

**DISSERTAÇÃO DE MESTRADO PROFISSIONALIZANTE EM  
ADMINISTRAÇÃO**

**ANÁLISE POR ÁRVORE DE FALHAS  
USANDO ESTIMADORES DE  
PLAUSIBILIDADE MÁXIMA E LÓGICA  
FUZZY**

**HIME AGUIAR E OLIVEIRA JUNIOR**

Orientadora: Profa. Dra. Maria Augusta Soares Machado

Rio de Janeiro, 16 de Fevereiro de 2006

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

**ANÁLISE POR ÁRVORE DE FALHAS  
USANDO ESTIMADORES DE  
PLAUSIBILIDADE MÁXIMA E LÓGICA  
FUZZY**

Dissertação apresentada ao Curso de Mestrado  
Profissionalizante em Administração das Faculdades  
Ibmec como parte dos requisitos para obtenção do  
título de Mestre em Administração  
**Área de concentração: Apoio à decisão**

**HIME AGUIAR E OLIVEIRA JUNIOR**

Orientadora: Profa. Dra. Maria Augusta Soares Machado

Rio de Janeiro, 16 de Fevereiro de 2006



# **FACULDADES IBMEC**

**PROGRAMA DE PÓS-GRADUAÇÃO E PESQUISA EM  
ADMINISTRAÇÃO E ECONOMIA**

**HIME AGUIAR E OLIVEIRA JUNIOR**

**ANÁLISE POR ÁRVORE DE FALHAS USANDO ESTIMADORES DE  
PLAUSIBILIDADE MÁXIMA E LÓGICA FUZZY**

Dissertação apresentada ao Curso de Mestrado Profissionalizante em Administração das Faculdades Ibmec como parte dos requisitos para obtenção do título de Mestre em Administração.  
**Área de concentração: Apoio à decisão**

Aprovada em:

**BANCA EXAMINADORA:**

---

Profa. Dra. Maria Augusta Soares Machado

---

Prof. Dr. Edson José Dalto

---

Prof. Dr. Reinaldo Castro de Souza

Rio de Janeiro, 16 de Fevereiro de 2006

## ANÁLISE POR ÁRVORE DE FALHAS USANDO ESTIMADORES DE PLAUSIBILIDADE MÁXIMA E LÓGICA *FUZZY*

### RESUMO

O presente trabalho visa propor modelo de análise por árvore de falhas utilizando abordagens probabilística e possibilística e apresentar respectivos resultados obtidos através de implementação prototípica. Como aspecto inovador na vertente probabilística, são apresentados procedimentos de estimação de parâmetros das funções de distribuição por meio do Método da Plausibilidade Máxima (*Maximum Likelihood Method*) em conjugação com a técnica estocástica de otimização global conhecida como *Fuzzy ASA (Adaptive Simulated Annealing)*. O caso possibilístico lança mão de resultados fundamentais da Aritmética *Fuzzy* para propagação das possibilidades de falha no sentido ascendente.

Palavras-chave: Lógica *Fuzzy*; Inferência Estatística; Otimização Global.

# **FAULT TREE ANALYSIS USING MAXIMUM LIKELIHOOD ESTIMATORS AND FUZZY LOGIC**

## **ABSTRACT**

The present work aims to propose a fault tree analysis model using probabilistic and possibilistic approaches and to show results from a corresponding prototypical implementation. As an innovative feature in the probabilistic viewpoint, we present statistical estimation procedures based on the Maximum Likelihood Method conjugated to the stochastic global optimization technique known as Fuzzy ASA (Adaptive Simulated Annealing). The possibilistic case uses fundamental results of Fuzzy Arithmetic to propagate fault possibilities in a bottom-up manner.

Key-words: Fuzzy Logic; Statistical Inference; Global Optimization .

## ÌNDICE

<b>1. DEFINIÇÃO DO PROBLEMA.....</b>	<b>6</b>
<b>2. OBJETIVOS DA PESQUISA.....</b>	<b>8</b>
<b>3. ABORDAGEM PROPOSTA.....</b>	<b>10</b>
<b>4. METODOLOGIA DETALHADA.....</b>	<b>11</b>
4.1. INTRODUÇÃO .....	11
4.2. LÓGICA <i>FUZZY</i> .....	12
4.3. <i>FUZZY ADAPTIVE SIMULATED ANNEALING (FUZZY ASA)</i> .....	25
4.4. ESTIMADORES COM PLAUSIBILIDADE MÁXIMA USANDO <i>FUZZY ASA</i> .....	38
4.5. ÁRVORES DE FALHAS POSSIBILÍSTICAS E PROBABILÍSTICAS .....	47
4.6. INTEGRAÇÃO DE RESULTADOS INTERMEDIÁRIOS .....	57
4.7. ARQUITETURA DE SOFTWARE UTILIZADA NA IMPLEMENTAÇÃO .....	60
<b>5. RESULTADOS .....</b>	<b>61</b>
<b>6. CONCLUSÃO.....</b>	<b>66</b>
<b>7. REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>67</b>

## 1. DEFINIÇÃO DO PROBLEMA

Em qualquer ambiente – industrial ou comercial - onde a operação ininterrupta seja uma necessidade ou períodos de inatividade sejam altamente nocivos às atividades-fim da Organização, ferramentas que forneçam subsídios para construir estimativas e/ou alertas relativos a potenciais eventos de falha certamente serão bem-vindos, tendo em vista o impacto positivo de sua utilização no aumento da eficiência operacional, redução de estoques com peças de reposição, ampliação da capacidade de planejamento, redução de custos com serviços de manutenção emergenciais, etc. Sendo assim, o problema a resolver (construção de um tal tipo de ferramenta) pode ser traduzido em algumas poucas perguntas objetivas:

- ♦ Que tipo de abordagem adotar?
- ♦ Qual é o modelo mais adequado (eficaz e eficiente) para resolver o problema em questão?
- ♦ Como implementar a solução?
- ♦ A relação custo/benefício é satisfatória?

Foram escolhidas as seguintes características para a referida ferramenta de software:

- ♦ Abordagens probabilística e possibilística.
- ♦ Modelo hierárquico em árvore, tendo como folhas os eventos considerados elementares.



- Implementação por meio de linguagem de alto nível (C++) compilável em código de máquina.
- A relação custo/benefício será satisfatória se forem utilizadas ferramentas adequadas e de baixo custo, como diversas bibliotecas e compiladores *open source* (GNU GPL - *General Public License*, por exemplo) de boa qualidade. Tais ferramentas tipicamente apresentam custos operacionais residuais e investimento inicial nulo. Para a realização do programa descrito neste trabalho serão utilizados o compilador C++ Borland BC 5. 5, a biblioteca de classes para construção de interfaces gráficas FOX-TOOLKIT ([www.fox-toolkit.org](http://www.fox-toolkit.org)), a implementação do algoritmo de otimização global estocástica (*FUZZY*) ASA ([www.ingber.com](http://www.ingber.com) ) e rotinas numéricas para cálculos probabilísticos elementares (PROB - [http://www.csit.fsu.edu/~burkardt/cpp\\_src/prob/prob.html](http://www.csit.fsu.edu/~burkardt/cpp_src/prob/prob.html)).

## 2. OBJETIVOS DA PESQUISA

Projetar e implementar programa passível de execução em ambientes computacionais de larga utilização e baixo custo, com funcionalidades relativas à realização de análise por árvore de falhas para sistemas genéricos. Como aspecto inovador, a estimação de parâmetros das funções de distribuição envolvidas será realizada por meio do Método da Plausibilidade Máxima (*Maximum Likelihood Method*) em conjugação com a técnica estocástica de otimização global conhecida como *Fuzzy ASA (Adaptive Simulated Annealing)*.

Logo, um dos resultados da pesquisa deverá ser a validação de um novo método de estimação estatística que evitará os frequentes problemas decorrentes do “aprisionamento” em mínimos locais, quando da execução do processo de otimização da função de plausibilidade, característicos de esquemas baseados no conceito de gradiente de funções diferenciáveis. Outro aspecto a ser ressaltado é a eliminação da exigência de diferenciabilidade (ou até continuidade) para as funções sob processo de maximização, tendo em vista a generalidade adicional dos métodos numéricos utilizados. Como outro subproduto, podemos citar a possibilidade da obtenção simultânea de todos os parâmetros das distribuições, eliminando as dificuldades relacionadas à solução de equações não lineares decorrentes da anulação dos gradientes, quando as mesmas possuem solução conhecida em forma fechada. Finalmente, o produto deverá ser suficientemente versátil para possibilitar a substituição completa de distribuições de probabilidade com mínimo impacto estrutural no sistema como um todo.

Na vertente possibilística, as operações determinadas pela estrutura da árvore de falhas, que explicita a dependência operacional entre os vários subsistemas, serão realizadas por meio dos operadores lógicos fundamentais da Lógica *Fuzzy* (OR, AND, etc.). No nível mais baixo da

hierarquia arbórea (as folhas), as possibilidades de falha deverão ser arbitradas por números *fuzzy*, representando as opiniões de especialistas em cada uma das áreas envolvidas. Ao contrário do caso probabilístico, estes *scores* serão assumidos como verdadeiros, tendo em vista a ausência (ou insuficiência) de dados históricos, indispensáveis para os procedimentos usuais de inferência estatística. Em suma, o objetivo é fornecer ferramentas quantitativas para avaliação de riscos em qualquer caso: com ou sem volume significativo de informações históricas. Para efeito de delimitação de escopo, foi determinado que a implementação utilizará somente conectivos lógicos OR e AND, ficando para etapas posteriores a inclusão de alguns outros que venham a ser considerados convenientes. Tal escolha não afeta o alcance da ferramenta, tendo em vista que as operações escolhidas representam conexões em série e paralelo de subsistemas. De qualquer forma, a arquitetura de software idealizada acomodará novos operadores estruturais com mínimo esforço de programação.

### 3. ABORDAGEM PROPOSTA

Integrar resultados fundamentais da Lógica *Fuzzy* e Estatística para síntese de modelos funcionais de árvores de falhas possibilísticas e probabilísticas, utilizando inferência por Plausibilidade Máxima (*Maximum Likelihood*), otimização por meio de *Simulated Annealing*, operações *fuzzy* elementares, resultados básicos de Cálculo de Probabilidades, etc. Cada uma das abordagens dará origem a modelo específico, ou seja, serão sintetizados verdadeiramente dois modelos distintos, embora mantenham similaridade estrutural superficial.

## 4. METODOLOGIA DETALHADA

### 4.1. INTRODUÇÃO

Para realizar a tarefa proposta serão utilizadas técnicas da Teoria das Probabilidades, assim como estruturas computacionais que permitirão a modelagem das relações de dependência entre os diversos subsistemas envolvidos. Assim, serão construídas árvores de falha invertidas cuja raiz simbolizará o macrossistema sob estudo e os nós intermediários corresponderão aos vários componentes de cujo funcionamento depende o nó imediatamente superior. As “folhas” da árvore corresponderão aos elementos funcionais considerados atômicos ou elementares, cujo comportamento operacional definirá a confiabilidade global.

Para permitir o acompanhamento temporal das probabilidades de falha, cada tipo de componente-folha será associado a conjuntos de registros de falhas anteriores, que integrarão amostras consideradas independentes e identicamente distribuídas. A partir da definição do tipo de distribuição correspondente aos eventos de falha do elemento em questão e de seu conjunto amostral, será possível aplicar o Método da Plausibilidade Máxima para obtenção dos parâmetros que aproximarão os da distribuição real. Uma vez obtidos os parâmetros de todo os componentes será possível computar as probabilidades de falha dos eventos compostos, incluindo o correspondente à raiz. Para realizar os procedimentos de maximização da função de plausibilidade, será usado o método de otimização global estocástica conhecido como *Fuzzy ASA*, aplicado ao inverso multiplicativo do logaritmo natural da referida função. Ao minimizar o inverso do logaritmo, será maximizada a função original, que é o efeito desejado. Ao final, serão construídos módulos de software, em linguagem C++, a título de ilustração das idéias aqui apresentadas.

## 4.2. LÓGICA FUZZY

Embora seja sempre difícil definir conceitos fundamentais, podemos estabelecer que Lógica Fuzzy é um conjunto de métodos baseados no conceito de conjunto *fuzzy* e operações *fuzzy*, possibilitando a modelagem realista e flexível de sistemas. O aspecto mais notável dessa metodologia é a possibilidade de se capturar, em um modelo matemático, conceitos intuitivos como graus de satisfação, conforto, adequação, etc. Nada melhor do que um exemplo para ilustrar a idéia [Oliveira Jr., 1999]:

Suponhamos que em determinada cidade, a temperatura varie anualmente entre 15 e 42 graus Celsius. Ao questionarmos um grupo de habitantes sobre sua sensação térmica, escolhida entre as alternativas {FRIA, NEUTRA, QUENTE}, certamente todos concordarão que a de 15 graus é FRIA, 42 graus é QUENTE e a, digamos, 25 graus é NEUTRA.

A pergunta crucial é:

Em que temperaturas a sensação FRIA se torna NEUTRA e esta se transforma em QUENTE?

Difícilmente haverá coincidência de opiniões, tendo em vista que os conceitos envolvidos são subjetivos e apresentam áreas "sombrias", nas quais não se consegue escolher entre duas alternativas mutuamente exclusivas simplesmente porque ambas estão parcialmente presentes [Oliveira Jr., 1999].

É importante ressaltar que, apesar do nome, não se trata de uma vertente da Lógica Matemática, nem uma nova Teoria de Conjuntos e muito menos um paradigma conflitante com a Teoria das Probabilidades.

Embora empregue os conceitos de inferência *fuzzy*, conjunto *fuzzy* e distribuição de possibilidades, a teoria é formulada por meio de conceitos da Teoria dos Conjuntos de Cantor e se mostra como importante ferramenta também no tratamento de sistemas estocásticos, aliada ao conhecimento já estabelecido nos modelos estatísticos. Devemos entender com clareza que o tipo de incerteza tratado pela Estatística difere daquele tratado pela Lógica *Fuzzy*.

Apesar do elevado percentual de exemplos relacionados à área de Engenharia, nunca é demais reafirmar que o conceito de difusão (*fuzziness*) tem larga utilidade nas ciências sociais e médicas, onde a interação com conceitos até então não "matematizáveis" é grande. A relação a seguir mostra alguns segmentos já contemplados com soluções *fuzzy*:

- ♦ Controle econômico de sistemas de condicionamento de temperatura
  
- ♦ Controle de máquinas de lavar
  
- ♦ Controle de secadoras de roupa
  
- ♦ Controle de freios ABS
  
- ♦ Injeção eletrônica de combustível em veículos automotores
  
- ♦ Sistemas de transmissão automática em veículos
  
- ♦ Sistemas de tratamento de água
  
- ♦ Direção automática de trens

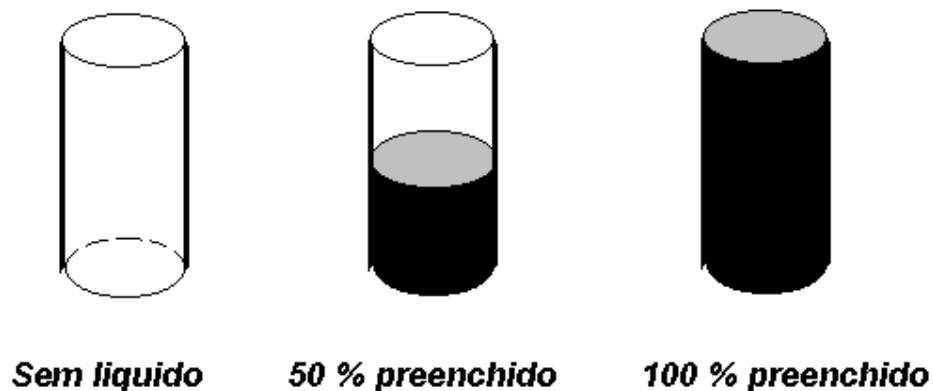
- ♦ Controle de concentração de boro em reatores nucleares (PWR)
  
- ♦ Sistemas de reconhecimento de voz e imagem
  
- ♦ Controle de tráfego urbano e rodoviário
  
- ♦ Detecção de fraudes em cartões de crédito
  
- ♦ Reconhecimento de assinaturas
  
- ♦ Diagnóstico de câncer por ultra-sonografia
  
- ♦ Carga rápida de baterias
  
- ♦ Estabilização de imagens em filmadoras
  
- ♦ Controle integrado de grupos de elevadores
  
- ♦ Controle de anestesia
  
- ♦ Controle de ventilação em túneis

Vale dizer que a grande maioria das aplicações de mercado foi desenvolvida em países da Ásia e Europa, em particular, Japão e Alemanha, onde o caminho *fuzzy* foi escolhido como o mais adequado para a realização de metas aparentemente conflitantes, como redução de custos com aumento de qualidade.



O aspecto principal desta abordagem é, como foi dito, a capacidade de capturar com clareza e concisão as várias nuances dos conceitos psicológicos utilizados pelos seres humanos em seu raciocínio usual, sem a necessidade de "violentá-lo" por meio de enquadramento obrigatório em modelos por vezes incompatíveis com o grau de difusão apresentado no contexto em questão. Como exemplo, podemos citar uma situação banal, mas ilustrativa:

Se formos solicitados a classificar o status dos 3 copos idênticos da figura I. 1, quanto a seu grau de preenchimento, e só utilizarmos os predicados CHEIO e VAZIO, como expressar a condição do copo central?



O copo está meio CHEIO ou meio VAZIO?

Na verdade, o copo está meio CHEIO **E** meio VAZIO. O problema é que os paradigmas tradicionais de classificação costumam exigir a tomada de decisões abruptas ("CRISP", na terminologia difusa), descartando os tons de "cinza" entre o absolutamente "preto" e o absolutamente "branco", além de proibir a coexistência de conceitos qualitativamente antagônicos.

O analista *fuzzy* responderia:

1. O copo central está CHEIO com grau de 50 % e VAZIO com grau de 50 %.
2. O copo da esquerda está CHEIO com grau de 0 % e VAZIO com grau de 100 %.
3. O copo da direita está CHEIO com grau de 100 % e VAZIO com grau de 0 %.

Outra grande vantagem apresentada se relaciona com a análise de sistemas de extrema complexidade, cuja modelagem por ferramentas convencionais se mostra extremamente difícil, ou até impossível.

Mis uma vez, exemplifiquemos [Oliveira Jr., 1999]:

Suponhamos que o administrador de um fundo mútuo de investimento em ações deseje construir um programa de apoio à decisão, que tenha como entradas os diversos "sinais vitais" da Economia (cotações de diversos ativos e suas tendências), Política (decisões governamentais) e séries temporais "intraday" do mercado acionário em si (atualização em tempo real).

As saídas desejadas são:

1. Emissão automática de alertas sobre eventos significativos
2. Emissão de séries temporais , contendo alterações em sua
3. carteira atual, de modo a maximizar o volume total de recursos, incluindo a possibilidade de negociação *eletrônica*, ou seja, sem intervenção humana.

## REQUISITOS FUNDAMENTAIS

1. SEGURANÇA TOTAL QUANTO A FALHAS DE JULGAMENTO
2. VISÃO CONSERVADORA
3. AGRESSIVIDADE (não se limita a ações de primeira linha)
4. COMPORTAMENTO ADAPTATIVO

É certo que um bom analista de mercado de capitais tem um modelo mental de como tal sistema deveria atuar. O problema é que tal modelo está repleto de subjetividade e nuances inerentes ao modo pelo qual pensamos.

Outro fato crucial se apresenta nas milhares de variáveis fortemente acopladas que traduzem a dinâmica não linear dos mercados, impossibilitando, em termos práticos, a solução ótima do problema por meios manuais e/ou tradicionais. É nesse momento que a abordagem difusa se mostra como alternativa viável de modelagem e até de implementação (por meio de linguagens *fuzzy* declarativas, por exemplo). Toda a experiência do especialista pode (e deve) ser capturada, transformando subjetividade em *graus de pertinência*, raciocínio em *base de regras*, tomada de decisão em *inferência / "defuzzificação"*.

A formalização dos conceitos difusos por Zadeh permitiu, entre outras coisas, o preenchimento de uma lacuna existente desde tempos imemoriais, ou seja, a descrição sistemática e integral da experiência acumulada em termos precisos e a solução de paradoxos seculares. No entanto, pode-se atribuir ao grande Bertrand Russel o início da "era *fuzzy oficial*", quando da publicação do

artigo "Vagueness" em 1923, após se deparar com alguns paradoxos da Lógica Matemática clássica.

Existe bastante confusão sobre o significado do conceito de difusão (*fuzziness*) e escopo de sua aplicação no tratamento da imprecisão.

Imprecisão (ou incerteza) associada com a ocorrência futura de algum evento está ligada ao tratamento probabilístico. Imprecisão associada à descrição de sistemas leva naturalmente ao tratamento *fuzzy*, ou seja, o conhecimento que possuímos a respeito da estrutura interna do objeto em estudo não é suficiente para formularmos equações (convencionais) precisas, por exemplo.

Existe uma íntima relação entre ambigüidade e difusão, pois a primeira é a propriedade de se possuir simultaneamente vários significados plausíveis ou possíveis e estes, normalmente, possuem diferentes pesos ou ponderações, dependendo de nossa interpretação dentro do contexto em que ocorrem.

Vamos a um exemplo "ecológico" de ambigüidade sem difusão:

Um fumante pergunta a outro:

- "Quer um cigarro?"

A resposta é:

- "Obrigado!"

O primeiro certamente dirá:

- "Obrigado SIM ou obrigado NÃO?"

Vemos que há DUAS possibilidades BEM delimitadas, pois o interlocutor QUER ou NÃO QUER o cigarro, não havendo margem de imprecisão, uma vez que ocorreu apenas um deslize na comunicação.

Já a seguinte frase, pode dar origem a uma situação *fuzzy*, mesmo após a remoção da ambigüidade:

- "Esse prato é quente !"

Há 3 possibilidades:

- ♦ O prato está na moda
- ♦ O prato está apimentado
- ♦ O prato está com temperatura elevada

Nos dois primeiros casos, a incerteza é removida após o esclarecimento.

No terceiro caso, o conceito de QUENTE dá oportunidade para o tratamento *fuzzy*, tendo em vista que as pessoas divergem bastante em sua conceituação de sensações térmicas.

Ao contrário de muitas áreas da Ciência contemporânea, o estudo da Lógica *Fuzzy* não exige grande quantidade de conhecimentos matemáticos. Pode-se observar que os principais pré-requisitos para entender e aplicar os conceitos fundamentais, em nível básico, são maturidade

intelectual e alguns elementos básicos de Teoria dos Conjuntos. Esta constatação reforça um aspecto importante da Teoria *Fuzzy*, visto que democratiza sua aplicação em terrenos não "matematizados". Por outro lado, quando aplicamos a teoria em problemas de Engenharia, Computação e áreas afins, tem se mostrado importante o domínio total das ferramentas técnicas convencionais até então utilizadas no segmento em questão, mesmo que seja no único intuito de comparar os diferentes resultados. Podemos citar um exemplo bastante simples, ilustrando o que foi dito acima sobre a relativa simplicidade das soluções *fuzzy* alternativas:

Imaginemos uma situação em que um operador de câmera realiza uma filmagem de um jogo de futebol.

Tendo que acompanhar o movimento da bola durante o andamento do jogo, ele "fecha a malha" e implementa um sistema de controle manual *fuzzy*, centrando a bola na tela intuitivamente, por meio da previsão de suas posições futuras. Com a abordagem fuzzy, o que fazemos é capturar, implementar e ajustar o algoritmo de controle, sem desprezar qualquer experiência anteriormente adquirida, incluindo modelos matemáticos preestabelecidos. Se analisarmos a linha de raciocínio utilizada pelo operador anteriormente citado, veremos que seu modelo mental de rastreamento dificilmente será baseado em inferências estatísticas ou filtros de Kalman. Mas, apesar de tudo, é recomendável a utilização de quaisquer subsídios para o enriquecimento e/ou avaliação do método *fuzzy*. [Oliveira Jr., 1999].

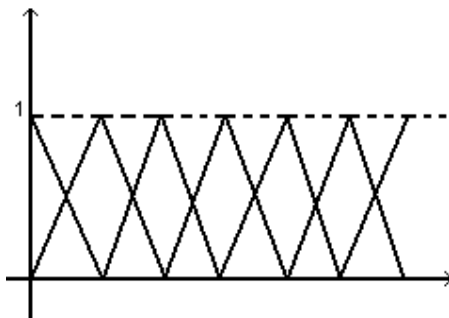
Formalmente, um conjunto *fuzzy*  $F$  é definido como o conjunto de pares ordenados contendo o elemento e seu grau de pertinência no conjunto:

$$F = \{ (u, \mu_F(u)) / u \in U \}$$

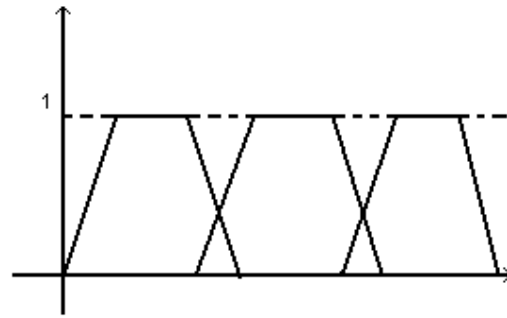
Onde  $U$  é o chamado universo de discurso e  $\mu_F(u)$  a função de pertinência associada ao conjunto  $F$ , representada por:

$$\mu_F(u): U \rightarrow [0, 1]$$

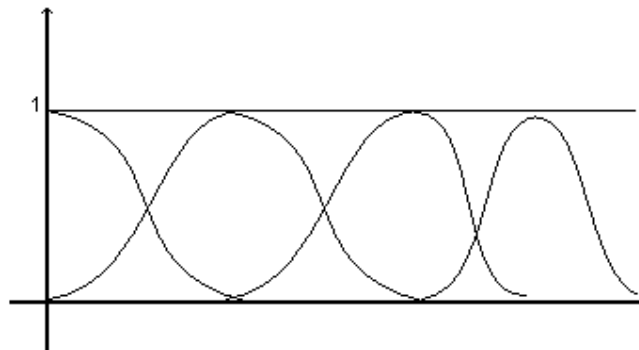
As funções de pertinência fazem o papel das curvas de possibilidades na teoria clássica da Lógica *Fuzzy*. Sendo os conjuntos *fuzzy* apropriados para representar noções vagas, freqüentemente encontradas no mundo real, como, por exemplo, alto, quente, frio, rápido, etc., é a função de pertinência que define as fronteiras desses conjuntos. A princípio, qualquer função com domínio  $U$  e imagem em  $[0, 1]$  pode ser utilizada como função de pertinência. No entanto, as formas triangular, trapezoidal e gaussiana são as mais utilizadas.



Triangulares



Trapezoidais



Gaussianas

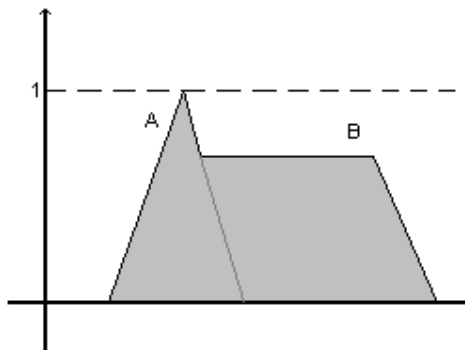
Zadeh apresenta o conceito de conjunto *fuzzy* como uma generalização da noção de conjunto clássico, sendo, num certo sentido, uma extensão da Teoria dos Conjuntos. Abaixo estão definidas as principais relações e operações entre conjuntos *fuzzy*:

## UNIÃO

A função de pertinência resultante da união de dois conjuntos *fuzzy* é o maior valor da pertinência aos dois conjuntos específicos e de cada um dos seus elementos. Em outras palavras, o que se quer para o grau de associação de um elemento, quando listado, na união de dois grupos *fuzzy*, é o valor máximo deste grau de associação quando dois grupos *fuzzy* formam uma união. Dados dois conjuntos *fuzzy* A e B, denomina-se conjunto *fuzzy* união de A com B, ao conjunto *fuzzy* C definido como  $C = A \cup B$ , cuja função de pertinência é relacionada às funções de pertinência de A e B por:

$$\mu_C(x) = \max[\mu_A(x); \mu_B(x)]$$

A união de dois conjuntos *fuzzy* corresponde, na Álgebra Booleana, à operação OR.



União entre conjuntos *Fuzzy*

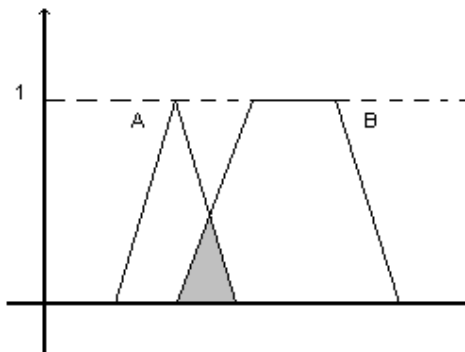


## INTERSEÇÃO

A interseção de dois conjuntos *fuzzy* caracteriza-se por ser a parte comum de ambos os conjuntos *fuzzy* e, em consequência, é sempre menor que qualquer um dos conjuntos individuais. A função de pertinência resultante da interseção de dois conjuntos *fuzzy* é o menor valor da pertinência, aos dois conjuntos específicos e de cada um dos seus elementos. O grau de associação de um elemento na interseção de dois grupos *fuzzy* é o mínimo, ou o menor valor de seu grau de associação individualmente nos dois grupos que formam a interseção. Dados dois conjuntos *fuzzy* A e B, denomina-se conjunto *fuzzy* interseção de A com B, ao conjunto *fuzzy* C definido como  $C = A \cap B$ , cuja função de pertinência é relacionada às funções de pertinência de A e B por:

$$\mu_C(x) = \min[\mu_A(x) ; \mu_B(x)]$$

A interseção de dois conjuntos *fuzzy* corresponde, na Álgebra Booleana, à operação AND.



Interseção entre conjuntos *Fuzzy*

## COMPLEMENTO

Dado o conjunto *fuzzy*  $A$ , denomina-se complemento de  $A$ , o conjunto *fuzzy*  $C$  representado por  $\bar{A}$  e satisfazendo:

$$\mu_C(x) = 1 - \mu_A(x)$$

O complemento de um conjunto *fuzzy* corresponde, na Álgebra Booleana, à operação NOT.

## DIFERENÇA

Dados dois conjuntos *fuzzy*  $A$  e  $B$  sobre um universo de discurso  $X$ , com graus de pertinência de  $x$  iguais a  $\mu_A(x)$  e  $\mu_B(x)$  em  $A$  e  $B$ , respectivamente, dizemos que  $A \neq B$  se  $\mu_A(x) \neq \mu_B(x)$  para, pelo menos, um elemento de  $x \in X$ .

## IGUALDADE

Dados dois conjuntos *fuzzy*  $A$  e  $B$  sobre um universos de discurso  $X$ , com graus de pertinência de  $x$  iguais a  $\mu_A(x)$  e  $\mu_B(x)$  em  $A$  e  $B$ , respectivamente, dizemos que  $A = B$ , se  $\mu_A(x) = \mu_B(x)$  para todo  $x \in X$ .

## INCLUSÃO

Dados dois conjuntos *fuzzy*  $A$  e  $B$  sobre um universo de discurso  $X$ , com graus de pertinência de  $x$  iguais a  $\mu_A(x)$  e  $\mu_B(x)$ , respectivamente, dizemos que  $A$  está contido em  $B$ , e representamos por  $A \leq B$ , se  $\mu_A(x) \leq \mu_B(x)$  para todo  $x \in X$ .

## SUPORTE

O suporte de um conjunto *fuzzy*  $A$ , em um conjunto universo  $X$ , é o conjunto *crisp* que contém todos os elementos de  $X$  que possuem grau de pertinência diferente de zero em  $A$ .

$$\text{Suporte}(A) = \{ x / \mu_A(x) \geq 0 \}$$

## NÚCLEO

O núcleo de um conjunto *fuzzy*  $A$  é o conjunto *crisp* de todos os pontos  $x \in X$  tais que  $\mu_A(x) = 1$ :

$$\text{Núcleo}(A) = \{ x / \mu_A(x) = 1 \}$$

## NORMAL

Um conjunto *fuzzy*  $A$  é dito normal se o seu núcleo não é vazio, ou seja, se há pelo menos um ponto  $x \in X$  tal que  $\mu_A(x) = 1$

### 4.3. FUZZY ADAPTIVE SIMULATED ANNEALING (FUZZY ASA)

O problema de otimização global de funções numéricas tem grande importância em várias áreas do Conhecimento. Ele aparece em campos como Engenharia, Finanças, Gerenciamento, Medicina, etc. Em casos práticos, a função a ser minimizada aparece como um índice de performance multivariável e está sujeita a certos vínculos, impostos por seu contexto. Quando a função-objetivo é “bem comportada”, existem vários métodos para se obter seus pontos de mínimo, satisfazendo aos referidos vínculos.

Os problemas começam quando a dada função apresenta vários mínimos locais, cada um deles possuindo sua própria bacia de atração, fazendo tipicamente o resultado final depender do ponto inicial, utilizado para “lançar” um dado algoritmo particular. Infelizmente, a maioria dos problemas práticos dá origem a funções-objetivo bastante complexas, sendo frequentemente não-lineares, descontínuas, multi-modais, etc. .

Para resolver tal classe de problemas, métodos não-determinísticos parecem ser uma boa alternativa, se não a única, em várias situações. Algoritmos genéticos e *Simulated Annealing* (abreviadamente SA) aparecem como abordagens bastante exploradas em otimização estocástica global. O problema neste caso está relacionado à velocidade de convergência e, no caso genético, garantia do alcance do extremo global, sob condições suficientemente gerais. Métodos puros de SA, por outro lado, possuem resultados assegurando sua convergência a ótimos globais com probabilidade 1, mas seu desempenho em geral não é satisfatório. Apesar disso, alguns pesquisadores conseguiram superar tais limitações, levando, por exemplo, a alternativas como VFSR (*Very Fast Simulated Reannealing*), que é um método de otimização global estocástica realmente eficaz. VFSR é particularmente adequado a aplicações envolvendo sistemas *neuro-fuzzy* e treinamento de redes neurais artificiais, em virtude de seu excelente desempenho e relativa simplicidade. ASA (*Adaptive Simulated Annealing*) é uma implementação do método VFSR (na linguagem de programação C) que nos traz os benefícios de ser pública, parametrizável e bem mantida. Além disso, ASA se mostra como uma alternativa em relação aos algoritmos genéticos, levando-se em conta os testes de desempenho já realizados, que demonstram sua boa qualidade. Infelizmente, todos os algoritmos estocásticos de otimização global compartilham algumas características pouco desejáveis como, por exemplo, longos períodos de estagnação. Em implementações de SA, isto é devido ao chamado processo de *resfriamento*, cuja velocidade é limitada pelas características das funções de densidade de probabilidade (PDFs) usadas para a geração de novos pontos candidatos. Desta maneira, se escolhermos empregar o resfriamento de

Boltzmann (BA), a “temperatura” é reduzida a uma taxa máxima correspondente a  $T(k) = T(0) / \ln(k)$ . No caso do resfriamento rápido (FA), a evolução da temperatura passa a  $T(k) = T(0) / k$ , se desejarmos assegurar convergência com probabilidade 1, resultando numa evolução mais rápida do processo geral. ASA tem um esquema ainda mais eficiente, dado por:

$$\mathbf{T}_i(\mathbf{k}) = \mathbf{T}_i(\mathbf{0}) \times \exp(-\mathbf{C}_i \mathbf{k}^{\frac{1}{D}})$$

(  $\mathbf{C}_i$  = parâmetro definido pelo usuário )

graças à sua excelente PDF de geração de pontos. Note que os subscritos indicam evolução independente de temperaturas para cada dimensão. O usuário também tem a possibilidade de utilizar o processo de *Simulated Quenching* (SQ), resultando em

$$\mathbf{T}_i(\mathbf{k}) = \mathbf{T}_i(\mathbf{0}) \times \exp(-\mathbf{C}_i \mathbf{k}^{\frac{Q_i}{D}})$$

(  $Q_i$  = parâmetro de quenching )

Se usarmos parâmetros de *quenching* maiores do que 1, existirá ganho de velocidade mas a convergência ao ótimo global não é mais assegurada. Tal procedimento poderia ser usado em casos de domínios com dimensões elevadas (> 100, por exemplo) e recursos computacionais escassos. Apesar (ou por causa) de toda esta flexibilidade, há muitos ajustes a realizar, do ponto de vista do usuário (“Sistemas não-lineares são, tipicamente, atípicos.”, como diz Lester Ingber - o criador do sistema ASA). No que segue, descreveremos um método bem sucedido de aceleração aplicado ao sistema ASA, que, pela utilização de um controlador *fuzzy* Mamdani, ajusta dinamicamente parâmetros relacionados ao procedimento de *quenching*. Mostraremos que, pelo aumento da percepção de situações de estagnação ou fraco desempenho, é possível tomar ações corretivas e reduzir substancialmente (talvez eliminar) as tarefas de ajuste paramétrico manual. Tudo é feito sem alterar o código do sistema ASA [Oliveira Jr., 2005].

Algoritmos baseados em SA utilizam princípios idealizados por N. Metropolis e outros, sendo conhecidos genericamente pelo rótulo de métodos de Monte Carlo.

A abordagem usa três componentes fundamentais que têm grande impacto na implementação final:

- Uma densidade de probabilidade  $g(\cdot)$ , usada na geração de novos pontos candidatos.
- Uma densidade de probabilidade  $a(\cdot)$ , usada na aceitação/rejeição de novos pontos.
- Um esquema de redução de temperaturas  $T(\cdot)$ , que determina como as mesmas variarão durante a execução do algoritmo, ou seja, seu perfil dinâmico.

A estratégia básica é gerar um ponto inicial, escolhido de acordo com critérios convenientes, e ajustar a temperatura inicial de modo que o espaço de fase possa ser *suficientemente* explorado. A seguir, novos pontos são gerados de acordo com a PDF  $g(\cdot)$  e probabilisticamente aceitos ou rejeitados, conforme determinar a PDF  $a(\cdot)$ .

Se ocorrer a aceitação, o ponto candidato é elevado à condição de ponto básico vigente. Durante a execução do algoritmo as temperaturas são reduzidas, provocando a redução da probabilidade de aceitação de novos pontos posteriormente gerados apresentando valores da função-objetivo superiores àquele do ponto básico corrente (no caso de minimização de funções). Entretanto, existe uma probabilidade não nula de escolha de pontos situados *acima* deste último, tornando possível uma eventual “fuga” de mínimos locais.

Como citado anteriormente, ASA, que é uma realização prática de VFSR, é baseado no conceito de *simulated annealing*, possuindo um grande número de aspectos positivos. Dentre eles, podemos citar:

- ♦ *Re-annealing* – trata-se do re-escalamento dinâmico das temperaturas paramétricas, adaptando PDFs geradoras para cada dimensão de acordo com as sensibilidades exibidas em cada direção de busca. Em poucas palavras, se a função objetivo não apresenta variações significativas quando alteramos um dado parâmetro, pode ser proveitoso estender a amplitude do intervalo de busca naquela dimensão em particular, e vice-versa.
- ♦ Facilidades de *quenching* – como dissemos anteriormente, a implementação do sistema ASA apresenta a possibilidade de ajuste manual (por parte do usuário) de vários parâmetros estruturais relacionados ao processo de *quenching*, o que poderá resultar em maior velocidade de convergência. Assim, é possível moldar a evolução das temperaturas paramétricas de modo amplo, fácil e limpo.
- ♦ Alto nível de parametrização – o sistema ASA foi codificado de tal modo que possamos alterar virtualmente qualquer subsistema sem esforço significativo. Sendo assim, é possível mudar o comportamento dos processos de geração/aceitação, critérios de conclusão, geração de pontos iniciais, nível de detalhe do arquivo de log, etc.

ASA foi projetado para encontrar mínimos globais pertencentes a um dado subconjunto compacto do espaço Euclidiano n-dimensional. Ele gera pontos componente a componente, de acordo com:

$$\begin{aligned}
x_{i+1} &= x_i + \Delta x_i, \\
\text{com } \Delta x_i &= y_i (B_i - A_i), \\
[A_i, B_i] &= \text{faixa de variação da } i\text{-ésima dimensão,} \\
y_i \in [-1, 1] &\text{ é dado por} \\
y_i &= \text{sgn}(u_i - 1/2) T_i [(1 + 1/T_i)^{|2u_i - 1|} - 1] \text{ onde} \\
u_i \in [0, 1] &\text{ é gerado por meio da distribuição uniforme,} \\
T_i &= \text{temperatura atual relativa à dimensão } i.
\end{aligned}$$

A compacidade do espaço de busca não é uma severa limitação na prática, e na falta de prévia informação sobre a possível localização de mínimos globais, basta escolher domínios hiper-retangulares suficientemente abrangentes [Oliveira Jr., 2005].

Conforme citação anterior, o uso do mecanismo de *quenching* pode melhorar substancialmente a eficiência do processo de convergência, com a assunção do risco de alcance prematuro de mínimos não globais. Em certos casos, contudo, podemos simplesmente não ter alternativa, como é o caso de funções com domínios de elevado número de dimensões. Para resolver o problema, um controlador *fuzzy* foi projetado. A abordagem é simples: consideramos ASA como um sistema dinâmico MISO (*Multiple Input Single Output*) e "fechamos a malha", pela amostragem da saída de ASA (valor corrente da função objetivo) e atuação em suas entradas (um subconjunto de parâmetros ajustáveis em *run-time*, relacionados ao processo de *quenching*) de acordo com uma *lei fuzzy* (algoritmo de controle), que nada mais faz do que emular o raciocínio humano sobre o processo subjacente. Assim, pelo uso de um controlador inteligente podemos acelerar e retardar a evolução das temperaturas, além de sermos capazes de tomar ações evasivas em caso de convergência prematura [Oliveira Jr., 2005].

Enfrentamos dois principais obstáculos para alcançar tal objetivo:



1. Como as saídas amostradas (valores da função objetivo) podem informar o estado atual do processo de minimização em andamento?
2. Como podemos alterar dinamicamente as entradas do sistema ASA de modo a eliminar situações indesejáveis, como permanência junto a mínimos não globais ou progresso insatisfatório?

A primeira questão foi resolvida graças ao conceito de função de sub-energia, usada no método TRUST [Barhen et al., 1997].

A função de sub-energia é dada por:

$$SE(x, x_0) = \log(1/[1 + \exp(-(f(x) - f(x_0)) - a)])$$

onde  $a$  é uma constante real, e  $x_0$  é o "ponto básico" atual.

O ponto básico é o melhor ponto de mínimo encontrado até então. Logo, a função SE comporta-se qualitativamente como a original  $f(\cdot)$  quando o processo de busca "visita" pontos melhores do que o mínimo corrente e tende a se achatar em pontos piores. Assim, é possível avaliar quando a busca está concentrada acima, nas imediações ou abaixo do ponto mínimo atual pela inspeção dos valores assumidos pela função de sub-energia. Tal processo de detecção resulta em conclusões aproximadas como

A busca está *PRÓXIMA* do mínimo vigente.

ou

A busca está *MUITO DISTANTE* do mínimo vigente.

O que leva naturalmente a uma oportunidade de modelagem *fuzzy*.

A segunda questão acima está relacionada às partes conseqüentes da base de regras *fuzzy*, na qual temos que inserir ações corretivas para eventuais desvios em relação às diretrizes preestabelecidas para o processo de minimização. Isto foi feito pela variação dos graus de *quenching* para PDFs de geração e aceitação. A implementação usou fatores de *quenching* individuais para cada dimensão.

A base de regras do controlador fuzzy contém asserções como

- ♦ SE AveSub está PRÓXIMA a zero ENTÃO  
  
    aumente o nível de Quenching.
  
- ♦ SE AveSub está PRÓXIMA ao mínimo corrente ENTÃO  
  
    aumente o nível de Quenching
  
- ♦ SE StdDevSub é ZERO ENTÃO  
  
    decreça o nível de Quenching

onde:

- ♦ AveSub é uma variável lingüística correspondente à média abrupta (*crisp*) dos 100 últimos valores de sub-energia
  
- ♦ StdDevSub é uma variável lingüística correspondente ao desvio-padrão abrupto (*crisp*) dos 100 últimos valores de sub-energia

Tendo descrito a estrutura geral da abordagem adotada, é hora de mostrar alguns resultados práticos, obtidos através da otimização de algumas funções de difícil tratamento [Oliveira Jr., 2005].

Apresentaremos quatro testes nos quais funções multi-modais foram submetidas a três métodos: ASA, *Fuzzy* ASA (descrito neste trabalho) e um algoritmo genético de ponto flutuante eficiente e bastante conhecido, cujo nome não será citado por motivos éticos. ASA e *Fuzzy* ASA mantiveram exatamente os mesmos parâmetros em cada caso ; a única diferença entre as execuções sendo a ativação do controlador inteligente em *Fuzzy* ASA, localizado em módulo externo e ativado de dentro da função de custo. O controlador ficou inalterado em todos os testes, evidenciando sua independência em relação às características da função objetivo e/ou dimensão de seu domínio.

O algoritmo genético foi utilizado com as seguintes características:

- ♦ Tamanho da população – 75
- ♦ Elitismo – Operante
- ♦ População inicial idêntica à semente de ASA
- ♦ 3 operadores de reprodução (*crossover*)
- ♦ 5 operadores de mutação

As funções de teste são:

Função 1:

Domínio:  $\{x \in \mathbb{R}^3 : x_i \in [-10000, 10000]\}$

$$f(x) = x_1^2(2 + \sin(120x_2)) + x_2^2(2 + \sin(220x_1)) + x_3^2(2 + \sin(50x_1))$$

Mínimo global em  $(0,0,0)$ .

Valor mínimo = 0.

Função 2:

Domínio:  $\{x \in \mathbb{R}^4 : x_i \in [-10, 10]\}$

$$f(x) = 100(x_2 - x_1)^2 + (1 - x_1)^2 + 90(x_1 - x_3)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$$

Mínimo global em  $(1,1,1,1)$ .

Valor mínimo = 0.

Função 3:

Domínio :  $\{ x \in \mathbb{R}^{50} : x_i \in [-10, 10] \}$

$$f(x) = f_1(x) + f_2(x) + f_3(x)$$

$$f_1(x) = \sum_{i=1}^{50} (ix_i^2)$$

$$f_2(x) = \sum_{i=1}^{50} (x_{i-1} + 5\sin x_i + x_{i+1}^2)^2$$

$$f_3(x) = \sum_{i=1}^{50} \ln^2(1 + |\sin^2 x_{i-1} + 2x_i + 3x_{i+1}|)$$

com  $x_0 = x_{50}$  e  $x_{51} = x_1$

Mínimo global em  $0 \in \mathbb{R}^{50}$ .

Valor mínimo = 0.

Função 4:

Domínio : Mesmo da função 3.

$$f(x) = f_1(x) + f_2(x) + f_3(x)$$

$$f_1(x) = \sum_{i=1}^{50} (ix_i^2)$$

$$f_2(x) = \sum_{i=1}^{50} \sin^2(x_{i-1} \sin x_i - x_i + \sin x_{i+1})$$

$$f_3(x) = \sum_{i=1}^{50} \ln(1 + i(x_{i-1}^2 - 2x_i + 3x_{i+1} - \cos x_i + 1)^2)$$

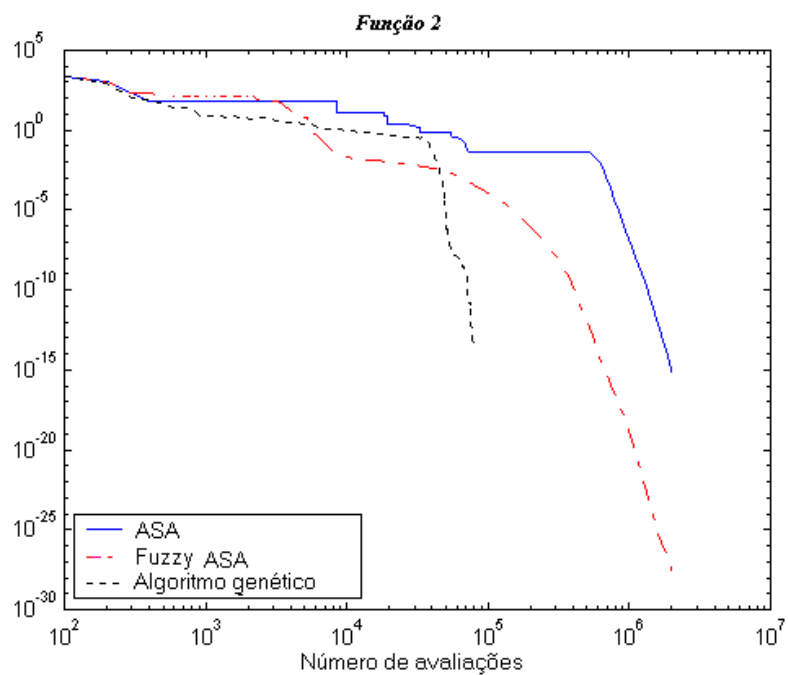
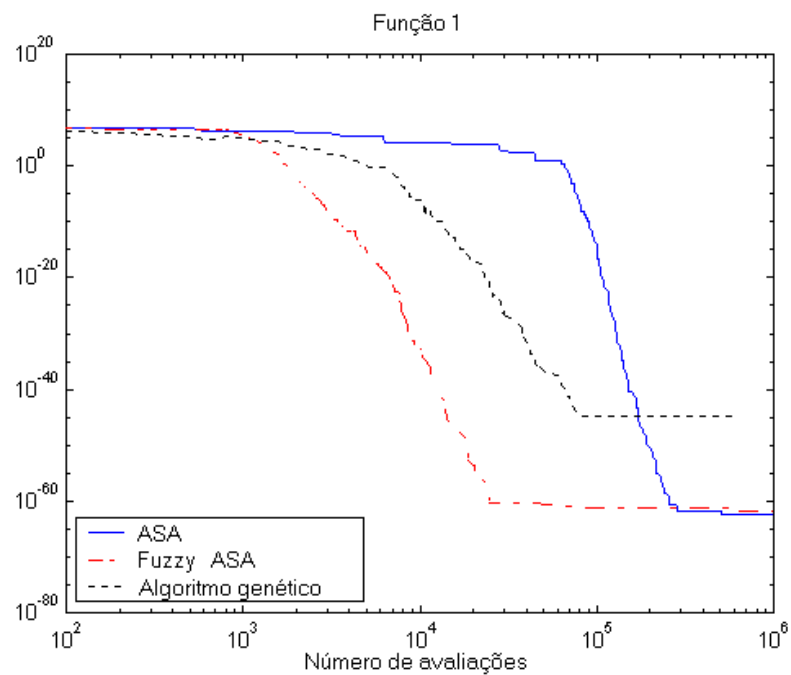
com  $x_0 = x_{50}$  e  $x_{51} = x_1$

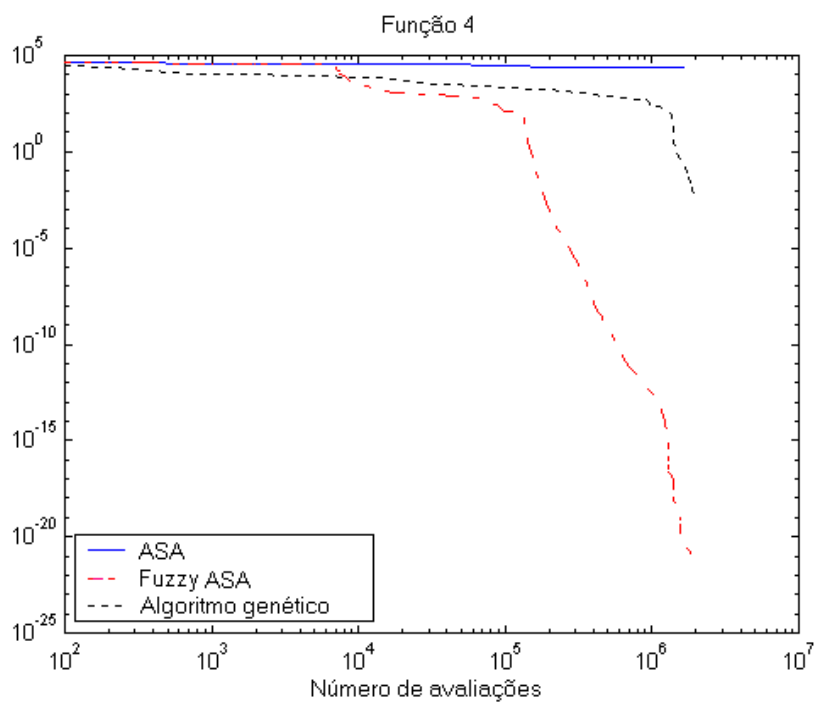
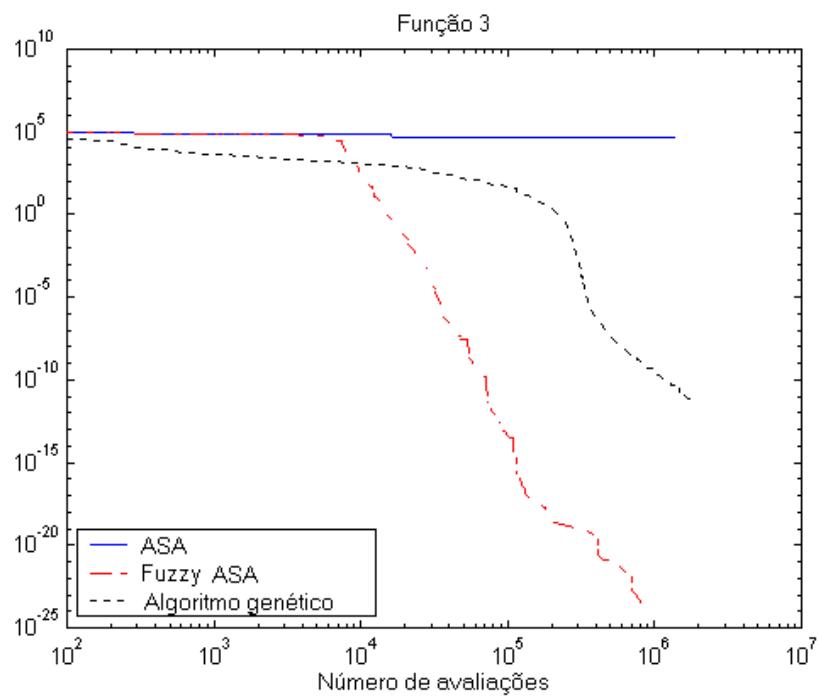
Mínimo global em  $0 \in \mathbb{R}^{50}$ .

Valor mínimo = 0.

A seguir, mostramos a evolução dos processos de minimização. No eixo das abscissas temos o número de avaliações da função-objetivo e no das ordenadas o melhor valor encontrado até então. Como se pode constatar pelos gráficos, o algoritmo proposto apresenta desempenho sempre

superior ao sem realimentação, superando ainda, nos casos de altas dimensões, o FPGA (*loating Point Genetic Algorithm*). Isto o coloca entre as soluções viáveis para problemas práticos.





#### 4.4. ESTIMADORES COM PLAUSIBILIDADE MÁXIMA USANDO FUZZY ASA

O método da plausibilidade máxima é uma técnica bastante popular para obtenção de estimadores de parâmetros de distribuições de probabilidades.

O procedimento geral é simples:

Dada uma amostra independente e identicamente distribuída ( $X_1, X_2, \dots, X_n$ ) de uma população com suposta função densidade de probabilidade

$$f(\mathbf{x} | \theta_1; \theta_2; \dots; \theta_k),$$

função de plausibilidade é definida por [Casella et al., 2001]

$$L(\boldsymbol{\theta} | \mathbf{x}) = L(\theta_1, \theta_2, \dots, \theta_k | x_1, x_2, \dots, x_n) = \prod f(x_i | \theta_1; \theta_2; \dots; \theta_k).$$

Quando as premissas de independência e/ou igualdade de distribuições não são satisfeitas para o conjunto amostral, a função de plausibilidade é definida como a densidade conjunta  $f(x_1, x_2, \dots, x_n | \boldsymbol{\theta})$ , considerada como função do vetor  $\boldsymbol{\theta}$ , uma vez que a amostra é fixa.

O vetor  $\boldsymbol{\theta}^*$  que maximiza  $L(\boldsymbol{\theta} | X)$  é denominado estimador de plausibilidade máxima sendo, como o nome indica, uma aproximação do vetor paramétrico correspondente à distribuição real da população ou processo sob estudo.



Embora tenha sido definida por meio da expressão acima, a maximização propriamente dita é realizada normalmente utilizando seu logaritmo, tendo em vista que tal transformação tende a prevenir a ocorrência de *underflows*, além de simplificar a expressão da função de plausibilidade original. Tendo em vista a monotonia da função logaritmo (base 10, em particular), o maximizante da função transformada coincidirá com o da função original. Na literatura pertinente encontramos inúmeros casos onde o procedimento de otimização é realizado por meio da anulação do gradiente da função-produto e posterior solução das equações resultantes.

Infelizmente, tal abordagem não pode ser adotada nos casos em que a função não é diferenciável e se mostra pouco eficaz caso a função-objetivo apresente diversos pontos de máximo locais. Outro ponto desfavorável desta abordagem tradicional ocorre quando as equações obtidas pela anulação do gradiente são de difícil tratamento e/ou não possuem expressão de solução em forma fechada. Logo, métodos computacionais alternativos podem vir a contribuir para ampliar o campo de aplicação do método da plausibilidade máxima.

Seria desejável que tais métodos possuíssem as seguintes características:

- ♦ Prescindir da continuidade ou diferenciabilidade das funções-objetivo.
- ♦ Realizar otimização global, ou seja, não ser “aprisionados” em pontos de máximo locais.
- ♦ Possuir comprovação teórica de convergência, ao menos em probabilidade, a pontos de máximo global.

O método apresentado anteriormente (*Fuzzy Adaptive Simulated Annealing*) apresenta todas as características acima e se mostra adequado para uso na tarefa em questão.

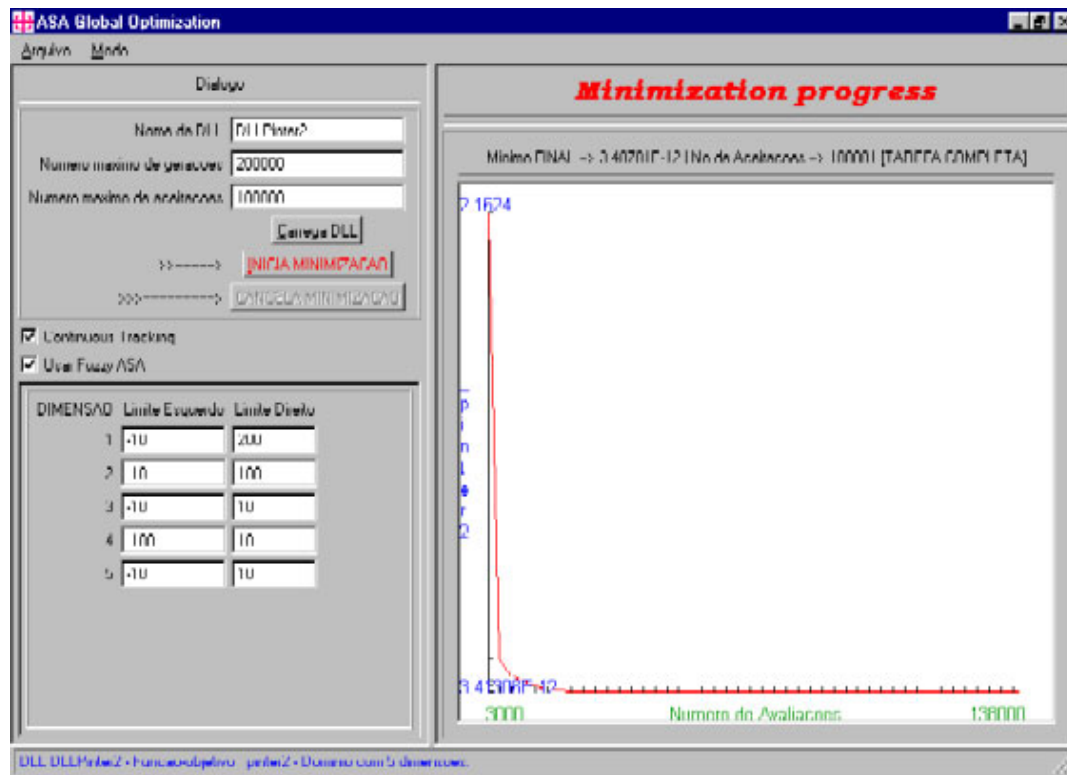
Logo, o processo de maximização será radicalmente alterado e sua dinâmica será aquela de um processo estocástico que varre o espaço paramétrico, buscando convergência ao ótimo global por mecanismos completamente diferentes. Outro aspecto digno de nota é que trata-se de um abordagem fortemente computacional e de maior abrangência, tendo em vista o relaxamento de inúmeras características das funções densidade de probabilidade sob análise.

Neste ponto da pesquisa, foi necessário construir alguns artefatos de software para validar as idéias expostas acima e dar solidez aos resultados finais desejados.

Para tanto, foi feito um programa extensível cuja estrutura consiste de um núcleo fixo, interface gráfica amigável e que permite a incorporação de *plug-ins* contendo o código das funções a otimizar. Como resultado final, são fornecidos os parâmetros correspondentes aos pontos ótimos pertencentes ao domínio das funções sob tratamento. A opção foi por oferecer uma “máquina de minimização global” geral, pois, em caso de maximização, basta inverter (em termos aditivos) o sinal da função-objetivo no módulo descritor da mesma. A idéia é extremamente simples: para investigar os parâmetros de uma dada distribuição, dadas as amostras aleatórias, basta construir um módulo de software (cuja estrutura será detalhada abaixo) contendo alguns poucos métodos com interfaces bem definidas. Dentre eles, um será responsável por avaliar a função-objetivo (inverso aditivo do logaritmo da função de plausibilidade, no caso de MLE ) em pontos arbitrários do domínio e devolver o valor correspondente ao chamador. Tal tipo de arquitetura permite desacoplar o mecanismo de otimização propriamente dito dos cálculos particulares relativos a uma dada função numérica. Logo, após os procedimentos de ajuste iniciais, o núcleo central inicia um processo iterativo, chamando continuamente a função responsável pelo cálculo dos valores numéricos e simulando um processo estocástico que visita estados com “energias” cada vez menores e rumo à configuração que deverá conter algum ótimo global para o problema – é claro que as condições para convergência deverão ser cumpridas para que o resultado desejado seja

atingido. Além disso, pode ser necessário ajustar alguns parâmetros para obtenção de melhor desempenho

A seguir, vemos algumas aspectos da interface gráfica



## INTERFACE GRÁFICA

O programa foi codificado na linguagem C++, de modo a permitir seu fácil transporte a diversos ambientes, dentre eles vários “sabores” do sistema operacionais Unix (Linux, em particular). Para utilizá-lo, basta codificar um módulo carregável dinamicamente que contenha, no mínimo, 3 rotinas específicas do problema em questão, a saber:

- ◆ extern "C" char \* NomeDaFuncaoObjetivo(void)

Devolve uma *string* contendo o nome da função-objetivo a ser investigada.

- extern "C" unsigned int NumeroDeParametrosDaFO(void)

Devolve a dimensão do espaço no qual o domínio da função-objetivo está imerso.

- extern "C" double FuncaoObjetivoPropriamenteDita

(int NoDeDimensoes, double \*Vetor, void \*Custom )

Devolve o valor da função, dados o ponto de cálculo e um ponteiro contendo informações de responsabilidade do usuário.

A título de exemplo, mostramos abaixo o código-fonte de um módulo completo na linguagem C:

---

```
#include <windows. h>
#include <math. h>

char NomeComUnderscore[] = "_pinter1";

BOOL DllMain(HINSTANCE hInst,DWORD dwReason,LPVOID Reservado)
{
    switch (dwReason)
    {
        case DLL_PROCESS_ATTACH:           break;
        case DLL_PROCESS_DETACH:          break;
        case DLL_THREAD_ATTACH:           break;
        case DLL_THREAD_DETACH:           break;
    }

    return TRUE;
}

extern "C" double __declspec(dllexport) pinter1(
    int nodedimensoes, // No de elementos no vetor abaixo
    double *x, // Vetor
    void *Custom // Informacao especifica do usuario
```

```

    )
{
int i;

double f1,f2,f3,aux;

    f1=0;
    for (i=0;i<nodedimensoes;i++)
        f1 += (i+1)*x[i]*x[i];

    f2 = pow(x[nodedimensoes-1]+5*sin(x[0])+x[1]*x[1],2);
    aux = pow(x[nodedimensoes-2]+5*sin(x[nodedimensoes-1])+x[0]*x[0],2) ;

    f2 = f2 + aux;

    for (i=1 ; i < nodedimensoes-1 ; i++)
        f2 += pow(x[i-1]+5*sin(x[i])+x[i+1]*x[i+1],2);

    f3 = log(1+fabs(pow(x[nodedimensoes-1],2)+2*x[0]+3*x[1]));
    f3 = f3*f3 ;
    aux = log(1+fabs(nodedimensoes*pow(x[nodedimensoes-
2],2)+2*x[nodedimensoes-1]+3*x[0]));
    aux = aux*aux;

    f3 = f3+aux;

    for (i=1 ; i < nodedimensoes-1 ; i++)
    {
        aux = log(1+fabs((1+i)*pow(x[i-1],2)+2*x[i]+3*x[i+1]));
        f3 += aux*aux ;
    }

    aux=f1+f2+f3;

    return aux ;

}

extern "C" char * __declspec(dllexport) NomeDaFuncaoObjetivo(void)
{
    return (char *) NomeComUnderscore;
}

extern "C" unsigned int __declspec(dllexport) NumeroDeParametrosDaFO(void)
{
    return 5;
}

```

---

A partir deste ponto, é obviamente necessário compilar e “link-editar” o programa-fonte para criação do binário (*shared object*, DLL, etc.) que será carregado em *run-time* pelo programa principal, após as devidas configurações via interface gráfica. Os controles disponíveis na presente versão são relativos ao nome do arquivo que contem o código da biblioteca dinamicamente carregável, ao número de aceitações e gerações do algoritmo de otimização, à definição do domínio de otimização que restringe o espaço de estados do processo a ser simulado e ao tipo de algoritmo a ser utilizado durante a execução. Como instrumento complementar, uma janela mostra a evolução dos valores numéricos assumidos pelos pontos intermediários, na busca pelo extremo global. Mostraremos agora o resultado da aplicação do programa acima à obtenção simultânea dos 2 parâmetros de uma distribuição lognormal, tendo como entrada amostras geradas *a priori* por meio da CDF ideal (distribuição com parâmetros 4 e 2) e compostas por 100, 500, 5000 e 8000 elementos – cabe ressaltar que o procedimento não lançará mão de qualquer método relacionado a gradientes e outros mecanismos do Cálculo Diferencial, além de desconsiderar por completo aspectos particulares desta distribuição específica. Primeiramente, apresentamos o código-fonte da DLL utilizada:

---

```
#include <windows. h>
#include <stdio. h>
#include <stdlib. h>
#include <math. h>
#include <exceptio. h>
# include <cstdlib>
# include <iostream>
# include <iomanip>
# include <cmath>
# include <ctime>
#include "prob. h"

#define ARQUIVO "AMOSTRAS. DAT"
#define NOMAXIMODEENTRADAS 10000

char NomeComUnderscore[] = "_MLELognormal";

static int NoDeAmostras ;
double SampleArray[NOMAXIMODEENTRADAS];
```

```

BOOL DIIMain(HINSTANCE hInst,DWORD dwReason,LPVOID Reservado)
{
FILE *fp;
char amostra[100] ;

switch (dwReason)
{
    case DLL_PROCESS_ATTACH:

        NoDeAmostras = 0;
        fp = fopen(ARQUIVO,"r");

        while ( fscanf(fp,"%s",amostra) != EOF && \
                NoDeAmostras < NOMAXIMODEENTRADAS
                )
            {
                SampleArray[NoDeAmostras] = strtod(amostra,NULL);
                NoDeAmostras++; ;
            }
        fclose(fp);

        if (NoDeAmostras == 0) return FALSE; // Inicializacao falhou
            // => impede a carga

            break;

        case DLL_PROCESS_DETACH:                break;
        case DLL_THREAD_ATTACH:                break;
        case DLL_THREAD_DETACH:                break;
    }

return TRUE;

}

extern "C" double __declspec(dllexport) MLELognormal(
    int nodedimensoes, // No de elementos no vetor abaixo
    double *x, // Vetor
    void *Custom // Informacao especifica do usuario
    )
{
double aux = 0 ;
double resu ;

for ( int i=0 ; i < NoDeAmostras ; i++ )
    {
        resu = log_normal_pdf(SampleArray[i],x[0],x[1]) ;
    }
}

```

```

    if (resu < 1e-20)
        aux += -30;
    else aux += log(resu) ;

}

return -aux ; // Note a inversao aditiva para maximização
}

extern "C" char * __declspec(dllexport) NomeDaFuncaoObjetivo(void)
{
    return (char *) NomeComUnderscore;
}

extern "C" unsigned int __declspec(dllexport) NumeroDeParametrosDaFO(void)
{
    return 2;
}

```

---

Foram gerados 4 conjuntos de amostras independentes contendo 100, 500, 5000 e 8000 pontos distribuídos de acordo com a lognormal com parâmetros (4,2), cada um dos quais tendo sido utilizado para estimar os 2 parâmetros da referida distribuição de acordo com o método acima apresentado. Os resultados são mostrados na tabela abaixo:

No de pontos na amostra	Parâmetro A estimado	Parâmetro B estimado
100	3.696686	2.019308
500	3.964747	1.92806
5000	3.982984	1.968152
8000	3.994115	1.990389

Pode-se observar que o comportamento assintótico do procedimento proposto é coerente com os resultados teóricos, uma vez que o par de parâmetros estimados tende visivelmente ao par (4,2) quando o número de amostras aumenta.



Finalmente, temos os resultados da aproximação dos parâmetros relativos a uma população distribuída de acordo com Cauchy (3.5 , 1), tendo como entrada amostras geradas *a priori* por meio da CDF ideal e compostas por 100, 1000, 4000 e 9000 elementos.

No de pontos na amostra	Parâmetro A estimado	Parâmetro B estimado
100	3. 487382	0. 7965445
1000	3. 519669	0. 8635537
4000	3. 540199	0. 8407481
9000	3. 526978	0. 8521081

Embora os exemplos tenham sido baseados em distribuições unidimensionais, o método apresentado pode ser aplicado diretamente a distribuições multivariadas – é claro que, quanto maiores forem a complexidade da função de plausibilidade, a cardinalidade do conjunto amostral e o número de dimensões, maior será o esforço computacional para realizar a aproximação paramétrica.

Após este processo intermediário de validação das ferramentas, é possível prosseguir rumo aos estágios finais da pesquisa.

#### 4.5. ÁRVORES DE FALHAS POSSIBILISTICAS E PROBABILISTICAS

A Teoria da Confiabilidade, que tem por escopo os métodos, critérios e estratégias a serem usados nas fases de concepção, projeto, desenvolvimento, operação e manutenção de sistemas, objetiva garantir o máximo de eficiência, segurança, economia e durabilidade dos mesmos. Em especial, visa o prolongamento da atividade de sistemas a plena carga e de modo contínuo.

Fundamentalmente, tem como objetivos principais [Machado, 2005]:

- ♦ Estabelecer as leis estatísticas da ocorrência de falhas nos dispositivos e nos sistemas.

- ♦ Estabelecer métodos que permitam melhorar os dispositivos e sistemas mediante a introdução de estratégias capazes da alteração de índices quantitativos e qualitativos relativos a falhas.

Uma das finalidades da Confiabilidade é a elaboração de regras que permitam a concepção de sistemas muito complexos (computadores, redes elétricas, indústrias químicas, sistemas de geração elétrica, aviões, naves espaciais, sistema de controle e proteção, etc.) capazes de funcionar satisfatoriamente mesmo com a ocorrência de falhas em alguns dos seus componentes mais críticos. Os princípios da Teoria da Redundância nasceram deste problema.

A metodologia para análise de confiabilidade cujos dados não estão disponíveis para a utilização de modelos probabilísticos, através da determinação de funções densidade de probabilidade que melhor se ajuste aos dados, baseia-se em modelos possibilísticos, utilizando opiniões de especialistas na área específica que estiver sendo estudada.

Os dados possibilísticos são tratados utilizando-se a álgebra dos números *fuzzy*. Cada possibilidade associada a um evento é, por exemplo, um número triangular. As operações envolvidas para a determinação dos caminhos críticos de uma árvore de falhas, bem como a possibilidade de falha do evento principal, são operações algébricas com números *fuzzy*.

A metodologia para análise de confiabilidade cujos dados encontram-se disponíveis utiliza, tipicamente, modelos probabilísticos. Em caso de estudos teóricos utiliza-se uma norma como por exemplo, a MIL – HDBK americana. As normas contêm taxas médias de falha para cada tipo de componente físico utilizado na produção de sistemas, tais como transistores, diodos, resistores, etc. . Um cartão ou placa é um conjunto desses componentes, logo a taxa de falhas de cada cartão pode ser calculada e, então, a do sistema como um todo.

Sabe-se, por exemplo, que vários componentes eletrônicos apresentam taxas de falha praticamente constantes, o que indica um modelo probabilístico exponencial; componentes mecânicos têm modelo probabilístico de Erlang e, em geral, a taxa de falhas do equipamento é crescente ou constante.

No caso de sistemas de software, admitindo-se a não inserção de erros durante a manutenção, a taxa de falhas é decrescente. Supondo conhecidas as taxas apresentadas pelos eventos básicos de uma árvore de falhas, pode-se assumir o modelo probabilístico de Weibull que, através da alteração de valores dos parâmetros característicos da distribuição, permite obter taxas de falhas crescentes, decrescentes ou constantes.

Admitindo-se não haver volume adequado de dados históricos para a taxa de falhas de um dado sistema, os modelos probabilísticos não devem ser adotados, pela inviabilidade de realizar procedimentos de inferência estatística com número insuficiente de amostras. Logo, modelos alternativos podem ser adotados através da Teoria das Possibilidades. Os modelos possibilísticos utilizam banco de dados baseados em opiniões de especialistas [Machado, 2005].

A metodologia consiste em:

- ♦ Criar de um banco de dados de possibilidades de falha, baseado em opiniões de especialistas;
- ♦ Determinar os caminhos críticos da dinâmica de funcionamento e identificar os módulos que mais contribuem para a falha do sistema como um todo;
- ♦ Calcular a confiabilidade através da árvore de falhas;

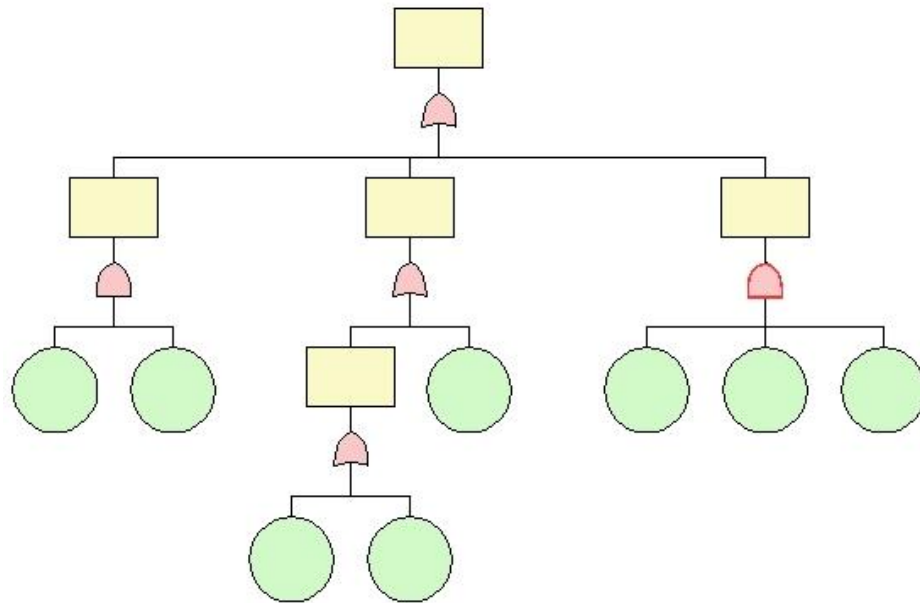
- Determinar as causas e efeitos das falhas possíveis, considerando-se diversos níveis (efeitos sobre outros módulos, sobre o sistema e sobre o ambiente);
- Determinar as falhas admissíveis (falhas que não são catastróficas) e as falhas inadmissíveis (falhas catastróficas).

A Análise por Árvore de Falhas (FTA – *Fault Tree Analysis*) mostra:

- Os aspectos relevantes ao sistema;
- A degradação de desempenho do sistema, quando são consideradas as falhas relevantes dos subsistemas;
- As possíveis falhas do sistema;
- os eventos básicos do evento principal;
- A informação gráfica das falhas do sistema.

A técnica da árvore de falhas estuda a probabilidade/possibilidade de falha do sistema global e tem por objetivo determinar a confiabilidade do mesmo. Para obter este resultado, cada um dos eventos básicos da árvore de falhas terá probabilidades/possibilidades de falhas associadas aos mesmos. Estas são caracterizadas por funções de pertinência determinadas de modo ‘subjetivo’ ou probabilidades obtidas por meios inferenciais estatísticos.

O evento principal é dependente dos sub-eventos e eventos básicos. Estes eventos básicos são considerados eventos das falhas iniciais, que é o nível mais baixo do estudo. O evento principal, sub-eventos e eventos básicos estão interrelacionados através de operações lógicas. A análise por árvore de falhas se baseia na Álgebra Booleana, podendo utilizar os operadores básicos AND, OR, NOT, XOR e sua composição arbitrária.



### EXEMPLO DE ÁRVORE DE FALHAS

O banco de dados baseado nas opiniões de especialistas se refere às possibilidades de falhas dos eventos básicos ou atômicos.

Para os módulos críticos, FMECA (*Failure Mode and Effect and Crittically Analysis*) devem ser realizadas com a finalidade de se avaliar as falhas admissíveis, podendo-se minimizar custos.

No caso da abordagem probabilística, o evento principal é obtido do seguinte modo:

- ♦ assume-se que os eventos básicos são independentes;
- ♦ um ou mais eventos básicos podem causar uma falha em um ou mais eventos intermediários e assim sucessivamente;
- ♦ a probabilidade de falha de cada evento básico é conhecida; para o evento básico ‘ $A_i$ ’, esta probabilidade será notada por  $p(A_i)$
- ♦ a probabilidade de cada evento intermediário da árvore de falhas é obtida através da substituição dos operadores lógicos AND, OR e NEG pelas operações de adição, produto ou complemento dos valores probabilísticos
- ♦ para uma seqüência de ‘n’ sub-eventos independentes, com o operador lógico AND, a probabilidade de falha é dada por:

$$P^{AND}(t) = \prod P_{A_i}(t)$$

e a confiabilidade é dada por:

$$R^{AND}(t) = 1 - P^{AND}(t)$$

- ♦ para uma seqüência de ‘n’ sub-eventos independentes, relacionados com o operador lógico OR, a probabilidade de falha é computada por meio da fórmula análoga, que expressa a probabilidade da união de eventos, dadas as probabilidades de cada um deles.

Do mesmo modo, é possível se adotar essas expressões para um modelo possibilístico utilizando-se a álgebra da Teoria dos Conjuntos *Fuzzy*.

As opiniões dos especialistas são caracterizadas por números *fuzzy*. Para obtenção do caminho crítico da árvore de falhas, bem como determinar a possibilidade de falha do evento principal, são utilizadas as operações aritméticas adequadas aos números *fuzzy*, dependendo do tipo de dependência lógica expressa por meio da árvore de falhas [Machado, 2005].

Após o estabelecimento destes pré-requisitos, passaremos a tratar a questão central desta seção.

É necessário compreender que a redundância de recursos é um pré-requisito para obtenção de um alto grau de disponibilidade, pois é necessária para eliminar os pontos únicos de falha (*Single Point Of Failures* - SPOF). É importante lembrar que os SPOF não se restringem somente aos aspectos físicos das máquinas, existindo também em:

- ♦ Redes de Computadores (*switches*, roteadores, hubs, cabeamento, etc.) que disponibilizam serviços ao mundo externo;
- ♦ Redes Elétricas internas;
- ♦ Localizações dos Recursos. Deve-se lembrar que desastres (terremotos, furacões, ataques terroristas, etc) podem inviabilizar toda a operação de uma companhia, se esta mantiver suas informações e sistemas operando em apenas um local.

Assim, fica claro que quando tratamos de um sistema de alta disponibilidade, este deve englobar muito mais do que um simples sistema de redundância de recursos. É claro que tudo isso depende

do grau de disponibilidade que se deseja atingir; por exemplo, uma disponibilidade contra ataques terroristas não é pretendida na maioria das situações. Também, é fácil perceber que quanto maior o grau de disponibilidade que desejamos atingir, maiores serão os gastos com a solução. Outro detalhe a ser abordado sobre alta disponibilidade é como medi-la. Em termos técnicos, a disponibilidade de certo serviço é dita como sendo a probabilidade de encontrá-lo operando normalmente em determinado momento. Portanto, tal probabilidade leva em conta qual o provável *uptime* (tempo em que os serviços estarão funcionando) e o provável *downtime* (tempo em que os serviços ficarão fora do ar).

Pode-se calcular a disponibilidade através da fórmula:

$$A = \frac{MTBF}{MTBF + MTTR}$$

onde A é o grau de disponibilidade expresso como uma percentagem, MTBF é o tempo médio entre falhas (*Mean Time Between Failures*), e MTTR é o tempo médio para reparos (*Mean Time To Repair*).

As dificuldades associadas com a estimativa e previsão dos tempos das tarefas ligadas às atividades de manutenção resultam do fato de que esses tempos não são determinísticos. Há o elemento humano presente no processo de manutenção nas diversas fases existentes e, conseqüentemente, observa-se grande variabilidade estocástica associada aos tempos relacionados às atividades de manutenção.



Manutenibilidade é um índice associado a um equipamento sob reparo. É a probabilidade de que um equipamento em falha seja reparado dentro de um tempo "t". Se "T" é uma variável aleatória representando o tempo de reparo, então a manutenibilidade é definida como descrito abaixo:

Para o caso em que o tempo de reparo é exponencialmente distribuído, com uma taxa de reparo ( $\mu$ ), a função densidade de probabilidade é dada por:

$$m(t) = \mu \cdot \exp(-\mu t)$$

e a CDF, por:

$$M(t) = \int_0^t \mu \cdot e^{-\mu t} \cdot dt = 1 - e^{-\mu t}$$

Logo:

O valor esperado do tempo de reparo (TMPR - Tempo Médio Para Reparo) é dado por:

$$\text{TMPR} = \int_0^{\infty} [1 - M(t)] \cdot dt = \frac{1}{\mu}$$

Exemplo:

Quando da falha de um microcomputador, a empresa contratada para manutenção leva, em média, 3 horas para efetuar o reparo. Qual a probabilidade de que o reparo leve até 2,5 horas?

$$\mu(\text{taxa de reparo}) = \frac{1}{\text{TMFR}} = \frac{1}{3} = 0,3333 \text{ reparos/hora}$$

Logo:

Para "t" igual ou menor a 2,5 horas, a probabilidade é dada pela função de manutenibilidade, ou seja:

$$M(2,5) = 1 - \exp(0.333 * 2,5) = 0.565$$

Portanto, a probabilidade de que o reparo seja efetuado em 2,5 horas ou menos é de 56,5%. A distribuição exponencial negativa é utilizada tipicamente para os tempos de reparo de equipamentos eletrônicos (modularizados). A exponencial representa aqueles equipamentos, os quais requerem ajustes relativamente freqüentes de curta duração ou aqueles que podem retornar para o serviço por meio de uma rápida operação de troca.

Há casos em que o tempo de reparo é regido pela distribuição lognormal. Logo, temos que considerar o fato de que a utilização da mesma nos estudos de manutenibilidade está baseada na Lei de Weber-Fechner, que provou que o tempo de resposta de animais e elementos humanos a simples estímulos físicos ou percepções são, freqüentemente, lognormais. Estudos experimentais mais recentes do tempo de reação de elementos humanos para modelos de percepção mais complicados, envolvendo algum grau de aprendizagem, indicaram que esses tempos de reação seguem uma distribuição lognormal. É também verdadeiro que o processo de localização e reparo de falhas em equipamentos segue uma seqüência lógica. Desse modo, essas considerações indicam que os tempos de paralisação, quando da manutenção de sistemas, são lognormalmente

distribuídos, visto que a localização e correção de falhas em equipamentos é uma extensão do processo lógico de classificação em categorias.

As fórmulas abaixo representam a manutenibilidade ( $M(t)$ ) e o tempo médio para reparo (TMPR), neste caso:

$$M(t) = \Phi\left(\frac{\text{Log}_e t - \text{Log}_e \beta}{\alpha}\right)$$

$$\text{TMPR} = \beta \cdot \text{Exp}\left(\frac{\alpha^2}{2}\right)$$

#### 4.6. INTEGRAÇÃO DE RESULTADOS INTERMEDIÁRIOS

Para produzir os resultados finais esperados, os elementos teóricos descritos anteriormente foram integrados de modo que os procedimentos de avaliação possam ser realizados de modo sistemático, ou seja, seguindo um algoritmo precisamente definido e de fácil execução por parte do usuário final. Ao mesmo tempo, trata-se de um modelo lógico simples e extensível de aferição de confiabilidade, comportando agregação de novos métodos quantitativos que possam colaborar para aperfeiçoamento de resultados finais.

Em qualquer uma das abordagens (possibilística ou probabilística), é necessária a construção de um diagrama esquemático de dependência operacional, definindo detalhadamente como subsistemas ou componentes estão relacionados entre si e como interferem no funcionamento uns dos outros, em termos lógicos.

Vencida esta etapa fundamental, é recomendável a realização de uma análise do histórico de falhas da planta como um todo:

- ♦ Há volume suficiente de dados históricos para uma análise estatística?
- ♦ Os dados são confiáveis?
- ♦ Os dados são atuais?
- ♦ O procedimento de registro de falhas é realizado sistemática e periodicamente?
- ♦ A periodicidade de coleta de informações relativas a falhas é suficiente ?

Se todas as perguntas anteriores forem respondidas afirmativamente, podemos optar diretamente por qualquer uma das duas abordagens, sendo a probabilística a mais indicada.

Em caso de inexistência de informações históricas adequadas e/ou inviabilidade de implantação de procedimentos de coleta em prazos considerados satisfatórios, certamente obteremos melhores resultados com a abordagem possibilística, usando opiniões de especialistas.

Para por em prática as idéias acima, relativas à árvore de falhas, foram adotadas duas abordagens distintas:

A realização possibilística parte das estimativas de confiabilidade fornecidas por especialistas para as folhas e, recursivamente, calcula os graus de falha dos subsistemas de mais alto grau hierárquico até chegar à raiz, que representa o sistema completo sob estudo. Para tanto, utiliza os operadores *fuzzy* AND e OR, como citado anteriormente.

A realização probabilística requer procedimentos mais elaborados, no que tange às estimativas de probabilidades de falha das folhas (ou componentes elementares), tendo em vista que, a partir dos dados históricos de falha, devemos obter os parâmetros estatísticos de cada um dos componentes atômicos e as probabilidades de falha em um dado momento, a partir das funções de distribuição de probabilidades obtidas por tal procedimento.

Para fins de realização do presente trabalho, houve a opção pela adoção da distribuição exponencial para modelagem dos tempos de falha, correspondentes aos componentes elementares. Logo, a cada avaliação de confiabilidade, os dados de cada tipo de componente deverão ser recuperados, os parâmetros das distribuições exponenciais estimados segundo o princípio da plausibilidade máxima e as probabilidades de falha computadas, de acordo com a respectiva CDF exponencial. Devido à extrema flexibilidade estrutural da abordagem adotada, podemos facilmente adotar qualquer outra distribuição com mínimo esforço de manutenção, graças à abordagem predominantemente computacional do método MLE usando *Fuzzy* ASA - na abordagem convencional, seria necessário sintetizar novos algoritmos a cada mudança de tipo de distribuição, tendo em vista a utilização de métodos do Cálculo Diferencial e resolução das respectivas equações.

Vencida a etapa de cálculo dos graus ou probabilidades de falha de cada componente, os procedimentos são muito similares nos dois casos. A diferença básica é que, no caso possibilístico, usamos resultados da Lógica *Fuzzy* para obtenção de uniões e interseções de conjuntos *fuzzy*, e, no caso probabilístico, entram em cena as fórmulas para obtenção de união e interseção de eventos, citadas anteriormente.

#### 4.7. ARQUITETURA DE SOFTWARE UTILIZADA NA IMPLEMENTAÇÃO

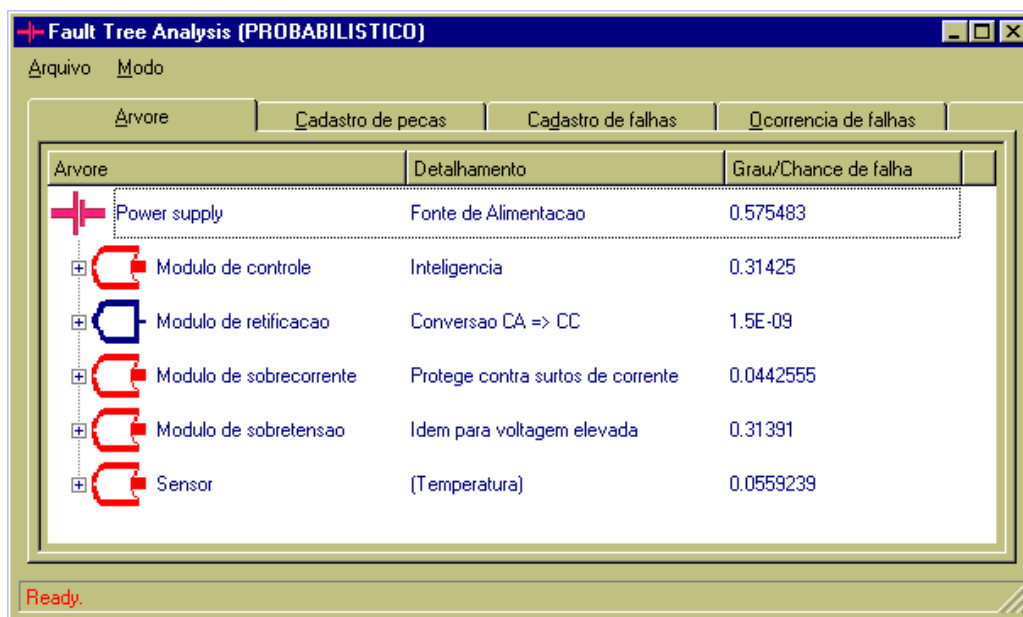
Na construção dos módulos de software, foi adotada a abordagem de orientação por objetos, tendo sido usada a linguagem C++ para sua efetiva programação. Para acelerar a construção do sistema, foram utilizadas algumas bibliotecas de classes especializadas em interfaces gráficas e manipulação de arquivos indexados. Em particular, a biblioteca de interface gráfica (FOX-TOOLKIT) apresenta alto grau de maturidade, tendo sido utilizada com sucesso em significativo número de aplicações dos mais variados tipos, além de apresentar relativa facilidade de utilização. O modelo de execução é ativado por eventos e, após a fase de inicialização, os vários *handlers* preestabelecidos são ativados quando da ocorrência de diversos estímulos como, por exemplo, acionamento de botões, recomposição de telas, etc. .

A operação do programa pode ser dividida em duas fases: inicialização e operação em regime. Na primeira fase ocorre a criação de inúmeros controles, ativação de gerenciadores de *layout* e inicialização das estruturas de dados que serão usadas no decorrer da execução propriamente dita. Na segunda fase, todo o aparato preestabelecido é exercitado por meio de eventos provocados tanto pelo operador quanto pelos processos internos.

## 5. RESULTADOS

Seguindo as diretrizes descritas anteriormente, foi implementado um programa que permite exercitar e validar o modelo sintetizado inicialmente. Tal artefato de software foi construído de modo a possibilitar a geração, em tempo de compilação, de dois módulos executáveis correspondentes aos casos possibilístico e probabilístico, bastando para isso ajustar um par de constantes, presentes no código fonte.

Conforme citado anteriormente, esta primeira versão realiza os dois tipos de árvores de falha (possibilística e probabilística) do modo mais geral, permitindo extensão hierárquica arbitrária, somente limitada por recursos computacionais particulares da máquina em uso. A título de exemplo, lançaremos mão de um sistema eletrônico simplificado, no qual foram escolhidos algumas componentes para figuração nas árvores realizadas. A seguir são apresentadas algumas telas reais com resultados obtidos pelo sistema em questão, nos casos probabilístico e possibilístico.













Árvore	Detalhamento	Grau/Chance de falha
Power supply	Fonte de Alimentacao	0.575483
Modulo de controle	Inteligencia	0.31425
Modulo de retificacao	Conversao CA => CC	1.5E-09
Modulo de sobrecorrente	Protege contra surtos de corrente	0.0442555
Modulo de sobretensao	Idem para voltagem elevada	0.31391
Sensor	(Temperatura)	0.0559239

**Fault Tree Analysis (PROBABILISTICO)**

Arquivo Modo

Arvore Cadastro de pecas Cadastro de falhas Correncia de falhas











Arvore	Detalhamento	Grau/Chance de falha
 Power supply	Fonte de Alimentacao	0.575483
 Modulo de controle	Inteligencia	0.31425
 Resistor 1	Precisao de 5%	0.09
 Resistor 2	Precisao de 5%	0.09
 Resistor 3	Precisao de 5%	0.09
 Capacitor 1	100 pF	0.09
 Modulo de retificacao	Conversao CA => CC	1.5E-09
 Modulo de sobrecorrente	Protege contra surtos de corrente	0.0442555
 Modulo de sobretensao	Idem para voltagem elevada	0.31391
 Sensor	(Temperatura)	0.0559239

Ready.

**Fault Tree Analysis (PROBABILISTICO)**

Arquivo Modo

Arvore Cadastro de pecas Cadastro de falhas Correncia de falhas

Arvore	Detalhamento	Grau/Chance de falha
 Power supply	Fonte de Alimentacao	0.575483
 Modulo de controle	Inteligencia	0.31425
 Resistor 1	Precisao de 5%	0.09
 Resistor 2	Precisao de 5%	0.09
 Resistor 3	Precisao de 5%	0.09
 Capacitor 1	100 pF	0.09
 Modulo de retificacao	Conversao CA => CC	1.5E-09
 Modulo de sobrecorrente	Protege contra surtos de corrente	0.0442555
 Modulo de sobretensao	Idem para voltagem elevada	0.31391
 Sensor	(Temperatura)	0.0559239

Ready.



**Fault Tree Analysis (PROBABILISTICO)**

Arquivo Modo

Arvore Cadastro de pecas Cadastro de falhas Ocorrencia de falhas

Arvore	Detalhamento	Grau/Chance de falha
Modulo de sobrecorrente	Protege contra surtos de corrente	0.0442555
Modulo de sobretensao	Idem para voltagem elevada	0.31391
Transistor T1		0.01
Transistor T2		0.01
Sensor 1	Circuito discreto	2.9701E-08
Transistor T3		0.01
Transistor T4		0.01
Transistor T5		0.01
Circuito experimental		0.029701
Sensor 2		0
Subsistema	Circuito auxiliar	0.29998
Sensor	(Temperatura)	0.0559239

Ready.

Agora, detalhes do caso possibilistico:

**Fault Tree Analysis (POSSIBILISTICO)**

Arquivo Modo

Arvore Cadastro de pecas Cadastro de falhas Ocorrencia de falhas

Arvore	Detalhamento	Grau/Chance de falha
Power supply	Fonte de Alimentacao	0.09
Modulo de controle	Inteligencia	0.09
Modulo de retificacao	Conversao CA => CC	0.001
Resistor 1	(Potencia)	0.01
Resistor 2	(Potencia)	0.01
Resistor 3	(Potencia)	0.001
Transformador		0.015
Modulo de sobrecorrente	Protege contra surtos de corrente	0.015
Modulo de sobretensao	Idem para voltagem elevada	0.019
Sensor	(Temperatura)	0.019

Ready.

**Fault Tree Analysis (POSSIBILISTICO)**

Arquivo Modo

Arvore Cadastro de pecas Cadastro de falhas Qcorrencia de falhas

Arvore	Detalhamento	Grau/Chance de falha	
+	Modulo de controle	Inteligencia	0.09
+	Modulo de retificacao	Conversao CA => CC	0.001
+	Modulo de sobrecorrente	Protege contra surtos de corrente	0.015
+	Modulo de sobretensao	Idem para voltagem elevada	0.019
-	Transistor T1		0.01
-	Transistor T2		0.01
+	Sensor 1	Circuito discreto	0.01
+	Sensor 2		0.007
+	Subsistema	Circuito auxiliar	0.019
+	Sensor	(Temperatura)	0.019

Ready.

**Fault Tree Analysis (POSSIBILISTICO)**

Arquivo Modo

Arvore Cadastro de pecas Cadastro de falhas Qcorrencia de falhas

Arvore	Detalhamento	Grau/Chance de falha	
+	Power supply	Fonte de Alimentacao	0.09
+	Modulo de controle	Inteligencia	0.09
+	Modulo de retificacao	Conversao CA => CC	0.001
+	Modulo de sobrecorrente	Protege contra surtos de corrente	0.015
+	Modulo de sobretensao	Idem para voltagem elevada	0.019
-	Sensor	(Temperatura)	0.019
-	Termistor 1		0.019
-	FET 1	Field Effect Transistor	0.019
-	Fotodiodo 1		0.019

Ready.

Embora a versão atual do programa já contenha diversos módulos de gerenciamento de tabelas com informações sobre os componentes, históricos de falhas, etc. , a codificação de tais funcionalidades não está no escopo da presente dissertação, tendo sido dada prioridade à conclusão da parte relativa ao cálculo dos parâmetros da árvore de falhas em si, a título de prova de conceito.

## 6. CONCLUSÃO

Conforme proposto, foi construída uma ferramenta de software para análise hierárquica de falhas, com abordagens probabilística e possibilística, além da validação off-line de método computacional para obtenção de parâmetros de funções densidade de probabilidade de distribuições que careçam dos requisitos analíticos mínimos para aplicação de métodos baseados em operadores gradiente e outros mecanismos do Cálculo Diferencial clássico. Tal validação foi desacoplada do objeto principal da pesquisa com o objetivo de demonstrar tanto sua utilidade geral como para não desviar o foco do tema central. De qualquer modo, os trabalhos de exploração de novas aplicações deverão prosseguir no futuro. Quanto ao protótipo apresentado, deverá ganhar novas funcionalidades que possam dotá-lo de maior abrangência e permitir sua aplicação a um maior espectro de problemas.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

ABONYI, Janos. **Fuzzy Model Identification for Control**. Birkhäuser, 2003. ISBN 0817642382.

BARHEN, J.; PROTOPOPESCU, V.; REISTER. D. TRUST - A Deterministic Algorithm for Global Optimization, **Science**, Vol. 276, p. 1094-1097, May 1997.

BOLCH, Gunter et al. **Queueing Networks and Markov Chains**. 1. ed. New York: John Wiley & Sons, 1998. ISBN 0471193666.

CASELLA, George; BERGER, Roger L. **Statistical Inference**. 2. ed. New York: Duxbury Press, 2001. ISBN 0534243126.

DRIANKOV, Dimiter; PALM Rainer. **Advances in fuzzy control**. New York: Physica-Verlag, 1998.

GREENE, William H. **Econometric Analysis**. 5. ed. New Jersey: Prentice-Hall, 2002. ISBN 0130661899.

INGBER, Lester. **Simulated Annealing: Practice versus Theory**. Disponível em: <[http://www.ingber.com/asa93\\_sapvt.pdf](http://www.ingber.com/asa93_sapvt.pdf)>. Acesso em: 13 jun. 2005.

INGBER, Lester. **Adaptive Simulated Annealing (ASA): Lessons Learned**. Disponível em: < [http://www.ingber.com/asa96\\_lessons.pdf](http://www.ingber.com/asa96_lessons.pdf)>. Acesso em: 13 jun. 2005.

INGBER, Lester. **Very Fast Simulated Re-Annealing**. Disponível em: < [http://www.ingber.com/asa89\\_vfsr.pdf](http://www.ingber.com/asa89_vfsr.pdf) >. Acesso em: 13 jun. 2005.

JAMSHIDI, Mohammad. **Applications of Fuzzy Logic – Towards High Machine Intelligence Quotient Systems**. New Jersey: Prentice-Hall PTR, 1997.

KLIR, George J.; YUAN, Bo. **Fuzzy sets and Fuzzy Logic – Theory and Applications**. Prentice-Hall, 1995

LEE M. A.; TAKAGI H. **Dynamic Control of Genetic Algorithms using Fuzzy Logic Techniques**. In: PROCEEDINGS OF THE 5TH INT’L CONFERENCE ON GENETIC ALGORITHMS (ICGA’ 93). Urbana Champaign,IL, USA, July 17-21,1993. p 76-83.

MACHADO, M. A. S. **Comunicação verbal**, 2005.

NASA OFFICE OF LOGIC DESIGNS. **The Team Approach of Fault-tree Analysis**. Disponível em: <[http://klabs.org/DEI/References/design\\_guidelines/analysis\\_series/1312msfc.pdf](http://klabs.org/DEI/References/design_guidelines/analysis_series/1312msfc.pdf)>. Acesso em 13 jun. 2005.

OLIVEIRA Jr., H. A. **Fuzzy Control of Stochastic Global Optimization Algorithms and Very Fast Simulated Reannealing**. Disponível em: < [http://www.optimization-online.org/DB\\_FILE/2003/11/779.pdf](http://www.optimization-online.org/DB_FILE/2003/11/779.pdf)>. Acesso em: 13 jun. 2005.

OLIVEIRA Jr., H. A. **Lógica Difusa – Aspectos Práticos e Aplicações**. Rio de Janeiro: Editora Interciência, 1999.

PAWITAN, Yudi. **In All Likelihood: Statistical Modelling and Inference Using Likelihood**. 1. ed. Oxford: Oxford University Press, 2001. ISBN 0198507658.

ROBERT, Christian. P.; CASELLA, George. **Monte Carlo Statistical Methods**. 2. ed. New York: Springer-Verlag, 2004. ISBN 0387212396.

ROSS, Timothy. **Fuzzy Logic with Engineering Applications**. New York: McGraw-Hill, 1995.

SEVERINI, Thomas A. **Likelihood Methods in Statistics**. 1. ed. Oxford: Oxford University Press, 2000. ISBN 0198506503.

SOUZA, João M. C.; KAYMAK, Uzay. **Fuzzy Decision Making in Modeling and Control**. World Scientific, 2002. ISBN 9810248776.

STROUSTRUP, Bjarne. **The C++ Programming Language**. 3. ed. Reading: Addison-Wesley, 1997. ISBN 0201889544.

U. S. COAST GUARD. **Risk-based Decision-making Guidelines**. Disponível em <<http://www.uscg.mil/hq/g-m/risk/e-guidelines/RBDM/html/vol3/09/v3-09-cont.htm>>. Acesso em: 13 jun. 2005.

VERDEGAY, José-Luis. **Fuzzy Sets Based Heuristics for Optimization**. Berlin:Springer Verlag, 2003. ISBN 354000551.

YAGER, Ronald R.; FILEV, Dimitar P. **Essentials of Fuzzy Modeling and Control**. NewYork: John Wiley & Sons, 1994.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)