

RAINER ZANGHI

Metodologia com base em Algoritmos Genéticos para Análise de Solicitações de Desligamentos em Redes Elétricas

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Computação Científica e Sistemas de Potência.

Orientadores: Prof. Julio Cesar Stacchini de Souza, D.Sc.
Prof. Milton Brown Do Couto Filho, D.Sc.

Niterói

2011

**Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de
Computação da UFF**

Z29 Zanghi, Rainer

Metodologia com base em algoritmos genéticos para análise de solicitações de desligamentos em redes elétricas / Rainer Zanghi. – Niterói, RJ : [s.n.], 2011.

140 f.

Dissertação (Mestrado em Computação) - Universidade Federal Fluminense, 2011.

Orientadores: Julio Cesar Stacchini de Souza, Milton Brown Do Couto Filho.

1. Sistema elétrico de potência. 2. Otimização combinatória.
3. Algoritmo genético. 4. Controle de sistemas de energia elétrica.
I. Título.

CDD 621.31

RAINER ZANGHI

METODOLOGIA COM BASE EM ALGORITMOS GENÉTICOS PARA ANÁLISE DE SOLICITAÇÕES DE DESLIGAMENTOS EM REDES ELÉTRICAS

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Computação Científica e Sistemas de Potência.

Aprovada em dezembro de 2011.

BANCA EXAMINADORA

Prof. Julio Cesar Stacchini de Souza, D.Sc. – Orientador
Universidade Federal Fluminense

Prof. Milton Brown Do Coutto Filho, D.Sc. – Orientador
Universidade Federal Fluminense

Prof. Alexandre Plastino de Carvalho, D.Sc.
Universidade Federal Fluminense

Prof. Vitor Hugo Ferreira, D.Sc.
Universidade Federal Fluminense

Prof. Glauco Nery Taranto, D.Sc.
Universidade Federal do Rio de Janeiro

Niterói
2011

“Este mundo, igual para todos, nenhum dos deuses e nenhum dos homens o fez; sempre foi, é e será um fogo eternamente vivo, acendendo-se e apagando-se conforme a medida.”

“A mais bela harmonia cósmica é semelhante a um monte de coisas atiradas.”

Heráclito de Éfeso
(aprox. 540 a.C. - 470 a.C.)

AGRADECIMENTOS

Agradeço a todos aqueles que me apoiaram nesta jornada de retorno à Academia, em especial minha amada esposa Alice por seu carinho, dedicação e acolhimento.

Aos orientadores Julio Cesar Stacchini de Souza e Milton Brown Do Coutto Filho pelo incentivo e condução durante o curso, que culminaram na elaboração desta Dissertação.

Ao Eng. Marcio Leonardo Roberto pela cooperação essencial para a elaboração deste trabalho.

Aos meus filhos Enzo e Gael pelos momentos de alegria e paz proporcionados pelo nosso convívio em família. Aos meus familiares, em especial às avós de meus filhos, que proporcionaram momentos em que foi possível me dedicar a este trabalho.

Aos meus pais, em especial à minha mãe Therezinha, que se dedicou em proporcionar ensino de qualidade ao seu filho, despertando e incentivando meu prazer em aprender.

RESUMO

Em um sistema de potência interligado sobre o qual atuam diversos agentes responsáveis pela manutenção de equipamentos, cabe à coordenação da operação efetuar a análise e o agendamento de solicitações de desligamentos de elementos da rede elétrica. Em sistemas interligados, tal como no Sistema Interligado Nacional (SIN) brasileiro, o desligamento de uma linha de transmissão, ainda que programado, altera suas condições operativas. Por exemplo, no caso de desligamentos simultâneos, limites operacionais de alguns equipamentos podem ser violados severamente, inviabilizando o atendimento às solicitações recebidas. Além disso, prioridades devem ser atribuídas aos pedidos de desligamentos, de modo a considerar os diversos níveis de urgência associados a cada tipo de intervenção no sistema.

Assim sendo, para o responsável pela programação de desligamentos, a busca de uma solução que atenda aos solicitantes, observados diversos requisitos operativos, se constitui em um problema de otimização combinatória complexo, com inúmeras soluções. Tal característica torna atraente a aplicação de meta-heurísticas.

Esta Dissertação introduz uma metodologia eficiente, com base em Algoritmos Genéticos, para a solução do problema de avaliação de solicitações de desligamentos de elementos de redes elétricas interligadas, sujeitas a restrições operativas. Resultados de testes utilizando os sistemas IEEE 14 e 30 barras são apresentados e discutidos.

Palavras-chave: ***Operação de sistemas de energia elétrica;
Programação de desligamentos;
Algoritmos Genéticos.***

ABSTRACT

In an interconnected power system on which various agents are responsible for equipment maintenance, the coordination of the operation must perform the analysis and scheduling requests for the disconnection of grid elements. In interconnected systems, such as the National Interconnected System (SIN) of Brazil, the disconnection of a transmission line, even if scheduled, change its operating conditions. For example, in case of simultaneous outages, operational limits of a given equipment can be severely violated, making it impossible to accomplish the requests received. In addition, priority should be given to requests for disconnections in order to consider the different levels of urgency associated with each type of intervention in the system.

Thus, to the person in charge of scheduling disconnections, the search for a solution that meets the applicants, observing various operating requirements, constitutes a complex combinatorial optimization problem with many solutions. This feature makes it attractive to apply meta-heuristics.

This Dissertation introduces an efficient methodology based on genetic algorithms for solving the problem of evaluating requests for disconnections of interconnected grids elements, subject to operational constraints. Test results using the IEEE 14 bus and 30 bus systems are presented and discussed.

Keywords: ***Power System Operation;***
 Maintenance Schedule;
 Genetic Algorithm.

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1 CONSIDERAÇÕES INICIAIS	1
1.2 OBJETIVOS.....	3
1.3 PUBLICAÇÕES.....	4
1.4 ESTRUTURA DA DISSERTAÇÃO	4
CAPÍTULO 2 – PROGRAMAÇÃO DE DESLIGAMENTOS.....	6
2.1 INTRODUÇÃO.....	6
2.2 CARACTERIZAÇÃO DO PROBLEMA	6
2.3 CRITÉRIOS DE ANÁLISE PARA SOLICITAÇÕES DE INTERVENÇÕES	8
2.4 INEQUAÇÕES PARA MONITORAMENTO DE CARREGAMENTO	9
2.5 TÉCNICAS DE SOLUÇÃO	11
CAPÍTULO 3 – META-HEURÍSTICAS	15
3.1 INTRODUÇÃO.....	15
3.2 META-HEURÍSTICAS.....	17
3.2.1 <i>Intensificação e diversificação</i>	18
3.3 ALGORITMOS GENÉTICOS	19
3.3.1 <i>Histórico</i>	19
3.3.2 <i>Implementação</i>	21
3.3.3 <i>Codificação</i>	24
3.3.4 <i>Operadores e estratégias</i>	30
3.3.5 <i>Parâmetros</i>	46
3.4 GERADORES DE NÚMEROS PSEUDO ALEATÓRIOS	47
3.5 CONCLUSÕES	49
CAPÍTULO 4 – METODOLOGIA PROPOSTA	50
4.1 FORMULAÇÃO DO PROBLEMA	50
4.2 OTIMIZAÇÃO DA PROGRAMAÇÃO DE DESLIGAMENTOS VIA ALGORITMOS GENÉTICOS.....	51
4.2.1 <i>Codificação da programação</i>	52
4.2.2 <i>Condições de carregamento</i>	54
4.2.3 <i>Definição dos parâmetros do Algoritmo Genético</i>	55
4.2.4 <i>Inclusão de indivíduos na população inicial</i>	56
4.2.5 <i>Elitismo associado à reinicialização da população</i>	57
4.2.6 <i>Conjunto elite com um indivíduo</i>	58
4.2.7 <i>Conjunto elite de tamanho automático</i>	58
4.3 MÉTRICA PARA AVALIAÇÃO DA APTIDÃO DAS PROGRAMAÇÕES DE DESLIGAMENTOS.....	59
4.3.1 <i>Requisito de prioridades na reprogramação das intervenções</i>	60
4.3.2 <i>Requisito de atendimento à carga</i>	61
4.3.3 <i>Requisito de atendimento aos limites operativos dos equipamentos</i>	61
4.3.4 <i>Requisito de atendimento às inequações para monitoramento de carregamento</i>	62
4.3.5 <i>Requisito de atendimento a contingências simples</i>	64
4.4 DEFINIÇÃO DO GERADOR DE NÚMEROS PSEUDO-ALEATÓRIOS	65
4.4.1 <i>Algoritmo para determinação das sementes</i>	66
4.5 FLUXOGRAMA DO ALGORITMO PROPOSTO	67
4.6 CONCLUSÕES	68
CAPÍTULO 5 – RESULTADOS	69
5.1 DESCRIÇÃO DAS SIMULAÇÕES	69
5.2 IMPLEMENTAÇÃO DO ALGORITMO PROPOSTO.....	71
5.3 TESTES COMPARATIVOS CONSIDERANDO O SISTEMA IEEE 14 BARRAS	73
5.3.1 <i>Descrição dos testes 6, 7, 8, 12, 13, 14 de [Robe04]</i>	73
5.3.2 <i>Comparação com os testes 6, 7, 8, 12, 13 e 14 de [Robe04]</i>	75
5.3.3 <i>Análise Qualitativa do AG proposto</i>	81
5.4 TESTES CONSIDERANDO O SISTEMA IEEE 30 BARRAS	99

5.4.1	<i>Teste para 5 desligamentos</i>	100
5.4.2	<i>Teste para 6 desligamentos</i>	101
5.4.3	<i>Teste para 7 desligamentos</i>	102
5.4.4	<i>Teste para 8 desligamentos</i>	103
5.4.5	<i>Teste para 9 desligamentos</i>	105
5.4.6	<i>Teste para 10 desligamentos</i>	106
5.4.7	<i>Comentários sobre os resultados com o sistema IEEE 30 barras</i>	108
5.5	CONCLUSÕES	110
CAPÍTULO 6 – CONCLUSÕES E TRABALHOS FUTUROS		111
REFERÊNCIAS		114
APÊNDICE A – SISTEMAS TESTADOS		122
A.1	CONFIGURAÇÃO BÁSICA DO SISTEMA IEEE 14	122
A.2	DADOS DE REDE	123
A.3	DADOS DE CARGA	123
A.4	CONFIGURAÇÃO BÁSICA DO SISTEMA IEEE 30	124
A.5	DADOS DE REDE	125
A.6	DADOS DE CARGA	126
APÊNDICE B – GERADORES DE NÚMEROS PSEUDO-ALEATÓRIOS		127
B.1	TESTES DE RANDOMICIDADE	127
B.2	GERADOR DE FIBONACCI SUBTRATIVO DEFASADO	128
B.3	MERSENNE TWISTER	129
B.4	TRANSFORMADA DE BOX-MULLER	130

ÍNDICE DE FIGURAS

Figura 3.1 – Ótimo local x ótimo global	16
Figura 3.2 – Técnicas de busca	20
Figura 3.3 - Fluxograma básico de um AG.....	22
Figura 3.4 - Codificação do fenótipo	24
Figura 3.5 - Cromossomos, genes e alelos.....	25
Figura 3.6 - Codificação binária para diversas variáveis	26
Figura 3.7 - Problemas no mapeamento fixo da representação binária	27
Figura 3.8 - Estratégia de Cruzamento Multivariável Uniforme 50%	35
Figura 3.9 - Cruzamento binário de 1 ponto.....	36
Figura 3.10 - Distribuição de probabilidade de β para cruzamento binário de 1 ponto	39
Figura 3.11 - Distribuições de probabilidade para diferentes n	40
Figura 3.12 - Pais distantes	42
Figura 3.13 - Pais próximos	42
Figura 3.14 - Distribuição de probabilidade para diversos n na mutação polinomial ...	44
Figura 4.1 - Codificações empregadas no AG.....	53
Figura 4.2 - Exemplo de programação com diferentes cenários	55
Figura 4.3 - Fluxograma principal da metodologia	68
Figura 5.1 - Diversificação para Teste 6 com codificação GRAY	83
Figura 5.2 - Diversificação para Teste 6 com codificação INTEIRA	83
Figura 5.3 - Diversificação para Teste 7 com codificação GRAY	84
Figura 5.4 - Diversificação para Teste 7 com codificação INTEIRA	84
Figura 5.5 - Diversificação para Teste 8 com codificação GRAY	85
Figura 5.6 - Diversificação para Teste 8 com codificação INTEIRA	85
Figura 5.7 - Diversificação para Teste 12 com codificação GRAY	86
Figura 5.8 - Diversificação para Teste 12 com codificação INTEIRA	86
Figura 5.9 - Diversificação para Teste 13 com codificação GRAY	87
Figura 5.10 - Diversificação para Teste 13 com codificação INTEIRA	87
Figura 5.11 - Diversificação para Teste 14 com codificação GRAY	88
Figura 5.12 - Diversificação para Teste 14 com codificação INTEIRA	88
Figura 5.14 - Indivíduos no Conjunto Elite Teste 7 GRAY x INTEIRA	90
Figura 5.15 - Indivíduos no Conjunto Elite Teste 8 GRAY x INTEIRA	90
Figura 5.16 - Indivíduos no Conjunto Elite Teste 12 GRAY x INTEIRA	91
Figura 5.17 - Indivíduos no Conjunto Elite Teste 13 GRAY x INTEIRA	91
Figura 5.18 - Indivíduos no Conjunto Elite Teste 14 GRAY x INTEIRA	92
Figura 5.19 - Gráfico tttplots para o Teste 6 - Conjunto Elite.....	96
Figura 5.20 - Gráfico tttplots para o Teste 6 – Codificação	96
Figura 5.21 - Gráfico tttplots para o Teste 7 - Conjunto Elite.....	97
Figura 5.22 - Gráfico tttplots para o Teste 7 – Codificação	97
Figura 5.23 - Gráfico tttplots para o Teste 8 - Conjunto Elite.....	98
Figura 5.24 - Gráfico tttplots para o Teste 8 – Codificação	98
Figura A.1 - Sistema IEEE 14 barras [Chri99].....	122
Figura A. 2 - Sistema IEEE 30 barras [Chri99].....	124

ÍNDICE DE TABELAS

Tabela 2.1 - Evolução dos eventos para análise de sobrecarga	10
Tabela 3.1 - Hamming Cliffs na codificação binária	28
Tabela 3.2 - Comparação entre distância de Hamming de binário e Gray	29
Tabela 5.1 - Descrição dos testes 6, 7, 8, 12, 13 e 14 de [Robe04]	74
Tabela 5.2 - Programações utilizadas nos testes comparativos IEEE 14 barras	74
Tabela 5.3 - Lista de contingências dos testes com o sistema IEEE 14 barras	75
Tabela 5.4 - Comparação de resultados para o Teste 6	76
Tabela 5.5 - Comparação de resultados para o Teste 7	76
Tabela 5.6 - Comparação de resultados para o Teste 8	77
Tabela 5.7 - Comparação de resultados para o Teste 12	78
Tabela 5.8 - Comparação de resultados para Teste 13	78
Tabela 5.9 - Comparação de resultados para Teste 14	79
Tabela 5.10 - Tempos computacionais dos testes com o sistema IEEE 14 barras.....	80
Tabela 5.11 - Gerações necessárias até a solução - testes IEEE 14 barras	80
Tabela 5.12 - Comparação das diferentes estratégias com diferentes GNPA's	93
Tabela 5.13 - Resultados comparativos para diferentes estratégias	94
Tabela 5.14 - Lista de contingências dos testes com o sistema IEEE 30 barras	99
Tabela 5.15 - Programação de 5 desligamentos – IEEE 30 Barras.....	100
Tabela 5.16 - Programação Original x Otimizada - 5 desligamentos.....	100
Tabela 5.17 - Resumo das violações (em %) para 5 desligamentos	100
Tabela 5.18 - Programação de 6 desligamentos – IEEE 30 Barras.....	101
Tabela 5.19 - Programação Original x Otimizada - 6 desligamentos.....	101
Tabela 5.20 - Resumo das violações (em %) para 6 desligamentos	102
Tabela 5.21 - Programação de 7 desligamentos – IEEE 30 Barras.....	102
Tabela 5.22 - Programação Original x Otimizada - 7 desligamentos.....	103
Tabela 5.23 - Resumo das violações (em %) para 7 desligamentos	103
Tabela 5.24 - Programação de 8 desligamentos – IEEE 30 Barras.....	104
Tabela 5.25 - Programação Original x Otimizada - 8 desligamentos.....	104
Tabela 5.26 - Resumo das violações (em %) para 8 desligamentos	105
Tabela 5.27 - Programação de 9 desligamentos – IEEE 30 Barras.....	105
Tabela 5.28 - Programação Original x Otimizada - 9 desligamentos.....	106
Tabela 5.29 - Resumo das violações (em %) para 9 desligamentos	106
Tabela 5.30 - Programação de 10 desligamentos – IEEE 30 Barras.....	107
Tabela 5.31 - Programação Original x Otimizada - 10 desligamentos.....	107
Tabela 5.32 - Resumo das violações (em %) para 10 desligamentos	108
Tabela 5.33 - Número de gerações até a convergência – Testes IEEE 30 Barras....	108
Tabela 5.34 - Tempo de execução das 700 gerações – Testes IEEE 30 Barras.....	109
Tabela A. 1 - Dados de rede para o sistema IEEE 14 barras.....	123
Tabela A. 2 - Dados de carga para o sistema IEEE 14 barras	123
Tabela A. 3 - Dados de rede para o sistema IEEE 30 barras.....	125
Tabela A. 4 - Dados de carga para o sistema IEEE 30 barras	126

1.1 CONSIDERAÇÕES INICIAIS

A tarefa de programar desligamentos de equipamentos em um sistema interligado de transmissão de energia elétrica diz respeito ao operador do sistema. Esta tarefa deve atender a requisitos técnicos, econômicos e de segurança. Em um sistema em que diversas empresas operam de forma interligada, se faz necessária a ação de um operador que gerencie pedidos de desligamentos através de critérios amplamente aceitos, descritos em procedimentos estabelecidos de comum acordo com tais empresas.

A coordenação e controle da operação das instalações de geração/transmissão de energia elétrica do Sistema Interligado Nacional (SIN) cabem ao Operador Nacional do Sistema (ONS) sendo este constituído por diversos membros associados (agentes) e participantes (Ministério das Minas e Energia, Conselhos de Consumidores, entre outros). O SIN compõe-se de empresas de diversas regiões do País: Sul, Sudeste, Centro-Oeste, Nordeste e parte da região Norte. Pequena capacidade de produção de energia (cerca de 3,4%) encontra-se fora do SIN, em sistemas localizados principalmente na região amazônica.

No Brasil, a Rede Básica ou SIN é operada pelo ONS que leva a efeito procedimentos homologados pela ANEEL (Agência Nacional de Energia Elétrica). Estes procedimentos de rede devem ser atendidos pelos diversos agentes que possuem equipamentos nesta Rede de Geração e Transmissão de Energia Elétrica.

Sob o ponto de vista dos agentes, os desligamentos de equipamentos são necessários para efetuar procedimentos internos de manutenção preventiva, manutenção corretiva e expansão (para inclusão de novos

equipamentos). Cabe aos agentes informar o motivo do desligamento e o período necessário até a normalização operacional do equipamento em questão.

As intervenções requisitadas pelos agentes para efetuar desligamentos de equipamentos podem afetar significativamente o sistema; até expô-lo à situação de indisponibilidade, caso um evento fortuito propicie o desligamento de um ou mais equipamentos, além do que tenha sido previsto.

Atualmente, devido à ausência de ferramentas que tratem o problema de desligamentos de elementos da rede elétrica de forma integrada, as estratégias usualmente empregadas são baseadas em análises simplificadas e na experiência de operadores. Em geral, daí resulta um deslocamento das solicitações de desligamentos para períodos onde se acredita que seu impacto seja minimizado. Por exemplo, intervenções na rede básica do SIN que levem a violações de quaisquer dos requisitos descritos nos Procedimentos de Rede do ONS [ONS09] são programadas de modo a incluir o maior número possível de horas nos períodos de carga leve e mínima, finais de semana ou feriados, de forma a minimizar a perda de carga e o tempo de exposição do sistema ao risco de uma operação inadequada. Tal estratégia, além de não considerar todas as possíveis soluções que possam garantir o bom funcionamento do sistema, não busca atender a solicitação inicial de desligamentos.

A solução do problema deve ser obtida de modo a minimizar as possíveis violações de restrições, considerando-as ponderadamente, de modo a mensurar a adequação do resultado final aos requisitos desejados.

Conforme [Robe04], o modelo adotado é fortemente influenciado pelas ponderações associadas às restrições, as quais propiciam também uma maior flexibilidade ao operador para, ao formular o problema, incluir aspectos qualitativos que levem também em consideração a experiência operativa.

Resultados relatados na literatura [Robe04; Robe05; Souza11] mostram que o uso de ferramentas de otimização combinatória, especialmente

aquelas classificadas dentre os algoritmos evolucionários, já demonstrou potencial para a solução deste problema. No entanto, é necessário ainda explorar o benefício de tal aplicação, com a identificação de meta-heurísticas que apresentem melhor desempenho e também a inclusão de outras restrições de interesse para o problema.

Esta Dissertação apresenta uma metodologia que permite a realização da programação de desligamentos em redes elétricas considerando-se simultaneamente e de forma integrada a análise de desempenho e observação de diferentes restrições associadas à operação de sistemas de potência. A metodologia apresentada em [Robe04] é estendida, de forma a investigar a aplicação de uma meta-heurística mais poderosa, capaz de levar a soluções de melhor qualidade em tempos computacionais menores. Testes são realizados com os sistemas IEEE 14 e 30 barras de forma a avaliar o desempenho da metodologia proposta. Quando pertinente, comparações com resultados obtidos em [Robe04] são também apresentados.

1.2 OBJETIVOS

Esta Dissertação tem como objetivo propor uma metodologia para solução do problema do agendamento de desligamentos em redes de energia elétrica utilizando um Algoritmo Genético. Neste trabalho pretende-se:

- apresentar uma metodologia integrada para a solução do problema de programação de desligamentos, o qual é tratado como um problema de otimização combinatória onde diferentes restrições operativas são simultaneamente consideradas;
- introduzir alterações no algoritmo genético empregado em [Robe04] de modo a melhorar a eficiência e eficácia do processo de busca;
- investigar diferentes codificações para representar as soluções do problema, identificando a mais conveniente;

- incluir restrições operativas associadas à monitoração de carregamento nos ramos que fazem parte de certos arranjos topológicos de interesse.

1.3 PUBLICAÇÕES

As publicações listadas a seguir possuem relação direta com esta Dissertação:

R. Zanghi, J. C. Stacchini de Souza; M. B. Do Coutto Filho, “Manutenção Otimizada de Equipamentos de Transmissão de Energia Elétrica Através de Algoritmos Genéticos”, XLIII Simpósio Brasileiro de Pesquisa Operacional, pp. 1-12, Ubatuba, SP, 2011.

R. Zanghi, J. C. Stacchini de Souza; M. B. Do Coutto Filho, “Aplicação de Algoritmos Genéticos para a Programação Ótima de Desligamentos em Redes de Energia Elétrica”, XI Simpósio Brasileiro de Automação Inteligente, pp. 1-6, São João Del Rei, MG, 2011.

1.4 ESTRUTURA DA DISSERTAÇÃO

Esta Dissertação está estruturada da seguinte forma:

No Capítulo 2, descreve-se o problema a ser analisado por este trabalho.

O Capítulo 3 apresenta as bases da metodologia desenvolvida, fornecendo uma breve abordagem das heurísticas e estratégias a serem implementadas.

O Capítulo 4 constrói uma metodologia para o problema em estudo, compreendendo: formulação do problema; escolhas de codificação; operadores e parâmetros do Algoritmo Genético adotado; e demais heurísticas.

No Capítulo 5 são reunidos os resultados dos estudos de simulação realizados com os sistemas IEEE 14 barras e 30 barras, bem como análises comparativas com resultados de trabalhos anteriores.

O Capítulo 6 apresenta conclusões do trabalho e também propostas para trabalhos futuros.

PROGRAMAÇÃO DE DESLIGAMENTOS

2.1 INTRODUÇÃO

Neste capítulo será apresentada uma descrição do problema de avaliação de solicitações de desligamentos de elementos de redes elétricas. Tal problema situa-se entre os de natureza combinatória, nos quais busca-se minimizar uma função que represente a reprogramação de desligamentos solicitados por empresas que operem em redes elétricas interligadas, sujeitos a restrições de regime quase permanente.

Por fim, na mesma linha de pesquisa encontrada na literatura técnica para o tratamento do problema de planejamento de manutenções, adotam-se meta-heurísticas para tratar problemas desta natureza.

2.2 CARACTERIZAÇÃO DO PROBLEMA

A programação de desligamentos de elementos de redes elétricas que seja capaz de atender a diversas restrições operativas (e.g., horários próprios à manutenção, segurança operativa, contingências) é tarefa do operador do sistema. No Brasil, conforme descrito em [Robe04], essa avaliação de desligamentos programados é realizada na Gerência de Programação e Desligamentos (GPD), pertencente à Diretoria de Planejamento e Programação da Operação (DPP) do ONS, seguindo os critérios contidos nos submódulos dos procedimentos de rede homologados pela ANEEL.

Atualmente, esta tarefa é realizada através de aplicativos que calculam o fluxo de potência de todo o sistema para cada cenário em que estejam incluídos os desligamentos dos equipamentos de uma programação específica. Cabe ao analista simular e verificar individual e manualmente cada cenário de desligamentos simultâneos, analisando as ocorrências de violações de cada elemento do sistema, propondo novas programações que

por sua vez deverão ter seus cenários de desligamentos simultâneos testados, em um ciclo contínuo, até que uma programação ótima sem violações seja encontrada. Usualmente, devido à complexidade e custo associado à implementação desta tarefa, as programações são estabelecidas em horários de baixa demanda energética e sem simultaneidade, imputando aos agentes uma adequação dos horários pretendidos à nova programação estabelecida. Nos casos em que a simultaneidade é mandatória, o recurso de redespacho de potência é utilizado para minimizar e eliminar as violações dos limites operacionais dos equipamentos. Conforme estabelecido em [ONS09] e ressaltado em [Robe04], intervenções que impliquem em restrições de despacho de usinas em relação à programação sem restrições elétricas deverão ser programadas para dias e horários em que seja minimizado o afastamento em relação às metas energéticas. Caso, na solicitação original, essa condição não seja atendida, cabe ao ONS propor ao Agente períodos alternativos para a realização do serviço.

A presença de equipamentos de disponibilidade crítica ou mais sensíveis a alterações das condições operacionais de funcionamento do sistema elétrico interligado suscita verificações de estados de contingência, onde o sistema é testado no caso do desligamento fortuito destes equipamentos em cada cenário de desligamentos programados. Desta forma, o analista pode prever o impacto operacional destas atuações imprevistas, caso ocorram dentro do período programado para cada desligamento.

Também é de interesse para o analista, em determinados arranjos topológicos, a verificação do efeito de sobrecarga que o desligamento de cada equipamento provoca nos adjacentes. Caso o fluxo de potência ativa em determinado equipamento, mesmo atendendo aos seus limites operativos, seja tal que gere sobrecarga em um ou mais equipamentos adjacentes no caso de seu desligamento fortuito, a análise dos fatores de superposição para cada cenário de desligamento indicará esta situação. A adição desta verificação a cada cenário de programação é uma medida

saneadora que indicará que a programação escolhida colocará o sistema em um estado de alerta, conforme indicado em [ONS04].

O operador também deverá verificar a inviabilidade de determinadas programações tendo como base os valores de fluxo de potência no sistema e considerando as capacidades de geração e transmissão de energia dos equipamentos interligados. Estas programações inviáveis devem ser descartadas e os desligamentos simultâneos que ocasionam estes cenários programados em horários distintos, considerando a prioridade de cada desligamento.

A construção de uma ferramenta computacional, parametrizável, que permita efetuar a análise das solicitações de desligamentos, em tempo computacional reduzido e de forma automática, torna-se de grande valia. A solução deste problema de otimização combinatória já foi abordada em [Robe04], sendo necessário refinar e ampliar a metodologia lá encontrada para a consideração de novas restrições.

2.3 CRITÉRIOS DE ANÁLISE PARA SOLICITAÇÕES DE INTERVENÇÕES

Uma solicitação de intervenção na rede elétrica será concedida quando atender certos requisitos, dentre os quais são considerados neste trabalho [ONS09]:

- a) Não interrupção de fornecimento às cargas, em regime normal de operação.
- b) O sistema deve suportar contingência simples, sem perda de carga.
- c) Não devem existir violações de limites operativos (definidos para regime normal) em qualquer elemento da rede em estudo.

Quando diferentes pedidos de intervenção se sobrepuserem (ocorrência em um mesmo período de tempo), não sendo compatíveis entre si, a prioridade será dada de acordo com os critérios pré-estabelecidos [ONS09].

Intervenções de urgência, que são aquelas em que se caracteriza risco aos equipamentos e/ou pessoas, são realizadas em qualquer período de carga, devendo-se buscar realizá-las no período mais favorável para o sistema. As intervenções são consideradas não compatíveis, quando a análise indicar que sua realização simultânea, em qualquer período de carga, leva à violação de qualquer dos critérios anteriormente descritos. Tal violação pode ser eliminada ou minimizada, se as intervenções forem não coincidentes.

Na análise para obtenção da programação mais favorável, alguns dados são necessários:

- Dados fornecidos pelos agentes que requisitam as intervenções (data inicial, duração desejada, linha a ser desligada e prioridade).
- Topologia do sistema.
- Perfis de carga de todas as barras do sistema.

Vale ressaltar que, assim como em [Robe04], este trabalho não considera a análise de segurança (estudos dinâmicos) e de natureza transitória (fenômenos eletromagnéticos) do sistema elétrico para o agendamento dos desligamentos, o que será deixado para futuras pesquisas.

2.4 INEQUAÇÕES PARA MONITORAMENTO DE CARREGAMENTO

Em determinados arranjos topológicos, como linhas de transmissão em anel, transformadores de duas ou mais derivações e linhas em paralelo a verificação do efeito de sobrecarga que o desligamento de cada equipamento provoca nos adjacentes deve ser cuidadosamente levada a efeito pelo analista de operação.

A Norma Técnica 136/2004 do ONS [ONS04] indica uma metodologia para cálculo dos fatores de superposição e verificação das inequações para monitoramento de carregamento. Através destas inequações, é possível verificar os casos onde o fluxo de potência ativa em determinado

equipamento, mesmo atendendo aos seus limites operativos, seja tal que gere sobrecarga em um ou mais equipamentos próximos, no caso de seu desligamento fortuito.

Conforme [ONS04], a formulação mais usual para obtenção destes fatores prevê a utilização do fluxo de potência linearizado, em que os fatores de distribuição de carregamento por saída de equipamentos ou fatores de superposição dependerão apenas da matriz de susceptância de barra (B') que representa as características construtivas e topológicas da rede, mantendo estes fatores constantes e independentes do ponto de operação.

Nesta formulação linearizada, considerando que um determinado equipamento indicado pelo n° 2 seja desligado e deseja-se avaliar a sobrecarga sobre o outro de n° 1, vem a seguinte evolução de eventos descritos na Tabela 2.1 [ONS04].

Tabela 2.1 - Evolução dos eventos para análise de sobrecarga

Equipamento	Fluxo do Equipamento no Momento Inicial (t_i)	Fluxo do Equipamento no Momento Final (t_f)
1	F_{1i}	F_{1f}
2	F_{2i}	0

Para evitar que o carregamento final supere o limite do equipamento:

$$F_{1f} < Limite \quad (2.1)$$

para

$$F_{1f} = F_{1i} + k_{21} \times F_{2i} \quad (2.2)$$

Onde cada fluxo de potência ativa corresponde a:

F_{1i} equipamento 1 no momento inicial

F_{1f} equipamento 1 no momento final

F_{2i} equipamento 2 no momento inicial

k_{21} fator de superposição do equipamento 2 sobre o equipamento 1

Limite é o valor máximo para o fluxo de potência ativa no equipamento 1

Este fator de superposição representa (percentualmente) quanto do carregamento do equipamento 2 antes da ocorrência do evento será adicionado ao carregamento do equipamento 1, caso o equipamento 2 seja desligado. Tal fator pode ser calculado pela equação (2.3).

$$k_{21} = \frac{F_{1f} - F_{1i}}{F_{2i}} \quad (2.3)$$

2.5 TÉCNICAS DE SOLUÇÃO

Diversas técnicas têm sido propostas na literatura para a programação de manutenção de equipamentos, em geral associadas à manutenção de unidades geradoras e baseadas em técnicas de otimização, tais como: programação dinâmica [Zurn76], programação inteira [Egan76; Dopa89; Muke91; Chen91; Chat95], decomposição de Benders [Yell92; Al-K92; Silv95], métodos heurísticos [Garv72; Stre81; El-S84; Cont89; Juan97], etc.

A avaliação da influência de desligamentos na confiabilidade do sistema de Ontário é analisada em [And02]. Em [Fu09], um problema de otimização é resolvido para a coordenação de planejamento de médio prazo de desligamentos considerando compromisso de curto prazo de unidades com restrições de segurança. Sistemas inteligentes têm sido aplicados ao problema, em especial, sistemas especialistas [Lin92], nos quais a experiência e conhecimento humanos são explorados para a construção de um sistema baseado em regras que forma a máquina de inferência responsável pela manutenção do programa de agendamentos. No entanto, sistemas especialistas têm desempenho satisfatório somente em situações que já foram consideradas durante o desenvolvimento de sua base de conhecimento, não tendo a capacidade de lidar com novos dados. Aplicações de meta-heurísticas como algoritmos genéticos [Wang00],

enxame de partículas [Yare08; Yare10], e busca tabu [Sawa99; Kim97] também podem ser encontradas.

Em [Robe04], [Robe05] e [Souz11] o uso de algoritmos genéticos (AGs) se mostrou potencialmente interessante para a programação de desligamentos de equipamentos de transmissão. Porém, um AG muito simples foi empregado, sendo necessário maior investimento em seu aperfeiçoamento e a avaliação do efeito sobre a eficiência e eficácia da busca pela solução ótima. Além disso, é também de interesse a consideração de outras restrições importantes para o problema, conforme preconizado em [ONS09].

Os AGs são inspirados no paradigma epistemológico da seleção natural trazido por Charles Darwin e Alfred Russel Wallace. Na seleção natural, características hereditárias que contribuem para a aptidão do indivíduo ao meio se tornam mais comuns em uma população, pois estes indivíduos tem maior probabilidade de reproduzir e transferir seu material genético aos seus descendentes.

Os mecanismos da seleção natural, aplicados a algoritmos, permitem resolver problemas de otimização combinatória, em que as soluções ótimas desejadas correspondem aos indivíduos mais aptos. O material genético destes indivíduos é a codificação dos parâmetros que devem ser otimizados. Estes indivíduos são adaptados no processo evolutivo através de operadores genéticos, assim como ocorre na seleção natural. Os objetivos do problema e as restrições do meio, impostas aos indivíduos no mecanismo da seleção natural, são representadas em uma função de aptidão no AG.

A capacidade exploratória de um algoritmo de otimização é medida na diversificação da procura pela maior parte possível do espaço de soluções ou de busca. Para evitar convergência prematura ou intensificação na busca próxima a uma solução ótima local, o algoritmo deve manter sua capacidade de diversificação, conforme citado em [Coel00] e [Blum03]. Os AGs mantêm

em seu processo evolutivo uma população de indivíduos que representam possíveis soluções, possibilitando uma procura robusta do espaço de busca.

Dando continuidade à pesquisa realizada em [Robe04] que utilizou uma codificação binária simples, na presente Dissertação foram exploradas as codificações inteira e Gray. A codificação inteira é adequada quando a característica a ser otimizada pode ser representada por números inteiros. Na codificação Gray, as características são representadas por números binários no código Gray. Neste código, também chamado de código binário refletido, somente um bit muda de um número para o próximo na sequência. Para esta codificação, alterações de apenas um bit podem representar grandes alterações no valor final, permitindo uma maior exploração do espaço de soluções, conforme indicado em [Rowe04].

Os tipos de operadores de cruzamento empregados para AGs dependem da codificação implementada. A utilização do operador SBX [Deb94] se demonstrou robusta suficiente para ser utilizada no cruzamento de cromossomos de indivíduos com codificação inteira ou real. O operador binário de um ponto é usualmente utilizado para cruzamento de indivíduos com codificação em números binários.

O operador de mutação altera de maneira aleatória parte do material genético do indivíduo e permite que parte das características desejáveis perdidas durante o processo evolutivo seja recuperada.

Algumas estratégias ou heurísticas foram empregadas em AGs [Mich96; Coel00; Blum03; Haup04; Siva08]. O elitismo é uma estratégia de seleção de um ou mais indivíduos, garantindo sua permanência em gerações futuras. A estratégia de inclusão de certos indivíduos em uma população inicial bem diversificada também permite um melhor desempenho do AG.

Já foi explorada na literatura existente [Mich96; Haup04; Siva08] a utilização de AGs, com populações iniciais distintas, executados em paralelo

e com possibilidade de migração de determinados indivíduos no final ou durante o processo evolutivo.

Os métodos de seleção escolhidos para as operações de cruzamento e mutação devem considerar aleatoriedade e aptidão. O método do torneio consiste na escolha aleatória de uma quantidade parametrizável de indivíduos em que somente o par mais apto é selecionado para cruzamento. O cruzamento e a mutação são processos aleatórios que ocorrem com probabilidade previamente especificada em parâmetros do AG. O processo de sintonia destes parâmetros é amplamente discutido na literatura [Mich96; Coel00; Haup04; Siva08], sendo usual a aplicação de probabilidades de cruzamento altas e de mutação bem reduzidas.

Todos os cálculos que envolvem probabilidade em um AG devem ser feitos a partir de um gerador de números pseudoaleatórios (GNPA) que possa garantir a randomicidade dos números e a reprodutibilidade dos valores gerados com uma mesma semente. O uso de séries com período longo e randomicidade estatisticamente testada [Knut97] associadas a transformadas que permitem adequar os valores a uma distribuição normal [Box58] se demonstrou eficiente para a geração de números pseudoaleatórios em algoritmos que utilizam cálculos probabilísticos.

3.1 INTRODUÇÃO

A otimização em conjuntos finitos é estudada pelo ramo da Ciência da Computação denominado Otimização Combinatória. Alguns problemas de otimização, devido à sua complexidade, apresentam dificuldades intratáveis em sua formulação ou resolução exatas. A aplicação de métodos não exatos para a solução destes problemas se demonstrou adequada e, em muitos casos, a única implementação possível considerando os recursos disponíveis, sejam eles computacionais ou não.

Nos métodos não exatos para solução de problemas de otimização com restrições, é feito um processo de busca de uma solução ótima que atenda às restrições impostas às variáveis analisadas, através da avaliação de uma função objetivo que considera as violações destas restrições.

Em [Papa82], um modelo de otimização com restrições pode ser formalizado como:

Encontre x para

minimizar ou *maximizar* função objetivo $f(x)$

sujeito a restrições principais $g_i(x) \geq 0 \quad i=1, \dots, m$

$h_j(x) = 0 \quad j=1, \dots, p$

onde f , g_i e h_j são funções gerais do parâmetro $x \in R^n$.

Se o valor mínimo ou máximo, conforme a métrica empregada, possível para a função objetivo é atingido, atendendo a todas as restrições impostas, a solução encontrada é considerada ótima global. Para casos onde a metodologia não consegue encontrar valores melhores na vizinhança da solução encontrada, define-se esta solução como um ótimo local. Na Figura

3.1 está ilustrado um exemplo de função objetivo com ótimo local L e ótimo global G para um caso de minimização.

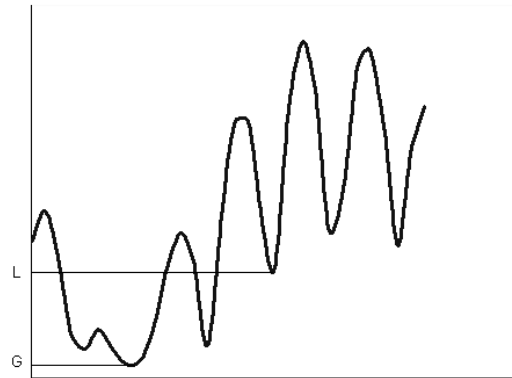


Figura 3.1 – Ótimo local x ótimo global

Deve-se notar que se o valor mínimo ou máximo possível para a função objetivo, dadas as restrições impostas, não é sabido a priori, a solução ótima global pode ser considerada utópica, fazendo com que todas as soluções encontradas sejam consideradas ótimos locais.

Ainda em [Papa82], vemos que a vizinhança pode ser formalizada como um subconjunto $N(s) \subseteq S$ onde S é o espaço de busca do problema e s uma solução do problema. Este subconjunto é composto por todas as soluções próximas à solução s a partir de uma métrica com base na sua representação. A correlação entre a vizinhança e o esquema de codificação utilizado para representar o problema é ressaltado em [Rowe04].

Na formalização encontrada em [Papa82], o ótimo local, para problemas de minimização, pode ser definido como :

$$f(s) \leq f(s') \quad \forall s' \in N(s) \quad (3.1)$$

O ótimo global, para o mesmo tipo de problema, pode ser formulado como:

$$f(s) \leq f(s') \quad \forall s' \in S \quad (3.2)$$

O processo de busca das possíveis soluções de um problema necessita de uma metodologia onde serão definidos todos os elementos e as ações necessárias para se alcançar a solução desejada.

Neste momento do presente trabalho, se faz necessário explicar, mesmo que sucintamente, estas definições como são entendidas hoje pelos campos de conhecimento científico correspondentes. Isto é feito nos itens a seguir.

3.2 META-HEURÍSTICAS

Conforme citado em [Blum03], “o termo meta-heurística primeiramente introduzido por Glover (1986) deriva da composição de duas palavras gregas. Heurística deriva do verbo *heuriskein* (εὕρισκειν) que significa “encontrar”, enquanto o sufixo meta significa “além, em um nível superior.” As meta-heurísticas podem ser definidas como metodologias que utilizam métodos de busca para explorar de forma mais eficiente o espaço de busca e encontrar uma solução desejável para o problema apresentado.

Entende-se como espaço de busca o domínio de todas as soluções do problema, sejam elas adequadas ou não ao que se deseja como resultado ou resposta final da meta-heurística. Para o problema de otimização com restrições, consideramos soluções no espaço de busca estando elas adequadas ou não às restrições impostas à função objetivo.

A forma atualmente mais adequada à aplicação destas meta-heurísticas no campo da Ciência da Computação é a algorítmica. Nesta, as ações são codificadas em linguagem algorítmica e os elementos de entrada e saída codificados de forma a permitir a aplicação da metodologia a um problema específico. Esta forma favorece a sua implementação em uma linguagem de programação, mantendo a clareza do entendimento da metodologia empregada para a busca.

Diversos algoritmos classificados como meta-heurísticas são propostos, discutidos e aplicados na literatura para resolução de problemas de Otimização Combinatória. Dentre eles podemos citar: Busca Tabu, Recozimento Simulado, Otimização de Colônia de Formigas, Otimização por Enxame de Partículas e Algoritmos Genéticos.

Conforme indicado em [Haupt04] a interação entre as variáveis de um problema de otimização é definida como epistasia (do Grego *epi*, sobre, *stasis*, parada, inibição) na literatura relacionada a Algoritmos Genéticos. [Haupt04] comenta que para problemas com epistasia ou interação gênica média para alta, os Algoritmos Genéticos apresentam melhor desempenho.

3.2.1 Intensificação e diversificação

Os operadores e estratégias de cada meta-heurística permitem percorrer o espaço de busca à procura da solução mais adequada à resolução do problema. Duas características deste processo de busca são amplamente discutidas na literatura [Blum03]: diversificação e intensificação. A diversificação se refere à exploração ampla do espaço de busca e a intensificação está relacionada com a busca intensa na vizinhança de uma determinada solução, priorizando a experiência acumulada na história do processo de busca.

Conforme citado em [Blum03], “o equilíbrio entre diversificação e intensificação (...) é importante, de um lado para identificar rapidamente regiões no espaço de busca com soluções de alta qualidade e de outro lado para não desperdiçar muito tempo em regiões do espaço de busca que já foram exploradas ou não proveem soluções de alta qualidade”. É desejável um equilíbrio dinâmico entre estas duas características ao longo de todo o processo de busca, de modo a evitar uma convergência prematura para uma solução sub-ótima.

3.3 ALGORITMOS GENÉTICOS

3.3.1 Histórico

A implementação algorítmica dos processos da seleção natural descritos por Charles Darwin em seu livro “On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life” de 1859 foi formulada inicialmente por matemáticos como Alex Fraser no final da década de 1950 e Nils Aall Barricelli no início da década de 1960.

Dois conceitos básicos estão presentes na seleção natural proposta por Charles Darwin: evolução e adaptação. A evolução é a mudança de características hereditárias de uma população de indivíduos de uma geração para outra através dos mecanismos da seleção natural. A adaptação é a característica ou comportamento adquirido que permite que aquele indivíduo sobreviva no ambiente em que vive. Uma maior adaptação confere maior aptidão ao indivíduo, isto é, permite que ele tenha mais chances de reproduzir seu material genético.

A conciliação entre a evolução por seleção natural e a hereditariedade mendeliana com a utilização de uma modelagem genética para descrever a fonte das características hereditárias ocorre apenas em 1930 com a síntese evolutiva moderna (Ronald Fisher e outros). Esta síntese é apoiada por trabalhos realizados nas décadas de 1940 e 1950 na identificação do DNA como material genético (Oswald Avery e colegas) e na fundamentação física da hereditariedade (James Watson e Francis Creek).

Os Algoritmos Genéticos (AG) foram propostos inicialmente por John Holland e publicados em seu livro “Adaptation in Natural and Artificial Systems” de 1975. Esta metodologia surgiu após a introdução dos conceitos de Estratégias de Evolução publicados na tese “Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution” de Ingo Rechenberg em 1971. A tese “Numerische Optimierung von

Computer-Modellen” de Hans-Paul Schwefel em 1974 também contribuiu na genealogia desta produção científica de Holland. No entanto, conforme apontado por [Mitt99], em contraste com Schwefel e Rechenberg, Holland buscava estudar o fenômeno da adaptação e como ele ocorre na natureza, e desenvolver meios para que estes mecanismos da seleção natural fossem importados para sistemas computacionais. Em seu livro, Holland apresenta os AGs como uma abstração da evolução biológica e fornece uma base teórica para a adaptação no AG.

Com o desenvolvimento de diversos ramos de pesquisa na grande área denominada Computação Evolutiva, os AGs representam hoje uma subdivisão deste campo de estudo, conforme pode ser visto na Figura 3.2 [Siva08].

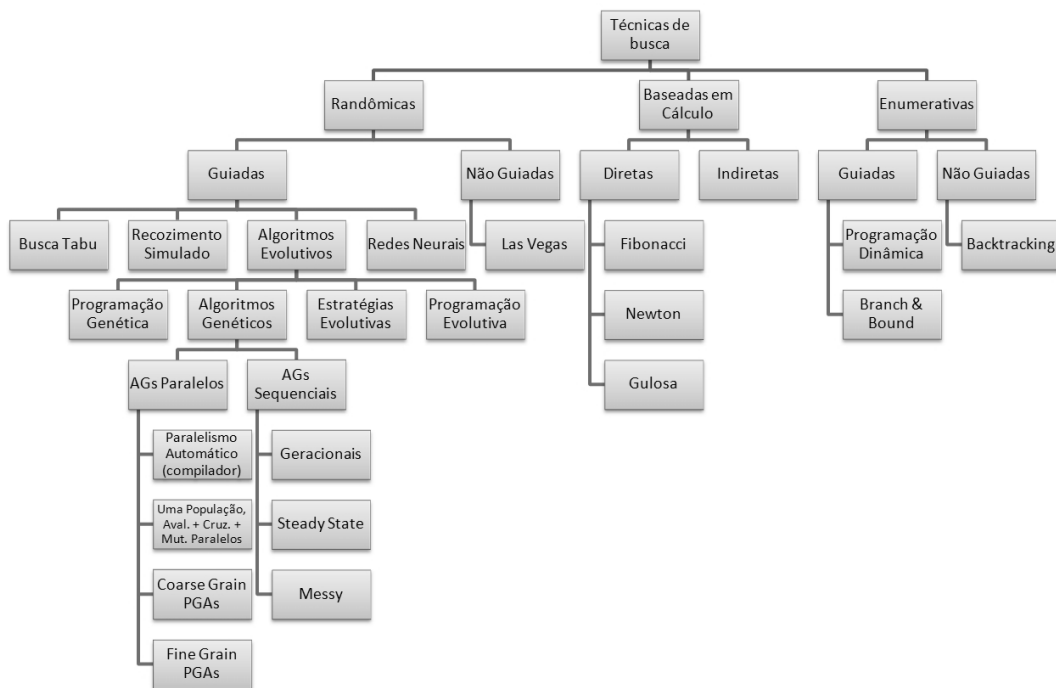


Figura 3.2 – Técnicas de busca

Sob o ponto de vista epistemológico, a concepção positivista da natureza introduzida por Darwin encontra ressonância com os métodos não exatos com base probabilística utilizados para resolução de problemas de otimização combinatória. A migração dos conceitos biológicos introduzidos com a Teoria da Evolução de Darwin das ciências naturais para as ciências

exatas se mostra como mais uma faceta do crescente movimento de interdisciplinaridade que permeia, até hoje, a produção científica.

3.3.2 Implementação

Conforme estipulado em [Gref86] e em [De J07] os AGs são procedimentos iterativos com base probabilística conduzidos por parâmetros e estratégias que devem ser ajustados para sua implementação em cada aplicação.

Em [De J07] são ressaltadas distinções sobre AGs que possuem parâmetros adaptativos, i.e. alterados automaticamente durante o processo iterativo, e outros que se utilizam de parametrização manual, realizada a priori e mantida fixa durante toda a execução do algoritmo. [De J07] ressalta que o uso de parâmetros adaptativos para implementação de AGs sem parâmetros ajustados pelo usuário ainda é um desafio e o uso de estratégias adaptativas deve ser avaliado para cada aplicação.

A representação algorítmica de um AG simples pode ser descrita nos passos do fluxograma da Figura 3.3.

Conforme descrito em [Mich96], uma **população** de **indivíduos** é mantida durante todo o processo iterativo do AG onde cada indivíduo corresponde a uma possível solução do problema de otimização. Este indivíduo é uma estrutura de dados composta por uma ou mais variáveis codificadas em um ou mais cromossomos.

O tamanho da população é um dos parâmetros que deve ser estudado e ajustado conforme a aplicação e a implementação do AG. Esta tarefa não é trivial e possui referências na literatura em [Gref86], [Gold92], [Mich96], [De J07] e outros autores nos 50 anos de existência deste campo de atuação.

Conforme citado em [Gref86] e [Mich96], a população inicial é usualmente composta por indivíduos gerados aleatoriamente de modo a

possibilitar uma melhor exploração do espaço de busca. O uso de estratégias que incluem algumas soluções sugeridas na população inicial também tem apresentado bons resultados, conforme indicado em [Gref86], [Mich96] e [Merz99].

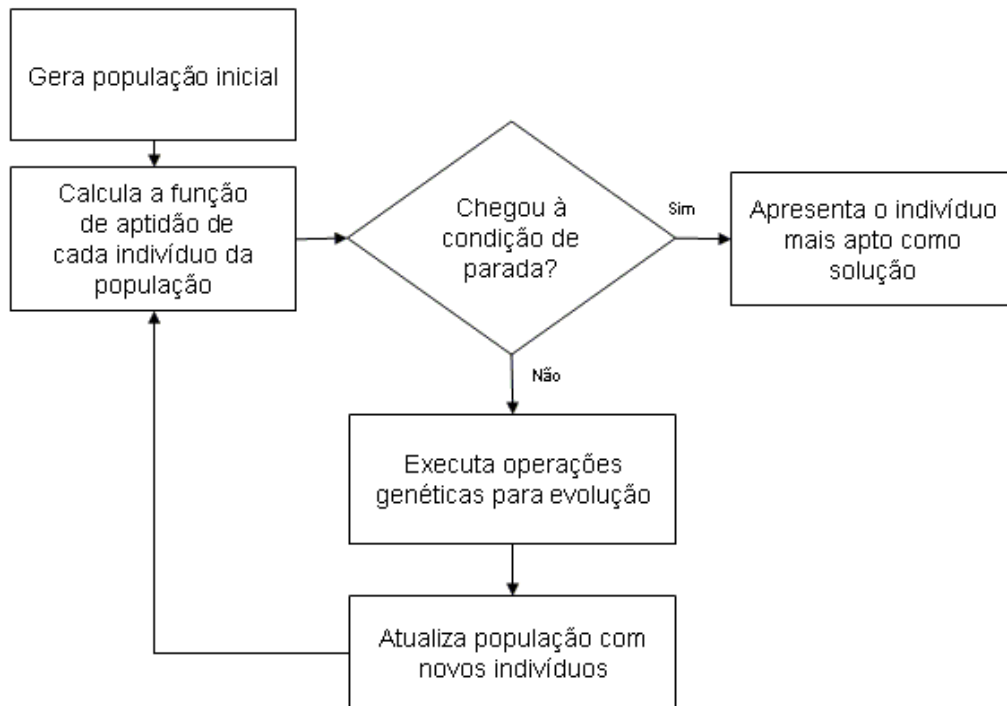


Figura 3.3 - Fluxograma básico de um AG

Em [Gold92] é estudada a influência do tamanho da população para a convergência prematura do AG para soluções sub-ótimas. [De J07] ressalta que o tamanho da população pode ser modificado e adaptado ao longo do processo iterativo.

Assim como em um processo evolutivo biológico, cada indivíduo é avaliado através de uma **função de aptidão** que fornecerá uma métrica do quão apta esta solução se apresenta dada as restrições do problema. Para cada restrição é associada uma penalidade à função de aptidão. Nos problemas de minimização a violação da restrição é adicionada ao valor de aptidão. Para os problemas de maximização, a penalidade é subtraída do valor da função de aptidão para o indivíduo. De forma a ajustar a importância

de cada restrição no problema analisado, é feita uma ponderação dos valores associados à função de aptidão para cada tipo de violação encontrada no processo de avaliação do indivíduo.

A população de indivíduos avaliada é submetida ao processo de seleção. Os indivíduos escolhidos passarão por transformações em seus cromossomos, formando uma nova população de indivíduos. A este ciclo é dado o nome de **geração**. Pretende-se que estas transformações extraiam dos indivíduos mais aptos características presentes em seus cromossomos que permitam a formação de indivíduos ainda mais aptos.

As diversas gerações compõem o processo evolutivo e um critério de parada usual para AGs não adaptativos [De J07] ou com parametrização *offline* [Gref86] é o número máximo de gerações. Na utilização deste critério, o indivíduo mais apto obtido em qualquer uma das gerações do AG será considerado como a solução do problema de otimização.

Conforme observado em [De J07], nos seus comentários sobre [Eshe91] e I. Rechenberg, a utilização de **gerações aninhadas e reinicializações** utilizando informações de um conjunto de gerações anterior é promissora e já apresentou bons resultados.

O algoritmo CHC de [Eshe91] reinicializa a população do AG com indivíduos gerados a partir de um indivíduo mais apto, aplicando-se a ele um operador chamado de mutação cataclísmica. Esta nova população é então composta de uma cópia deste indivíduo mais apto e de todos os indivíduos gerados a partir deste operador citado. Esta estratégia introduz diversificação no processo evolutivo, aumentando a exploração do espaço de busca. Neste algoritmo, a ativação desta estratégia é feita através da avaliação das diferenças entre os valores de aptidão dos indivíduos de cada população. Caso exista pouca dissimilaridade entre os valores de aptidão dos indivíduos, o algoritmo convergiu para uma solução em um local do espaço de soluções e perdeu a capacidade de efetivamente explorar outras áreas com soluções possivelmente melhores.

3.3.3 Codificação

O processo de codificação consiste na adequação da informação fenotípica, proveniente do conjunto de variáveis que compõem o indivíduo, para um cromossomo, conforme ilustrado na Figura 3.4. O cromossomo é uma representação genotípica do indivíduo e permite a delimitação de um espaço de busca conhecido, facilitando o processo de busca de uma solução ótima.

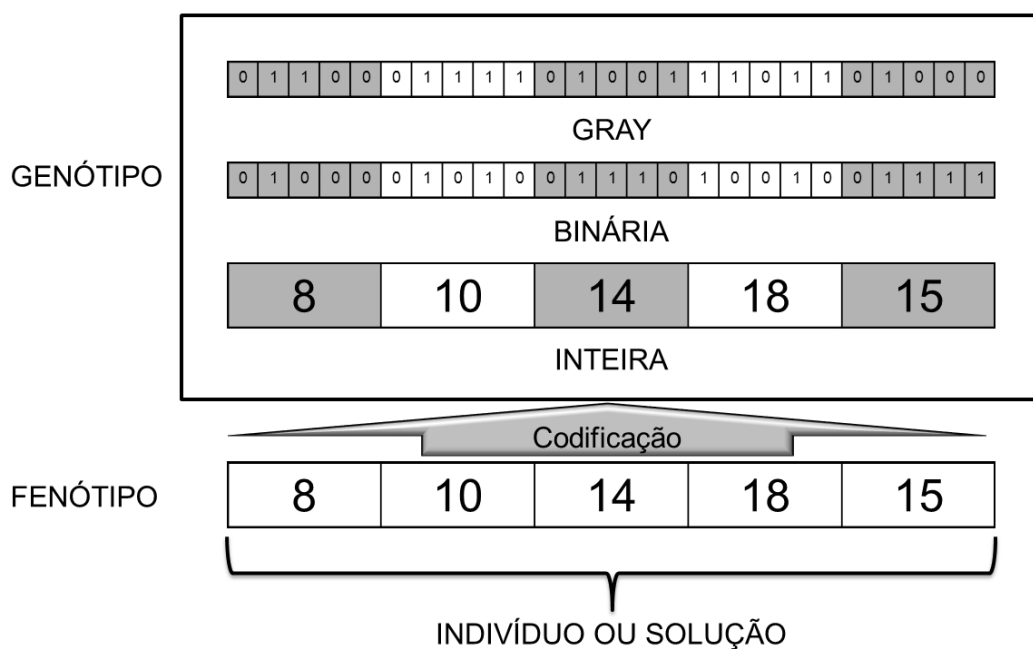


Figura 3.4 - Codificação do fenótipo

Por definição, independente do tipo de representação utilizada, o cromossomo é composto de genes, onde cada gene corresponde a um trecho significativo do fenótipo, como, por exemplo, uma variável do problema. Os alelos são os valores possíveis para cada gene, conforme ilustrado na Figura 3.5.

Em [Deb94] é citada a importância da escolha da codificação para que não exista perda de informação neste processo. O espaço de busca fenotípico é uma redução do problema real, pois abrange apenas as

variáveis selecionadas para representar o problema. A escolha da codificação deve ser tal a minimizar ou eliminar a subsequente perda de informação na transposição do espaço fenotípico para o genotípico.

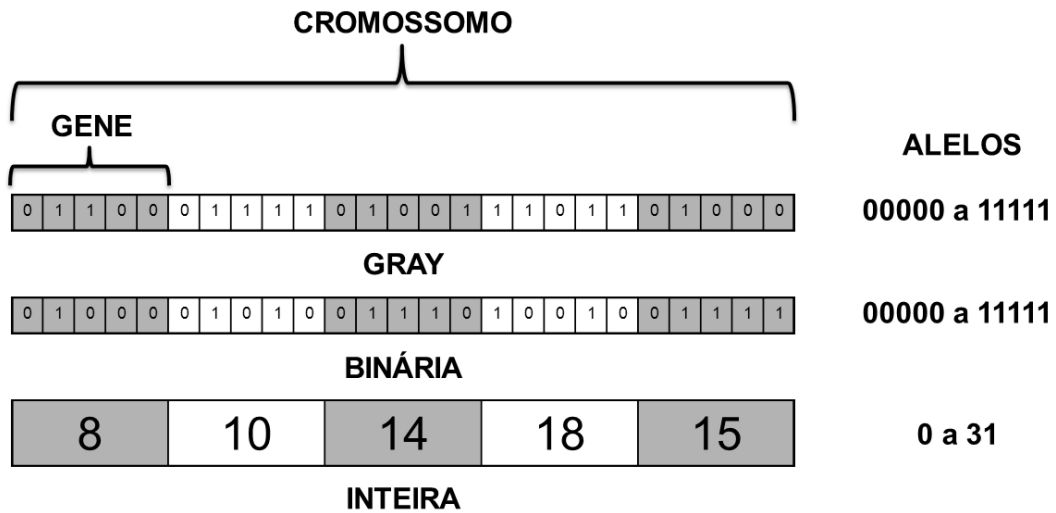


Figura 3.5 - Cromossomos, genes e alelos

Em [Deb94] é citada a importância da escolha da codificação para que não exista perda de informação neste processo. O espaço de busca fenotípico é uma redução do problema real, pois abrange apenas as variáveis selecionadas para representar o problema. A escolha da codificação deve ser tal a minimizar ou eliminar a subsequente perda de informação na transposição do espaço fenotípico para o genotípico.

Ainda em [Deb94], pode-se notar que o desempenho do AG é afetado pela escolha do par codificação e operador de cruzamento, sugerindo uma forte ligação entre estes dois elementos para o sucesso do processo de otimização. Esta relação também é destacada pelos resultados obtidos em [Chak03].

É importante ressaltar que a codificação pode ser utilizada para delimitar o domínio das soluções para aquelas denominadas viáveis. Isto é feito através da imposição de limites rígidos que não poderão ser ultrapassados pelas variáveis que compõem a solução.

3.3.3.1 Codificação binária

Na codificação binária, cada característica fenotípica do indivíduo, identificada por uma variável, é representada através do sistema numérico binário. Em sua implementação mais usual, os números binários que representam estas variáveis neste sistema são agrupados em um único número binário que será o cromossomo deste indivíduo no AG, conforme ilustrado na Figura 3.6.

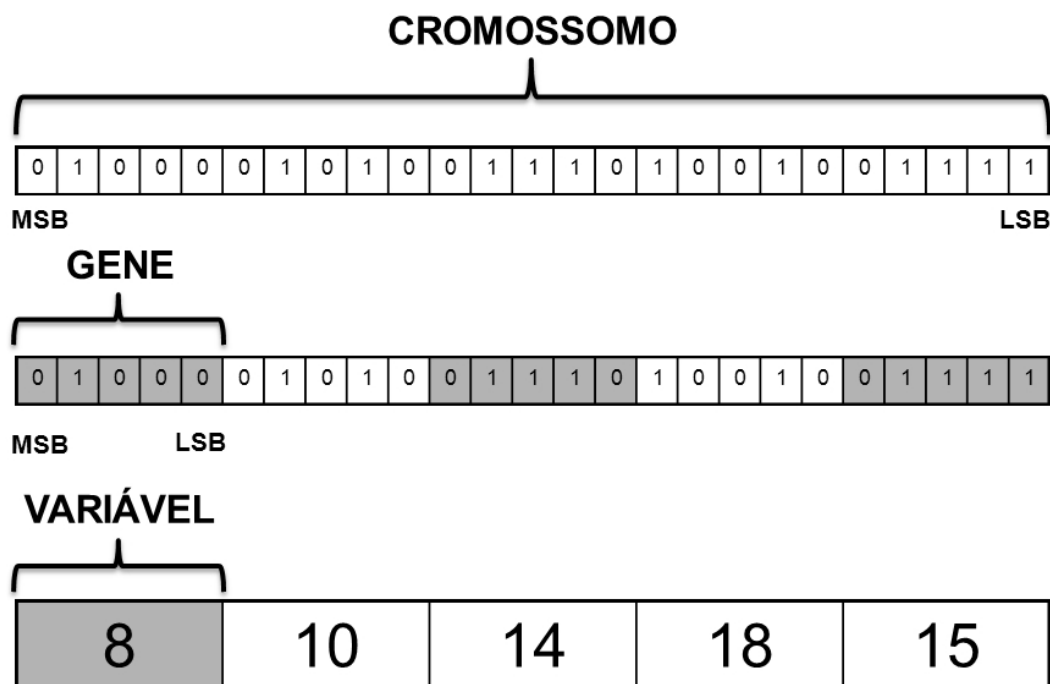


Figura 3.6 - Codificação binária para diversas variáveis

Vale ressaltar que, na codificação binária de cada variável, o número de bits selecionado para representar esta característica fenotípica deve ser tal que permita abranger todo o domínio possível ou desejado do fenótipo. Esta relação de compromisso entre representatividade máxima do domínio do fenótipo e simplificação do espaço de busca é crucial para dimensionar o custo computacional e desempenho dos AGs. Certas características fenotípicas não possuem domínios conhecidos a priori, o que dificulta ainda mais esta escolha. Em [Deb94] é ressaltado que, para variáveis representadas originariamente por números reais, a representação binária, com seu mapeamento fixo, impõe limites máximos, mínimos e de precisão às

possíveis soluções que o AG pode alcançar, conforme ilustrado na Figura 3.7.

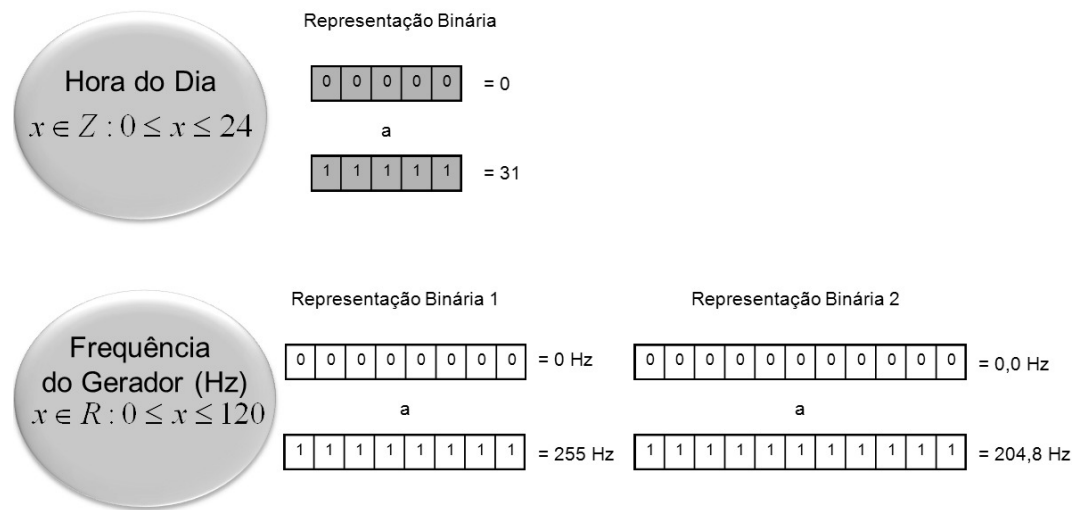


Figura 3.7 - Problemas no mapeamento fixo da representação binária

Conforme apontado em [Deb94], a codificação binária apresenta algumas dificuldades para representar problemas que possuem um espaço de busca contínuo. A primeira dificuldade diz respeito aos *Hamming Cliffs* associados a transições entre certos números binários, onde uma transição para uma solução vizinha no espaço real ou fenotípico requer a alteração de mais de um bit no genótipo. Esta característica, segundo [Deb94], causa um impedimento artificial para uma busca gradual no espaço de busca contínuo.

A questão dos *Hamming Cliffs* é também citada em [Luke09] que comenta que se a codificação do indivíduo é tal que permita que pequenas mudanças no genótipo possam representar pequenas mudanças na aptidão, o trabalho realizado pelo algoritmo de otimização é facilitado. A quantidade de bits alterados em uma codificação binária com 3 bits é ilustrada na Tabela 3.1.

Estas características da codificação binária são amplamente discutidas na literatura com outras referências em [Chak03] e [Rowe04].

Tabela 3.1 - Hamming Cliffs na codificação binária

Decimal (fenótipo)	Binário (genótipo)	Qtde. de Bits Alterados
0	000	n.a.
1	001	1
2	010	2
3	011	1
4	100	3
5	101	1
6	110	2
7	111	1

3.3.3.2 Codificação Gray

O código binário refletido, nomeado de código Gray em homenagem a Frank Gray, pesquisador dos Laboratórios Bell, foi introduzido em 1947 e se constitui em um sistema numérico binário onde dois números sucessivos se diferenciam em apenas 1 bit.

A diferença entre dois números binários com o mesmo número de bits pode ser medida através da distância de *Hamming*, nomeada em homenagem a Richard Hamming. Neste sistema numérico, esta distância corresponde ao número de bits diferentes entre dois números binários. Para o código binário refletido, ou código Gray, a distância de *Hamming* entre dois números sucessivos é sempre igual a 1. Esta comparação está descrita na Tabela 3.2.

Conforme descrito em [Chak03] a representação de um número em código Gray pode permitir a redução do número de ótimos locais, o que, dependendo do problema a ser otimizado, permite um melhor desempenho do AG com relação a outras codificações. Em [Rowe04] é demonstrado que uma função f de um conjunto $\Omega \rightarrow \mathbb{R}$ não pode ter mais ótimos locais em uma codificação Gray do que teria em sua codificação natural.

Tabela 3.2 - Comparação entre distância de Hamming de binário e Gray

Decimal (fenótipo)	Binário (genótipo)	Qtde. de Bits Alterados	Gray (genótipo)	Qtde. de Bits Alterados
0	000	n.a.	000	n.a.
1	001	1	001	1
2	010	2	011	1
3	011	1	010	1
4	100	3	110	1
5	101	1	111	1
6	110	2	101	1
7	111	1	100	1

[Deb94] e [Rowe04] indicam que a distância de *Hamming* igual a 1 existente na codificação Gray elimina o problema dos *Hamming Cliffs* existente na codificação binária, fazendo com que uma pequena variação na característica fenotípica também corresponda a uma pequena variação genotípica. Esta característica propicia uma procura gradual no espaço de busca.

É importante ressaltar que esta representação ainda possui as desvantagens de precisão e mapeamento fixo ressaltadas por [Deb94] para a representação binária, tendo em vista que a escolha do número de bits para representar cada variável impõe limites máximos e mínimos e de precisão para as variáveis da solução.

3.3.3.3 Codificação real

A codificação real é utilizada em problemas onde a codificação natural das variáveis que compõem o fenótipo é melhor descrita no domínio dos números reais.

Com esta codificação direta é possível eliminar problemas de outras codificações que limitam ou diminuem a precisão das características fenotípicas. [Deb94] cita a dificuldade de um operador binário em propagar

blocos significativos dos cromossomos dos pais em um cruzamento. Na codificação real, as próprias variáveis são utilizadas pelo operador, eliminando este problema.

Conforme visto em [Deb94], a escolha de operadores específicos para esta codificação deve levar em conta o seu poder de busca.

3.3.3.4 Codificação inteira

Na codificação inteira, o cromossomo de cada indivíduo é composto por genes com valores pertencentes ao conjunto dos números inteiros. É usual aplicar esta codificação quando a codificação natural das variáveis que compõem o fenótipo é inteira.

Também para esta codificação, existem operadores específicos projetados para permitir uma transmissão eficiente da característica genotípica dos indivíduos selecionados.

3.3.4 Operadores e estratégias

3.3.4.1 Seleção por torneio

A seleção é o operador do AG responsável por selecionar os indivíduos da população que irão participar da recombinação ou cruzamento.

Conforme descrito por [Gold90] “uma forma de seleção por torneio atribuída a um trabalho não publicado por Wetzel foi estudada na Dissertação de Brindle (1981), e estudos mais recentes utilizando esquemas de torneio são encontrados em alguns trabalhos (Goldberg, Korb and Deb, 1989; Muhlenbein, 1990; Suh and Van Gucht, 1987).”

Ainda em [Gold90], a seleção por torneio pode ser descrita como o processo que escolhe um número de indivíduos aleatoriamente de uma população (com ou sem reposição), seleciona o melhor indivíduo deste grupo para o processamento genético subsequente e repete as etapas anteriores

enquanto for necessário (usualmente até quando todos os candidatos a cruzamento foram escolhidos).

Conforme descrito em [Gold90] este algoritmo tem complexidade baixa ($O(n)$), vinculada apenas ao tamanho da população. Para populações grandes, este método apresenta vantagem em relação a outros métodos no aspecto tempo ou custo computacional. Conforme ressaltado em [Luke09] esta é a técnica de seleção preferencialmente utilizada para AGs. Para [Gold90] e [Luke09] esta seleção favorece o uso de processamento paralelo e não necessita de adaptações ou escalonamentos do valor de aptidão.

Em [Luke09] é ressaltada a possibilidade de sintonização da seletividade do algoritmo através do parâmetro tamanho do torneio, que representa o número de indivíduos que participarão deste operador de seleção. Se o valor é muito alto (tendendo ao tamanho da população) a probabilidade que o indivíduo de maior aptidão seja escolhido tende a 100%, o que corresponde ao operador de elitismo. Se o valor é igual a 1, a seleção será equivalente a uma busca aleatória, onde um indivíduo qualquer é selecionado na população. Segundo [Luke09], o valor mais utilizado para torneio em AGs é igual a 2.

A eficiência e a eficácia deste método é ressaltada em [Cham95], [Mitt99], [Haupt04] e [Siva08].

3.3.4.2 Cruzamento

O cruzamento é um operador de recombinação que reproduz os processos biológicos de meiose e cariogamia na reprodução sexuada onde cada membro do par de indivíduos que participa do processo fornece metade de seu material genético para formação de um ovo ou zigoto. Este zigoto possui um conjunto de cromossomos composto pelo material genético dos dois indivíduos selecionados como pais. Sob o ponto de vista da seleção natural, este processo deve gerar indivíduos com material genético que deem

continuidade à adaptação dos pais ao meio e tenham ainda a capacidade de melhorá-la.

Nos AGs, como estipulado por [Deb94], o operador de cruzamento tem o propósito de procurar pelo espaço de busca de modo a criar boas sequências cromossômicas e de combinar estas sequências para formar outras ainda melhores. Ainda segundo [Deb94] e de acordo com [Gold92] e [De J07], os *schemata* propostos por Holland indicavam que partes de alguns genes deveriam permanecer inalteradas para se alcançar boas soluções, exigindo um cuidado do operador de cruzamento em não partir e perder estas combinações. Estas combinações foram definidas por Holland como *building blocks* ou blocos construtivos, em uma tradução livre.

Em [Luke09] é ressaltado que o operador de cruzamento não é equivalente a um tipo de mutação global. Isso significa dizer que não é possível obter qualquer indivíduo possível através do cruzamento de dois indivíduos quaisquer. Conforme ressaltado em [Deb94], esta capacidade de gerar mais indivíduos distintos é proporcional à diferença entre os dois cromossomos dos indivíduos selecionados para cruzamento. [Deb94] classifica o poder de busca de um operador de cruzamento como uma medida do quão flexível o operador é na criação de um ponto arbitrário do espaço de busca. Em uma codificação binária com 3 bits onde, por exemplo, os vetores 100 e 000 seriam aqueles selecionados para o cruzamento, este operador gerará indivíduos com alguns bits de um vetor e outros do segundo vetor selecionado. Neste caso, onde os vetores possuem somente 1 bit de diferença, só os indivíduos 100 e 000 poderão ser gerados por este operador.

Durante o processo evolutivo de um AG, a aplicação dos operadores de seleção e cruzamento ocasiona um aumento de indivíduos similares na população. [De J07] comenta que esta convergência da população em indivíduos com muita similaridade impede que o operador de cruzamento gere indivíduos distintos, reduzindo sua capacidade de diversificar a população. O objetivo do AG é convergir para uma solução de melhor

aptidão. No entanto, se esta convergência ocorre de maneira prematura, esta solução poderá ser sub-ótima, o que não é desejável. [Blum03] e [Coel00] ressaltam que manter a diversidade da população do AG durante o processo é essencial para evitar o efeito da **convergência prematura**, pois permite que o AG explore o espaço de busca antes de intensificar a busca na vizinhança da melhor solução encontrada.

Conforme citado em [Coel00] e em [Luke09], De Jong experimentou o uso de um operador de *crowding* associado ao cruzamento, onde cada indivíduo filho substituiria outro indivíduo da população anterior com o qual tivesse um determinado grau de similaridade. Este processo penaliza os indivíduos similares, removendo-os da população. O objetivo deste operador é aumentar a diversidade da população, prevenindo uma convergência prematura.

Outra técnica desenvolvida para impedir a convergência prematura é o *sharing*, proposto por Goldberg e Richardson e ampliada por Deb e Goldberg, conforme citado em [Coel00]. Nesta abordagem, a similaridade entre os indivíduos é medida nos dois espaços de busca possíveis: fenotípico (variáveis decodificadas) e genotípico (cromossomo codificado). Uma função de *sharing* é obtida para cada espaço e uma modificação na aptidão de cada indivíduo é aplicada com base no valor obtido. Conforme descrito em [Coel00] o uso do *sharing* no espaço fenotípico é melhor para manter a diversidade do que no espaço genotípico e ambos os casos são melhores que o *crowding*.

[Coel00] também cita que técnicas de restrição de cruzamento foram empregadas na literatura, considerando somente indivíduos similares ou somente dissimilares (prevenção de incesto).

As operações de recombinação executadas pelos AG podem gerar **soluções** consideradas **inviáveis** para o problema a ser otimizado. O tratamento do AG para estas soluções deve ser considerado na modelagem empregada. [Blum03] citam 3 métodos mais comuns no tratamento deste

problema: eliminação da solução inviável, reparo da solução e penalização na função de aptidão do indivíduo.

Nos casos de representação genotípica de **múltiplas variáveis** onde se aplica uma codificação inteira ou real em cada variável, pode-se empregar uma estratégia para o cruzamento individual de cada gene ou variável. Esta estratégia consiste na escolha de uma ou mais variáveis para cruzamento com base em uma probabilidade. Desta forma, além da verificação da probabilidade de cruzamento entre dois indivíduos selecionados pelo operador de seleção, também seria verificada uma probabilidade para cada par de variáveis ser selecionada para cruzamento.

Esta estratégia foi apresentada no algoritmo Simple GA [Deb01] do Kanpur Genetic Algorithms Laboratory (KanGAL) dirigido pelo Prof. Kalyanmoy Deb e introduz diversificação no operador de cruzamento, prevenindo uma convergência prematura.

A Figura 3.8 ilustra um exemplo de aplicação da Estratégia de Cruzamento Multivariável Uniforme 50% (Cross Over Uniform 50%), presente no Simple GA do KanGAL.

Neste exemplo, cada indivíduo possui 5 variáveis ou genes em seu cromossomo e, para cada variável, um sorteio é feito com 50% de probabilidade de escolha de um par de variáveis para o cruzamento (caso das variáveis 0 e 3). As variáveis que não são sorteadas para cruzamento são copiadas dos pais para os indivíduos filhos (variáveis 1, 2 e 4 no exemplo).

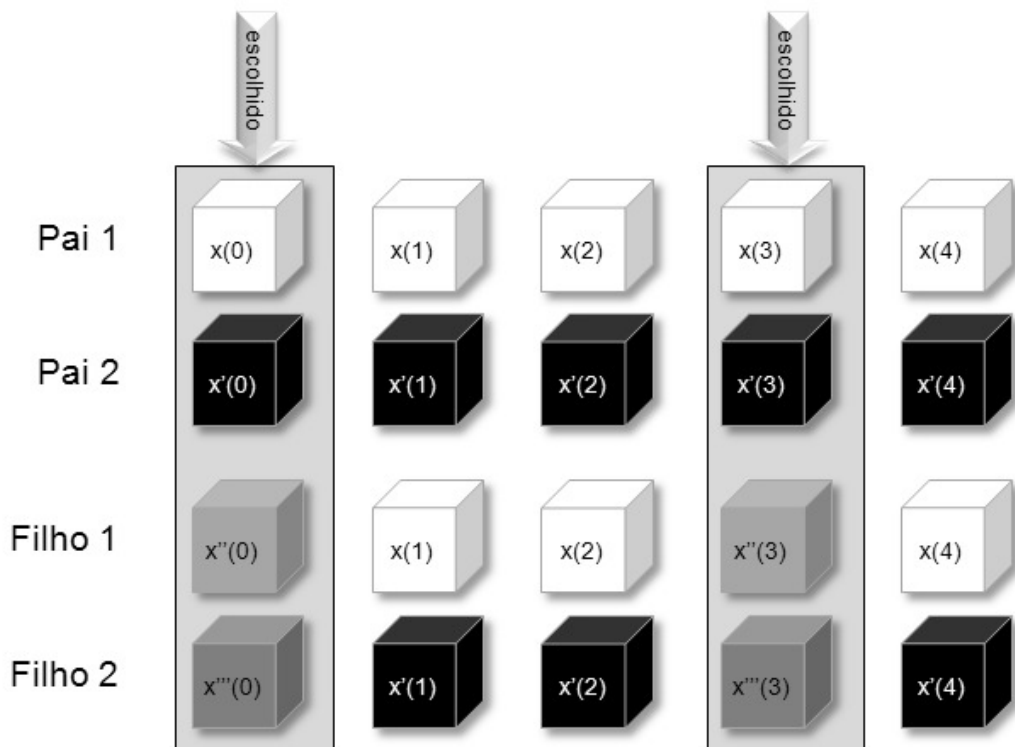


Figura 3.8 - Estratégia de Cruzamento Multivariável Uniforme 50%

a) Cruzamento binário de um ponto

Nesta estratégia, sorteia-se a posição n do cromossomo com codificação binária ou Gray como ponto de corte. A partir deste bit n , o conteúdo menos significativo de um pai é dado a um filho junto com o conteúdo mais significativo do outro pai. O outro filho gerado por este tipo de cruzamento recebe o material genético restante. Um exemplo de cruzamento binário de um ponto com indivíduos de codificação binária de 10 bits é ilustrado na Figura 3.9.

Conforme indicado em [Deb94] e [Luke09] a vulnerabilidade deste operador reside num efeito denominado *linkage* ou ligação entre os genes do cromossomo. Existe uma maior probabilidade que bits no extremo da cadeia sejam separados no cruzamento de 1 ponto. Caso estes bits tenham uma relação forte de dependência entre si e alta influência no valor de aptidão, o

operador de cruzamento poderá descartar boas soluções com maior frequência.

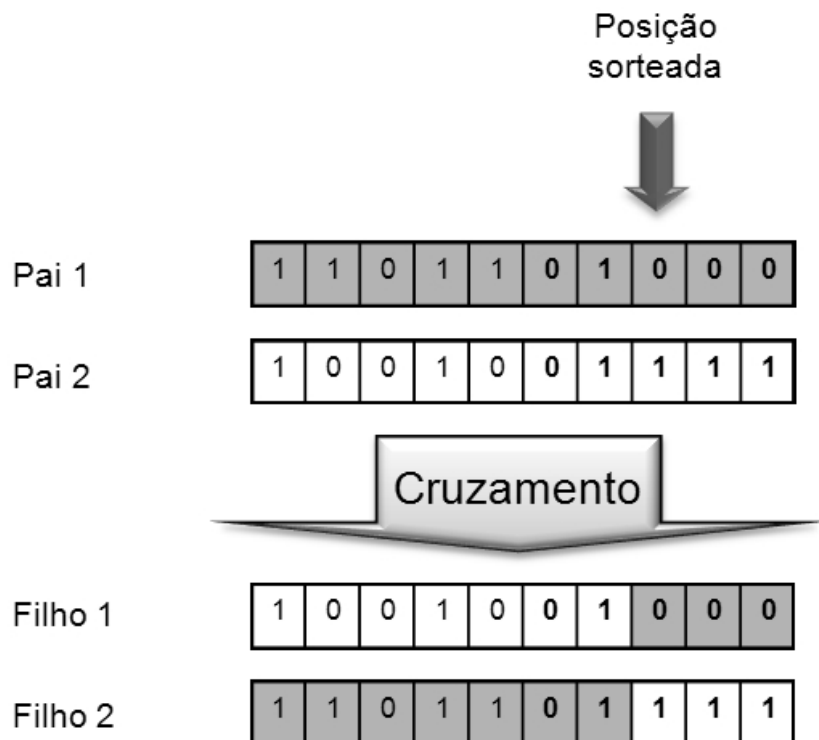


Figura 3.9 - Cruzamento binário de 1 ponto

b) **SBX (Simulated Binary Crossover)**

Conforme indicado em [Luke09] o uso de codificações inteiras ou reais para a representação das variáveis que compõem o fenótipo de um indivíduo em um AG demanda a aplicação de operadores de cruzamento diferentes daqueles utilizados em codificações binárias. Estes novos operadores devem realizar operações matemáticas entre as variáveis que são diferentes da simples troca de bits empregada no cruzamento binário.

[Deb94] avaliou o que chamou de poder de busca dos operadores de cruzamento e propôs o operador SBX (Simulated Binary Crossover) para o cruzamento com números reais. Conforme demonstrado em [Deb94], o operador SBX possui desempenho equivalente ao operador binário de 1 ponto usando codificação binária e superior ao operador real BLX-0,5 de

[Eshe92]. [Deb94] ressalta que o uso de codificação real com SBX resolve os problemas inseridos com o uso da codificação binária: perda de precisão, presença de *Hamming cliffs* e mapeamento fixo que restringe os valores possíveis para a solução.

[Deb94] indica que, num problema de otimização com múltiplas variáveis, cada variável é codificada em um certo número de bits e estes grupos são combinados para formar o cromossomo do AG. O operador de cruzamento binário de 1 ponto pode desfazer o agrupamento de bits de uma mesma variável ou uma combinação de variáveis contíguas simultaneamente.

O operador de cruzamento binário de 1 ponto possui uma propriedade que é mantida no operador SBX : a média dos valores decodificados dos dois pais $((x_1+x_2)/2)$ é igual à média dos valores decodificados dos dois filhos $((y_1+y_2)/2)$. Isso significa dizer que os pontos correspondentes aos filhos são equidistantes do ponto médio dos dois pontos correspondentes aos pais. Esta propriedade de gerar filhos proporcionais ao espalhamento entre os pais, também é verificada no operador BLX- α [Eshe92] com $\alpha=0,5$, conforme ressaltado em trabalho publicado pelo Kanpur Genetic Algorithms Laboratory (KanGAL) em 2005 [KanG05]. Segundo este trabalho, esta é uma condição essencial para o algoritmo exibir auto adaptação e auxiliar o processo de convergência ou divergência para melhores soluções no espaço de busca.

Seguindo esta propriedade da média no cruzamento binário de 1 ponto, os filhos poderão se situar em três posições distintas : dentro do intervalo definido entre os pais, sobre os pais e além do intervalo definido entre os pais. Um fator de espalhamento β determina esta posição e pode ser definido como a razão entre a diferença absoluta dos valores dos filhos e a diferença dos valores dos pais. Este fator é indicado na equação (3.3).

$$\beta = \left| \frac{f_1 - f_2}{p_1 - p_2} \right| \quad (3.3)$$

Caso $\beta > 1$, temos os filhos (f_1 e f_2) além do intervalo definido pelos pais (p_1 e p_2), para $\beta = 1$, os filhos são idênticos aos pais e no caso de $\beta < 1$ temos os filhos dentro do intervalo definido pelos pais.

Conforme verificado em [Deb94], a distribuição de probabilidades do fator β , dado o cruzamento entre dois indivíduos, possui as seguintes características:

- A probabilidade de ocorrência do fator de espalhamento $\beta \approx 1$ é maior do que a de qualquer outro valor de β .
- Para o intervalo de $\beta > 1$, a probabilidade de gerar filhos com β alto é pequena.
- Para valores de β entre 0 e 1, à medida que β aumenta a probabilidade também aumenta. Desta forma, a probabilidade de criar filhos próximos aos pais é maior que aquela associada a filhos distantes de seus pais.

A Figura 3.10, extraída de [Deb94], mostra a distribuição de probabilidades do fator de espalhamento β para o cruzamento binário de 1 ponto, com a codificação binária dos indivíduos de 15 bits de comprimento.

Para alcançar as mesmas características descritas para o operador binário de 1 ponto no operador SBX, com codificação real, em [Deb94] é proposto que a obtenção dos filhos deve obedecer à propriedade da média e a distribuição de probabilidade do fator de espalhamento deve possuir as mesmas características do operador binário de 1 ponto.

Na formulação de [Deb94], para atender a propriedade da média, os filhos gerados pelo operador SBX deverão ser calculados conforme as equações (3.4) e (3.5) descritas a seguir:

$$f_1 = \bar{x} - \frac{1}{2}\beta(p_2 - p_1) \tag{3.4}$$

$$f_2 = \bar{x} + \frac{1}{2}\beta(p_2 - p_1) \quad (3.5)$$

para

$$p_2 > p_1$$

onde

$$\bar{x} = \frac{1}{2}(p_1 + p_2)$$

Logo :

$$\bar{f} = \bar{x}$$

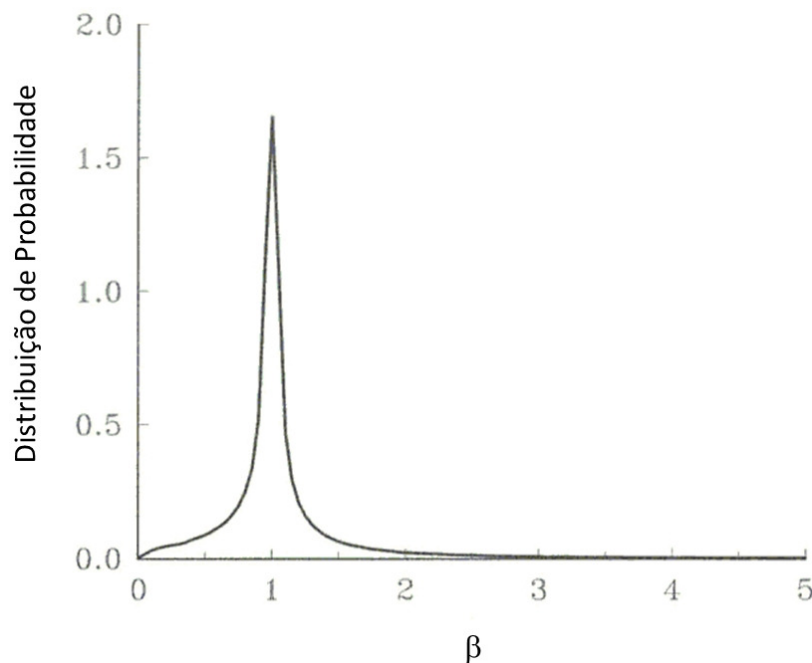


Figura 3.10 - Distribuição de probabilidade de β para cruzamento binário de 1 ponto

A distribuição de probabilidade do fator de espalhamento β no operador SBX deve possuir a mesma forma da distribuição de probabilidade de β no operador binário de 1 ponto, para que mantenha seu poder de busca, segundo [Deb94]. Para isso, [Deb94] propôs a equação (3.6) para a distribuição de probabilidade de β , nos intervalos $\beta \leq 1$ e $\beta > 1$ respectivamente.

$$P(\beta) = \begin{cases} \frac{1}{2}(n+1)\beta^n, & \text{para } \beta \leq 1; \\ \frac{1}{2}(n+1)\frac{1}{\beta^{n+2}}, & \text{para } \beta > 1. \end{cases} \quad (3.6)$$

Segundo [Deb94], os valores de n entre 2 e 5 se aproximam dos resultados obtidos para o operador de cruzamento binário de 1 ponto. A Figura 3.11 extraída de [Deb94] ilustra três casos onde $n = 0$, $n = 2$ e $n = 20$.

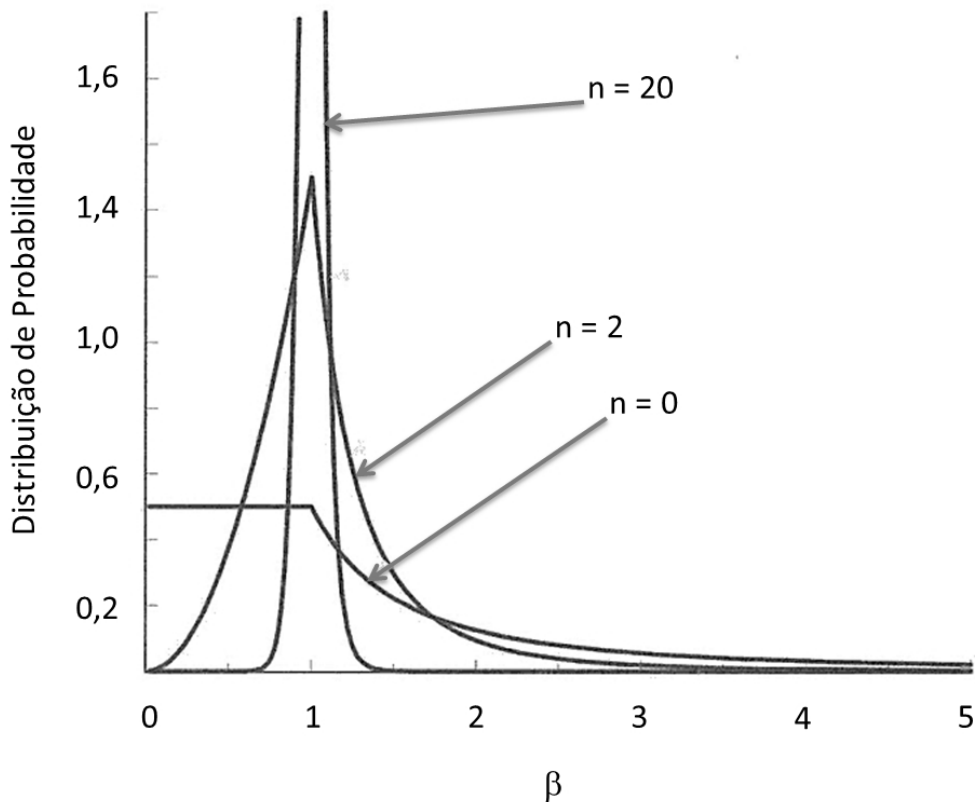


Figura 3.11 - Distribuições de probabilidade para diferentes n

Em trabalhos posteriores [KanG05] o fator de espalhamento β do operador SBX é formulado como um número aleatório que segue a função de distribuição de probabilidade proposta na equação (3.6). Para seu cálculo, é obtido um número pseudoaleatório u segundo uma distribuição uniforme

definida entre $[0,1)$. A ordenada β_q será aquela que faz a área sob a curva de distribuição de probabilidade de β do trecho de 0 a β_q igual ao número u . Considerando a área descrita como a integral definida entre 0 e β_q da equação (3.6) e igualando-a a u , é possível calcular a ordenada β_q conforme a equação (3.7) resultante:

$$\beta_q = \begin{cases} (2u)^{\frac{1}{n+1}}, & \text{para } u \leq 1/2; \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{n+1}}, & \text{para } u > 1/2. \end{cases} \quad (3.7)$$

O valor de β_q obtido é atribuído ao β nas equações (3.4) e (3.5) para a obtenção dos números reais que correspondem aos filhos resultantes da operação de cruzamento SBX.

Ainda conforme [KanG05], é importante ressaltar que os filhos obtidos com este operador terão, essencialmente, duas propriedades:

- A amplitude entre as soluções filhas é proporcional àquela entre as soluções dos pais, e
- Soluções mais próximas dos pais são monotonicamente mais prováveis que aquelas distantes dos pais.

Estas propriedades podem ser verificadas na Figura 3.12 e na Figura 3.13 extraídas de [KanG05], onde são ilustradas as distribuições de probabilidade para soluções filhas onde $n=2$ e os pais estão distantes ($p_1=2,0$ e $p_2=5,0$) ou próximos ($p_1=2,0$ e $p_2=2,5$).

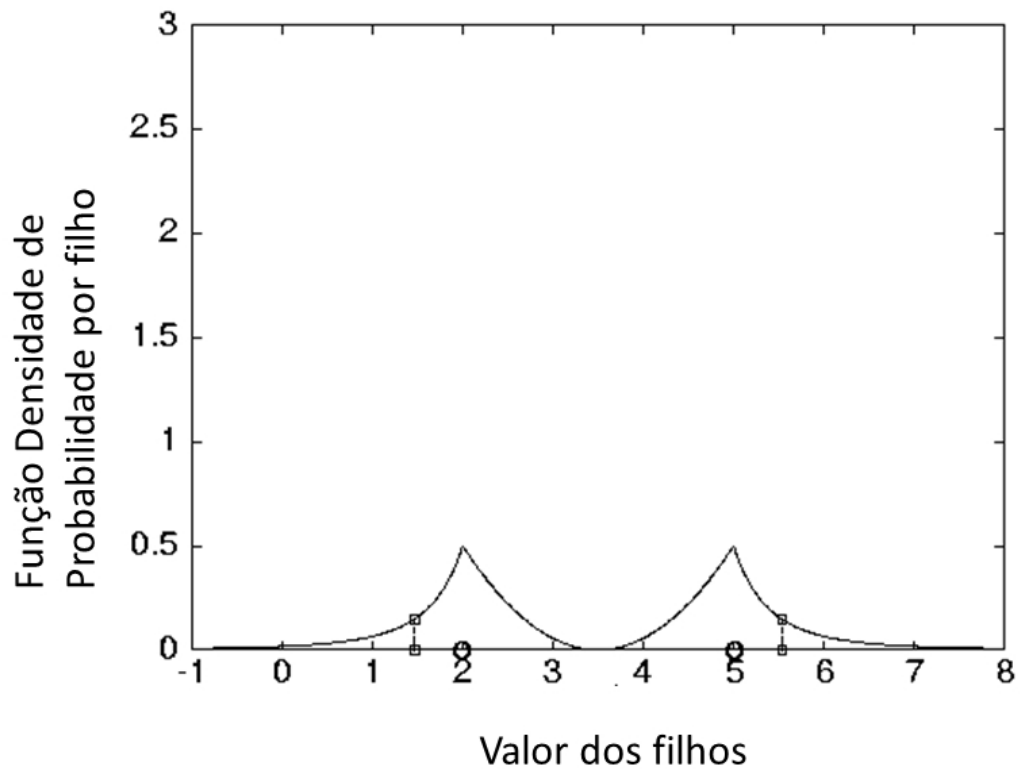


Figura 3.12 - Pais distantes

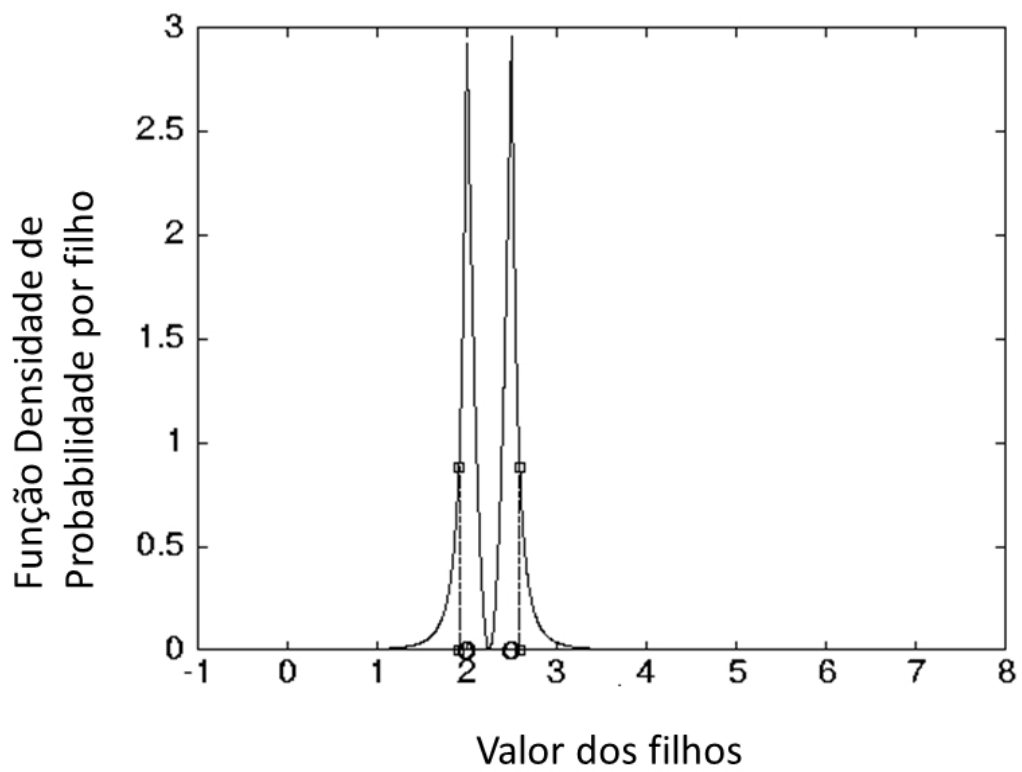


Figura 3.13 - Pais próximos

3.3.4.3 Mutaç o

A muta o   um operador que deve ser aplicado ap s o cruzamento e tem como objetivo restabelecer genes e sequ ncias cromoss micas perdidas durante o processo evolutivo do AG. Conforme definido em [Siva08] este operador   considerado do tipo assexuado, pois atua individualmente em cada solu o filha gerada pelo cruzamento.

Na muta o, alguns genes selecionados atrav s de sorteio dos filhos gerados pelo cruzamento s o alterados para valores gerados aleatoriamente. Como no cruzamento, sua implementa o   vinculada ao tipo de codifica o empregada. Para a codifica o bin ria ou Gray, um ou mais bits s o sorteados para terem seus estados invertidos (0 para 1 e 1 para 0). Na codifica o inteira ou real, o operador de muta o mais usual   a muta o gaussiana ou normal onde um n mero aleat rio, pertencente a uma distribu o gaussiana e escolhido de modo a n o ultrapassar os limites definidos para cada vari vel,   adicionado   vari vel sorteada naquele determinado indiv duo que sofre a muta o.

[Deb96] propuseram, para a codifica o real, um operador de muta o polinomial onde o valor atual da vari vel   alterado para um valor vizinho, usando uma distribu o de probabilidade polinomial com o valor m dio igual ao valor atual da vari vel e sua vari ncia como uma fun o do  ndice de distribu o n . Para realizar esta muta o, um fator de perturba o δ   definido como:

$$\delta = \frac{f - p}{\Delta_{\max}} \quad (3.8)$$

Na equa o (3.8), Δ_{\max}   um valor constante, que representa a m xima perturba o permitida no pai p que gera o filho f . De forma similar ao operador de cruzamento SBX, o valor do filho   calculado a partir da

atribuição de uma função de distribuição de probabilidade a este fator de perturbação δ , indicada na equação (3.9).

$$P(\delta) = \frac{1}{2}(n+1)(1-|\delta|)^n, \quad \text{para } \delta \in (-1,1) \quad (3.9)$$

A Figura 3.14, extraída de [Deb96] demonstra a função de distribuição de probabilidade indicada na equação (3.9) para valores de n iguais a 0, 1 e 4.

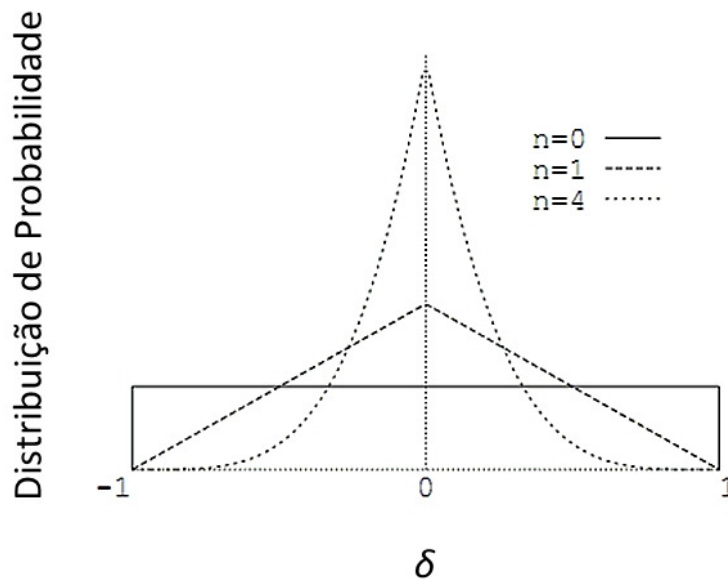


Figura 3.14 - Distribuição de probabilidade para diversos n na mutação polinomial

Para que o valor do filho seja calculado, é gerado um número pseudoaleatório u entre (0,1) que é igualado à integral definida entre 0 e uma ordenada δ_q da distribuição de probabilidade definida na equação (3.9), determinando δ_q para o índice de distribuição n escolhido. O valor δ_q é então obtido através da equação (3.10) descrita a seguir.

$$\delta_q = \begin{cases} (2u)^{\frac{1}{n+1}} - 1, & \text{para } u < 1/2; \\ 1 - [2(1-u)]^{\frac{1}{n+1}}, & \text{para } u \geq 1/2. \end{cases} \quad (3.10)$$

Com base na relação estipulada na equação (3.8), o valor do filho será calculado pela equação (3.11), descrita a seguir.

$$f = p + \delta_q \Delta_{\max} \quad (3.11)$$

3.3.4.4 Elitismo

O Elitismo é uma estratégia ou operador que preserva um conjunto de indivíduos para a população que irá compor a próxima geração. Através deste operador, é possível preservar a memória genética dos indivíduos mais aptos de todas as gerações.

De acordo com [Luke09] esta estratégia foi colocada inicialmente por Kenneth De Jong em 1975 em sua tese de Ph.D intitulada “An Analysis of the Behaviour of a Class of Genetic Adaptive Systems” para a University of Michigan.

[Blum03] ressalta que usualmente os algoritmos genéticos que possuem populações de tamanho fixo mantêm pelo menos o melhor indivíduo sempre na população atual.

[De J07] indica que a utilização de múltiplas reinicializações na população utilizando informações de cada reinicialização para melhorar a performance das execuções subsequentes é uma técnica promissora. Segundo este autor, o sucesso mais notável nesta área é o Algoritmo CHC desenvolvido por [Eshe91]. Neste algoritmo auto adaptativo, quando todos os indivíduos da população do AG atingem um determinado grau de similaridade, um operador denominado mutação cataclísmica renova a população do AG, mantendo o indivíduo com melhor aptidão.

[Rech11] propôs em sua Evolution Strategy (Estratégia Evolucionária ou Evolutiva) de 1994 o uso de execuções aninhadas de AGs partindo da premissa da causalidade suficientemente forte. Segundo este autor: “A teoria da Evolution-Strategy deriva a lei central do progresso como a ideia principal para otimização.” As populações de melhor qualidade, obtidas através do

aninhamento de AGs executados em paralelo em um padrão escalada de montanha (*Hill Climbing*), são preservadas e reutilizadas até que o critério mínimo de convergência seja atingido.

Em [Leun03], é descrita uma técnica que atua no redimensionamento da população com base em uma análise de dissimilaridade baseada na direção ascendente relativa entre suas aptidões aplicada a operadores genéticos de elitismo.

3.3.5 Parâmetros

Os parâmetros de um AG são atribuídos antes de sua execução e podem se manter fixos durante todo o processo evolutivo ou, conforme indicado em [De J07], podem ser alterados para valores distintos no decorrer do processo evolutivo. A estes últimos é atribuído o nome de adaptativos.

[De J07] descreve a sintonia dos parâmetros de um AG como uma tarefa complexa e que ainda precisa ser aperfeiçoada para ser implementada de uma forma automática ou adaptativa. Ainda em [De J07] é possível notar que este tema ainda está em aberto e é recorrente na literatura, tendo sido abordado por diversos trabalhos na área.

[Blum03] destaca a necessidade em se balancear a escolha dos parâmetros que determinarão a frequência de aplicação dos operadores, com base em uma avaliação da influência de cada operador na diversificação ou intensificação no processo de otimização.

3.3.5.1 Probabilidade de cruzamento

Este parâmetro define uma probabilidade para aplicação do operador de cruzamento em dois indivíduos selecionados da população atual do AG. Para [Gref86] e [De J07] os valores ideais para a probabilidade de cruzamento devem ser ajustados considerando que a utilização de valores muito altos pode ocasionar de maneira prematura a perda de estruturas que

representam boas soluções e valores muito baixos ocasionam uma estagnação da busca devido à baixa taxa de exploração. [Gref86] demonstra que valores de 90% para a probabilidade de cruzamento apresentaram bons resultados para diversas aplicações.

3.3.5.2 Probabilidade de mutação

A probabilidade de mutação é utilizada para avaliar em qual dentre os indivíduos filhos gerados após a operação de cruzamento será aplicado o operador de mutação.

Conforme verificado em [Gref86] e em [Blum03] valores altos neste parâmetro aumentam a característica aleatória no processo evolutivo do AG. Valores típicos de 1 a 5% permitem introduzir pequenas perturbações capazes de recuperar soluções perdidas durante a convergência do AG, permitindo sair de soluções sub-ótimas para soluções melhores. Conforme indicado em [Gref86] valores acima de 5% aproximam o AG de um algoritmo de busca randômica, apresentando resultados similares a este tipo de algoritmo.

3.4 GERADORES DE NÚMEROS PSEUDO ALEATÓRIOS

Todos os cálculos que envolvem probabilidade em um algoritmo com base probabilística devem ser feitos a partir de um gerador de números pseudoaleatórios (GNPA) que possa garantir a randomicidade dos números e a reprodutibilidade dos valores gerados para uma mesma semente. Conforme indicado em [Kenn05], a utilização de geradores de números realmente aleatórios não é indicada para simulações que precisam de repetibilidade.

Conforme visto em [Viei04], os geradores de números pseudoaleatórios são programas de computador cujo objetivo é imitar ou simular o comportamento típico de uma sequência de variáveis aleatórias

independentes. São algoritmos específicos, sequenciais e determinísticos (se inicializados em computadores ou momentos diferentes com o mesmo estado inicial ou semente, produzem a mesma sequência de números aleatórios).

Os GNPA's possuem um tamanho de ciclo ou período que indica quantos números podem ser gerados até que a série comece a se repetir. É desejável que os GNPA's possuam um tamanho de ciclo maior possível. Além disso, devem ter uma implementação de baixo custo computacional e os números gerados devem ser independentes e uniformemente distribuídos com correlação entre eles muito pequena.

O GNPA Mersenne Twister [Mats98] é capaz de fornecer números com randomicidade de alta qualidade e com baixo custo computacional, tendo sido projetado com técnicas desenvolvidas para retificar erros de algoritmos anteriores. Segundo [Mats98], o período alcançado com este GNPA é $2^{19937} - 1$. Este algoritmo permanece sendo aperfeiçoado, com implementações em diversas linguagens de programação e arquiteturas de hardware [Mats11].

Em [Knut97] alguns algoritmos de GNPA's são apresentados, sendo discutidas suas características juntamente com diversas técnicas de aperfeiçoamento da sua eficiência computacional. Na aplicação destes algoritmos para geração de números pseudoaleatórios para AGs, o quesito custo computacional deve ser levado em maior consideração do que os demais, dado o perfil repetitivo dos testes que exigem estes números. O uso de algoritmos de simples implementação como o Gerador de Fibonacci Subtrativo Defasado é adequado a este tipo de aplicação. A técnica de embaralhamento (*shuffling*) dos valores gerados pelo GNPA, demonstrada em [Knut97] é conveniente para seu emprego em AGs com processamento paralelo, possibilitando uma independência entre as séries geradas para cada linha de processamento. O uso de séries com período longo e com randomicidade estatisticamente testada [Knut97] associadas a transformadas que permitem adequar os valores a uma distribuição normal [Box58] se demonstrou eficiente para a geração de números pseudoaleatórios em

algoritmos que utilizam cálculos probabilísticos com o nível de exigência de aplicabilidade dos AGs.

Maiores detalhes sobre os GNPA's empregados nesta Dissertação podem ser encontrados no Apêndice B.

3.5 CONCLUSÕES

Neste Capítulo foram apresentados fundamentos para o uso de AGs em um problema de otimização combinatória. No próximo Capítulo, será descrita a metodologia proposta nesta Dissertação para otimizar uma programação de desligamentos de equipamentos em redes elétricas, utilizando um AG.

4.1 FORMULAÇÃO DO PROBLEMA

O problema de programação de desligamentos em redes de energia elétrica é formulado neste trabalho como um problema de otimização, no qual se busca a minimização de uma função que representa a reprogramação de um conjunto de desligamentos. É desejável, na medida do possível, manter as solicitações originadas de diferentes agentes. Para tal, as restrições básicas que devem ser observadas são: o atendimento a toda a demanda em regime normal de operação e na ocorrência de contingências, a não ocorrência de violação dos limites operativos em condição normal e em contingência e restrições específicas de carregamento em alguns ramos especiais. Além disso, devem também ser levadas em consideração as prioridades das intervenções.

Intervenções que levem a qualquer violação nas restrições anteriormente descritas deverão ser reprogramadas, com o mínimo de desvio possível em relação à solicitação original, para períodos onde as mesmas sejam eliminadas ou reduzidas. Quando diferentes pedidos de intervenções se sobrepuserem em um mesmo período, não sendo compatíveis entre si, deve-se levar em consideração a prioridade de cada uma. Logo, o problema pode ser formulado como:

Min (R_{int})

s/a atendimento à demanda;

inviolabilidade de limites operativos;

suportabilidade a contingências simples;

observação às prioridades das intervenções;

atendimento a restrições de carregamento

onde R_{int} representa a reprogramação dos desligamentos.

4.2 OTIMIZAÇÃO DA PROGRAMAÇÃO DE DESLIGAMENTOS VIA ALGORITMOS GENÉTICOS

Cada programação, definida pelo conjunto de horários iniciais de cada desligamento constitui o indivíduo que será otimizado pelo AG em seu processo evolutivo. Para cada programação, um grupo de cenários de diferentes níveis de carregamento deverá ser avaliado para a obtenção de um valor de aptidão. Este cálculo é feito para cada indivíduo de cada população. Com o valor de aptidão calculado, o AG executa seus operadores de seleção, cruzamento, mutação e elitismo na população atual para gerar uma nova população. Neste trabalho foram introduzidas algumas técnicas no AG:

- No **elitismo associado à reinicialização da população**, ou **repopulação**, o melhor indivíduo das últimas g gerações é inserido em uma nova população gerada aleatoriamente sempre com uma semente distinta no gerador de números pseudoaleatórios (GNPA). Esta técnica auxilia na robustez da característica de convergência do AG, permitindo que o algoritmo encontre boas soluções para diversas sementes testadas.

- No **conjunto elite**, o elitismo é executado com um conjunto de indivíduos selecionados ao final de cada g gerações com base em uma **similaridade com o indivíduo de melhor aptidão** associada a uma **análise de diferenças mínimas** entre os elementos do conjunto. Aplicada a uma população diversificada, esta estratégia possibilita a perpetuação no processo evolutivo de bons resultados com diversidade fenotípica.

- Na **escolha de indivíduos na população inicial** é possível incluir algumas programações sugeridas pelo usuário na população inicial, de modo a garantir a convergência para uma solução melhor que a determinada pelo analista e introduzir seu conhecimento no processo evolutivo.

- Para a **codificação inteira**, uma **estratégia de cruzamento de múltiplas variáveis** foi adotada.

4.2.1 Codificação da programação

Dado que o tempo de duração dos desligamentos é fixo e informado pelos agentes, na programação a ser otimizada figurarão apenas os horários iniciais de cada desligamento. É também assumido que todos os desligamentos devem ser agendados em uma janela de tempo pré-definida. Desta forma, o espaço de busca é simplificado sem prejudicar as restrições impostas à otimização.

O fenótipo utilizado na modelagem do AG é um vetor x constituído das variáveis x_1 a x_n correspondentes à hora inicial de cada um dos n desligamentos, referenciadas à hora inicial 00:00 do primeiro dia da janela de tempo com valor máximo $t_{max} > 0$. As variáveis x_1 a x_n podem assumir valores inteiros de 0 a t_{max} , sendo desprezados os minutos na representação fenotípica escolhida.

Para a representação do genótipo do AG, neste trabalho foram exploradas as potencialidades das codificações inteira e binária no sistema numérico Gray, além da codificação binária já testada em [Robe04].

Na codificação inteira, cada variável (hora inicial do desligamento) é representada por um número inteiro. O AG foi então formulado com múltiplas variáveis e objetivo único. Em função da codificação empregada, operações específicas de cruzamento e mutação, bem como estratégias de cruzamento apropriadas foram utilizadas no AG. Em todos os operadores, os resultados foram adequados dentro dos limites rígidos de 0 a t_{max} . Neste caso, os resultados das operações foram aproximados para pertencer ao domínio dos inteiros, onde o valor de cada variável corresponde sempre ao inteiro mais próximo.

Na codificação Gray, todas as variáveis x_1 a x_n do fenótipo são inicialmente convertidas em uma cadeia de números binários x_{b1} a x_{bn} . Cada número binário possui b bits, onde $t_{max} \leq 2^b - 1$, de modo a atender a janela de tempo 0 a t_{max} . Cada elemento desta cadeia é então convertido em um número binário em código Gray equivalente, formando a cadeia x_{G1} a x_{Gn} . As operações de cruzamento e mutação do AG são aplicadas ao número binário de $b \cdot n$ bits formado pela cadeia x_{G1} a x_{Gn} . Um exemplo de codificação pode ser visto na Figura 4.1.

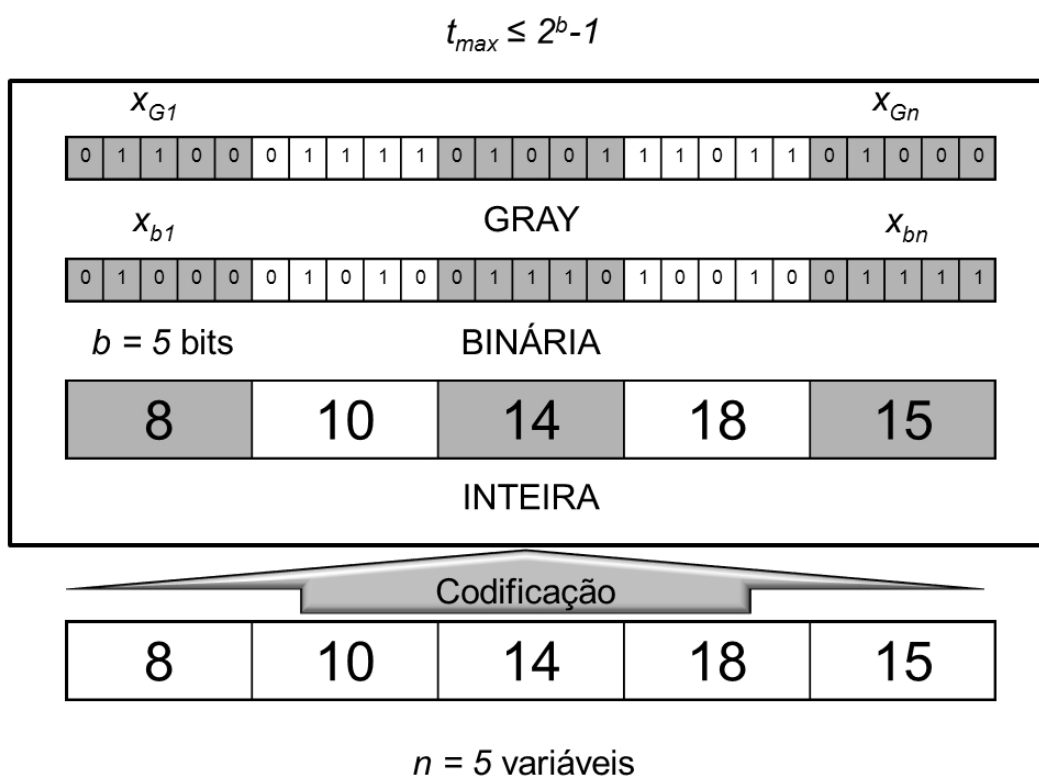


Figura 4.1 - Codificações empregadas no AG

Considerando que $t_{max} \leq 2^b - 1$, a janela de tempo na qual os desligamentos podem ser agendados é determinada pela máxima representatividade possível na codificação escolhida. Caso o número de bits da codificação Gray seja igual a 6 (seis), é possível apenas valores entre 0 e 63 que correspondem a 0:00 do primeiro dia às 15:00 do terceiro dia da janela de tempo. No caso da codificação inteira, a máxima representatividade também é definida em limites rígidos fora dos quais o AG não poderá explorar, considerando a adequação implementada nos resultados dos

operadores do AG. O tamanho ideal deste intervalo não é conhecido *a priori*, devendo ser definido pelo planejador, de tal forma que permita uma programação de desligamentos que atenda os objetivos definidos no problema de otimização anteriormente apresentado.

4.2.2 Condições de carregamento

Com base nos horários iniciais dos desligamentos e suas correspondentes durações é possível compor todos os cenários de operação, a cada hora da janela de tempo de análise. Estes cenários indicarão, a cada hora, quais ramos do sistema estarão desligados, assim como a condição de carregamento da topologia presente. A avaliação da condição de operação do sistema para cada cenário permitirá avaliar a adequação da programação de desligamentos proposta. Neste trabalho, para fins de análise de desempenho do sistema, foram considerados os seguintes níveis de carregamento em função da hora do dia:

- Entre 0:00h e 8:00h – nível de carga leve
- Entre 8:00h e 18:00h – nível de carga média
- Entre 18:00h e 24:00h – nível de carga pesada

Um exemplo de programação, com seus diferentes cenários, para uma análise de 5 desligamentos em uma janela de tempo de 28 horas pode ser visto na Figura 4.2. Nesta figura, os blocos numerados de 1 a 5 correspondem aos diferentes desligamentos considerados na programação com seus horários de início e término determinados pelo eixo x. Nesta programação, 7 cenários com diferentes características, são obtidos e avaliados. Para este exemplo, as horas da programação definidas entre o início da hora 10 e o final da 13 possuem as mesmas características topológicas e de carregamento, constituindo o cenário 2. Este cenário possui uma topologia na qual os equipamentos 1 e 2 estão desligados em condição de carregamento média do sistema.

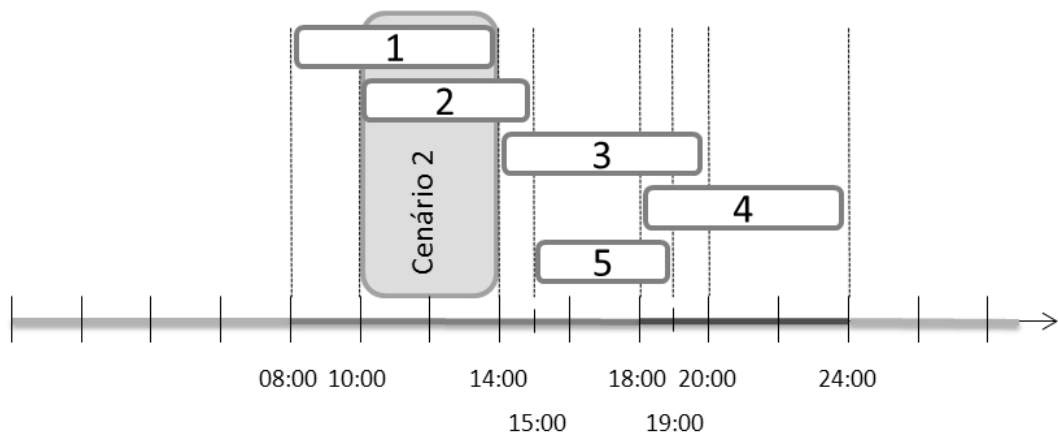


Figura 4.2 - Exemplo de programação com diferentes cenários

Cada cenário resultante da combinação entre as condições de carregamento e uma dada programação cujo fenótipo do AG possui as variáveis x_1 a x_n , determinará um perfil de carga e topologia (esta já considerando a presença dos desligamentos). Cada cenário será avaliado e um valor de aptidão atribuído. O somatório dos valores das aptidões obtidas para cada cenário de uma mesma programação constituirá o valor da função de aptidão do indivíduo do AG.

4.2.3 Definição dos parâmetros do Algoritmo Genético

O AG utilizado neste trabalho foi implementado a partir do Simple GA [Deb01] do Kanpur Genetic Algorithms Laboratory (KanGAL) que apresenta possibilidade de alteração *offline* de diversos de seus parâmetros. Todas as implementações realizadas no AG foram inseridas com parâmetros associados, de modo a facilitar os testes de sintonia de cada estratégia. A metodologia adotada para determinação destes parâmetros guiou-se pelas seguintes premissas :

- Para testes comparativos com [Robe04] foram adotados os mesmos parâmetros de tamanho de população, probabilidade de cruzamento, probabilidade de mutação e número máximo de gerações do trabalho referenciado.

- A sintonia fina dos parâmetros se deu de forma empírica para cada grupo de testes. Para todos os testes, foi efetuada variação individual ou em pares (p.ex. mutação e cruzamento) de valores de parâmetros avaliando as diferenças obtidas para a solução final e para o tempo de convergência em gerações.

Dada a complexidade do AG implementado, foram adotadas algumas relações de compromisso com referência na literatura, conforme indicado na seção 3.3.5 deste trabalho, de modo a minimizar o número de combinações exploradas entre os valores dos parâmetros.

4.2.4 Inclusão de indivíduos na população inicial

Neste trabalho foi adotada uma estratégia de inclusão na população inicial do AG de indivíduos representando programações de interesse, de modo a influenciar o algoritmo a migrar para soluções iguais ou melhores.

Uma vez que estes indivíduos não são impostos nas populações filhas das gerações posteriores e sua influência se dá apenas no início do processo evolutivo, o operador de elitismo foi sempre considerado. Com o elitismo, é garantida a permanência da melhor solução obtida na primeira geração, permitindo perpetuar a influência introduzida no início do processo evolutivo pelas programações de interesse, caso estas se revelem importantes.

As programações de interesse utilizadas na metodologia são aquela proposta pelos agentes e outra proposta pelo analista, podendo as duas figurar simultaneamente na população inicial do AG. Desta forma, a experiência do analista na escolha da programação pode ser adicionada ao processo de otimização, representando uma heurística que pode auxiliar o processo de busca.

Todos os demais indivíduos da população inicial foram gerados aleatoriamente, mantendo os limites rígidos para a codificação utilizada, de

modo a evitar o aparecimento de soluções inviáveis, com programações que extrapolam a janela de tempo estipulada.

4.2.5 Elitismo associado à reinicialização da população

A fim de introduzir diversificação na população do AG, mantendo a melhor solução obtida, foi adotada a estratégia de repopular o AG com novos indivíduos gerados aleatoriamente, mantendo o indivíduo com melhor aptidão (elitismo) após um número de gerações g . Este parâmetro g é determinado antes da execução do AG (*offline*), sendo escolhido de forma empírica, através de um teste de convergência que avalia em quantas gerações a população passa a ter pouca dissimilaridade entre seus indivíduos.

A geração de indivíduos aleatórios é feita após a inicialização do GNPA com uma nova semente, diferente daquela utilizada nas populações iniciais de ciclos de reinicialização anteriores. Esta decisão objetiva aumentar a robustez do AG quanto à influência do GNPA na introdução de possíveis erros de distribuição da população inicial pelo espaço de busca. É desejável que estes indivíduos estejam distribuídos de maneira uniforme pelo espaço de busca para garantir a introdução de diversidade no emprego desta estratégia.

Vale ressaltar que as populações das demais gerações são compostas pelo indivíduo elite e pelos indivíduos gerados pelos operadores de seleção, cruzamento e mutação. A estratégia de elitismo associado à reinicialização da população só será aplicada após g gerações.

Esta estratégia foi adotada inicialmente no presente trabalho de forma intuitiva, tendo sido posteriormente fundamentada em referências presentes na literatura como [Eshe91], [Rech11] e [De J07].

4.2.6 Conjunto elite com um indivíduo

Para a aplicação apresentada neste trabalho foi adotada a escolha do melhor indivíduo ou indivíduo elite a cada geração com base em seu valor de aptidão. A estratégia do elitismo garante a permanência da melhor solução na população do AG, verificada e atualizada a cada geração, durante todo o processo evolutivo.

Em sua implementação mais simples, somente o melhor indivíduo era selecionado para perdurar na próxima população, constituindo um conjunto elite com apenas um indivíduo.

4.2.7 Conjunto elite de tamanho automático

Como alternativa ao elitismo de apenas um indivíduo, foi considerada também a seleção de um conjunto elite de indivíduos que serão inseridos na população a ser reinicializada após g gerações, sendo isto realizado em duas etapas. Na primeira etapa, é formada uma lista restrita de candidatos a participarem do conjunto elite, com aqueles que representam soluções de boa qualidade quando comparadas ao indivíduo de melhor aptidão. Na segunda etapa, buscando eliminar redundância mas mantendo a diversidade do conjunto elite, apenas candidatos que representam programações significativamente diferentes entre si são preservados e integram tal conjunto. Tais etapas são mais detalhadamente descritas a seguir.

Etapa 1: Seleção de candidatos ao conjunto elite

A seleção dos candidatos é feita tendo como base uma medida de afastamento entre as aptidões dos indivíduos da população em relação ao melhor indivíduo nela presente, consistindo dos seguintes passos:

1. Ordenar os indivíduos em função dos valores de suas aptidões, iniciando pelo mais apto.

2. Selecionar como candidatos ao conjunto elite os k primeiros indivíduos cujas diferenças percentuais entre os valores das respectivas aptidões e o valor da aptidão do primeiro indivíduo listado não sejam maiores do que p , sendo p um parâmetro pré-definido.

Etapa 2: Determinação do conjunto elite

Com base na lista de candidatos formada na Etapa 1, é realizado o descarte de indivíduos que representam programações muito semelhantes. Mantendo a coerência com a estratégia elitista desejada, o indivíduo de melhor aptidão não é elegível para descarte. Os seguintes passos são executados:

1. Comparar o próximo indivíduo da lista com todos os demais.
2. Caso o indivíduo corrente se mostre significativamente diferente dos demais, é inserido no conjunto elite e volta-se ao passo 1. Caso contrário, ir para o passo 3.
3. Descartar o indivíduo corrente e voltar ao passo 1.

Para avaliar as diferenças fenotípicas mínimas com os demais elementos do grupo, um candidato é comparado com todos os demais e a diferença entre um número mínimo de variáveis deve ser maior ou igual a um valor em horas. Estes parâmetros são determinados antes da execução do AG e são escolhidos de forma empírica e em conjunto com o parâmetro p de modo a garantir um conjunto elite de no mínimo 2 indivíduos em mais de 80% das reinicializações do AG.

4.3 MÉTRICA PARA AVALIAÇÃO DA APTIDÃO DAS PROGRAMAÇÕES DE DESLIGAMENTOS

A função aptidão é a métrica utilizada pelo AG para guiar o processo evolutivo de acordo com os objetivos propostos. Esta função contém fatores

que penalizarão a aptidão de uma determinada programação ou indivíduo, em função do não atendimento aos objetivos e restrições do problema.

A função de aptidão associada a cada solução é calculada de acordo com a equação (4.1).

$$F_a = T_1 + T_2 + T_3 + T_4 + T_5 \quad (4.1)$$

Onde F_a corresponde ao valor de aptidão que o AG busca minimizar, T_1 representa o termo correspondente aos desvios com relação à programação inicial, T_2 o termo que quantifica todas as violações de atendimento à carga, T_3 a parcela de violações dos limites operativos de todos os equipamentos, T_4 a contribuição das violações das inequações de carregamento e T_5 as violações geradas nos casos de contingências simples. Tais termos serão detalhados a seguir.

4.3.1 Requisito de prioridades na reprogramação das intervenções

A cada desligamento é associada uma prioridade que irá ponderar os desvios de cada programação com relação à programação inicial, proposta pelos agentes que requisitaram os desligamentos. Tal prioridade depende da característica do desligamento solicitado.

A penalização aplicada ao não atendimento à programação inicial e suas prioridades pode ser representada através da equação (4.2).

$$T_1 = \left[\sum_{i=1}^n \Delta h(i) \times P_{pri}(i) \right] \times P_{dsv} \quad (4.2)$$

Onde $\Delta h(i)$ é o desvio, em horas, do i -ésimo desligamento de uma dada programação em relação ao correspondente horário na programação inicial, $P_{pri}(i)$ é a prioridade associada ao i -ésimo desligamento e P_{dsv} a penalidade

aplicada ao desvio na programação. O valor de n representa o número de desligamentos considerados na programação.

4.3.2 Requisito de atendimento à carga

Para cada indivíduo do AG, este requisito é avaliado com a execução de um programa de fluxo de potência para cada cenário onde ao menos um desligamento é considerado. Caso seja detectado o não atendimento à carga (através da não convergência do fluxo de potência), um incremento correspondente a uma penalidade é imposto à função de aptidão. Este termo da função de aptidão está descrito na equação (4.3).

$$T_2 = P_{dem} \times \sum_{t=1}^{na} k_t \quad (4.3)$$

Onde P_{dem} é a penalidade imposta pelo não atendimento à carga, sendo $k_t=1$ quando a carga não é atendida durante a análise do t -ésimo cenário e $k_t=0$ quando a carga é atendida em tal cenário, na representa o número de cenários analisados. Nota-se da equação (4.3) que, considerando o objetivo de minimizar R_{int} (reprogramação de desligamentos) o acréscimo da parcela correspondente à penalização pelo não atendimento a demanda pode fazer com que maiores afastamentos da programação original sejam admitidos para que o atendimento a demanda seja sempre respeitado.

O número de cenários na a analisar é dependente da programação de desligamentos representada pelo indivíduo do AG, considerando o período de cada desligamento, sua condição de carregamento e simultaneidade com demais desligamentos.

4.3.3 Requisito de atendimento aos limites operativos dos equipamentos

Este requisito é avaliado após a execução do programa de fluxo de potência para cada cenário da programação e, caso seja detectado o não

atendimento aos limites operativos dos equipamentos, um incremento correspondente a uma penalidade é imposto à função de aptidão. O termo adicionado à função de aptidão que expressa estas penalidades está descrito na equação (4.4).

$$T_3 = \left(P_v \times \sum_{i=1}^{nv} Violv_i \right) + \left(P_n \times \sum_{j=1}^{nFn} ViolFn_j \right) + \left(P_e \times \sum_{k=1}^{nFe} ViolFe_k \right) \quad (4.4)$$

Onde P_v é a penalidade imposta quando ocorrem violações de tensão na programação analisada, P_n é a penalidade imposta quando ocorrem violações de limites de fluxo nominal, P_e é a penalidade imposta quando ocorrem violações de limites de fluxo de emergência, $Violv_i$ é a magnitude da i -ésima violação de tensão observada, $ViolFn_j$ é a magnitude da j -ésima violação de limite de fluxo nominal observada, $ViolFe_k$ é a magnitude da k -ésima violação de limite de fluxo de emergência observada, nv é o número total de violações de tensão observadas na programação, nFn é o número total de violações de limites de fluxo nominal observadas na programação, nFe é o número total de violações de limites de fluxo de emergência observadas.

Para as violações dos fluxos de potência, existem dois limites superiores sendo observados: o limite nominal e o limite de emergência, este segundo de valor maior, porém devendo ser de curta duração. Para garantir uma maior severidade da violação do limite de emergência foram atribuídos valores das penalidades onde $P_e > P_n$.

Nas violações de tensão são analisados limites inferiores e superiores das tensões nas barras.

4.3.4 Requisito de atendimento às inequações para monitoramento de carregamento

Certos arranjos topológicos permitem uma análise simplificada da influência que o desligamento de uma linha de transmissão tem sobre o

carregamento de outros equipamentos. Dessa forma, através da execução *a priori* do fluxo de potência na situação onde todos os equipamentos estão ligados e outra execução onde somente a linha cujo carregamento se deseja analisar é retirada do sistema, calcula-se um fator de superposição desta linha sobre cada uma das demais linhas do arranjo. Este fator k representa percentualmente a quantidade do carregamento da linha removida que será transferida às demais linhas do arranjo após a remoção e pode ser calculado pela equação (4.5).

$$k_{12} = \frac{F_{1f} - F_{1i}}{F_{2i}} \quad (4.5)$$

Onde F_{1f} representa o fluxo de potência ativa da linha que poderá ser sobrecarregada quando a linha com desligamento programado foi desligada, F_{1i} o fluxo de potência ativa desta linha antes do desligamento ser efetuado e F_{2f} o fluxo de potência ativa da linha que será desligada, obtido antes do desligamento ser efetuado. É calculado um valor k para cada uma das demais linhas do arranjo.

Para cada cenário de programação onde estas duas linhas não estão programadas para desligamento, é verificada a inequação (4.6).

$$F_1 + (k_{12} \times F_2) > F_{1_{emerg}} \quad (4.6)$$

Onde F_1 representa o fluxo de potência ativa da linha que poderá ser sobrecarregada, F_2 representa o fluxo da potência ativa daquela que será desligada, $F_{1_{emerg}}$ é o limite de emergência para o fluxo de potência ativa da linha cuja sobrecarga está sendo avaliada e k_{12} representa o fator de superposição do carregamento da linha 2 sobre a linha 1.

Caso a inequação seja violada, uma penalização será adicionada à aptidão daquela programação, proporcional à diferença entre o limite de emergência e o novo fluxo da linha em sobrecarga. Esta penalidade será

incluída também nos casos das contingências consideradas, mas será desconsiderada caso o equipamento cujo efeito de carregamento que se deseja analisar já esteja definido na programação de desligamentos ou na lista de contingências. Cada inequação pode ser vista como uma análise de contingências simplificada, onde se deseja conhecer apenas o efeito local produzido pela saída de um dado ramo. É importante ressaltar que restrições deste tipo são usualmente observadas pelo ONS em suas análises [ONS04], sendo formuladas com base na experiência e conhecimento sobre o sistema a ser analisado. Na metodologia proposta, a inclusão de tais restrições permite, portanto, representar o conhecimento e experiência dos especialistas da operação do sistema em uma análise local.

Para o cálculo da função de aptidão, o termo correspondente à contribuição das violações das inequações de carregamento dos arranjos topológicos selecionados considerando cada cenário é definido na equação (4.7).

$$T_4 = P_{ineq} \times \sum_{m=1}^{nIn} ViolIneq_m \quad (4.7)$$

Onde: P_{ineq} é a penalidade atribuída a cada violação das inequações de carregamento, $ViolIneq$ é a magnitude da m-ésima violação das inequações de carregamento observada, nIn é o número total de violações das inequações de carregamento observadas na programação.

4.3.5 Requisito de atendimento a contingências simples

Para cada cenário da programação, este requisito é avaliado durante a análise de um conjunto de contingências e, caso seja detectado o não atendimento à carga ou o não atendimento aos limites operativos dos equipamentos, um incremento correspondente a uma penalidade é imposto à

função de aptidão. O termo correspondente às contingências simples é descrito na equação (4.8).

$$T_5 = \sum_{l=1}^{nctg} \left\{ \left(P_{dem}^{(l)} \times \sum_{t=1}^{na^{(l)}} k_t^{(l)} \right) + \left(P_v^{(l)} \times \sum_{i=1}^{nv^{(l)}} ViolV_i^{(l)} \right) + \right. \\ \left. \left(P_n^{(l)} \times \sum_{j=1}^{nFn^{(l)}} ViolFn_j^{(l)} \right) + \left(P_e^{(l)} \times \sum_{k=1}^{nFe^{(l)}} ViolFe_k^{(l)} \right) + \right. \\ \left. \left(P_{ineq}^{(l)} \times \sum_{m=1}^{nIn^{(l)}} ViolIneq_m^{(l)} \right) \right\} \quad (4.8)$$

Onde $nctg$ é o número de contingências a analisar.

Pode-se observar que, neste caso, o atendimento à demanda, aos limites operativos e às inequações de carregamento também deve ser verificado para os cenários da rede em contingência, podendo-se definir diferentes penalidades para cada contingência considerada. Nos casos onde existe contingência simples a ser considerada, cada contingência simples é adicionada ao cenário e suas violações adicionadas ao valor da aptidão da solução.

4.4 DEFINIÇÃO DO GERADOR DE NÚMEROS PSEUDO-ALEATÓRIOS

Foram avaliados dois algoritmos de GNPA's utilizando a mesma estratégia para determinação das sementes (vide seção 4.4.1).

O primeiro algoritmo avaliado é aquele presente no Simple GA da KanGAL [Deb01], atribuído no Algoritmo ao "sga.c (c) E.Goldberg 1986". Não foi possível obter mais informações sobre esta implementação dos autores, porém, pela interpretação do código implementado, nota-se que utiliza técnicas de embaralhamento sobre um gerador de séries de Fibonacci defasado, descritos em [Knut97].

O segundo algoritmo foi o Mersenne Twister obtido em [Mats11] na sua implementação para C/C++ desenvolvida por Richard J. Wagner na versão 1.1 de 28 de Setembro de 2009.

A avaliação foi baseada nos resultados do AG onde o Algoritmo Genético é executado novamente com os mesmos parâmetros, porém com um dos GNPA's selecionado para utilização. Os resultados avaliados foram : o melhor valor de aptidão encontrado em cada execução e o número de gerações necessário para encontra-lo em cada caso.

Para a análise estatística dos resultados obtidos, as simulações efetuadas com os dois GNPA's foram consideradas como amostras na composição do espaço amostral da análise.

4.4.1 Algoritmo para determinação das sementes

Para garantir randomicidade com repetibilidade foi adotada implementação existente no Simple GA da KanGAL [Deb01] que reinicializa o gerador de números pseudoaleatórios com uma nova semente a cada partida ou repopulação do AG.

O valor da semente de cada *partida* é determinado pela equação (4.9).

$$semente = semente_escolhida + \left[(1 - semente_escolhida) * \left(\frac{partida - 1}{max\ partida} \right) \right] \quad (4.9)$$

Onde:

partida é um inteiro variando de 1 até *maxpartida* que indica a partida atual do AG.

semente_escolhida é um número real de 0 a 1 determinado pelo usuário.

maxpartida é um número inteiro que indica o número máximo de reinicializações, conforme determinado pelo usuário.

Esta semente é utilizada na geração de todos os números pseudoaleatórios utilizados pelo AG em cada partida. Todos os cálculos envolvendo probabilidade, sejam eles seleção, cruzamento, mutação e criação de população inicial, utilizam os números pseudoaleatórios gerados por esta semente.

Além dos motivos descritos anteriormente, a utilização de diferentes sementes para cada partida tem como objetivo eliminar o “efeito borboleta” postulado por Edward Lorenz em 1963, onde as condições iniciais de determinados problemas podem influir significativamente no resultado final.

Na metodologia adotada, a reinicialização do GNPA é aplicada tanto nos casos onde a estratégia de repopulação com elitismo é utilizada como naqueles onde o elitismo entre populações não é empregado.

4.5 FLUXOGRAMA DO ALGORITMO PROPOSTO

A Figura 4.3 ilustra o fluxograma principal da metodologia implementada.

No fluxograma principal descrito na Figura 4.3 as linhas tracejadas indicam o laço correspondente à estratégia de repopulação com elitismo e as linhas ilustradas em cor diferenciada representam o laço correspondente ao ciclo de cada geração do processo evolutivo.

Vale ressaltar que o número máximo de gerações utilizado como critério de parada corresponde ao número máximo de repopulações ou partidas multiplicado pelo número de gerações g de cada repopulação.

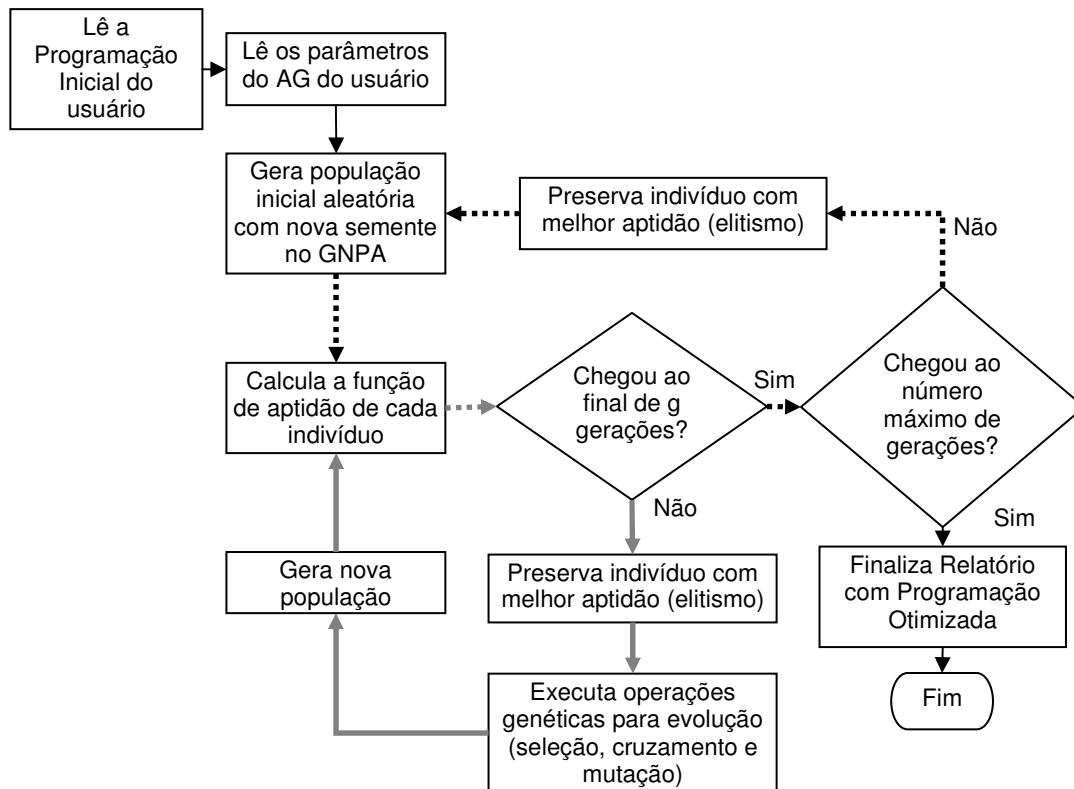


Figura 4.3 - Fluxograma principal da metodologia

4.6 CONCLUSÕES

Neste Capítulo foi exposta a metodologia proposta em sua formulação geral, onde a otimização da programação de desligamentos é entendida como um problema de otimização combinatória resolvido através de um AG. De modo a verificar a eficiência e eficácia da metodologia, resultados de simulações empregando este AG e as métricas e estratégias apresentadas são mostrados e discutidos no próximo Capítulo.

5.1 DESCRIÇÃO DAS SIMULAÇÕES

Diversas simulações foram realizadas visando testar a metodologia proposta. Para tal, foram utilizados os sistemas teste IEEE 14 barras e IEEE 30 barras [Chri99] cujos dados estão descritos no Apêndice A. Os testes com o sistema IEEE 14 barras foram realizados de forma a permitir uma análise comparativa entre os resultados obtidos com a metodologia proposta e aqueles relatados em [Robe04].

Nos testes com o sistema IEEE 14 barras, visando à comparação de resultados, foram feitas simulações utilizando a mesma programação inicial de desligamentos, perfis de carga e limites operativos empregados em [Robe04]. Para o sistema IEEE 30 barras foram feitos testes que avaliaram o desempenho do AG para otimizar a programação de 5, 6, 7, 8, 9 e 10 desligamentos. Nos testes para este sistema, foram adotados os mesmos patamares de carga (pesada, média e leve) escolhidos para o sistema IEEE 14 barras em [Robe04], onde os valores de carga média representam 85% da carga pesada e os valores de carga leve 80% da carga pesada. Para ambos os sistemas considerou-se que todos os desligamentos deveriam acontecer em uma janela de tempo de 32 horas.

Nos testes efetuados para o sistema IEEE 14 barras, foi feita a análise das inequações de carregamento para dois anéis formados pelos nós 6-12-13 e 2-3-4. Para os testes com o sistema IEEE 30 Barras, foram considerados os arranjos em anel 1-2-3-4 e 12-14-15.

Os mesmos valores para penalizações de violações das restrições de demanda e de limites operativos utilizadas em [Robe04] ($P_{dem} = 100$, $P_v = 100$, $P_n = 100$ e $P_e = 150$) foram utilizados. Utilizou-se valor igual a 1 para a

penalidade associada aos desvios da programação original (P_{dsv}) tendo sido verificado nas curvas de convergência do AG que este valor permite que o algoritmo busque novas soluções com desvios maiores que os anteriormente encontrados, de forma a reduzir as violações das demais restrições. Em [Robe04] esta penalidade foi adotada com valor igual a 0,2. Para violações das inequações de monitoramento de sobrecarga, introduzidas pela metodologia do AG proposto, foi adotada uma penalidade P_{ineq} igual a 70, a qual apresentou bons resultados nas simulações realizadas. Assim como feito em [Robe04], as mesmas penalizações escolhidas foram aplicadas para as análises de contingências simples.

Diversos testes foram feitos para a parametrização do AG com melhor relação de compromisso entre convergência rápida e não prematura, considerando os parâmetros disponíveis no Single GA [Deb01] da KanGAL utilizado e as informações disponíveis na literatura [Gref86; Gold92; Koza95; Mich96; Blum03; De J07; Luke09] descritas no Capítulo 3 desta Dissertação. Em todos os testes cujos resultados são apresentados, foi empregada uma parametrização básica do AG com população de 100 indivíduos, probabilidade de cruzamento de 90%, probabilidade de mutação de 2% e seleção por torneio com 3 indivíduos. Para os testes comparativos, foram selecionados de [Robe04] testes com os mesmos parâmetros correspondentes do AG.

Para codificação inteira, foi adotada estratégia de seleção de variáveis para cruzamento com distribuição polinomial, operador de cruzamento SBX e operador de mutação polinomial, ambos com parâmetro $n = 4$ conforme descrito nas equações (3.6) e (3.9). Para codificação Gray, foi utilizado operador de cruzamento binário de um ponto e operador de mutação binário simples, onde cada bit do cromossomo pode ser alterado para o alelo complementar conforme a probabilidade de mutação de 2%. Os limites para todas as variáveis na codificação inteira ou Gray foram de 0 a 31, utilizando 5 bits para cada variável codificada em Gray. Desta forma, a janela de tempo escolhida possui $t_{max} = 31$ e todas as soluções resultantes das operações são escalonadas nestes limites, mantendo-se sempre viáveis.

O elitismo entre gerações foi empregado para todas as simulações. Adotou-se um critério de parada de 700 gerações (70 repopulações com elitismo a cada 10 gerações) nos testes comparativos com [Robe04] no sistema IEEE 14 barras e nos testes com o sistema IEEE 30 Barras.

Todos os resultados apresentados que fazem uso da estratégia do Conjunto Elite, cujos parâmetros foram definidos na Seção 4.2.7, consideram o parâmetro p igual a 40% da aptidão do indivíduo elite, o número mínimo de variáveis diferentes igual a 1 e diferença mínima entre os horários comparados igual a 3 horas. Dos valores simulados para o ajuste desta estratégia, estes apresentaram os melhores resultados, mantendo uma boa relação de compromisso entre o tamanho do conjunto elite e a diversidade dos indivíduos que dele fazem parte.

Nas simulações onde a análise de contingência simples foi considerada, os termos T_2 , T_3 e T_4 da equação (4.1), que representam o caso base onde os equipamentos em contingência não são considerados, foram removidos da composição da função de aptidão das programações avaliadas pelo AG. Desta forma, considera-se somente o caso com mais restrições, reduzindo o custo computacional no cálculo da função de aptidão.

Com relação aos resultados apresentados neste capítulo, é importante destacar que, como um problema de minimização foi formulado neste trabalho, os melhores resultados obtidos são aqueles em que a função de aptidão assume os menores valores.

5.2 IMPLEMENTAÇÃO DO ALGORITMO PROPOSTO

O algoritmo implementado foi criado com base na junção e adaptação de dois módulos em um mesmo ambiente de desenvolvimento:

a) O Simple GA da KanGAL [Deb01] teve seu código adaptado de C para MS Visual C++ e serviu como base para todas as implementações do AG elaboradas neste trabalho.

b) O código em FORTRAN implementado em [Robe04] foi utilizado como base para o cálculo da função aptidão e adaptado para Visual Fortran para utilização do mesmo ambiente de desenvolvimento do código do AG.

Após a adaptação do código do Simple GA para o novo ambiente de programação foram feitas as seguintes implementações:

a.1) Parametrização completa do AG por meio de arquivo de configuração.

a.2) Codificação GRAY.

a.3) Codificação INTEIRA.

a.4) Elitismo entre gerações.

a.5) Estratégia de repopulação com elitismo.

a.6) Opção de inclusão de até duas programações heurísticas na população inicial.

a.7) Integração das funções em Visual FORTRAN para cálculo da aptidão.

a.8) Estratégia de conjunto elite de tamanho automático associada à repopulação com elitismo.

a.9) Pós processamento dos dados obtidos durante todo o processo evolutivo.

a.10) Inclusão de GNPA baseado no Mersenne Twister [Mats11].

As seguintes adequações das funções em FORTRAN para o cálculo da aptidão de cada solução à nova metodologia foram feitas após o código ser portado para o novo ambiente de programação:

b.1) Adição das inequações de carregamento no cálculo da aptidão.

b.2) Parametrização do código para diferentes tamanhos e topologias de rede.

b.3) Integração com o código do AG em MS Visual C++.

5.3 TESTES COMPARATIVOS CONSIDERANDO O SISTEMA IEEE 14 BARRAS

5.3.1 Descrição dos testes 6, 7, 8, 12, 13, 14 de [Robe04]

Para análise comparativa, foram selecionados testes descritos em [Robe04] considerados representativos do problema apresentado e feitos no mesmo sistema teste IEEE 14 barras. Foram eliminados da análise os testes apresentados em [Robe04] que apresentavam parâmetros do AG distintos para o mesmo problema (testes 1, 2, 3, 9, 10 e 11), aqueles que apresentavam versões mais simples do mesmo objetivo (testes 4 e 5) e testes em outro sistema de estudo cujos parâmetros não estavam disponíveis para simulações (testes 15, 16 e 17).

Os testes presentes em [Robe04], utilizados como referência para comparação dos resultados obtidos neste trabalho para o sistema IEEE 14 barras, estão sumarizados na Tabela 5.1.

As programações 1 e 2 utilizadas em [Robe04] para os testes com o sistema IEEE 14 barras descritos na Tabela 5.1 são transcritas na Tabela 5.2.

As contingências utilizadas nos testes do sistema IEEE 14 barras propostas em [Robe04] são transcritas na Tabela 5.3.

Tabela 5.1 - Descrição dos testes 6, 7, 8, 12, 13 e 14 de [Robe04]

Teste	Descrição	Programação Proposta Utilizada
6	Atendimento à prioridade de urgência (4) no desligamento do ramo 12-13 e prioridade 3 no desligamento do ramo 2-4, atendendo às restrições operativas e de demanda.	1
7	Análise considerando 3 contingências, sem considerar desvios e atendendo às restrições operativas e de demanda.	1
8	Análise considerando 3 contingências, com prioridade de urgência (4) no desligamento do ramo 2-5 e atendendo às restrições operativas e de demanda.	1
12	Análise com prioridade de urgência (4) no desligamento do ramo 2-5 e atendendo às restrições operativas e de demanda.	2
13	Análise com prioridade de urgência (4) no desligamento do ramo 4-7 e atendendo às restrições operativas e de demanda.	2
14	Análise com prioridade de urgência (4) no desligamento do ramo 12-13, prioridade 3 no desligamento do ramo 2-4 e atendendo às restrições operativas e de demanda.	2

Tabela 5.2 - Programações utilizadas nos testes comparativos IEEE 14 barras

Intervenção	Circuito Desligado		Programação 1		Programação 2	
	De	Para	Horário Inicial	Duração (h)	Horário Inicial	Duração (h)
1	2	5	08:00	6	08:00	11
2	2	4	10:00	5	10:00	7
3	4	7	14:00	6	14:00	6
4	12	13	18:00	6	18:00	10
5	10	11	15:00	4	15:00	4

Tabela 5.3 - Lista de contingências dos testes com o sistema IEEE 14 barras

Contingência	Circuito Desligado	
	De	Para
1	3	4
2	6	13
3	13	14

5.3.2 Comparação com os testes 6, 7, 8, 12, 13 e 14 de [Robe04]

A seguir são apresentados resultados obtidos com o AG proposto neste trabalho, os quais são comparados com aqueles encontrados em [Robe04].

Para os testes comparativos, foram utilizados os mesmos limites operativos (tensão máxima e mínima e fluxo de potência ativa de emergência e nominal) e de demanda (carga leve, média e pesada) efetuando-se a comparação do melhor resultado de cada teste obtido através do AG proposto nesta Dissertação com a programação otimizada do teste correspondente obtida em [Robe04]. Para isso, a programação obtida em [Robe04] foi incluída na população inicial do AG sendo verificada sua aptidão, violações e desvios dentro da métrica para a função de aptidão apresentada nesta Dissertação. Nesta comparação foram avaliados os desvios da programação original, as restrições violadas e o número de gerações para obtenção da solução encontrada como medida do tempo computacional. Para a análise de inequações, foram utilizados os arranjos em anel 2-3-4 e 6-12-13, onde a influência da retirada das linhas 2-4 e 6-12 sobre as demais de cada arranjo foi analisada.

Todos os resultados para o AG proposto apresentados nesta seção foram obtidos com a aplicação da codificação GRAY, estratégia de conjunto elite automático e repopulação com elitismo. Assim como em [Robe04], os resultados aqui apresentados representam o melhor resultado obtido de diversas simulações executadas para cada teste.

Tabela 5.4 - Comparação de resultados para o Teste 6

[Robe04]				AG Proposto			
Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
03:00	-5	4,0	10,36	04:00	-4	2,8	8,87
10:00	0			10:00	0		
03:00	-11			04:00	-10		
18:00	0			18:00	0		
11:00	-4			15:00	0		

Para o Teste 6, os dois AG empregados apresentaram resultados sem violações dos limites operativos e de demanda testados, porém o AG proposto neste trabalho apresentou um desvio médio menor, reduzindo os desvios individuais de três dos 5 desligamentos.

Tabela 5.5 - Comparação de resultados para o Teste 7

[Robe04]				AG Proposto			
Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
00:00	-8	9,6	34,75	01:00*	+17	12,8	28,65
01:00*	+15			01:00	-9		
00:00	-14			01:00*	+11		
24:00	+6			00:00	-18		
20:00	+5			06:00	-9		

*dia seguinte

Para o Teste 7, o AG proposto neste trabalho apresentou um valor de aptidão melhor, através do deslocamento do quinto desligamento de um horário de carga pesada (20h) para um horário em carga leve (6h), o que minimizou as violações geradas pelo desligamento do circuito 10-11, em detrimento de um valor maior no desvio em relação à programação solicitada (de 9,6h para 12,8h). As violações presentes nos resultados otimizados são consequência do desligamento do circuito 6-13, analisado como uma das

contingências na proposição de [Robe04], utilizada como parâmetro comparativo. Foi possível observar que ambos AGs foram capazes de minimizar as violações, sem contudo conseguir eliminá-las. Isto se explica pelo fato de que, para os valores dos limites operativos empregados, a contingência do ramo 6-13 sempre leva a violações das restrições, independente do patamar de carregamento considerado.

Tabela 5.6 - Comparação de resultados para o Teste 8

[Robe04]				AG Proposto			
Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
08:00	0	3,8	46,02	07:00	-1	6,8	36,88
03:00	-7			01:00*	+15		
08:00	-6			07:00	-7		
24:00	+6			24:00	+6		
15:00	0			13:00	-2		

*dia seguinte

Para o Teste 8, o AG proposto neste trabalho apresentou um valor de aptidão melhor, através do deslocamento do quinto desligamento de um horário de carga pesada (15h até 18h) para um horário em carga média (13h até 16h), o que minimizou as violações geradas pelo desligamento do circuito 10-11. Isso só foi possível com o deslocamento dos desligamentos 1 e 3 para um horário 1h mais cedo, de modo a evitar o aparecimento de mais um cenário figurando os desligamentos 1, 3 e 5 simultaneamente. Estas minimizações de violações ocorreram em detrimento de um valor maior no desvio em relação à programação solicitada (de 3,8h para 6,8h). Assim como no Teste 7, as violações presentes nos resultados otimizados são consequência do circuito 6-13 ser uma das contingências consideradas a qual sempre leva a violações. Mesmo assim, ambos AGs foram capazes de minimizar as violações com a reprogramação de desligamentos.

Tabela 5.7 - Comparação de resultados para o Teste 12

[Robe04]				AG Proposto			
Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
07:00	-1	3,8	15,45	06:00	-2	6,0	13,56
11:00	+1			24:00	+14		
24:00	+10			11:00	-3		
20:00	+2			21:00	+3		
20:00	+5			17:00	+2		

Para o Teste 12, o AG proposto neste trabalho apresentou um valor de aptidão melhor que aquele encontrado no mesmo teste em [Robe04], eliminando a violação de 4,38% no limite de emergência do fluxo de potência ativa, existente no circuito 2-3, em detrimento de um valor maior no desvio em relação a programação solicitada (de 3,8h para 6,0h). A solução final encontrada pelo AG proposto atende a todas as restrições impostas sem nenhuma violação.

Tabela 5.8 - Comparação de resultados para Teste 13

[Robe04]				AG Proposto			
Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
09:00	+1	3,8	19,37	24:00	+16	7,4	14,06
01:00	-9			10:00	0		
14:00	0			11:00	-3		
14:00	-4			11:00	-7		
20:00	+5			17:00	+2		

Para o Teste 13, o AG proposto neste trabalho apresentou um valor de aptidão melhor que aquele encontrado no mesmo teste em [Robe04], eliminando todas as violações existentes nos circuitos 2-4, 4-9, 5-6 e 6-13 e reduzindo a violação limite nominal do fluxo de potência ativa do circuito 13-

14 de 14,57% para 5,98%, em detrimento de um valor maior no desvio da programação solicitada (de 3,8h para 7,4h).

Tabela 5.9 - Comparação de resultados para Teste 14

[Robe04]				AG Proposto			
Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
06:00	-2	3,0	21,28	24:00	+16	4,4	13,94
10:00	0			10:00	0		
18:00	+4			11:00	-3		
18:00	0			18:00	0		
06:00	-9			18:00	+3		

Para o Teste 14, o AG proposto neste trabalho apresentou um valor de aptidão melhor que aquele encontrado no mesmo teste em [Robe04], eliminando todas as violações existentes nos circuitos 2-3, 4-9, 5-6 e 6-13 e reduzindo a violação limite nominal do fluxo de potência ativa do circuito 13-14 de 19,31% para 9,29%, em detrimento de um valor maior no desvio à programação solicitada (de 3,0h para 4,4h).

Para avaliação do tempo computacional de cada metodologia, foram executados testes de tempo de execução da implementação de [Robe04] e do AG proposto nesta Dissertação, considerando os mesmos parâmetros dos testes propostos em [Robe04] e a mesma plataforma computacional para execução das duas implementações. Os resultados destes testes são demonstrados na Tabela 5.10, onde o valor em segundos corresponde ao tempo total de execução das duas implementações, utilizando a parametrização de cada teste com um total de 700 gerações. A plataforma computacional utilizada para os testes possuía processador Intel Core 2 Duo CPU 8400 @ 2.26GHz em sistema operacional Windows XP Professional SP3.

Tabela 5.10 - Tempos computacionais dos testes com o sistema IEEE 14 barras

Teste	AG [Robe04] (s)	AG Proposto (s)
6	608	498
7	915	648
8	919	690
12	636	503
13	623	506
14	625	505

A partir dos resultados da tabela, é possível verificar que a metodologia implementada em [Robe04] apresentou tempos de execução maiores que o AG proposto, em todos os testes. Vale ressaltar que o AG proposto utilizou a codificação GRAY e as estratégias de conjunto elite automático e repopulação com elitismo em todas as execuções descritas.

No tocante à eficiência computacional, foram comparados os números de gerações necessárias para que o AG proposto e o de [Robe04] encontrassem as programações descritas da Tabela 5.4 até a Tabela 5.9. Os resultados de [Robe04] foram obtidos das Figuras B.6 a B.8 e B12 a B14 daquele documento. O número de gerações descritos na Tabela 5.11 para o AG proposto considera uma contagem absoluta de gerações desde a inicialização do AG e a execução da estratégia de repopulação com elitismo a cada grupo de 10 gerações não reinicializa este contador.

Tabela 5.11 - Gerações necessárias até a solução - testes IEEE 14 barras

Teste	AG [Robe04]	AG Proposto
6	680	12
7	620	7
8	280	14
12	490	9
13	300	9
14	90	6

Levando em consideração os resultados obtidos para os tempos computacionais da execução de 700 gerações dos dois AGs descritos na Tabela 5.10, e os números de gerações para se alcançar a programação

otimizada descritos na Tabela 5.11, pode-se observar uma redução significativa no tempo computacional para se alcançar a melhor solução no AG proposto.

5.3.3 Análise Qualitativa do AG proposto

A seguir será apresentada uma análise qualitativa do desempenho do AG proposto, no que diz respeito à diversificação das soluções no processo de busca e à característica de convergência, tendo como foco a influência do tipo de codificação e de estratégias empregadas. Para tal, serão apresentados resultados observados nos testes com o sistema IEEE 14 barras.

1) Análise da diversificação durante o processo de busca

Foram realizados testes com o AG, com diferenças em sua codificação e operadores de mutação e cruzamento, para avaliar o impacto da utilização das codificações GRAY e INTEIRA nos problemas apresentados nos testes 6, 7, 8, 12, 13 e 14 de [Robe04].

Análise da Diversificação das Populações

De modo a analisar a diversificação das populações quando se utilizou as codificações GRAY e INTEIRA, foi adotado o procedimento descrito a seguir. Para cada população do processo evolutivo foram calculadas as variâncias das variáveis do problema (cinco variáveis, cada uma correspondendo ao horário inicial de um dos desligamentos a serem agendados). No caso da codificação GRAY, considerou-se para o cálculo das variâncias os números inteiros que se encontravam representados no espaço codificado, os quais representam os horários iniciais de cada desligamento. Logo, foram calculadas as variâncias de cada variável (horário inicial de um dado desligamento), em cada geração, podendo ser estas

consideradas medidas de espalhamento do horário inicial de cada desligamento, em uma dada população. Da Figura 5.1 até a Figura 5.12 estão plotados os valores de variância obtidos em cada geração do processo evolutivo, para diferentes testes. Em cada uma das figuras, cada ponto plotado corresponde à variância do horário inicial de um dado desligamento, para cada geração. Os resultados são apresentados para um total de 11 gerações, englobando a população inicial da primeira repopulação do AG e as 10 gerações subsequentes.

Nos resultados apresentados da Figura 5.1 até a Figura 5.12 a diversificação do AG em função da codificação empregada pode ser avaliada considerando uma relação direta entre a diversificação e a variância de cada variável ou desligamento. As situações em que um maior valor de variância é observado correspondem a uma maior diversificação das populações, pois indica a presença de uma maior diversidade de indivíduos ao longo do processo de busca. Vale ressaltar que a capacidade de diversificação não está associada apenas à codificação adotada, mas também ao uso de outros operadores de cruzamento e mutação decorrentes do emprego de cada tipo de codificação. Das figuras apresentadas, pode-se perceber que a codificação GRAY permite manter uma maior diversidade durante o processo de busca, o que pode ser benéfico para escapar de ótimos locais.

Em todos os casos foi possível perceber que, quando se utilizou a codificação INTEIRA, os indivíduos obtidos ao final de cada grupo de 11 gerações se concentram na vizinhança do indivíduo elite, indicando que a população neste momento fica composta, na sua maioria, de clones deste indivíduo que representa a melhor solução. No caso da codificação GRAY, a presença de valores altos de variância nos gráficos apresentados, indica que a população permaneceu diversificada.

A manutenção da diversidade obtida com a codificação Gray foi demonstrada nesta seção. Tal característica ajuda a reduzir o risco de convergência prematura, sendo benéfico para a eficácia do processo de busca [Blum03].

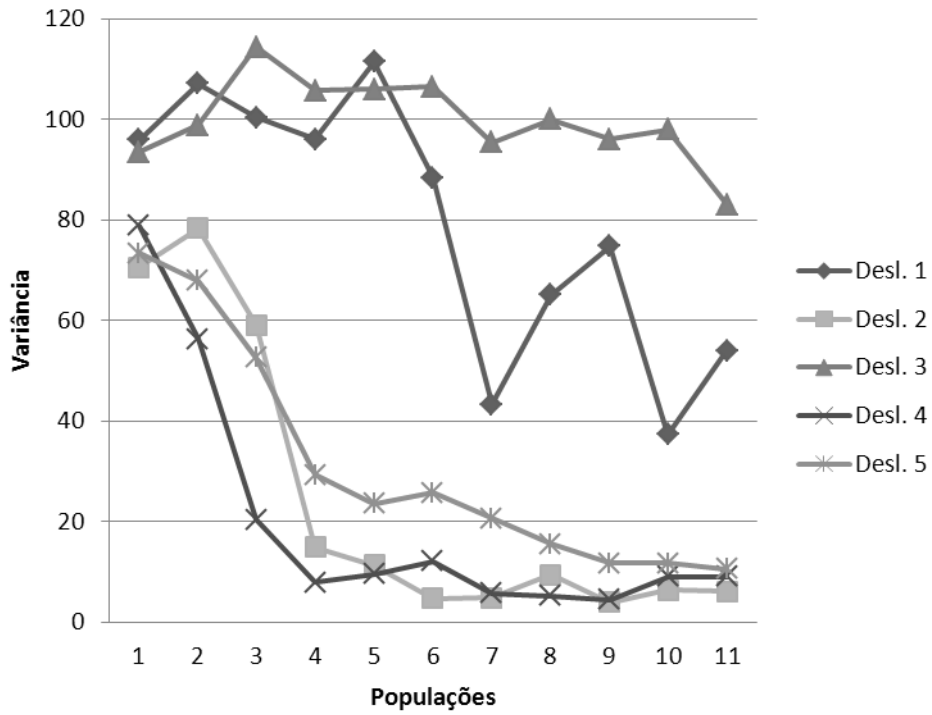


Figura 5.1 - Diversificação para Teste 6 com codificação GRAY

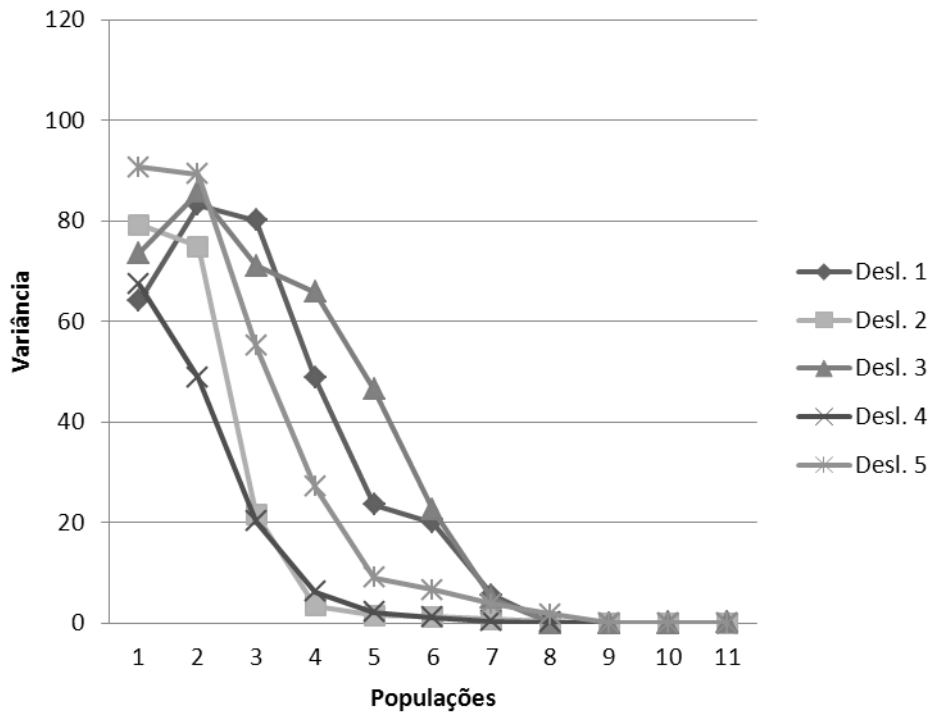


Figura 5.2 - Diversificação para Teste 6 com codificação INTEIRA

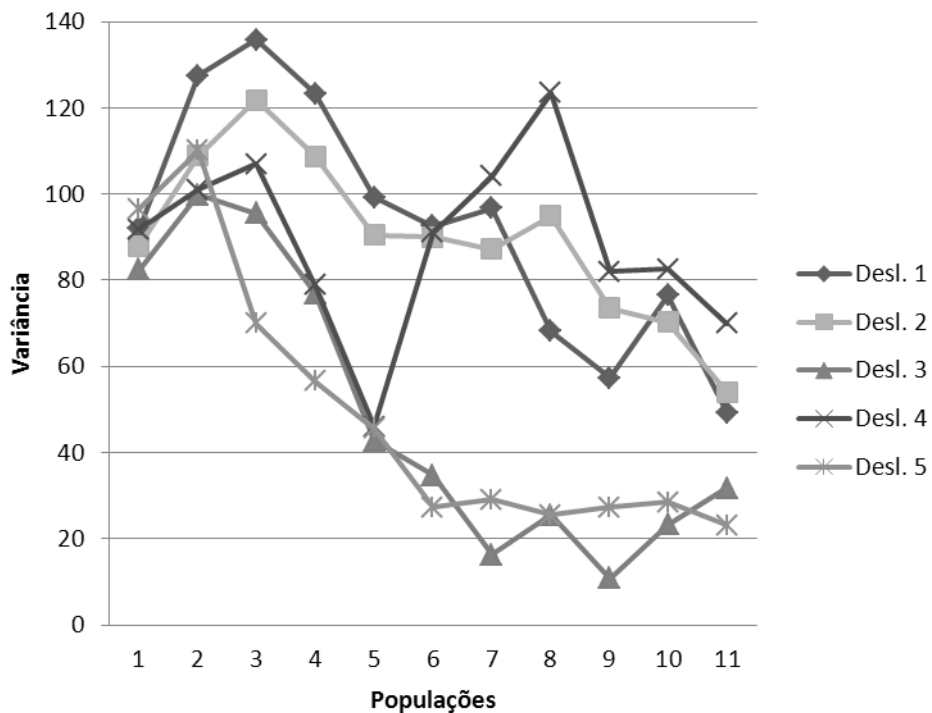


Figura 5.3 - Diversificação para Teste 7 com codificação GRAY

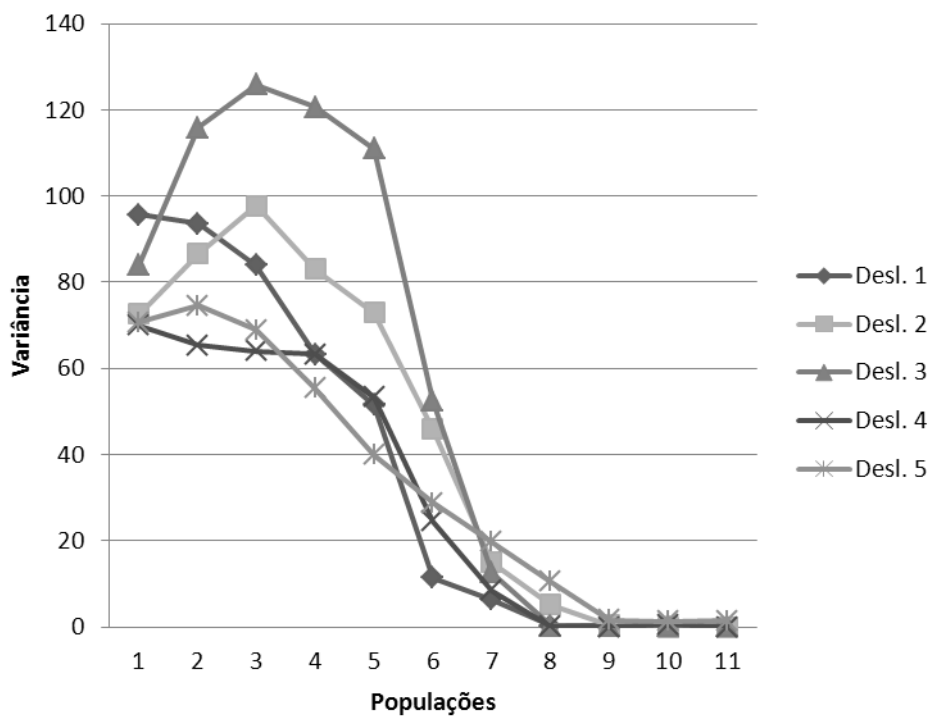


Figura 5.4 - Diversificação para Teste 7 com codificação INTEIRA

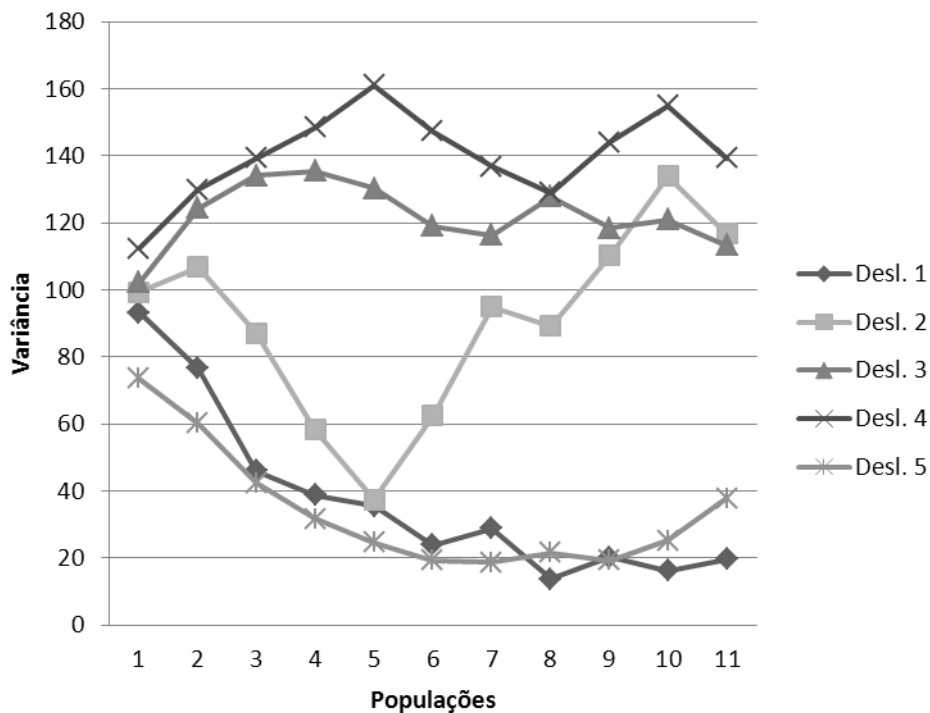


Figura 5.5 - Diversificação para Teste 8 com codificação GRAY

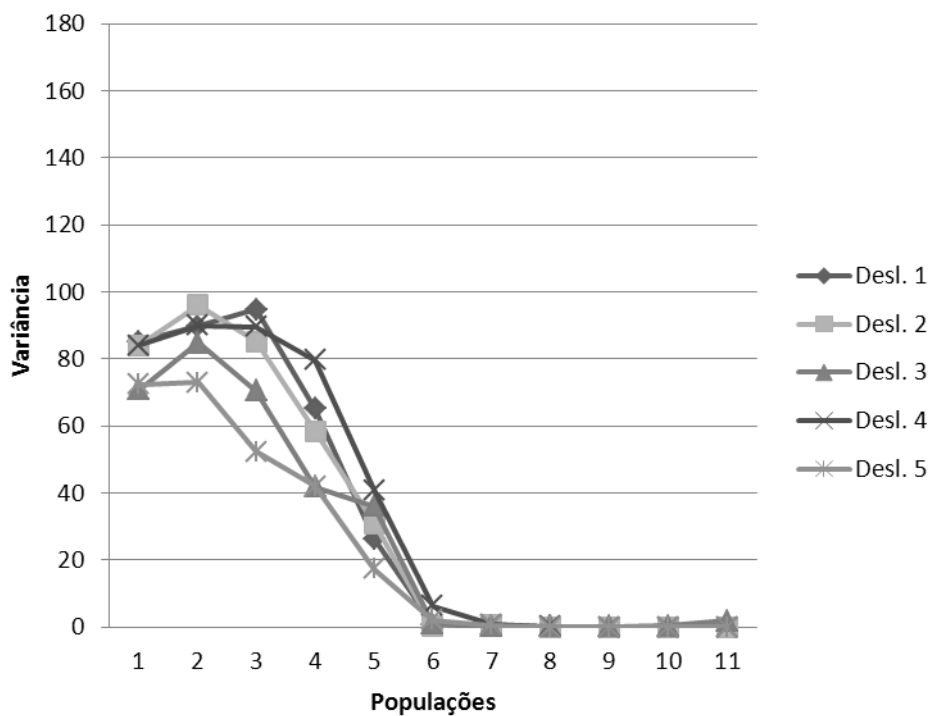


Figura 5.6 - Diversificação para Teste 8 com codificação INTEIRA

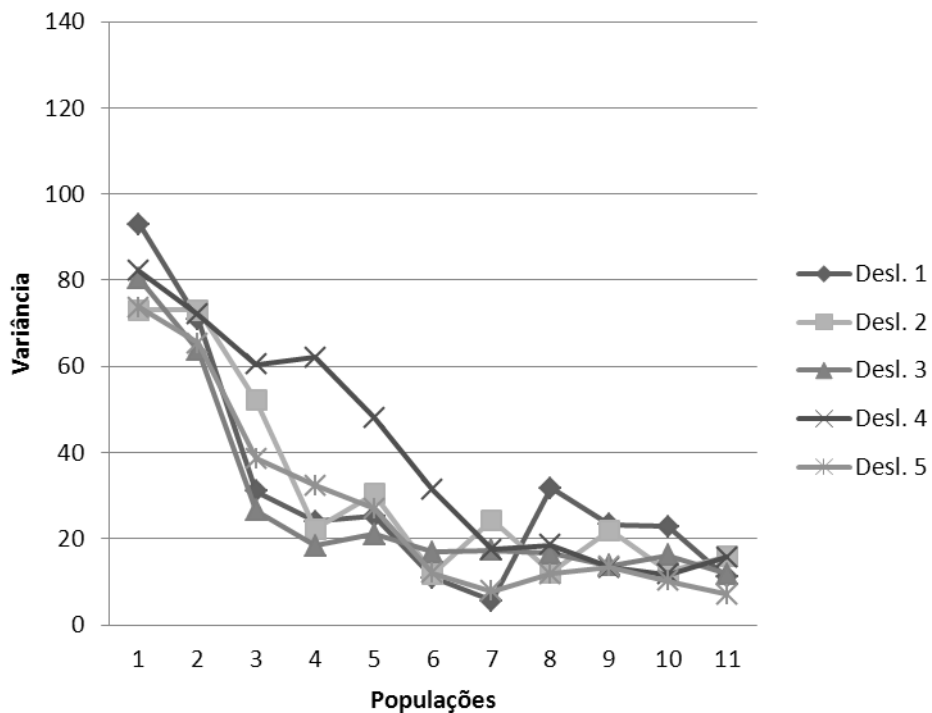


Figura 5.7 - Diversificação para Teste 12 com codificação GRAY

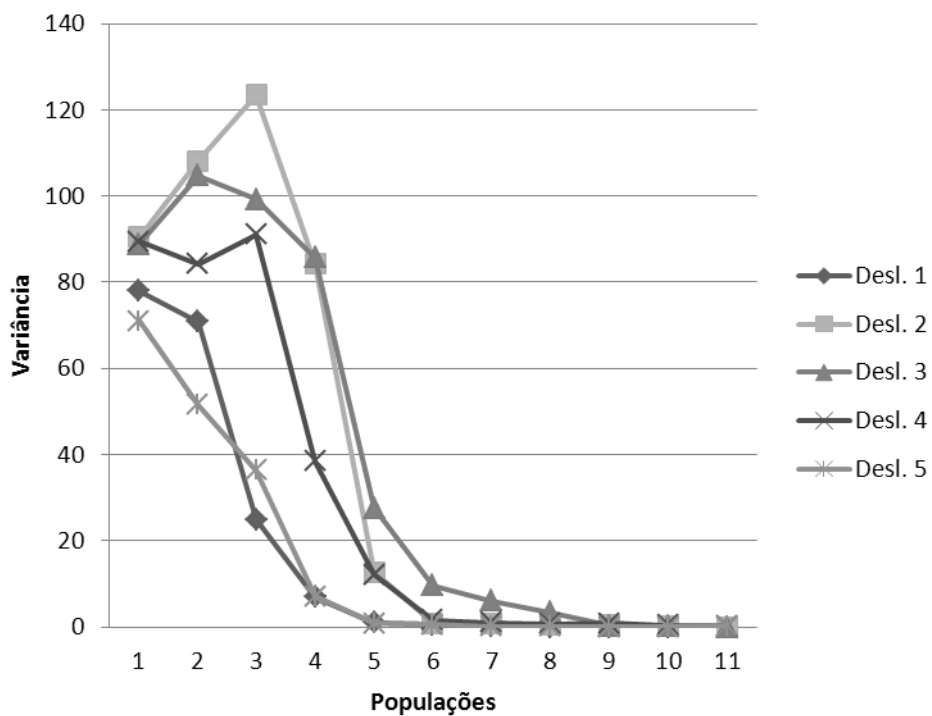


Figura 5.8 - Diversificação para Teste 12 com codificação INTEIRA

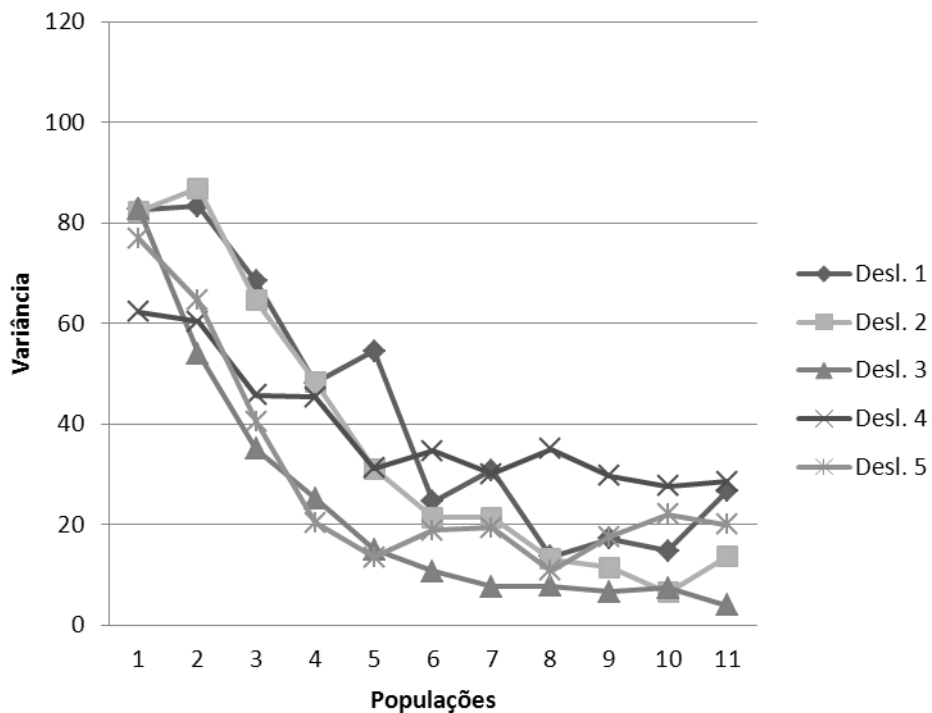


Figura 5.9 - Diversificação para Teste 13 com codificação GRAY

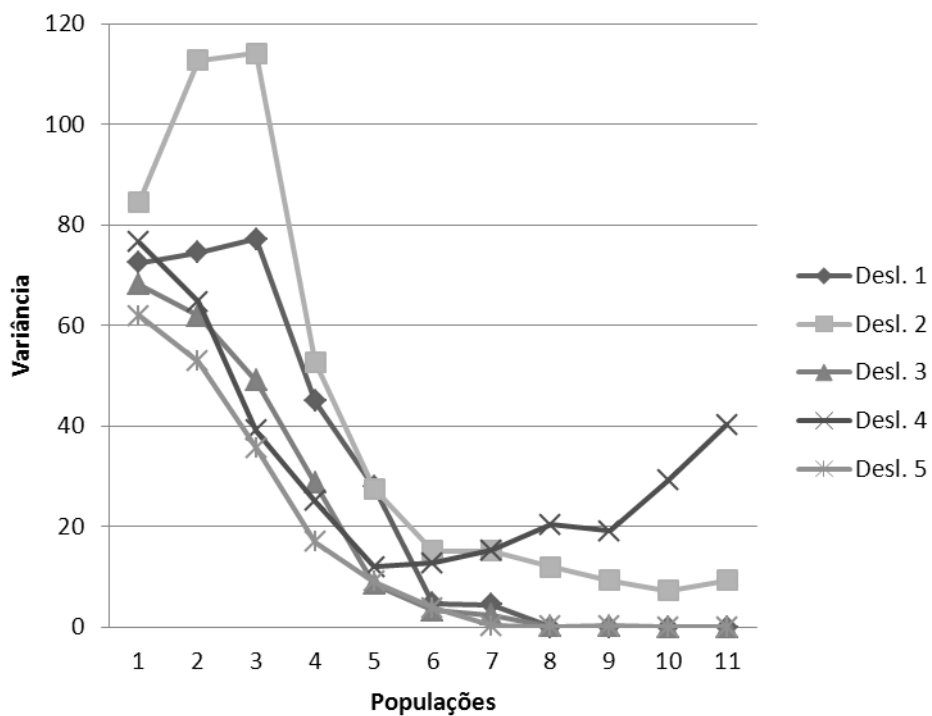


Figura 5.10 - Diversificação para Teste 13 com codificação INTEIRA

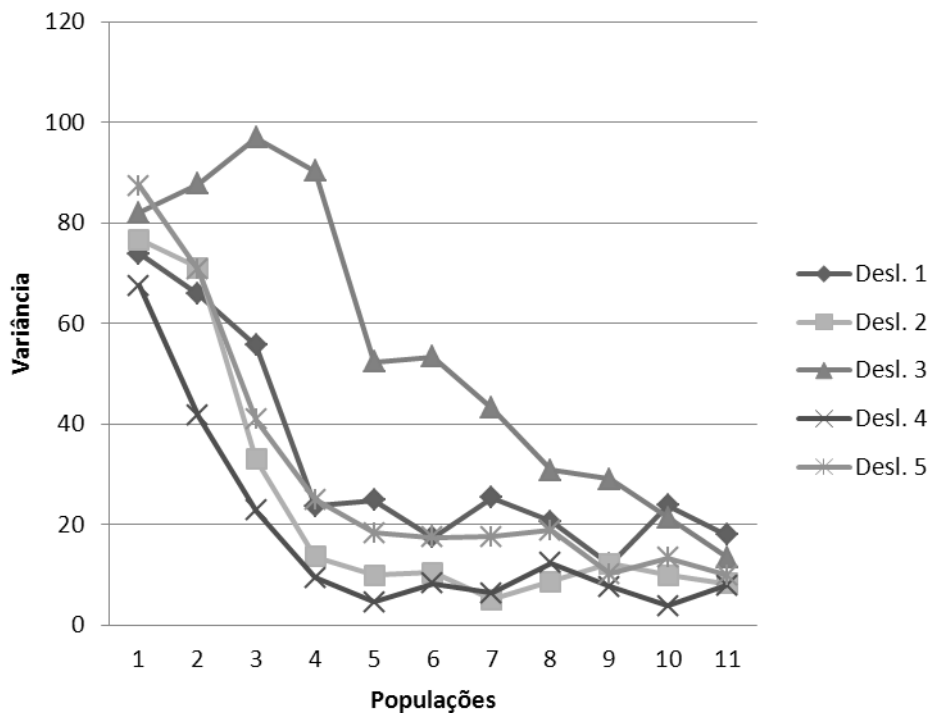


Figura 5.11 - Diversificação para Teste 14 com codificação GRAY

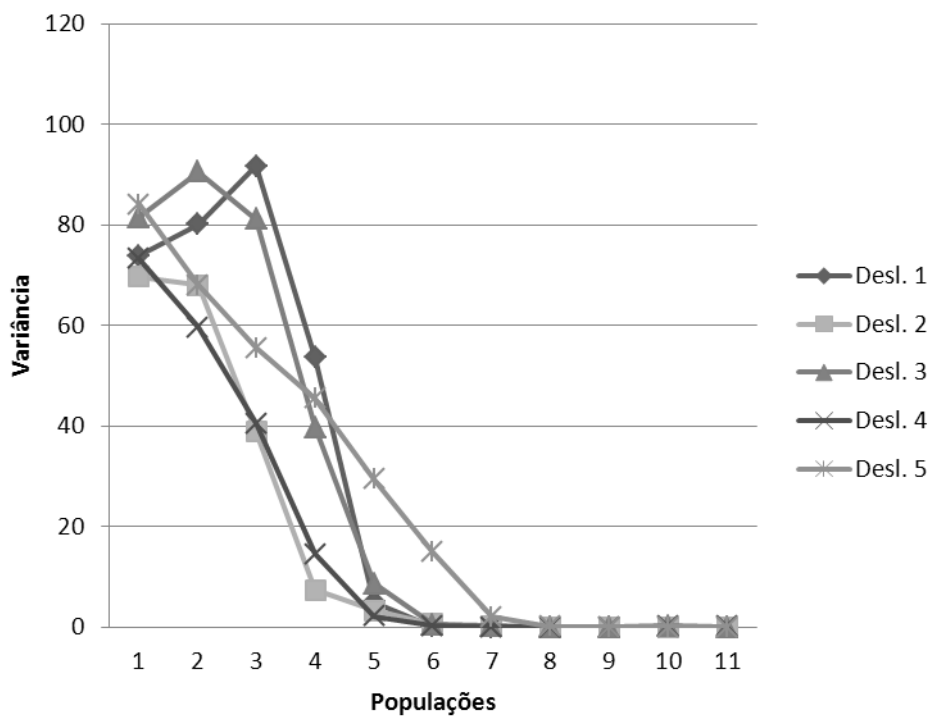


Figura 5.12 - Diversificação para Teste 14 com codificação INTEIRA

Análise do Tamanho do Conjunto Elite

A análise comparativa da capacidade de diversificação do AG proposto, considerando as estratégias empregadas e as codificações testadas, foi realizada também com base nas quantidades de indivíduos selecionados pelo emprego da estratégia de Conjunto Elite ao longo das 70 repopulações. Como pode ser visto da Figura 5.13 até a Figura 5.18, o número de indivíduos selecionados para o Conjunto Elite permaneceu expressivo quando se utilizou a codificação GRAY, enquanto poucos foram selecionados quando se empregou a codificação INTEIRA. Este resultado corrobora a capacidade de diversificação proporcionada pela codificação GRAY no problema tratado, revelando ainda que tal diversidade é também encontrada entre os indivíduos mais aptos. Isto mostra que o emprego da codificação GRAY permite uma exploração mais ampla do espaço de busca e a existência de um maior número de indivíduos muito aptos e significativamente diferentes indica a eficácia da exploração de diferentes regiões do espaço de solução.

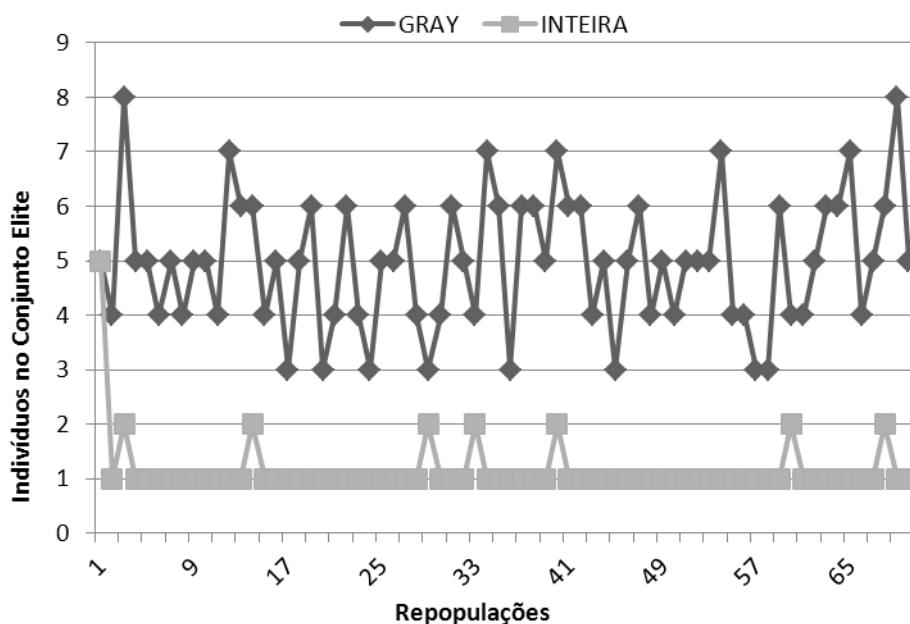


Figura 5.13 - Indivíduos no Conjunto Elite Teste 6 GRAY x INTEIRA

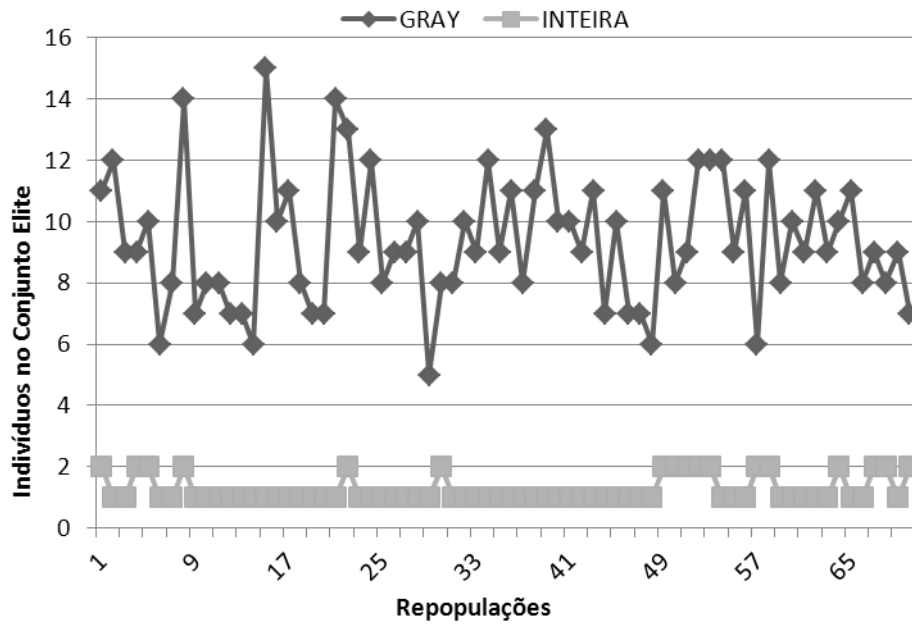


Figura 5.14 - Indivíduos no Conjunto Elite Teste 7 GRAY x INTEIRA

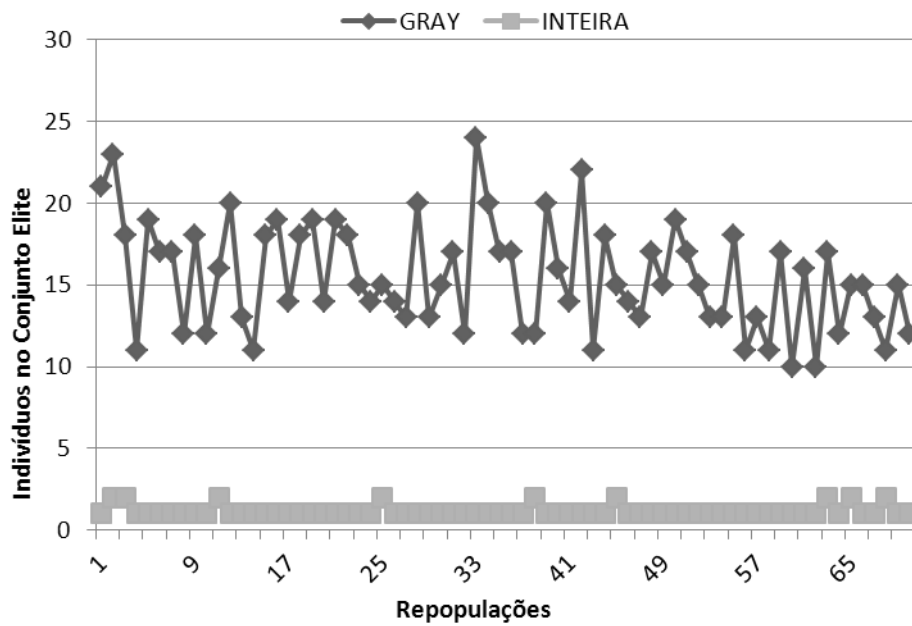


Figura 5.15 - Indivíduos no Conjunto Elite Teste 8 GRAY x INTEIRA

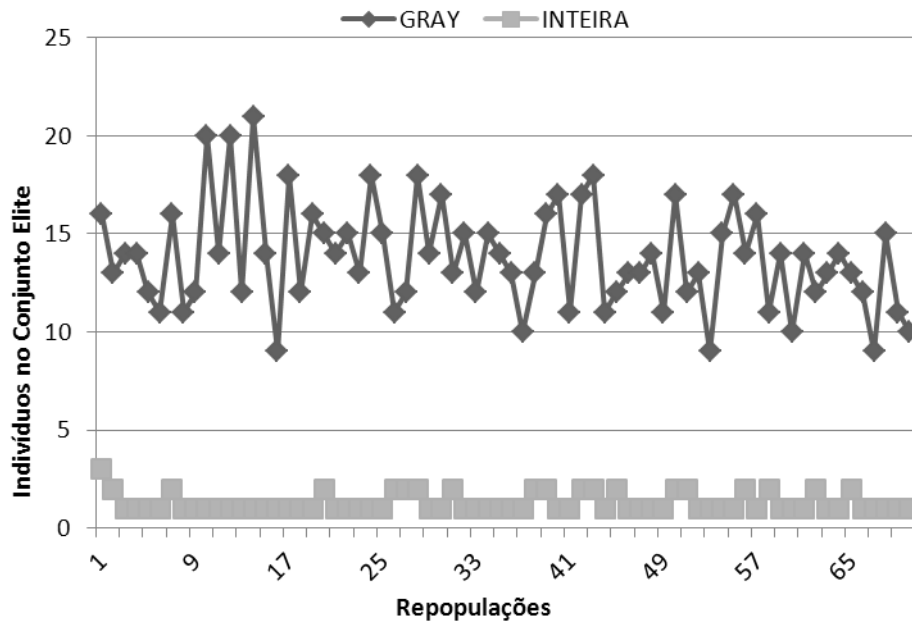


Figura 5.16 - Indivíduos no Conjunto Elite Teste 12 GRAY x INTEIRA

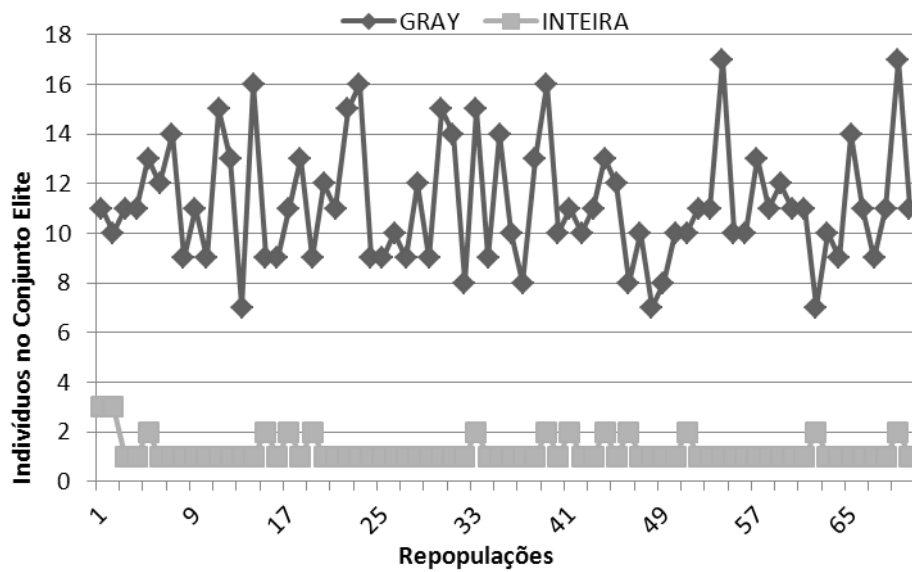


Figura 5.17 - Indivíduos no Conjunto Elite Teste 13 GRAY x INTEIRA

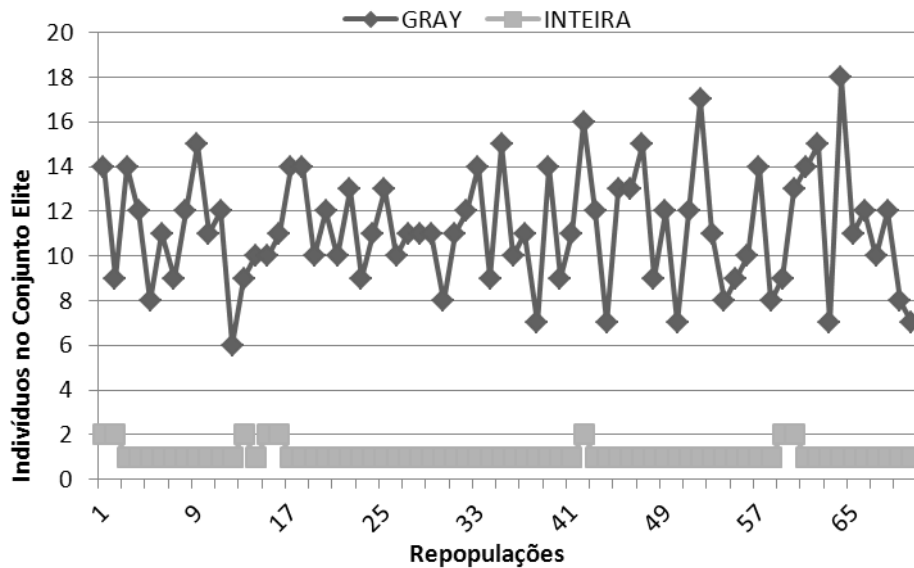


Figura 5.18 - Indivíduos no Conjunto Elite Teste 14 GRAY x INTEIRA

2) Análise da característica de convergência

A característica de convergência do AG proposto foi analisada com base em simulações considerando os desligamentos descritos nos testes 6, 7 e 8 de [Robe04] e para as seguintes situações:

- Emprego da codificação GRAY, sem considerar inequações de carregamento e formando o conjunto elite
- Emprego da codificação GRAY, considerando inequações de carregamento e formando o conjunto elite
- Emprego da codificação INTEIRA, considerando inequações de carregamento e formando o conjunto elite
- Emprego da codificação GRAY, considerando inequações de carregamento, sem formar o conjunto elite

Os testes foram realizados utilizando 50 sementes diferentes, geradas a partir de uma distribuição uniforme com cada um dos dois GNPA's implementados, perfazendo um total de 100 simulações por cada teste. Como critérios de comparação foram adotados para cada teste os melhores valores da função de aptidão obtidos, o valor médio da aptidão obtido, o

desvio padrão entre os valores de aptidão obtidos e a probabilidade acumulada para que o AG alcance a convergência em um número de gerações máximo. Os três primeiros critérios são descritos na Tabela 5.12 e na Tabela 5.13. O último critério será discutido adiante.

Na Tabela 5.12, as médias, mínimos, desvios padrão dos 50 valores de aptidão das soluções obtidas utilizando cada um dos dois GNPA's para cada teste são apresentados. Em cada linha da Tabela 5.12 está listado um teste que representa 50 execuções do mesmo AG para o mesmo problema, porém com diferentes sementes para o GNPA indicado (Mersenne Twister ou SGA). Estes resultados buscam ilustrar o quão díspares as soluções encontradas pelo AG foram, para cada um dos testes. Valores de desvios padrão baixos indicam que os valores de aptidão obtidos para a solução em cada teste variaram pouco.

Tabela 5.12 - Comparação das diferentes estratégias com diferentes GNPA's

Valores de Aptidão	SGA	MT	SGA	MT	SGA	MT
Teste	Média	Média	Mínimo	Mínimo	Desvio Padrão	Desvio Padrão
Teste6GRAYsemiNEQ	8,87643	8,87643	8,87643	8,87643	0,000	0,000
Teste6GRAYcomiNEQ	8,87643	8,87643	8,87643	8,87643	0,000	0,000
Teste6INTcomiNEQ	9,26548	9,30923	8,87643	8,87643	0,751	0,720
Teste6GRAYcomiNEQsemConjElite	8,88443	8,87643	8,87643	8,87643	0,057	0,000
Teste7GRAYsemiNEQ	28,67912	28,65046	28,65046	28,65046	0,203	0,000
Teste7GRAYcomiNEQ	28,65046	28,67912	28,65046	28,65046	0,000	0,203
Teste7INTcomiNEQ	29,26204	29,41870	28,65046	28,65046	0,825	0,775
Teste7GRAYcomiNEQsemConjElite	28,65046	28,65046	28,65046	28,65046	0,000	0,000
Teste8GRAYsemiNEQ	37,17676	37,24174	36,88378	36,88378	0,459	0,471
Teste8GRAYcomiNEQ	37,04378	37,18778	36,88378	36,88378	0,323	0,447
Teste8INTcomiNEQ	38,51881	38,47661	36,88378	36,88378	0,966	1,243
Teste8GRAYcomiNEQsemConjElite	37,31473	37,20574	36,88378	36,88378	0,566	0,515

Onde:

- SGA indica que o GNPA do SGA do KanGAL [Deb01] foi utilizado e
- MT indica que o GNPA Mersenne Twister [Mats11] foi utilizado.

Os resultados apresentados na Tabela 5.12 mostram que as diferenças oriundas do uso dos dois GNPA's não revelaram uma superioridade de um deles em relação ao outro.

Na Tabela 5.13, as médias, mínimos e desvios padrão dos 100 valores de aptidão das soluções obtidas utilizando os dois GNPA's para cada teste são apresentados. Nesta tabela as 50 amostras obtidas com cada GNPA são agrupadas em um mesmo conjunto TOTAL de 100 amostras. Cada amostra representa uma execução do mesmo AG, para o mesmo problema, porém com uma semente diferente no GNPA. Cada linha da tabela sumariza os resultados de um teste específico, tendo sido tais testes descritos na Seção 5.3.1. Valores baixos de desvio padrão indicam maior robustez do AG, visto que, nestes casos, o resultado apresentado pelo algoritmo variou pouco entre uma execução e outra para o mesmo tipo de teste.

Tabela 5.13 - Resultados comparativos para diferentes estratégias

Valores de Aptidão	TOTAL	TOTAL	TOTAL
Teste	Média	Mínimo	Desvio Padrão
Teste6GRAYsemINEQ	8,87643	8,87643	0,000
Teste6GRAYcomINEQ	8,87643	8,87643	0,000
Teste6INTcomINEQ	9,28735	8,87643	0,732
Teste6GRAYcomINEQsemConjElite	8,88043	8,87643	0,040
Teste7GRAYsemINEQ	28,66479	28,65046	0,143
Teste7GRAYcomINEQ	28,66479	28,65046	0,143
Teste7INTcomINEQ	29,34037	28,65046	0,800
Teste7GRAYcomINEQsemConjElite	28,65046	28,65046	0,000
Teste8GRAYsemINEQ	37,20925	36,88378	0,464
Teste8GRAYcomINEQ	37,11578	36,88378	0,395
Teste8INTcomINEQ	38,49771	36,88378	1,107
Teste8GRAYcomINEQsemConjElite	37,26024	36,88378	0,541

Os resultados apresentados na Tabela 5.12 e na Tabela 5.13 demonstram que os testes realizados com codificação GRAY apresentaram menor média e desvio padrão que aqueles nos quais se empregou a codificação INTEIRA. Foi possível observar também que a consideração da análise das inequações de carregamento nos ramos que formam os arranjos

topológicos de interesse não influenciou significativamente nos valores de aptidão obtidos nas melhores soluções geradas pelo AG, indicando que o mesmo conseguiu encontrar soluções que eliminaram possíveis violações em tais ramos.

A probabilidade acumulada para que o AG alcance a convergência em um número máximo de gerações foi avaliada utilizando a ferramenta *ttplots* [Aiex07]. Através desta ferramenta é possível avaliar a probabilidade do algoritmo alcançar um objetivo alvo (*target*) em um determinado tempo. Para cada caso testado, a ferramenta *ttplots* considerou um conjunto de dados de entrada contendo, para cada simulação realizada, o número de gerações em que o AG obteve a melhor aptidão. Esta análise busca comparar as diferentes estratégias utilizadas em cada teste, com base na probabilidade de encontrar uma boa solução em um baixo tempo computacional. Em tal análise, considerou-se que é baixo o desvio padrão entre os valores de aptidão encontrados em cada teste e que as soluções obtidas em todas as simulações foram de boa qualidade. Estas premissas podem ser verificadas na Tabela 5.12 e na Tabela 5.13.

Da Figura 5.19 até a Figura 5.24 estão ilustrados os resultados obtidos com a ferramenta *ttplots* [Aiex07], considerando os resultados das 100 simulações realizadas para cada teste. Como referência para a análise, as curvas posicionadas mais a esquerda indicam melhores resultados no tocante à probabilidade de se alcançar uma boa solução em um menor número de gerações.

Os resultados apresentados mostram que o emprego da codificação GRAY e da estratégia de formação de um conjunto elite influenciam positivamente a convergência do AG proposto.

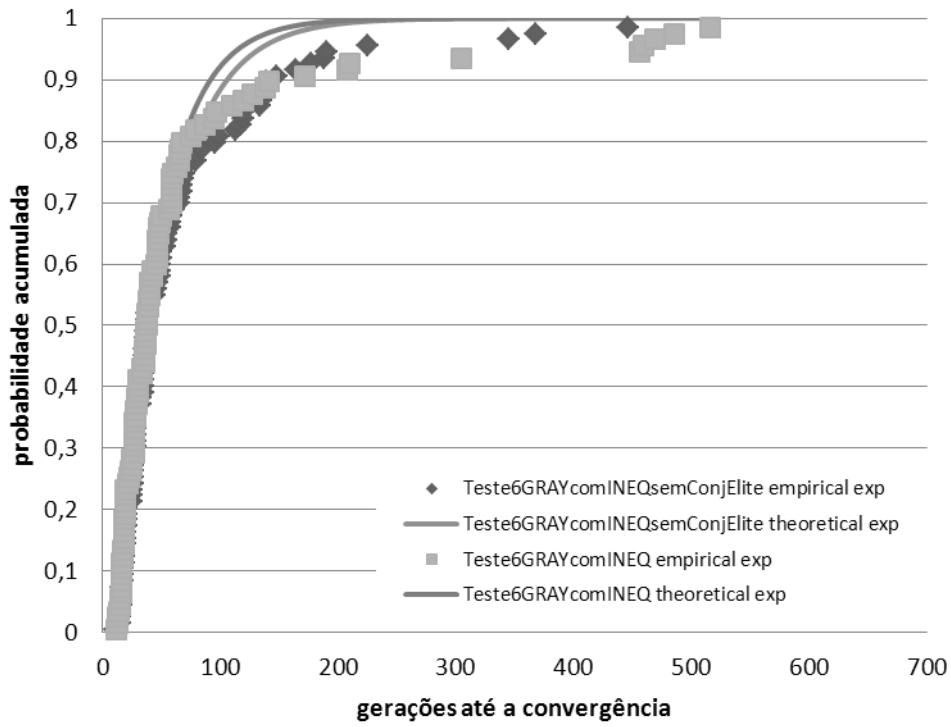


Figura 5.19 - Gráfico ttpplots para o Teste 6 - Conjunto Elite

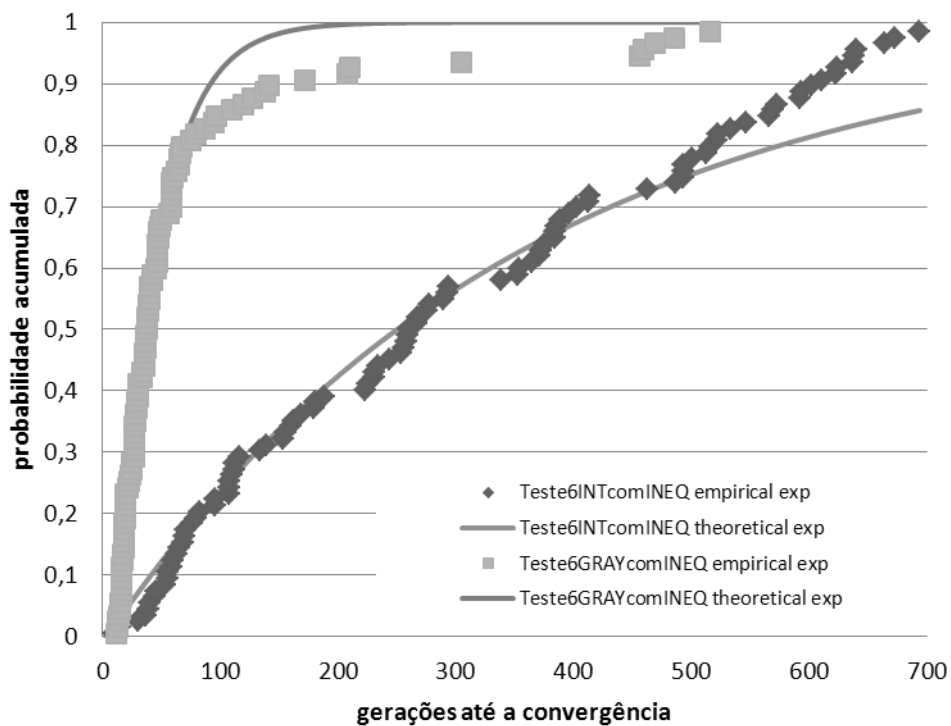


Figura 5.20 - Gráfico ttpplots para o Teste 6 - Codificação

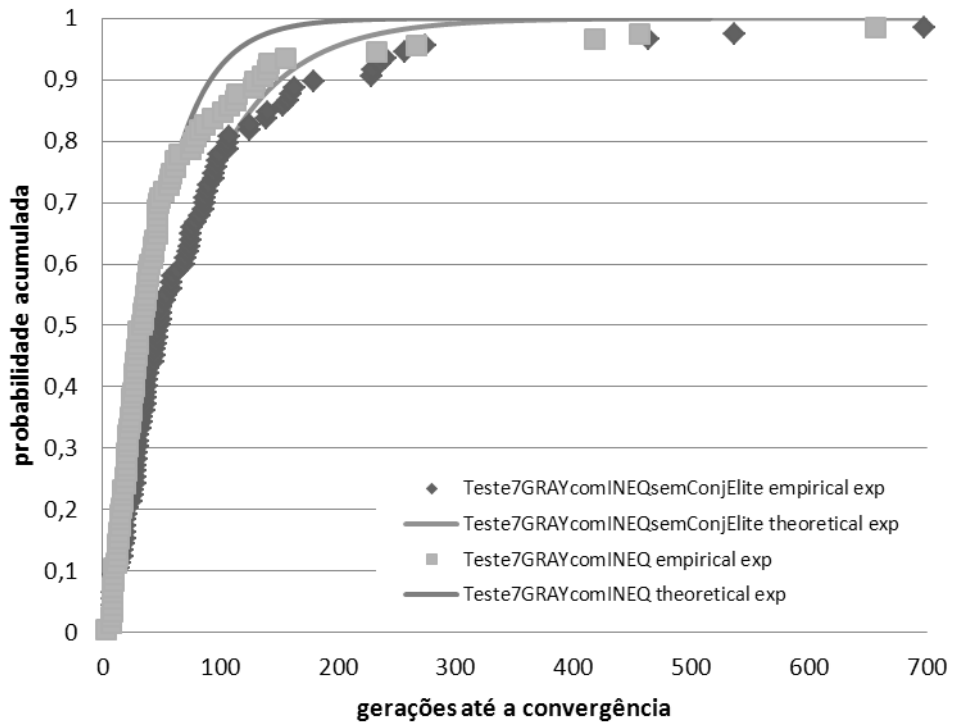


Figura 5.21 - Gráfico ttpplots para o Teste 7 - Conjunto Elite

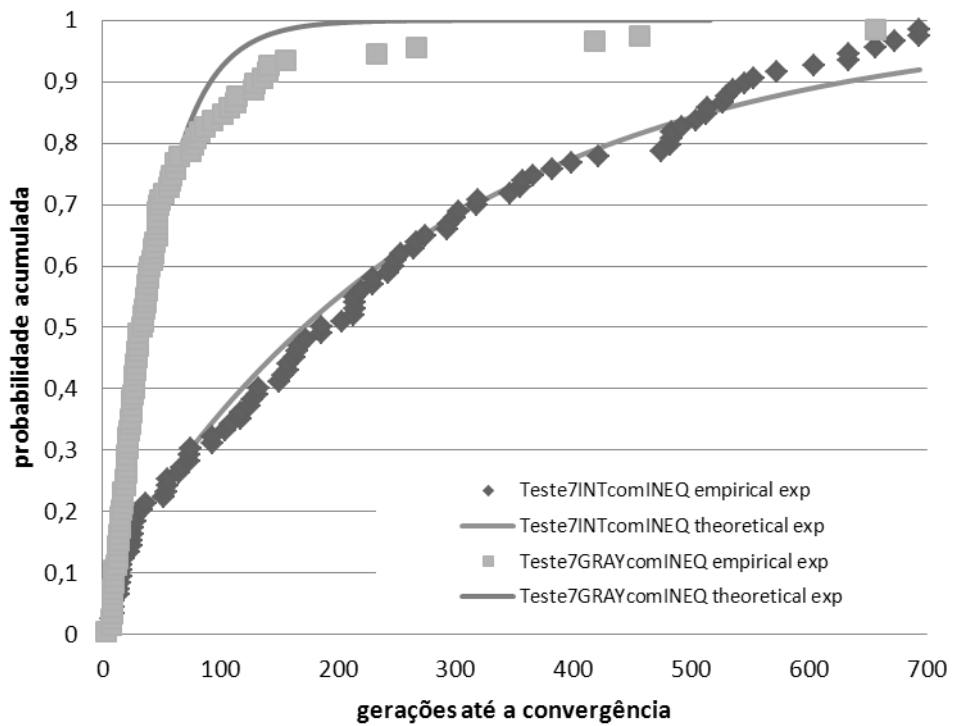


Figura 5.22 - Gráfico ttpplots para o Teste 7 - Codificação

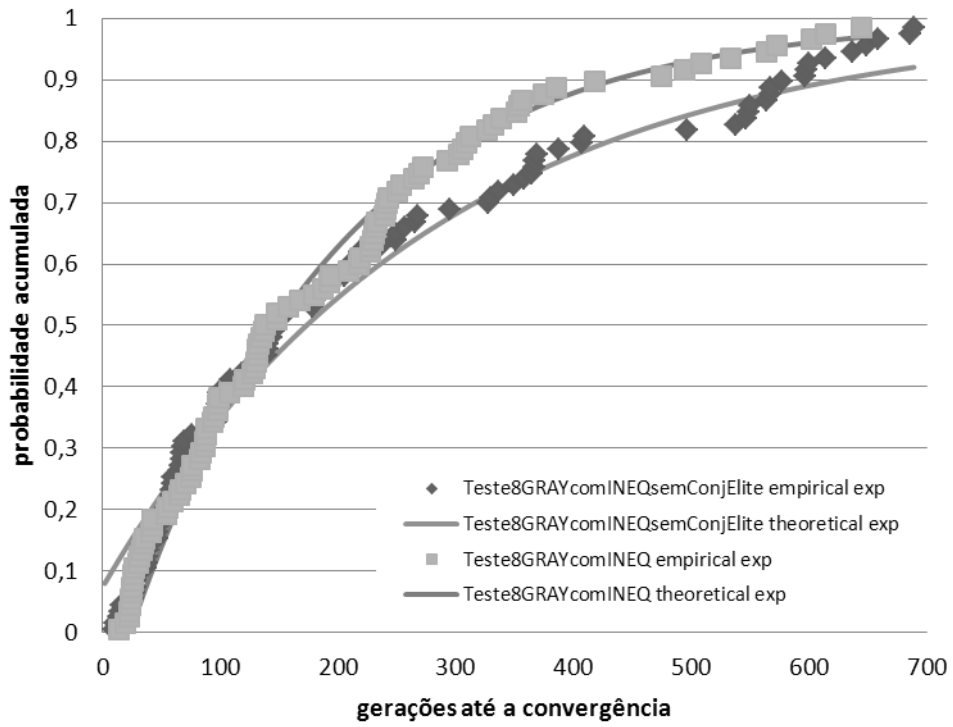


Figura 5.23 - Gráfico ttpplots para o Teste 8 - Conjunto Elite

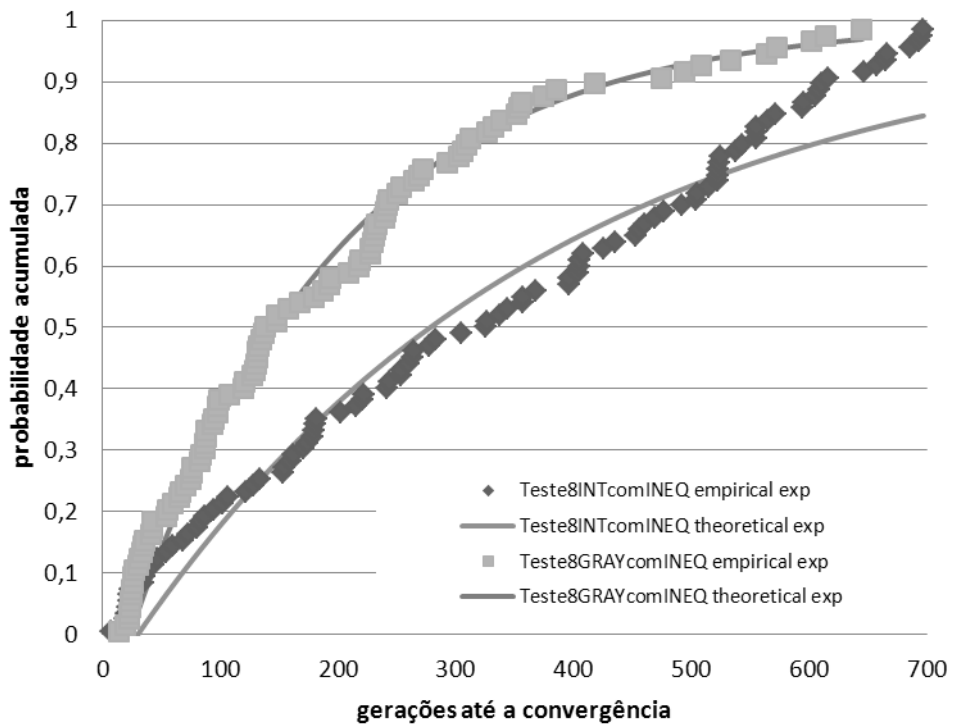


Figura 5.24 - Gráfico ttpplots para o Teste 8 - Codificação

5.4 TESTES CONSIDERANDO O SISTEMA IEEE 30 BARRAS

Os testes para o sistema IEEE 30 barras foram feitos considerando uma lista de contingências adotada para todos os testes, tendo sido mantida constante em todas as programações. Para a análise de inequações, foram utilizados os arranjos em anel 1-2-3-4 e 12-14-15, onde a influência da retirada das linhas 1-3 e 12-15 sobre as demais de cada arranjo era analisada. As estratégias de repopulação com elitismo a cada 10 gerações e a de conjunto elite automático foram adotadas em todos os testes, assim como o emprego da codificação GRAY.

As contingências utilizadas nos testes do sistema IEEE 30 barras são descritas na Tabela 5.14.

Tabela 5.14 - Lista de contingências dos testes com o sistema IEEE 30 barras

Contingência	Circuito Desligado	
	De	Para
1	2	4
2	12	15
3	15	18

Para análise dos resultados foram avaliados, para cada teste, as violações observadas e o valor de aptidão obtido para a programação original e para a otimizada considerando a função de aptidão formulada neste trabalho, bem como os desvios verificados entre a programação original e a otimizada. Também figuram nos resultados apresentados as violações da programação original e otimizada sem a presença de contingências, de modo a ilustrar a influência da inclusão de contingências na análise das programações. São apresentados nas tabelas de resultados apenas os ramos que apresentaram violações e os valores das maiores violações encontradas nestes ramos.

5.4.1 Teste para 5 desligamentos

Para o teste com 5 desligamentos foi utilizada a programação indicada da Tabela 5.15. Os resultados obtidos são apresentados na Tabela 5.16 e na Tabela 5.17.

Tabela 5.15 - Programação de 5 desligamentos – IEEE 30 Barras

Intervenção	Circuito Desligado		Programação		
	De	Para	Horário Inicial	Duração (h)	Prioridades
1	2	4	08:00	6	4
2	2	6	10:00	5	1
3	6	9	14:00	6	1
4	14	15	18:00	6	1
5	16	17	15:00	4	1

Tabela 5.16 - Programação Original x Otimizada - 5 desligamentos

Programação					
Original		Otimizada			
Horário Inicial	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
08:00	114,90	08:00	0	4,0	7,26
10:00		09:00	-1		
14:00		24:00	+10		
18:00		11:00	-7		
15:00		13:00	-2		

Tabela 5.17 - Resumo das violações (em %) para 5 desligamentos

Ramo	Sem Contingência		Contingência 1		Contingência 2		Contingência 3	
	Original	Otimizada	Original	Otimizada	Original	Otimizada	Original	Otimizada
2-4	20,75 ^l	-	-	-	20,83 ^l	-	20,68 ^l	-
6-10	32,39 ⁿ 0,49 ^e	-	22,31 ⁿ 2,44 ^e	-	6,00 ⁿ 56,79 ^e	4,42 ⁿ	27,57 ⁿ 16,96 ^e	3,81 ⁿ
12-15	15,51 ⁿ	-	11,57 ⁿ	-	-	-	-	-
12-14	-	-	-	-	77,61 ⁿ	5,44 ⁿ	-	-
12-16	-	-	-	-	82,36 ⁿ	3,59 ⁿ	18,65 ⁿ	-
14-15	-	-	-	-	32,75 ⁿ	-	-	-
28-27	-	-	-	-	31,82 ⁿ	-	-	-
10-20	-	-	-	-	23,01 ⁿ	4,52 ⁿ	-	-
6-28	-	-	-	-	16,08 ⁿ	-	-	-

^l – Inequações de Carregamento; ⁿ – Fluxo Nominal; ^e – Fluxo de Emergência

5.4.2 Teste para 6 desligamentos

Para o teste com 6 desligamentos foi utilizada a programação indicada da Tabela 5.18. Os resultados obtidos são apresentados na Tabela 5.19 e na Tabela 5.20.

Tabela 5.18 - Programação de 6 desligamentos – IEEE 30 Barras

Intervenção	Circuito Desligado		Programação		
	De	Para	Horário Inicial	Duração (h)	Prioridades
1	2	4	08:00	6	4
2	2	6	10:00	5	1
3	6	9	14:00	6	1
4	14	15	18:00	6	1
5	16	17	15:00	4	1
6	22	24	08:00	6	1

Tabela 5.19 - Programação Original x Otimizada - 6 desligamentos

Programação					
Original		Otimizada			
Horário Inicial	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
08:00	117,08	08:00	0	4,5	8,17
10:00		09:00	-1		
14:00		24:00	+10		
18:00		11:00	-7		
15:00		13:00	-2		
08:00		01:00	-7		

Tabela 5.20 - Resumo das violações (em %) para 6 desligamentos

Ramo	Sem Contingência		Contingência 1		Contingência 2		Contingência 3	
	Original	Otimizada	Original	Otimizada	Original	Otimizada	Original	Otimizada
2-4	20,75 ^f	-	-	-	20,83 ^f	-	20,68 ^f	-
6-10	32,39 ⁿ 0,49 ^e	-	22,31 ⁿ 2,44 ^e	-	6,00 ⁿ 56,79 ^e	4,42 ⁿ	27,57 ⁿ 16,96 ^e	3,81 ⁿ
12-15	15,51 ⁿ	-	11,57 ⁿ	-	-	-	-	-
12-14	-	-	-	-	77,61 ⁿ	5,44 ⁿ	-	-
12-16	-	-	-	-	82,36 ⁿ	3,59 ⁿ	18,65 ⁿ	-
14-15	-	-	-	-	32,75 ⁿ	-	-	-
28-27	-	-	-	-	31,82 ⁿ	-	-	-
10-20	-	-	-	-	23,01 ⁿ	4,52 ⁿ	-	-
6-28	-	-	-	-	16,08 ⁿ	-	-	-
16-17	-	-	-	-	56,17 ⁿ	-	-	-

^f – Inequações de Carregamento; ⁿ – Fluxo Nominal; ^e – Fluxo de Emergência

5.4.3 Teste para 7 desligamentos

Para o teste com 7 desligamentos foi utilizada a programação indicada da Tabela 5.21. Os resultados obtidos são apresentados na Tabela 5.22 e na Tabela 5.23.

Tabela 5.21 - Programação de 7 desligamentos – IEEE 30 Barras

Intervenção	Circuito Desligado		Programação		
	De	Para	Horário Inicial	Duração (h)	Prioridades
1	2	4	08:00	6	4
2	2	6	10:00	5	1
3	6	9	14:00	6	1
4	14	15	18:00	6	1
5	16	17	15:00	4	1
6	22	24	08:00	6	1
7	8	28	10:00	6	1

Tabela 5.22 - Programação Original x Otimizada - 7 desligamentos

Programação					
Original		Otimizada			
Horário Inicial	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
08:00	126,51	08:00	0	3,86	7,53
10:00		09:00	-1		
14:00		24:00	+10		
18:00		11:00	-7		
15:00		13:00	-2		
08:00		02:00	-6		
10:00		11:00	+1		

Tabela 5.23 - Resumo das violações (em %) para 7 desligamentos

Ramo	Sem Contingência		Contingência 1		Contingência 2		Contingência 3	
	Original	Otimizada	Original	Otimizada	Original	Otimizada	Original	Otimizada
2-4	20,75 ^l	-	-	-	20,82 ^l	-	20,68 ^l	-
6-10	32,39 ⁿ	-	22,31 ⁿ	-	6,03 ⁿ	4,42 ⁿ	27,57 ⁿ	3,81 ⁿ
	0,49 ^e	-	2,44 ^e	-	56,79 ^e	-	16,96 ^e	-
12-15	15,51 ⁿ	-	11,57 ⁿ	-	-	-	-	-
12-14	-	-	-	-	77,61 ⁿ	5,44 ⁿ	-	-
12-16	-	-	-	-	82,36 ⁿ	3,59 ⁿ	18,65 ⁿ	-
14-15	-	-	-	-	32,75 ⁿ	-	-	-
28-27	-	-	-	-	31,82 ⁿ	-	-	-
10-20	-	-	-	-	23,01 ⁿ	4,50 ⁿ	-	-
6-28	-	-	-	-	16,08 ⁿ	-	-	-
16-17	-	-	-	-	56,17 ⁿ	-	-	-

^l - Inequações de Carregamento; ⁿ - Fluxo Nominal; ^e - Fluxo de Emergência

5.4.4 Teste para 8 desligamentos

Para o teste com 8 desligamentos foi utilizada a programação indicada da Tabela 5.24. Os resultados obtidos são apresentados na Tabela 5.25 e na Tabela 5.26.

Tabela 5.24 - Programação de 8 desligamentos – IEEE 30 Barras

Intervenção	Circuito Desligado		Programação		
	De	Para	Horário Inicial	Duração (h)	Prioridades
1	2	4	08:00	6	4
2	2	6	10:00	5	1
3	6	9	14:00	6	1
4	14	15	18:00	6	1
5	16	17	15:00	4	1
6	22	24	08:00	6	1
7	8	28	10:00	6	1
8	27	29	14:00	5	1

Tabela 5.25 - Programação Original x Otimizada - 8 desligamentos

Programação					
Original		Otimizada			
Horário Inicial	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
08:00	127,79	08:00	0	3,75	7,44
10:00		09:00	-1		
14:00		24:00	+10		
18:00		10:00	-8		
15:00		12:00	-3		
08:00		02:00	-6		
10:00		10:00	0		
14:00		12:00	-2		

Tabela 5.26 - Resumo das violações (em %) para 8 desligamentos

Ramo	Sem Contingência		Contingência 1		Contingência 2		Contingência 3	
	Original	Otimizada	Original	Otimizada	Original	Otimizada	Original	Otimizada
2-4	20,86 ^f	-	-	-	20,94 ^f	-	20,79 ^f	-
6-10	32,59 ⁿ 0,65 ^e	-	22,31 ⁿ 2,60 ^e	-	6,16 ⁿ 57,10 ^e	4,42 ⁿ	27,57 ⁿ 17,15 ^e	3,81 ⁿ
12-15	15,77 ⁿ	-	11,83 ⁿ	-	-	-	-	-
12-14	-	-	-	-	77,97 ⁿ	5,44 ⁿ	-	-
12-16	-	-	-	-	82,36 ⁿ	3,59 ⁿ	18,65 ⁿ	-
14-15	-	-	-	-	33,09 ⁿ	-	-	-
28-27	-	-	-	-	32,32 ⁿ	-	-	-
10-20	-	-	-	-	23,01 ⁿ	4,50 ⁿ	-	-
6-28	-	-	-	-	16,45 ⁿ	-	-	-
16-17	-	-	-	-	56,17 ⁿ	-	-	-

^f – Inequações de Carregamento; ⁿ – Fluxo Nominal; ^e – Fluxo de Emergência

5.4.5 Teste para 9 desligamentos

Para o teste com 9 desligamentos foi utilizada a programação indicada da Tabela 5.27. Os resultados obtidos são apresentados na Tabela 5.28 e na Tabela 5.29.

Tabela 5.27 - Programação de 9 desligamentos – IEEE 30 Barras

Intervenção	Circuito Desligado		Programação		
	De	Para	Horário Inicial	Duração (h)	Prioridades
1	2	4	08:00	6	4
2	2	6	10:00	5	1
3	6	9	14:00	6	1
4	14	15	18:00	6	1
5	16	17	15:00	4	1
6	22	24	08:00	6	1
7	8	28	10:00	6	1
8	27	29	14:00	5	1
9	1	4	18:00	4	1

Tabela 5.28 - Programação Original x Otimizada - 9 desligamentos

Programação					
Original		Otimizada			
Horário Inicial	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
08:00	130,81	08:00	0	4,0	7,69
10:00		09:00	-1		
14:00		24:00	+10		
18:00		10:00	-8		
15:00		12:00	-3		
08:00		02:00	-6		
10:00		10:00	0		
14:00		12:00	-2		
18:00		12:00	-6		

Tabela 5.29 - Resumo das violações (em %) para 9 desligamentos

Ramo	Sem Contingência		Contingência 1		Contingência 2		Contingência 3	
	Original	Otimizada	Original	Otimizada	Original	Otimizada	Original	Otimizada
2-4	20,86 ^l	-	-	-	20,94 ^l	-	20,79 ^l	-
6-10	32,59 ⁿ	-	22,31 ⁿ	-	6,16 ⁿ	4,42 ⁿ	27,57 ⁿ	3,81 ⁿ
	0,65 ^e	-	2,60 ^e	-	57,10 ^e	-	17,15 ^e	-
12-15	15,78 ⁿ	-	11,83 ⁿ	-	-	-	-	-
12-14	-	-	-	-	77,97 ⁿ	5,44 ⁿ	-	-
12-16	-	-	-	-	82,36 ⁿ	3,59 ⁿ	18,65 ⁿ	-
14-15	-	-	-	-	33,09 ⁿ	-	-	-
28-27	-	-	-	-	32,32 ⁿ	-	-	-
10-20	-	-	-	-	23,01 ⁿ	4,52 ⁿ	-	-
6-28	-	-	-	-	16,45 ⁿ	-	-	-
16-17	-	-	-	-	56,17 ⁿ	-	-	-

^l – Inequações de Carregamento; ⁿ – Fluxo Nominal; ^e – Fluxo de Emergência

5.4.6 Teste para 10 desligamentos

Para o teste com 10 desligamentos foi utilizada a programação indicada da Tabela 5.30. Os resultados obtidos são apresentados na Tabela 5.31 e na Tabela 5.32.

Tabela 5.30 - Programação de 10 desligamentos – IEEE 30 Barras

Intervenção	Circuito Desligado		Programação		
	De	Para	Horário Inicial	Duração (h)	Prioridades
1	2	4	08:00	6	4
2	2	6	10:00	5	1
3	6	9	14:00	6	1
4	14	15	18:00	6	1
5	16	17	15:00	4	1
6	22	24	08:00	6	1
7	8	28	10:00	6	1
8	27	29	14:00	5	1
9	1	4	18:00	4	1
10	15	18	15:00	5	1

Tabela 5.31 - Programação Original x Otimizada - 10 desligamentos

Programação					
Original		Otimizada			
Horário Inicial	Aptidão	Horário Inicial	Desvio (h)	Desvio Médio (h)	Aptidão
08:00	136,86	08:00	0	2,5	6,35
10:00		09:00	-1		
14:00		24:00	+10		
18:00		14:00	-4		
15:00		16:00	+1		
08:00		02:00	-6		
10:00		10:00	0		
14:00		15:00	+1		
18:00		16:00	-2		
15:00		15:00	0		

Tabela 5.32 - Resumo das violações (em %) para 10 desligamentos

Ramo	Sem Contingência		Contingência 1		Contingência 2		Contingência 3	
	Original	Otimizada	Original	Otimizada	Original	Otimizada	Original	Otimizada
2-4	20,86 ^l	-	-	-	20,94 ^l	-	20,79 ^l	-
6-10	32,59 ⁿ 0,65 ^e	-	30,69 ⁿ 18,47 ^e	-	6,16 ⁿ 57,72 ^e	4,42 ⁿ	27,57 ⁿ 17,15 ^e	3,81 ⁿ
12-15	15,78 ⁿ	-	-	-	-	-	-	-
12-14	-	-	-	-	55,20 ⁿ	5,44 ⁿ	-	-
12-16	-	-	14,04 ⁿ	-	82,19 ⁿ	3,59 ⁿ	18,65 ⁿ	-
14-15	-	-	-	-	11,16 ⁿ	-	-	-
28-27	-	-	-	-	31,84 ⁿ	-	-	-
10-20	-	-	-	-	23,01 ⁿ	4,27 ⁿ	-	-
6-28	-	-	-	-	16,06 ⁿ	-	-	-
16-17	-	-	-	-	56,00 ⁿ	-	-	-
22-24					-	5,56 ⁿ		

^l – Inequações de Carregamento; ⁿ – Fluxo Nominal; ^e – Fluxo de Emergência

5.4.7 Comentários sobre os resultados com o sistema IEEE 30 barras

Para os testes com 5, 6, 7, 8, 9 e 10 desligamentos no sistema IEEE 30 barras, o AG Proposto atendeu às restrições impostas, minimizando desvios da nova programação com base na prioridade definida, mantendo um desvio mínimo na intervenção com prioridade 4 da programação. Para a codificação GRAY, utilizando a estratégia de repopulação com elitismo após 10 gerações e conjunto elite automático, foi possível alcançar as soluções anteriormente descritas no número de gerações indicado na Tabela 5.33.

Tabela 5.33 - Número de gerações até a convergência – Testes IEEE 30 Barras

Número de Desligamentos Solicitados	Número de Gerações até a convergência
5	8
6	17
7	25
8	19
9	43
10	99

Assim como nos testes para o sistema IEEE 14 barras, o número de gerações descritos na Tabela 5.33 para o AG proposto considera uma contagem absoluta de gerações desde a inicialização do AG e a execução da estratégia de repopulação com elitismo a cada grupo de 10 gerações não reinicializa este contador.

Foram registrados os tempos de execução das 700 gerações do AG proposto para cada um dos testes realizados no sistema IEEE 30 barras descritos na Seção 5.4. Os resultados destes testes são demonstrados na Tabela 5.34, onde o valor em segundos corresponde ao tempo total de execução de cada teste considerando as 700 gerações definidas para a execução do AG. A plataforma computacional utilizada para os testes possuía processador Intel Core 2 Duo CPU 8400 @ 2.26GHz em sistema operacional Windows XP Professional SP3. Vale ressaltar que, em todos os testes, foi utilizada a codificação GRAY em conjunto com a estratégia de repopulação com elitismo após 10 gerações e conjunto elite automático.

Tabela 5.34 - Tempo de execução das 700 gerações – Testes IEEE 30 Barras

Número de Desligamentos Solicitados	Tempo de execução das 700 gerações (s)
5	1342
6	1503
7	1626
8	1791
9	1880
10	1991

Nos resultados apresentados pode-se notar que a programação otimizada obtida em todos os testes eliminou completamente as violações existentes quando as contingências simples não eram consideradas. Em todos os casos, a programação otimizada eliminou ou reduziu expressivamente as violações, mesmo considerando a lista de contingências proposta. Também é importante ressaltar que todas as violações de limites

de emergência foram eliminadas nas programações otimizadas obtidas. Na análise da segunda contingência do teste de 10 desligamentos, uma nova violação de limite nominal de fluxo de potência ativa de 5,56% para o ramo 22-24 foi gerada na programação otimizada, devido à severidade da análise desta contingência na programação de desligamentos desejada para aquele teste.

5.5 CONCLUSÕES

Este capítulo apresentou resultados da aplicação da metodologia proposta à reprogramação de desligamentos nos sistemas teste IEEE 14 e 30 barras. As estratégias propostas no AG foram testadas, bem como o emprego de diferentes codificações para as variáveis do problema, tendo sido analisada a sua influência sobre o processo de otimização.

Os resultados obtidos revelaram a capacidade da metodologia proposta tratar o problema de programação de desligamentos como um problema de otimização com restrições, levando a soluções de alta qualidade. Quando comparada a resultados encontrados na literatura, tal metodologia apresentou soluções de melhor qualidade, obtidas a um custo computacional mais baixo. Destacam-se os benefícios encontrados com as estratégias de repopulação e de formação do conjunto elite, bem como do emprego da codificação GRAY.

CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma metodologia para a avaliação de solicitações de desligamentos de elementos em redes de transmissão de energia elétrica. Tal metodologia se baseia no emprego de um AG, tendo sido exploradas diferentes formas de codificação da solução. Além disso, foi empregada uma estratégia de repopulação de gerações, com elitismo, o que gerou uma melhor exploração do espaço de soluções. Com o emprego da codificação Gray foi possível também obter uma maior diversificação de indivíduos, o que impede uma convergência prematura.

As programações de desligamentos obtidas em testes realizados apresentaram pequenos desvios em relação às solicitações originais, sendo atendidas todas as restrições, inclusive as inequações de sobrecarga em ramos especiais. Os operadores utilizados, a codificação adotada, a estratégia de elitismo entre gerações associada à repopulação permitiram encontrar soluções de alta qualidade e de baixo custo computacional, quando comparados com outros resultados da literatura.

Deve-se destacar que a metodologia proposta estabelece uma forma sistemática de tratar o problema da programação de desligamentos em redes de transmissão, visto como um problema de otimização.

Também foram apresentadas neste trabalho algumas formas de análise do processo evolutivo do AG. O uso da ferramenta *ttplots* e a análise gráfica de diversificação permitiram avaliar a influência de diferentes estratégias e codificações adotadas no AG.

O emprego da estratégia de conjunto elite automático apresentou resultados promissores quanto à prevenção de convergência prematura no AG proposto.

O trabalho desenvolvido nesta Dissertação valida e amplia os estudos realizados em [Robe04], demonstrando ser possível aperfeiçoamentos através da inclusão de novas hipóteses na formulação do problema e do emprego de novas técnicas / estratégias no processo de busca.

A utilização de outras heurísticas tais como Otimização por Enxame de Partículas, Estratégias Evolutivas ou ainda de ferramentas para Mineração de Dados pode ser considerada para aplicação em trabalhos futuros, tomando por base os resultados apresentados nesta Dissertação.

Considerando os resultados dos testes realizados, caso os desligamentos dos equipamentos vistos como contingências violem os limites atribuídos ao sistema analisado com qualquer desligamento programado escolhido, esta metodologia indica que não é possível eliminar tais violações apenas com a reprogramação dos desligamentos. Em trabalhos futuros, esta avaliação poderá ser incorporada à metodologia.

A presença de grandes penalidades para violações de demanda em algumas soluções pode deslocar a busca do AG para outras soluções sem estas violações, mas com aquelas (indesejáveis) de limites operativos. Em trabalhos futuros, propõe-se o emprego do redespacho de potência nestas soluções com violações de demanda e sem ultrapassagem de limites operativos na busca por melhores soluções com a metodologia proposta.

A implementação de uma ferramenta que seja capaz de apresentar as soluções encontradas durante o processo evolutivo do AG, em tempo de execução, pode acelerar ainda mais a tarefa de análise de uma programação ótima de desligamentos. Assim, seria desejável que os pesos e parâmetros pudessem ser ajustados durante a execução do AG, de modo a permitir que

o analista atue sobre o processo evolutivo de modo a alcançar uma programação que atenda às restrições desejadas.

O refino da estratégia do Conjunto Elite, com o uso de uma parametrização automática (baseada no número de desligamentos e no espalhamento das soluções no espaço de busca), pode ser também objeto de estudos futuros.

Por fim, indica-se para a continuidade desta pesquisa a inclusão de outros critérios de interesse para a análise dos desligamentos, tal como a análise da segurança (aspectos dinâmicos) dos cenários estudados.

REFERÊNCIAS

[Aiex07] R. M. Aiex, M.G.C. Resende and C.C. Ribeiro , “TTTTPLOTS: A Perl program to create time-to-target plots”, *Optimization Letters*, Vol.1, No.4, Springer Berlin / Heidelberg, pp. 355-366, 2007.

[Al-K92] T.M. Al-Khamis, S. Vemuri, L. Lemonidis and J. Yellen, “Unit Maintenance Scheduling with Fuel Constraints”, *IEEE Trans. on PWRS*, Vol.7, No.2, pp. 933-939, 1992.

[Ande02] G. Anders, G. Hamoud, A. M. Leite da Silva, and L. A. F. Manso, “Optimal outage scheduling – example of application to a large power system”, in Proc. 14th Power Syst. Comput. Conf. (PSCC), Seville, Spain, Session 41, paper 5, pp. 1-7, Jun. 2002.

[Bell68] J.R. Bell, “Algorithm 334: Normal random deviates”, *Communications of the ACM* 11, No. 7, p.498, Jul. 1968.

[Blum03] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison”, *ACM Computing Surveys*, Vol. 35, Issue 3, pp.268-308, Set. 2003.

[Box58] G.E.P. Box and M.E. Muller, “A Note on the Generation of Random Normal Deviates”, *The Annals of Mathematical Statistics (1958)*, Vol. 29, No. 2 pp. 610–611, 1958.

[Chak03] U.K. Chakraborty and C.Z. Janikow, “An analysis of Gray versus binary encoding in genetic search”, *Information Sciences*, 156(3-4), pp.253-269, 2003.

[Cham95] D.L. Chambers ed., “Practical handbook of genetic algorithms: new frontiers Vol. 2”, *CRC Press*, ISBN 0-8493-2529-3, 1995.

[Chat95] D. Chattopadhyay, K. Bhattacharya and J. Parikh, "A Systems Approach to Least-Cost Maintenance Scheduling for an Interconnected Power System", *IEEE Trans. on Power Systems*, vol. 10, 4, pp.2002-2007, 1995.

[Chen91] L. Chen and J. Toyoda, "Optimal Generating Unit Maintenance Scheduling for Multi-Area System with Network Constraints", *IEEE Trans. On Power Syst.*, vol. 6, pp.1168-1174, 1991.

[Chri99] R.D. Christie, "Power Systems Test Case Archive" [online], Disponível em: <http://www.ee.washington.edu/research/pstca/>, [acessado em 28/04/2011], 1999.

[Coel00] C.A. Coello, "An updated survey of GA-based multiobjective optimization techniques", *ACM Computing Surveys*, Vol. 32, Issue 2, pp.109-143, Jun. 2000.

[Cont89] G.C. Contaxis, S.D. Kavatza and C.D. Vournas, "An Interactive Package for Risk Evaluation and Maintenance Scheduling", *IEEE Trans. On Power Syst.*, vol. 4, 2, pp.389-395, 1989.

[De J07] K. De Jong, "Parameter Setting in EAs: a 30 Year Perspective", *In: Studies in Computational Intelligence (SCI) 54, 2007, Springer-Verlag Berlin Heidelberg*, pp.1-18, 2007.

[Deb94] K. Deb and R.B. Agrawal , "Simulated Binary Crossover for Continuous Search Space", *Complex Systems*, Vol. 9, pp. 1-34, 1994.

[Deb96] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design", *Computer Science and Informatics*, Vol. 26, pp.30-45, 1996.

[Deb01] K. Deb, "Simple GA code in C (for Windows and Linux)", Disponível em <http://www.iitk.ac.in/kangal/codes/sga/sga.tar>, [Acessada em março de 2011], 2001.

[Dopa89] J.F. Dopazo and H.M. Merrill, "Optimal Generator Maintenance Scheduling using Integer Programming", *IEEE Trans. on PA Syst.*, vol. 4, 2, pp.389-395, 1989.

[Egan76] G.T. Egan, T.S. Dillon and K. Morsztyn, "An Experimental Method of Determination of Optimal Maintenance Schedules in Power Systems Using the Branch and Bound Technique", *IEEE Trans. Man & Cybs*, Vol. SMC-6, No. 8, pp.1269-1278, 1976.

[El-S84] F.A. El-Sheikhi and R. Billinton, "Generating Unit Maintenance Scheduling for Single and Two Interconnected Systems", *IEEE Trans. on PAS*, vol. PAS-103, 5, pp.1038-1044, 1984.

[Eshe91] L.J. Eshelman, "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination", In: Gregory J. E. Rawlins ed., *Proceedings of the First Workshop on Foundations of Genetic Algorithms*. pp. 265-283. *Morgan Kaufmann*, ISBN 1-55860-170-8, 1991.

[Eshe92] L.J. Eshelman and J.D. Schaffer, "Real-Coded Genetic Algorithms and Interval-Schemata", in *Proc. FOGA*, pp.187-202, 1992.

[Fu09] Y. Fu, Z. Li, M. Shahidehpour, T. Zheng, and E. Litvinov, "Coordination of midterm outage scheduling with short-term security-constrained unit commitment", *IEEE Trans. Power Syst.*, vol. 24, no. 4, pp. 1818-1830, Nov. 2009.

[Garv72] L.L. Garver, "Adjusting Maintenance Schedules to Levelize Risk", *IEEE Trans. on PAS*, vol. PAS-91, pp. 2057-2063, 1972.

[Gold90] D.E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms", *In Proceedings of FOGA. 1990*, pp.69-93, 1990.

[Gold92] D.E. Goldberg, K. Deb and J.H. Clark, "Genetic Algorithms, Noise, and the Sizing of Populations", *Complex Systems*, Vol. 6, pp.333-362, 1992.

[Gref86] J. J. Grefenstette , "Optimization of control parameters for genetic algorithms", *IEEE Trans. Systems, Man, and Cybernetics*, SMC-16(1), pp.122-128, 1986.

[Haupt04] R.L. Haupt and S.E. Haupt, "Practical Genetic Algorithms", 2nd ed, *New York: Wiley*, ISBN 0-471-45565-2, 2004.

[Juan97] J. Juan and I. Ortega, "Reliability Analysis for Hydrothermal Generating Systems Including the Effect of Maintenance Scheduling", *IEEE Trans. on Power Syst.*, vol. 12, 4, pp.1561-1568, 1997.

[KanG05] Kanpur Genetic Algorithms Laboratory (KanGAL), "Real-coded Genetic Algorithms", [online], Disponível em : <http://www.iitk.ac.in/kangal/resources.shtml#2>, [Acessado em janeiro de 2011], 2005.

[Kenn05] C. Kenny , "Random Number Generators: An Evaluation and Comparison of Random.org and Some Commonly Used Generators", *Management Science and Information Systems Studies Project Report, TRINITY COLLEGE DUBLIN*, [pdf]. Disponível em: <http://www.random.org/analysis/Analysis2005.pdf>, [Acessado em julho de 2011], 2005.

[Kim97] H. Kim, Y. Hayashi, and K. Nara, "An algorithm for thermal unit maintenance scheduling through combined use of GA, SA, and TS", *IEEE Trans. Power Syst.*, vol. 12, no. 1, pp.329-335, Fev. 1997.

[Knop69] R. Knop, "Remark on algorithm 334 [G5]: normal random deviates", *Communications of the ACM*, vol. 12, No. 5., 1969.

[Knot10] R. Knott, "Fibonacci Numbers and Nature", Disponível em: <http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/fibnat.html>, [Acessado em julho de 2011], 2010.

[Knut97] D.E. Knuth, "The Art of Computer Programming, Vol 2, Seminumerical Algorithms", Third Edition, *Reading, Massachusetts: Addison-Wesley*, ISBN 0-201-89684-2, 1997.

[Koza95] J.R. Koza, "Survey of Genetic Algorithms and Genetic Programming", In: In Proceedings of the Wescon 95 - Conference Record: Microelectronics, Communications Technology, Producing Quality Products, Mobile and Portable Power, Emerging Technologies, *IEEE Press*, pp. 589-594, 1995.

[Leun03] K.S. Leung and Y. Liang, "Adaptive Elitist-Population Based Genetic Algorithm for Multimodal Function Optimization", In: Cantú-Paz, E. et al. Eds., GECCO 2003, Lecture Notes in Computer Science Vol. 2723, *Springer-Verlag Berlin Heidelberg*, pp.1160-1171, 2003.

[Luke09] S. Luke, "Essentials of Metaheuristics" [pdf], Disponível em: <http://cs.gmu.edu/~sean/book/metaheuristics/>, [Acessado em setembro de 2009], 2009.

[Mats98] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator", *ACM Transactions on Modeling and Computer Simulation (TOMACS)* - Special issue on uniform random number generation Volume 8 Issue 1, Jan. 1998.

[Mats11] M. Matsumoto, “Mersenne Twister Home Page”, Disponível em: <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html>, [Acessada em julho de 2011], 2011.

[Merz99] P. Merz and B. Freisleben, “A Comparison of Memetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem”, In *1999 Congress on Evolutionary Computation (CEC'99)*, IEEE Press, Piscataway, NJ, pp.2063–2070, 1999.

[Mich96] Z. Michalewicz, “Genetic Algorithms + Data Structures = Evolution programs”, Third Edition, *New York: Springer*, ISBN 3-540-60676-9, 1996.

[Mitc99] M. Mitchell, “An Introduction to Genetic Algorithms”, *Bradford Book The MIT Press*, ISBN 0-262-63185-7, 1999.

[Muke91] R. Mukerji, H.M. Merrill, R. Erickson, J.H. Parker and R.E. Freidman, “Power Plant Maintenance Scheduling: Optimizing Economics and Reliability”, *IEEE Trans. on Power Syst.*, vol. 6, 2, pp.476-483, 1991.

[ONS04] ONS (Operador Nacional do Sistema Elétrico), NT-ONS-136-2004, “INEQUAÇÕES PARA MONITORAMENTO DE CARREGAMENTO”, *Rio de Janeiro: ONS*, 2004.

[ONS09] ONS (Operador Nacional do Sistema Elétrico), Procedimentos de Rede, Submódulo 6.5, “Programação de Intervenções em Instalações da Rede de Operação”, *Rio de Janeiro: ONS.*, Disponível em : <http://www.ons.org.br/procedimentos/index.aspx>, [Acessado em janeiro de 2011], 2009.

[Papa82] C.H. Papadimitriou, K. Steiglitz, “Combinatorial Optimization: Algorithms and Complexity”, *Dover Publications, INC., Mineola, New York*, ISBN 0-486-40258-4, 1982.

[Rech11] I. Rechenberg, “The Evolution-Strategist”, [online], Disponível em: <http://www.bionik.tu-berlin.de/institut/xs2evost.html>, [Acessado em janeiro de 2011], 2011.

[Robe04] M.L.R. Roberto, Programação Ótima de Desligamentos em Redes de Energia Elétrica Utilizando Algoritmos Genéticos, Dissertação de Mestrado, *Instituto de Computação, Universidade Federal Fluminense*, [pdf]. Disponível em: <http://www.ic.uff.br/PosGraduacao/Dissertacoes/481.pdf>, [Acessado em janeiro de 2011], 2004.

[Robe05] M.L.R. Roberto, J.C.S. Souza, M.B. Do Coutto Filho and M.T. Schilling. “Planejamento de Intervenções em Redes de Energia Elétrica Utilizando Algoritmos Genéticos”, *VII Simpósio Brasileiro de Automação Inteligente*, 2005, São Luis, Anais do VII SBAI, 2005. v. 1. pp.1-6, 2005.

[Rowe04] J. Rowe, D. Whitley, L. Barbulescu and J.P. Watson, “Properties of Gray and Binary Representations”, *Evolutionary Computation*, Vol. 12, Issue 1, pp.47-76, Mar. 2004.

[Sawa99] T. Sawa, T. Furukawa, M. Nomoto, T. Nagasawa, T. Sasaki, K. Deno, and T. Maekawa, “Automatic scheduling method using tabu search for maintenance outage tasks of transmission and substation system with network constraints”, in Proc. IEEE Power Eng. Soc. 1999 Winter Meeting, New York, USA, vol. 2, pp.895-900, Fev. 1999.

[Silv95] E.L. Silva, M. Morozowski, L.G.S. Fonseca, G.C. Oliveira, A.C.G. Melo and J.C.O. Mello, “Transmission Constrained Maintenance Scheduling of Generating Units: A Stochastic Programming Approach”, *IEEE Trans. on Power Syst.*, vol. 10, 2, pp.695-701, 1995.

[Siva08] S.N. Sivanandam, S.N. Deepa, “Introduction to Genetic Algorithms”, *New York: Springer*, ISBN 9783540731894, 2008.

[Souz11] J.C.S. Souza, M.B. Do Coutto Filho, M.L.R. Roberto, “A Genetic-Based Methodology for Evaluating Requested Outages of Power Network Elements”, *IEEE Transactions on Power Systems*, Vol. 26, pp.2242-2249, 2011.

[Stre81] J.P. Stremmel, Maintenance Scheduling for Generation System Planning, *IEEE Trans. on PAS*, vol. PAS-100, 3, pp. 1410-1419, 1981.

[Viei04] C.E.C. Vieira, C. C. Ribeiro and R. de C. e Souza, “Geradores de Números Aleatórios” [pdf], PUC-RioInf.MCC22/04 Junho, 2004, Disponível em : ftp://139.82.16.194/pub/docs/techreports/04_22_vieira.pdf, 2004.

[Wang00] Y. Wang and E. Handschin, “A new genetic algorithm for unit maintenance scheduling of power systems”, *Elect. Power & Energy Syst.*, vol. 22, no. 5, pp.343-348, Maio 2000.

[Yare08] Y. Yare, G. K. Venayagamoorthy and U. O. Aliyu , “Optimal generator scheduling using a modified discrete PSO”, *IET Gener. Transm. & Distrib.*, vol.2, no. 6, pp.834-846, Nov. 2008.

[Yare10] Y. Yare and G. K. Venayagamoorthy , “Optimal maintenance scheduling of generators using multiple swarms-MDPSO framework”, *Eng. Applic. of Artific. Intellig.*, vol. 23, no. 6, pp. 895-910, Set. 2010.

[Yell92] J. Yellen, T.M. Al-Khamis, S. Vemuri and L. Lemonidis, A Decomposition Approach to Unit Maintenance Scheduling, *IEEE Trans. on PWRS*, vol. 7, 2, pp. 726-733, 1992.

[Zurn76] H.H. Zurn and V.H. Quintana, Several Objectives Criteria for Optimal Generator Preventive Maintenance Scheduling, *IEEE Trans. on PWRS*, vol. 96, 3, pp. 984-992, 1976.

A.1 CONFIGURAÇÃO BÁSICA DO SISTEMA IEEE 14

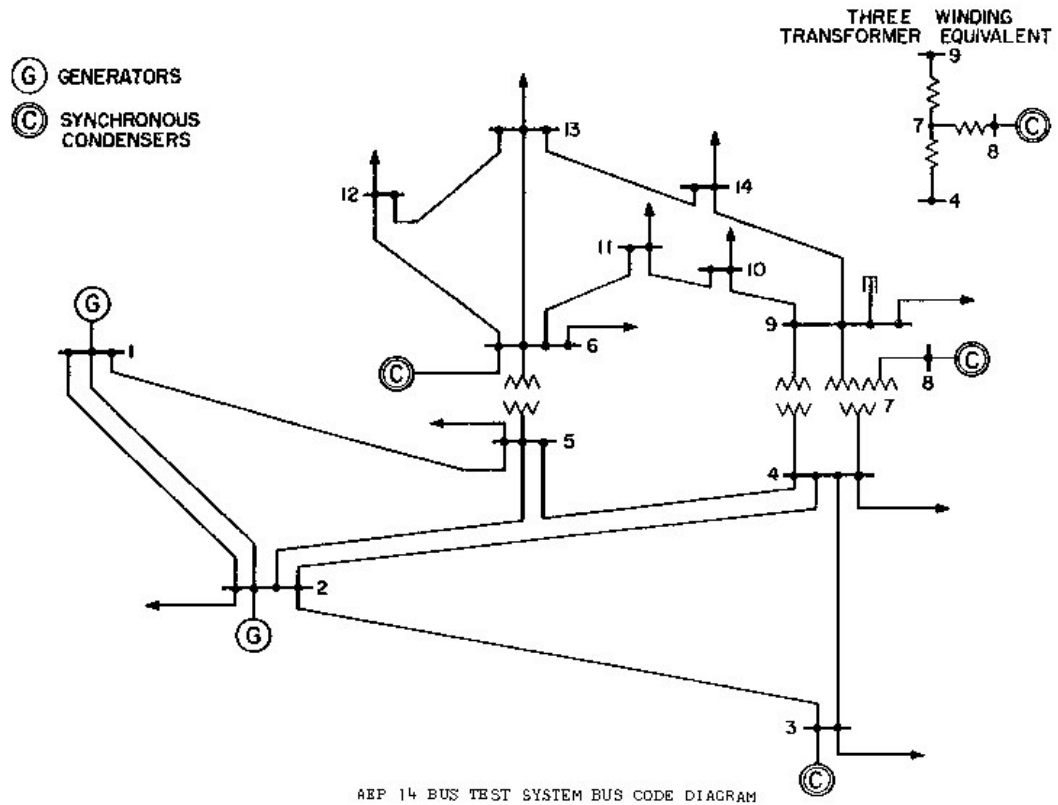


Figura A.1 - Sistema IEEE 14 barras [Chri99]

A.2 DADOS DE REDE

Tabela A. 1 - Dados de rede para o sistema IEEE 14 barras

Circuito		Parâmetros		
De	Para	R(pu)	X(pu)	B(pu)
1	2	0,01938	0,05917	0,0528
1	5	0,05403	0,22304	0,0492
2	3	0,04699	0,19797	0,0438
2	4	0,05811	0,17632	0,0374
2	5	0,05695	0,17388	0,0340
3	4	0,06701	0,17103	0,0346
4	5	0,01335	0,04211	0,0128
4	7	0	0,20912	0
4	9	0	0,55618	0
5	6	0	0,25202	0
6	11	0,09498	0,19890	0
6	12	0,12291	0,25581	0
6	13	0,06615	0,13027	0
7	8	0	0,17615	0
7	9	0	0,11001	0
9	10	0,03181	0,08450	0
9	14	0,12711	0,27038	0
10	11	0,08205	0,19207	0
12	13	0,22092	0,19988	0
13	14	0,17093	0,34802	0

A.3 DADOS DE CARGA

Tabela A. 2 - Dados de carga para o sistema IEEE 14 barras

Patamar de Carga Média					
Barra	Tipo de Barra (*)	Tensão Inicial (pu)	Potência Ativa (MW)	Carga Ativa (MW)	Carga Reativa (Mvar)
1	2	1,060	1589,075	0,0000	0,0000
2	1	1,045	34,000	12,3085	7,1995
3	1	1,010	0,000	54,2725	10,9395
4	0	1,019	-	27,4125	-2,2355
5	0	1,020	-	4,3265	0,9095
6	1	1,070	0,000	6,3835	4,2755
7	0	1,062	-	0,0000	0,0000
8	1	1,090	0,000	0,0000	0,0000
9	0	1,056	-	16,6345	9,3585
10	0	1,051	-	5,1178	3,2986
11	0	1,057	-	1,9892	1,0256
12	0	1,055	-	3,4517	0,9014
13	0	1,050	-	7,6245	3,2725
14	0	1,036	-	8,4915	2,8475

- (*) Tipos de Barra
- Tipo 0 – Barra PQ
- Tipo 1 – Barra PV
- Tipo 2 – Barra de Referência

A.4 CONFIGURAÇÃO BÁSICA DO SISTEMA IEEE 30

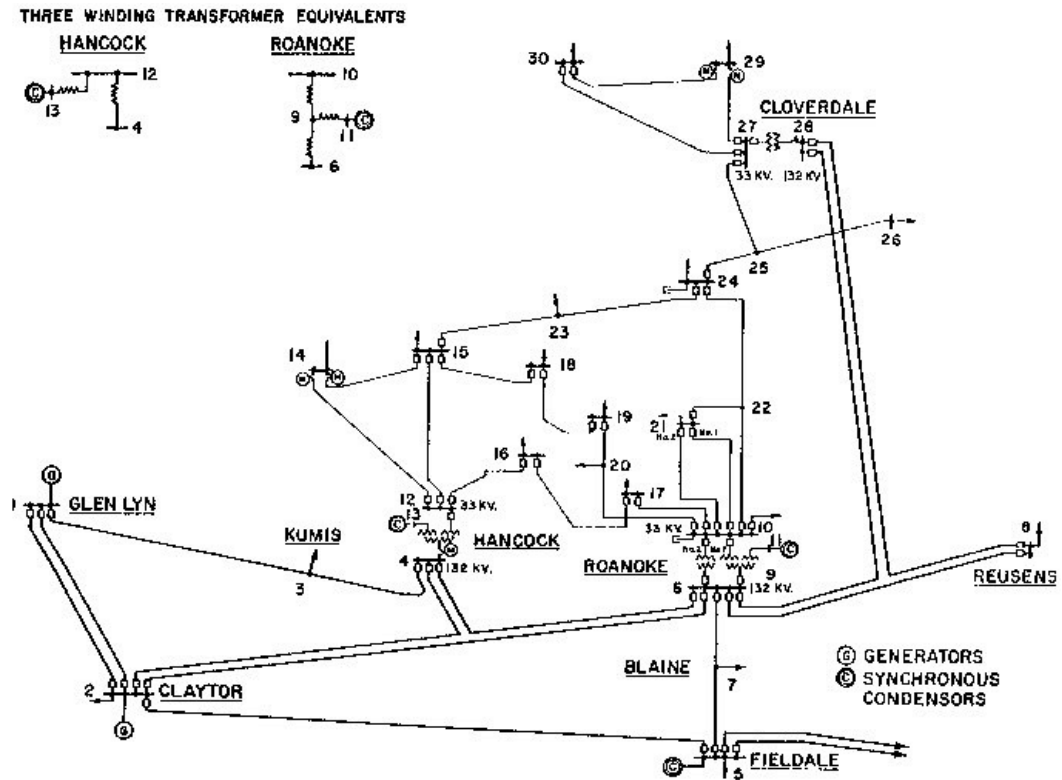


Figura A. 2 - Sistema IEEE 30 barras [Chri99]

A.5 DADOS DE REDE

Tabela A. 3 - Dados de rede para o sistema IEEE 30 barras

Circuito		Parâmetros		
De	Para	R(pu)	X(pu)	B(pu)
1	2	0,0194	0,0592	0,0528
1	3	0,0452	0,1652	0,0408
2	4	0,0570	0,1763	0,0368
3	4	0,0132	0,0379	0,0084
2	5	0,0472	0,1983	0,0418
2	6	0,0581	0,1763	0,0374
4	6	0,0119	0,0414	0,0090
5	7	0,0460	0,1160	0,0204
6	7	0,0267	0,0820	0,0170
6	8	0,0120	0,0420	0,0090
6	9	0,0000	0,2080	0,0000
6	10	0,0000	0,5560	0,0000
9	11	0,0000	0,2080	0,0000
9	10	0,0000	0,1100	0,0000
4	12	0,0000	0,2560	0,0000
12	13	0,0000	0,1400	0,0000
12	14	0,1231	0,2559	0,0000
12	15	0,0662	0,1304	0,0000
12	16	0,0945	0,1987	0,0000
14	15	0,2210	0,1997	0,0000
16	17	0,0524	0,1923	0,0000
15	18	0,1073	0,2185	0,0000
18	19	0,0639	0,1292	0,0000
19	20	0,0340	0,0680	0,0000
10	20	0,0936	0,2090	0,0000
10	17	0,0324	0,0845	0,0000
10	21	0,0348	0,0749	0,0000
10	22	0,0727	0,1499	0,0000
21	22	0,0116	0,0236	0,0000
15	23	0,1000	0,2020	0,0000
22	24	0,1150	0,1790	0,0000
23	24	0,1320	0,2700	0,0000
24	25	0,1885	0,3292	0,0000
25	26	0,2544	0,3800	0,0000
25	27	0,1093	0,2087	0,0000
28	27	0,0000	0,3960	0,0000
27	29	0,2198	0,4153	0,0000
27	30	0,3202	0,6027	0,0000
29	30	0,2399	0,4533	0,0000
8	28	0,0636	0,2000	0,0428
6	28	0,0169	0,0599	0,0130

A.6 DADOS DE CARGA

Tabela A. 4 - Dados de carga para o sistema IEEE 30 barras

Patamar de Carga Média					
Barra	Tipo de Barra (*)	Tensão Inicial (pu)	Potência Ativa (MW)	Carga Ativa (MW)	Carga Reativa (Mvar)
1	2	1,060	147,4467	0,0000	0,0000
2	1	1,043	22,6667	12,2967	7,1967
3	0	1,021	-	1,3600	0,6800
4	0	1,012	-	4,3067	0,9067
5	1	1,010	0,0000	53,3800	10,7667
6	0	1,010	-	0,0000	0,0000
7	0	1,002	-	12,9200	6,1767
8	1	1,010	0,0000	17,0000	17,0000
9	0	1,051	-	0,0000	0,0000
10	0	1,045	-	3,2867	1,1333
11	1	1,082	0,0000	0,0000	0,0000
12	0	1,057	-	6,3467	4,2500
13	1	1,071	0,0000	0,0000	0,0000
14	0	1,042	-	3,5133	0,9067
15	0	1,038	-	4,6467	1,4167
16	0	1,045	-	1,9833	1,0200
17	0	1,040	-	5,1000	3,2867
18	0	1,028	-	1,8133	0,5100
19	0	1,026	-	5,3833	1,9267
20	0	1,030	-	1,2467	0,3967
21	0	1,033	-	9,9167	6,3467
22	0	1,033	-	0,0000	0,0000
23	0	1,027	-	1,8133	0,9067
24	0	1,021	-	4,9300	3,7967
25	0	1,017	-	0,0000	0,0000
26	0	1,000	-	1,9833	1,3033
27	0	1,023	-	0,0000	0,0000
28	0	1,001	-	0,0000	0,0000
29	0	1,000	-	1,3600	0,5100
30	0	0,992	-	6,0067	1,0767

(*) Tipos de Barra

Tipo 0 – Barra PQ

Tipo 1 – Barra PV

Tipo 2 – Barra de Referência

GERADORES DE NÚMEROS PSEUDO-ALEATÓRIOS

B.1 TESTES DE RANDOMICIDADE

A medida de correlação entre os números gerados pelo GNPA pode ser feita através de testes estatísticos da hipótese de randomicidade que avaliam a imparcialidade entre os valores. Um destes testes é o teste chi-quadrado, descrito em [Knut97] pela equação (B.1).

$$V = \sum_{s=1}^k \frac{(Y_s - np_s)^2}{np_s} \quad (\text{B.1})$$

onde k corresponde ao número de categorias, n é o número de observações independentes, p_s é a probabilidade que cada observação se encaixe na categoria s e Y_s é o número de observações que efetivamente se encaixam na categoria s .

Conforme indicado em [Knut97] deve-se levar em conta que a distribuição chi-quadrado é uma aproximação válida apenas para valores de n suficientemente grandes. Este autor sugere valores maiores que 5 para o produto $n \times p_s$, preferencialmente muito maiores que 5 para um teste mais robusto. Para testar a hipótese de imparcialidade entre os valores obtidos nas observações, o valor V obtido na equação (B.1) deverá ser comparado ao da distribuição chi-quadrado, usualmente tabelado pela probabilidade de ocorrência, considerando o número de graus de liberdade $v=k-1$. O valor x da tabela com a coluna p e linha v indica que a quantidade V na equação (B.1) será menor que ou igual a x com a probabilidade aproximada p , se n é suficientemente grande. Por exemplo, para $v=10$ (10 graus de liberdade), teremos $V > 18,31$ somente cerca de 5% das vezes.

Ainda segundo [Knut97], para garantir a randomicidade entre os valores gerados pela série, devemos verificar se V é menor que o valor obtido na distribuição chi-quadrado com probabilidade 1% ou maior que o valor tabelado para 99%. Neste caso, os números obtidos não são suficientemente randômicos. Se o valor de V estiver entre os valores da distribuição correspondentes a 1% e 5% ou entre 95% e 99%, os números podem ser considerados “suspeitos”. Os casos onde V estiver entre os valores correspondentes a 5% a 10% e 90% a 95% podem ser considerados “quase suspeitos”.

[Knut97] recomenda que este teste deva ser feito no mínimo três vezes com diferentes grupos de números e se pelo menos dois dos três resultados são suspeitos, os números gerados pela série podem ser considerados insuficientemente randômicos.

Conforme ressaltado em [Kenn05], existem conjuntos de testes que avaliam a randomicidade da série através de outras hipóteses, com diversos graus de confiança baseados na aplicação onde os geradores serão utilizados. Dentre estes conjuntos podemos citar o do National Institute of Standards and Technology (NIST) e o DieHard de George Marsaglia.

B.2 GERADOR DE FIBONACCI SUBTRATIVO DEFASADO

O gerador de Fibonacci Subtrativo Defasado ou Gerador de Atraso Subtrativo de Fibonacci [Viei04] é baseado na sequência de Fibonacci definida pela equação (B.2).

$$x_n = x_{n-1} + x_{n-2} \text{ com } x_0 = 0 \text{ e } x_1 = 1 \quad (\text{B.2})$$

[Knut97] indica que a generalização desta sequência para uso do gerador de Fibonacci Subtrativo Defasado pode ser descrita conforme a equação (B.3).

$$x_n = (x_{n-l} - x_{n-k}) \bmod m \text{ com } 0 < l < k \quad (\text{B.3})$$

onde:

l e k são denominados atrasos (*lags*) com diversos valores estudados na literatura

m corresponde a uma potência de 2 para a maioria das aplicações de interesse

Com esta configuração, é necessário que sejam determinados *a priori* os l elementos (de x_0 a x_{l-1}). Conforme citado em [Knut97], para garantir boa randomicidade e comprimento para a sequência de atraso de Fibonacci foram propostos em 1958 por G.J.Mitchell e D.P.Moore os atrasos $l = 24$ e $k = 55$. Além disso, é necessário que pelo menos um dos 55 elementos definidos *a priori* seja ímpar.

Existe uma relação entre a sequência de Fibonacci e elementos encontrados na natureza que suporta, de maneira intuitiva, o uso desta classe de GNPA para AGs. A espiral encontrada em diversas formas da natureza como a folha da bromélia e na concha do nautilus marinho segue a sequência de Fibonacci. Exemplos desta sequência [Knot10] também são encontrados em pétalas de flores, caules de árvores, arranjos de sementes em flores de girassol e em árvores genealógicas de coelhos e abelhas.

B.3 MERSENNE TWISTER

Este algoritmo para um GNPA foi desenvolvido por Makoto Matsumoto e Takuji Nishimura em 1996/1997 [Mats98], com melhorias incorporadas em sua inicialização no ano 2002 [Mats11].

O Mersenne Twister é uma variante do gerador *Twisted Generalised Feedback Shift Register* (TGFSR) que utiliza um registrador de deslocamento de 32 bits cuja entrada é alimentada por uma operação linear (ou exclusivo)

entre alguns dos bits da palavra presente no registrador. Em sua parametrização proposta em [Mats98], este algoritmo é capaz de gerar uma série com um período de $2^{19937}-1$ (um número primo de Mersenne) e uma equidistribuição de 623 dimensões para uma precisão de 32bits, utilizando uma área de trabalho interna de apenas 624 words. Além disso, por utilizar apenas operações de baixo custo computacional como E, OU e OU exclusivo devido à escolha do campo de dois elementos para suas operações, este algoritmo possui execução mais rápida que outros equivalentes, considerando seu período e área de trabalho interna.

Variações deste algoritmo, implementadas desde sua criação, melhoraram alguns aspectos encontrados na versão inicial e algumas implementações direcionadas a hardware específico, para acelerar seu processamento, estão disponíveis em [Mats11].

B.4 TRANSFORMADA DE BOX-MULLER

Conforme ressaltado em [Luke09] é desejável que os geradores de números pseudoaleatórios possuam a funcionalidade de gerar números pertencentes a uma mesma Distribuição Normal ou Gaussiana. O algoritmo em [Box58] permite gerar números pertencentes a uma Distribuição Normal com variância unitária e média zero, a partir de dois números pseudoaleatórios pertencentes a uma distribuição uniforme.

Em sua forma polar [Bell68], este algoritmo apresenta um menor custo computacional, pois não utiliza as funções seno e cosseno, presentes em sua concepção original [Box58]. Uma terceira versão foi proposta por [Knop69], com base na forma polar proposta por [Bell68] e é amplamente utilizada atualmente.