Os resultados apresentados correspondem a melhor solução encontrada (incumbente) e o número de PL's necessários (PL 1) para chegar a esse valor. Foram executados em cada teste 1200 PL's. Nas tabelas 7.1 a 7.4 tem-se os respectivos resultados obtidos com as seleções: proporcional, escalonamento, ordenamento e torneio.

No.	Incumbente	PL 1
1	248	990
2	200	690
3	200	810
4	272	450
5	200	630
6	200	570
7	200	720
8	200	870
9	200	840
10	301	570
11	200	720
12	200	960
13	200	330
14	231	210
15	200	540
16	200	1080
17	272	240
18	282	390
19	200	780
20	231	990

Tabela 7.1: Seleção Proporcional

No.	Incumbente	PL 1
1	316	900
2	200	1020
3	200	420
4	200	450
5	238	930
6	341	330
7	200	750
8	200	1140
9	248	1020
10	200	1020
11	200	1230
12	200	450
13	296	1200
14	200	1230
15	200	780
16	200	570
17	200	840
18	200	300
19	200	540
20	231	900

Tabela 7.2: Seleção de Escalonamento

No.	Incumbente	PL 1
1	282	390
2	200	540
3	200	690
4	200	1110
5	200	390
6	200	660
7	329	1080
8	200	570
9	200	660
10	231	1200
11	200	1230
12	200	510
13	200	480
14	231	1140
15	200	420
16	200	360
17	200	1230
18	231	390
19	200	270
20	200	420

Tabela 7.3: Seleção de Ordenamento

No.	Incumbente	PL 1
1	200	480
2	282	360
3	200	570
4	328	240
5	200	360
6	200	750
7	200	960
8	200	780
9	200	420
10	200	750
11	200	480
12	282	270
13	315	270
14	272	300
15	200	900
16	282	270
17	200	270
18	334	270
19	200	270
20	200	270

Tabela 7.4: Seleção de Torneio

Os esquemas de seleção de ordenamento e torneio foram submetidos a novos testes variando alguns parâmetros citados anteriormente. Na seleção baseada em ordenamento mudou-se o número máximo de configurações (nmp) e atribuiu-se 2 ao número máximo de descendentes por configuração (nmd). Na seleção de torneio variou-se o número de configurações participantes do torneio (ncj). Obteve-se os seguintes resultados:

No.	Incumbente	PL 1
1	231	840
2	200	690
3	200	600
4	200	600
5	231	1130
6	200	990
7	200	570
8	200	840
9	200	570
10	200	960
11	200	900
12	200	1230
13	200	630
14	200	360
15	200	690
16	200	480
17	200	420
18	200	390
19	200	570
20	200	690

Tabela 7.5 (a)

No.	Incumbente	PL 1
1	200	750
2	200	270
3	200	390
4	200	450
5	200	660
6	200	600
7	200	510
8	200	750
9	200	480
10	200	390
11	282	630
12	200	810
13	200	390
14	200	900
15	200	960
16	200	420
17	200	480
18	230	1230
19	200	1230
20	200	670

Tabela 7.5 (b)

No.	Incumbente	PL 1
1	200	750
2	220	1230
3	200	510
4	272	690
5	200	420
6	231	690
7	200	450
8	231	660
9	200	690
10	238	450
11	341	270
12	200	540
13	200	840
14	200	480
15	200	780
16	200	480
17	200	1020
18	200	450
19	200	480
20	200	720

Tabela 7.5 (c)

As tabelas 7.5: (a), (b) e (c) apresentam os resultados obtidos com a seleção de ordenamento.

No.	Incumbente	PL 1
1	200	540
2	200	270
3	374	450
4	200	240
5	260	960
6	282	420
7	282	450
8	240	200
9	341	450
10	200	390

Tabela 7.6 (a)

No.	Incumbente	PL 1
1	200	870
2	200	270
3	200	240
4	200	450
5	200	450
6	260	810
7	376	210
8	364	210
9	200	360
10	200	780

Tabela 7.6 (b)

No.	Incumbente	PL 1
1	200	450
2	200	420
3	200	510
4	200	630
5	328	470
6	343	900
7	200	720
8	200	600
9	341	690
10	316	270

Tabela 7.6 (c)

As tabelas 7.6: (a), (b) e (c) apresentam os resultados obtidos com a seleção de torneio.



7.2. Testes Realizados Com o Sistema Sul Brasileiro Sem Redespacho

Empregou-se os seguintes parâmetros nos testes realizados com o Sistema Sul Brasileiro:

- Custo de racionamento (nafa): $\alpha = 1000$
- Tamanho da população: 60
- Valor máximo da incumbente (kincumb): 300.000.000
- Número máximo de gerações (njera): 100
- Taxa de cruzamento (ycross): 0.90
- Taxa de variação das mutações (bconst): 0.99
- Número máximo de mutações por configuração (nmutar): 4
- Número máximo de configurações que participam da seleção (nmp): 26
- Número máximo de descendentes por configuração (nmd): 2
- Número de configurações que entram no jogo (ncj): 6

Os métodos de seleção empregados nos testes encontraram a solução ótima (US\$ 154.420.000), onde a população inicial foi gerada de maneira aleatória e com o algoritmo de Garver. Obteve-se um melhor desempenho do algoritmo inicializado por Garver em que foram reduzidos o número de PL's para encontrar a solução ótima. Foram executados em cada teste 6000 PL's.

Nas tabelas 7.7, 7.8, 7.9 e 7.10 tem-se os respectivos resultados obtidos com as seleções: proporcional, escalonamento, ordenamento e torneio, com a população gerada aleatoriamente.



No.	Incumbente	PL 1
1	173.697	4020
2	154.420	4440
3	154.420	3900
4	154.420	6000
5	162.734	5760
6	175.946	4440
7	154.420	5040
8	195.895	5760
9	171.780	5520
10	164.752	5220
11	166.113	3901
12	215.033	4680
13	154.420	6000
14	157.198	4980
15	164.752	3480
16	154.420	4980
17	173.697	5940
18	176.886	5640
19	154.420	5940
20	168.095	5700

Tabela 7.7: Seleção Proporcional

No.	Incumbente	PL 1
1	154.420	5640
2	164.514	4500
3	154.420	4860
4	164.752	4380
5	175.352	5340
6	154.420	5520
7	164.113	4300
8	183.667	5160
9	154.420	4320
10	174.300	4440
11	154.420	4380
12	157.863	4380
13	199.559	4500
14	181.098	6000
15	170.005	5720
16	178.728	3720
17	154.420	5240
18	162.734	4720
19	178.465	4740
20	154.420	3300

Tabela 7.8: Seleção por Escalonamento Linear

No.	Incumbente	PL 1
1	164.880	4560
2	164.880	2280
3	203.787	4080
4	172.273	1500
5	164.880	5460
6	154.420	2880
7	154.420	3660
8	162.188	6420
9	162.730	4740
10	174.174	2340
11	154.420	5400
12	154.420	4320
13	154.420	2220
14	172.482	2880
15	164.880	3240
16	154.420	6480
17	169.822	6420
18	162.734	8700
19	154.420	2760
20	198.302	3360

Tabela 7.9: Seleção de Ordenamento

No.	Incumbente	PL 1
1	164.752	5700
2	164.752	3960
3	173.697	3480
4	154.420	3780
5	164.752	5040
6	154.420	4380
7	173.697	4560
8	167.333	6060
9	164.880	3060
10	172.273	4620
11	189.182	5760
12	164.752	4320
13	172.164	3240
14	152.420	5880
15	154.420	4200
16	164.752	4500
17	154.420	3300
18	154.420	4380
19	165.512	5100
20	164.752	4800

Tabela 7.10: Seleção de Torneio

Nas tabelas 7.11, 7.12, 7.13 e 7.14 tem-se os respectivos resultados obtidos com as seleções: proporcional, escalonamento, ordenamento e torneio, com a população gerada por Garver.

No.	Incumbente	PL 1
1	154.420	2350
2	154.420	2100
3	154.420	2200
4	191.546	1700
5	154.420	1000
6	154.420	3300
7	154.420	1650
8	154.420	2550
9	154.420	2150
10	154.420	2300
11	154.420	1250
12	178.603	1050
13	154.420	1050
14	181.995	3450
15	154.420	2900
16	154.420	1000
17	183.157	1400
18	154.420	1250
19	154.420	1600
20	154.420	2450

Tabela 7.11: Seleção Proporcional

No.	Incumbente	PL 1
1	154.420	1000
2	190.173	1000
3	154.420	1200
4	154.420	1000
5	154.420	1000
6	157.198	1240
7	154.420	1400
8	154.420	1000
9	154.420	1000
10	154.420	1680
11	154.420	1120
12	154.420	1680
13	154.420	1240
14	154.420	1680
15	154.420	1140
16	183.511	1600
17	183.157	1430
18	154.420	1760
19	178.603	1260
20	154.420	1000

Tabela 7.12: Seleção por Escalonamento Linear

No.	Incumbente	PL 1
1	154.420	1450
2	190.173	1850
3	154.420	3350
4	154.420	3000
5	154.420	1550
6	154.420	2300
7	154.420	1750
8	154.420	1000
9	154.420	1000
10	154.420	1000
11	154.420	2150
12	154.420	1650
13	154.420	2150
14	154.420	1800
15	154.420	1400
16	154.420	1200
17	191.546	1350
18	154.420	1900
19	181.995	1150
20	154.420	1700

Tabela 7.13: Seleção de Ordenamento

No.	Incumbente	PL 1
1	181.995	1000
2	154.420	1000
3	154.420	1000
4	191.546	1000
5	183.911	2000
6	181.995	3450
7	154.420	1000
8	154.420	1000
9	178.603	1500
10	154.420	1000
11	154.420	1000
12	154.420	1000
13	154.420	1000
14	154.420	1000
15	154.420	1850
16	154.420	1600
17	154.420	1300
18	154.420	1000
19	178.603	1000
20	191.546	1000

Tabela 7.14: Seleção de Torneio

7.3. Testes Realizados com o Sistema Norte-Nordeste Brasileiro (Plano-2008)

A população inicial foi gerada pelo algoritmo de Garver, foram utilizados os seguintes parâmetros:

- Custo de racionamento (nafa): $\alpha = 1000$
- Tamanho da população: 180
- Valor máximo da incumbente (kincumb): 4.500.000.000
- Número máximo de gerações (njera): 700
- Número de configurações geradas por Garver (npara): 180
- Taxa de recombinação (ycross): 0.90
- Número máximo de linhas (nmax): 12
- Taxa de variação das mutações (bconst): 0.996

- Número máximo de mutações por configuração (nmutar):24
- Limite de corte de carga na mutação: 150
- Número máximo de configurações que participam da seleção (nmp): 100
- Número máximo de descendentes por configuração (nmd):3
- Número de configurações que entram no jogo (ncj): 8

Geração	Incumbente	PL 1
1	3.635.946	446
10	3.015.446	4296
20	2.871.992	8576
30	2.817.091	12876
40	2.783.248	17145
50	2.769.303	21453
60	2.717.153	25761
70	2.704.332	30112
80	2.703.897	34456
90	2.698.547	38778
100	2.695.982	43101
110	2.691.821	47365
120	2.687.606	51726
130	2.685.136	56115
200	2.675.968	86426
203	2.672.503	87726
205	2.653.772	88588
210	2.642.177	90320
212	2.641.761	91638
220	2.639.716	95073
225	2.636.251	97281
230	2.628.036	99471
255	2.626.257	109863
263	2.622.692	113743
298	2.621.692	128846
339	2.619.132	146583
383	2.614.272	165595
460	2.613.832	198966

Tabela 7.15: Resultados obtidos utilizando seleção proporcional para o sistema Norte-Nordeste Brasileiro.



Na tabela 7.16 apresenta as melhores configurações (incumbente) descobertas em testes preliminares utilizando os diferentes tipos de seleção apresentados neste trabalho para o sistema Norte-Nordeste Brasileiro, onde NPL está em milhares de PL's e o investimento V_{best} está em milhares de dólares.

Tipo de seleção	V_{best}	NPL
Seleção Proporcional	2594517	302
Seleção de Escalonamento	2595866	289
Seleção Baseada em Ranking	2603000	318
Seleção de Torneio	2595957	277

Tabela 7.16: Melhores Resultados Obtidos Utilizando os Diferentes Tipos de Seleção para o Sistema Norte-Nordeste Brasileiro – Plano 2008.

Foram realizados novos testes, relaxando o critério de parada do algoritmo, para avaliar a possibilidade de encontrar configurações de melhor qualidade. A tabela 7.17 apresenta as melhores configurações encontradas com cada método de seleção:

Tipo de seleção	V_{best}	NPL
Seleção Proporcional	2594216	428
Seleção de Escalonamento	2582843	344
Seleção Baseada em Ranking	2601773	551
Seleção de Torneio	2595957	268

Tabela 7.17: Melhores Resultados Obtidos Relaxando o Critério de Parada do Algoritmo Utilizando os Diferentes Tipos de Seleção para o Sistema Norte-Nordeste Brasileiro – Plano 2008.

Encontrou-se as seguintes topologias:

- Utilizando o método de seleção proporcional obteve-se a seguinte configuração com investimento de $V = 2594216$:



$n_{1-2} = 1$, $n_{2-4} = 1$, $n_{4-5} = 5$, $n_{4-81} = 3$, $n_{5-56} = 1$, $n_{5-58} = 4$, $n_{13-15} = 4$, $n_{14-17} = 1$,
 $n_{14-15} = 3$, $n_{15-46} = 1$, $n_{16-44} = 6$, $n_{18-50} = 11$, $n_{15-46} = 1$, $n_{16-44} = 6$, $n_{18-50} = 11$, $n_{18-74} = 5$,
 $n_{21-57} = 2$, $n_{22-58} = 2$, $n_{24-43} = 1$, $n_{25-55} = 4$, $n_{30-31} = 1$, $n_{30-63} = 2$, $n_{35-46} = 1$, $n_{35-51} = 2$,
 $n_{36-46} = 2$, $n_{39-86} = 1$, $n_{40-45} = 1$, $n_{40-46} = 2$, $n_{41-64} = 2$, $n_{42-44} = 1$, $n_{43-55} = 2$, $n_{43-58} = 3$,
 $n_{48-49} = 1$, $n_{49-50} = 4$, $n_{52-59} = 1$, $n_{53-86} = 1$, $n_{54-58} = 1$, $n_{54-63} = 1$, $n_{56-57} = 1$, $n_{61-85} = 3$,
 $n_{61-86} = 1$, $n_{63-64} = 1$, $n_{67-68} = 1$, $n_{67-69} = 1$, $n_{67-71} = 3$, $n_{71-72} = 1$, $n_{72-73} = 1$, $n_{73-74} = 2$,
 $n_{73-75} = 1$, $n_{75-81} = 1$

- Utilizando o método de seleção de escalonamento obteve-se a seguinte configuração com investimento de $V = 2582843$:

$n_{1-2} = 1$, $n_{2-4} = 1$, $n_{4-5} = 5$, $n_{4-81} = 3$, $n_{5-56} = 1$, $n_{5-58} = 4$, $n_{6-75} = 1$, $n_{13-15} = 4$,
 $n_{14-59} = 1$, $n_{15-16} = 3$, $n_{15-46} = 1$, $n_{16-44} = 6$, $n_{18-50} = 11$, $n_{18-74} = 5$, $n_{21-57} = 2$, $n_{22-58} = 2$,
 $n_{24-43} = 1$, $n_{25-55} = 4$, $n_{30-31} = 1$, $n_{30-63} = 2$, $n_{35-46} = 1$, $n_{35-51} = 2$, $n_{36-46} = 2$, $n_{39-86} = 2$,
 $n_{40-45} = 1$, $n_{40-46} = 2$, $n_{41-64} = 2$, $n_{42-44} = 2$, $n_{43-55} = 2$, $n_{43-58} = 3$, $n_{48-49} = 1$, $n_{49-50} = 4$,
 $n_{52-59} = 1$, $n_{53-86} = 1$, $n_{54-58} = 1$, $n_{54-63} = 1$, $n_{56-57} = 1$, $n_{61-85} = 2$, $n_{61-86} = 1$, $n_{63-64} = 1$,
 $n_{67-68} = 1$, $n_{67-69} = 1$, $n_{67-71} = 3$, $n_{69-87} = 1$, $n_{71-72} = 1$, $n_{72-73} = 1$, $n_{73-74} = 2$, $n_{73-75} = 1$,
 $n_{75-81} = 1$

- Utilizando o método de seleção baseada em "ranking" obteve-se a seguinte configuração com investimento de $V = 2601773$:

$n_{1-2} = 1$, $n_{2-4} = 1$, $n_{4-5} = 5$, $n_{4-81} = 3$, $n_{5-56} = 1$, $n_{5-58} = 4$, $n_{6-75} = 1$, $n_{12-15} = 2$,
 $n_{13-15} = 3$, $n_{14-59} = 1$, $n_{15-16} = 3$, $n_{18-50} = 11$, $n_{15-46} = 3$, $n_{16-44} = 6$, $n_{18-50} = 11$, $n_{18-74} = 5$,
 $n_{21-57} = 2$, $n_{21-57} = 2$, $n_{22-58} = 2$, $n_{24-43} = 1$, $n_{25-55} = 4$, $n_{30-31} = 1$, $n_{30-63} = 2$, $n_{35-51} = 1$,
 $n_{36-46} = 2$, $n_{39-86} = 1$, $n_{40-45} = 1$, $n_{40-46} = 1$, $n_{41-64} = 2$, $n_{42-44} = 2$, $n_{43-55} = 2$, $n_{43-58} = 3$,
 $n_{48-49} = 1$, $n_{49-50} = 4$, $n_{52-59} = 1$, $n_{53-86} = 1$, $n_{54-58} = 1$, $n_{54-63} = 1$, $n_{56-57} = 1$, $n_{61-85} = 2$,
 $n_{61-86} = 1$, $n_{63-64} = 1$, $n_{67-68} = 1$, $n_{67-69} = 1$, $n_{67-71} = 3$, $n_{69-87} = 1$, $n_{71-72} = 1$, $n_{72-73} = 1$,
 $n_{73-74} = 2$, $n_{73-75} = 1$, $n_{75-81} = 1$

- Utilizando o método de seleção de torneio obteve-se a seguinte configuração com investimento de $V = 2595957$:

$n_{1-2} = 1$, $n_{2-4} = 1$, $n_{4-5} = 5$, $n_{4-81} = 3$, $n_{5-56} = 1$, $n_{5-58} = 4$, $n_{13-15} = 4$, $n_{14-59} = 1$,
 $n_{15-16} = 3$, $n_{15-44} = 6$, $n_{18-50} = 11$, $n_{18-74} = 5$, $n_{21-57} = 2$, $n_{22-58} = 2$, $n_{24-43} = 1$, $n_{25-55} = 4$,
 $n_{30-31} = 1$, $n_{30-63} = 2$, $n_{35-46} = 1$, $n_{35-51} = 1$, $n_{36-46} = 2$, $n_{39-86} = 1$, $n_{40-45} = 1$, $n_{40-46} = 2$,
 $n_{41-64} = 2$, $n_{42-44} = 1$, $n_{43-55} = 2$, $n_{43-58} = 3$, $n_{49-50} = 5$, $n_{52-59} = 2$, $n_{53-86} = 1$, $n_{54-58} = 1$,
 $n_{54-63} = 1$, $n_{56-57} = 1$, $n_{61-85} = 3$, $n_{61-86} = 1$, $n_{63-64} = 1$, $n_{67-68} = 1$, $n_{67-69} = 1$, $n_{67-71} = 3$,
 $n_{71-72} = 1$, $n_{72-73} = 1$, $n_{73-74} = 2$, $n_{73-75} = 1$, $n_{75-81} = 1$



7.4. Comentários dos Resultados

A população inicial do sistema de 06 barras foi gerada aleatoriamente pois o algoritmo construtivo de Garver encontra a solução ótima, tornando desnecessária a utilização do algoritmo genético. Para os quatro tipos de seleção testados com o sistema de 06 barras não existe muita diferença na qualidade dos resultados apresentando taxas de êxito de 65 a 75%, isto é, sobre um conjunto de 20 testes os métodos de seleção encontraram a solução ótima entre 13 e 15 dos testes. Entretanto, a seleção baseada em torneio encontra a melhor configuração mais rapidamente que os outros métodos de seleção apresentando uma média de 462 PL's.

Para o sistema Sul brasileiro foram realizados testes gerando a população inicial de duas formas: aleatoriamente e usando o algoritmo heurístico de Garver. Novamente, os quatro métodos de seleção encontram a configuração ótima do problema com taxas de êxito muito parecidas (entre 6 a 7 vezes sobre 20 testes com população gerada aleatoriamente e entre 13 e 17 vezes sobre 20 testes com população gerada usando o algoritmo de Garver). Entretanto, nesses testes aparece uma grande diferença entre os resultados encontrados, utilizando uma população inicial gerada aleatoriamente e outra utilizando o algoritmo de Garver. Essa diferença aparece na taxa de êxito em encontrar a solução ótima, assim como na diminuição do esforço computacional necessário para encontrar a melhor configuração. Assim, por exemplo, a seleção baseada em ordenamento incrementa sua taxa de êxito de 6 para 17 em 20 testes e o esforço computacional médio diminui de 4506 PL's para 1735 PL's.

Dos testes realizados com o sistema Sul brasileiro, pode-se concluir que é muito importante encontrar uma população inicial usando algoritmos heurísticos construtivos adequados. A melhoria encontrada é significativa e este fato faz com que o desempenho do algoritmo genético, na prática, dependa do ponto inicial de partida, especialmente em aplicações práticas. Pode-se observar também um desempenho ligeiramente superior da seleção proporcional e da seleção baseada em ordenamento na taxa de êxito em encontrar a configuração ótima, mas os métodos baseados em escalonamento linear e em torneio encontram mais rapidamente a melhor configuração (o que leva a um menor esforço computacional).



Para o sistema Norte-Nordeste brasileiro – Plano 2008, o desempenho dos quatro métodos de seleção apresentam resultados muito parecidos. A vantagem de seleção baseada em torneio é muito pequena. Nesse caso, todos os testes foram realizados gerando a população inicial usando o algoritmo de Garver. Quando é utilizado a população gerada aleatoriamente, dificilmente a melhor configuração encontrada encontra-se na faixa dos 2.600.000.000 US\$ e com esforços computacionais muito maiores.

A segunda bateria de testes realizados, incrementando a possibilidade de esforço computacional, foi especialmente interessante para a seleção baseada em escalonamento que encontrou uma topologia de melhor qualidade. Porém, os outros métodos de seleção apresentam nenhuma ou pequenas melhorias quando é incrementado o esforço computacional.

7.5. Resultados Encontrados em Relação ao Algoritmo Genético Apresentado em [1].

O algoritmo genético desenvolvido e apresentado neste trabalho é uma continuação natural do algoritmo genético desenvolvido em [1] e também contou com a colaboração de pesquisadores que participaram de [1].

1. Em relação ao algoritmo apresentado em [1] foram introduzidas melhorias na formulação dos novos caminhos (blocos construtivos). Os diferentes métodos de seleção foram analisados na sua parte teórica e experimentalmente. Para o sistema de 06 barras o esforço computacional é menor no algoritmo apresentado no presente trabalho para taxas de êxito equivalentes. Particularmente, quando é usada a seleção baseada em torneio o esforço computacional diminui significativamente.
2. Para o sistema de 46 barras (sistema de médio porte) o esforço computacional do algoritmo genético apresentado também é menor para taxas de êxito equivalentes. Esta diminuição do esforço computacional acontece com todos os métodos de seleção e particularmente, quando é usado o algoritmo de Garver modificado para gerar uma população inicial. Em [1] o algoritmo converge após 8.000 a 9.000 chamadas do algoritmo de PL quando a população inicial é gerada aleatoriamente e no algoritmo



apresentado esse esforço computacional fica na faixa de 4.000 a 6.000 PL's. Quando a população inicial é gerada usando o algoritmo heurístico de Garver o esforço computacional fica na faixa de 1.500 a 3.000 PL's, enquanto que no algoritmo apresentado em [1] esse esforço computacional está na faixa de 3.500 a 4.000 PL's.

3. Para o sistema Norte-Nordeste brasileiro de grande porte foram descobertas topologias melhores que as topologias encontradas em [1]. Três dos quatro métodos de seleção encontraram topologias de melhor qualidade, sendo a melhor delas a topologia com investimento de 2.582.843.000 US\$ encontrada usando o método de seleção com escalonamento. Entretanto, as topologias encontradas apresentam investimentos muito próximos. Deve-se mencionar que ainda não se conhece a solução ótima do sistema de grande porte analisado. Em resumo, o algoritmo apresenta menor esforço computacional para taxas de êxito equivalentes em sistemas pequenos e médios, e em sistemas de grande porte encontra topologias um pouco melhores que o algoritmo apresentado em [1]. Para encontrar melhorias adicionais, deve-se analisar e modificar outras partes do algoritmo genético.
4. A melhor topologia encontrada em [1] para o sistema Norte-Nordeste apresenta um investimento de 2.600.593.000 US\$ com a seguinte configuração:

$n_{1-2} = 1,$	$n_{2-4} = 1,$	$n_{4-5} = 4,$	$n_{4-81} = 3,$	$n_{5-56} = 1,$	$n_{5-58} = 4,$	$n_{6-75} = 1$	$n_{12-15} = 1,$
$n_{13-15} = 4,$	$n_{14-59} = 1,$	$n_{15-16} = 3,$	$n_{15-46} = 2,$	$n_{16-44} = 6,$	$n_{18-50} = 11,$	$n_{18-74} = 5,$	$n_{21-57} = 2,$
$n_{22-58} = 2,$	$n_{24-43} = 1,$	$n_{25-55} = 4,$	$n_{30-31} = 1,$	$n_{30-63} = 2,$	$n_{35-51} = 2,$	$n_{36-46} = 2,$	$n_{39-86} = 1,$
$n_{40-45} = 1,$	$n_{40-46} = 1,$	$n_{41-64} = 2,$	$n_{42-44} = 1,$	$n_{43-55} = 2,$	$n_{43-58} = 3,$	$n_{48-49} = 1,$	$n_{49-50} = 4,$
$n_{52-59} = 1,$	$n_{53-86} = 1,$	$n_{54-58} = 1,$	$n_{54-63} = 1,$	$n_{56-57} = 1,$	$n_{61-85} = 3,$	$n_{61-86} = 1,$	$n_{63-64} = 1,$
$n_{67-68} = 1,$	$n_{67-69} = 1,$	$n_{67-71} = 3,$	$n_{69-87} = 1,$	$n_{71-72} = 1,$	$n_{72-73} = 1,$	$n_{73-74} = 2,$	$n_{73-75} = 1,$
$n_{75-81} = 1$							

CAPÍTULO 8

Conclusões

Em relação ao trabalho desenvolvido, pode-se mencionar as seguintes conclusões:

1. O algoritmo implementado é eficiente e robusto, encontrando facilmente as topologias ótimas de sistemas elétricos de pequeno e de médio porte.
2. Os quatro métodos de seleção usados são competitivos e o desempenho comparativo entre eles depende do tamanho do problema. Particularmente o método baseado em torneio é eficiente em sistemas de pequeno e médio porte. Para sistemas de grande porte, o método baseado em escalonamento encontra topologias de melhor qualidade. No caso de sistemas de grande porte o esforço computacional assim como o número de gerações é elevado e era esperado que a seleção baseada em escalonamento apresenta-se bom desempenho pois esse método não perde seletividade quando as funções objetivos das configurações ficam muito próximas.

3. Os resultados encontrados são competitivos com os encontrados com outros algoritmos combinatoriais tais como, "Simulated Annealing" e "Tabu Search" para o problema de planejamento de sistemas de transmissão, ver referências [4,5].
4. Para o planejamento de sistemas reais, deve-se usar algoritmos heurísticos construtivos eficientes para gerar a população inicial. Esta proposta diminui muito o esforço computacional e permite encontrar topologias de excelente qualidade. Gerar a população inicial aleatoriamente leva a esforços computacionais proibitivos.
5. Melhorias devem ser buscadas através de modificações de outras partes do algoritmo. Uma proposta interessante consiste em encontrar uma estratégia eficiente de codificação de preferência usando estruturas de dados e eliminar a necessidade de uso do algoritmo de PL com a consequente diminuição do esforço computacional. Esta proposta transforma o algoritmo genético num algoritmo evolutivo.
6. Outra proposta de pesquisa consiste em realizar o planejamento dinâmico de sistemas de transmissão separando o horizonte de planejamento em um número de períodos ou estágios menores. Pode-se assim, considerar o problema de planejamento de sistemas de transmissão num sistema de mercado aberto ou de livre acesso.



REFERÊNCIAS BIBLIOGRÁFICAS

- [1] R. A. Gallego, A . Monticelli, R. Romero: "Transmission System Expansion Planning by Extended Genetic Algorithm", IEE Proceeding, Genetic Transmission Distribution, Vol.145, no. 3, pp.329-335, may 1998.
- [2] L. L. Garver: "Transmission Network Estimation Using Linear Programming", IEEE Trans. Power App. Syst., Vol. PAS-89, pp. 1688-16967, September-October, 1970.
- [3] A . Monticelli, A . Santos Jr., M. V. F. Pereira, S. H. Cunha. B. J. Parker, J. C. G. Praça: "Interactive Transmission Network Planning Using a Least-Effort Criterion", IEEE Trans. Power App. Systems, Vol. PAS-101, No. 10, October, 1982.
- [4] R. Romero, R. A . Gallego, A . Monticelli: "Transmission System Expansion Planning by Simulated Annealing ", IEEE Transactions on Power Systems, Vol. 11, No. 1, pp 364-369, February 1996.
- [5] R. A . Gallego, A . Monticelli, R. Romero: "Comparative Studies of Non-Convex Optimization Methods for Transmission System Expansion Planning", IEEE Transactions in Power Systems, Vol. 13, no. 3, pp. 822-828, August 1998.
- [6] M. Srinivas, L.M. Patnaik: "Genetic Algorithms", IEEE Computer, pp 17-26, 1994.

- [7] V. CA. Levi, M. S. Calovic : "A New Decomposition Based Method for Optimal Expansion Planning of Large Transmission Networks", IEEE Transactions on Power System, Vol. 6, No. 3, pp 937-943, October, 1991.
- [8] K. P. Wong, Y. W. Wong: "Combined Genetic Algorithms/ Simulated Annealing/ Fussy Set Approach to Short-Term Generation Scheduling with Take-or-Pay Fuel Contract", IEEE Transactions on Power Systems, Vol. 11, No. 1, pp 128-136, February 1996.
- [9] D. E. Goldberg: "Genetics Algorithms in Search, Optimization and Machine Learning", Addison Wesley, Reading, Massachusetts, 1989.
- [10] S. A . Kazarlis, A. G. Bakirtizis, V. Petridis: "A Genetics Algorithm Solution to the Unit Commitment Problem", IEEE Transactions on Power Systems, Vol. II, No. 1, pp 83-92, February 1996.
- [11] A . Diaz, F. Glover, H. Ghaziri, J. Gonzalez, M. Laguna, P. Moscato, F. Tseng: "Optimización Heurística y Redes Neuronales", Editorial Paraninfo, Madrid, 1996.
- [12] M. Mitchel: "Introduction to Genetics Algorithms", Vol.III, Editorial Cambridge: Mit Press, 1996.
- [13] Hancock P.J.B. : "Selecion Methods for Evolulutionary Algorithms: New Frontiers", Vol.II, Ed. by Lance Chambers, CRC Press, 1995.
- [14] Back T.: "Evolucionary Algorithm in Theory and Practice", Oxford University Press, 1996.
- [15] Goldberg D.E. , Deb K. A.: "Comparative Analysis of Selection Schemes Used in Genetic Algorithms", Rawlins G. J. E. Ed., Morgan Kaufmann, 1991.



- [16] Michalewicz Z.: "Genetic Algorithms + Data Structures = Evolution Programs", Artificial Intelligence, Springer, Berlin, 1992.
- [17] Goldberg D. E.: "Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley, 1989.
- [18] Pereira M. V. F. , Pinto L. M. V., Oliveira G. C., S. H. F. Cunha: " Composite Generation-Transmission Expansion Planning ", EPRI-Research Project 2473-9, Stanford University, 1987.
- [19] R. Romero, A. Monticelli: " A Hierarchical Decomposition Approach for Transmission Network Expansion Planning ", IEEE Transactions on Power Systems, Vol. 9, NO. 1, pp. 373-380, february 1994.



APÊNDICE A

Lista de Figuras

Figura 4.1: Codificação Binária.....	28
Figura 4.2: Representação Binária de uma Variável Contínua.....	29
Figura 4.3a: Configurações Candidatas a Recombinação.....	32
Figura 4.3b: Configurações Recombinadas.....	32
Figura 4.4: Exemplo de Mutação.....	34
Figura 4.5: Representação de um Esquema.....	36
Figura 4.6: Representação dos Possíveis Esquemas.....	36
Figura 4.7: Representação do Espaço Completo de Configurações.....	36
Figura 4.8: Representação de um Esquema de Ordem 3.....	37
Figura 4.9: Comprimento de um Esquema.....	37
Figura 5.1: Divisão da Roleta.....	44
Figura 5.2: Divisão da Roleta Padronizada.....	45
Figura 5.3: Seleção Baseada em Ordenamento.....	54
Figura 5.4: Exemplo de População.....	62
Figura 5.5: População Formada pela Melhor Configuração.....	62
Figura 6.1: O problema da Codificação.....	72

Figura 6.2: Diferentes Formas de Codificação.....	73
Figura 6.3: Recombinação e Mutação.....	75
Figura 6.4: Recombinação e Mutação no Sistema Binário.....	76
Figura 6.5: “Building Blocks”.....	82

Lista de Tabelas

Tabela 5.1: Esquema de Seleção Proporcional	42
Tabela 5.2: Início do Processo de Seleção Utilizando a Roleta Padronizada	44
Tabela 5.3: Distribuição do Número de Descendentes para Cada Configuração.....	45
Tabela 5.4: Seleção Estocástica do Resíduo.....	47
Tabela 5.5: Implementação da Seleção Determinística.....	49
Tabela 5.6: Configurações da População.....	50
Tabela 5.7: Número de Descendentes.....	51
Tabela 5.8: Implementação do Mecanismo de Seleção Usando Ordenamento Linear....	53
Tabela 5.9: Atribuição do Número de Descendentes para Cada Configuração.....	56
Tabela 5.10: Exemplo da Implementação da Técnica de Janelamento.....	58
Tabela 5.11: Escalonamento Sigma.....	59
Tabela 5.12: Esquema de seleções: Proporcional e Torneio.....	61
Tabela 5.13: Tempo de “Take Over” Associado a cada Tipo de Seleção.....	64
Tabela 5.14: Complexidade Computacional Associado a cada Tipo de Seleção	67
Tabela 7.1: Seleção Proporcional.....	88
Tabela 7.2: Seleção de Escalonamento.....	88



Tabela 7.3: Seleção de Ordenamento.....	89
Tabela 7.4: Seleção de Torneio.....	89
Tabela 7.5 (a),(b) e (c): Resultados Obtidos com Seleção de Ordenamento.....	90
Tabela 7.6 (a),(b) e (c): Resultados Obtidos com Seleção de Torneio.....	90
Tabela 7.7: Seleção Proporcional.....	92
Tabela 7.8: Seleção por Escalonamento Linear.....	92
Tabela 7.9: Seleção de Ordenamento	92
Tabela 7.10: Seleção de Torneio.....	92
Tabela 7.11: Seleção Proporcional.....	93
Tabela 7.12: Seleção por Escalonamento Linear.....	93
Tabela 7.13: Seleção de Ordenamento	94
Tabela 7.14: Seleção de Torneio.....	94
Tabela 7.15: Resultados Obtidos Utilizando Seleção Proporcional para o Sistema Norte-Nordeste Brasileiro.....	95
Tabela 7.16: Melhores Resultados Obtidos Utilizando os Diferentes Tipos de Seleção Para o Sistema Norte-Nordeste – Plano 2008.....	96
Tabela 7.17: Melhores Resultados Obtidos Relaxando o Critério de Parada do Algoritmo Utilizando os Diferentes Tipos de Seleção para o Sistema Norte-Nordeste – Plano 2008.....	96

Lista de Equações

Equação 2.1: Representação do Modelo DC.....	07
Equação 2.2: Representação do Modelo DC Equivalente a (2.1).....	08
Equação 2.3: Representação do Modelo DC Modificado.....	09



Equação 3.1: Representação do Modelo de Transportes.....	13
Equação 3.5: Representação do Modelo Híbrido.....	19

Lista de Símbolos e Abreviações

PNLIM – Programação não Linear Inteira Mista.....	01
PLIM - Programação Linear Inteira Mista.....	14
GA – Genetic Algorithms.....	03
SA – Simulating Annealing.....	03
TS – Tabu Search.....	03
v – Investimento.....	07
c_{ij} – Custo de uma Linha no Caminho i - j	07
n_{ij} – Número de Linhas Adicionados no Caminho i - j	08
n^0_{ij} – Número de Circuitos na Configuração Base no Caminho i - j	08
$\bar{n}_{ij}$ - Vetor de Número Máximo de Adições Permitidas no Caminho i - j	08
S – Matriz Incidência Nó-Ramo do Sistema Elétrico.....	07
f – Vetor de Fluxos.....	07
$\bar{f}_{ij}$ - Fluxo Máximo Permitido para um Circuito no Caminho i - j	07
f_{ij} – Fluxo Total no Caminho i - j	07
g – Vetor de Geração.....	07
$\bar{g}$ - Vetor de Máxima Capacidade de Geração nas Barras de Geração.....	07
d – Vetor de Demanda.....	07
θ_j e θ_i - Ângulos de Tensão nas Barras Terminais do Ramo i - j	08



Ω - Conjunto de Todas as Linhas Candidatas à Adição.....	07
γ_{ij}^0 - Susceptância Equivalente dos Circuitos Existentes no Ramo i-j na Configuração Base.....	08
γ_{ij} - Susceptância de um Circuito no Ramo i-j.....	08
x_{ij} - Susceptância Equivalente dos Circuitos Adicionados no Ramo i-j.....	08
B - Matriz de Susceptância do Sistema Elétrico.....	09
Ω_1 - Conjunto de todos os Ramos i-j em que Existem Circuitos na Configuração Base.	09
Ω_2 - Conjunto dos Ramos Novos onde é Possível Adicionar Circuitos.....	09
r_k - Geração Artificial (ou Fictícia) na Barra k.....	10
Γ - Conjunto de Todas as Barras k que Existe Demanda.....	10
$Sf+g = d$ - Equação Correspondente à Primeira Lei de Kirchhoff.....	07
$ f_{ij} \leq (n_{ij} + n_{ij}^0)$ - Capacidade de Transmissão dos Circuitos.....	07
$0 \leq g \leq \bar{g}$ - Limites de Geração em cada Caminho Candidato i-j.....	07
$0 \leq n_{ij} \leq \bar{n}_{ij}$ - Limites de Circuitos Adicionados em Cada Caminho i-j.....	07
$f_{ij} - (\gamma_{ij}^0 + x_{ij})(\theta_j - \theta_i) = 0$ - Restrições Correspondentes à Segunda Lei de Kirchhoff	07
$ f_{ij} \leq (x_{ij} + \gamma_{ij}^0)\bar{\phi}_{ij}$ - Restrições sobre a Capacidade de Transmissão.....	07
α - Parâmetro de Penalidade Suficientemente Grande para Tornar Pouco Atrativa Alternativas de Investimento com Valores de r_k Diferentes de Zero.....	10
MCC - Mínimo Corte de Carga.....	17
IS _{MCC} - Índice de Sensibilidade do Algoritmo de Mínimo Corte de Carga.....	17
P_c - Taxa de Recombinação.....	32
P_m - Taxa de Mutação.....	34

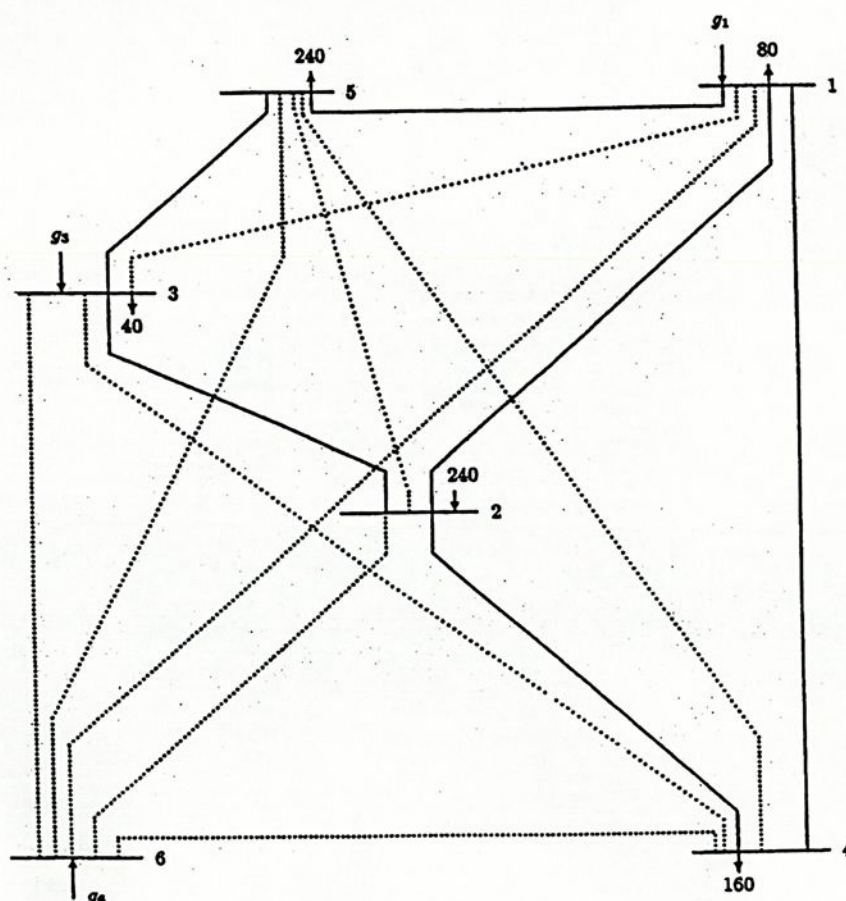


N_i – Número de Descendentes.....	42
$Z_i(x)$ – Valor da Função Objetivo de cada Configuração.....	42
$Z_m(x)$ – Média da população.....	42
N_{km} – Número máximo de descendentes atribuído as melhores configurações (seleção proporcional).....	48
$\bar{f}_m$ - Média dos Valores das Funções Objetivo.....	48
T_p - Tamanho da População.....	84
$K_{máx}$ – Critério de Parada.....	84
K_{stop} – Número Máximo de Configurações Permitidas.....	84
$K1$ – Contador do Número de Gerações em que a Incumbente Permanece Inalterada .	85
$K2$ – Contador do Número de Gerações.....	85
n_m – Número de Mutações.....	85

APÊNDICE B

Dados dos Sistemas Testados

B.1. Sistema de 06 Barras de Garver



Sistema de 06 Barras

B.1.1. Dados das barras

B.1.1. 1. Dados das barras considerando a reprogramação da geração

Nº da barra	Geração (MW)	Carga (MW)
1	150	80
2	0	240
3	360	40
4	0	160
5	0	240
6	600	0

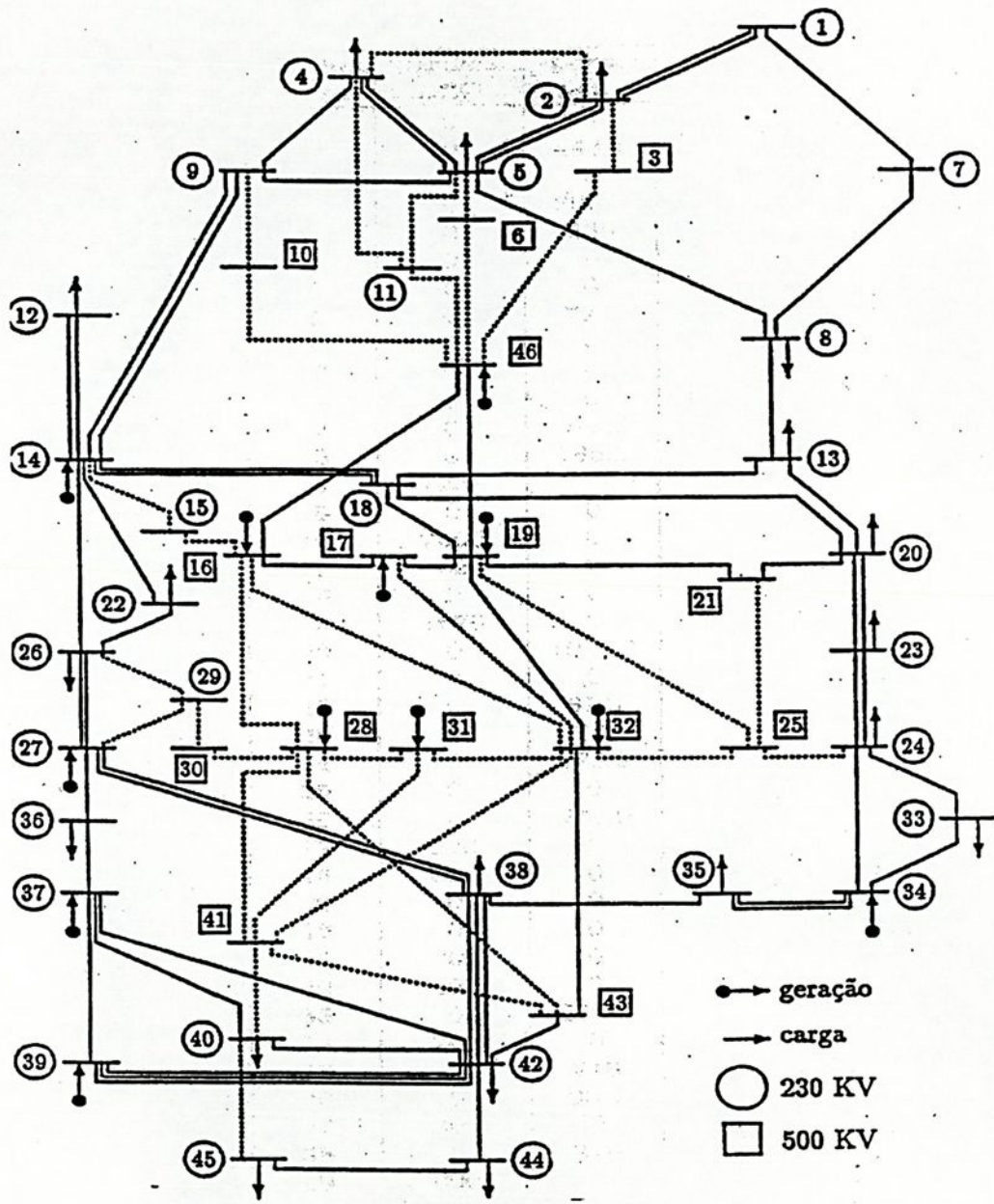
B. 1.1. 2. Dados das barras sem considerar a reprogramação da geração

Nº da barra	Geração (MW)	Carga (MW)
1	50	80
2	0	240
3	165	40
4	0	160
5	0	240
6	545	0

B. 1.2. Dados das linhas

Nº da linha	Barra inicial	Barra final	Capacidade de transmissão (MW)	Reatância (Ω)	Custo (us\$.10 ³)
1	1	2	100	0,4	40
2	1	3	100	0,38	38
3	1	4	80	0,6	60
4	1	5	100	0,2	20
5	1	6	70	0,68	68
6	2	3	100	0,2	20
7	2	4	100	0,4	40
8	2	5	100	0,31	31
9	2	6	100	0,30	30
10	3	4	82	0,59	59
11	3	5	100	0,20	20
12	3	6	100	0,48	48
13	4	5	75	0,63	63
14	4	6	100	0,3	30
15	5	6	78	0,61	61

B.2. Dados do Sistema Sul Brasileiro de 46 Barras



Sistema de 46 Barras



B. 2. 1. Dados das barras

B. 2. 1.1. Dados das barras com reprogramação da geração

Nº da barra	Geração (MW)	Carga (MW)
1	0	0
2	0	443,1
3	0	0
4	0	300,7
5	0	238
6	0	0
7	0	0
8	0	72,8
9	0	0
10	0	0
11	0	0
12	0	511,9
13	0	185,8
14	1257	0
15	0	0
16	2000	0
17	1050	0
18	0	0
19	1670	0
20	0	1091,2
21	0	0
22	0	81,9
23	0	458,1
24	0	478,2
25	0	0
26	0	231,9
27	220	0
28	800	0
29	0	0
30	0	0
31	700	0
32	500	0
33	0	229,1
34	748	0
35	0	216
36	0	90,1
37	300	0
38	0	216
39	600	0
40	0	262,1
41	0	0
42	0	1607,9
43	0	0
44	0	79,1
45	0	86,7
46	700	0

B. 2. 1. 2 Dados das barras sem a reprogramação da geração

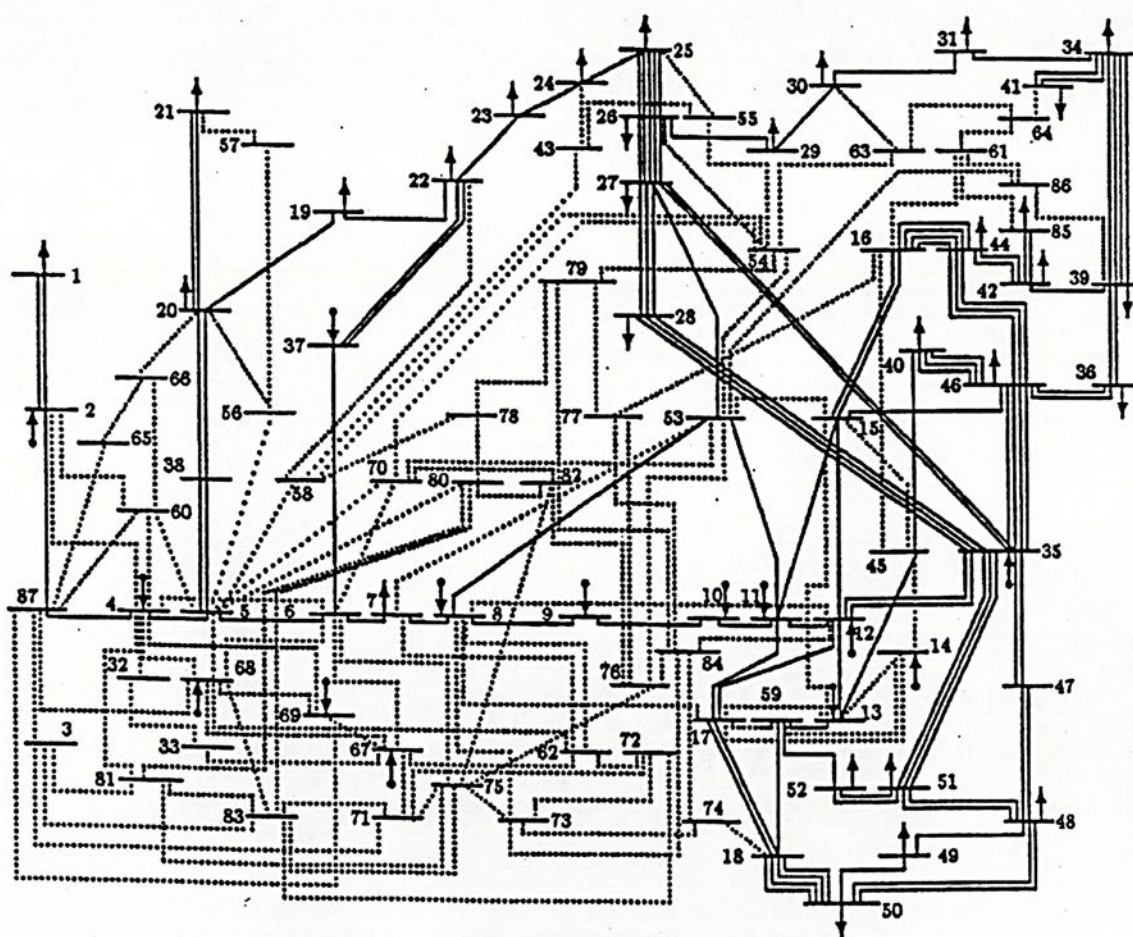
N ^o da barra	Geração (MW)	Carga (MW)
1	0	0
2	0	443,1
3	0	0
4	0	300,7
5	0	238,0
6	0	0
7	0	0
8	0	72,2
9	0	0
10	0	0
11	0	0
12	0	511,9
13	0	185,8
14	944,0	0
15	0	0
16	1366,0	0
17	1000,0	0
18	0	0
19	773,0	0
20	0	1091,2
21	0	0
22	0	81,9
23	0	458,1
24	0	478,2
25	0	0
26	0	231,9
27	54,0	0
28	730,0	0
29	0	0
30	0	0
31	310,0	0
32	450,0	0
33	0	229,1
34	221,0	0
35	0	216,0
36	0	90,1
37	212,0	0
38	0	216,0
39	221,0	0
40	0	262,1
41	0	0
42	0	1607,9
43	0	0
44	0	79,1
45	0	86,7
46	599,0	0

B. 2. 2. Dados das linhas

Nº da linha	Barra inicial	Barra final	Capacidade de transmissão (MW)	Reatância (Ω)	Custo (us\$.10 ³)
1	1	7	270	0,0616	4349
2	1	2	270	0,1065	7076
3	4	9	270	0,0924	6217
4	5	9	270	0,1173	7732
5	5	8	270	0,1132	7480
6	7	8	270	0,1023	6823
7	4	5	270	0,0566	4046
8	2	5	270	0,0324	2581
9	8	13	240	0,1348	8793
10	9	14	220	0,1756	11267
11	12	14	270	0,0740	5106
12	14	18	240	0,1514	9803
13	13	18	220	0,1805	11570
14	13	20	270	0,1073	7126
15	18	20	200	0,1997	12732
16	19	21	1500	0,0278	32632
17	16	17	2000	0,0078	10505
18	17	19	2000	0,0061	8715
19	14	26	220	0,1614	10409
20	14	22	270	0,0840	5712
21	22	26	270	0,0790	5409
22	20	23	270	0,0932	6268
23	23	24	270	0,0774	5308
24	26	27	270	0,0832	5662
25	24	34	220	0,1647	10611
26	24	33	240	0,1448	9399
27	33	24	270	0,1265	8288
28	27	36	270	0,0915	6167
29	27	38	200	0,2080	13237
30	36	37	270	0,1057	7025
31	34	35	270	0,0491	3591
32	35	38	200	0,1980	12631
33	37	39	270	0,0283	2329
34	37	40	270	0,1281	8389
35	37	42	200	0,2105	13388
36	39	42	200	0,2030	12934
37	40	42	270	0,0932	6268
38	38	42	270	0,0907	6116
39	32	43	1400	0,0309	35957
40	42	44	270	0,1206	7934
41	44	45	200	0,1864	11924
42	19	32	1800	0,0195	23423
43	46	19	1800	0,0222	26365
44	46	16	1800	0,0203	24309
45	18	19	600	0,0125	8178
46	20	21	600	0,0125	8178
47	42	43	600	0,0125	8178
48	2	4	270	0,0882	5965
49	14	15	270	0,0374	2884
50	46	10	2000	0,0081	10889
51	4	11	240	0,2246	14247
52	5	11	270	0,0915	6167
53	46	6	2000	0,0128	16005
54	46	3	1800	0,0203	24319
55	16	28	1800	0,0222	26365
56	16	32	1400	0,0311	36213
57	17	32	1700	0,0232	27516
58	19	25	1400	0,0325	37748
59	21	25	2000	0,0174	21121
60	25	32	1400	0,0319	37109
61	31	32	2000	0,0046	7052
62	28	31	2000	0,0053	7819
63	28	30	2000	0,0058	8331
64	27	29	270	0,0998	6672

65	26	29	270	0,0541	3894
66	28	41	1300	0,0339	39283
67	28	43	1200	0,0406	46701
68	31	41	1500	0,0278	32632
69	32	41	1400	0,0309	35957
70	41	43	2000	0,0139	17284
71	40	45	180	0,2205	13994
72	15	16	600	0,0125	8178
73	46	11	600	0,0125	8178
74	24	25	600	0,0125	8178
75	29	30	600	0,0125	8178
76	40	41	600	0,0125	8178
77	2	3	600	0,0125	8178
78	5	6	600	0,0125	8178
79	9	10	600	0,0125	8178

B. 3. Sistema Norte – Nordeste Brasileiro de 87 Barras



Sistema de 87 Barras

B. 3.1. Dados das barras com reprogramação da geração

Nº da barra	Geração (MW)	Carga (MW)
1	0	1857
2	4048	0
3	0	0
4	517	0
5	0	0
6	0	0
7	0	31
8	403	0
9	465	0
10	538	0
11	2200	0
12	2257	0
13	4510	0
14	542	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	86
20	0	125
21	0	722
22	0	291
23	0	58
24	0	159
25	0	1502
26	0	47
27	0	378
28	0	189
29	0	47
30	0	189
31	0	110
32	0	0
33	0	0
34	0	28
35	1635	0
36	0	225
37	169	0
38	0	0
39	0	186
40	0	1201
41	0	520
42	0	341
43	0,5	0
44	0	4022

45	0	0
46	0	205
47	0	0
48	0	347
49	0	777
50	0	5189
51	0	290
52	0	707
53	0	0
54	0,5	0
55	0	0
56	0	0
57	0	0
58	0	0
59	0	0
60	0	0
61	0,5	0
62	0	0
63	0	0
64	0	0
65	0,5	0
66	0	0
67	1242	0
68	888	0
69	902	0
70	0	0
71	0,5	0
72	0,5	0
73	0	0
74	0	0
75	0,5	0
76	0	0
77	0	0
78	0	0
79	0,5	0
80	0,5	0
81	0	0
82	0,5	0
83	0,5	0
84	0,5	0
85	0	487
86	0	0
87	0	0

B. 3.2 Dados das linhas

Nº da linha	Barra inicial	Barra final	Capacidade de transmissão (MW)	Reatância (Ω)	Custo (us\$.10 ³)
1	1	2	1000	0,0374	44056
2	2	4	1000	0,0406	48880
3	2	60	1000	0,0435	52230
4	2	87	1000	0,0259	31192
5	3	71	3200	0,0078	92253
6	3	81	3200	0,0049	60153
7	3	83	3200	0,0043	53253
8	3	87	1200	0,0058	21232
9	4	5	1000	0,0435	52230
10	4	6	1000	0,0487	58260
11	4	32	300	0,0233	7510
12	4	60	1000	0,0215	26770
13	4	68	1000	0,007	10020
14	4	69	1000	0,0162	20740
15	4	81	1200	0,0058	21232
16	4	87	1000	0,0218	26502
17	5	6	1000	0,0241	29852
18	5	38	600	0,0117	8926
19	5	56	1000	0,0235	29182
20	5	58	1000	0,022	27440
21	5	60	1000	0,0261	32130
22	5	68	1000	0,0406	48880
23	5	70	1000	0,0464	55580
24	5	80	1200	0,0058	21232
25	6	7	1000	0,0288	35212
26	6	37	300	0,0233	7510
27	6	67	1000	0,0464	55580
28	6	68	1000	0,0476	56920
29	6	70	1000	0,0371	44860
30	6	75	1200	0,0058	21232
31	7	8	1000	0,0234	29048
32	7	53	1000	0,0452	54240
33	7	62	1000	0,0255	31460
34	8	1	1000	0,0186	23420
35	8	12	1000	0,0394	47540
36	8	17	1000	0,0447	53570
37	8	53	1200	0,0365	44190
38	8	62	1000	0,0429	51560
39	8	73	1200	0,0058	21232
40	9	10	1000	0,0046	7340
41	10	11	1000	0,0133	17390
42	11	12	1200	0,0041	6670
43	11	15	1200	0,0297	36284
44	11	17	1200	0,0286	35078
45	11	53	1000	0,0254	31326
46	12	13	1200	0,0046	7340
47	12	15	1200	0,0256	31594
48	12	17	1200	0,0246	30388
49	12	35	600	0,0117	8926
50	12	84	1200	0,0058	21232
51	13	14	1200	0,0075	10690
52	13	15	1200	0,0215	26770
53	13	17	1200	0,0232	28780
54	13	45	1200	0,0290	35480
55	13	59	1200	0,0232	28780
56	14	17	1200	0,0232	28780
57	14	25	1200	0,0232	28780
58	14	59	1200	0,0157	20070
59	15	16	1200	0,0197	24760
60	15	45	1200	0,0103	13906
61	15	46	600	0,0117	8926
62	15	53	1000	0,0423	50890
63	16	44	600	0,0117	8926



64	16	45	1200	0,022	27440
65	16	61	1000	0,0128	16720
66	16	77	1200	0,0058	21232
67	17	18	1200	0,017	21678
68	17	59	1200	0,017	21678
69	18	50	600	0,0117	8926
70	18	59	1200	0,0331	40170
71	18	74	1200	0,0058	21232
72	19	20	170	0,0934	5885
73	19	22	170	0,1877	11165
74	20	21	300	0,0715	6960
75	20	38	300	0,1382	12840
76	20	56	600	0,0117	8926
77	20	66	170	0,2064	12210
78	21	57	600	0,0117	8926
79	22	23	170	0,1514	9130
80	22	37	170	0,2015	11935
81	22	58	300	0,0233	7510
82	23	24	170	0,1651	9900
83	24	25	170	0,2153	12705
84	24	43	300	0,0233	7510
85	25	26	300	0,1073	29636
86	25	55	600	0,0117	8926
87	26	27	300	0,1404	25500
88	26	29	170	0,1081	6710
89	26	54	600	0,0117	8926
90	27	28	170	0,0826	5335
91	27	35	300	0,1367	25000
92	27	53	600	0,0117	8926
93	28	35	170	0,1671	9900
94	29	30	170	0,0688	4510
95	30	31	170	0,0639	4235
96	30	63	300	0,0233	7510
97	31	34	170	0,1406	8525
98	32	33	170	0,1966	11660
99	33	67	300	0,0233	7510
100	34	39	170	0,116	7150
101	34	41	170	0,0993	6215
102	35	46	170	0,2172	12705
103	35	47	170	0,1327	8085
104	35	51	170	0,1602	9625
105	36	39	170	0,1189	7315
106	36	46	170	0,0639	4235
107	39	42	170	0,0973	6105
108	39	86	300	0,0233	7510
109	40	45	600	0,0117	8926
110	40	46	170	0,0875	5500
111	41	64	300	0,0233	7510
112	42	44	170	0,0698	4565
113	42	85	170	0,0501	3465
114	43	55	1000	0,0254	31326
115	43	58	1000	0,0313	38160
116	44	46	170	0,1671	10010
117	47	48	170	0,1966	11660
118	48	49	170	0,0757	4895
119	48	50	170	0,0256	2090
120	48	51	170	0,2163	12760
121	49	50	170	0,0835	5335
122	51	52	170	0,056	3795
123	52	59	600	0,0117	8926
124	53	54	1000	0,027	32120
125	53	70	1000	0,0371	44860
126	53	76	1200	0,0058	21232
127	53	86	1000	0,0389	46870
128	54	55	1000	0,0206	25028
129	54	58	1000	0,051	60943
130	54	63	1000	0,0203	25430
131	54	70	1000	0,036	43520
132	54	79	1200	0,0058	21232
133	56	57	1000	0,0122	16050



134	58	78	1200	0,0058	21232
135	60	66	300	0,0233	7510
136	60	87	1000	0,0377	45530
137	61	64	1000	0,0186	23420
138	61	85	300	0,0233	7510
139	61	86	1000	0,0139	18060
140	62	67	1000	0,0464	55580
141	62	68	1000	0,0557	66300
142	62	72	1200	0,0058	21232
143	63	64	1000	0,029	35480
144	65	66	170	0,3146	18260
145	65	87	300	0,0233	7510
146	67	68	1000	0,029	35480
147	67	69	1000	0,0209	26100
148	67	71	1200	0,0058	21232
149	68	69	1000	0,0139	18060
150	68	83	1200	0,0058	21232
151	68	87	1000	0,0186	23240
152	69	87	1000	0,0139	18060
153	70	82	1200	0,0058	21232
154	71	72	3200	0,0108	125253
155	71	75	3200	0,0108	125253
156	71	83	3200	0,0067	80253
157	72	73	3200	0,01	116253
158	72	83	3200	0,013	149253
159	73	74	3200	0,013	149253
160	73	75	3200	0,013	149253
161	73	84	3200	0,0092	107253
162	74	84	3200	0,0108	125253
163	75	76	3200	0,0162	185253
164	75	81	3200	0,0113	131253
165	75	82	3200	0,0086	101253
166	75	83	3200	0,0111	128253
167	76	77	3200	0,013	149253
168	76	82	3200	0,0086	101253
169	76	84	3200	0,0059	70953
170	77	79	3200	0,0151	173253
171	77	84	3200	0,0115	132753
172	78	79	3200	0,0119	137253
173	78	80	3200	0,0051	62253
174	79	82	3200	0,0084	98253
175	80	81	3200	0,0101	117753
176	80	82	3200	0,0108	125253
177	80	83	3200	0,0094	110253
178	81	83	3200	0,0016	23253
179	82	84	3200	0,0135	155253



APÊNDICE C

Artigo Publicado:

Experimental Analysis of Selection Methods in a Genetic Algorithm to the Planning Electrical Transmission.. In: Internacional Conference on CAD/CAM , Robotics, and Factories of the Future, Águas de Lindóia, SP, Brazil, 18-20 August, 15 th ISPE/IEE, 1999

Experimental Analysis of Selection Methods in a Genetic Algorithm Applied to the Planning Electrical Transmission Systems

R. Romero - M. Mantovani
UNESP-Ilha Solteira-Brasil
15385.000 Ilha Solteira-SP
ruben@dee.feis.unesp.br

R.A. Gallego
UTP-Colombia
Pereira-Risaralda
ralfonso@utp.edu.co

A. Monticelli
UNICAMP-Campinas-Brasil
13081.970 Campinas-SP
alcir@dsee.fee.unicamp.br

Abstract: This paper discusses several alternative implementations of a genetic algorithm applied to power network expansion planning. Different selection techniques have been implemented and tested for comparison purposes. The selection operation in GA algorithms has been a hot research topic in recent years. This paper is an extension of [1] with significant improvements regarding the selection operation. Tests have been performed with systems available in the literature.

Keywords: genetic algorithms, selection techniques, transmission network expansion planning.

INTRODUCTION

This paper presents a comparative analysis of several selection techniques in a genetic algorithm for transmission network expansion planning. Only the static transmission planning problem is considered: in this problem, given an initial configuration and the data regarding generation/loads at a future year (the horizon year) a minimum cost expansion plan is sought. The static expansion problem is part of a more general problem called the dynamic expansion planning problem which solves the expansion planning problem through time. While the static planning problem answer the question *where* and *what type* of new equipment should be added, the dynamic formulation also tells *when* these pieces of equipment should be added (Hence, dynamic planning take into consideration the economies of scale.)

The transmission network is modeled by a DC power flow. The static expansion problem is then formulated as a mixed integer nonlinear program. This is a complex combinatorial problem that can suffer from combinatorial explosion. (For a more detailed discussion about the transmission expansion problem see [7, 8, 9].)

MATHEMATICAL FORMULATION: THE DC MODEL

In the following the static transmission expansion planning problem is formulated as a mixed integer nonlinear programming problem in which the power network is represented by a DC power flow model:

$$\begin{aligned} \text{Min} \quad & v = \sum_{ij} c_{ij} n_{ij} + \sum_i \alpha_i r_i \\ \text{s.a.} \quad & B(x + \gamma^o)\theta + g + r = d \\ & (x_{ij} + \gamma_{ij}^o)|\theta_i - \theta_j| \leq (x_{ij} + \gamma_{ij}^o)\bar{\phi}_{ij} \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq r \leq d \\ & 0 \leq n_{ij} \leq \bar{n}_{ij} \end{aligned} \quad (1)$$

where c_{ij} is the cost of the addition of a circuit in branch $i - j$, x_{ij} is the total susceptance in branch $i - j$, $B(\cdot)$ is the susceptance matrix, θ is the vector of initial susceptances, whose elements are γ_{ij}^o , i.e., the summation of the susceptances in branch $i - j$ at the beginning of the optimization, n_{ij} is the number of the circuits added in branch $i - j$, $\bar{n}_{ij} = \frac{\bar{f}_{ij}}{\gamma_{ij}}$, where γ_{ij} is the susceptance of the new circuits, $\bar{\phi}_{ij}$ is the defined as the ratio: $\bar{\phi}_{ij} = \frac{\bar{f}_{ij}}{\gamma_{ij}}$, where $\bar{f}_{ij}$ is the maximum flow in a circuit $i - j$, d is the demand vector, g is the generation vector, $\bar{g}$ is the vector of maximum generation capacity, r is the vector of artificial generation and α penalty parameter associated with loss of load caused by lack of transmission capacity.

Problem (1) is a mixed integer non linear problem. It is a typical hard combinatorial problem prone to combinatorial explosion as the number of decision variables increases. An extra complication regards the fact that there are cases in which planning does not simply mean the reinforcement of an existing network. Sometimes we have to start from scratch, at least for parts of the network due to the addition of new load, transition and generation buses. In the case of the Brazilian example systems, the simple addition of one or two circuits (lines or transformers) may not be enough to guarantee network connectivity: situation are not uncommon in which entire paths have to be built to put all the pieces together in a single network. In these cases, the combinatorial burden is even heavier than it would be in a simpler, reinforcement only, type of problem.

For a given set of x_{ij}^k , which may form a tentative solution, Problem (1) becomes a linear programming problem, Problem (2), which is solved in order to test



the adequacy of a candidate solution; adequacy is indicated by zero loss of load. Notice that Problem (1) is always feasible due to the presence of the loss of load factor $\sum_i \alpha_i r_i$ in the objective function; thus whenever

a tentative solution set x_{ij}^k is inadequate, feasibility is achieved by the use of artificial generators (loss of load). It has been observed that this feature facilitates the evolution of the process in order to come out from local minima by temporarily moving through regions of inadequate solutions but still keeping the feasibility of the mathematical problem;

$$\begin{aligned} \text{Min } w &= \sum_i \alpha_i r_i \\ \text{s.t. } & B(x^k + \gamma^o)\theta + g + r = d \\ & (x_{ij}^k + \gamma_{ij}^o)|\theta_i - \theta_j| \leq (x_{ij}^k + \gamma_{ij}^o)\bar{\phi}_{ij} \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq r \leq d \end{aligned} \quad (2)$$

THE GENETIC ALGORITHM FOR TRANSMISSION PLANNING

The genetic algorithm described here is an extension of the algorithm discussed in [1]. Several alternative selection schemes have been tried using the same basic algorithm. Other parts of the algorithm of Ref. [1] have been modified as well in order to profit from the special characteristics of the transmission planning problem.

For simplicity of the presentation, the implementation of a genetic algorithm can be decomposed into the following steps:

- codification and definition of the objective function,
- definition of a selection operator,
- definition of the recombination and mutation mechanisms,
- definition of the population size and of the stop criterion,
- implementation of special features of the problem being solved.

Codification and representation of the objective function:

Problem (1) is formulated in terms of integer and continuous variables. When the integer variables are fixed, it results the linear program (2). Hence, codification can be implemented by defining a configuration by the integer variables and to determine the values of the real variables by solving this linear program [1].

An alternative implementation of codification consists in representing configurations in binary form. In this method the integer variables are transformed into binary numbers. As discussed in [1], however, this

alternative is not very efficient. A third type of codification consists in defining a configuration by both integer and real variables in which case the subsidiary problem (2) is eliminated from consideration. This approach is currently being investigated; the genetic operators are modified and the problem is transformed into an evolutive programming problem.

The value of the objective function is determined after LP runs are carried out for each configuration in Problem (2).

The transmission planning problem is a cost minimization problem. This problem is transformed into a maximization problem by using one of the following: (a) $\min v \iff \max 1/v$, or (b) $\min v \iff \max \{K - v\}$. This transformation is used only for those types of selection where there is a need to put the original problem into a standard maximization format.

Implementation of the Selection Scheme

Several selection techniques are available in the literature. A selection operator determines the number of descendents of each configuration. Initially the proportional selection scheme has been proposed, in which case the number of descendents is proportional to the value of fitness function of each configuration. Later on a variety of selection operators have been introduced. Current research focus both theoretical aspects as well as experimental testing of the various approaches [2, 3, 4].

Method 1: Proportional selection.

There are several different ways of performing proportional selection, depending on (a) how the current value of the fitness function is taken into consideration and (b) how the number of descendents is approximated to an integer number. In the research reported in this paper the *stochastic remainder technique* with an upper bound on the number of descendents allowed for the best current configurations has been chosen [1]. As for the determination of an integer number of descendents it proceeds in two steps: (1) the integer part is determined deterministically, and (2) the fractional part is determined stochastically in the same way it is done in the proportional selection. This proposal avoids problems related to superconfigurations and improves the processing speed since part of the problem is solved deterministically. Tuning requirements are also reduced, since the only parameter that has to be set is the maximum number of descendents per configuration.

Method 2: Scaling selection.

This alternative has been suggested as a means of avoiding two common problems associated with the proportional selection: (1) the occurrence of superconfigurations at early stages of the optimization process (premature convergence), and (2) the development of a homogeneous set of configurations at the final stages of the optimization process (loss of selectivity). In this



type of selection the adaptation functions are modified by the transformation

$$f' = af + b \quad (3)$$

where f is the original adaptation function and f' is the transformed adaptation function. The values of the parameters a and b are normally obtained using the following transformation points: (1) the average value is constant, hence, $f'_m = f_m$, and (2) f' associated with the best configuration is a multiple of that associated with the mean value. As in the previous method, in this strategy there is only one parameter to be specified, the parameter δ which appears in the expression $f'_{max} = \delta f_m$, where normally $\delta \in [1.2, 2]$.

Method 3: Rank based selection

The ranking is based on the relative values of the adaptation functions. This type of selection also avoids the two problems affecting the proportional selection mentioned above. In addition, (1) it does not depend whether the problem is formulated as a minimization or maximization problem, and (2) it is not affected by the values of the adaptation function (positive or negative). Hence, selection depends only on the ranking.

To determine the number of descendents it is used an assignment function $\alpha(x)$ with some special features. Alternatively, a linear decreasing function can be used which depends on only two parameters: (1) the number of descendents of the best configuration, and (2) the number of configurations that should form part of the selection process; this can be part of or the entire population.

Method 4: Tournament selection

Both conceptually and computationally this is a very simple scheme. Given a population with n_{pop} configurations n_{pop} games are played where in each game there are n_k players (configurations). The winner configuration is the one with the highest quality and this one has the right to have one descendent. As with the two previous methods this method also avoids the deficiencies observed in simple proportional selection approach (superconfigurations and homogeneity). Also, as with the previous scheme, the approach works with both minimization and maximization, regardless the values assumed by the adaptation function. Additionally, it yields an integer number of descendents, which does not happen with the other alternative schemes. Only one parameter has to be chosen in this method, namely the number of configurations that take part of the game, n_k .

Crossover e mutation:

The crossover operator is conducted in the traditional way by choosing the crossover multipoint where the number of crossover points is increased. It is taken into consideration the size of the power grid, or the dimension of the vector of configurations.

The mutation operator is redefined to take into consideration the decimal codification. Hence, once the mutation point is chosen, the current position should be increased or decreased by one. In the transmission planning problem, this means that the number of circuits added to a right-of-way is increased or decreased by one.

Alternatively, the mutation operator can be implemented in such a way that multiple additions/removals can be performed; e.g., two or three circuits per move. Such mutation operator, however, should have a decreasing acceptance rate [1, 5].

Population size:

The size of the population is chosen taking into consideration the rates of crossover and mutation. Three control variables have to be tuned for each specific problem being solved. The size of population, for example, increases with the size of the problem. A detailed discussion about parameter tuning can be found in [1].

Stopping criterion:

The main part of the computational burden in this type of problem is related to the repeated solution the LP programs described above. In [1] a mixed stopping criterion have been used. If any of the following conditions is satisfied the algorithm stops: (1) maximum number of LP runs is exceeded and (2) the incumbent does not improve for a certain number of generations. This criterion has been also adopted here for the large scale example. As for the small and medium size networks the stopping criterion is simply based on the number of LP routine calls.

Special operators:

Special operators such as the "building blocks" described in [1] have also been used herein. These operators take advantage of the special characteristics of the transmission planning problem.

TEST RESULTS

In order to evaluate the relative performance of the selection operators discussed above tests have been conducted on three networks: the 6-bus Garver system and two real-life Brazilian systems (the Southern and the North-Northeastern systems). In the following, 20 representative test results are presented showing two main characteristics: the number of times that the algorithm finds the optimal solution, and (2) the number of LP solutions needed to find the optimal solution (or the best configuration find in the process) for the first time. In all cases the algorithm stops after running a given number of LP solutions. Generation levels have been considered to be fixed (No rescheduling allowed, which make the transmission planning problem harder to solve.)

Table 1: Results of the tests with the Garver system.

No.	Selection 1		Selection 2	
	<i>v</i>	NPL	<i>v</i>	NPL
1	248	990	316	900
2	200	690	200	1020
3	200	810	200	420
4	272	450	200	450
5	200	630	238	930
6	200	570	341	330
7	200	720	200	750
8	200	870	200	1140
9	200	840	248	1020
10	301	570	200	1020
11	200	720	200	1230
12	200	960	200	450
13	200	330	296	1200
14	231	210	200	1230
15	200	540	200	780
16	200	1080	200	570
17	272	240	200	840
18	282	390	200	300
19	200	780	200	540
20	231	990	231	900

Table 1: Continued.

No.	Selection 3		Selection 4	
	<i>v</i>	NPL	<i>v</i>	NPL
1	282	390	200	480
2	200	540	282	360
3	200	690	200	570
4	200	1110	328	240
5	200	390	200	360
6	200	660	200	750
7	329	1080	200	960
8	200	570	200	480
9	200	660	200	480
10	231	1200	200	480
11	200	1230	200	480
12	200	510	282	480
13	200	480	315	480
14	231	1140	272	480
15	200	420	200	480
16	200	360	282	480
17	200	1230	200	480
18	231	390	334	480
19	200	270	200	480
20	200	420	200	480

Table 2: Summary of results of tests for the Garver system

Selection	NOP	NMPL
1	13	669
2	14	801
3	15	687
4	13	462

The Garver network

All data for this system can be found in [7]. In this case, the configuration of the initial population were obtained randomly (in the other cases reported herein initial population were determined by a heuristic constructive algorithm as discussed above). The optimal solution for this problem is known (Figure 1) and the associated cost is of $v = 200$.

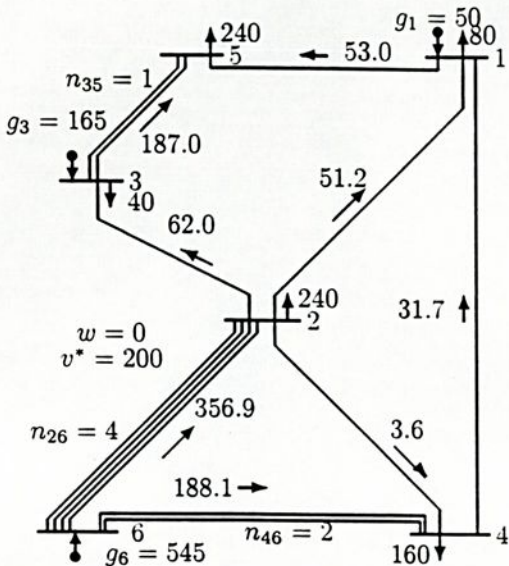


Figure 1: The optimal solution for the Garver problem

The following parameters have been used in this case: $\alpha = 10$, population $n_{pop} = 30$, maximum number of generations $n_{max} = 40$, crossover rate $\rho_c = 0.90$, and mutation rate $\rho_m = 0.13$. In all tests experimentally tuned data have been used as in [1]. For Method 1, the maximum number of descendents was set at $n_{md} = 4$, for Method 2, $\delta = 1.5$, for Method 3 the maximum number of descendents was set at $n_{md} = 2$ with 25 configuration taking part of the selection, and for Method 4 six configurations take part in each game.

Table 1 summarizes the main characteristics of the tests performed. The corresponding results are shown in Table 2 where v is the cost of the incumbent at the end of the process, NOP is the number of times that this incumbent is found during the process, and NMPL is the number average of LP runs in the 20 tests cases performed.

The Brazilian Southern System

All relevant data for this system can be found in [9]. In this case the initial population was generated (1) by random process and (2) using a modified version of the Garver constructive algorithm [1, 7]. The optimal solution for his example is \$ US $v = 154.420.000$.

The following parameters have been used: $\alpha = 1000$, population size of $n_{pop} = 50$ for population gener-

ated by the modified Garver algorithm and of 60 for populations generated randomly, the maximum number of generations of $n_{max} = 100$, crossover rate of $\rho_c = 0.90$, and mutation rate of $\rho_m = 0.05$. For Method 1 $n_{md} = 2$, for Method 2, $\delta = 1.5$, for Method 3 $n_{md} = 2$ with and 40 configurations, and for Method 4 three configurations take part in each game.

Table 3 summarized the main results obtained.

Table 3: Summary of the results obtained for the Southern system.

Selection	Random		Garver's	
	<i>NOP</i>	<i>NPL</i>	<i>NOP</i>	<i>NPL</i>
1	7	5117	16	1935
2	7	4758	15	1272
3	6	4506	17	1735
4	7	4185	13	1285

The Brazilian North-Northeastern System

This is a large scale combinatorial problem since the network has 87 buses, 187 branches and power demand of 29754 MW (See a topology of this system in Figure 2). In this case the initial population was obtained from the modified Garver approach. The number of LP calls per generation is significantly higher than for the other two examples discussed above due to the need for a more efficient mutation scheme. The stopping criterion is the same used in [1].

The following parameters have been used: $\alpha = 1000$, population size of $n_{pop} = 180$, maximum number of generations of $n_{max} = 700$, crossover rate of $\rho_c = 0.90$, and a variable mutation rate in the range $\rho_m = [0.01, 0.05]$. For Method 1 $n_{md} = 3$, for Method 2, $\delta = 1.5$, for Method 3 $n_{md} = 2$ and with the number of configurations in the range $[120, 140]$, and for Method 4 two configurations take part in each game.

Table 4 summarizes the best configuration found in the process (costs in $10^3 \times US\$$), for each selection operator after conducting 20 tests for each method. The average value of LP runs is also given in 1000 runs.

Table 4: Summary of the tests for the North-Northeastern system.

Selection	<i>v_{best}</i>	<i>NPL</i>
1	2594517	302
2	2595866	289
3	2603000	318
4	2595957	277

CONCLUSIONS

A efficient genetic algorithm for transmission network planning has been presented in the paper. The proposed method was able to find optimal solutions for

small and medium size networks. For large, complex networks excellent sets of solutions were found, both regarding costs and the structure of the proposed configurations.

Regarding the various types of selection studied in the paper, it was observed that all of them work satisfactorily, although the tournament selection was the best for finding the optimal configuration for the first time.

It is also noteworthy the improvement in the performance of the genetic algorithm when the initial population was obtained using a heuristic constructive approach (the modified Garver's approach). As above, the improvements were observed regarding both cost and quality of the solutions.

References

- [1] Gallego R.A., Monticelli A., Romero R.: "Transmission System Expansion Planning by an Extended Genetic Algorithm", IEE Proc.-Gener. Transm. Distrib., Vol. 145, No. 3, May 1998.
- [2] Hancock P.J.B.: "Selection Methods for Evolutionary Algorithms", Practical Handbook of Genetic Algorithms: New Frontiers, Vol. II, Ed. by Lance Chambers, CRC Pres, 1995.
- [3] Back T.: "Evolutionary Algorithms in Theory and Practice", Oxford University Pres, 1996.
- [4] Goldberg D.E., Deb K.A.: "Comparative Analysis of Selection Schemes Used in Genetic Algorithms", Rawlins G.J.E. Ed., Morgan Kaufmann, 1991.
- [5] Michalewicz Z.: "Genetic Algorithms + Data Structures = Evolution Programs", Artificial Intelligence, Springer, Berlin, 1992.
- [6] Goldberg D.E.: "Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley, 1989.
- [7] Garver L.L.: "Transmission Network Estimation Using Linear Programming", IEEE Trans. Power App. Syst., Vol. PAS-89, pp. 1688-1697, 1970.
- [8] Monticelli A., Santos Jr. A., Pereira M.V., Cunha S.H., Parker B.J., Praça J.C.: "Interactive Transmission Network Planning Using a Least-Effort Criterion", IEEE Trans. Power App. Syst., Vol. PAS-101, No. 10, October 1982.
- [9] Pereira M.V.F., Pinto L.M.V., Oliveira G.C., S.H.F. Cunha: "Composite Generation-Transmission Expansion Planning", EPRI-Research Project 2473-9, Stanford University, 1987.



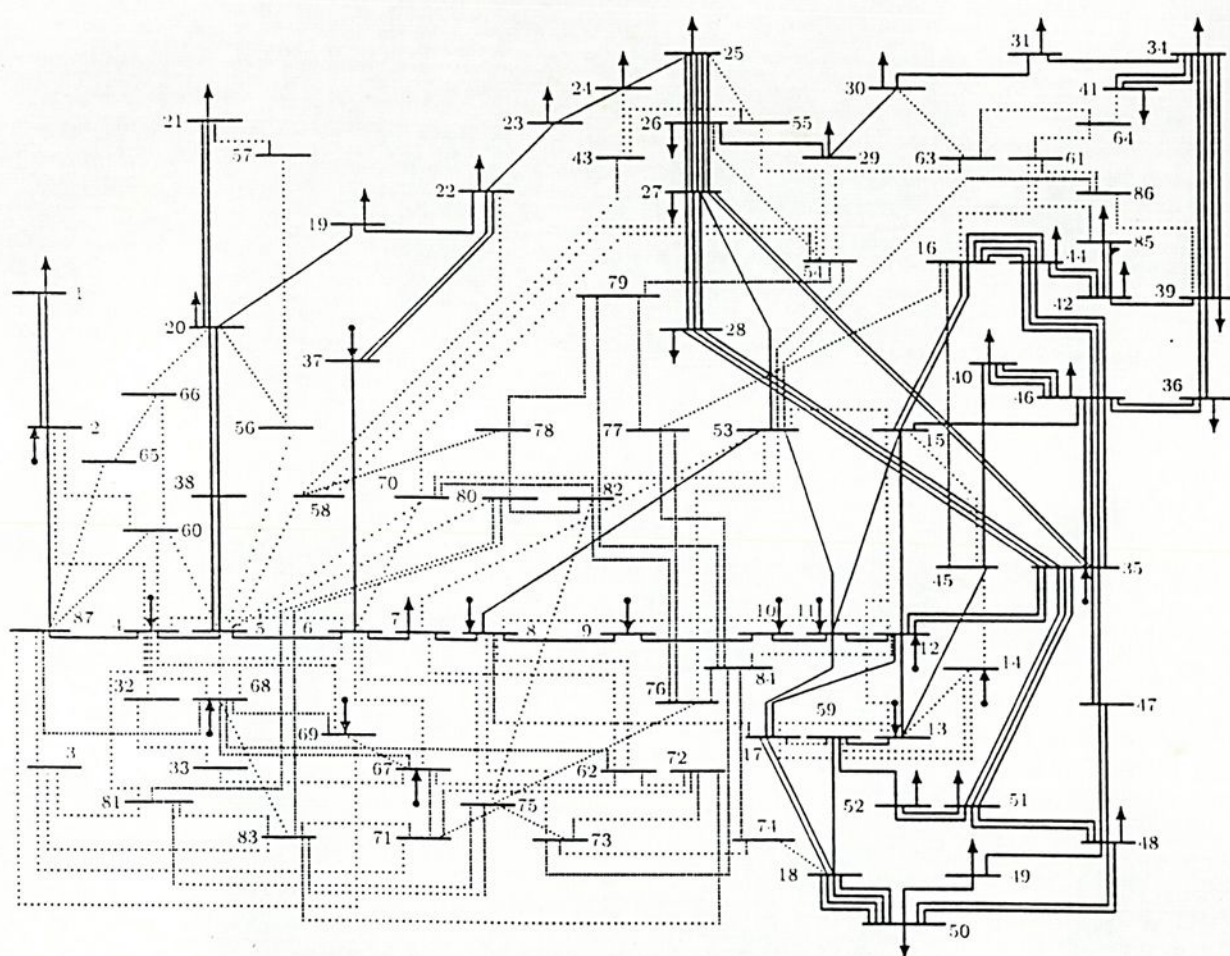


Figure 2: North-Notheastern Network.

**UNIVERSIDADE ESTADUAL PAULISTA - UNESP
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
AV. BRASIL, 56 - CAIXA POSTAL 31
15385-000 - ILHA SOLTEIRA - SP**

