

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

# **Otimização da Confiabilidade e Disponibilidade em Sistemas Redundantes**

Autor: Hélio Fiori de Castro  
Orientador: Profa. Dra. Katia Lucchesi Cavalca

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO**

# **Otimização da Confiabilidade e Disponibilidade em Sistemas Redundantes**

Autor: Hélio Fiori de Castro  
Orientador: Kátia Lucchesi Cavalca

Curso: Engenharia Mecânica  
Área de Concentração: Mecânica dos Sólidos e Projeto Mecânico

Dissertação de mestrado acadêmico apresentada à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Mestre em Engenharia Mecânica

Campinas, 2003  
S.P. – Brasil

**UNIVERSIDADE ESTADUAL DE CAMPINAS**  
**FACULDADE DE ENGENHARIA MECÂNICA**  
**COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**  
**DEPARTAMENTO DE PROJETO MECÂNICO**

DISSERTAÇÃO DE MESTRADO ACADÊMICO

**Otimização da Confiabilidade e  
Disponibilidade em Sistemas Redundantes**

Autor: Hélio Fiori de Castro  
Orientador: Kátia Lucchesi Cavalca

---

**Profa. Dra. Katia Lucchesi Cavalca**  
Universidade Estadual de Campinas

---

**Prof. Dr. Robson Pederiva**  
Universidade Estadual de Campinas

---

**Prof. Dr. Fernando Antônio Forcellini**  
Universidade Federal de Santa Catarina

Campinas, 7 de fevereiro de 2003

**Dedicatória:**

Dedico este trabalho aos meus amados pais, Lorivaldo e Tacira.

## **Agradecimentos**

Este trabalho não poderia ser terminado sem a ajuda de diversas pessoas às quais presto minha homenagem

Aos meus pais pelo incentivo em todos os momentos da minha vida.

À minha tia Yrma pela ajuda e apoio sempre que necessário.

À minha orientadora Professora Katia Lucchesi Cavalca, que mostrou os caminhos a serem seguidos e me apoiou nos momentos mais críticos.

Aos Professores Robson Pederiva e Franco Dedini pelas preciosas sugestões e conselhos no desenvolvimento do trabalho.

A todos os demais professores, funcionários e colegas do departamento, que ajudaram de forma direta e indireta na conclusão deste trabalho.

Ao Engenheiro Alexandre Degani e a ZF do Brasil S.A., pela colaboração com informações imprescindíveis para o trabalho.

*“Ó profundidade da riqueza, tanto da sabedoria, como do conhecimento de Deus! Quão insondáveis são os seus juízos e quão inescrutáveis são os seus caminhos”, Romanos 8.33*

## **Resumo**

CASTRO, Hélio Fiori de, *Otimização da Confiabilidade e Disponibilidade em Sistemas Redundantes*, Campinas,: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2003. 139 p. Dissertação (Mestrado)

Neste trabalho, buscou-se aplicar métodos de otimização a problemas de confiabilidade de sistemas redundantes. Sete problemas, envolvendo confiabilidade ou disponibilidade de sistemas redundantes, foram formulados, onde as variáveis de otimização eram o número de redundâncias e, no caso dos problemas de otimização da disponibilidade, a razão de dependabilidade, a qual relaciona os tempos de falha e manutenção dos componentes. Os métodos de otimização utilizados foram Multiplicadores de Lagrange, Estratégia Evolutiva para dois Indivíduos e Algoritmo Genético. Todos os métodos foram implementados para resolver os três problemas de otimização da confiabilidade, e apenas o Algoritmo Genético foi aplicado na solução dos problemas de otimização da disponibilidade, pois é o método mais indicado para problemas mais complexos. Os métodos de otimização foram comparados entre si, através de exemplos retirados da literatura. Exemplos reais foram aplicados aos problemas de otimização da disponibilidade. A metodologia desenvolvida neste trabalho pode ser aplicada a problemas mais complexos de otimização da confiabilidade em sistemas redundantes, devido à robustez dos métodos propostos.

### *Palavras Chave*

- Confiabilidade, Disponibilidade, Sistemas Redundantes, Otimização, Multiplicadores de Lagrange, Estratégia Evolutiva, Algoritmo Genético

## **Abstract**

CASTRO, Hélio Fiori de, *Reliability and Availability Optimization applied to redundant systems*, Campinas,: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2003. 139 p. Dissertação (Mestrado)

In this investigation, it was intended to apply optimizations methods, in order to solve reliability problems of redundant systems. Seven problems, that involved reliability or availability of redundant system, were formulated, where the optimization variables were the number of redundancies for all cases and the dependability ratio, which relates the failure times and the maintenance times, for the cases where the availability was involved. The optimization methods applied are Lagrange Multipliers, Evolution Strategies for two individuals and Genetic Algorithm. All methods were implemented to solve three problems of reliability optimization. And only the Genetic Algorithm was applied in the solution of availability optimization, because it has a better performance for complex problem. The optimization methods were compared to each other through examples from the technical literature. Real data were applied to availability optimization problems. The methodology developed in this investigation can be applied to other complex reliability optimization problems in redundant systems, due to the robustness of the considered methods.

### *Key Words*

- Reliability, Availability, Redundant System, Optimization, Lagrange Multipliers, Evolution Strategy, Genetic Algorithm

# Índice

Lista de Figuras .....	iv
Lista de Tabelas.....	vi
Nomenclatura .....	vii
1. Introdução.....	1
2. Revisão Bibliográfica.....	7
3. Conceitos de Confiabilidade associados à Redundância de Sistemas.....	17
3.1 Definições Gerais.....	18
3.2 Distribuições Estatísticas .....	21
3.2.1 Distribuição Exponencial .....	22
3.2.2 Distribuição de Weibull .....	24
3.3 Análise de Disponibilidade.....	26
3.3.1 Cálculo da Disponibilidade .....	26
3.3.2 Disponibilidade e Projeto.....	29
3.4 Análise de Dependabilidade .....	31
3.5 Sistemas redundantes.....	34

3.5.1	Redundância Ativa .....	36
3.5.2	Redundância Passiva .....	37
3.5.3	Redundância de Unidades .....	37
4.	Métodos de Otimização .....	41
4.1	Conceitos Gerais .....	41
4.2	Método do Cálculo Diferencial .....	44
4.3	Métodos de Busca para problemas com uma variável .....	45
4.3.1	Intervalos Iguais .....	46
4.3.2	Razão Áurea .....	47
4.3.3	Método de Fibonacci .....	48
4.4	Métodos de Busca para problemas com multi-variáveis .....	49
4.4.1	Steepest Descent .....	50
4.5	Multiplicadores de Lagrange .....	52
4.6	Programas ou Algoritmos Evolutivos .....	55
4.6.1	Algoritmos Genéticos .....	58
4.6.2	Estratégia Evolutiva .....	61
5.	Formulação dos Problemas e Implementação Computacional .....	65
5.1	Formulação de Problemas de Otimização de Redundância .....	65
5.1.1	Problema 1: Minimização do Custo para uma Confiabilidade Mínima .....	66
5.1.2	Problema 2: Maximização da confiabilidade para um Custo Máximo .....	67
5.1.3	Problema 3: Maximização da Confiabilidade para Custo e Peso Máximo .....	68
5.1.4	Problema 4: Maximização da Disponibilidade com Redundância e MTTR como variáveis de otimização .....	69
5.1.5	Problema 5: Maximização da Disponibilidade com Redundância e MTTF como variáveis de otimização .....	71
5.1.6	Problema 6: Maximização da Disponibilidade para Equipes de Manutenção .....	72

5.1.7	Problema 7: Maximização da Disponibilidade para recursos de Manutenção .....	74
5.2	Implementação dos Multiplicadores de Lagrange .....	76
5.2.1	Problema 1 .....	78
5.2.2	Problema 2 .....	83
5.2.3	Problema 3 .....	87
5.3	Estratégia Evolutiva.....	91
5.4	Algoritmo Genético .....	93
5.4.1	Maximização da Confiabilidade e Minimização do Custo.....	93
5.4.2	Maximização da Disponibilidade.....	95
5.5	Desenvolvimento do Programa.....	96
5.5.1	Otimização da Confiabilidade.....	98
5.5.2	Otimização da Disponibilidade .....	100
5.5.3	Análise Gráfica.....	103
6.	Aplicações e Exemplos .....	107
6.1	Comparação entre os Métodos.....	107
6.2	Parâmetro do Algoritmo Genético .....	115
6.3	Exemplos Reais .....	123
6.3.1	Processo de Produção.....	124
6.3.2	Dados das Máquinas Ferramentas.....	125
6.3.3	Simulação da maximização da Disponibilidade com o MTTR como variável .....	126
6.3.4	Simulação da maximização da Disponibilidade para Equipes de Manutenção.....	128
6.3.5	Simulação da maximização da Disponibilidade para Recursos de Manutenção .....	131
7.	Conclusões e Perspectivas Futuras.....	135
	Referências Bibliográficas .....	141

## Lista de Figuras

Figura 3.1 - Curva da Banheira .....	19
Figura 3.2 – Disponibilidade como parâmetro de Projeto .....	29
Figura 3.3 - Relação Linear entre MTTF e MTTR .....	30
Figura 3.4 - Região Economicamente Aceita .....	31
Figura 3.5 - Variação da Disponibilidade pela Razão de Dependabilidade.....	32
Figura 3.6 - Variação da Dependabilidade pelo MTBF ( $\lambda t$ ).....	34
Figura 3.7 - Sistema em Série .....	35
Figura 3.8 - Sistema em Paralelo .....	36
Figura 3.9 - Redundância Passiva .....	37
Figura 3.10 - Sistema Redundante .....	38
Figura 4.1 - Máximos e Mínimos locais e globais , pontos de inflexão e descontinuidade de uma Função $U(x)$ .....	45
Figura 4.2 - Exemplo ilustrativo para o Método da Razão Áurea e Método de Fibonacci.....	47
Figura 4.3 - Curvas de nível para função $U$ .....	50
Figura 4.4 - Fluxograma do Algoritmo Genético.....	59
Figura 4.5 - Mutação e Crossover .....	60
Figura 4.6 - Inversão .....	61
Figura 4.7 - Fluxograma da Estratégia Evolutiva para dois Indivíduos.....	63
Figura 5.1 - Atuação de Equipes de manutenção .....	72
Figura 5.2 - Fluxograma do Multiplicadores de Lagrange para o Problema 1 .....	82

Figura 5.3 - Fluxograma do Multiplicadores de Lagrange para o Problema 2 .....	86
Figura 5.4 - Fluxograma do Multiplicadores de Lagrange para o Problema 3 .....	90
Figura 5.5 – Formação do indivíduo inicial na Estratégia Evolutiva.....	91
Figura 5.6 – Processo de Mutação na Estratégia Evolutiva .....	92
Figura 5.7 – Crossover com dois pontos de interseção .....	96
Figura 6.1 – Evolução do exemplo um com o tempo para Estratégia Evolutiva .....	110
Figura 6.2 – Evolução do exemplo um com o tempo para Algoritmo Genético.....	110
Figura 6.3 – Gerações do Exemplo 1 .....	111
Figura 6.4 - Evolução do exemplo dois com o tempo para Algoritmo Genético.....	113
Figura 6.5 - Variação da Confiabilidade Média com o total de Gerações .....	116
Figura 6.6 – Variação do Tempo de Convergência com o Total de Gerações.....	116
Figura 6. 7 – Variação da Confiabilidade Média com o Tamanho da População.....	118
Figura 6. 8– Variação do Tempo de Convergência com o Tamanho da população .....	118
Figura 6.9 – Variação do tempo de convergência com a probabilidade de crossover .....	119
Figura 6.10 – Variação da Média da Confiabilidade com a probabilidade de Crossover.....	120
Figura 6.11 – Variação do tempo de convergência com a probabilidade de mutação .....	122
Figura 6.12 - Variação do tempo de convergência com a probabilidade de mutação.....	122
Figura 6.13 - Carça do Freio.....	123
Figura 6.14 - Processo de Produção .....	124
Figura 6.15 - Linha de Produção .....	124
Figura 6.16 - Torno Vertical Robotizado .....	125
Figura 6.17 – Centro de Usinagem.....	126
Figura 6.18 – Entalhadora .....	126
Figura 6.19 – Otimização do Problema 4.....	127
Figura 6.20 – Análise Gráfica do Problema 4 na primeira geração .....	128
Figura 6.21 – Otimização do Problema 6.....	130
Figura 6.22 – Análise Gráfica do Problema 4 na geração 238.....	131
Figura 6.23 – Otimização do Problema 7.....	132

## Lista de Tabelas

Tabela 6.1 – Exemplo de sistema para o Problema 1.....	108
Tabela 6.2 – Exemplo de sistema para o Problema 2.....	108
Tabela 6.3 – Exemplo de sistema para o Problema 3.....	108
Tabela 6.4 – Resulta ótimo para o exemplo 1 .....	109
Tabela 6.5 – Resulta ótimo para o exemplo 2 .....	112
Tabela 6.6 – Resulta ótimo para o exemplo 3 .....	112
Tabela 6.7 – Exemplo com nove estágios .....	114
Tabela 6.8 - Variação do total de gerações .....	115
Tabela 6.9 - Variação do tamanho da população .....	117
Tabela 6.10 – Variação da probabilidade de crossover.....	119
Tabela 6.11 – Variação da Probabilidade de Mutação com probabilidade de crossover igual a 0% .....	121
Tabela 6.12 – Variação da Probabilidade de Mutação com probabilidade de crossover igual a 10%.....	121
Tabela 6.13 – Dados dos Equipamentos .....	125

## Nomenclatura

### *Letras Latinas*

A – Disponibilidade em Regime Permanente

A(t) – Disponibilidade Instantânea

A\*(T) – Disponibilidade em um período de Tempo T

$\tilde{A}(t)$  – Complementar da Disponibilidade Instantânea

b – cadeia binária

c – Custo de um componente [unidade monetária]

cd – Custo de melhoria da razão de dependabilidade de um componente [unidade monetária]

cm – Custo de manutenção de um componente [unidade monetária]

ceq – Custo de equipe de manutenção em um componente [unidade monetária]

crec – Custo de recurso de manutenção em um componente [unidade monetária]

C – Custo Global, ou Custo do Sistema [unidade monetária]

CM – Custo Global de Manutenção [unidade monetária]

d – Razão de Dependabilidade

eq – número de equipes de manutenção

D(t) – Dependabilidade Temporal

$D_{\text{MIN}}$  – Dependabilidade Mínima  
 $f(t)$  – Função densidade de probabilidade  
 $F$  – Número de Fibonacci  
 $feq$  – fator de influencia de equipes de manutenção  
 $g$  – Restrição de desigualdade  
 $h$  – Restrição de igualdade  
 $I$  – impacto dos recursos de manutenção  
 $k$  – Tamanho de um cadeia binária  
 $L$  – Comprimento de um intervalo  
 $LE$  – Expressão de Lagrange  
 $LR$  – Comprimento de redução  
 $MTBF$  – Tempo médio entre falhas [unidade de tempo]  
 $MTTF$  – Tempo médio até a falha [unidade de tempo]  
 $MTTR$  – Tempo médio até o reparo [unidade de tempo]  
 $M(T)$  – Manutenibilidade de um dispositivo em um período de tempo  $T$   
 $m$  – Número de equções de restrições  
 $n$  – Número de sub-sistema ou estágios  
 $\bar{n}$  – Vetor normal unitário  
 $N$  – Número máximo de iterações no Método de Fibonacci  
 $\bar{N}$  – Vetor normal  
 $P_{\text{ch}}$  – Probabilidade de sucesso no chaveamento de uma unidade para outra  
 $Q(T)$  – Probabilidade de falha de um sistema ou componente no período de tempo  $T$   
 $rec$  – Porcentagem de recursos de manutenção [%]  
 $R(T)$  – Confiabilidade do sistema ou componente no período de tempo  $T$ .

$t$ – Variável Tempo	[unidade de tempo]
$\bar{t}$ – Vetor tangencial unitário	
$T$ – Período de Tempo	[unidade de tempo]
$U$ – Função objetivo	
$v$ – Volume de um componente	[unidade volumétrica]
$V$ – Volume Global, ou Volume do Sistema	[unidade volumétrica]
$w$ – Peso de um componente	[unidade de peso]
$W$ – Peso Global, ou Peso do Sistema	[unidade de peso]
$x$ – Variável de um problema qualquer de otimização	
$XL$ – Limite inferior de um intervalo de busca em uma dimensão	
$XU$ – Limite superior de um intervalo de busca em uma dimensão	
$y$ – Número de componentes em paralelo em um sub-sistema ou estágio	
$z$ – Vetor de mutação na Estratégia Evolutiva	

### ***Letras Gregas***

$\alpha$ – Parâmetro de Escala para análise de falha.	
$\beta$ – Parâmetro de forma para análise de falha.	
$\delta$ – Parâmetro de Escala para análise de manutenção.	
$\gamma$ – Parâmetro de forma para análise de manutenção.	
$\lambda$ - Taxa de Falha	[no. de falhas / unidade de tempo]
$\mu$ - Taxa de Reparo	[no. de reparos / unidade de tempo]
$\Delta$ - Incremento em uma iteração do Steepest Descent (Ascent)	
$\nabla$ – Gradiente	

$\varepsilon$  - Tolerância

$\Gamma$  – Função Gamma

$\psi$  - Função de restrição no Método dos multiplicadores de Lagrange

$\nu$  - Multiplicadores de Lagrange

$\sigma$  - desvio padrão de um distribuição normal

$\vartheta$  - Porcentagem de mutações que obtiveram sucesso

$\phi$  - Parametrização da Confiabilidade

### ***Subscritos***

ch – corresponde ao chaveamento de um sistema stand-by

i – índice correspondente ao sub-sistema ou estágio de um sistema redundante

j – série de Fibonacci

k – número de variáveis de um problema de otimização genérico

l – índice correspondente a um termo de uma cadeia binária

max – referente a valor máximo permitido

min – referente a valor mínimo permitido

o – corresponde ao componente ativo de um sistema stand-by

r – corresponde ao componente passivo de um sistema em stand-by

s -corresponde a um sistema de engenharia, formado por sub-sistemas ou estágios

t – iteração ou geração

## ***Superescritos***

\* - solução ótima

## ***Abreviações***

GA – Algoritmo Genético (do inglês, Genetic Algorithm)

ES - Estratégia Evolutiva (do inglês, Evolution Strategies)

EA – Algoritmos Evolutivos (do inglês, Evolutionary Algorithms)

PC – Personal Computer

SA – Simulated Annealing

TB – Tabu Search

## ***Siglas***

DPM – Departamento de Projeto Mecânico



# Capítulo 1

## Introdução

A forte tendência tecnológica atual caminha em direção à qualidade temporal do desempenho da função de projeto de componentes e sistemas, além de medidas que assegurem tal performance. No processo de concepção e desenvolvimento do projeto de máquinas e sistemas, é indispensável a preocupação com ferramentas que possam prever desempenho e performance de sistemas. Portanto, o conceito de confiabilidade, ou seja, a probabilidade de um componente ou sistema realizar uma função especificada em projeto, sob condições de operação pré-determinadas, e num certo período de tempo, torna-se uma importante grandeza probabilística na fase de projeto.

Após a 1<sup>a</sup>. Guerra Mundial, o conceito de confiabilidade começou a ser empregado de uma maneira qualitativa, devido ao crescente desenvolvimento da indústria aeronáutica. Na década de 30, houve uma quantificação inicial deste conceito através da taxa média de falha e do número médio de falhas. Durante a segunda guerra, os mísseis alemães V-1 e V-2 tiveram uma grande contribuição para o desenvolvimento e aplicação dos conceitos de confiabilidade. A não disponibilidade de equipamentos eletrônicos durante a guerra da Coreia fez surgir um grande interesse militar dos Estados Unidos em confiabilidade. Da mesma forma, o black-out de 1965 nos estados do norte deste país desencadeou um forte impulso à aplicação dos conceitos de confiabilidade.

Outro conceito importante é a capacidade de recolocação em funcionamento de um componente ou sistema após uma falha. Entende-se por falha, o fato de um sistema não poder mais cumprir sua função de projeto. Esta capacidade pode ser quantificada por outra grandeza probabilística, a manutenibilidade. Aliando a confiabilidade e manutenibilidade, pode-se determinar a probabilidade de que tal sistema esteja pronto para funcionamento, ou seja, o quanto disponível o sistema será para uso. Esta grandeza é conhecida como disponibilidade.

A redundância de componentes em sistemas é uma das formas de incrementar a confiabilidade, pois na falha de um componente, a unidade redundante no sistema passa a executar o objetivo de projeto. Entretanto, um sistema mais confiável é um sistema mais caro. O custo de projeto cresce muito com o aumento da confiabilidade. Porém, em alguns casos, a segurança e confiabilidade são primordiais. É o caso da indústria aeronáutica, onde a falha pode significar a morte de centenas de pessoas.

Outros custos são importantes na concepção e funcionamento de sistemas. Os custos de manutenção e o custo por paradas podem ser previstos durante a fase de projeto, através dos conceitos de manutenibilidade e disponibilidade.

O aumento da confiabilidade encarece o produto, porém proporciona maior disponibilidade, o que reduz os custos de manutenção e parada. O uso de técnicas de otimização, com o intuito de minimizar o custo global, é uma proposta interessante para se determinar o número de redundâncias ideais em um projeto, tendo em conta grandezas como confiabilidade e manutenibilidade. Outra proposta de otimização seria a maximização da confiabilidade ou da disponibilidade, tendo como restrição, além de volume e da massa, os custos de projeto e manutenção.

Existe uma enorme variedade de processos em nosso cotidiano que podem ser facilmente descritas como sistemas, desde sistemas físicos reais, como plantas de processos químicos, por exemplo, até entidades teóricas como modelos matemáticos. A eficiência de operação destes sistemas requer uma tentativa de otimização dos diversos parâmetros que influenciam na performance do sistema e agregam valor ao mesmo.

Com o recente avanço da tecnologia computacional, diversos métodos de otimização passaram a ser possíveis, o que proporcionou um grande avanço em resoluções de problemas com ótimo *design*.

Devem ser considerados dois tipos de efeitos inerentes, associados a sistemas de engenharia mecânica: efeitos desejáveis, como longa vida útil, alta eficiência da energia de saída, boa capacidade de transmissão de potência; e efeitos indesejáveis, como peso excessivo, grandes deflexões e vibrações mecânicas e altos custos. Então, podemos definir ótimo *design* como sendo a condição de projeto onde os efeitos indesejáveis são mínimos e os desejáveis máximos.

Para se formular problemas de otimização, é necessária uma função objetivo, que será otimizada, como, por exemplo, o custo, bem como restrições de projeto, como nível de confiabilidade, disponibilidade, peso e outros parâmetros, assumindo formas de equações ou inequações.

Os modelos matemáticos de otimização de projeto podem ser classificados em duas categorias: modelos analíticos, que incluem cálculo diferencial, multiplicadores de Lagrange e outros; e modelos numéricos, como a programação linear (método simplex) e programação não-linear.

Nos casos onde mais de um parâmetro deve ser otimizado, é necessário o uso de métodos de otimização multi-objetivo. A vantagem destes métodos se comparados com aqueles de otimização com um único critério é que o conjunto de soluções alternativas pode ser obtido.

Recentemente, métodos de busca baseados em fenômenos biológicos e físicos foram desenvolvidos e aplicados na otimização de sistemas de engenharia. Destacam-se a Estratégia Evolutiva, a qual se baseia no modelo de evolução, através de mutação de organismos vivos; o Algoritmo Genético, que alia ao conceito de mutação a idéia de cruzamento, que é representado pelo *crossover* de genes. O *Simulated Annealing* é um método de otimização que se baseia no processo físico do recozimento.

Propõe-se aplicar e analisar alguns métodos de otimização (Multiplicadores de Lagrange, Estratégia Evolutiva para dois Indivíduos e Algoritmo Genético) na otimização de redundância em confiabilidade de sistemas, determinando qual método se aplica melhor nos diversos problemas da área. Além disto, objetiva-se utilizar o método escolhido em problemas reais no cotidiano de empresas, mostrando a aplicabilidade do trabalho.

O Capítulo 2 deste trabalho fará uma revisão dos trabalhos publicados relevantes para a otimização da confiabilidade de sistemas redundantes.

Os conceitos de confiabilidade associados à redundância de sistemas serão tratados no Capítulo 3.

Os métodos de otimização utilizados no trabalho serão tratados no Capítulo 4, enquanto que a formulação dos problemas de otimização e sua implementação computacional serão detalhadas no Capítulo 5.

As aplicações dos métodos desenvolvidos, com comparações com trabalhos publicados e um exemplo do cotidiano industrial serão analisadas no Capítulo 6 e as conclusões finais do trabalho e perspectivas futuras formarão o Capítulo 7.



## Capítulo 2

### Revisão Bibliográfica

Uma performance confiável de sistemas para uma tarefa sobre várias condições é de extrema importância em muitas situações na indústria e na vida diária. Embora os conceitos qualitativos de confiabilidade sejam bastante antigos, seus aspectos quantitativos passaram a ser desenvolvidos na década de cinquenta.

Este tratamento quantitativo baseia-se nos tempos de falhas de sistemas de engenharia, tratados com ferramentas estatísticas.

De forma semelhante, o conceito de manutenibilidade é tratado, considerando os tempos de reparo.

Aliando a confiabilidade e a manutenibilidade, define-se a disponibilidade e a dependabilidade. O primeiro conceito relaciona o tempo em que o equipamento está disponível para uso com seu tempo de vida, enquanto o segundo relaciona os tempos de falha e de reparo.

O desenvolvimento técnico destes conceitos engloba:

- tratamentos estatísticos;

- aspectos combinatórios de confiabilidade de sistemas;
- técnicas gerais de estimativa de valores da confiabilidade, manutenibilidade e disponibilidade de sistemas complexos;
- aplicação destes conceitos em ambientes industriais, engenharia de produto e engenharia de manutenção;
- técnicas experimentais para determinação da confiabilidade de novos produtos.

Estes conceitos são explorados nas obras de Ertas (1993), Ramakumar (1993), Lewis (1996) e Doty (1989), as quais serviram de base para o desenvolvimento do presente trabalho.

O conceito de dependabilidade foi desenvolvido por Wohl (1966) e apresenta uma importante relação custo-benefício entre a manutenção e o tempo de funcionamento dos sistemas.

Os aspectos relacionados à redundância em sistemas de engenharia da confiabilidade, manutenibilidade, disponibilidade e dependabilidade serão detalhados mais adiante no Capítulo 3.

A otimização da confiabilidade tem atraído trabalhos científicos desde a década de 1960, devido à importância crítica da confiabilidade em vários sistemas. Para maximizar a confiabilidade, as opções seguintes podem ser consideradas:

1. Aumentar a confiabilidade dos componentes do sistema;
2. Alocar componentes redundantes em paralelo;
3. Uma combinação de (1) e (2);

#### 4. Utilização de componentes intercambiáveis.

A primeira opção apresentada foi bastante explorada nas décadas de 1960 e 1970. Após 1977, os trabalhos publicados se concentraram principalmente nas demais opções.

Entretanto, o trabalho “Optimization Techniques for System Reliability with Redundancy – A Review” de Tillman, Hwang e Kuo (1977) apresentaram uma revisão dos trabalhos publicados até aquela data sobre otimização da confiabilidade em sistemas redundantes (opções 2 e 3). Destacaram-se os métodos de otimização utilizados nas publicações, e as várias configurações de sistemas de confiabilidade: série, paralelo, série-paralelo, paralelo-série, *standby* e sistemas que não seguem estes padrões. Concluiu-se que a maior parte dos métodos empregados nas publicações resolviam uma pequena escala dos problemas de otimização relacionados a confiabilidade de sistemas.

Nesta mesma edição, Tillman, Hwang e Kuo publicaram um trabalho onde otimizaram a confiabilidade de um sistema em paralelo com redundância, tendo como variáveis o número de redundâncias e a confiabilidade dos componentes e como restrição o custo, volume e peso. Eles utilizaram uma combinação entre os métodos de busca de Hooke e Jeeves com a aproximação heurística de Aggarwal.

Banerjee e Rajamani (1973), em uma das publicações analisadas por Tillman et al., desenvolveram algoritmos, baseados em Multiplicadores de Lagrange, para maximizar a confiabilidade em sistemas redundantes, tendo como restrição o custo (algoritmo 1), minimizar o custo com a confiabilidade como restrição (algoritmo 2) e maximizar a confiabilidade para múltiplas restrições (algoritmo 3). Lopes (1998) implementou os algoritmos propostos por este trabalho em linguagem Visual Basic 5.0.

No Capítulo 5, será demonstrada a implementação computacional destes três algoritmos, pois servirão de comparação com os métodos propostos neste trabalho.

A disponibilidade do sistema também pode ser otimizada em sistemas redundantes. Sasaki, Kaburaki e Yanagi (1977) propuseram um método de maximização da disponibilidade de sistemas redundantes com custo e peso como restrições. O custo também pode ser minimizado tendo os demais parâmetros como funções restritivas.

Um algoritmo rápido para a otimização da confiabilidade em sistemas redundantes com várias restrições, não necessariamente lineares, foi proposto por Narasimhalu e Sivaramakrishnan (1978). Este método garante valores próximos ao ótimo global do sistema.

As contribuições feitas na otimização baseada na confiabilidade de sistemas de 1977 a 2000 são revisadas na publicação de Kuo e Prasad (2000). Neste período, foi dada maior atenção no desenvolvimento de métodos heurísticos e metaheurísticos. Os métodos de otimização do período analisado foram divididos em 7 categorias:

1. Métodos heurísticos para sistemas redundantes;
2. Algoritmos metaheurísticos para sistemas redundantes;
3. Algoritmos exatos para sistemas redundantes;
4. Métodos heurísticos para otimização da confiabilidade dos componentes em sistemas redundantes;
5. Otimização Multicritérios de sistemas;
6. Otimização de componentes intercambiáveis;
7. Outros métodos.

Os algoritmos metaheurísticos destacados foram: Algoritmos Genéticos (GA), *Simulated Annealing* (SA) e *Tabu Search* (TS).

O GA é um algoritmo metaheurístico que imita processos naturais de evolução e genética, e foi utilizado para a resolução de problemas de otimização da confiabilidade de sistemas redundantes. Eles são usados para resolver problemas de otimização complexos e discretos e não requerem o uso de matemática sofisticada, além de serem facilmente implementados em computadores. A maior desvantagem do método se dá na escolha imprópria dos valores de seus parâmetros, o que pode fazer com que a solução convirja rapidamente para um ótimo local ou exija muitas iterações para se atingir o ótimo global.

Outro algoritmo metaheurístico é o *Simulated Annealing* (SA), o qual imita o processo físico do recozimento. Este método pode ser usado na otimização de grandes problemas combinatórios e não explora uma estrutura especial que exista na função objetivo ou nas restrições. A maior desvantagem do método é a necessidade de muita memória computacional, pois faz muitos testes de funções.

O método metaheurístico *Tabu Search* (TS), assim como o método SA, é usado para a otimização de grandes problemas combinatórios. O método utiliza a memória de cálculo para direcionar a procura do ponto ótimo. Um TS simples pode ser facilmente implementado e pode ser usado para a resolução de problemas de otimização de redundâncias. A maior dificuldade do método é definir a estrutura e estratégia de uso da memória. Para isto é necessário um conhecimento do problema natural e muitos experimentos numéricos.

Os métodos de soluções exatas não foram muito explorados no período de 1977 a 2000, pois o fato de serem problemas não lineares traz uma maior dificuldade na implementação computacional, além de exigir mais memória computacional. Então, concentrou-se esforço em

métodos heurísticos. Entretanto, alguns métodos exatos são vantajosos quando o problema não é muito extenso, além de servirem para ajustar métodos heurísticos.

A otimização de componentes intercambiáveis caracteriza-se por serem estes problemas não lineares. Mas, de um ponto de vista matemático, a natureza destes problemas se diferencia do problema de otimização de redundância de sistemas, porque não há a necessidade de recursos para a otimização. Foram desenvolvidos para a solução destes problemas métodos heurísticos e exatos.

No projeto da confiabilidade de um sistema, faz-se necessário maximizar confiabilidade, manutenibilidade (probabilidade de que o reparo do sistema ou componente seja feito dentro de um intervalo temporal pré-determinado) e disponibilidade, a qual é uma combinação da confiabilidade e manutenibilidade. Além de se minimizar os recursos consumidos, como o custo, e algumas características físicas do sistema como a massa e o volume. Então, métodos de otimização com objetivos múltiplos (ou otimização multicritérios) são necessários para a solução destes problemas. Entretanto, esta ferramenta foi pouco explorada na otimização de sistemas redundantes.

Os problemas de otimização da confiabilidade de sistemas caracterizam-se por serem problemas inteiros, ou problemas com variáveis reais e inteiras, o que dificulta muito o desenvolvimento de métodos de otimização. Um método não linear foi proposto por Hikita et al. (1992) para a solução de vários sistemas, como sistemas em série, sistemas complexos e sistemas representados por árvores de falha.

Os métodos randômicos e heurísticos foram bastante utilizados no período analisado por Kuo et al (2000), pois se adaptam a natureza do problema de otimização da confiabilidade (variáveis inteiras e reais). Entretanto, a utilização destes métodos se iniciou na década de 1970. Kuo, Hwang e Tillman (1978) fizeram uma comparação entre seis métodos de otimização heurísticos, aplicados a otimização de confiabilidade de sistema, desenvolvidos na época.

Deve-se entender por método de otimização heurístico, como um algoritmo que se adequa especificamente ao problema tratado, podendo sofrer ajustes em sua estrutura, até que os resultados sejam satisfatórios.

Os métodos de busca aleatória são um importante exemplo de métodos heurísticos. Entre eles destacam-se os Programas Evolutivos, dos quais fazem parte os Algoritmos Genéticos, as Estratégias Evolutivas e outros, Simulated Annealing e Tabu Search. Um exemplo de método de busca aleatória foi desenvolvido por Mohan e Shanker (1988). Propôs-se otimizar a confiabilidade de sistemas complexos, como sistemas de ponte.

O Algoritmo de busca Simulated Annealing (SA), o qual simula o processo físico do recozimento, é uma opção que pode ser aproveitada na otimização da confiabilidade em sistemas redundantes. Ravi, Murty e Reddy (1997) desenvolveram um algoritmo, baseado no SA, para otimizar a confiabilidade em sistemas complexos. Um destes sistemas é o sistema redundante, tratado neste trabalho.

Os Algoritmos Genéticos (GA) passaram a ser utilizados na otimização de sistemas redundantes a partir da década de 1990. Este método foi criado por Holland (1992, 2ª. edição) em 1975. O objetivo inicial de Holland não era desenvolver um algoritmo para solucionar problemas específicos, mas criar um simulador para o estudo da evolução e adaptação natural das espécies. Estudos posteriores validaram o GA como método de otimização e de busca.

A obra de Goldberg (1989) apresenta, além dos fundamentos criados por Holland, esquemas para implementação computacional, aplicações do método, operadores avançados e técnicas de busca para o GA.

Assim como Goldberg, Mitchell (1996) mostra os fundamentos, aplicações e implementações dos GA. Davis (1996) editou um Handbook de GA, onde aplicações práticas em diversos problemas são mostrados.

Michalewicz (1996) explora, além do GA clássico, outros métodos de otimização, que se baseiam na evolução. Juntos, todos estes métodos podem ser classificados como Programas Evolutivos.

Os programas Evolutivos clássicos são os Algoritmos Genéticos, a Estratégia Evolutiva, Programação Evolutiva e a Programação Genética. Qualquer método de busca ou otimização, que derive dos métodos citados acima ou use os mesmos conceitos pode ser classificado como um Programa Evolutivo.

Uma importante característica dos Programas Evolutivos é o fato de que o algoritmo pode ser modificado de acordo com a estrutura de dados do problema.

A Estratégia Evolutiva (ES, do inglês Evolution Strategies) foi desenvolvida por Rechenberg (1973). Schwefel (1995) apresenta os métodos desenvolvidos por Rechenberg (Estratégia Evolutiva para dois indivíduos e Estratégia Evolutiva para muitos indivíduos) e os compara com métodos tradicionais, mostrando as vantagens e desvantagens de sua aplicação.

Os fundamentos e características dos GA e ES serão tratados no Capítulo 4. A implementação computacional dos algoritmos desenvolvidos a partir dos conceitos destes dois métodos será mostrada no Capítulo 5.

A otimização da confiabilidade para uma expansão de uma rede computacional, usando GA, foi desenvolvida por Kumar, Pathak e Gupta (1995). Quatro problemas de otimização foram testados, onde a confiabilidade deveria ser maximizada para certas restrições, ou outros parâmetros, como a distância média, deveria ser minimizado. O GA sofreu algumas modificações

em relação ao modelo clássico e foi chamado de GANE (Genetic Algorithm for Network Expansion).

Painton e Campbell (1995) desenvolveram um GA para a otimização do MTBF (Tempo Médio entre Falhas) de um PC doméstico. O GA mostrou-se robusto a ruídos estatísticos e ótimos locais, o que é uma característica importante na otimização da confiabilidade.

O trabalho de Rubinstein et al. (1997) desenvolveu a otimização de redundância, baseada na confiabilidade, de um sistema série-paralelo com incertezas nos parâmetros do sistema. O método de otimização utilizado foi o GA.

O trabalho de Levitin (2001) considerou o problema de otimização de redundância para um sistema com vários estágios de elementos que consomem uma quantidade fixa de recursos para atingir seus objetivos, ou um número de recursos gerados por subsistemas. Utilizou-se GA para encontrar a melhor estrutura do sistema, através da escolha de elementos que são caracterizados por custo, produtividade e disponibilidade. A função objetivo é minimizar a soma de investimentos e custo, tendo como restrição a disponibilidade do sistema.

Hsieh, Chen e Bricker (1998) desenvolveram um trabalho, onde GA foi usado para maximizar a confiabilidade de sistemas complexos, tendo como variáveis de otimização o número de redundância e a confiabilidade de cada componente. As funções de restrição consideradas foram custo, volume e peso. O problema de otimização estudado caracteriza-se por conter variáveis reais e inteiras. Uma grande vantagem dos GA é a possibilidade de se usar uma codificação binária, fazendo com que números reais e inteiros, transformados em cadeia binária, possam interagir entre si.

Devido ao desenvolvimento de novas técnicas de otimização, este trabalho visa comparar alguns destes métodos, a fim de determinar o método que melhor se adequa aos problemas de otimização de sistemas redundantes.



## **Capítulo 3**

### **Conceitos de Confiabilidade associados à Redundância de Sistemas**

Define-se confiabilidade como sendo a probabilidade de um dispositivo, ou sistema, desempenhar suas funções de projeto adequadamente, por um determinado período de tempo, sob condições de operação estabelecidas.

Um dos elementos básicos desta definição é a probabilidade. Associa-se, então, a confiabilidade a conceitos estatísticos. Diversas distribuições estatísticas podem ser usadas para modelar o comportamento desta probabilidade. Serão destacadas as distribuições exponencial e de Weibull, pois apresentam características pertinentes a este trabalho.

O desempenho, outro elemento destacado na definição de confiabilidade, é associado a um padrão, cujas variações de valores em torno do mesmo são designadas tolerâncias, que por sua vez, definem as especificações de engenharia.

O tempo esperado pode ser entendido como tempo ou período determinado. Ele está relacionado com posições ou restrições gerenciais, como tempo permitido de operação.

Outros aspectos importantes, ligados a definição de confiabilidade, são as condições de operação, que estão associadas a condições de carga, capacidade, ou mesmo condições ambientais que, eventualmente, possam influenciar os resultados.

### 3.1 Definições Gerais

Além da definição de confiabilidade, a qual foi tratada no item anterior, outros conceitos são importantes, e devem ser explorados, para o entendimento dos problemas de otimização de sistemas redundantes.

A probabilidade de falha é o complementar da confiabilidade. Portanto, é definida como sendo a probabilidade de que um dispositivo, ou sistema, falhe, ou deixe de desempenhar suas funções de projeto, em um período temporal definido, sob certas condições operacionais.

Sendo a confiabilidade, para um período de tempo  $T$ , representada por  $R(T)$ , a probabilidade de falha  $Q(T)$  é dada por:

$$Q(T) = 1 - R(T) \quad 3.1$$

A taxa de falha  $\lambda$  é uma importante função da confiabilidade e é definida como sendo o número de ocorrências de falha em um período de tempo dado. A análise do comportamento da taxa de falha de um equipamento por um longo período de tempo pode ser representada por uma curva que possui a forma de uma banheira. Por isto, conhecida como Curva da Banheira.

Como observado na Figura 3.1, há três regiões distintas nesta curva. A Região I corresponde as falhas no início de funcionamento, que surgem devido a problemas no uso inicial do sistema. Esta fase é conhecida como falhas de juventude

A Região II representa o tempo de vida útil do componente ou sistema. Neste período, as falhas ocorrem de forma aleatória. A taxa de falha constante é uma característica de componentes

eletrônicos. Em sistemas mecânicos, esta região não é percebida, pois os desgastes inerentes a tais sistemas e os processos de fadiga de material já colocam o equipamento em um processo de degradação, que caracteriza a Região III, também conhecida como falhas de velhice.

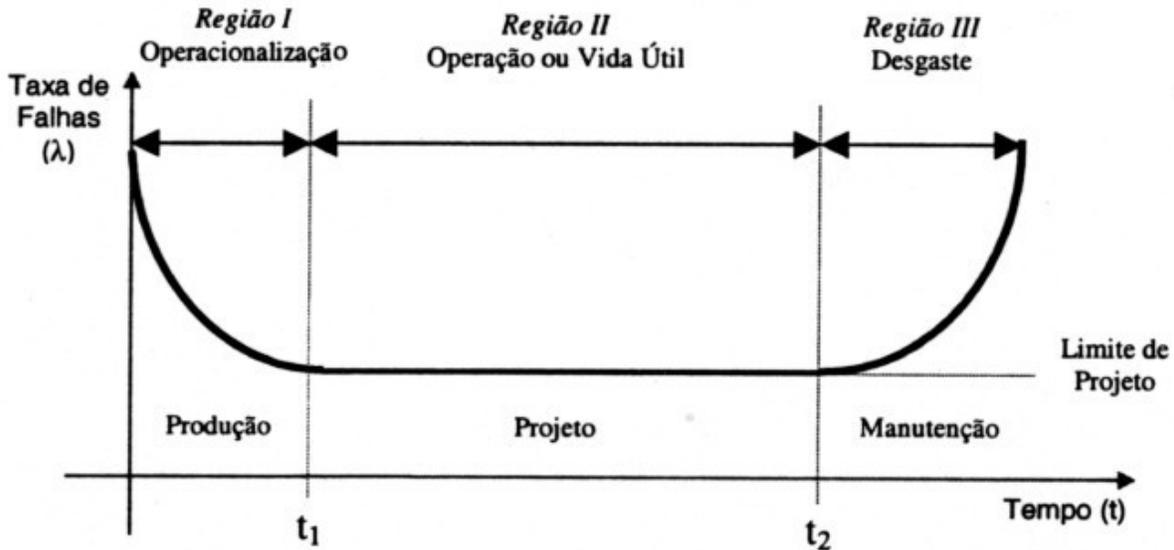


Figura 3.1 - Curva da Banheira

Devido aos desgastes ocorridos na Região III, são necessárias intervenções de manutenção. A intervenção de manutenção pode ser corretiva, quando envolve uma ação no sentido de restaurar um item logo após uma falha, para uma condição especificada, ou preventiva, quando envolve uma ação periódica no sentido de manter um item operando em uma determinada condição.

A grandeza probabilística que mede a capacidade de manter ou recolocar um componente, ou uma unidade, em um estado que permita seu bom desempenho, nas condições de operação determinadas, é a *manutenabilidade*. Ela pode ser definida como sendo a probabilidade de que um componente, ou sistema, seja reparado e recolocado em funcionamento, dentro de um período de tempo determinado. Sendo este período representado por  $T$ , a *manutenabilidade* é dada por  $M(T)$ .

A taxa de reparo  $\mu$  é a razão entre o número de reparos e o tempo necessário para efetuar tais reparos.

Outra importante função em probabilidade é a função densidade de falha, representada por  $f(t)$  e dada pela seguinte expressão:

$$f(t) = -\frac{d}{dt}R(t) = \frac{d}{dt}Q(t) \quad 3.2$$

A taxa de falha  $\lambda$  relaciona-se com a confiabilidade  $R$  e a função densidade de falha  $f$  segundo a Equação 3.3.

$$\lambda(t) = \frac{f(t)}{R(t)} \quad 3.3$$

O valor esperado da variável aleatória contínua denominada tempo até falha, é definido como tempo médio até a falha, ou simplesmente, MTTF. Em muitas situações práticas, o conhecimento do MTTF é suficiente para caracterizar a qualidade e disponibilidade de um certo item.

Da mesma forma, o tempo médio até o reparo é definido. A soma destes dois tempos resulta no tempo médio entre falhas, MTBF.

Analiticamente, o MTTF pode ser dado pela seguinte expressão:

$$MTTF = \int_0^{\infty} R(t)dt \quad 3.4$$

Assim como o MTTF depende da confiabilidade, o MTTR depende da Manutenibilidade, e pode ser expresso pela 3.5:

$$MTTR = \int_0^{\infty} (1 - M(t)) dt \quad 3.5$$

As expressões representadas pelas Equações 3.1, 3.2 e 3.3 fornecem relações entre todas as funções de confiabilidade. Portanto, o conhecimento de apenas uma, possibilita a determinação das demais. As distribuições estatísticas assumem expressões simbólicas, as quais aproximam os dados representados pelas amostras reais.

### 3.2 Distribuições Estatísticas

São várias as funções que podem modelar a distribuição probabilística de uma variável aleatória. A escolha do modelo matemático estatístico a ser utilizado está diretamente relacionada aos tipos de testes de falhas realizados, bem como ao tamanho e tipo de amostragem analisada.

Serão destacadas as distribuições exponencial e de Weibull, pois a primeira apresenta taxas de falha e reparo constantes, além de se adequar a equipamento com longa vida útil (eletrônicos), e a segunda simula equipamentos mecânicos. Entretanto, outras distribuições estatísticas são importantes para modelar determinadas características das variáveis aleatórias estudadas.

A distribuição binomial é mais adequada a problemas confiabilísticos do tipo combinatório, geralmente aplicada a variáveis discretas e grandes amostras. Uma condição importante para sua aplicação é que o número de ensaios ou triagens deve ser fixo, com apenas duas possibilidades de resposta: sucesso ou fracasso.

Outra distribuição aplicada a variáveis discretas é a distribuição de Poisson. Ela se aplica a análise de partes defeituosas, o que corresponde, essencialmente, a um controle de qualidade

utilizado para prever a probabilidade de ocorrência de defeitos, num intervalo de tempo contínuo, para melhor controle da produção.

Variáveis aleatórias contínuas podem ser modeladas por uma série de distribuições estatísticas. A distribuição de Rayleigh caracteriza-se por ter uma taxa de falha que varia linearmente, e pode aproximar as regiões de falha de juventude e desgaste da curva da banheira.

A distribuição normal constitui a mais importante lei das probabilidades em estatísticas, e forma o fundamento teórico para muitas outras distribuições de probabilidade. Em confiabilidade, a distribuição normal é utilizada para analisar produtos durante o início de vida e na fase de degradação natural, ou ainda, falhas por desgaste ou fadiga.

A distribuição Gamma aplica-se na modelagem de reparo ideal, enquanto que a distribuição Lognormal aplica-se em tempos de reparo para uma manutenção normal de falhas por desgastes.

Há outras distribuições estatísticas, como as distribuições Retangular, Beta e Valores Extremos. Entretanto, estas distribuições não têm relevância neste estudo.

### **3.2.1 Distribuição Exponencial**

A distribuição Exponencial caracteriza-se por uma taxa de falha  $\lambda$  constante. É uma distribuição uniparamétrica, onde o parâmetro principal é a própria taxa de falha  $\lambda$ .

Sua aplicação se dá no período de vida útil do item analisado. Ela se mostra muito eficiente na análise de componentes eletrônicos, que em condições e ambientes normais de operação, praticamente, não sofrem processos de degradação e possuem longa vida útil.

Considerando as relações dadas pelas Equações 3.1 a 3.3 e a função taxa de falha constante:

$$\lambda(t) = \lambda \quad 3.6$$

Obtém-se as demais funções de confiabilidade:

$$R(t) = e^{-\lambda \cdot t} \quad 3.7$$

$$Q(t) = 1 - e^{-\lambda \cdot t} \quad 3.8$$

$$f(t) = \lambda \cdot e^{-\lambda \cdot t} \quad 3.9$$

O MTTF é um valor constante e o inverso da taxa de falha:

$$MTTF = \frac{1}{\lambda} \quad 3.10$$

Semelhantemente, o MTTR é o inverso da taxa de reparo, caso seja distribuída exponencialmente:

$$MTTR = \frac{1}{\mu} \quad 3.11$$

Esta característica do MTTF e MTTR será essencial nos modelos propostos para as análises de disponibilidade e dependabilidade, itens 3.3 e 3.4 respectivamente.

### 3.2.2 Distribuição de Weibull

A distribuição de Weibull é, geralmente, uma distribuição bi-paramétrica. Os parâmetros que caracterizam esta distribuição são o parâmetro de escala  $\alpha$  (ou  $\delta$ , para análise de manutenção) e o parâmetro de forma  $\beta$  (ou  $\gamma$ , para análise de manutenção).

A função taxa de falhas, para  $\alpha > 0$ ,  $\beta > 0$  e  $t \geq 0$  é dada por:

$$\lambda(t) = \frac{\beta \cdot t^{\beta-1}}{\alpha^\beta} \quad 3.12$$

Considerando as relações entre as funções de confiabilidade dadas pelas Equações 3.1 a 3.3, obtém-se:

$$R(t) = e^{\left[-\left(\frac{t}{\alpha}\right)^\beta\right]} \quad 3.13$$

$$Q(t) = 1 - e^{\left[-\left(\frac{t}{\alpha}\right)^\beta\right]} \quad 3.14$$

$$f(t) = \frac{\beta \cdot t^{\beta-1}}{\alpha^\beta} \cdot e^{\left[-\left(\frac{t}{\alpha}\right)^\beta\right]} \quad 3.15$$

A distribuição de Weibull se adequa melhor a sistemas mecânicos do que a distribuição exponencial, pois esta última não consegue simular os efeitos de desgaste e fadiga, inerentes a tais sistemas.

A variação do parâmetro de forma  $\beta$  faz com que a distribuição de Weibull se ajuste a vários modelos físicos. Portanto, a distribuição de Weibull é amplamente utilizada em confiabilidade.

Caso o parâmetro de forma  $\beta$  seja igual a unidade. A distribuição de Weibull se iguala a uma distribuição exponencial, onde a taxa de falha é dada por:

$$\lambda(t) = \frac{1}{\alpha} \quad 3.16$$

Para  $\beta$  igual a 2, a distribuição de Weibull se reduz a uma distribuição de Rayleigh. Para  $\beta$  igual a 3,4 a distribuição de Weibull se aproxima a normal (geralmente este valor varia de 2,7 a 3,7).

Como citado no item anterior, valores constantes de MTTF e MTTR, característica da distribuição exponencial, serão considerados nas propostas nos itens seguintes. Entretanto, isto não é válido para a distribuição de Weibull. As expressões seguintes (Equações 3.17 e 3.18), fornecem um valor constante aproximado de MTTR e MTTR, considerando os parâmetros da distribuição de Weibull.

$$MTTF = \alpha \cdot \Gamma\left(\frac{1}{\beta} + 1\right) \quad 3.17$$

$$MTTR = \delta \cdot \Gamma\left(\frac{1}{\gamma} + 1\right) \quad 3.18$$

Onde a função gamma  $\Gamma(a)$  é dada por:

$$\Gamma(a) = \int_0^{\infty} e^{-x} \cdot x^{a-1} dx \quad 3.19$$

Caso a variável  $a$  seja inteira:

$$\Gamma(a) = (a-1)! \quad 3.20$$

A curva de Weibull pode ser expressa por uma reta em uma escala bi-logarítmica. Considerando a Equação 3.14, esta reta pode ser escrita na forma:

$$\ln \left[ \ln \left( \frac{1}{1-Q(t)} \right) \right] = \ln \left[ \left( \frac{t}{\alpha} \right)^\beta \right] = \beta \cdot \ln \left( \frac{t}{\alpha} \right) = \beta \cdot \ln(t) - \beta \cdot \ln(\alpha) \quad 3.21$$

Através desta reta, os dados experimentais de  $Q(t)$  podem ser ajustados a uma reta e os parâmetros da distribuição de Weibull podem ser obtidos a partir dos coeficientes angular e linear desta reta.

### 3.3 Análise de Disponibilidade

#### 3.3.1 Cálculo da Disponibilidade

Uma importante grandeza em sistemas reparáveis é a Disponibilidade. A Disponibilidade Instantânea  $A(t)$  é definida como sendo a de que um sistema esteja disponível para o uso no tempo  $t$ .

É necessário, entretanto, definir também a disponibilidade em um intervalo de tempo, e não apenas em um instante. Considerando  $T$  como sendo este intervalo, temos:

$$A^*(T) = \frac{1}{T} \int_0^T A(t) dt \quad 3.22$$

Este intervalo T deve ser projetado como sendo o tempo de vida útil do produto ou o tempo para cumprir uma determinada tarefa.

Se após um tempo inicial, a disponibilidade instantânea assumir um valor independente do tempo, podemos ter uma disponibilidade em regime permanente, a qual pode ser obtida fazendo o limite de T tendendo ao infinito na Equação 3.22.

$$A = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T A(t) dt \quad 3.23$$

Ou então, através da definição de disponibilidade, podemos deduzir que a disponibilidade em regime permanente é a razão entre o tempo de uso pelo tempo total de vida do equipamento.

$$A = \frac{\text{tempo de uso}}{\text{tempo total}} = \frac{\text{tempo de uso}}{\text{tempo de uso} + \text{tempo de reparo}} \quad 3.24$$

Considerando o tempo de uso como sendo o MTTF e tempo de reparo igual ao MTTR, temos:

$$A = \frac{MTTF}{MTTF + MTTR} \quad 3.25$$

Se a taxa de falha  $\lambda$  e a taxa de reparo  $\mu$  forem constantes (Distribuição Exponencial), podemos considerar as Equações 3.10 e 3.11. Desta forma, a Equação 3.25 se reduz a:

$$A = \frac{\mu}{\mu + \lambda} \quad 3.26$$

Para determinar a equação da disponibilidade instantânea, devemos considerar duas funções complementares, a disponibilidade  $A(t)$  e a “não-disponibilidade”  $\tilde{A}(t)$ .

$$A(t) + \tilde{A}(t) = 1 \quad 3.27$$

Devemos considerar as mudanças na disponibilidade entre  $t$  e  $t + \Delta t$ . As falhas no intervalo  $\Delta t$  serão  $\lambda \Delta t$ , portanto a perda de disponibilidade será  $\lambda \Delta t A(t)$ . Os reparos no mesmo intervalo serão  $\mu \Delta t$ , então o ganho de disponibilidade será de  $\mu \Delta t \tilde{A}(t)$ . Portanto, temos:

$$A(t + \Delta t) - A(t) = \mu \Delta t \tilde{A}(t) - \lambda \Delta t A(t) \quad 3.28$$

Fazendo  $\tilde{A}(t) = 1 - A(t)$  e dividindo a equação anterior por  $\Delta t$ , temos:

$$\frac{A(t + \Delta t) - A(t)}{\Delta t} = -(\lambda + \mu)A(t) + \mu \quad 3.29$$

Se  $\Delta t \rightarrow 0$ , obtém-se a seguinte equação diferencial:

$$\frac{d}{dt} A(t) = -(\lambda + \mu)A(t) + \mu \quad 3.30$$

Resolvendo a Equação 3.30, obtemos a disponibilidade instantânea:

$$A(t) = \frac{\mu}{\mu + \lambda} + \frac{\mu}{\mu + \lambda} \cdot e^{[-(\mu + \lambda)t]} \quad 3.31$$

Para elevados valores de tempo, o termo variante da expressão acima tende a zero, o que comprova a Equação 3.26.

### 3.3.2 Disponibilidade e Projeto

Ertas (1993) levantou uma importante relação entre disponibilidade e custo, o que afeta diretamente a performance de projeto. A Figura 3.2, a seguir, mostra a relação entre disponibilidade, custo e performance de projeto.

Deve-se escolher o ponto de melhor relação custo benefício, pois a disponibilidade cai com o aumento da performance e o custo aumenta com a disponibilidade. O perfil 2 apresenta uma solução razoável, pois apresenta valores médios entre disponibilidade, custo e desempenho do projeto.

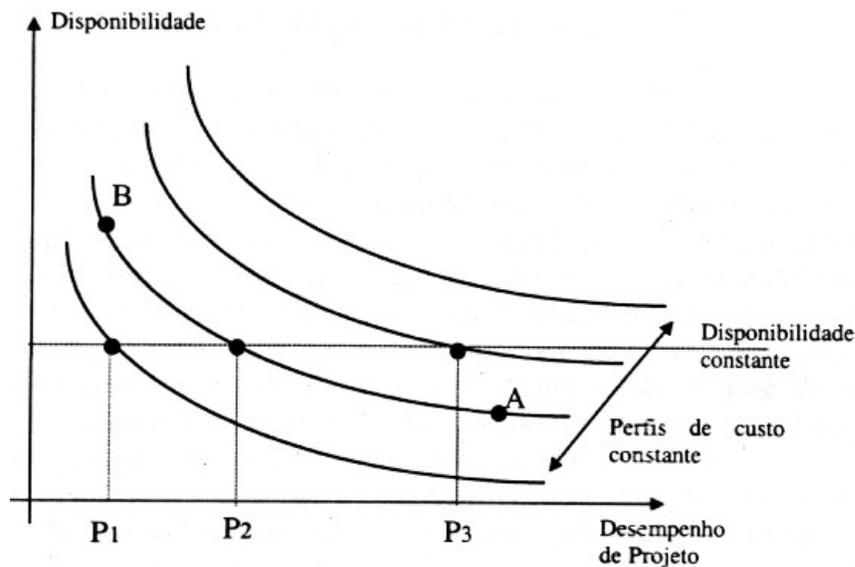


Figura 3.2 – Disponibilidade como parâmetro de Projeto

Em sistemas onde é necessário alta confiabilidade e performance elevada, o custo global do sistema é elevado, pois as operações de manutenção, assim como os métodos de melhoria devem receber investimentos. Portanto, processos de otimização devem ser aplicados para se obter o máximo destes parâmetros dentro dos limites de custo determinados.

Por outro lado, quando a disponibilidade e a performance do sistema não são pontos críticos no projeto, podendo atingir patamares mínimos pré-determinados, faz-se necessário o uso de métodos de otimização, a fim de se obter valores de custo mínimos, sem que os limites de disponibilidade e performance sejam quebrados.

Levando em conta a Equação 3.25 determina-se uma relação linear entre o MTTF e o MTTR:

$$MTTR = \left( \frac{1-A}{A} \right) MTTF \quad 3.32$$

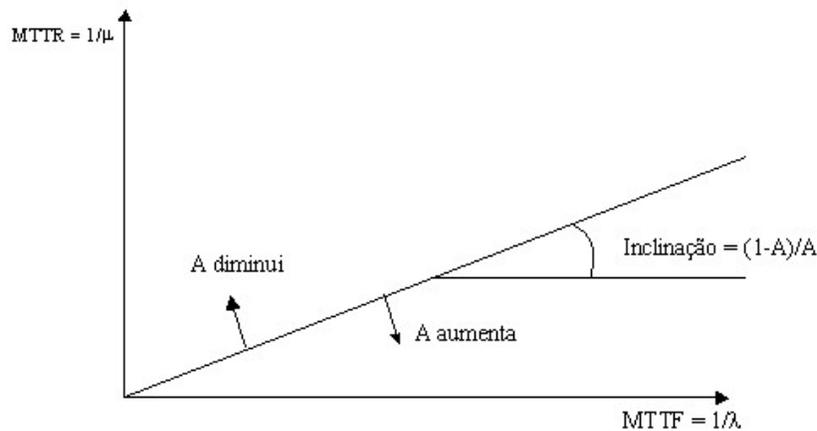


Figura 3.3 - Relação Linear entre MTTF e MTTR

Assumido um valor mínimo para o MTTF, ou seja, o tempo mínimo em que o sistema deverá estar disponível para o uso, um valor máximo para o MTTR, tempo máximo em que o sistema permanece sem o reparo necessário, e uma disponibilidade mínima permitida, pode-se

traçar a região economicamente aceita, onde o sistema poderá atuar, sem que os limites econômicos sejam quebrados.

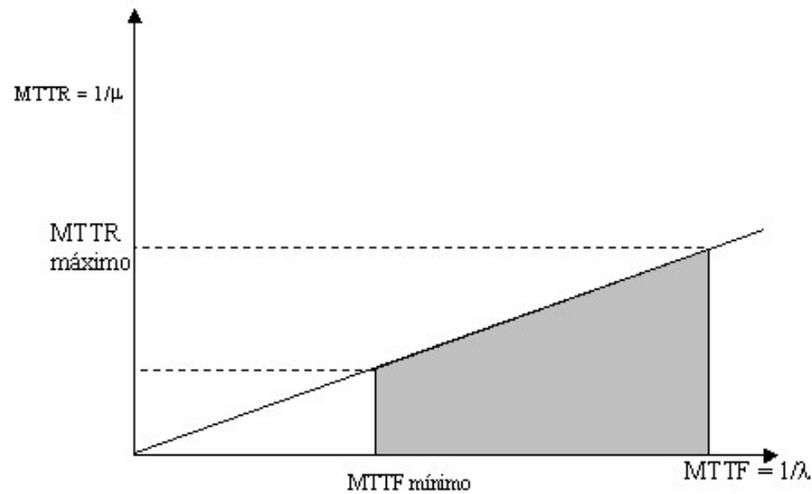


Figura 3. 4 - Região Economicamente Aceita

### 3.4 Análise de Dependabilidade

Um outro importante parâmetro de projeto é a dependabilidade, pois provê uma única medida de condições, através da combinação dos tempos de falha e manutenção. Este conceito pode ser definido como a probabilidade que um componente não falhe, ou falhe e seja reparado, dentro de um intervalo temporal aceitável. Uma característica importante da dependabilidade é a permissão da análise de custo entre a confiabilidade e manutenibilidade.

A razão de dependabilidade é a razão entre o MTTF e o MTTR. Se considerado uma distribuição estatística, esta razão pode ser escrita em termos da taxa de falha  $\lambda$  e a taxa de reparo  $\mu$ .

$$d = \frac{MTTF}{MTTR} = \frac{\mu}{\lambda} \quad 3.33$$

Quanto maior for o valor de  $d$ , maior será a necessidade de manutenção no sistema. Portanto, aliado a valores de custo, pode se determinar a viabilidade, ou não, do equipamento.

A relação entre a disponibilidade e a razão de dependabilidade pode ser obtida da combinação entre as Equações 3.25 e 3.33:

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{MTTR \cdot d}{MTTR \cdot (1 + d)} = \frac{d}{1 + d} \quad 3.34$$

Observa-se na Figura 3.5, um crescimento muito grande da razão de dependabilidade para disponibilidade acima de 0,9 e um decréscimo acentuado para disponibilidade inferior a 0,1. Isto mostra uma sensibilidade muito grande da razão de dependabilidade nestas regiões. Analisando a região onde a disponibilidade é superior a 0,9, nota-se que é necessário um aumento excessivo da manutenção para um crescimento pequeno acréscimo da disponibilidade, o que acarreta um custo alto. Portanto, a região onde a disponibilidade varia de 0,1 a 0,9 é mais adequada para se trabalhar no aumento da razão de dependabilidade.

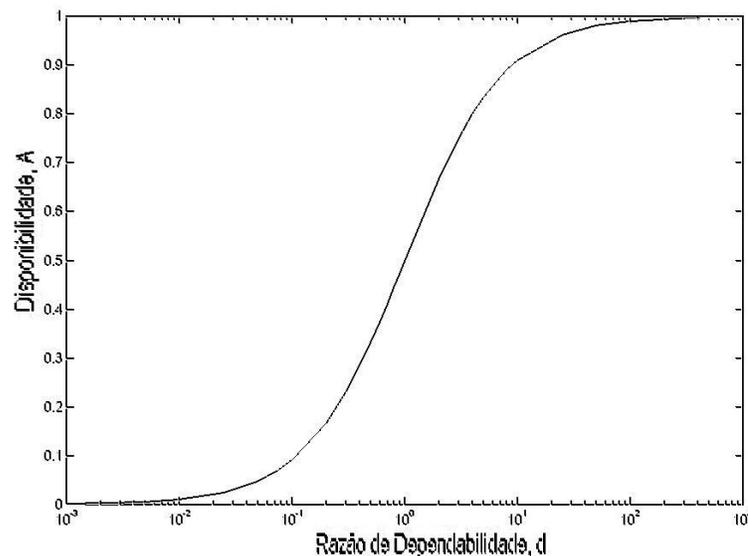


Figura 3.5 - Variação da Disponibilidade pela Razão de Dependabilidade

A razão de dependabilidade pode servir como restrição de um problema de otimização, onde se busca o incremento ótimo na disponibilidade dos componentes de um sistema, determinando um valor máximo para a mesma que seja viável economicamente.

Este incremento pode ser obtido na melhoria dos processos de manutenção, que resultaria em tempos de reparo menores.

Outra possibilidade seria o aumento dos tempos até a falha. Métodos de melhoria poderiam ser utilizados para se detectar os principais mecanismos de falha e propor soluções viáveis para os mesmos.

Estas opções acarretam um aumento de custo, o que deve ser considerado para se determinar os valores limites da razão de dependabilidade.

Dependabilidade temporal pode ser definida por:

$$D(t) = 1 - \left( \frac{1}{d-1} \right) \left( e^{-\lambda t} \right) \left[ 1 - e^{-(d-1)\lambda t} \right] \quad 3.35$$

Considerando  $\lambda t$ , que pode representar o MTBF (Tempo Médio entre Falhas), a variável da equação, temos a dependabilidade mínima como sendo:

$$D_{MIN} = 1 - \left( \frac{1}{d-1} \right) \left( e^{-\ln d / d-1} - e^{-d \ln d / d-1} \right) \quad 3.36$$

A variação da dependabilidade temporal pelo  $\lambda t$ , pode ser vista na Figura 3.6, a seguir, para vários valores da razão de dependabilidade,  $d$ .

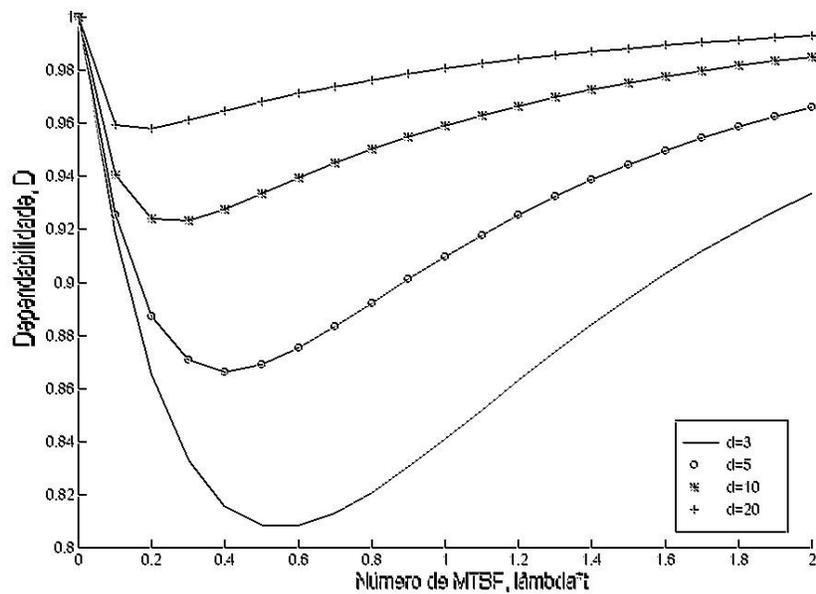


Figura 3.6 - Variação da Dependabilidade pelo MTBF ( $\lambda t$ )

Quanto maior for a razão de dependabilidade, menor é a probabilidade de que o sistema falhe, e que após a falha, os processos de manutenção não atinjam seu objetivo dentro do limite de tempo esperado. Isto se deve ao aumento dos tempos de falha e a queda dos tempos de manutenção, que resultam no aumento da razão de dependabilidade.

### 3.5 Sistemas redundantes

O aumento da confiabilidade (ou disponibilidade) de um sistema pode ser obtido de duas formas: ou o da confiabilidade (ou disponibilidade) dos componentes do sistema, ou a utilização de redundância de componentes ou sistemas.

Na redundância de sistemas, a função de projeto é exercida por dois, ou mais sistemas, em paralelo, enquanto que na redundância de unidades, a função de cada componente é redundada.

As redundâncias podem ser ativas ou passivas. Redundância passiva, ou redundância *stand-by*, trata de uma estrutura disposta, logicamente, em paralelo, onde o componente redundante passa a funcionar a partir da falha do outro componente. Caso os componentes paralelos funcionem simultaneamente, considera-se redundância ativa.

O conceito de estruturas de sistemas em paralelo e sistemas em série é essencial para o estudo de sistemas redundantes.

Um conjunto de componentes é considerado disposto em série do ponto de vista da confiabilidade se o sucesso da operação do sistema depende do sucesso da operação de cada componente.

Os componentes não necessitam estar fisicamente em série. A Figura 3.7 representa a disposição de um sistema em série.



Figura 3.7 - Sistema em Série

Como o funcionamento de todos os componentes é fundamental, a confiabilidade do sistema em série é dada pelo produto da confiabilidade de seus componentes:

$$R_S = \prod_{i=1}^n R_i \quad 3.37$$

Onde n é o número total de componentes.

As considerações feitas para a confiabilidade também são válidas para a disponibilidade.

### 3.5.1 Redundância Ativa

Como citado anteriormente, uma redundância ativa constitui um conjunto de componentes em paralelo, onde o funcionamento dos mesmos não depende da falha de alguns deles.

Um conjunto de componentes é considerado disposto logicamente em paralelo, do ponto de vista da confiabilidade, se o sucesso da operação do sistema depende ao menos do sucesso de um de seus componentes. Se a confiabilidade é a medida de bom desempenho do sistema, então se refere a probabilidade de sucesso do mesmo, enquanto que a distribuição de falhas, refere-se a probabilidade de fracasso  $Q$ . Assim:

$$R_S = \prod_{i=1}^n (1 - Q_i) \quad 3.38$$

A representação lógica de um sistema em paralelo é mostrada na Figura 3.8:

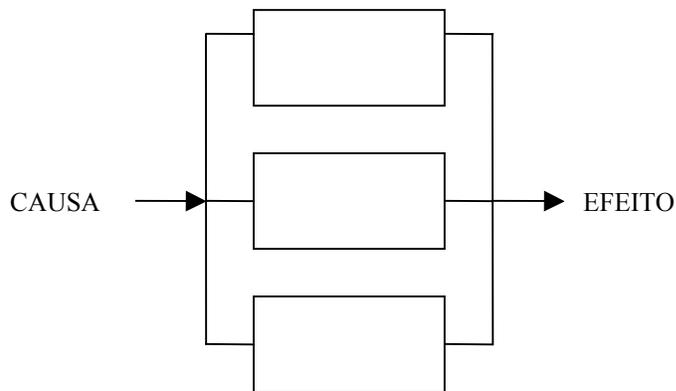


Figura 3.8 - Sistema em Paralelo

### 3.5.2 Redundância Passiva

Trata-se de uma estrutura em paralelo, onde um dos componentes passa a funcionar após a falha do outro. A taxa de falha na unidade de espera é, naturalmente, mais baixa que aquela da unidade em operação. O esquema mostrado na Figura 3.9 representa um sistema *stand-by*:

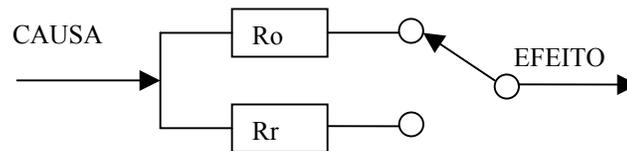


Figura 3.9 - Redundância Passiva

Considerando  $P_{ch}$  como sendo a probabilidade de sucesso de chaveamento de uma unidade para outra e  $R_{ch}$  a confiabilidade da conexão ou da chave, a confiabilidade do sistema  $R_S$  é dada, segundo Ramakumar (1993), por:

$$R_S = (1 - Q) \cdot R_{ch}$$

onde,

$$Q = Q_o - Q_o P_{ch} (1 - Q_r)$$
3.39

### 3.5.3 Redundância de Unidades

O modelo escolhido para o estudo de redundâncias de unidades considera um sistema em série, onde seus componentes são redundados ativamente.

Cada item em série pode ser considerado um sub-sistema, sendo que o número máximo de sub-sistemas é representado pela variável  $n$ . O número de redundância em cada estágio é

variável, e determinado por processos de otimização, considerando confiabilidade (ou disponibilidade), custo, peso e outros parâmetros de projeto do sistema. Este sistema é representado pela Figura 3.10.

Considerando as Equações 3.37e 3.38, obtém-se a expressão para o sistema estudado:

$$R_S = \prod_{i=1}^n [1 - (1 - R_i)^{y_i}] \quad 3.40$$

Onde  $y_i$  é o número de componentes no sub-sistema  $i$ .

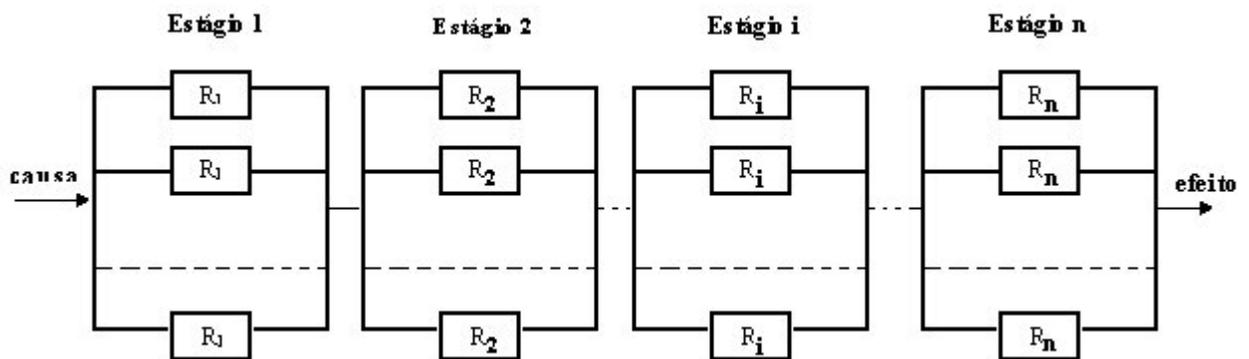


Figura 3.10 - Sistema Redundante

O aumento de componentes redundantes acarreta em aumento da confiabilidade do sistema. Por outro lado, incrementam o custo de implementação. O peso do sistema e o volume ocupado também crescem. Estes parâmetros de projeto não são desejáveis em muitos dos sistemas de engenharia.

Na indústria aeronáutica e aeroespacial é necessário que o peso seja mínimo, devido à segurança dos usuários de aviões e ao aumento de custo, pois mais potência seria necessária para

sustentar mais massa. Por outro lado, sistemas mais confiáveis oferecem maior segurança, o que é essencial, pois vidas humanas estão envolvidas.

Em muitos sistemas de engenharia, o espaço disponível é reduzido, o que limita a possibilidade de se redundar o sistema excessivamente.

A Equação 3.40 também pode ser utilizada para a disponibilidade. Desta forma, pode-se obter o número ótimo de redundâncias para sistemas, onde a disponibilidade é um parâmetro de projeto considerado.

Além disto, pode-se otimizar, simultaneamente, a disponibilidade dos componentes de cada subsistema, buscando o aumento da razão de dependabilidade, a qual depende dos tempos de falha e reparo.



## Capítulo 4

### Métodos de Otimização

#### 4.1 Conceitos Gerais

Problemas associados ao *design* ótimo têm sido considerados com bastante atenção nos últimos anos. Durante as décadas passadas o campo da otimização teve um considerável progresso em relação ao projeto de sistemas, controle de mecanismos e análise de engenharia. Com os recentes avanços da tecnologia computacional, muitos métodos de otimização modernos têm sido desenvolvidos.

Existem dois tipos de efeitos inerentemente associados a qualquer elemento mecânico ou sistema:

- Efeitos indesejáveis como alto custo, peso excessivo etc.
- Efeitos desejáveis como uma longa vida útil, eficiente energia de saída, boa capacidade de transmissão de potência e boa capacidade de refrigeração etc.

*Design* ótimo pode ser definido como a melhor configuração possível do ponto de vista do mais significativo efeito, ou seja, minimizando o mais significativo dos efeitos indesejáveis e/ou maximizando o mais significativo dos efeitos desejáveis.

A aplicação da otimização para um problema de projeto requer a formulação de uma função objetivo como o peso, custo ou forma, e as expressões para as restrições de projeto, como equações ou inequações. A função objetivo,  $U$ , em termos de variáveis independentes (variáveis de projeto) da qual uma solução ótima é procurada, pode ser escrita como segue:

$$U = U(x_1, x_2, \dots, x_n) \quad 4.1$$

Onde  $x$  é a variável de projeto e  $n$  é o número de variáveis de projeto.

Em um processo de otimização, existem certas condições a serem satisfeitas pelas variáveis de projeto. Estas condições ou limitações são chamadas de restrições. A procura da solução ótima para uma dada função objetivo, obedece as seguintes equações e inequações,

$$\begin{aligned} h_i(x) = h_i(x_1, x_2, \dots, x_n) &= 0 & i = 1, 2, \dots, m \\ g_j(x) = g_j(x_1, x_2, \dots, x_n) &\geq 0 & j = 1, 2, \dots, p \end{aligned} \quad 4.2$$

Onde  $h_i(x)$  e  $g_j(x)$  são as equações e inequações de restrições, respectivamente. A variável de projeto pode ser uma propriedade de um material, dimensões de estruturas (espessura, largura e comprimento) ou dados geométricos (coordenadas de pontos de nós de uma estrutura).

Problemas de otimização podem envolver o seguinte:

1. Somente restrições na forma de equações.

2. Somente restrições na forma de inequações.
3. Restrições dos dois tipos anteriores, 1 e 2.
4. Sem restrições.

Para o caso (4), o qual é chamado de problema de otimização sem restrições ou limitações de projeto, não existem imposições de restrição às variáveis.

Em geral existem dois tipos de problemas de otimização:

1. Problemas de otimização com único critério - no qual o objetivo de projeto é minimizar ou maximizar somente uma função objetivo. Nesta situação o ponto ótimo é simplesmente um máximo ou mínimo.
2. Problemas de otimização multicritério ou multiobjetivo - no qual o objetivo é minimizar ou otimizar mais do que uma função objetivo simultaneamente. Nesta situação, todas as funções objetivo devem ser consideradas para encontrar a solução ótima.

Diversos métodos de otimização têm sido desenvolvidos desde o estabelecimento da teoria do cálculo diferencial. Siddal (1982) e Arora (1989) apresentam alguns métodos consagrados em otimização de sistema.

Cada método é mais adequado a um certo tipo de problema, devido a sua natureza de operação e base teórica. Métodos simbólicos, como o método do cálculo diferencial, ou o método dos Multiplicadores de Lagrange, aplicam-se muito bem a problemas onde as funções que determinam o problema são diferenciáveis e bem determinadas analiticamente.

Segundo Schwefel (1995), métodos numéricos ou diretos são aqueles onde a solução ótima é obtida por iterações ou passos, e a cada passo o valor da função objetivo aproxima-se do ótimo. Por outro lado, os métodos analíticos, ou indiretos, são aqueles onde a solução ótima é obtida em apenas um passo.

Schwefel também classifica outros métodos. Os métodos experimentais aplicam-se a problemas onde a relação entre as variáveis de otimização e a função objetivo não é conhecida. Os métodos estáticos mantêm as condições e os parâmetros de otimização constantes, enquanto que os métodos dinâmicos variam tais condições, adaptando o processo de convergência ao sistema de otimização.

## 4.2 Método do Cálculo Diferencial

O método do cálculo diferencial usa a primeira e segunda derivadas para encontrar o(s) valor(es) máximo(s) ou mínimo(s) de uma função diferenciável,  $U(x)$ . Para encontrar um máximo ou mínimo usando este método, tome a derivada primeira da função objetivo,  $U(x)$ , e iguale esta a zero; então resolva para a variável independente  $x$ :

$$\frac{dU}{dx} = 0 \quad 4.3$$

A equação acima fornece os valores dos pontos críticos (pontos ótimos). Se a derivada segunda, calculada para o ponto crítico, for menor do que zero, então o ponto é um máximo. O ponto é um mínimo se a derivada segunda é maior do que zero.

$$\frac{d^2U}{dx^2} < 0 \quad \text{implica em um máximo local} \quad 4.4$$

$$\frac{d^2U}{dx^2} > 0 \quad \text{implica em um mínimo local}$$

A função deve ser contínua para ter um máximo ou mínimo. Se a derivada segunda for zero o ponto é um ponto de inflexão. Um gráfico esquemático de uma função qualquer pode ser visualizado na Figura 4.1.

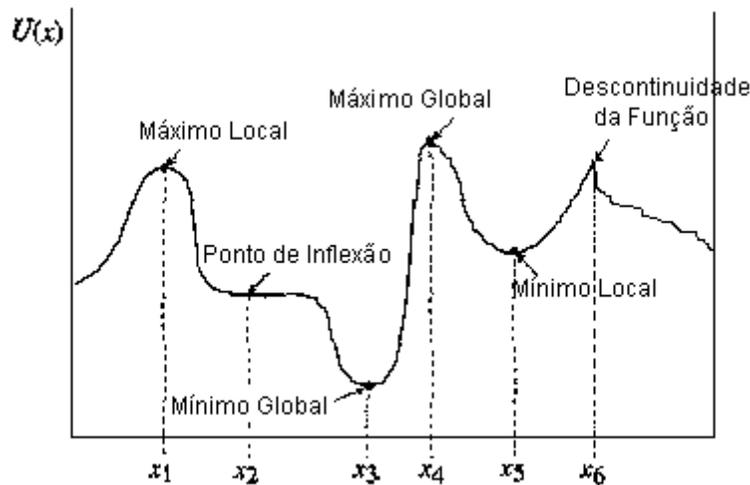


Figura 4.1 - Máximos e Mínimos locais e globais , pontos de inflexão e descontinuidade de uma Função  $U(x)$

### 4.3 Métodos de Busca para problemas com uma variável

Métodos de busca baseiam-se em examinar soluções simultâneas ou seqüenciais sobre todo o domínio da região factível para determinar qual é o ponto ótimo. Estes métodos provêm informações sobre a região na qual o ponto ótimo está alocado. Entretanto, pelo fato de que estes métodos não são exatos, em geral, uma rápida taxa de convergência não é possível.

Três métodos de busca para problemas que dependem de apenas uma única variável são apresentados a seguir: intervalos iguais, razão áurea e método de Fibonacci.

### 4.3.1 Intervalos Iguais

Assumindo que uma função objetivo  $U = U(x)$  é uma função de uma simples variável  $x$  e tem um único valor mínimo,  $U_{min}$  em  $x^*$ . Usando o método de procura de intervalos iguais, a função  $U(x)$  é calculada nos pontos com incrementos iguais  $\Delta x$ . Os valores calculados da função para dois pontos sucessivos,  $x_i$  e  $x_{i+1} = x_i + \Delta x$ , são então comparados. Quando um aumento no valor da função para qualquer ponto final,  $x_f = x_{i+1}$ , é sentido, então:

$$U(x_f - \Delta x) < U(x_f) \quad 4.5$$

O mínimo da função estará entre os pontos detectados:

$$x_f - \Delta x < x^* < x_f \quad 4.6$$

Onde  $x^*$  é o ponto mínimo de  $U(y)$ . Para reduzir o intervalo de incerteza a um valor aceitável, o processo de procura é revertido por uma mudança de sinal com um decremento reduzido pela metade. Agora o processo de procura é reiniciado, calculando o valor da função em  $x_f - \Delta x/2$  e o processo é repetido para um próximo intervalo de incerteza ainda menor, até que o valor de  $\Delta x$  seja menor que uma tolerância pré-determinada.

Uma desvantagem deste método é sua sensibilidade a máximos ou mínimos locais. Entretanto, é bem simples e fácil de ser implementado.

### 4.3.2 Razão Áurea

No método de procura com intervalos iguais um incremento  $\Delta x$  é mantido constante até que uma mudança no sentido da função seja sentida. O método de procura da razão áurea provê um procedimento alternativo, onde o incremento varia para cada passo de procura. Este método de procura encontra o valor ótimo de uma dada função sobre um intervalo especificado de comprimento  $L$ . O primeiro passo deste método é encontrar dois pontos internos ao intervalo no qual a função possa ser calculada.

Considerando um intervalo com um limite inferior  $XL$  e um limite superior de  $XU$ , como pode ser visto na

Figura 4.2, o comprimento do intervalo é dado por:

$$L = XU - XL \quad 4.7$$

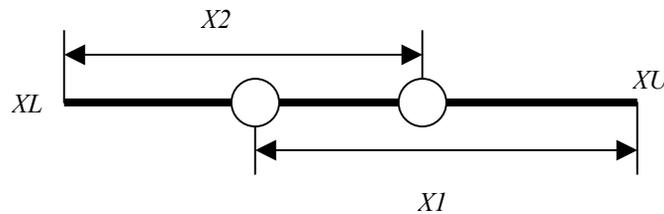


Figura 4.2 - Exemplo ilustrativo para o Método da Razão Áurea e Método de Fibonacci

Considerando  $k$  uma constante entre 0 e 1, os valores de  $x_1$  e  $x_2$  são dados por:

$$\begin{aligned} x_1 &= XL + 0,618 \cdot L \\ x_2 &= XU - 0,618 \cdot L \end{aligned} \quad 4.8$$

Como pode ser visto na Equação 4.8 acima, os pontos interiores no qual a função é calculada não são selecionados ao acaso. Eles são baseados numa razão de 0.618, conhecida como razão áurea.

Assume-se que o objetivo é encontrar o mínimo valor para a função dada. Se o valor da função calculada em  $x_1$  é maior do que o valor em  $x_2$ , a região a direita de  $x_1$  é eliminada. Um novo intervalo vai de  $XL$  até  $XU = x_1$ , e os cálculos são repetidos. Se o valor calculado em  $x_2$  é maior do que o valor calculado em  $x_1$ , a região à esquerda de  $x_2$  é eliminada. Neste caso, o novo intervalo vai de  $XL = x_2$  até  $XU$ . Este processo é contínuo até que uma desejada precisão seja obtida. A precisão é determinada pelo comprimento restante do intervalo. O contrário se aplica para encontrar o máximo da função.

### 4.3.3 Método de Fibonacci

Este método de busca, assim como o método da razão áurea, é um método que se baseia na divisão de intervalos. Ele opera a partir de uma série criada por Leonardo de Pisa, que tinha como pseudônimo o nome de Fibonacci.

A série de Fibonacci é dada por  $F_j$ , onde:

$$\begin{aligned} F_0 = F_1 &= 1 \\ F_j &= F_{j-1} + F_{j-2}, \quad \text{para } j \geq 2 \end{aligned} \tag{4.9}$$

Considerando um intervalo dado pela

Figura 4.2 e Equação 4.7, os valores intermediários  $x_1$  e  $x_2$  são dados por:

$$\begin{aligned}
 x_1 &= XL + \frac{F_{N-j-1}}{F_{N-j}} \cdot L \\
 x_2 &= XU - \frac{F_{N-j-1}}{F_{N-j}} \cdot L
 \end{aligned}
 \tag{4.10}$$

Onde  $N$  é o número máximo de iterações.

As considerações feitas para escolha dos limites do novo intervalo na seção anterior são válidas também para este método.

O número de iterações ideal para os métodos de Fibonacci e razão áurea pode ser dado por:

$$N = \ln\left(\frac{L_{inicial}}{\varepsilon}\right)
 \tag{4.11}$$

Onde  $\varepsilon$  é a tolerância ou erro mínimo.

#### 4.4 Métodos de Busca para problemas com multi-variáveis

Muitos problemas de otimização em engenharia possuem mais de uma variável. Portanto, as estratégias adotadas no item anterior não podem ser aplicadas.

Os métodos de busca multidimensionais podem ser classificados em relação ao tipo de informação que eles necessitam:

1. Métodos de Busca direta: Necessitam apenas do valor da função objetivo;
2. Métodos do Gradiente: É necessário, além do valor da função objetivo, o valor das derivadas de primeira ordem;
3. Método de Newton: Necessita-se de informações das derivadas de segunda ordem.

Além de apresentar a Estratégia Evolutiva, Schwefel (1995) apresenta muitas estratégias de busca multidimensional e as compara com os métodos evolutivos.

Um método simples de busca multidimensional é o *Steepest Descent* (para minimização) ou *Steepest Ascent* (maximização). Este método necessita do conhecimento do gradiente da função.

#### 4.4.1 Steepest Descent

Considerando um problema bidimensional, para facilitar a formulação do método, a função objetivo  $U$  é representada por:

$$U = U(x_1, x_2) \quad 4.12$$

As curvas de nível da função  $U$  podem ser obtidas para diferentes valores de  $x_1$  e  $x_2$ , como pode ser visto na Figura 4.3.

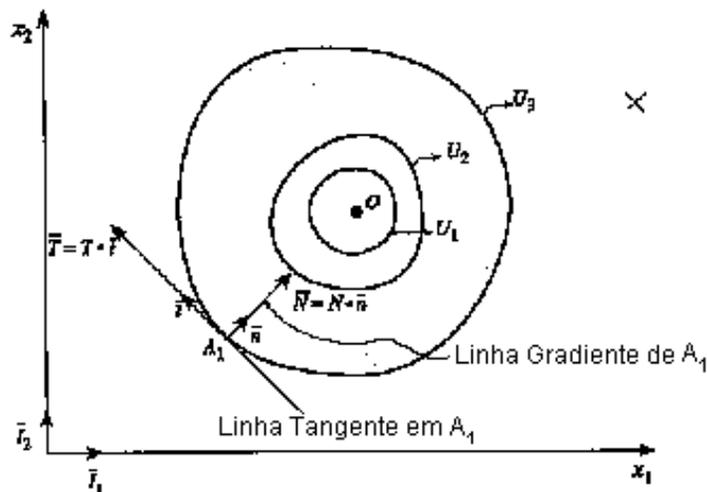


Figura 4.3 - Curvas de nível para função  $U$

O ponto  $O$  é um ponto ótimo para a função  $U$ .  $A_1$  é um ponto arbitrário de início, chamado de ponto base. Os vetores  $\bar{i}$  e  $\bar{n}$  são, respectivamente, os vetores unitários tangente e normal a uma dada curva de nível. O vetor normal no ponto  $A_1$ , chamado de vetor gradiente, é definido por:

$$\bar{N} = \nabla U = \left( \frac{\partial U}{\partial x_1} \right) \bar{i}_1 + \left( \frac{\partial U}{\partial x_2} \right) \bar{i}_2 \quad 4.13$$

Portanto, o vetor unitário pode ser escrito como:

$$\bar{n} = \frac{\left( \frac{\partial U}{\partial x_1} \right) \bar{i}_1 + \left( \frac{\partial U}{\partial x_2} \right) \bar{i}_2}{\sqrt{\left( \frac{\partial U}{\partial x_1} \right)^2 + \left( \frac{\partial U}{\partial x_2} \right)^2}} \quad 4.14$$

O que mostra a direção do gradiente como um vetor direcional para o próximo passo. O método steepest pode ser usado para problemas de busca multidimensional, com um incremento na direção do gradiente. Para mover-se na direção do vetor gradiente  $\bar{N}$ , o incremento  $\Delta$  é multiplicado pelo vetor unitário  $\bar{n}$ .

$$\bar{\Delta} = \Delta \bar{n} = \left[ \frac{\left( \frac{\partial U}{\partial x_1} \right) \Delta \bar{i}_1 + \left( \frac{\partial U}{\partial x_2} \right) \Delta \bar{i}_2}{\sqrt{\left( \frac{\partial U}{\partial x_1} \right)^2 + \left( \frac{\partial U}{\partial x_2} \right)^2}} \right] \quad 4.15$$

A equação acima pode ser escrita como:

$$\bar{\Delta} = \Delta x_1 \bar{I}_1 + \Delta x_2 \bar{I}_2 \quad 4.16$$

Onde:

$$\Delta x_1 = \frac{\left(\frac{\partial U}{\partial x_1}\right)_{A_1} \Delta}{\sqrt{\left(\frac{\partial U}{\partial x_1}\right)_{A_1}^2 + \left(\frac{\partial U}{\partial x_2}\right)_{A_1}^2}} \quad 4.17$$

$$\Delta x_2 = \frac{\left(\frac{\partial U}{\partial x_2}\right)_{A_1} \Delta}{\sqrt{\left(\frac{\partial U}{\partial x_1}\right)_{A_1}^2 + \left(\frac{\partial U}{\partial x_2}\right)_{A_1}^2}}$$

O mesmo  $\Delta$  e direção do gradiente são utilizados até que um aumento em  $U$  é percebido. A direção do gradiente é, então, mudada e  $\Delta$  é reduzido para  $\Delta/2$ . Este procedimento é repetido até que  $\Delta$  seja reduzido a um valor correspondente ao critério de convergência  $\varepsilon_1$ . A próxima direção do gradiente é, então, determinada, e os mesmos passos são repetidos até que  $\partial U/\partial x_1$  ou  $\partial U/\partial x_2$  seja reduzido a um valor que corresponda ao critério de convergência  $\varepsilon_2$ . Quando  $x_1$  e  $x_2$  alcançam um mínimo, seus valores serão iguais às raízes das funções  $\partial U/\partial x_1$  ou  $\partial U/\partial x_2$ . Então, quanto menor for a derivada parcial,  $\partial U/\partial x_i$ , mais perto estarão os valores  $x_i$  das raízes. Um procedimento semelhante é adotado para o Steepest Ascent (maximização).

#### 4.5 Multiplicadores de Lagrange

A técnica dos multiplicadores de Lagrange, como descrita classicamente, consiste na aplicação do cálculo diferencial para encontrar o máximo ou o mínimo de uma função  $U$  de variáveis contínuas  $(x_1, x_2, x_3, \dots, x_n)$  sujeitas a funções de restrições  $\psi$ . Em outras palavras, é uma técnica para encontrar as coordenadas para um ponto de projeto ideal em alguns problemas. Este

método pode ser aplicado para um sistema de equações constituído de uma função objetivo, que é a função que desejamos otimizar, e das  $m$  funções de restrição:

$$U = U(x_1, x_2, \dots, x_n) \quad 4.18$$

$$\begin{aligned} \psi_1 &= g_1(x_1, x_2, \dots, x_n) = 0 \\ \psi_2 &= g_2(x_1, x_2, \dots, x_n) = 0 \\ &\cdot \\ &\cdot \\ \psi_m &= g_m(x_1, x_2, \dots, x_n) = 0 \end{aligned} \quad 4.19$$

Deseja-se encontrar os valores  $x_i$ , onde  $k = 1, \dots, n$ , para que a derivada da função objetivo seja igual a zero,  $dU = 0$ , respeitando as funções de restrição. Então, aplicando-se a teoria de diferenciação para um dado sistema de equações, devem ser satisfeitas as condições:

$$\begin{aligned} dU &= \sum_{i=1}^n \frac{\partial U}{\partial x_k} dx_k = 0 \\ d\psi_1 &= \sum_{i=1}^n \frac{\partial \psi_1}{\partial x_k} dx_k = 0 \\ d\psi_2 &= \sum_{i=1}^n \frac{\partial \psi_2}{\partial x_k} dx_k = 0 \\ &\cdot \quad \quad \quad \cdot \\ &\cdot \quad \quad \quad \cdot \\ d\psi_m &= \sum_{i=1}^n \frac{\partial \psi_m}{\partial x_k} dx_k = 0 \end{aligned} \quad 4.20$$

Multiplica-se cada uma das equações acima, exceto a primeira, por um parâmetro indeterminado, chamado multiplicador de Lagrange,  $(v_1, v_2, \dots, v_m)$ , obtendo-se o seguinte:

$$\begin{aligned}
\nu_1 d\psi_1 &= \sum_{k=1}^n \nu_1 \frac{\partial \psi_1}{\partial x_k} dx_k = 0 \\
\nu_2 d\psi_2 &= \sum_{k=1}^n \nu_2 \frac{\partial \psi_2}{\partial x_k} dx_k = 0 \\
&\cdot \qquad \qquad \cdot \\
&\cdot \qquad \qquad \cdot \\
\nu_m d\psi_m &= \sum_{k=1}^n \nu_m \frac{\partial \psi_m}{\partial x_k} dx_k = 0
\end{aligned}
\tag{4.21}$$

Adicionando as equações, agora derivadas, obtém-se:

$$dU + \nu_1 d\psi_1 + \nu_2 d\psi_2 + \dots + \nu_m d\psi_m = 0 \tag{4.22}$$

Portanto:

$$\sum_{k=1}^n \left( \frac{\partial U}{\partial x_k} + \nu_1 \frac{\partial \psi_1}{\partial x_k} + \nu_2 \frac{\partial \psi_2}{\partial x_k} + \dots + \nu_m \frac{\partial \psi_m}{\partial x_k} \right) dx_k = 0 \tag{4.23}$$

Considerando que todas as variáveis sejam independentes entre si, satisfaz-se todas as  $n$  equações da forma:

$$\begin{aligned}
\frac{\partial U}{\partial x_k} + \nu_1 \frac{\partial \psi_1}{\partial x_k} + \nu_2 \frac{\partial \psi_2}{\partial x_k} + \dots + \nu_m \frac{\partial \psi_m}{\partial x_k} &= 0 \\
\text{Para } k &= 1, 2, \dots, n
\end{aligned}
\tag{4.24}$$

Todas as  $m$  equações de restrição devem ser satisfeitas. Então, há  $(m + n)$  equações disponíveis, formando um sistema representado pelas Equações 4.19 e 4.24. Também há  $(m + n)$  incógnitas, que são as variáveis  $(x_1, x_2, x_3, \dots, x_n)$  e os multiplicadores de Lagrange  $(v_1, v_2, \dots, v_m)$ , tornando possível solucionar o sistema de equações proposto.

#### **4.6 Programas ou Algoritmos Evolutivos**

A imitação de princípios biológicos para o desenvolvimento da tecnologia acompanha a humanidade desde os primórdios de sua história. A lenda de Dédalo e Ícaro, que construíram asas de cera, a fim de voar como os pássaros, e as asas movidas a pedal, de Leonardo da Vinci, mostram como o ser humano associa observações da natureza em suas invenções.

Os avanços no estudo da genética proporcionaram que pesquisadores associassem o processo adaptativo dos seres vivos a problemas de engenharia, como otimização, inteligência artificial, treinamento de máquinas. Ashby (1960), Rechenberg (1973) Fogel, et al (1966) e Holland (1975) desenvolveram algoritmos que são conhecidos como Programas ou Algoritmos Evolutivos.

Programas ou Algoritmos Evolutivos (EA) são métodos aleatórios de busca que simulam o processo de evolução genética, tendo como ferramentas os princípios de seleção, recombinação, reprodução e mutação.

Uma aplicação recente destes métodos é no projeto dos mais variados sistemas, pois os resultados podem ser codificados em possíveis soluções dos problemas, as quais podem ser avaliadas segundo o desempenho que cada uma produz. Bentley (1999) mostra em sua obra como o “projeto evolutivo” pode ser implementado e mostra diversos exemplos de aplicação. Um destes exemplos (Capítulo 8) é a aplicação de Algoritmo Genético em otimização de sistemas redundantes.

Alguns termos de genética assumem importante papel na representação destes algoritmos. Logo, é necessário que suas funções nestes algoritmos sejam esclarecidas.

Uma possível solução é representada por um indivíduo, o qual é representado por seu gene. Um conjunto de possíveis soluções é uma população.

Deve-se entender por genótipo a representação da variável de resposta do problema dentro da estrutura do algoritmo. Fenótipo é interpretado como sendo os valores reais da variável de resposta. Em muitos casos os genes, ou solução possível, são formados por uma cadeia de variáveis binárias. Portanto, a cadeia binária é o genótipo, enquanto que os valores reais das variáveis (números inteiros ou reais) devem ser entendidos como fenótipo. Entretanto, os genes podem ser formados pelos valores reais. Neste caso, o genótipo é igual ao fenótipo.

O processo de seleção é uma importante fase dos EA's, pois é feita a escolha dos indivíduos mais aptos a se reproduzirem ou recombinarem e participarem da próxima geração (iteração). A possibilidade de um indivíduo ser selecionado depende de seu *fitness*. O *fitness* mede o quanto um indivíduo é mais apto do que o outro. No caso da otimização, mostra o quanto um indivíduo está mais próximo do ponto ótimo, e pode ser representado pela própria função objetivo ou uma função da mesma. A seleção pode ser um processo probabilístico, ou seja, um indivíduo com melhor *fitness* tem maior probabilidade de ser selecionado, ou determinísticos, selecionando os indivíduos com melhor *fitness*.

Um importante conceito na seleção é o elitismo, o indivíduo com o melhor *fitness*, ou um grupo formado pelos melhores indivíduos, é mantido na geração seguinte, garantindo que a solução ótima de uma geração seja mantida.

Os indivíduos sofrem modificações através de mutações, onde um gene ou uma parte da carga genética é modificada. A carga genética dos indivíduos pode se recombinar por *crossover*

ou recombinação. Neste processo, dois indivíduos produzem filhos, os quais possuem carga genética dos pais recombinada. Há outros operadores nos algoritmos evolutivos como, por exemplo, a inversão, que altera a ordem da cadeia genética.

Algoritmos Evolutivos podem ser considerados paralelos quando duas populações se desenvolvem ao mesmo tempo, podendo haver migrações entre elas, ou seja, há um intercâmbio entre as populações.

A estrutura completa de um algoritmo evolutivo deve seguir a seguinte seqüência:

1. Inicialização: a população inicial (ou populações iniciais) são formadas aleatoriamente;
2. Codificação: quando o genótipo difere do fenótipo, o fenótipo deve ser calculado;
3. Avaliação: o *fitness* é calculado a partir do fenótipo;
4. Seleção: Indivíduos são selecionados de acordo com o *fitness* para gerarem a próxima geração.
5. Reprodução: Novos indivíduos são reproduzidos através de mutação, *crossover* e outros operados;
6. Seleção Negativa: Indivíduos são eliminados, ou substituídos, caso seu *fitness* seja inadequado;
7. Migração: No caso de haver mais de uma população, indivíduos podem migrar para outras populações;

8. Parada: caso o máximo número de gerações, ou algum outro critério de parada seja atingido, o programa para, senão volta ao passo 2.

A seqüência mostrada é completa e nem todos os algoritmos evolutivos seguem todos os passos. Os passos essenciais são o passo 1, 2, 5 e 8.

Destacam-se dois algoritmos evolutivos, os Algoritmos Genéticos (GA) e a Estratégia Evolutiva (ES). Outros métodos, entretanto, são bastante difundidos, como a Programação Genética e Programação Evolutiva.

#### **4.6.1 Algoritmos Genéticos**

O Algoritmo Genético é o Programa Evolutivo mais conhecido. Ele foi desenvolvido por Holland (1975) com o objetivo de explicar os processos adaptativos de sistemas naturais e projetar sistemas artificiais baseados nos sistemas naturais.

Uma grande vantagem do GA é sua robustez. Ele é insensível a ótimos locais e ruídos no espaço de busca.

A otimização é apenas uma das aplicações do GA. Este método de busca também é usado em programação automática, diagnose e predição de aspectos relacionados a máquinas (machine learning), economia, imunidade de sistemas, ecologia, genética das populações, estudo de evolução, etc.

O algoritmo apresentado por Holland sofreu algumas inovações ao longo do tempo. Mudanças nos operadores do GA e na forma de codificação foram propostas e se adaptam melhor para determinados tipos de problemas.

O Algoritmo básico pode ser representado pela Figura 4.4 a seguir:

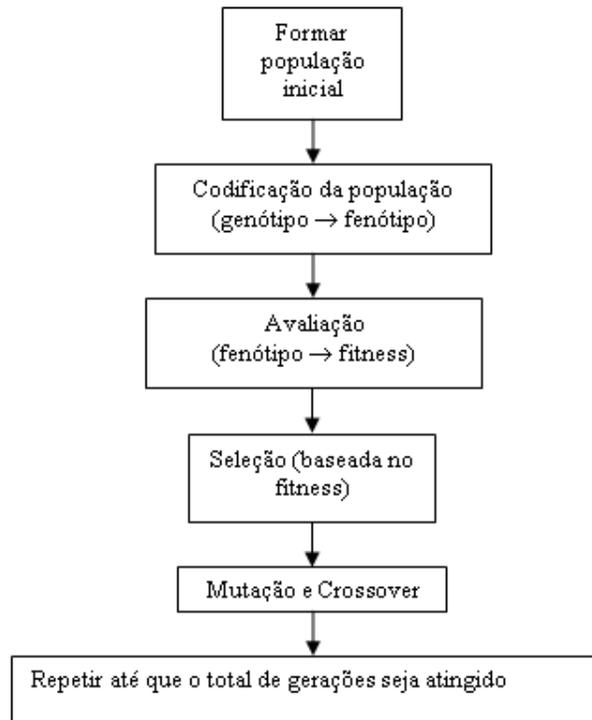


Figura 4.4 - Fluxograma do Algoritmo Genético

Os operadores básicos GA são a seleção, mutação e crossover.

A seleção proposta por Holland baseia-se no *fitness* e é feita de uma forma probabilística. O indivíduo com melhor *fitness* tem maior probabilidade de ser selecionado, ou tem uma maior fertilidade. Existem outras formas de seleção. Os principais tipos de seleção são:

- Roda de roleta: Neste método cada indivíduo é uma fatia da roda e seu tamanho é proporcional ao seu *fitness*. Desta forma, um indivíduo com um melhor *fitness* tem uma maior probabilidade de ser selecionado, pois a probabilidade que sua fatia seja sorteada é maior. Este foi o método proposto por Holland.

- Elitismo: Os indivíduos com melhor *fitness* são selecionados e farão parte da próxima geração, impedindo que o melhor resultado seja perdido.
- Seleção através do Rank: Neste caso, os indivíduos são classificados de acordo com seu fitness, descartando, porém o valor absoluto do fitness. Os indivíduos de melhor classificação possuem a maior probabilidade de serem selecionados.

Existem outras formas de seleção como a seleção de Boltzman, que possui um critério semelhante ao simulated annealing, escala sigma, seleção por torneio, seleção em regime permanente e outros. Mitchell (1996) apresenta vários métodos de seleção no Capítulo 5 de sua obra.

Os indivíduos sofrem modificações por operadores como a mutação e o crossover (ou recombinação). O primeiro método baseia-se na modificação de genes de um indivíduo, enquanto que a recombinação é baseada no cruzamento da carga genética de dois indivíduos. A Figura 4.5 exemplifica casos simples de crossover e mutação:

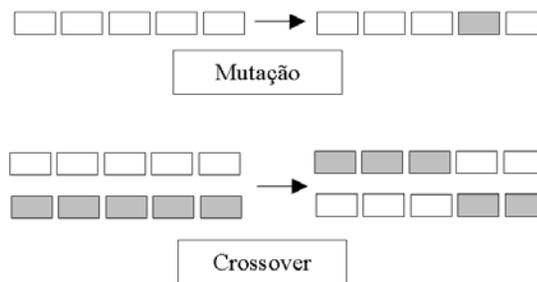


Figura 4.5 - Mutação e Crossover

Existem outras formas mais complexas de crossover, como o crossover de dois pontos, onde há dois pontos na cadeia genética e a troca é feita neste intervalo.

Outro método de transformação que foi proposto por Holland é a inversão. Este método não é uma analogia biológica, mas é aplicável em sistemas artificiais, principalmente se a codificação

for binária. Ele consiste na inversão da cadeia genética, como uma imagem em um espelho, e está exemplificado na Figura 4.6:

0100011110010110110 → 0110110100111100010

Figura 4.6 - Inversão

O critério de parada do GA é o total de gerações, que deve ser definido previamente. Outros parâmetros do GA são o tamanho da população, probabilidade de ocorrência de mutação, crossover e inversão. Estes parâmetros devem ser escolhidos de acordo com a natureza do problema e as características do algoritmo implementado.

#### 4.6.2 Estratégia Evolutiva

A Estratégia Evolutiva (ES, do inglês Evolution Strategy) foi desenvolvida por Rechenberg (1973) durante as décadas de 1960 e 1970. Este método foi desenvolvido ao mesmo tempo em que o Algoritmo Genético de Holland, mas com o objetivo restrito apenas a otimização. Entretanto, seus princípios podem ser aplicados a qualquer problema possível de ser atacado pelos Algoritmos Evolutivos.

Uma das principais diferenças entre a ES e o GA inicial é a codificação. Na ES, Rechenberg propôs que os genes fossem representados pelos valores reais das variáveis, ao passo que Holland propôs uma codificação binária. Atualmente, a codificação proposta por Rechenberg passou a ser aplicada no GA, pois para determinados problemas é mais adequada. Por outro lado, problemas que possuem variáveis de natureza numérica diferente, como por exemplo, um conjunto de variáveis reais e outro de variáveis inteiras, não são resolvidos pela codificação que é representada pelo valor real das variáveis.

Outra importante diferença está no processo de seleção. Na ES a seleção acontece após a geração de novos indivíduos, e todos os indivíduos (geradores e gerados) são comparados, sendo escolhidos aqueles que possuem um melhor fitness.

A reprodução de indivíduos é feita apenas por mutação, não havendo, portanto, o crossover ou recombinação. Todos os genes podem ser modificados. Considerando  $x_t$  um vetor resposta da geração  $t$ ,  $x_{t+1}$  pode ser dado pela Equação 4.25:

$$x_{t+1} = x_t + z \quad 4.25$$

Onde  $z$  é um vetor composto de valores aleatórios, os quais são distribuídos normalmente com média zero e desvio padrão  $\sigma$ .

O desvio padrão  $\sigma$  é controlado pela regra do 1/5. Onde após um certo número de gerações, se a razão entre as mutações que obtiveram sucesso (melhoraram o fitness) com o total de mutações for maior do que 1/5 o desvio padrão  $\sigma$  é multiplicado por 1,22. Caso esta relação seja menor do que 1/5, o desvio padrão  $\sigma$  é multiplicado por 0,82 (1/1,22). Sendo  $f$  esta porcentagem de mutações com sucesso, a regra do 1/5 é dada por:

$$\sigma_{t+1} = \begin{cases} 0,82 \cdot \sigma_t \rightarrow \varphi < 1/5 \\ 1,22 \cdot \sigma_t \rightarrow \varphi > 1/5 \\ \sigma_t \rightarrow \varphi = 1/5 \end{cases} \quad 4.26$$

As constantes 0,82 e 1,22 foram usadas por Schwefel (1995) em suas experiências.

Há duas propostas de Estratégia Evolutiva. A estratégia evolutiva para dois indivíduos, onde há apenas um gerador e a comparação é feita entre pai e filho. E a estratégia evolutiva para populações e indivíduo, onde uma população gera outra, e as comparações são feitas entre todos

os indivíduos das duas gerações. Neste caso, o tamanho da população geradora não é, necessariamente, o tamanho da população gerada, e são selecionados indivíduos em número necessário para formar a população geradora.

O critério de parada é baseado no total de gerações e na diminuição do desvio padrão.

O fluxograma da ES para dois indivíduos é representado pela Figura 4.7:

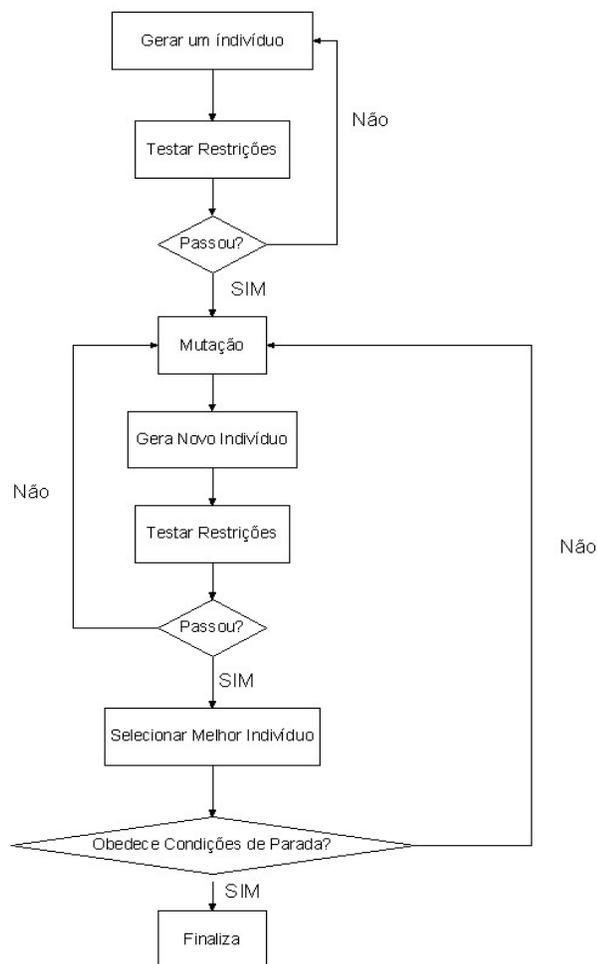


Figura 4.7 - Fluxograma da Estratégia Evolutiva para dois Indivíduos



## Capítulo 5

### Formulação dos Problemas e Implementação Computacional

#### 5.1 Formulação de Problemas de Otimização de Redundância

O uso de redundância de componentes é uma forma de se aumentar a confiabilidade ou disponibilidade de sistemas de engenharia. O uso deste recurso, porém, aumenta o custo de implementação, peso, volume e outras características indesejáveis ao sistema.

Além de componentes redundantes, a melhora nos tempos de manutenção, ou a diminuição dos tempos entre falhas, aumentam a disponibilidade dos sistemas. A razão de dependabilidade  $d$ , representada pela Equação 3.33, mostra a relação entre o MTTF, tempo médio até a falha, e MTTR, tempo médio até o reparo. A diminuição dos tempos de reparo, ou o aumento dos tempos até a falha, incrementam a razão de dependabilidade  $d$ , que por sua vez aumenta a disponibilidade, como mostra a Figura 3.5. Por outro lado, é necessário investimentos em manutenção para diminuir o MTTR, ou em melhoria dos processos, a fim de aumentar o MTTF, acarretando em aumento do custo global do sistema.

Portanto, problemas de otimização foram formulados, a fim de obter valores máximos de confiabilidade ou disponibilidade, ou valores mínimos de custo, para casos onde um limite mínimo de confiabilidade é estabelecido.

Foram formulados sete problemas. Os três primeiros são referentes à Dissertação de Conclusão de Graduação de Lopes (1998), que utilizou o método dos Multiplicadores de Lagrange e considerou uma parametrização da confiabilidade elaborada por Banerjee e Rajamani (1973).

Além do método dos Multiplicadores de Lagrange, foram aplicados a Estratégia Evolutiva e os Algoritmos Genéticos, a fim de que uma comparação entre os métodos de otimização pudesse ser feita.

Os demais problemas envolveram a maximização da disponibilidade, buscando obter não só o número ótimo de redundâncias, mas também a melhora da razão de dependabilidade. Devido à complexidade destes problemas, apenas o Algoritmo Genético foi implementado.

O detalhamento dos problemas de otimização é descrito nos itens seguintes.

### **5.1.1 Problema 1: Minimização do Custo para uma Confiabilidade Mínima**

Este problema tem como objetivo atingir o custo mínimo, para um sistema redundante, com uma confiabilidade mínima desejada.

O sistema é representado pela Figura 3.10. A análise feita no item 3.5.3 sobre sistemas redundantes é pertinente aos problemas de otimização desenvolvidos.

O custo é dado pela soma dos custos dos componentes que fazem parte do sistema. Sendo  $c_i$  o custo de um componente no subsistema  $i$ , o custo do sistema  $C_s$  é expresso por:

$$C_S = \sum_{i=1}^n c_i \cdot y_i \quad 5.1$$

Onde  $y_i$  é o número de componentes no subsistema  $i$ , que é a variável do sistema.

A função de restrição é dada pela confiabilidade do sistema, que é representada pela Equação 3.40. O valor mínimo permitido da confiabilidade do sistema é expresso por  $R_{\min}$ . Desta forma, o problema de otimização pode ser escrito segundo a Equação 5.2.

$$\begin{aligned} \min U &= \sum_{i=1}^n c_i \cdot y_i \\ \text{para :} & \\ \prod_{i=1}^n [1 - (1 - R_i)^{y_i}] &\geq R_{\min} \end{aligned} \quad 5.2$$

### 5.1.2 Problema 2: Maximização da confiabilidade para um Custo Máximo

O aumento da confiabilidade de sistemas pode ser obtido com mais componentes redundantes. Entretanto, isto acarreta em aumento de custo. Propõe-se obter a confiabilidade máxima de um sistema (Equação 3.40), para um custo máximo permitido  $C_{\max}$ , variando o número de componentes redundantes.

$$\max U = \prod_{i=1}^n [1 - (1 - R_i)^{y_i}]$$

para :

5.3

$$\sum_{i=1}^n c_i \cdot y_i \leq C_{\max}$$

### 5.1.3 Problema 3: Maximização da Confiabilidade para Custo e Peso Máximo

Além do incremento no custo, o aumento de componentes redundantes acarreta em maior peso e volume do sistema.

Este problema busca a confiabilidade máxima, tendo como restrição, além do custo máximo, um dado peso máximo permitido do sistema. Sistemas aeronáuticos e aeroespaciais têm o peso como um importante parâmetro de projeto. Portanto, limitar o peso do sistema é uma interessante restrição para um problema de maximização da confiabilidade.

Assim como o custo, o peso é dado pela soma dos pesos dos componentes do sistema. Ele pode ser expresso pela Equação seguinte:

$$W_S = \sum_{i=1}^n w_i \cdot y_i \quad 5.4$$

O problema é dado por:

$$\max U = \prod_{i=1}^n \left[ 1 - (1 - R_i)^{y_i} \right]$$

para :

$$\sum_{i=1}^n c_i \cdot y_i \leq C_{\max} \quad 5.5$$

$$\sum_{i=1}^n w_i \cdot y_i \leq W_{\max}$$

#### 5.1.4 Problema 4: Maximização da Disponibilidade com Redundância e MTTR como variáveis de otimização

Assim como a confiabilidade, a disponibilidade é uma grandeza probabilística. Portanto, todas as considerações feitas no item 3.5.3 sobre confiabilidade também valem para a disponibilidade. Desta forma, a disponibilidade de um sistema redundante  $A_S$  é dada por:

$$A_S = \prod_{i=1}^n \left[ 1 - (1 - A_i)^{y_i} \right] \quad 5.6$$

Considerando que a disponibilidade de um componente pode ser dada pela Razão de Dependabilidade  $d_i$ , como mostra a Equação 3.34, a disponibilidade do sistema é reduzida a:

$$A_S = \prod_{i=1}^n \left[ 1 - \left( \frac{1}{1 + d_i} \right)^{y_i} \right] \quad 5.7$$

Propõem-se maximizar a disponibilidade do sistema, com o aumento do número de componentes redundantes  $y_i$  e da razão de dependabilidade  $d_i$ . Este incremento em  $d_i$  pode ser obtido pela diminuição do MTTR ou pelo aumento do MTTF.

A melhoria na razão de dependabilidade de um componente acarreta em custo, representado por  $cd_i$ , o qual é oriundo de investimentos na manutenção, ou aumento da vida útil do componente. Portanto, o custo global do sistema ganha um novo termo, e é expresso por:

$$C_S = \sum_{i=1}^n (c_i \cdot y_i + cd_i \cdot d_i) \quad 5.8$$

O problema 4 busca otimizar os tempos de reparo simultaneamente com o número de redundâncias. Além de considerar o custo (Equação 5.8) e o peso (Equação 5.4) como restrição, o volume do sistema também é levado em conta. O volume é representado pela Equação 5.9.

$$V_S = \sum_{i=1}^n v_i \cdot y_i \quad 5.9$$

Portanto, o problema de otimização, onde as variáveis são  $y_i$  e  $MTTR_i$  é dado por:

$$\max U = \prod_{i=1}^n \left[ 1 - \left( \frac{1}{1 + d_i} \right)^{y_i} \right]$$

para :

$$d_i = \frac{MTTF_i}{MTTR_i}$$

$$\sum_{i=1}^n (c_i \cdot y_i + cd_i \cdot d_i) \leq C_{\max} \quad 5.10$$

$$\sum_{i=1}^n w_i \cdot y_i \leq W_{\max}$$

$$\sum_{i=1}^n v_i \cdot y_i \leq V_{\max}$$

### 5.1.5 Problema 5: Maximização da Disponibilidade com Redundância e MTTF como variáveis de otimização

Assim como no item anterior, busca-se maximizar a disponibilidade do sistema com o aumento do número de redundância e da razão de dependabilidade dos componentes. A diferença está na maneira de se obter a melhoria na razão de dependabilidade. A variável a ser considerada neste caso será o MTTF.

A formulação do problema é idêntica ao caso anterior (Equação 5.10). Entretanto, as variáveis do problema passam a ser o número de componentes  $y_i$  e o tempo médio até a falha  $MTTF_i$ .

### 5.1.6 Problema 6: Maximização da Disponibilidade para Equipes de Manutenção

Os problemas desenvolvidos nos itens 5.1.4 e 5.1.5 consideraram um aumento na razão de dependabilidade de uma forma genérica, não considerando como esta melhoria poderia ser obtida.

A utilização de equipes de manutenção pode diminuir o MTTR, influenciando diretamente na razão de dependabilidade. Considerando  $MTTR_{li}$  como sendo o MTTR da atuação de uma equipe de manutenção em um componente do subsistema  $i$ , a razão de dependabilidade deste subsistema é dada por:

$$d_i = \frac{MTTF_i}{feq \cdot MTTR_{li}} = \frac{1}{feq} \cdot d_{li} \quad 5.11$$

Onde  $d_{li}$  é a razão de dependabilidade resultante da atuação de uma equipe no subsistema e  $feq$  é um fator de influência do número de equipes de manutenção que atuam simultaneamente no subsistema.

A influência do número de equipes de manutenção no subsistema pode ser compreendida no esquema representado pela Figura 5.1.

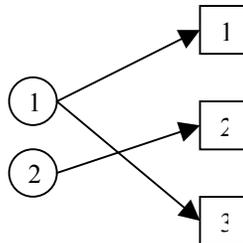


Figura 5.1 - Atuação de Equipes de manutenção

Cada componente é reparado por apenas uma equipe de manutenção. O número de equipes de manutenção atuando em um subsistema não é superior ao número de componentes. A Figura 5.1 representa um subsistema com três componentes e duas equipes de manutenção atuantes. Desta forma, se os três componentes falharem, as duas equipes atuarão simultaneamente nos dois primeiros componentes, e a primeira a finalizar o trabalho, irá reparar o terceiro componente. Então, o fator de influência das equipes de manutenção será de 2, pois o MTTR do subsistema é igual a duas vezes o MTTR da atuação de uma equipe de manutenção em um componente.

De uma forma genérica, o fator de influência  $feq$  é dado por:

$$feq = \begin{cases} \frac{y_i}{eq_i} & \text{se } \frac{y_i}{eq_i} \in Z \\ \text{int}\left(\frac{y_i}{eq_i}\right) + 1 & \text{se } \frac{y_i}{eq_i} \notin Z \end{cases} \quad 5.12$$

Onde  $eq_i$  é o número de equipes de manutenção atuando no subsistema  $i$ .

Pode-se considerar um custo de manutenção  $CM$ , onde o custo das equipes de manutenção  $ceq_i$  e o custo de manutenção de cada componente  $cm_i$  são considerados:

$$CM = \sum_{i=1}^n eq_i \cdot ceq_i + \sum_{i=1}^n q_i \cdot y_i \cdot cm_i \quad 5.13$$

Onde  $q_i$  é a probabilidade de falha, que para uma distribuição exponencial (MTTF e MTTR constantes) é dado pela Equação 3.8.

A Equação a seguir, representa a problema de otimização para este caso:

$$\max U = \prod_{i=1}^n \left[ 1 - \left( \frac{1}{1+d_i} \right)^{y_i} \right]$$

para :

$$d_i = \frac{MTTF_i}{feq \cdot MTTR_{1i}} = \frac{1}{feq} \cdot d_{1i}$$

$$\sum_{i=1}^n c_i \cdot y_i \leq C_{\max}$$

5.14

$$\sum_{i=1}^n w_i \cdot y_i \leq W_{\max}$$

$$\sum_{i=1}^n v_i \cdot y_i \leq V_{\max}$$

$$\sum_{i=1}^n eq_i \cdot c_{eqi} + \sum_{i=1}^n q_i \cdot y_i \cdot c_{mi} \leq CM_{\max}$$

### 5.1.7 Problema 7: Maximização da Disponibilidade para recursos de Manutenção

Nem todos os sistemas de manutenção podem ser representados como no item anterior. Portanto, uma formulação mais genérica é necessária, a fim de abranger uma gama maior de sistemas reais.

Considerando *rec*, sendo uma variável percentual para representar os recursos de manutenção disponíveis para aplicação, pode se desenvolver um problema de otimização mais genérico que o caso anterior.

A variável impacto *I* é definida como o impacto na razão de dependabilidade, se aplicado 100% dos recursos de manutenção. Então, o impacto em um subsistema *i* é dado pelo produto da variável *rec<sub>i</sub>* pelo impacto *I*. Desta forma, a razão de dependabilidade de um subsistema *d<sub>i</sub>* é dada por:

$$d_i = [(I - 1) \cdot rec_i + 1] \cdot d_{1i} \quad 5.15$$

Onde  $d_{1i}$  é a razão de dependabilidade sem aplicação de recursos de manutenção. Caso o recurso seja 0%, não haverá nem um impacto na razão de dependabilidade. Se o recurso for 100% o impacto na razão de dependabilidade será  $I$ .

O custo de manutenção é expresso por:

$$CM = \sum_{i=1}^n rec_i \cdot c_{rec_i} + \sum_{i=1}^n q_i \cdot y_i \cdot cm_i \quad 5.16$$

Então, o problema de manutenção pode ser representado pela Equação 5.17 a seguir:

$$\begin{aligned} \max U &= \prod_{i=1}^n \left[ 1 - \left( \frac{1}{1 + d_i} \right)^{y_i} \right] \\ \text{para :} \\ d_i &= [(I - 1) \cdot rec_i + 1] \cdot d \\ \sum_{i=1}^n c_i \cdot y_i &\leq C_{\max} \\ \sum_{i=1}^n w_i \cdot y_i &\leq W_{\max} \\ \sum_{i=1}^n v_i \cdot y_i &\leq V_{\max} \\ \sum_{i=1}^n rec_i \cdot c_{rec_i} + \sum_{i=1}^n q_i \cdot y_i \cdot cm_i &\leq CM_{\max} \end{aligned} \quad 5.17$$

## 5.2 Implementação dos Multiplicadores de Lagrange

Uma das dificuldades de se trabalhar com Multiplicadores de Lagrange é a necessidade do uso de cálculo diferencial. A fim de simplificar a Equação 3.40, Banerjee e Rajamani (1973) desenvolveram uma parametrização da confiabilidade, que possui derivadas parciais da Equação da confiabilidade do sistema, mais simples de serem obtidas. Esta parametrização, representada por  $\phi$ , é dada por:

$$\phi = \frac{1-R}{R} \quad 5.18$$

Portanto, a confiabilidade é dada por:

$$R = \frac{1}{1+\phi} \quad 5.19$$

Considerando um sistema em série, representado pela Figura 3.7, substituindo a Equação 5.19 na Equação 3.37 e invertendo a expressão obtida, tem-se a expressão do parâmetro  $\phi$  para a confiabilidade de um sistema em série:

$$(1+\phi_s) = \prod_{k=1}^n (1+\phi_k) \quad 5.20$$

Considerando um sistema com dois elementos,  $n = 2$ , o parâmetro do sistema será,

$$\phi_s = \phi_1 + \phi_2 + \phi_1\phi_2 \quad 5.21$$

Assumindo  $\phi \ll 1$ , os termos de ordens mais elevadas na Equação 5.21 podem ser desprezados. Então o parâmetro correspondente ao sistema em série  $\phi_s$  pode ser aproximado por:

$$\phi_s = \sum_{k=1}^n \phi_k \quad 5.22$$

Em sistemas paralelos, Figura 3.8, obtém-se a expressão para o parâmetro da confiabilidade substituindo a Equação 5.19 na Equação 3.38:

$$\frac{\phi_p}{1 + \phi_p} = \prod_{k=1}^n \frac{\phi_k}{1 + \phi_k} \quad 5.23$$

Para valores pequenos de  $\phi$ , o termo  $(1 + \phi)$  pode ser aproximado para a unidade. Então, o parâmetro  $\phi_p$  para um sistema paralelo torna-se:

$$\phi_p = \prod_{k=1}^n \phi_k \quad 5.24$$

Considerou-se no item 3.5.3 que os componentes de um estágio (componentes redundantes) possuem a mesma confiabilidade. Neste caso, o parâmetro da confiabilidade no estágio  $i$  é dado por:

$$\phi_p = \phi_i^{y_i} \quad 5.25$$

Sendo o sistema redundante composto por  $n$  estágios em série o parâmetro da confiabilidade neste sistema é representado por:

$$\phi_{sp} = \sum_{i=1}^n \phi_i^{y_i} \quad 5.26$$

Sistemas complexos em confiabilidade podem ser determinados usando o método paramétrico com essas equações mais simples. Note que o método paramétrico assume que  $\phi < 1$ . Entretanto, quando  $R$  é menor do que 0.5,  $\phi > 1$ . Então, a faixa de  $\phi$  é um importante critério para considerar o erro no cálculo da confiabilidade quando o método paramétrico é utilizado. Portanto, o método paramétrico pode ser utilizado para determinar a confiabilidade de um sistema com componentes de alta confiabilidade, com um menor erro.

A expressão da Equação 5.26 possui derivadas mais simples de calcular do que a expressão da Equação 3.40, o que possibilita a implementação do método dos Multiplicadores de Lagrange. Por outro lado, pode ser aplicado apenas em sistemas com elevada confiabilidade.

### 5.2.1 Problema 1

Considerando o caso onde um sistema específico deve alcançar uma certa confiabilidade mantendo o menor preço possível. Nesta situação, a função objetivo a ser minimizada é:

$$U = \sum C_i * y_i \quad 5.27$$

A inequação de restrição que deve ser satisfeita é:

$$\phi \leq \sum \phi_i^{y_i} \quad 5.28$$

Onde  $\phi$  é dado pela Equação 5.18.

Desta forma, a função de restrição e a expressão de Lagrange  $LE$  são respectivamente:

$$\psi = \sum \phi_i^{y_i} - \phi \quad 5.29$$

$$LE = U + \nu * \psi \quad 5.30$$

Onde  $\nu$  é o multiplicador de Lagrange.

Portanto a expressão completa se torna:

$$LE = \sum C_i * y_i + \nu * \left( \sum \phi_i^{y_i} - \phi \right) \quad 5.31$$

Para otimizar o número de componentes  $y_i$  no estágio  $i$  a diferenciação da expressão de Lagrange com respeito a  $y_i$  resulta em:

$$\frac{\partial LE}{\partial y_i} = C_i + \nu * \ln \phi_i * \phi_i^{y_i} = 0 \quad 5.32$$

Para  $i = 1$ , a Equação 5.32 torna-se:

$$\nu * \ln \phi_1 \phi_1^{y_1} = -C_1 \quad 5.33$$

Eliminando  $\nu$  das Equações 5.32 e 5.33, resulta na seguinte equação,

$$\phi_i^{y_i} = K_i * \phi_1^{y_1} \quad 5.34$$

Onde:

$$K_i = \frac{\ln \phi_1 * C_i}{\ln \phi_i * C_1} \quad 5.35$$

Das equações 5.33 e 5.34,  $v$  pode ser calculado como:

$$v = -\frac{C_1}{S * \ln \phi_1} \quad 5.36$$

Onde:

$$S = \frac{\phi_i^{y_i}}{K} \quad 5.37$$

para,

$$K = \sum K_i$$

Então, o número total de componentes no estágio  $i$  é dado por:

$$y_i = \frac{\ln(K_i * S)}{\ln \phi_i} \quad 5.38$$

O número de componentes  $y_i$  deve ser aproximado para um número inteiro, a fim de alcançar o ponto ótimo.

Assumindo  $\bar{n}_1 = (n_{11}, n_{12}, \dots, n_{1N})$  um vetor resposta do problema e  $\bar{n}_2 = (n_{21}, n_{22}, \dots, n_{2N})$  outro conjunto de resposta, onde a função objetivo do sistema é melhor, pode-se afirmar que  $\bar{n}_1$  é uma seqüência não-dominada para o problema 1, se  $C(\bar{n}_2) < C(\bar{n}_1)$  implicar em  $R_i(n_{2i}) < R_i(n_{1i})$  para qualquer  $i$ , onde  $R_i$  é a confiabilidade no subsistema  $i$ . Uma seqüência não-dominada garante que as condições restritivas do problema sejam garantidas.

Considerando que, para qualquer  $i$ ,  $R_i(n_{2i}) > R_i(n_{1i})$ , o vetor resposta  $\bar{n}_1$  não será uma seqüência não dominada, pois existirá um outro conjunto resposta que gere um custo menor para uma confiabilidade maior.

A função objetivo atingirá seu ponto mínimo, dentro de limites restritivos, sendo uma função convexa, satisfazendo, desta forma, as condições de Kuhn-Tucker. A convexidade para a expressão da Equação 5.26 foi desenvolvida por Barlow e Proschan (1965).

Então, Banerjee e Rajamani (1973) determinaram a inequação da expressão 5.39, baseando-se na convexidade de  $\phi$ , a qual fornece seqüências não-dominadas até que os limites de restrições sejam atingidos e, conseqüentemente, a condição ótima do problema:

$$v * \frac{\phi_i^{y_i}}{(1 + \phi_i)^{(1+y_i)}} < c_i \quad 5.39$$

O algoritmo pode ser visto no fluxograma da Figura 5.2:

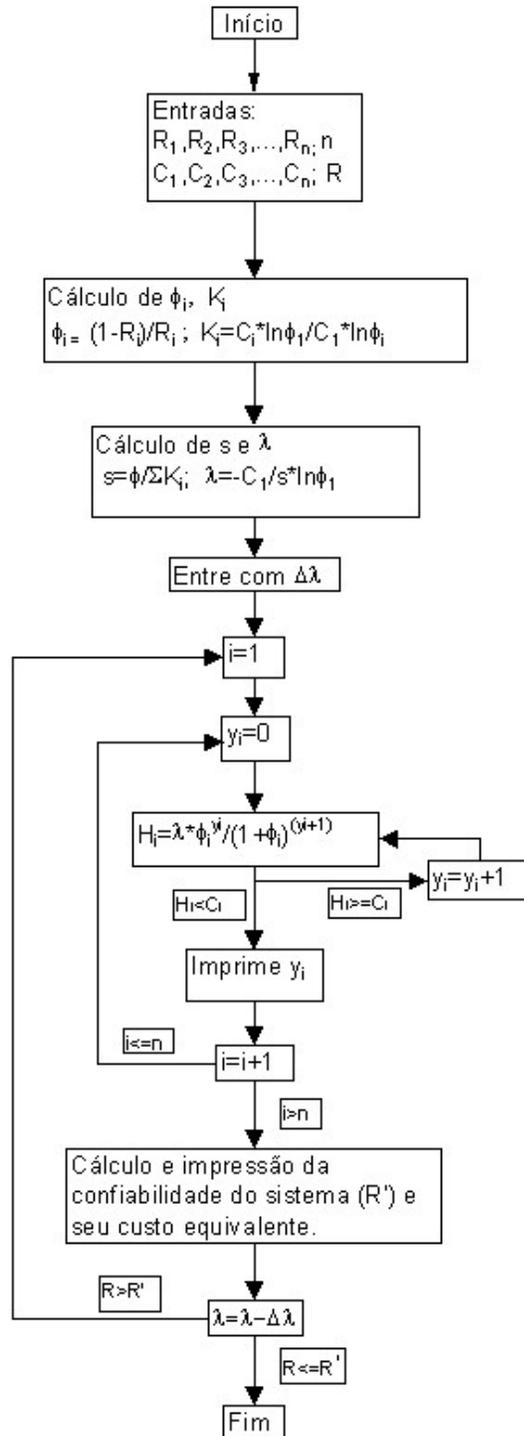


Figura 5.2 - Fluxograma do Multiplicadores de Lagrange para o Problema 1

### 5.2.2 Problema 2

Considere um sistema onde o número de estágios e a confiabilidade dos componentes são dados. É desejável que o custo total do sistema não exceda um dado custo limite  $C$ . Portanto, a função de restrição pode ser escrita como,

$$\psi = \sum C_i * y_i - C = 0 \quad 5.40$$

Como a confiabilidade está sujeita a otimização (minimizando  $\phi_i$ ), a função objetivo pode ser definida como:

$$U = \sum \phi_i^{y_i} \quad 5.41$$

Então, a expressão de Lagrange para a otimização da confiabilidade é:

$$LE = U + \nu * \psi \quad 5.42$$

Portanto a expressão completa se torna:

$$LE = \sum \phi_i^{y_i} + \nu * \left( \sum C_i * y_i - C \right) \quad 5.43$$

Para otimizar o número de componentes  $y_i$  no estágio  $i$ , a diferenciação da equação de Lagrange em relação a  $y_i$  resulta em:

$$\frac{\partial LE}{\partial y_i} = \ln \phi_i * \phi_i^{y_i} + \nu * C_i = 0 \quad 5.44$$

De acordo com a derivação de  $y_i$ , somente um estágio é considerado por vez. A equação acima pode ser reescrita da seguinte forma:

$$y_i = a_i * \ln \nu + b_i \quad 5.45$$

Onde:

$$\begin{aligned} a_i &= \frac{1}{\ln \phi_i} \\ b_i &= \frac{\ln K_i}{\ln \phi_i} \\ K_i &= -\frac{C_i}{\ln \phi_i} \end{aligned} \quad 5.46$$

Considerando as Equações 5.40 e 5.45, a expressão para o multiplicador de Lagrange  $\nu$  pode ser obtida da seguinte forma:

$$\nu = e^s \quad 5.47$$

Onde:

$$S = \frac{C - \sum C_i * b_i}{\sum C_i * a_i} \quad 5.48$$

Uma vez obtido  $v$ ,  $y_i$  pode ser obtida utilizando a Equação 5.45;  $y_i$ , obtida como uma variável contínua, é aproximada ao inteiro mais próximo para alcançar um projeto próximo ao ideal. Assim como no primeiro caso, determinou-se uma expressão condicional para a obtenção do ponto ótimo:

$$\frac{\phi_i^{y_i}}{(1 + \phi_i)^{(1+y_i)}} < v * C \quad 5.49$$

Esta inequação resulta diretamente da convexidade de  $\phi$ . Calculando  $v$  na Equação 5.47, o menor número  $n_i$  que irá satisfazer a inequação acima pode ser obtido. Alterando o valor de  $v$  em decrementos, diferentes seqüências que correspondem a cada  $v$  podem ser obtidas. Escolhendo apropriados decrementos de  $v$ , o número ótimo de componentes em cada estágio, que satisfaz a restrição de custo do sistema pode ser obtido usando o procedimento iterativo mostrado no fluxograma da Figura 5.3. Como pode ser visto no fluxograma, o valor de  $v$  muda com o custo e a confiabilidade de cada elemento. O decremento de  $v$  pode ser escolhido para diferentes dados de entrada. É mais seguro selecionar  $\Delta v$  pequeno para não perder a correta solução ótima. O único inconveniente é o tempo computacional.

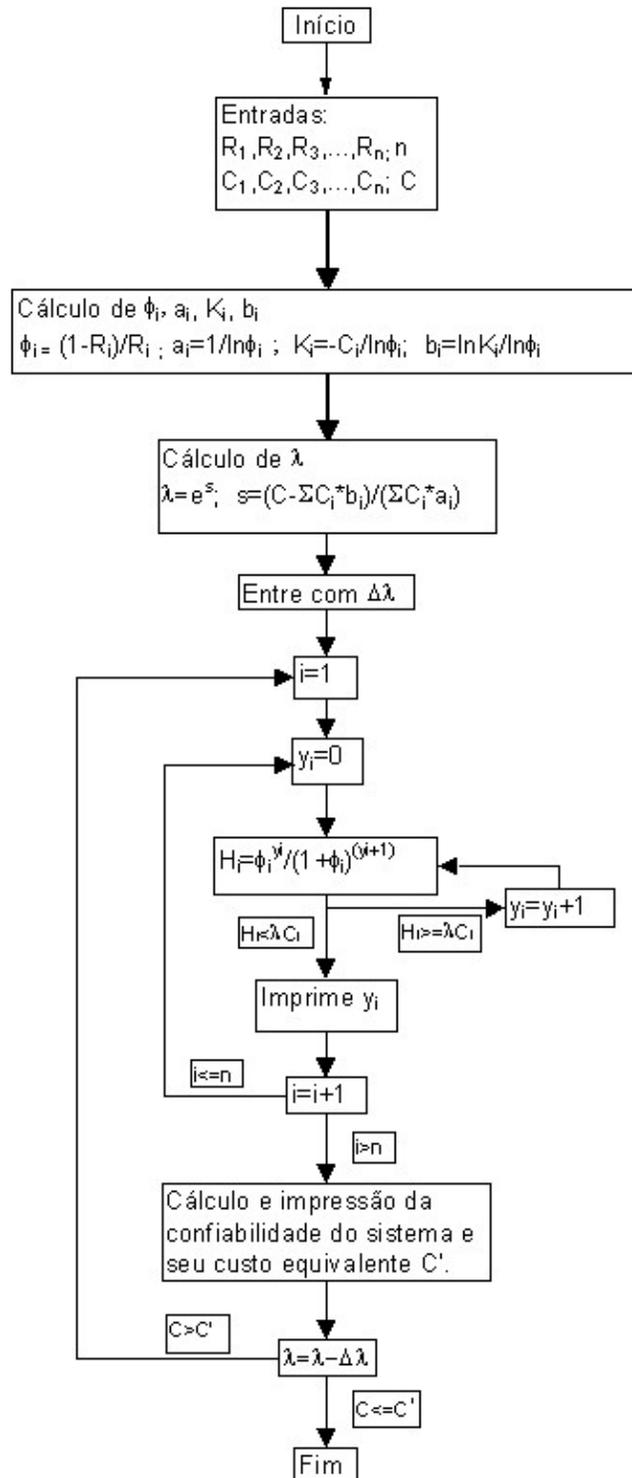


Figura 5.3 - Fluxograma do Multiplicadores de Lagrange para o Problema 2

### 5.2.3 Problema 3

Nos casos anteriores, o sistema está sujeito a apenas uma condição de restrição. Nesta seção, o trabalho é estendido para considerar múltiplas restrições. Considere o problema de maximização da confiabilidade com ambas as restrições lineares de custo e peso. A função objetivo que deve ser minimizada é:

$$U = \sum \phi_i^{y_i} \quad 5.50$$

O peso e o custo do sistema são ambos funções lineares do número de elementos e são utilizados como funções de restrição. O peso e o custo total do sistema não deve exceder um dado peso limite  $W$  e custo limite  $C$ :

$$\begin{aligned} C &\geq \sum C_i * y_i \\ W &\geq \sum W_i * y_i \end{aligned} \quad 5.51$$

As funções de restrição podem ser escritas como:

$$\begin{aligned} \psi_1 &= \sum C_i * y_i - C = 0 \\ \psi_2 &= \sum W_i * y_i - W = 0 \end{aligned} \quad 5.52$$

Então, a expressão de Lagrange pode ser expressa por:

$$LE = \sum \phi_i^{y_i} + \nu_1 * \left( \sum C_i * y_i - C \right) + \nu_2 * \left( \sum W_i * y_i - W \right) \quad 5.53$$

Diferenciando em relação a  $y_i$ , obtém-se:

$$\frac{\partial LE}{\partial y_i} = \ln(\phi_i * \phi_i^{y_i}) + \nu_1 * C_i + \nu_2 * W_i = 0 \quad 5.54$$

Rearranjando a Equação 5.54,

$$y_i = \frac{1}{\ln \phi_i} * [\ln(a_i * \nu_1 + b_i * \nu_2)] \quad 5.55$$

Onde:

$$a_i = -\frac{C_i}{\ln \phi_i} \quad 5.56$$

$$b_i = -\frac{W_i}{\ln \phi_i}$$

Substituindo a Equação 5.55 em 5.52:

$$C = -\sum a_i * [\ln(a_i * \nu_1 + b_i * \nu_2)] \quad 5.57$$

$$W = -\sum b_i * [\ln(a_i * \nu_1 + b_i * \nu_2)]$$

Resolvendo simultaneamente o sistema de equações não-lineares, representado pela expressão 5.57, encontra-se  $\nu_1$  e  $\nu_2$ . As equações não-lineares simultâneas podem ser resolvidas utilizando o método de Newton - Raphson.

Uma vez calculado  $v_1$  e  $v_2$ ,  $y_i$  pode ser encontrado pela Equação 5.55. Valores aproximados de  $v_1$  e  $v_2$  serão suficientes, pois os valores finais de  $y_i$  serão aproximados para inteiros. Como nos casos anteriores, desenvolveu-se uma expressão que condiciona o valor ótimo, gerando uma seqüência não-dominada. Esta expressão é representada pela Equação 5.58:

$$\frac{\phi_i^{y_i}}{(1 + \phi_i)^{(1+y_i)}} < v * (g_1 * C_i + g_2 * W_i) \quad 5.58$$

Onde:

$$v = v_1 + v_2$$

$$g_1 = \frac{v_1}{v_1 + v_2} \quad 5.59$$

$$g_2 = \frac{v_2}{v_1 + v_2}$$

O algoritmo para este problema pode ser visualizado no fluxograma da Figura 5.4.

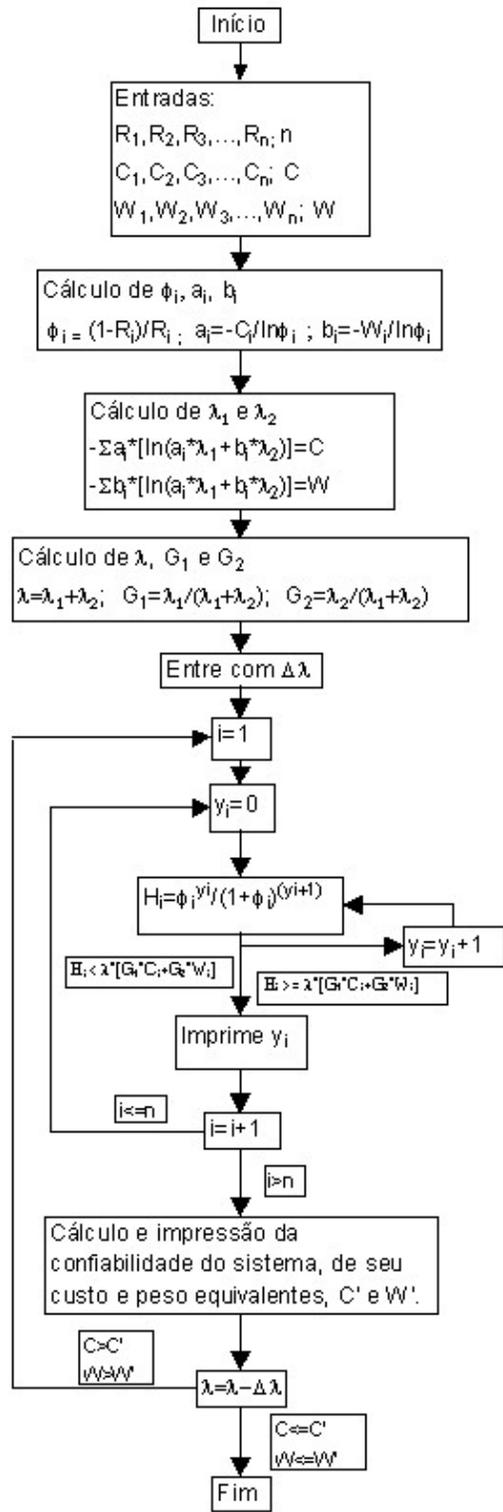


Figura 5.4 - Fluxograma do Multiplicadores de Lagrange para o Problema 3

### 5.3 Estratégia Evolutiva

A Estratégia Evolutiva para dois Indivíduos foi um dos métodos a serem aplicados na resolução dos problemas um, dois e três. A facilidade de aplicação é uma das principais vantagens deste método. A seção 4.6.2 introduz os aspectos teóricos desta metodologia de otimização.

Não há codificação binária neste método. Desta forma, o cromossomo do indivíduo é o vetor resposta do problema.

Inicialmente, um indivíduo é formado aleatoriamente, utilizando os processos randômicos da linguagem de programação. Cada gene do indivíduo é sorteado. Considerando que o sistema a ser otimizado tenha cinco estágios, cinco números inteiros serão sorteados, os quais representam o número de componentes em cada estágios. Não foram considerados números inteiros maiores do que dez, que é um valor significativo de componentes redundantes, por questões de ordem prática em sistemas mecânicos. A Figura 5.5 representa este passo de otimização:

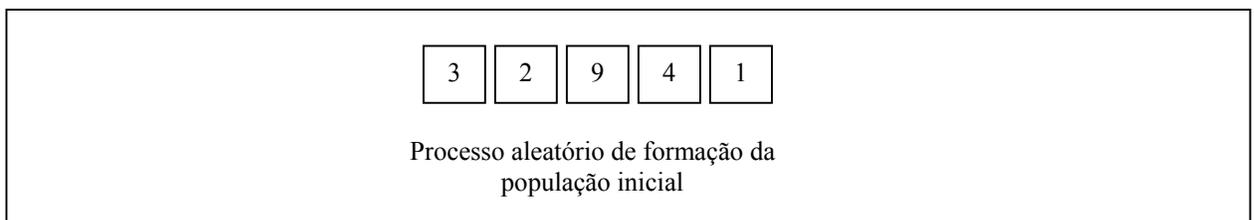


Figura 5.5 – Formação do indivíduo inicial na Estratégia Evolutiva

O indivíduo formado deve obedecer às condições de restrição. Caso isto não ocorra, o processo se repete até que um indivíduo esteja dentro da região factível.

Em seguida um novo indivíduo é formado por mutação. A mutação, neste caso, ocorre em todos os genes, como mostra a Figura 5.6:

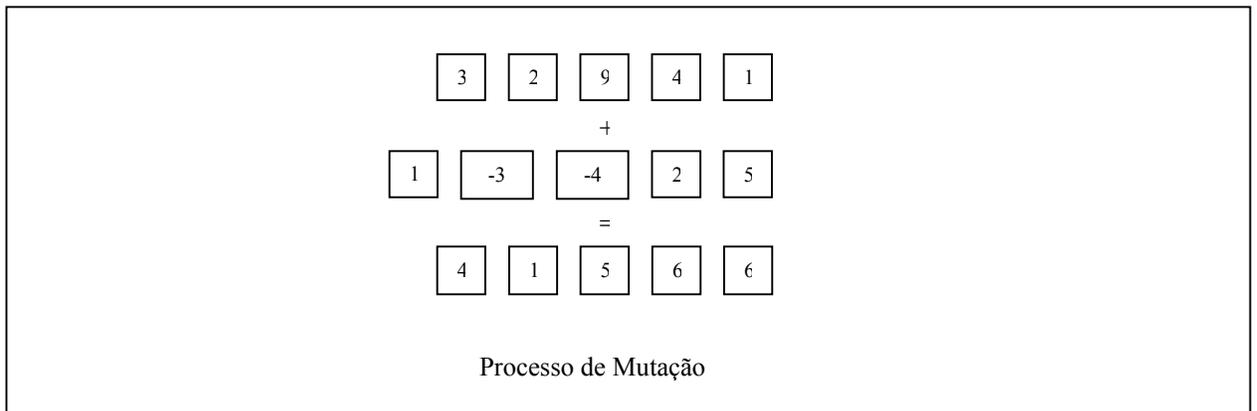


Figura 5.6 – Processo de Mutação na Estratégia Evolutiva

Não há número de componentes negativo em um sistema de engenharia. Portanto, se a mutação formar um gene com valor negativo, é assumido o módulo deste indivíduo. No caso de um gene assumir valor nulo, o cálculo do *fitness* será negativo ou as condições de restrição quebradas, então este indivíduo não será selecionado para a próxima geração.

A seleção é feita pela comparação entre os *fitness*, ou função objetivo, dos indivíduos. O indivíduo mais apto irá sofrer as mutações para a próxima geração.

Como o número de componentes é limitado até dez componentes, não é seguida a regra do 1/5, como apresentado anteriormente, pois a redução do desvio padrão atingiria valores menores do que um rapidamente.

O critério de parada é um parâmetro muito importante no algoritmo implementado. Três condições de paradas distintas foram desenvolvidas, pois problemas diferentes convergem de forma diferente, exigindo diferentes formas de paradas.

1. O vetor resposta deve ser selecionado um determinado número de vezes, que é um parâmetro determinado pelo usuário.

2. A diferença percentual entre um vetor resposta e o vetor anterior deve ser menor do que um erro pré-determinado. Isto só ocorre quando decorrido 75% do número máximo de iterações, ou total de gerações.
3. O número máximo de iterações (ou gerações) é atingido.

O algoritmo desenvolvido é representado pelo fluxograma da Figura 4.7.

A diferença entre os três problemas se dá no cálculo da função objetivo e das funções de restrição. O algoritmo base é o mesmo para os três casos.

## **5.4 Algoritmo Genético**

O Algoritmo Genético foi implementado para solucionar todos os problemas levantados. Os demais métodos não podem ser aplicados aos problemas quatro a sete, pois estes problemas possuem dois tipos de variáveis, número de componentes redundantes e outra variável que influencia a razão de dependabilidade dos componentes: MTTR, MTTF, equipes de manutenção e recursos de manutenção respectivamente para os problemas quatro a sete.

O item 4.6.1 expõe os pontos básicos do Algoritmo Genético. O fluxograma que representa o algoritmo é representado na Figura 4.4. É necessário, entretanto, determinar o tipo de seleção, codificação, mutação, *crossover* e inversão que irá ser implementado. Esta escolha depende diretamente da natureza das características do problema de otimização.

### **5.4.1 Maximização da Confiabilidade e Minimização do Custo**

Um mesmo Algoritmo Genético foi aplicado aos problemas de um a três, diferenciando-se a forma como a função objetivo (*fitness*) é calculada.

Adotou-se que os genes seriam compostos pelo número de componentes redundantes, ou seja, o fenótipo dos indivíduos é igual ao genótipo. Isto foi assumido, porque esta forma de codificação se mostrou mais eficiente, nestes problemas, do que uma codificação binária.

A formação da população inicial é similar ao método utilizado para a Estratégia Evolutiva (Figura 5.5), diferenciando-se apenas no número de indivíduos que é formado, o qual neste caso é igual ao tamanho da população.

Como no caso anterior, apenas indivíduos que obedeçam as condições de restrição são aceitos. Esta atitude aumenta o tempo de convergência, mas aumenta a probabilidade de que o ponto ótimo seja atingido, pois apenas indivíduos aptos são formados.

O processo de seleção associa o princípio de seleção por rank e o princípio do elitismo. Adotou-se que os 50% melhores indivíduos (melhor *fitness*) da população são selecionados para a reprodução e composição da geração seguinte. Isto faz com que a variabilidade de indivíduos seja muito pequena após algumas gerações, sendo necessário um alto índice de mutações.

Os métodos de seleção probabilísticos foram testados, mas não apresentaram resultados melhores do que o método adotado.

A reprodução é realizada pelo mecanismo de *crossover* e mutação. A Figura 4.5 representa estes mecanismos. No caso da mutação, foi utilizado o mesmo algoritmo da Estratégia Evolutiva (item 5.3), diferenciando apenas no fato de que a mutação é realizada em apenas um gene.

### 5.4.2 Maximização da Disponibilidade

Os problemas quatro a sete possuem dois tipos de variáveis, sendo que apenas no problema seis as duas são números inteiros. Nos demais problemas, os números são inteiros e reais. Desta forma, é necessário que a codificação do Algoritmo Genético seja binária, a fim de que os dados referentes a um tipo de variável possam interagir com os dados do outro tipo de variável.

A população inicial é formada aleatoriamente, onde os genes dos indivíduos (valor 0 ou 1) são sorteados. Os indivíduos são codificados segundo a Equação 5.60:

$$x = x_{\min} + \sum_{l=0}^{k-1} b_l \cdot 2^l \cdot \frac{x_{\max} - x_{\min}}{2^k - 1} \quad 5.60$$

Onde:  $x_{\min}$  e  $x_{\max}$  são respectivamente os limites inferior e superior do domínio da variável decimal  $x$  e  $b_l$  é uma cadeia binária de tamanho  $k$  (ou  $k$  bits).

Foi adotado  $k = 8$  para os problemas. Desta forma, se um sistema de engenharia possuir 5 estágios, com duas variáveis cada (número de redundâncias e razão de dependabilidade), a cadeia terá um total de 80 bits (o produto de 8 bits por variável, 5 estágios e 2 variáveis).

Após a codificação, o *fitness* de cada indivíduo é calculado de acordo com a função objetivo e as funções de restrição correspondentes (Equações 5.10, 5.14 e 5.17). Caso alguma condição de restrição seja quebrada, o *fitness* é considerado igual a zero. Na primeira geração, não é aceito nenhum indivíduo com *fitness* igual a zero, a fim de garantir apenas que indivíduos aptos sejam selecionados.

Os métodos de seleção e mutação são similares ao algoritmo implementado para os problemas de um a três. Entretanto, a mutação é simplificada, pois é necessário que se troque o valor zero por um, ou vice-versa.

O método de inversão é adotado como exemplificado na Figura 4.6.

Como a cadeia de genes é mais longa (produto do número de variáveis pelo número de bits adotado), foi assumido neste algoritmo que o crossover pode ter dois pontos de interseção aleatórios, como exemplificado na Figura 5.7:

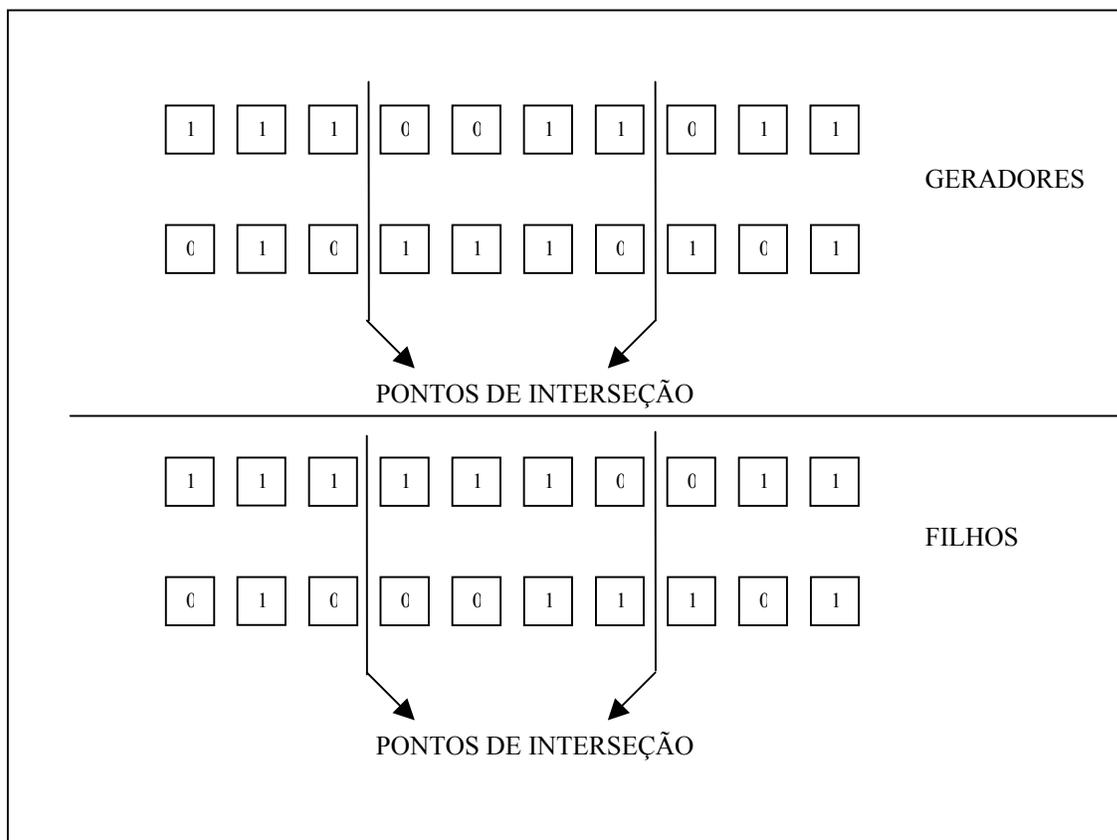


Figura 5.7 – Crossover com dois pontos de interseção

## 5.5 Desenvolvimento do Programa

O programa desenvolvido em Visual Basic 5.0 é dividido em três seções distintas. Otimização da confiabilidade, a qual resolve os problemas um a três, otimização da

disponibilidade, que representa os problemas quatro a sete e análise gráfica. Esta última seção tem como objetivo analisar a convergência da função objetivo e o desenvolvimento das populações no Algoritmo Genético.

Os diferentes tamanhos dos problemas de otimização exigem parâmetros de programas diferentes. Então, uma seção de mudança de parâmetros foi desenvolvida, como pode ser visto na Figura 5.8:

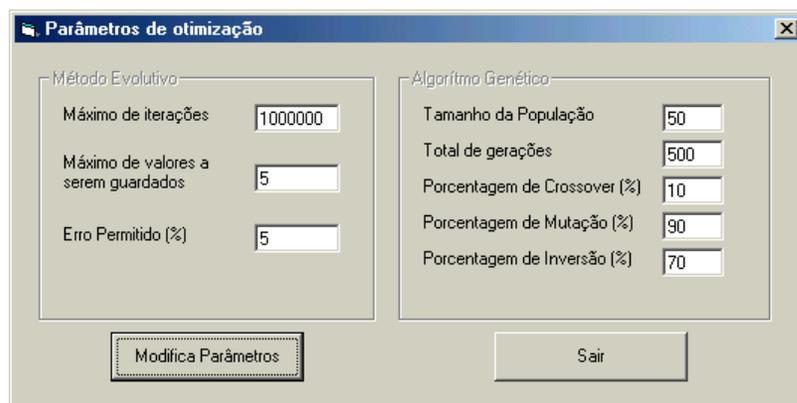


Figura 5.8 – Tela de mudança de parâmetros

Os parâmetros considerados são:

Estratégia Evolutiva:

- Máximo de iterações;
- Valores a serem guardados, ou número de repetições da função objetivo;
- Erro permitido.

Algoritmo Genético:

- Tamanho da População;
- Total de Gerações
- Probabilidade (ou porcentagem) de crossover;
- Probabilidade (ou porcentagem) de mutação;
- Probabilidade (ou porcentagem) de inversão;

Os parâmetros referentes a Estratégia Evolutiva são pertinentes a condições de paradas.

O método dos multiplicadores de Lagrange possui parâmetros característicos de sua operação que são entradas na execução do método.

### **5.5.1 Otimização da Confiabilidade**

Esta seção do programa otimiza os problemas de um a três. Apenas uma tela foi desenvolvida (Figura 5.9), pois as entradas para os problemas são as mesmas.

As entradas são:

- Problema a ser resolvido;
- Número de estágios ou subsistemas;
- Custo máximo permitido

- Peso máximo permitido
- Confiabilidade mínima permitida;
- Confiabilidade de cada componente;
- Custo de cada componente;
- Peso de cada componente.

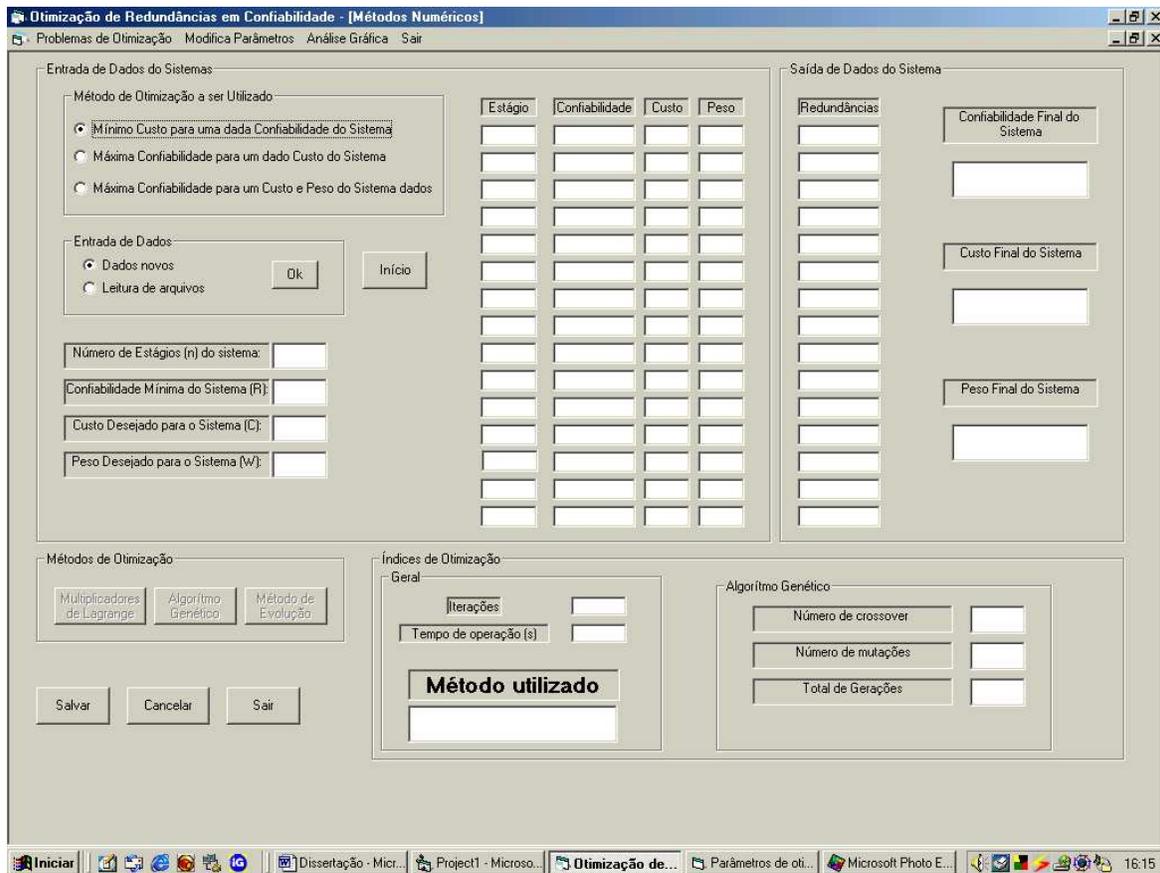


Figura 5.9 – Tela de otimização da Confiabilidade

As saídas do programa são:

- Número de redundâncias em cada estágio;
- Confiabilidade final do sistema;
- Peso final do sistema;
- Custo final do sistema;
- Tempo de processamento;
- Número de iterações (Estratégia Evolutiva);
- Número de crossover, mutação e geração (Algoritmo Genético);
- Método de otimização utilizado.

### **5.5.2 Otimização da Disponibilidade**

Como os problemas de otimização da disponibilidade possuem diferentes entradas e saídas, foi desenvolvida uma tela para cada problema.

As entradas comuns aos problemas são:

- Número de subsistemas;
- Número máximo de redundância;

- Peso máximo do sistema;
- Volume máximo do sistema;
- Custo, peso e volume de cada componente.

As saídas comuns são:

- Tempo de processamento;
- Número de redundâncias;
- Disponibilidade, custo, peso e volume final do sistema.

O problema quatro tem como objetivo otimizar o número de redundâncias e o MTTR. Desta forma, as entradas específicas deste problema são o MTTR máximo e mínimo e o MTTF de cada componente. Enquanto que as saídas específicas são o MTTR de cada componente e a taxa de reparo correspondente.

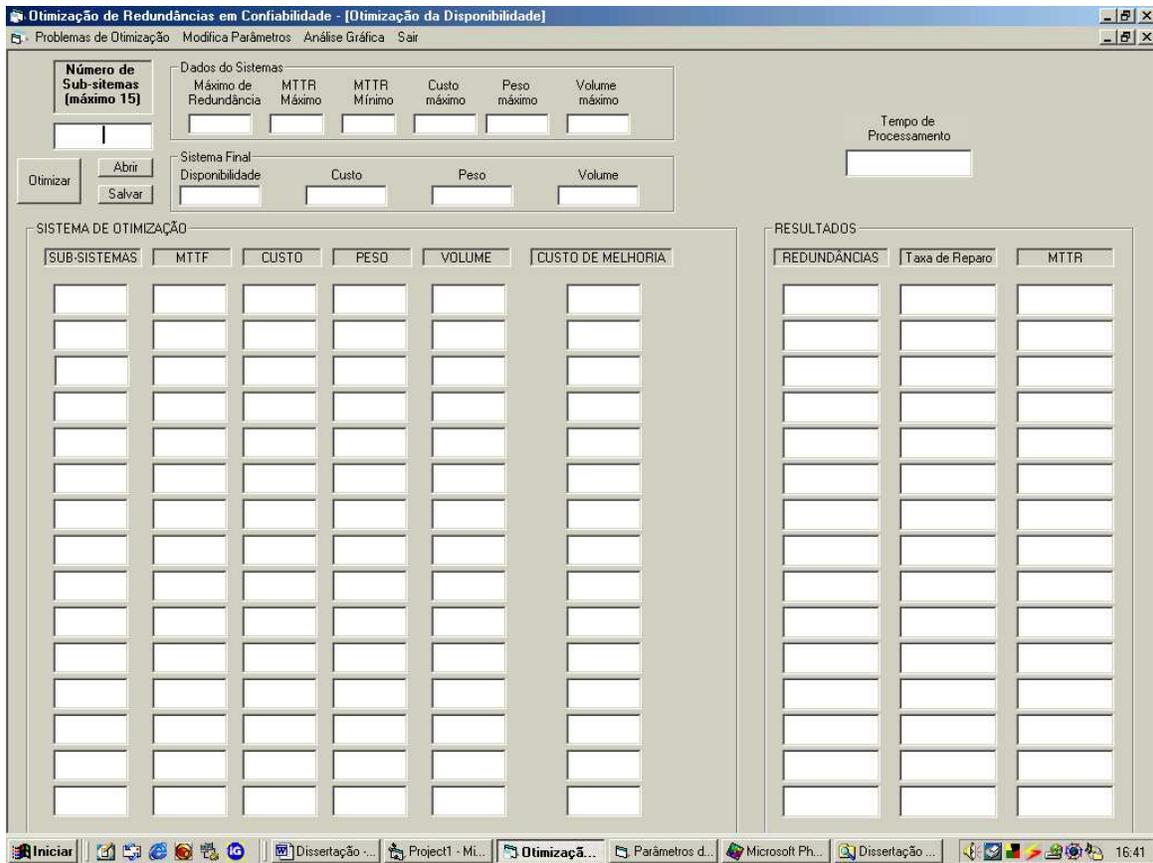


Figura 5.10 – Tela da Maximização da Disponibilidade (Problema 4 e 5)

O quinto problema é semelhante ao quarto, diferenciando apenas a variável que é otimizada. Neste caso, o MTTF, e conseqüentemente, a taxa de falha é variável, enquanto que o MTTR é entrada. A tela representada pela Figura 5.10 também é utilizada neste caso, mudando apenas os nomes das variáveis de entrada e saída.

Os problemas seis e sete têm o MTTR e o MTTF de cada componente como entrada. As saídas são respectivamente o número de equipes de manutenção e recursos de manutenção, além das saídas comuns a todos os problemas de maximização da disponibilidade. O total de equipes de manutenção é uma entrada específica do problema seis, e o impacto dos recursos de manutenção no MTTR é uma entrada do problema sete. A Figura 5.11 representa a tela dos problemas seis e sete, sendo que os nomes das variáveis mudam para cada um dos problemas.

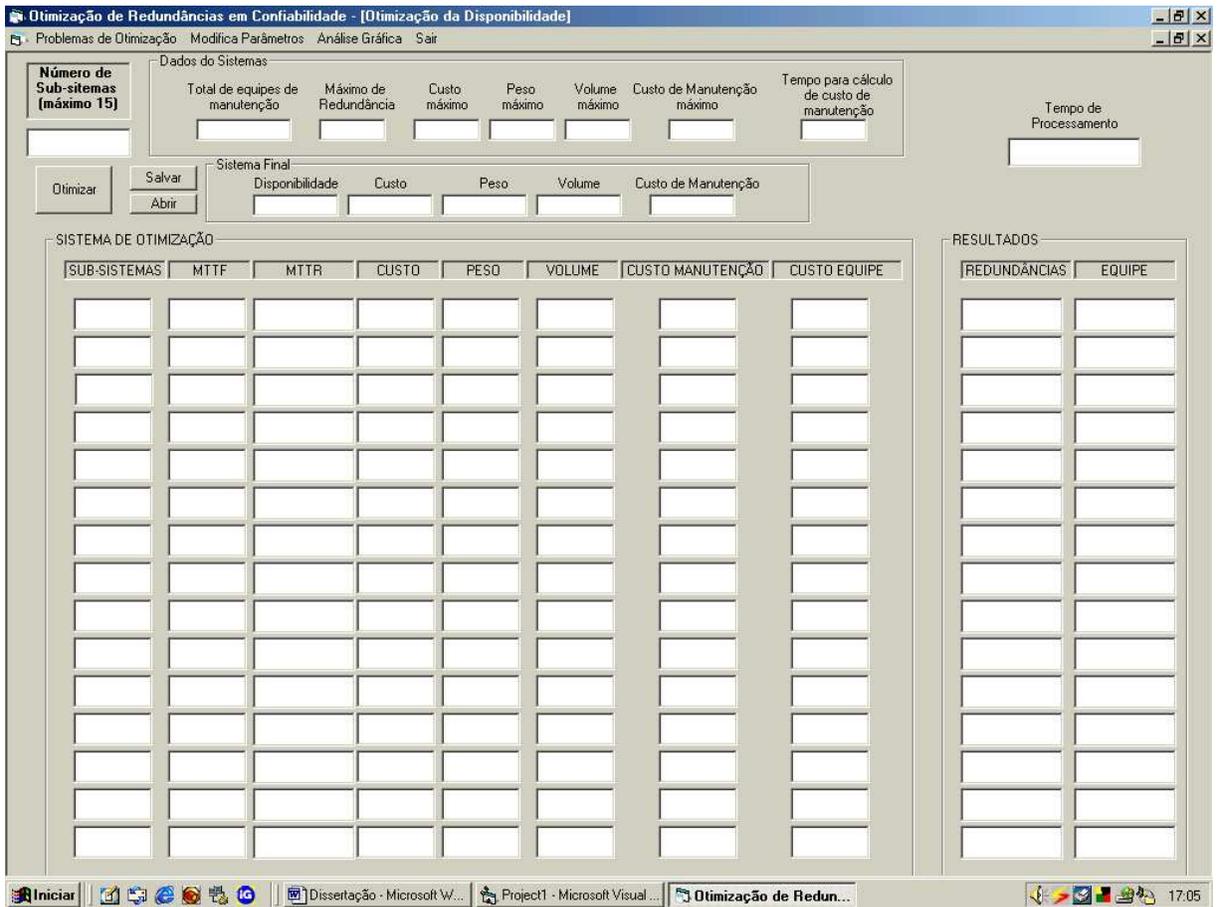


Figura 5.11 - Tela da Maximização da Disponibilidade (Problema 6 e 7)

### 5.5.3 Análise Gráfica

Uma seção de análise gráfica foi desenvolvida no programa, a fim de avaliar a convergência do problema e as gerações do Algoritmo Genético, conforme é mostrado na Figura 5.12:

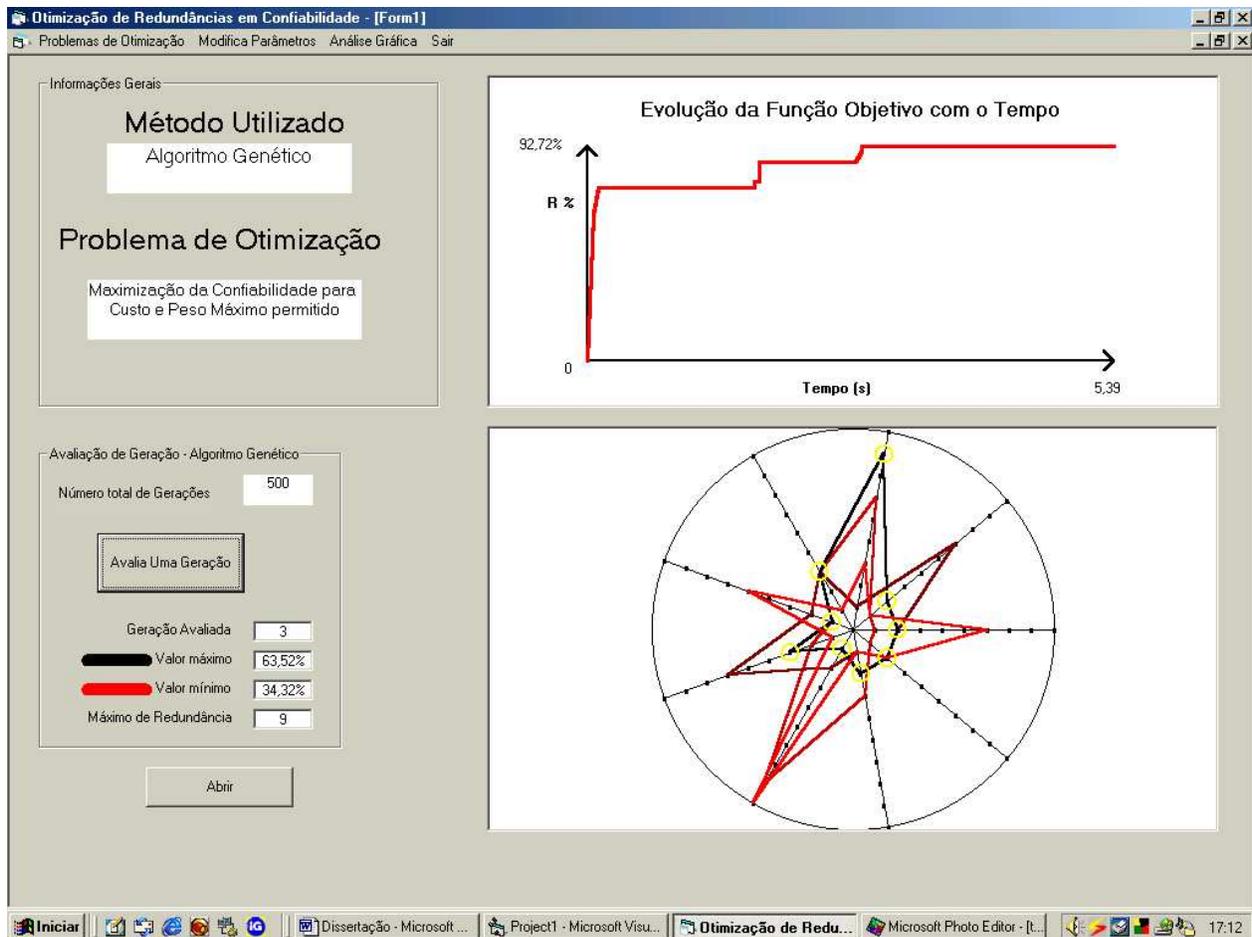


Figura 5.12 - Análise Gráfica (Otimização da Confiabilidade)

A gráfico superior representa a variação da função objetivo com o tempo, e é válido para todos os métodos, enquanto que o gráfico inferior representa uma geração do Algoritmo Genético para a otimização da confiabilidade. Cada seguimento do gráfico representa um subsistema e os pontos o número de redundâncias. O subsistema um é representado pelo seguimento a zero grau da horizontal e o crescimento é no sentido anti-horário. As linhas que ligam os pontos dos seguimentos representam os indivíduos da população, sendo que o indivíduo com maior fitness tem cor preta, e com menor tem cor vermelha. Uma linha que cruze o primeiro seguimento no ponto seis, representa um indivíduo que tem seis componentes no subsistema um.

Análise gráfica para a maximização da disponibilidade é representada pela Figura 5.13:

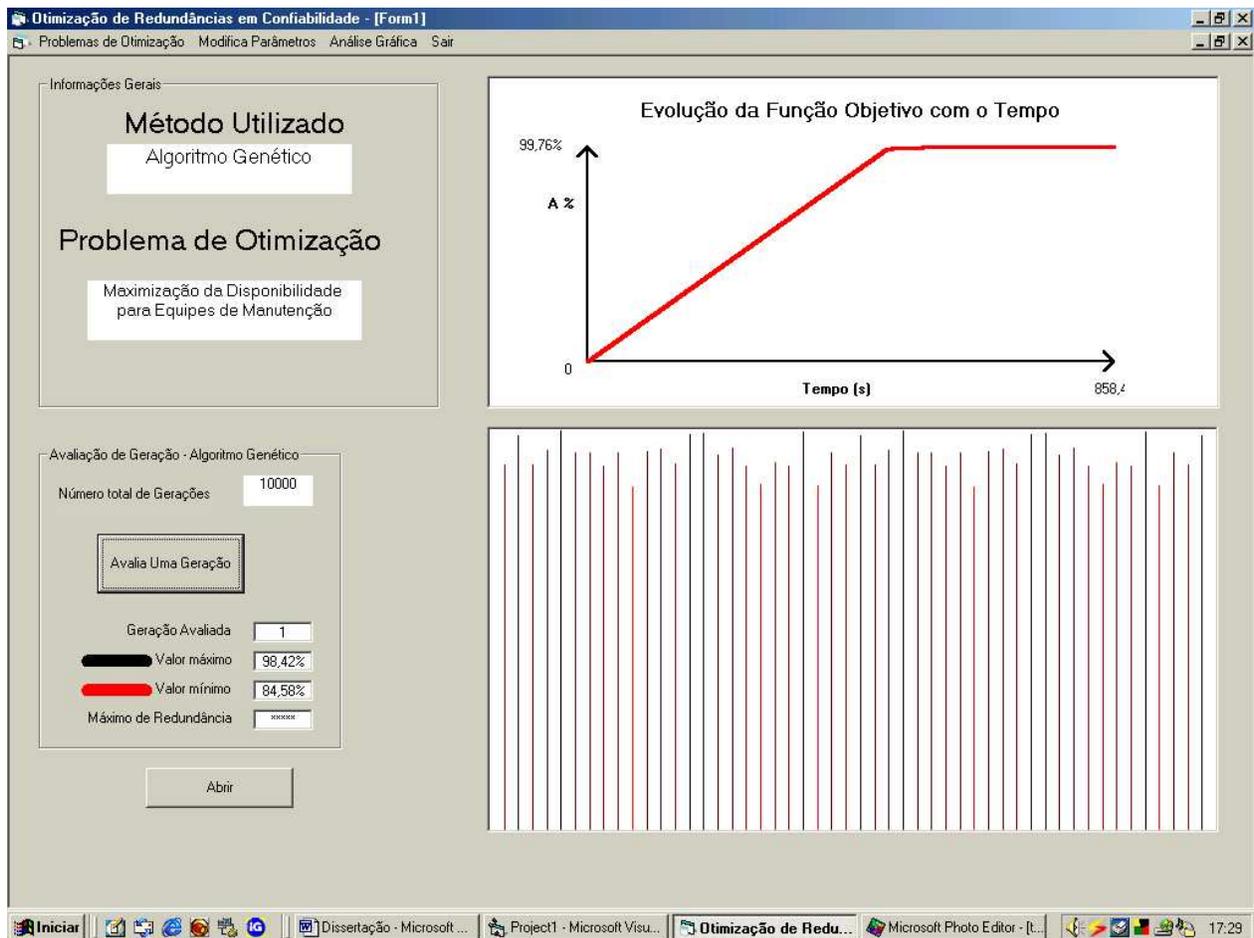


Figura 5.13 - Análise Gráfica (Otimização da Disponibilidade)

Neste caso, há dois tipos de variáveis de respostas, e em apenas um caso (Problema 6) as duas são números inteiros, o que inviabiliza a representação de uma geração como no gráfico anterior. Portanto, cada linha perpendicular representa um indivíduo da população e as cores e comprimentos da linha variam acordo com o valor do fitness do indivíduo. Então, um indivíduo com melhor fitness possui um comprimento maior e é preto, enquanto aquele que possui o fitness menos adequado, tem comprimento menor e é vermelho. Os demais indivíduos possuem cores e comprimentos intermediários.



## Capítulo 6

### Aplicações e Exemplos

Foram realizados testes numéricos para comparar os métodos de otimização dos problemas 1 a 3, assim como avaliar a influência dos parâmetros do Algoritmo Genético na convergência do problema.

Além disto, exemplos reais foram aplicados aos problemas 4 a 7, validando o método desenvolvido.

#### 6.1 Comparação entre os Métodos

Banerjee e Rajamani (1973) aplicaram três exemplos para os problemas 1 a 3. Para comparação dos métodos desenvolvidos, estes problemas foram representados pelas seguintes tabelas:

<b>Estágio</b>	<b>Custo</b>	<b>Confiabilidade</b>
1	3,0	95%
2	2,5	92%

3	2,0	90%
4	1,5	85%
Restrição		99,9%

Tabela 6.1 – Exemplo de sistema para o Problema 1

<b>Estágio</b>	<b>Custo</b>	<b>Confiabilidade</b>
1	4,0	90%
2	3,0	85%
3	2,0	80%
4	1,0	70%
Restrição	30,0	

Tabela 6.2 – Exemplo de sistema para o Problema 2

<b>Estágio</b>	<b>Custo</b>	<b>Peso</b>	<b>Confiabilidade</b>
1	5	8	90%
2	4	9	75%
3	9	6	65%
4	7	7	80%
5	7	8	85%
Restrição	100	104	

Tabela 6.3 – Exemplo de sistema para o Problema 3

A solução ótima do exemplo 1, assim como os valores finais do custo e confiabilidade do sistema são dados pela Tabela 6.4:

<b>Redundâncias no estágio 1</b>	<b>Redundâncias no estágio 2</b>	<b>Redundâncias no estágio 3</b>	<b>Redundâncias no estágio 4</b>	<b>Confiabilidade</b>	<b>Custo</b>
3	3	4	5	99,92%	\$32

Tabela 6.4 – Resultado ótimo para o exemplo 1

O passo de decremento  $\Delta v$  do multiplicador de Lagrange foi assumido como igual a dois. O total de iterações para a estratégia evolutiva foi de 1.000.000, enquanto que o número máximo de resultados repetidos foi de 10 e o erro percentual mínimo, para o segundo critério de parada (seção 5.3) foi de 1%. No Algoritmo Genético, considerou-se um total de 500 gerações para uma população de 50 indivíduos, com probabilidades de mutação e crossover iguais a 90% e 10% respectivamente.

Todos os métodos convergiram com os parâmetros que foram adotados. O método dos multiplicadores de Lagrange convergiu, praticamente, em zero segundos, enquanto que a ES e o GA convergiram, respectivamente, em 1,81 e 4,18 segundos em um processador Pentium 4 com 1,6 GHz de velocidade e 128 MB de memória RAM. As figuras seguintes, obtidas da análise gráfica do problema, mostram a evolução da função objetivo (custo) com os métodos ES e GA.

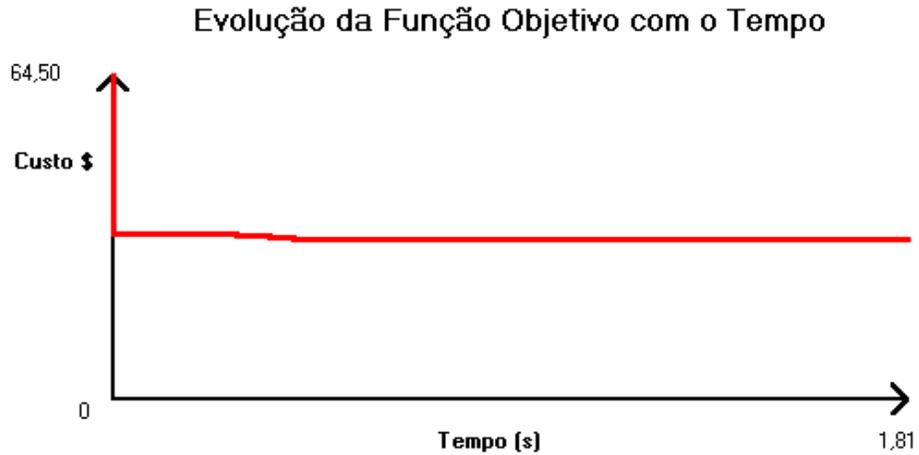


Figura 6.1 – Evolução do exemplo um com o tempo para Estratégia Evolutiva

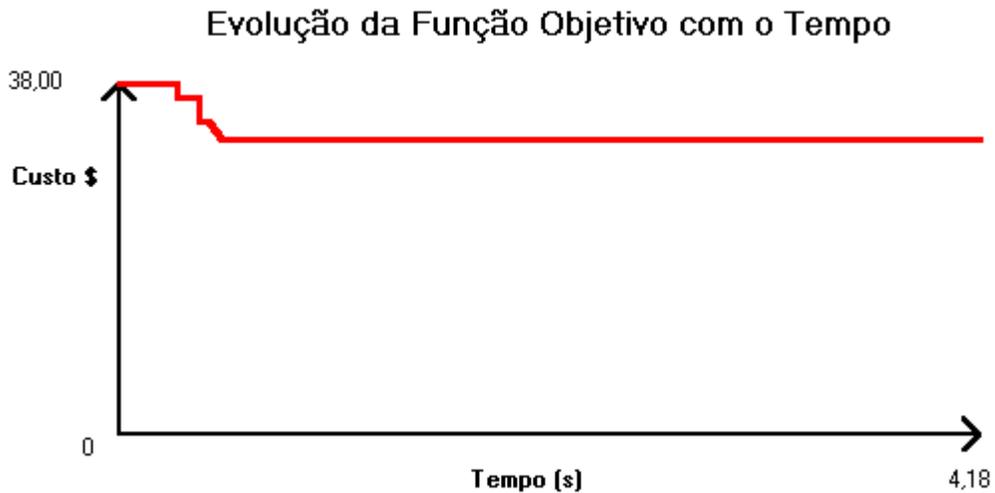


Figura 6.2 – Evolução do exemplo um com o tempo para Algoritmo Genético

O ponto ótimo é atingido rapidamente pelos métodos representados pelas figuras acima. Entretanto, o longo período para finalização garante a convergência do método. As condições de paradas podem ser mudadas para diminuir o tempo de convergência, porém a probabilidade de se atingir o ponto ótimo seria menor, e mesmo com os parâmetros adotados o tempo de convergência é pequeno.

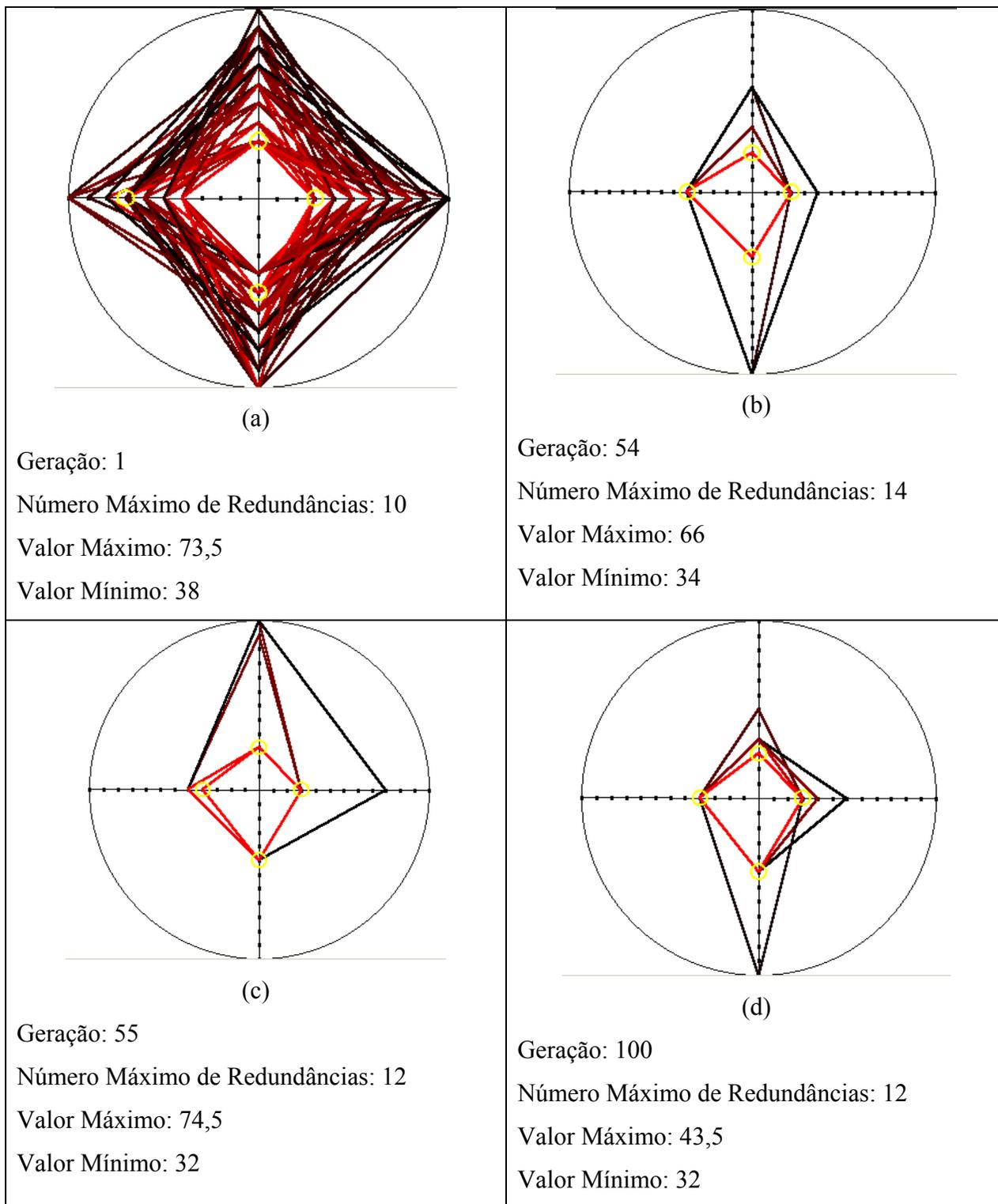


Figura 6.3 – Gerações do Exemplo 1

A Figura 6.3 acima mostra quatro gerações do GA utilizado na solução do Problema 1. Nota-se que na primeira geração (Figura 6.3 (a)) a diversidade de indivíduos é muito grande. O Ponto ótimo é atingido na geração 55, mostrado na comparação da Figura 6.3 (b) e Figura 6.3 (c).

Aos exemplos 2 e 3 também foram aplicados cada um dos métodos. O resultado ótimo de cada um deles é mostrado nas tabelas a seguir:

<b>Redundâncias no estágio 1</b>	<b>Redundâncias no estágio 2</b>	<b>Redundâncias no estágio 3</b>	<b>Redundâncias no estágio 4</b>	<b>Confiabilidade</b>	<b>Custo</b>
2	3	4	5	98,27%	\$30

Tabela 6.5 – Resultado ótimo para o exemplo 2

<b>Redundâncias nos estágios</b>					<b>Confiabilidade</b>	<b>Custo</b>	<b>Peso</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>			
2	3	4	3	2	93,08%	93	104

Tabela 6.6 – Resultado ótimo para o exemplo 3

O método multiplicadores de Lagrange convergiu com um passo de convergência de 0,001 no multiplicador  $v$ . O tempo de convergência foi praticamente instantâneo.

A fim de se obter o ponto ótimo através do algoritmo Genético, o número total de gerações foi aumentado para 1000 gerações, enquanto que os demais parâmetros se mantiveram os mesmos do exemplo anterior. O tempo de convergência foi de 9,34 segundos. A Figura 6.4 mostra a evolução da confiabilidade em função do tempo para este caso.

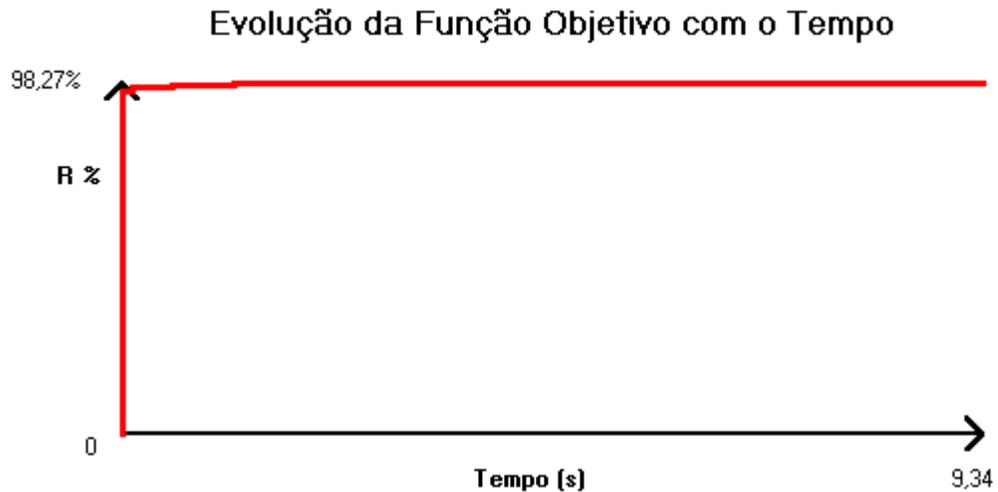


Figura 6.4 - Evolução do exemplo dois com o tempo para Algoritmo Genético

Para os mesmos parâmetros do exemplo anterior, o método da estratégia evolutiva convergiu em 1,76 segundos.

Com um valor inicial para  $v_1$  de 0,001 e  $v_2$  de 0,004 e um passo de 0,001, o método dos multiplicadores convergiu para o ponto ótimo exemplo 3. O Algoritmo Genético convergiu em 6,04 segundos, para os parâmetros de convergência iguais aos do exemplo 1. Da mesma forma, a Estratégia Evolutiva convergiu em 8,84 segundos.

O tempo de convergência mostrou que o algoritmo Genético é mais indicado do que a estratégia Evolutiva para dois indivíduos quando a complexidade do problema aumenta.

Um outro exemplo, representado pela Tabela 6.7, foi solucionado pelos métodos de otimização. Por se tratar de um exemplo mais complexo que os anteriores, ele permitiu a realização de uma análise de desempenho dos métodos em problemas complexos. O número ótimo de redundâncias e os resultados finais do sistema ótimo também estão mostrados na tabela:

<b>Estágio</b>	<b>Custo</b>	<b>Peso</b>	<b>Confiabilidade</b>	<b>Resultado Ótimo</b>
1	5	8	90%	2
2	4	9	75%	3
3	9	6	65%	5
4	7	7	80%	3
5	7	8	85%	3
6	8	9	75%	4
7	9	7	90%	2
8	8	8	85%	3
9	6	9	75%	3
Restrição	215	220		
Sistema Ótimo	201	219	92,72%	

Tabela 6.7 – Exemplo com nove estágios

O tamanho de população para o GA foi de 70 indivíduos, enquanto que o número total de gerações foi de 1500 gerações. O tempo de convergência neste caso foi de 42 segundos.

A Estratégia Evolutiva atingiu uma confiabilidade de 91,77% em 1.000.000 iterações, enquanto que o método dos multiplicadores de Lagrange não convergiu aos valores iniciais de  $v_1$  e  $v_2$ , que apresentaram incompatibilidade numérica na resolução do sistema não-linear representado pela Equação 5.57.

## 6.2 Parâmetro do Algoritmo Genético

Os parâmetros do Algoritmo Genético (tamanho da população, total de gerações, probabilidade de mutação e probabilidade de *crossover*) influenciam na convergência e no tempo de convergência. Com o objetivo de se avaliar tal influência, realizou-se uma série de testes variando cada um destes parâmetros. O sistema com nove estágios foi utilizado nestes testes.

O número de réplicas para cada variação foi de dez. Não se optou por realizar uma análise estatística, como planejamento fatorial, mas uma análise gráfica, pois se deseja determinar apenas o comportamento do valor da função objetivo e do tempo de convergência com a variação destes parâmetros.

O número total de variações partiu de 100 gerações, de 100 em 100, até que 1000 gerações fossem atingidas. A Tabela 6.8 mostra esta variação. O tamanho da população, a probabilidade de mutação e de *crossover* permaneceram constantes, valendo 50 indivíduos, 90% e 10 % respectivamente.

geração	Tempo	Média da Confiabilidade	Desv. Pad. Da Confiabilidade
100	0,60	85,03	5,14
200	1,10	86,86	3,88
300	1,61	90,33	1,77
400	2,13	91,05	1,31
500	2,65	91,02	1,73
600	3,16	91,71	1,34
700	3,68	92,03	0,92
800	4,20	92,47	0,45
900	4,72	92,19	0,74
1000	5,24	92,58	0,31

Tabela 6.8 - Variação do total de gerações

A confiabilidade média cresce lentamente com o aumento do total de gerações, como mostrado na Figura 6.5. O tempo de convergência apresenta uma relação linear com o total de gerações, como fica evidenciado na Figura 6.6.

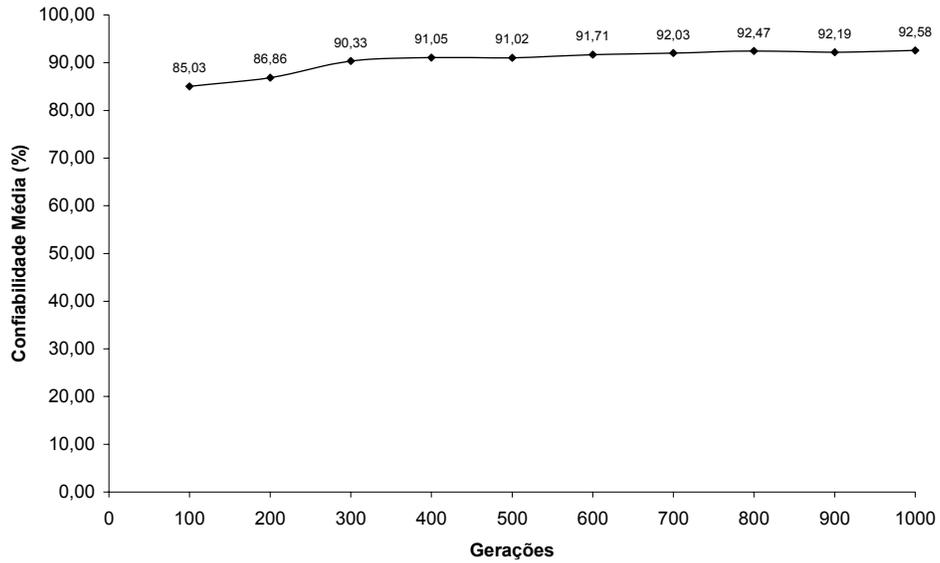


Figura 6.5 - Variação da Confiabilidade Média com o total de Gerações

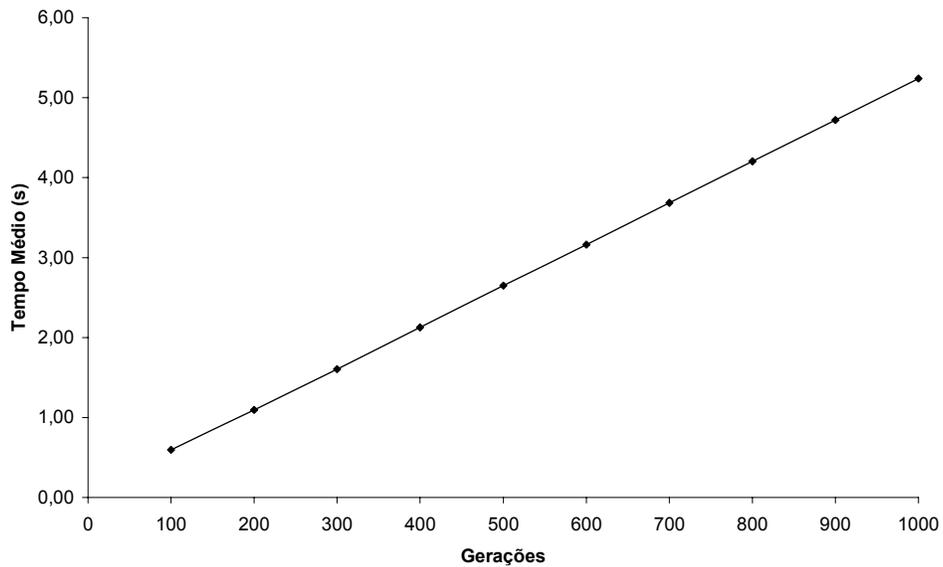


Figura 6.6 – Variação do Tempo de Convergência com o Total de Gerações

A influência do tamanho da população também foi avaliada da mesma forma. As médias da confiabilidade e do tempo de convergência, assim como o desvio padrão da confiabilidade para um total de 500 gerações e valores de probabilidades de mutação e crossover iguais aos da análise anterior, estão na Tabela 6.9:

<b>população</b>	<b>Tempo</b>	<b>Média da Confiabilidade</b>	<b>Desv. Pad. da Confiabilidade</b>
10	0,24	83,36	5,64
20	0,54	88,28	3,96
30	1,00	89,91	1,73
40	1,67	91,53	1,14
50	2,64	91,51	1,59
60	3,97	92,02	1,00
70	5,74	92,34	0,37
80	7,99	92,35	0,01
90	10,81	92,55	0,23
100	14,27	92,54	0,29

Tabela 6.9 - Variação do tamanho da população

A média da confiabilidade cresce lentamente com aumento do tamanho da população, mas o tempo de convergência apresenta um crescimento bastante acentuado. Os gráficos nas Figuras 6.7 e 6.8 ilustram estes comportamentos:

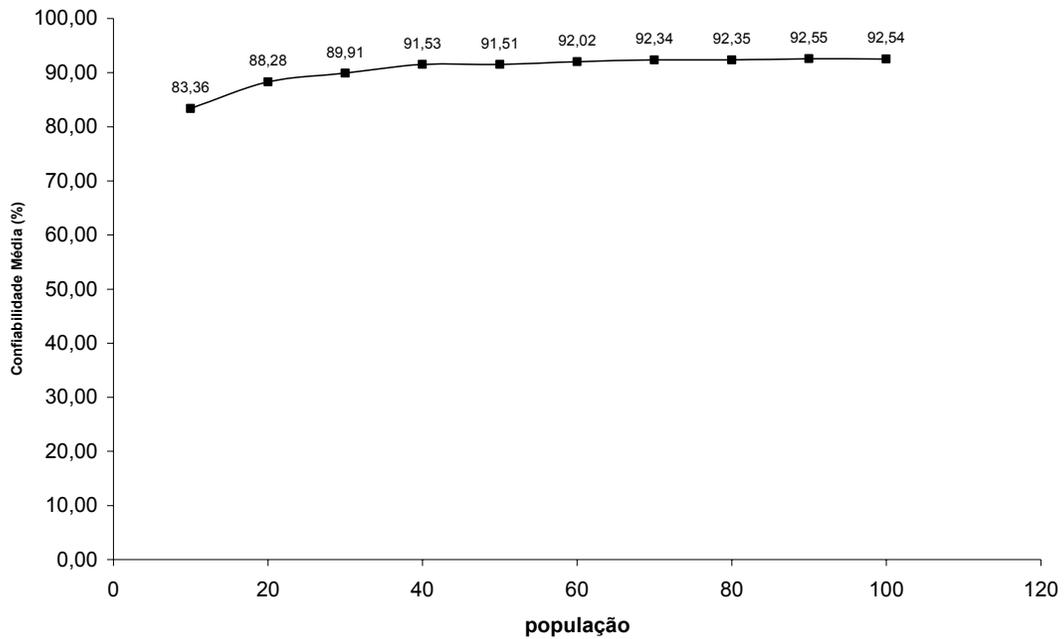


Figura 6.7 – Variação da Confiabilidade Média com o Tamanho da População

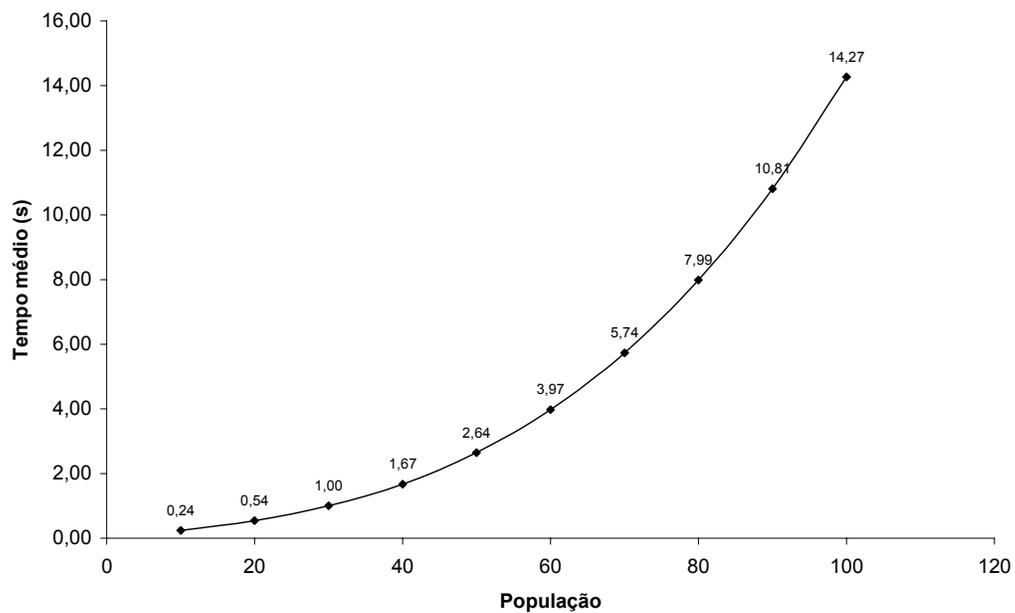


Figura 6.8– Variação do Tempo de Convergência com o Tamanho da população

Para a avaliação da influência da probabilidade de *crossover*, fixou-se um total de 1000 gerações, um tamanho de população de 50 indivíduos e uma probabilidade de mutação de 10%. A tabela seguinte mostra o resultado destes testes:

Probabilidade de crossover	Tempo médio	Média da Confiabilidade	Desv. Pad. Da Confiabilidade
10	5,02	85,54	5,73
20	5,01	77,68	10,25
30	5,00	74,25	9,04
40	5,00	77,34	11,44
50	5,01	74,27	10,24
60	5,01	77,70	6,83
70	5,02	71,55	13,26
80	5,02	75,16	8,91
90	5,02	64,21	7,39
100	5,03	65,77	4,94

Tabela 6.10 – Variação da probabilidade de crossover

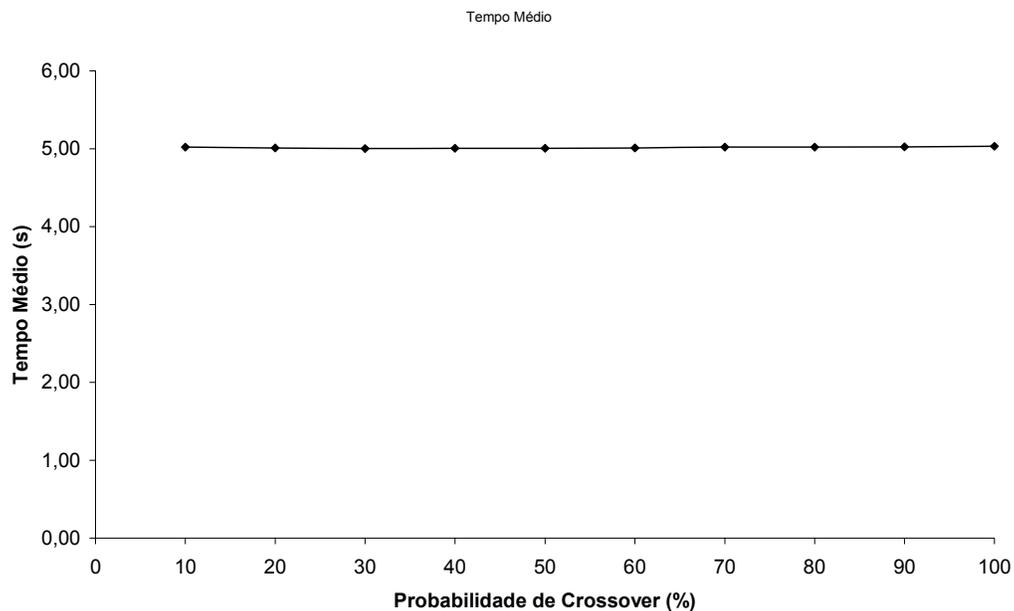


Figura 6.9 – Variação do tempo de convergência com a probabilidade de crossover

Conforme a Figura 6.9, o tempo de convergência não sofre alteração com a variação da probabilidade de crossover. A média da confiabilidade conforme a Figura 6.10, a seguir, decai com esta variação.

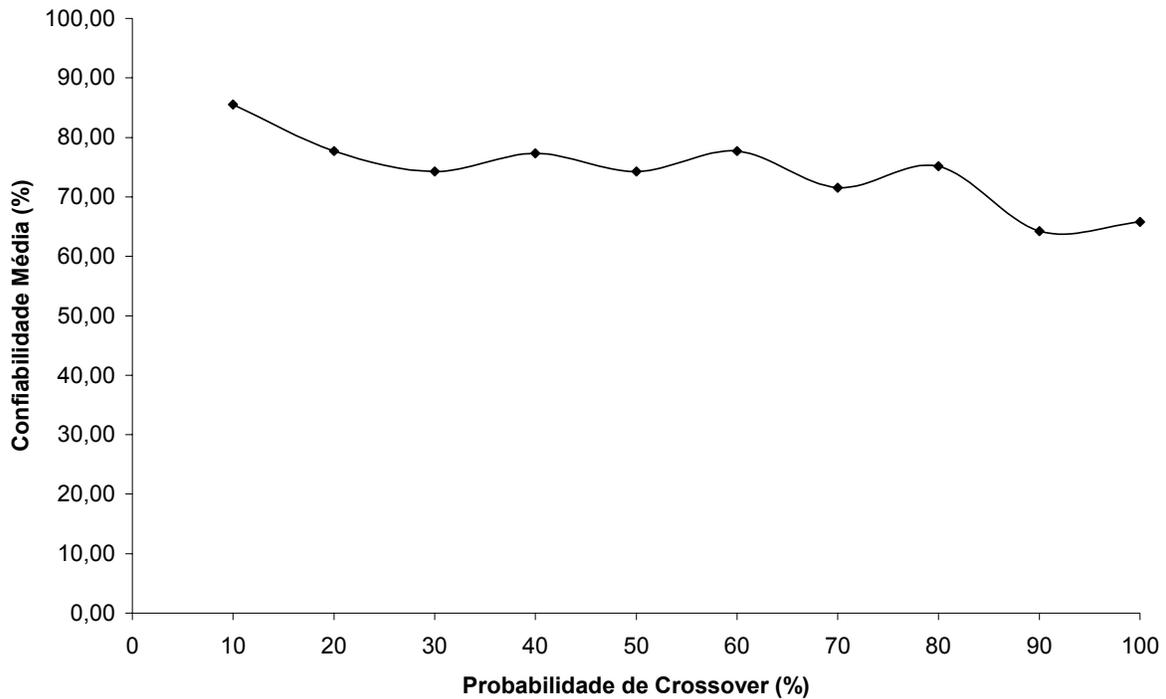


Figura 6.10 – Variação da Média da Confiabilidade com a probabilidade de *Crossover*

Para avaliar a influência da probabilidade de mutação, realizaram-se dois testes, um com probabilidade de *crossover* igual a 0% e outro igual a 10%. Em ambos os casos, o total de gerações foi de 1000 gerações e o tamanho da população foi de 50 indivíduos.

<b>Probabilidade de Mutação</b>	<b>Tempo</b>	<b>Média da Confiabilidade</b>	<b>Desv. Pad. da Confiabilidade</b>
10	5,20	84,09	5,86
20	5,25	88,60	2,22
30	5,29	89,72	1,57
40	5,31	89,31	1,77
50	5,33	88,99	3,71
60	5,35	89,13	2,03
70	5,36	88,42	0,77
80	5,37	89,11	1,52
90	5,38	89,27	2,34
100	5,40	88,48	2,04

Tabela 6.11 – Variação da Probabilidade de Mutação com probabilidade de crossover igual a 0%

<b>Probabilidade de Mutação</b>	<b>Tempo</b>	<b>Média da Confiabilidade</b>	<b>Desv. Pad. Da Confiabilidade</b>
10	5,02	85,54	5,73
20	5,06	90,00	1,64
30	5,10	90,66	2,08
40	5,13	91,73	0,88
50	5,16	92,31	0,54
60	5,18	92,00	0,51
70	5,20	92,50	0,33
80	5,22	92,19	0,70
90	5,24	92,35	0,90
100	5,26	92,51	0,26

Tabela 6.12 – Variação da Probabilidade de Mutação com probabilidade de crossover igual a 10%

As Figuras 6.11 e 6.12 mostram a influência da variação probabilidade de mutação no tempo de convergência e na confiabilidade média respectivamente:

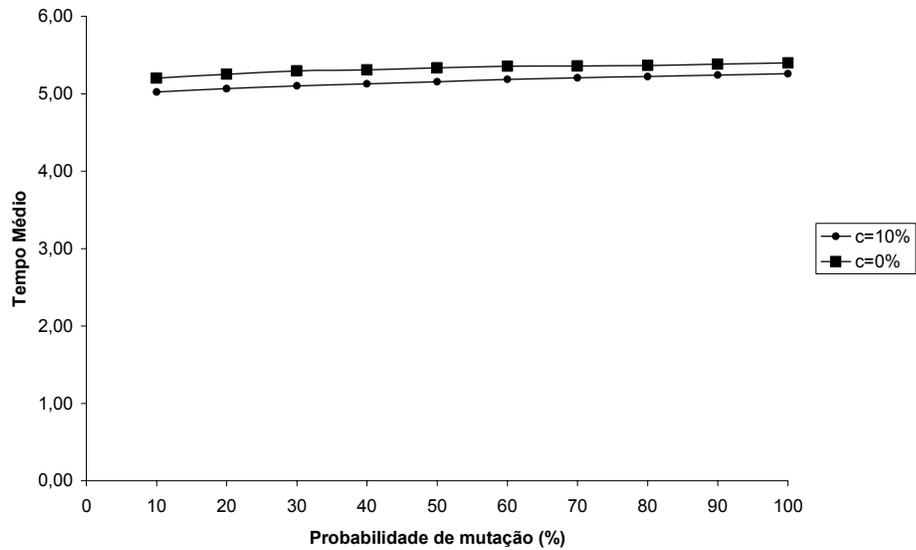


Figura 6.11 – Variação do tempo de convergência com a probabilidade de mutação

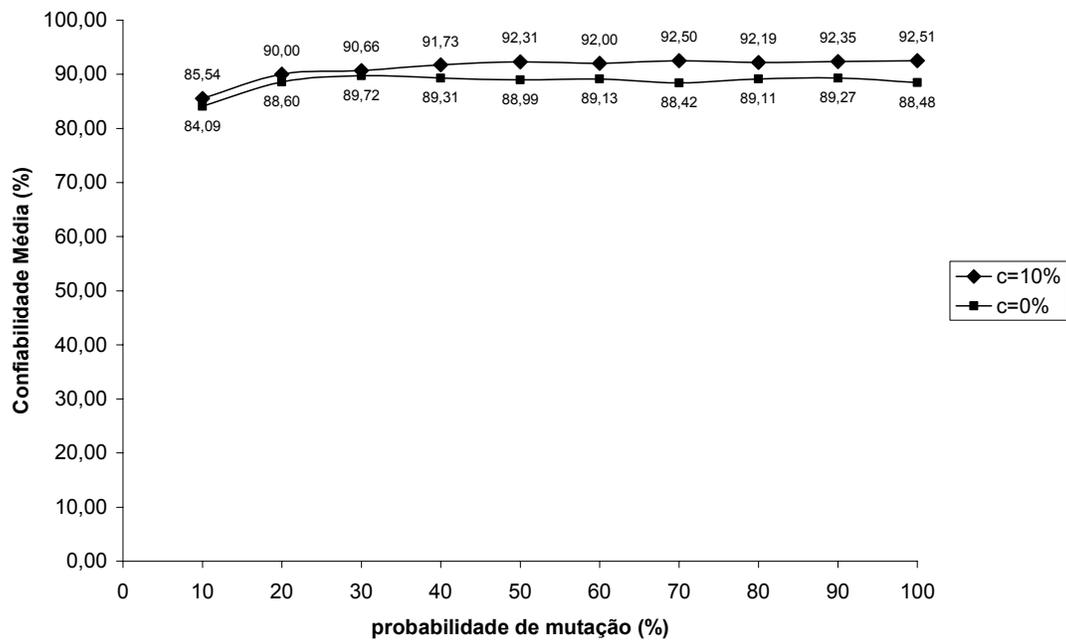


Figura 6.12 – Variação da Média da Confiabilidade com a probabilidade de mutação

Nota-se que com 10% de probabilidade de *crossover* o tempo de convergência foi menor e a média da confiabilidade maior. Entretanto, como mostrado na Figura 6.10, a confiabilidade tende a cair com o aumento da probabilidade de *crossover*. Portanto, para o algoritmo adotado, aconselha-se uma alta probabilidade de mutação, 90% por exemplo, e baixa probabilidade de *crossover*, 10 % por exemplo.

O tamanho da população e o total de gerações devem variar de acordo com a complexidade do sistema a ser otimizado. Devido à forte influência do crescimento da população no tempo de convergência, deve-se, preferencialmente, variar o número total de gerações.

### 6.3 Exemplos Reais

Para simular os problemas quatro a sete foi usado um exemplo de uma linha de produção da ZF do Brasil, onde é produzida uma carcaça de freio (Figura 6.13).



Figura 6.13 - Carcaça do Freio

### 6.3.1 Processo de Produção

Cinco máquinas-ferramenta estão envolvidas da produção desta peça. Primeiramente, dois tornos verticais robotizados em série ( T8901 e T8902), dois centros de usinagem em paralelo (CE25377 e CE25378) e uma entalhadora (EN51004). A Figura 6.14 representa o esquema de produção desta peça. O sistema de produção é mostrado na Figura 6.15.

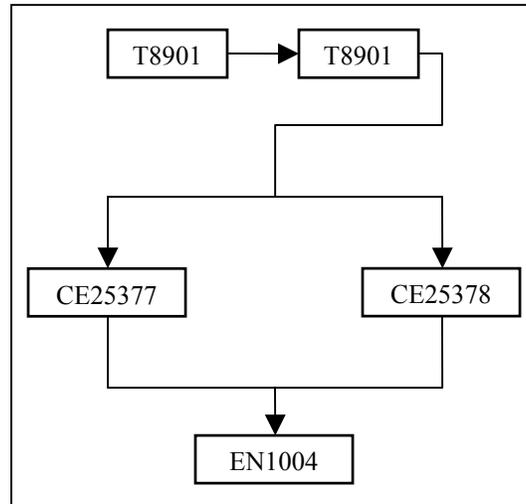


Figura 6.14 - Processo de Produção



Figura 6.15 - Linha de Produção

### 6.3.2 Dados das Máquinas-Ferramenta

A fim de se realizar as simulações, foram levantados os valores o MTTF e o MTTR para cada equipamento. Além disto, estimou-se preço de cada um dos equipamentos, assim como a área ocupada. Como a massa não é um parâmetro importante, será considerada nula para cada equipamento, para não influenciar na simulação. A Tabela 6.13 mostra os valores destes parâmetros.

<b>Equipamento</b>	<b>MTTF (horas)</b>	<b>MTTR (horas)</b>	<b>Preço (R\$)</b>	<b>Área (m<sup>2</sup>)</b>	<b>Figura</b>
T8901 e T 8902	7670	42	500.000	15	Figura 6.16
CE25377 e CE25378	5200	120	700.000	20	Figura 6.17
EN51004	5700	160	200.000	15	Figura 6.18

Tabela 6.13 – Dados dos Equipamentos

Nas análises realizadas não foi considerada a demanda de produção, apenas os tempos de falha e de reparo.



Figura 6.16 - Torno Vertical Robotizado



Figura 6.17 – Centro de Usinagem



Figura 6.18 – Entalhadora

### 6.3.3 Simulação da maximização da Disponibilidade com o MTTR como variável

Para simular este problema (Problema 4, seção 5.1.4), considerou-se o custo de melhoria da razão de dependabilidade  $cd_i$  igual a R\$ 200,00 para os tornos e R\$ 300,00 para os demais equipamentos, pois estes são mais antigos. O custo total do sistema foi assumido como sendo R\$



A disponibilidade máxima atingida foi de 99,94%. Como mostra a Figura 6.20, valores muito próximos a disponibilidade máxima são atingidos no início do processo de otimização. Portanto, seria interessante o uso de métodos de otimização multiobjetivo, onde o custo também seria uma função objetivo, pois é possível obter alta disponibilidade para um custo menor.

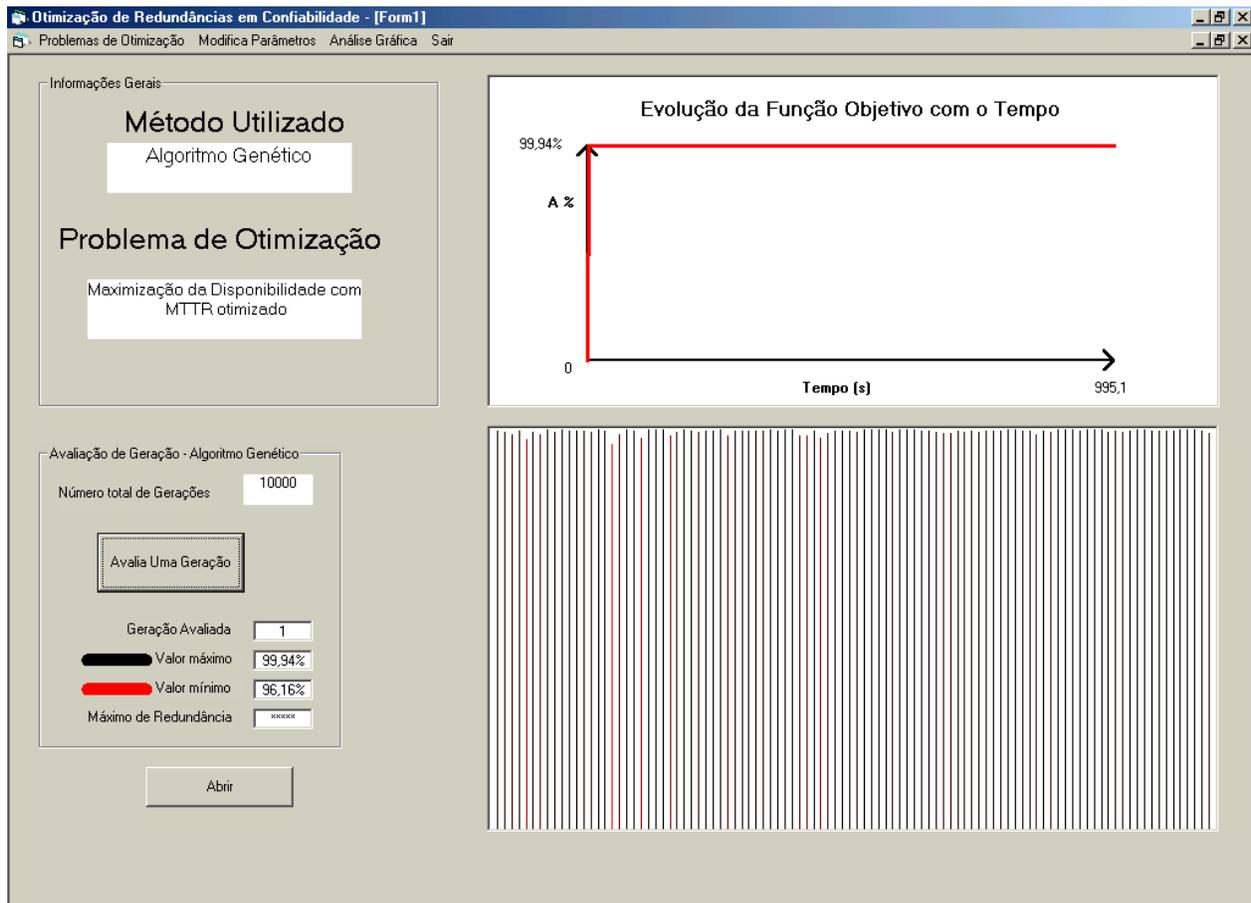


Figura 6.20 – Análise Gráfica do Problema 4 na primeira geração

#### 6.3.4 Simulação da maximização da Disponibilidade para Equipes de Manutenção

A utilização de equipes de manutenção é uma forma de se diminuir o MTTR de um subsistema. A formulação deste problema se encontra no item 5.1.6.

O custo mensal do técnico de manutenção é de R\$ 22,28. Foi considerado um total de 180 horas (próximo a um mês de trabalho) para o cálculo da probabilidade de falha, utilizada na expressão do custo de manutenção. Os demais dados de entrada estão na Tabela 6.13. Assim como no problema anterior, o custo total do sistema foi assumido como sendo R\$ 5.000.000,00 e a área total de 100 m<sup>2</sup>. O custo máximo de manutenção foi assumido como R\$ 3.000,00.

A fim de que a convergência do problema seja garantida, foi considerado um total de 5.000 gerações, enquanto que o tamanho da população se manteve em 50 indivíduos. A probabilidade de mutação, *crossover* e inversão valem respectivamente 90%, 10% e 70%.

A confiabilidade máxima atingida foi de 98,79%, para um sistema com 1 componente e 1 equipe nos dois primeiros estágios (tornos) e dois componentes e duas equipes nos demais estágios (centro de usinagem e entalhadora).

A principal restrição foi a área, pois os demais limites de restrição foram elevados para os resultados atingidos. A área total do sistema final foi de 100 m<sup>2</sup>, igual ao limite de restrição. A tela do software com o sistema ótimo é mostrada na Figura 6.21:



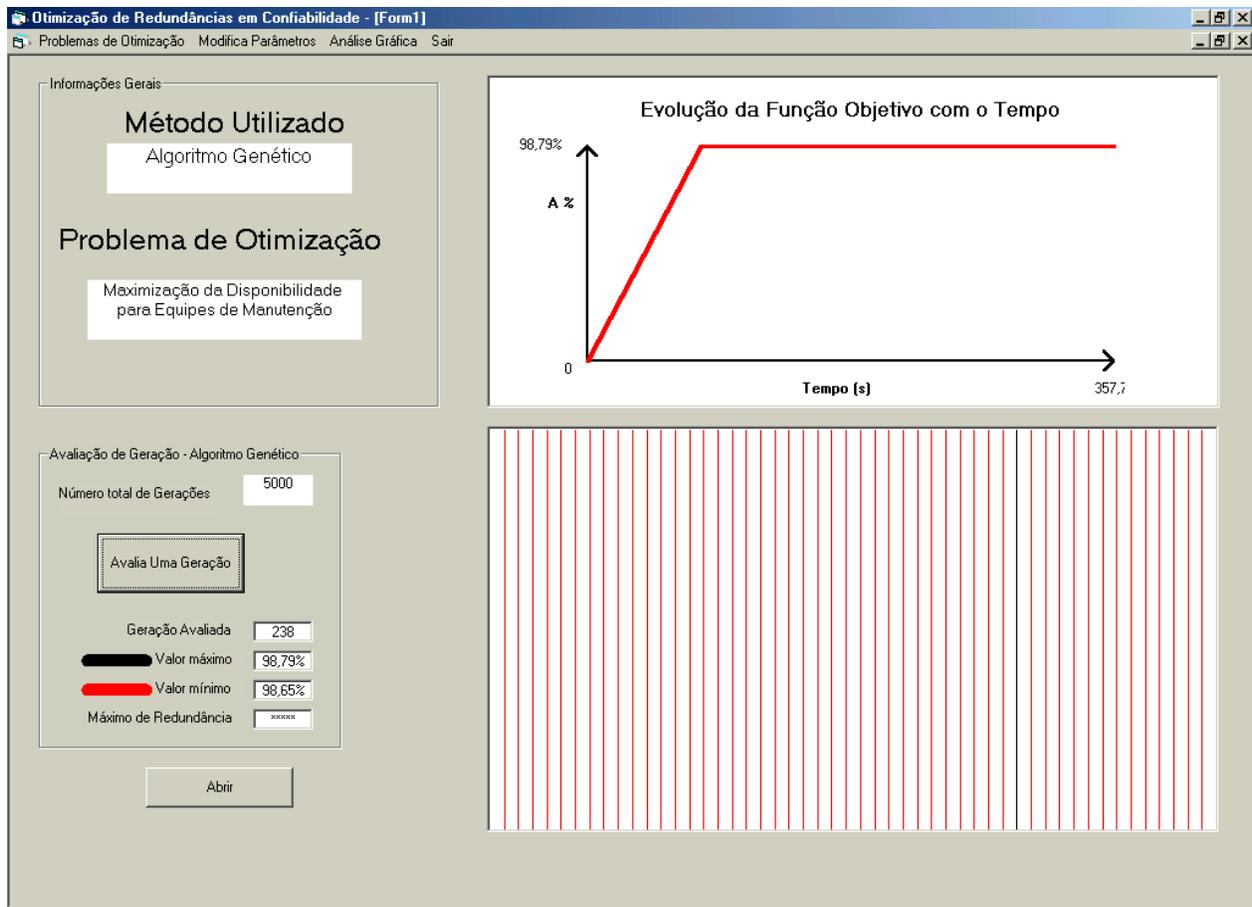


Figura 6.22 – Análise Gráfica do Problema 6 na geração 238

### 6.3.5 Simulação da maximização da Disponibilidade para Recursos de Manutenção

O problema simulado, desenvolvido no item 5.1.7, otimiza o número de redundância e os recursos de manutenção, que devem ser empregados em cada subsistema.

Os altos custos dos equipamentos e o espaço disponível na fábrica inviabilizam a instalação de equipamentos redundantes. Portanto, apenas os recursos de manutenção serão otimizados.



A disponibilidade máxima atingida foi de 98,51%, sendo que 12% de recurso de manutenção foi aplicado no subsistema 1, 20% no subsistema 2, 1% no subsistema 3 e 65% no subsistema 4.

Os tornos, devido à alta disponibilidade, não necessitam de muito recurso de manutenção. Como os centros de usinagens são redundantes, é necessária a aplicação de menos recursos de manutenção. Entretanto, não foram considerados os tempos de processo e de cada máquina, então a falha de um centro de usinagem pode levar a uma queda de produtividade da linha.

A aplicação do Algoritmo Genético em problemas complexos de otimização de sistemas redundantes é bastante eficiente, por causa da robustez inerente ao método. Outros métodos de otimização, como os multiplicadores de Lagrange, podem não ser eficientes quando aplicado a estes problemas, pois a não linearidade das funções e a necessidade do uso de conceitos matemáticos mais avançados dificultam a implementação computacional.



## Capítulo 7

### Conclusões e Perspectivas Futuras

O trabalho realizado objetivou a utilização e testes de métodos de otimização em problemas de confiabilidade e disponibilidade de sistemas redundantes de engenharia. Três métodos foram utilizados. O método dos Multiplicadores de Lagrange, utilizando algoritmos propostos por Banerjee e Rajamani (1973), a Estratégia Evolutiva para Dois Indivíduos, um algoritmo evolutivo proposto por Rechenberg (1973) e o Algoritmo Genético, que consiste no algoritmo evolutivo mais difundido e foi proposto por Holland (1975).

O método dos multiplicadores de Lagrange, por se tratar de um método analítico, exige que um complexo algoritmo seja desenvolvido para cada problema de otimização, como mostrado nas Figura 5.2 a Figura 5.4. Além disto, a necessidade da aplicação de cálculo diferencial e de conceitos matemáticos complexos dificultam a utilização em novos problemas de otimização da confiabilidade em sistemas redundantes. Por exemplo, não se procurou desenvolver algoritmos baseados neste método, para os problemas que envolviam o cálculo da disponibilidade através dos tempos de falha e manutenção.

Outro inconveniente dos algoritmos baseados no método dos multiplicadores de Lagrange, se mostrou na aplicação do problema três (maximização da confiabilidade para custo e peso máximo determinados) em um sistema com nove estágios, como mostrado no item 6.1. Neste

caso, o algoritmo apresentou inconsistência numérica, devido às entradas dos valores iniciais dos multiplicadores de Lagrange, nas iterações do método de Newton-Raphson, para a solução do sistema não-linear representado pela Equação 5.57.

Entretanto, a exatidão e velocidade de convergência deste método, para a maioria dos exemplos testados, devem ser levados em conta, pois os algoritmos são formulados para estes casos.

Tanto a Estratégia Evolutiva como o Algoritmo Genético apresentam fácil implementação computacional, devido a seu tratamento matemático trivial, que considera apenas as expressões da função objetivo e das funções de restrições. Estes métodos se diferenciam na aplicação em diferentes sistemas de engenharia. Para sistemas pequenos, ou mais simples, o Algoritmo Genético apresenta uma maior lentidão para se atingir o ponto ótimo, pois a quantidade de cálculos computacionais é excessiva neste caso. Por outro lado, para sistemas mais complexos (com mais variáveis), ele apresenta uma convergência mais rápida e eficiente, pois a Estratégia Evolutiva tem apenas um vetor de busca, enquanto que o Algoritmo Genético possui vários vetores (com número igual ao tamanho da população).

Logo, a Estratégia Evolutiva deve ser aplicada, preferencialmente, para sistema simples, como os exemplos da Tabela 6.1 e da Tabela 6.2.

A influência dos parâmetros do Algoritmo Genético no tempo e na precisão de convergência do Algoritmo Genético implementado nos problemas 1 a 3 foi verificada graficamente. Mostrou-se que o aumento da população (ou seja, o aumento dos vetores de busca) melhora a convergência do processo, mas o tempo de convergência apresenta crescimento acentuado (Figura 6.8). Em contra partida, o aumento do total de gerações apresenta um aumento linear do tempo de convergência (Figura 6.6), com influência semelhante na precisão de convergência. Desta forma, deve-se priorizar o aumento do total de gerações em problemas mais complexos.

Além da influência do tamanho da população e do total de gerações, as influências das probabilidades de mutação e crossover foram verificadas. O tempo de convergência não é afetado para as duas variáveis, mas o aumento das mesmas influencia, antagonicamente, na precisão de convergência. O aumento da probabilidade de crossover diminui o valor médio da função objetivo, como mostrado na Figura 6.10. Por outro lado, a média da função objetivo cresce com o aumento da probabilidade de mutação (Figura 6.12).

Entretanto, a Figura 6.12 mostra que para probabilidade de crossover nula, o aumento da probabilidade de mutação é menos significativo do que se a probabilidade de crossover for igual a 10%. Então, é necessário que se tenha um pouco de “mistura genética” nestes problemas, mas o principal instrumento adaptativo é a mutação.

A robustez do algoritmo Genético permite aplicá-lo a problemas de otimização mais complexos, como nos problemas 4 a 7. A utilização do método dos Multiplicadores de Lagrange a estes problemas exigiria o uso de matemática complexa. No caso da Estratégia Evolutiva para dois indivíduos, o custo computacional seria excessivo.

Além disto, a possibilidade da codificação binária nos Algoritmos Genéticos os credenciam para solução de problemas que envolvam variáveis de diferentes naturezas, como nos casos onde o número de redundância e a razão de dependabilidade foram as variáveis de otimização. Entretanto, esta abordagem aumenta, consideravelmente, o tempo de convergência. Nos problemas onde a confiabilidade foi otimizada, portanto apenas o número de redundância foi considerado, o tempo de convergência é consideravelmente menor, pois os efeitos dos operadores do Algoritmo Genético (mutação, crossover e inversão) são mais sensíveis. Por exemplo, a mutação, no caso binário, pode acarretar em nenhuma mudança no fenótipo do indivíduo, uma mudança muito brusca, onde as condições de restrição são quebradas, ou mudança aceitável. Por outro lado, no caso real, a mutação sempre acarretará em mudanças do fenótipo, que podem ser aceitas, ou que quebrem as condições de restrição por pouco.

A aplicação em casos reais mostrou que os problemas levantados podem ser aplicados industrialmente, mas muitas considerações devem ser feitas. Para os problemas quatro e cinco, o levantamento dos custos de melhoria da razão de dependabilidade é muito subjetiva. O problema seis apresenta uma aplicação com dados facilmente levantados, porém a utilização de equipes de manutenção da forma que é considerada restringe consideravelmente o campo de aplicação. Para o problema sete, utilizou-se o conceito de impacto da aplicação dos recursos de manutenção, o qual também é subjetivo, mas imprescindível para transformar a porcentagem de recursos em melhoria dos tempos de manutenção.

Os problemas levantados são úteis para a simulação de sistemas de engenharia, fornecendo sinalizações de onde devem ser aplicados os recursos (número de redundância, recursos de manutenção, recursos financeiros, equipes de manutenção) e com que ordem de grandeza esta aplicação deve ser feita.

Portanto, seria ideal que problemas gerados no ambiente industrial fossem levantados, a fim de aplicar o método de otimização desenvolvido, e não ajustar dados reais a problemas, a priori, formulados para testar os métodos. Uma sugestão para futuros trabalhos seria a aplicação do Algoritmo Genético a um problema de origem industrial.

Além disto, a Figura 6.20 e a Figura 6.22 mostraram, na evolução da função objetivo em função do tempo, um crescimento linear, seguido de um patamar horizontal, até o fim do processo de otimização. Este crescimento linear corresponde à primeira geração, onde apenas indivíduos que não quebrem as condições de restrição são computados. Desta forma, o valor da função objetivo na primeira geração é muito próximo ou igual ao valor ótimo, fazendo com que sejam necessárias várias gerações, para que ocorra alguma mudança no valor da função objetivo. No caso da simulação do problema 4, avaliado na Figura 6.20, não houve nenhuma mudança, e no caso da simulação do problema 6, uma única mudança ocorreu na geração 238, como é mostrado na Figura 6.22.

Por isso, a aplicação de métodos de otimização que acelerassem a convergência neste patamar é uma sugestão bastante útil para futuros trabalhos. Um método que pode ser utilizado é o *Simulated Annealing*, o qual simula o fenômeno físico do recozimento.

O principal parâmetro restritivo aplicado aos problemas propostos é o custo. Entretanto, devido à importância de se reduzir gastos, o custo pode ser tratado também como variável de otimização. Portanto, sugere-se o uso de métodos de otimização multiobjetivos (ou multicritérios), onde a confiabilidade (ou disponibilidade) seria otimizada juntamente com o custo.



## Referências Bibliográficas

Arora, J. S. *Introduction to Optimal Design*. New York: McGraw-Hill, 1989, 625p.

Ashby, W.R. *Design for a brain*. New York: John Wiley & Sons. Inc, 1960.

Banerjee, S. K., Rajamani, K. Optimization of System Using a Parametric Approach. *IEEE Transactions on Reliability*, v. R-22, no. 1, pp. 35-39, April 1973.

Barlow, R. E., Proschan, F. *Mathematical Theory of Reliability*. New York: John Wiley & Sons. Inc, 1965

Bentley, P. J. *Evolutionary Design by Computers*. San Francisco: Morgan Kaufman Publishers, Inc., 1999, 446p.

Cavalca, K. L., Mojena, M. A. R. *Analysis and reduction of structures in reliability*. In: INTERPACK 99 – ASME International, Intersociety Electronic & Photonic Packaging Conference, 1999, Hawaii – USA. EEP –Advances in Electronic Packaging. Hawaii – USA: ASME 1999, v. 26, pp. 1033-1040.

Cavalca, K. L., Mojena, M. A. R. *Avaliação da confiabilidade em sistemas pelo método da matriz de conexão* In: COBEM 99 – Congresso Brasileiro de engenharia Mecânica, Águas de Lindóia – SP: UNICAMP, 1999, v.1, p.1-10.

Davis, L. *Handbook of Genetic Algorithm*. London: IPT, 1996, 385p.

Doty, L. A. *Reliability for the technologies*. New York: ASQC Quality press Book. 1989, 307p.

Ertas A, *The Engineering Design Process*, John Wiley & Sons Inc, 1993, 525p.

Fogel, L.J., Owens, A. J., Walsh, M. J., *Artificial Intelligence through simulated evolution*. New York: John Wiley & Sons. Inc, 1966.

Goldberg, D. E. *Genetic Algorithm in search optimization and Machine Learning*. Reading, Mass.: Addison-Wesley, 1989, 412p.

Hikita, M., Nakagawa, Y., Nakashima, K. Marihisa, H. Reliability Optimization of Systems by a Surrogate Constrains Algorithm. *IEEE Transactions on Reliability*, v. 41, n. 3, pp.473-480, September 1992.

Holland, J. H. *Adaptation in natural and artificial system: an introduction with application to biology, control and artificial intelligence*, An Arbor, MI: University of Michigan Press, 1975.

Holland, J. H. *Adaptation in natural and artificial system: an introduction with application to biology, control and artificial intelligence*, 2<sup>nd</sup>. Ed. Cambridge, MA: MIT Press, 1992, 211p.

Hsieh, Y. C., Chen, T. C., Bricker, D. L. Genetic Algorithms for Reliability Design Problems. *Microelectronics Reliability*, v. 38, n. 10, pp. 1599-1605, 1998.

- Kumar, A., Pathak, R. M., Gupta, Y. P. Genetic-Algorithm-Based Reliability Optimization for Computer Network Expansion. *IEEE Transactions on Reliability*, v. 44, n. 1, pp. 63-72, March 1995.
- Kuo, W., Hwang C., Tillman, F. A. A Note on Heuristic Methods in Optimal System Reliability. *IEEE Transaction on Reliability*, v. R-27, n. 5, pp. 320-324, December 1978.
- Kuo, W., Prasad, V. R. An Annotated Overview of System Reliability Optimization. *IEEE Transactions on Reliability*, v. 49, n. 2, pp.176-187, June 2000.
- Levitin, G. Redundancy Optimization for Multi-state System with Fixed Resource-Requirements and Unreliable Sources. *IEEE Transactions on Reliability*, v. 50, n. 1, pp. 52-59, March 2001.
- Lewis, E. E. *Introduction to Reliability engineering*. John Wiley & Sons, Inc. 1996, 435p.
- Lopes, José Carlos Bueno de Paiva *Otimização de Sistemas Redundantes em Confiabilidade* Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1998, 91p., Trabalho de Conclusão (Graduação).
- Michalewicz, Z. *Genetic Algorithms + data structures = evolution programs*, 3<sup>rd</sup>. Ed. New York: Springer, 1996, 387p.
- Mitchell, M. *An Introduction to Genetic Algorithms*. Cambridge: MIT, 1996, 209p.
- Mohan, C., Shanker, K. Reliability Optimization of Complex System Using Random Search Technique. *Microelectronics and Reliability*, v. 28, n. 4, pp. 513-518, 1988.

- Mojena, Miguel Algel Reyes *Análise e Redução de Estruturas em Confiabilidade* Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1999, 158p., Dissertação (Mestrado).
- Moras, Celso Fabrício *Metodologia para implementação do modelo Markoviano em análise de confiabilidade de sistemas*. Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2002, 116p., Dissertação (Mestrado).
- Narasimhalu, A. D., Sivaramakrishnan, H. A Rapid Algorithm For Reliability Optimization of Parallel Redundant System. *IEEE Transactions on Reliability*, v. R-27, n. 4, pp. 261-263, October 1978.
- Painton, L., Campbell, J. Genetic Algorithm in Optimization of System Reliability. *IEEE Transactions on Reliability*, v. 44, n. 2, pp. 172-178, June 1995
- Ramakumar, R., *Engineering Reliability*, Ed. Prentice Hall, 1993, 482p.
- Ravi, V., Murty, B. S. N., Reddy, P. J. Nonequilibrium Simulated Annealing-Algorithm Applied to Reliability Optimization of Complex Systems. *IEEE Transaction on Reliability*, v. 46, n. 2, pp. 233-239, June 1997.
- Rechenberg, I. *Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog, 1973.
- Rubinstein, R. Y., Levitin, G., Lisnianski, A., Ben-Haim, H. Redundancy Optimization of Static Series-Parallel Reliability Models Under Uncertainty. *IEEE Transactions on Reliability*, v. 46, n. 4, pp. 503-511, December 1997.

Sasaki, M., Kaburaki, S., Yanagi, S. System Availability and Optimum Spare Units. *IEEE Transactions on Reliability*, v. R-26, n. 3, pp. 182-188, August 1977.

Schwefel, H. P. *Evolution and optimum Seeking*. New York: J. Wiley, 1995, 444p.

Siddal, J. N. *Optimal Engineering Design: Principles and applications*. New York: Marcel Decker, Inc., 1982, 523p.

Tillman, F. A., Hwang C. L., Kuo, W. Optimization Techniques for System Reliability with Redundancy – A Review. *IEEE Transactions on Reliability*, v. R-26, n. 3, pp. 148-155, August 1977.

Wohl, J. G. System Operational Readiness and Equipment Dependability. *IEEE Transactions on Reliability*, v. R-15, n. 1, pp. 1-6, May 1966.