

Contribuições ao Estudo de Programação Não-Linear com Incertezas

Ricardo Coêlho Silva
GRADUADO EM MATEMÁTICA – UERN

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como parte dos requisitos exigidos para a obtenção do título de **Mestre em Engenharia Elétrica**.

Orientador:
Prof. Dr. Akebo Yamakami
FEEC/UNICAMP

Co-Orientadora:
Prof. Dra. Luiza Amalia Pinto Cantão
UNESP/Campus de Sorocaba

Banca Examinadora

Prof. Dra. Marcia Tomie Takahashi	DEXA/UEFS
Prof. Dr. Anésio dos Santos Junior	FEEC/UNICAMP
Prof. Dr. Akebo Yamakami	FEEC/UNICAMP (presidente)

Maio de 2005
FEEC - UNICAMP

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Si38c	Silva, Ricardo Coêlho Contribuições ao estudo de programação não-linear com incertezas / Ricardo Coêlho Silva. – Campinas, SP: [s.n.], 2005. Orientador: Akebo Yamakami, Luiza Amália Pinto Cantão Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação 1. Conjuntos difusos. 2. Programação (Matemática). 3. Programação não-linear. 4. Algoritmos genéticos. I. Yamakami, Akebo. II. Luiza Amalia Pinto Cantão. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título
-------	---

Título em Inglês: Contributions to the study of nonlinear programming with uncertainties

Palavras-chave em Inglês: Theory fuzzy, Mathematical programming, Nonlinear programming e Genetic algorithms

Área de concentração: Matemática Aplicada e Pesquisa Operacional

Banca examinadora: Márcia Tomie Takahashi e Anésio dos Santos Junior.

Data da defesa: 25/05/2005

Resumo

Neste trabalho foram desenvolvidos alguns métodos iterativos e algoritmos meta-heurísticos, adaptados de modelos clássicos, que solucionam problemas de programação não-linear com parâmetros *fuzzy* na função objetivo e no conjunto de restrições. Apresentamos aqui uma relação entre alguns destes métodos iterativos e uma abordagem diferenciada das restrições de igualdade com parâmetros *fuzzy*. Comprovamos a eficiência dos algoritmos propostos comparando os seus resultados com os encontrados na literatura.

Abstract

In this work we develop some iterative methods and meta-heuristic algorithms that solve the nonlinear programming problems with uncertainties in the objective function and in the set of constraints. We derive a relation among some of this iterative methods and introduce a novel approach to the equality constraints with uncertainties. Selected examples from the literature are presented to validate the efficiency of the methods and algorithms addressed.

TODOS NÓS TEMOS PROBLEMAS,
PORÉM, O QUE NOS DIFERENCIA DOS
DEMAIS MORTAIS É COMO IREMOS
SOLUCIONÁ-LOS.

Anônimo

Agradecimentos

À minha família, meus pais (Antonio e Regina) e minha irmã (Daisy), que me permitiram descobrir o valor da vida e, mesmo com meu ponto-de-vista de vida conflitante aos deles, compreenderam os esforços feitos para atingir os meus objetivos.

Ao meu orientador, prof. Akebo que ultrapassa a barreira de um Orientador, pela oportunidade dada e creditado sua confiança em meu potencial para concluir este trabalho. A todos os professores que contribuíram para a minha formação, dentre os quais posso destacar: Ir. Eduardo e Ir. Aguiar pela amizade conquistada e o ensino básico nos moldes da ideologia Marista; João de Deus e Amorim, que me mostraram na graduação a beleza da matemática, em geral, quando demonstramos a sua importância na evolução do mundo.

Aos meus parentes, por me apoiarem e me ensinarem, mesmo cada um com sua maneira diferente de ver a vida, a buscar os meus desejos e vencer os desafios. Em especial ao meu avô, João Coelho (*in memoriam*), que prometi batalhar na busca dos meus sonhos e não desistir em alcançá-los independente das dificuldades encontradas. A tia Elba por me criar como um filho e aturar todas as minhas peripécias. A Maria e sua família, que cuidaram de mim desde criança e me mostraram que existem muitos caminhos na vida que um garoto mimado não percebe.

A todos os meus amigos que conquistei até este momento, os quais me proporcionam momentos importantes na minha vida e um excelente ambiente de trabalho, dos quais se destacam: Renato e Luiza pelas ajudas e diálogos que foram importantíssimo para a conclusão deste trabalho e para a minha caminhada particular; Fábio, Chicão e Bruno por estarem presentes em todos os momentos importantes nesta etapa de minha vida, formando assim o “trio parada dura”; Gelinho (*in memoriam*), pois mostrou a todos que o jeito brincalhão não prejudica a vida profissional, pelo contrário, deixa o ambiente de trabalho, ou estudo, muito mais descontraído e produtivo. Aos fundadores, membros e simpatizantes do grupo “Os Zero Hora”, que mostraram a força existente na união de pessoas para alcançar propósitos em comum a todos.

Finalmente, ao CNPq pelo apoio financeiro fornecido durante esta etapa de estudo e à realização deste trabalho (processo # 133041/2003-4).

Sumário

Resumo e Abstract	ii
Agradecimentos	iv
Lista de Figuras	vii
Lista de Tabelas	viii
Lista de Notações	ix
1 Introdução	1
2 Programação não-linear <i>fuzzy</i>	6
Introdução	6
2.1 Incertezas no conjunto de restrições	6
2.1.1 Programação não-linear <i>fuzzy</i> - Primeiro método	7
2.1.2 Programação não-linear <i>fuzzy</i> - Método duas fases	8
2.1.3 Relação entre os métodos	11
2.2 Parâmetros <i>fuzzy</i> na função objetivo	12
2.2.1 Programação não-linear irrestrita	13
2.2.1.1 Condições de otimalidade	13
2.2.1.2 Método de busca unidimensional com parâmetros <i>fuzzy</i>	14
2.2.1.3 Método de busca multidimensional com parâmetros <i>fuzzy</i>	16
2.2.2 Programação não-linear restrita	17
2.2.2.1 Condições de otimalidade	18
2.2.2.2 Métodos não-lineares restritos com parâmetros <i>fuzzy</i>	19
3 Programação matemática <i>fuzzy</i>	21
Introdução	21
3.1 Conjunto de restrições <i>fuzzy</i>	22
3.2 Métodos adaptados para programação matemática <i>fuzzy</i>	23
3.2.1 Método adaptado de Zimmermann	23
3.2.2 Método adaptado de Xu	25
3.2.3 Método com penalização na função objetivo	27
4 Meta-heurística <i>fuzzy</i>	28
Introdução	28
4.1 Critérios de decisão	29

4.1.1	Conjunto de restrições	29
4.1.2	Valor de <i>fitness</i>	30
4.1.3	Nível de satisfação	31
4.2	Algoritmos genéticos adaptados para programação matemática <i>fuzzy</i>	31
4.2.1	Algoritmo genético puro	32
4.2.1.1	Representação estrutural	32
4.2.1.2	Processo de inicialização	32
4.2.1.3	Função de avaliação	33
4.2.1.4	Processo de seleção	34
4.2.1.5	Operação de reprodução	34
4.2.1.6	Operação de mutação	35
4.2.1.7	Procedimento do algoritmo genético puro	36
4.2.2	Algoritmo genético com busca local	37
4.2.2.1	Operação de busca local	37
4.2.2.2	Procedimento do algoritmo genético com busca local	38
5	Resultados Numéricos	39
	Introdução	39
5.1	Formulação dos problemas	40
5.1.1	Incertezas no conjunto de restrições	40
5.1.1.1	Problema 1: Suporte de teto com três barras	40
5.1.1.2	Problema 2: Anteparo ondulado	41
5.1.1.3	Problema 3: Modelo de processo industrial	43
5.1.2	Incertezas na função objetivo e no conjunto de restrições	46
5.1.2.1	Problemas com restrições de desigualdade	46
5.1.2.2	Problemas com restrições de igualdade	47
5.1.2.3	Problemas com restrições de igualdade e desigualdade	47
5.2	Resultados e análise	47
5.2.1	Incertezas no conjunto de restrições	47
5.2.1.1	Problema 1: Suporte de teto com três barras	48
5.2.1.2	Problema 2: Anteparo ondulado	49
5.2.1.3	Problema 3: Modelo de processo industrial	50
5.2.2	Incertezas na função objetivo e no conjunto de restrições	52
6	Considerações Finais	55
6.1	Conclusões	55
6.2	Trabalhos futuros	56
	Bibliografia	58

Lista de Figuras

1.1	Função de Pertinência de $\mu_{\tilde{A}}(x)$	3
3.1	Função de pertinência da restrição de igualdade $h_l(\mathbf{x}) = c_l$	22
3.2	Função de pertinência da restrição de desigualdade $g_k(\mathbf{x}) \lesssim b_k$	22
3.3	Função de pertinência da restrição de desigualdade $g_k(\mathbf{x}) \gtrsim b_k$	22
4.1	Função de pertinência da restrição de igualdade transformada	30
5.1	Suporte de Teto com Três Barras	40
5.2	Visão lateral do anteparo ondulado	41
5.3	Especificação das variáveis do anteparo ondulado	42

Lista de Tabelas

5.1	Valores dos limites e tolerâncias para as restrições	40
5.2	Níveis de tolerância para as restrições	43
5.3	Descrição detalhada dos dados do problema 5.3	44
5.4	Níveis máximos de tolerância	46
5.5	Problemas com restrições somente de desigualdade	47
5.6	Problema com restrição somente de igualdade	47
5.7	Problemas com restrições de igualdade e desigualdade	48
5.8	Resultados da implementação do método da Seção 2.1.1	48
5.9	Resultados da implementação do método da Seção 2.1.2 - Primeira fase	49
5.10	Resultados da implementação do método da Seção 2.1.2 - Segunda fase	49
5.11	Resultados da implementação do método da Seção 2.1.1	50
5.12	Resultados da implementação do método da Seção 2.1.2 - Primeira fase	50
5.13	Resultados da implementação do método da Seção 2.1.2 - Segunda Fase	50
5.14	Resultados do exemplo numérico 5.3 com as restrições totalmente violadas . .	51
5.15	Resultado do exemplo numérico 5.3 dos modelos para comparação	51
5.16	Análise dos resultados do exemplo numérico 5.3	52
5.17	Níveis máximos e mínimos de tolerância para o conjunto de restrições	53
5.18	Resultado para o problema PG1	53
5.19	Resultado para o problema PG2	54
5.20	Resultado para o problema PG3	54
5.21	Resultado usando o algoritmo genético puro	54

Lista de notações

$\mathbb{R}$	Conjunto dos números reais.
$\mathbb{R}^n$	Espaço vetorial n -dimensional de componentes reais.
$\mathbf{x}$	Vetor de componentes reais.
$\sim$	Representa a incerteza inserida.
$\mathbb{F}(\mathbb{R})$	Conjunto dos números <i>fuzzy</i> .
$\mathbb{F}(\mathbb{R}^n)$	Conjunto de vetores n -dimensionais de componentes <i>fuzzy</i> .
$\tilde{\mathbf{x}}$	Vetor de componentes <i>fuzzy</i> .
$\text{Idf}(\tilde{a})$	Número defuzzyficado, que representa um número pertencente a $\mathbb{R}$.
$Q_\epsilon(x)$	Vizinhança esférica.
I	Conjunto das restrições de desigualdade ativas.
J	Fator de discretização da direção de busca <i>fuzzy</i> .
Γ	Valor do passo na direção aleatória do algoritmo genético.
n	Dimensão do problema a ser otimizado.

Capítulo 1

Introdução

O surgimento da pesquisa operacional (PO) forneceu uma base quantitativa e racional para as tomadas de decisão. A PO é um ramo da ciência que tem por finalidade desenvolver técnicas para otimizar o desempenho de sistemas. Suas aplicações encontram-se nas áreas industriais, de negócios, militares e governamentais, entre outras. O estudo da PO pode ser dividido em algumas sub-áreas, sendo a programação matemática (PM) uma delas. Essa tem como meta solucionar problemas que envolvem minimização (ou maximização) da função objetivo, onde estes podem ser do tipo irrestrito ou restrito. Os problemas de otimização com restrições podem ser de três naturezas diferentes: *(i)* restrições de igualdade; *(ii)* restrições de desigualdade e *(iii)* restrições mistas. Algumas teorias foram desenvolvidas para determinar uma solução ótima para o segundo tipo de problema, como por exemplo as condições de otimalidade de Fritz Jonh (FJ) desenvolvidas em 1948, porém estas teorias se restringiam a um número limitado de problemas. O surgimento das condições de otimalidade de Karush-Kunh-Tucker (KKT), desenvolvidas, de forma independente, pelos pesquisadores Karush, em 1939, e Kunh & Tucker, em 1951, adicionou requisitos às condições de FJ, descritos em (BAZARAA, SHERALI & SHETTY 1993, LUENBERGER 1989), permitindo resolver uma quantidade maior de problemas de programação matemática restritos.

Quando definimos problemas de programação matemática clássica (PMC), tanto as restrições quanto o objetivo devem descrever de forma precisa o problema a ser otimizado. Problemas convencionais são modelados na forma de PMC se estes têm definições matemáticas

breves e claras, da seguinte forma:

$$\begin{aligned}
 \min \quad & f(\mathbf{x}) \\
 \text{s.a} \quad & \mathbf{g}(\mathbf{x}) \leq \mathbf{b} \\
 & \mathbf{h}(\mathbf{x}) = \mathbf{c} \\
 & \mathbf{x} \in \Omega
 \end{aligned} \tag{1.1}$$

onde $\Omega \subset \mathbb{R}^n$, $\mathbf{g} : \mathbb{R}^k \times \mathbb{R}^n \rightarrow \mathbb{R}^k$ e $\mathbf{h} : \mathbb{R}^l \times \mathbb{R}^n \rightarrow \mathbb{R}^l$.

Quando um problema apresenta natureza vaga, com elementos de incerteza que podem ser modelados em termos matemáticos, então a PMC pode ser aplicado com análises de sensibilidade, paramétrica, ou estocástica.

Um grande avanço tecnológico e científico se deve a apresentação da teoria de conjuntos *fuzzy* idealizada por L. A. Zadeh, com a publicação do artigo “*Fuzzy Sets*” em 1965, veja (ZADEH 1965), introduzindo conceitos matemáticos de natureza vaga e ambígua. O surgimento desta teoria beneficiou o tratamento dos elementos de incerteza encontrados em várias áreas do conhecimento. Assim, essa teoria foi aplicada em diversos ramos das engenharias e da matemática, entre outros. Um conjunto *fuzzy*, descrito por BELLMAN & ZADEH (1970), pode ser definido como:

Definição 1.1 *Seja $X = \{x\}$ uma coleção de objetos (pontos) genericamente denotados por x . Assim, um conjunto fuzzy $\tilde{A}$ em X é um conjunto de pares ordenados:*

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \text{ tal que } x \in X\},$$

onde $\mu_{\tilde{A}}(x)$ é conhecida como FUNÇÃO DE PERTINÊNCIA de x em $\tilde{A}$, $\mu_{\tilde{A}} : X \rightarrow [0, 1]$ e $X \subseteq \mathbb{R}^n$.

A função $\mu_{\tilde{A}}(x)$ determina com que grau um objeto x pertence a um conjunto $\tilde{A}$. Em conjuntos clássicos, apenas duas situações para $\mu_{\tilde{A}}(x)$ são permitidas: o elemento pertence ou não pertence a um determinado conjunto. Essa particularidade classifica a função $\mu_{\tilde{A}}(x)$ como função característica. Em conjuntos *fuzzy*, os elementos pertencem ao conjunto com um valor real no intervalo $[0, 1]$. Esse valor é conhecido como GRAU DE PERTINÊNCIA, onde 0 exprime a completa exclusão do elemento no conjunto, enquanto que o valor 1 a pertinência completa.

Um conjunto *fuzzy* é definido pela função de pertinência $\mu_{\tilde{A}}(x)$, que estabelece para cada x um grau de pertinência ao conjunto $\tilde{A}$, com $\mu_{\tilde{A}} \in [0, 1]$. Assim, podemos expressar um

conjunto *fuzzy* como:

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x - \underline{a}}{a - \underline{a}} & x \in [\underline{a}, a] \\ \frac{\bar{a} - x}{\bar{a} - a} & x \in [a, \bar{a}] \\ 0 & \text{caso contrário,} \end{cases} \quad (1.2)$$

onde a é o valor modal, e $\underline{a}$ e $\bar{a}$ são os limitantes inferior e superior, respectivamente. A Figura 1.1 ilustra a função de pertinência descrita pela equação acima.

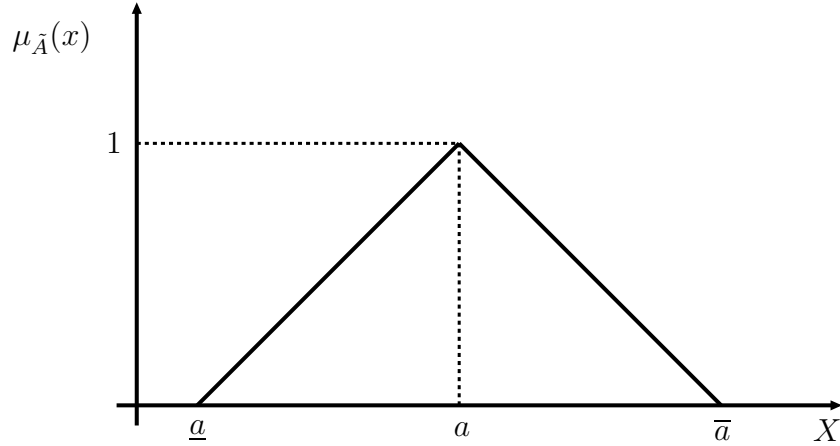


Figura 1.1: Função de Pertinência de $\mu_{\tilde{A}}(x)$.

A função de pertinência descrita pela Equação (1.2) e pela Figura 1.1 é conhecida como Função Triangular. Existem outros tipos de funções de pertinência, que podem ser: Função Trapezoidal, Função Gaussiana, Função tipo Exponencial, Γ -Função e S-Função. Essas funções são apresentadas por PEDRYCZ & GOMIDE (1998).

Além da função de pertinência ser normalmente utilizada para representar um conjunto *fuzzy*, existem outros parâmetros que podem caracterizar estes conjuntos:

- Suporte ($supp(\mu_{\tilde{A}})$): conjunto de elementos do universo para os quais o grau de pertinência é maior do que zero.

$$supp(\mu_{\tilde{A}}) = \{x \text{ tal que } \mu_{\tilde{A}}(x) > 0\};$$

- Altura ($htg(\tilde{A})$): valor máximo da função de pertinência, dado por

$$htg(\tilde{A}) = sup_x \mu_{\tilde{A}}(x);$$

- Conjunto normal: um conjunto *fuzzy* $\tilde{A}$ é normal se, e somente se, $htg(\tilde{A}) = 1$. Entretanto, se $htg(\tilde{A}) < 1$, então o conjunto é chamado de subnormal;

- União: $\mu_{(\tilde{A} \cup \tilde{B})}(x) = \max[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)], \quad \forall x \in X \subseteq \mathbb{R}^n;$
- Intersecção: $\mu_{(\tilde{A} \cap \tilde{B})}(x) = \min[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)], \quad \forall x \in X \subseteq \mathbb{R}^n;$
- Conjunto convexo: um conjunto *fuzzy* $\tilde{A}$ é convexo se

$$\mu_{\tilde{A}}(\lambda x_1 + (1 - \lambda)x_2) \geq \lambda \mu_{\tilde{A}}(x_1) + (1 - \lambda) \mu_{\tilde{A}}(x_2),$$

$$\forall x_1, x_2 \in \text{supp}(\mu_{\tilde{A}}) \text{ e } \lambda \in [0, 1];$$

- Conjunto quase-convexo: um conjunto *fuzzy* $\tilde{A}$ é quase-convexo se

$$\mu_{\tilde{A}}(\lambda x_1 + (1 - \lambda)x_2) \geq \min[\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)],$$

$$\forall x_1, x_2 \in \text{supp}(\mu_{\tilde{A}}) \text{ e } \lambda \in [0, 1];$$

- Função convexa: uma função *fuzzy* $f(\tilde{\mathbf{a}}; \mathbf{x}) : \mathbb{F}(\mathbb{R}^m) \times \mathbb{S} \rightarrow \mathbb{F}(\mathbb{R})$, sendo $\mathbb{S} \subseteq \mathbb{R}^n$ conjunto convexo não-vazio, é convexa em $\bar{\mathbf{x}} \in \mathbb{S}$ se:

$$f(\tilde{\mathbf{a}}; \lambda \bar{\mathbf{x}} + (1 - \lambda)\mathbf{x}) \lesssim \lambda f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) + (1 - \lambda)f(\tilde{\mathbf{a}}; \mathbf{x})$$

- Função quase-convexa: uma função *fuzzy* $f(\tilde{\mathbf{a}}; \mathbf{x}) : \mathbb{F}(\mathbb{R}^m) \times \mathbb{S} \rightarrow \mathbb{F}(\mathbb{R})$, sendo $\mathbb{S} \subseteq \mathbb{R}^n$ conjunto convexo não-vazio, é quase-convexa em $\bar{\mathbf{x}} \in \mathbb{S}$ se:

$$f(\tilde{\mathbf{a}}; \lambda \bar{\mathbf{x}} + (1 - \lambda)\mathbf{x}) \lesssim \max f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}), f(\tilde{\mathbf{a}}; \mathbf{x})$$

- Função pseudo-convexa: uma função *fuzzy* $f(\tilde{\mathbf{a}}; \mathbf{x}) : \mathbb{F}(\mathbb{R}^m) \times \mathbb{S} \rightarrow \mathbb{F}(\mathbb{R})$, sendo $\mathbb{S} \subseteq \mathbb{R}^n$ conjunto convexo não-vazio, é pseudoconvexa em $\bar{\mathbf{x}} \in \mathbb{S}$ se: $\nabla f(\tilde{\mathbf{a}}; \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \gtrsim \tilde{0}$ para cada $\mathbf{x} \in \mathbb{S}$, implica que $f(\tilde{\mathbf{a}}; \mathbf{x}) \gtrsim f(\tilde{\mathbf{a}}; \bar{\mathbf{x}})$.

- α -corte: o conjunto de α -cortes de um conjunto *fuzzy* $\tilde{A}$, denotado por $\tilde{A}_\alpha$, é definido como:

$$\tilde{A}_\alpha = \{x \in \text{supp}(\mu_{\tilde{A}}) \text{ tal que } \mu_{\tilde{A}}(x) \geq \alpha\}, \quad \forall \alpha \in [0, 1].$$

Em 1983, o pesquisador H. J. Zimmermann, vide (ZIMMERMANN 1983), introduziu incertezas na função objetivo e no conjunto de restrições de problemas de programação linear.

$$\begin{aligned} & \widetilde{\min} \quad \mathbf{c}^T \mathbf{x} \\ & s.a \quad \mathbf{A}\mathbf{x} \lesssim \mathbf{b} \\ & \quad \mathbf{x} \in \Omega \end{aligned} \tag{1.3}$$

onde $\Omega \subset \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$ e $A \in \mathbb{R}^m \times \mathbb{R}^n$. O símbolo $\sim$ explicita esta incerteza, que permite um nível de aspiração para a função objetivo e algumas violações na desigualdade do problema. O Capítulo 2 descreve como essas incertezas são tratadas.

Mais tarde, outros pesquisadores inseriram características *fuzzy* nas constantes da função objetivo e/ou nas restrições, no termo independente das restrições e nas variáveis de decisão do problema. Assim, podemos reescrever o Problema (1.3) de forma totalmente *fuzzy*:

$$\begin{aligned} \min \quad & \tilde{\mathbf{c}}^T \tilde{\mathbf{x}} \\ \text{s.a} \quad & \tilde{A}\tilde{\mathbf{x}} \lesssim \tilde{\mathbf{b}} \\ & \tilde{\mathbf{x}} \in \Omega \end{aligned} \tag{1.4}$$

onde $\Omega \subset \mathbb{F}(\mathbb{R}^n)$, $\mathbf{b} \in \mathbb{F}(\mathbb{R}^m)$, $A \in \mathbb{F}(\mathbb{R}^m) \times \mathbb{F}(\mathbb{R}^n)$.

Porém, poucos estudos se intensificaram na área de programação não-linear. Nessa área se destacam os trabalhos de TRAPPEY, LIU & CHANG (1988) e XU (1989), que abordam problemas com incertezas nas desigualdade do conjunto de restrições, e LIU & IWAMURA (1998a) e LIU & IWAMURA (1998b), que inserem incertezas nas constantes da função objetivo e das funções do conjunto de restrições.

Neste trabalho utilizamos (KAUFMANN & GUPTA 1984, KLIR & FOLGER 1998, PEDRYCZ & GOMIDE 1998) como literatura base na teoria de conjuntos *fuzzy*. As referências (BAZARAA et al. 1993, CANTÃO 2003, FERREIRA 2004, LUENBERGER 1989) contêm informações de como tratar problemas de programação não-linear e como identificar os critérios de otimalidade apropriados para a resolução destes problemas clássicos ou *fuzzy*.

Esta dissertação está organizada da seguinte forma: No Capítulo 2 é apresentado um resumo dos estudos realizados em problemas de programação não-linear *fuzzy* descritos em (CANTÃO 2003, LEE, YANG & MOON 1999, TRAPPEY et al. 1988, XU 1989). O Capítulo 3 descreve métodos adaptados de modelos iterativos clássicos que resolvem problemas com incerteza, tanto nas constantes da função objetivo quanto no conjunto de restrições, usando a derivada como principal direção ótima. As adaptações de algoritmos genéticos estão discriminadas no Capítulo 4, as quais solucionam problemas de programação não-linear com incertezas descritos no capítulo anterior. No Capítulo 5 estão os problemas testados nesta dissertação, junto com os resultados computacionais dos mesmos e alguns comentários sobre os resultados obtidos. Finalmente, no Capítulo 6 estão expressas as considerações finais sobre o trabalho apresentado nesta dissertação e alguns pontos interessantes para trabalhos futuros.

Capítulo 2

Programação não-linear *fuzzy*

Introdução

Este capítulo apresenta métodos iterativos desenvolvidos para solucionar problemas de programação não-linear *fuzzy*. As características *fuzzy* podem aparecer de várias formas no problema, podendo o mesmo ser totalmente *fuzzy*. Neste trabalho nos aprofundamos em incertezas introduzidas no conjunto de restrições, na função objetivo ou em ambos os casos (este último caso será explanado nos Capítulos 3 e 4).

A Seção 2.1 descreve problemas com incertezas inseridas no conjunto de restrições, para os quais a literatura apresenta dois métodos de resolução. Em (LEE et al. 1999, TRAPPEY et al. 1988, XU 1989) estão definidos os métodos mencionados neste parágrafo, entre os quais existe uma forte relação demonstrada nesta seção.

A Seção 2.2 apresenta problemas em que os coeficientes da função objetivo são dados por números incertos. Neste caso, os problemas de programação não-linear podem ser divididos em IRRESTRITOS e RESTRITOS, e as técnicas de resolução são adaptadas dos métodos de programação não-linear clássica, desenvolvidos por CANTÃO (2003).

2.1 Incertezas no conjunto de restrições

A Sub-seção 2.1.1 apresenta o primeiro método de programação não-linear *fuzzy*, encontrado em (LEE et al. 1999, TRAPPEY et al. 1988), que originou-se do método de programação linear *fuzzy* idealizado por ZIMMERMANN (1983).

A Sub-seção 2.1.2 descreve o segundo método, desenvolvido em (XU 1989), o qual utiliza duas fases para obter a solução ótima.

Analisando estes dois métodos foi encontrada uma relação entre os mesmos, cuja demonstração está apresentada na Sub-seção 2.1.3. Esta relação permite verificar que ambos os métodos obtêm a mesma solução ótima.

2.1.1 Programação não-linear *fuzzy* - Primeiro método

Trataremos aqui do método que transforma um problema de programação não-linear *fuzzy* (PPNLF) em um Problema de Programação Não-Linear Clássico (PPNLC). Esse método transforma a função objetivo em uma restrição, e a nova função objetivo tenta maximizar o nível de compatibilidade no espaço de factibilidade. Este nível de compatibilidade está determinado no intervalo real $[0,1]$, e diante de seu valor saberemos o quanto foram satisfeitas as restrições.

Um problema de programação não-linear com uma função objetivo *fuzzy* e restrições *fuzzy* pode ser escrito genericamente utilizando como base o Problema (1.1), da seguinte forma:

$$\begin{aligned} \widetilde{\min} \quad & g_0(\mathbf{x}) \\ \text{s.a} \quad & g_i(\mathbf{x}) \lesssim b_i, \quad i = 1, \dots, m \\ & x \geq 0, \end{aligned} \tag{2.1}$$

onde $\widetilde{\min}$ e $\lesssim$ representam as características *fuzzy* do problema. A formulação sem $\sim$ é um PPNLC. A função objetivo com sinal de *fuzzy* pode ser interpretada como:

$$g_0(\mathbf{x}) \lesssim b_0$$

onde b_0 é o nível de aspiração do decisor. Desta forma, pode-se acrescentar uma variável t_i que indica o nível de violação das restrições *fuzzy* e o nível de aspiração do problema (2.1), o qual pode ser reescrito na forma:

$$\begin{aligned} g_j(\mathbf{x}) &\leq b_j + t_j, \quad j = 0, \dots, m \\ \mathbf{x} &\geq \mathbf{0} \\ 0 &\leq t_j \leq T_j \end{aligned}$$

onde $t_j = [t_0, \dots, t_m]$ representa a violação da restrição j e $T_j = [T_0, \dots, T_m]$ o valor máximo de violação admissível para restrição j .

A função de pertinência $\mu_j(g_j(\mathbf{x}))$ representa o grau de satisfação que a solução $\mathbf{x}$ fornece para a restrição $g_j(\mathbf{x})$, que pode ser descrita como:

$$\mu_j(g_j(\mathbf{x})) = \begin{cases} 0, & \text{se } t_j > T_j \\ 1 - \frac{t_j}{T_j}, & \text{se } 0 \leq t_j \leq T_j \\ 1, & \text{se } t_j < 0 \end{cases}$$

No intuito de agregarmos todas as funções de pertinência μ_j , tomamos a intersecção dessas funções como:

$$\bigcap_{0 \leq j \leq m} \bigcap_{\mathbf{x}} \mu_j(g_j(\mathbf{x})) \in [0, 1]$$

$$0 \leq t_j \leq T_j$$

onde $\bigcap$ é o símbolo de intersecção.

Usando a regra de minimização para a agregação de múltiplos conjuntos *fuzzy*, a decisão satisfatória pode ser estabelecida dentro de cada restrição. Então, o novo objetivo consiste em tentar maximizar o grau de satisfação de todas as restrições. Fazendo $S = \min_{0 \leq j \leq m} \mu_j(g_j(\mathbf{x}))$, obtemos

$$S \leq 1 - \frac{t_j}{T_j}$$

$$T_j S \leq T_j - t_j$$

$$T_j S + t_j \leq T_j.$$

Logo, o modelo de PPNLF pode ser formulado como um PPNLC:

$$\begin{aligned} \max \quad & S \\ \text{s.a} \quad & (a) \quad T_j S + t_j \leq T_j \\ & (b) \quad g_j(\mathbf{x}) - t_j \leq b_j \\ & (c) \quad t_j \leq T_j \\ & S, \mathbf{x}, t_j \geq 0, \quad j = 0, \dots, m \end{aligned} \tag{2.2}$$

Nota-se que a função objetivo S e as restrições (a) e (c) são lineares. A restrição (b) pode ser linear ou não-linear.

2.1.2 Programação não-linear *fuzzy* - Método duas fases

Nesta sub-seção é abordado o conceito do método duas fases, desenvolvido por XU (1989). Na primeira fase, usa-se o método de níveis de α -cortes, transformando o PPNLF em um

PPNLC para determinar o conjunto de soluções para um problema parametrizado. Na segunda fase é aplicado o método de busca nos limites, utilizando o supremo e o ínfimo do conjunto de soluções factíveis da fase anterior, obtendo a solução ótima *fuzzy* no intervalo de confiança do espaço factível.

O método duas fases começa transformando um Problema *fuzzy* em um problema clássico equivalente, considerando o problema (2.1) como:

$$\begin{aligned} & \widetilde{\min} \quad g_0(\mathbf{x}) \\ & s.a \quad g_i(\mathbf{x}) \leq b_i + t_i, \quad i = 1, \dots, m \\ & \quad \quad x \geq 0 \end{aligned} \tag{2.3}$$

Contudo, utilizando uma função de pertinência $\mu_i(\mathbf{x}) : \mathbb{R}^n \rightarrow [0, 1]$, $\forall i = 1, \dots, m$, teremos diferentes graus de satisfação dentro do intervalo unitário $[0, 1]$, da seguinte forma:

$$\mu_i(x) = \begin{cases} 1, & g_i(\mathbf{x}) \leq b_i \\ \frac{(b_i + T_i) - g_i(\mathbf{x})}{T_i}, & b_i < g_i(\mathbf{x}) < b_i + T_i \\ 0, & g_i(\mathbf{x}) \geq b_i + T_i \end{cases} \tag{2.4}$$

Para tanto, é preciso parametrizar as restrições usando os níveis α -cortes propostos na teoria de conjuntos *fuzzy* (ZADEH 1965), tal que:

$$C_\alpha = \{\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n, \mu_c(\mathbf{x}) \geq \alpha\}, \quad \forall \alpha \in [0, 1]. \tag{2.5}$$

Depois de realizada a transformação e os cortes descritos pela Equação (2.5), faz-se a intersecção de todas as restrições com um operador de agregação, da seguinte forma:

$$\mu_C(\mathbf{x}) = \min_{i=1}^m \mu_i(\mathbf{x}), \quad \forall x \in \mathbb{R}^n \tag{2.6}$$

Logo, utiliza-se o mínimo em (2.6) como operador de agregação.

Por fim, consegue-se escrever o PPNLF em um PPNLC correspondente. Entretanto, existem vários valores da nossa função objetivo, dependendo do valor parametrizado de α . Portanto, o Problema (2.3) é reescrito como

$$\begin{aligned} & \widetilde{\min} \quad g_0(\mathbf{x}) \\ & s.a \quad g_i(\mathbf{x}) \leq b_i + T_i(1 - \alpha), \\ & \quad \quad \alpha \in [0, 1] \quad i = 1, \dots, m \end{aligned} \tag{2.7}$$

A solução do Problema (2.7) resulta em um conjunto de soluções, uma para cada valor de α diferente, e assim encerra-se a primeira fase do método.

A segunda fase consiste em fazer a intersecção das funções de pertinência da função objetivo com a intersecção de todas as restrições da fase anterior. Assim,

$$\mu_D = \mu_C \bigcap \mu_G,$$

onde $\mu_D, \mu_C, \mu_G : \mathbb{R}^n \rightarrow [0, 1]$.

A decisão ótima no espaço factível é obtida com a função de pertinência

$$\mu_D(\mathbf{x}^*) = \max_{\mathbf{x} \in \mathbb{R}^n} \mu_D(\mathbf{x}). \quad (2.8)$$

Este método de resolução é proposto por Bellman e Zadeh (BELLMAN & ZADEH 1970).

Pode-se obter da Equação (2.8) um nível ótimo α^* e o ponto ótimo x^* tal que

$$\mu_G(\mathbf{x}^*) = \max_{\mathbf{x} \in C_{\alpha^*}} \mu_G(\mathbf{x}), \quad (2.9)$$

onde C_{α^*} é o nível α^* -corte do conjunto de restrições *fuzzy* C .

A solução *fuzzy* é dada dentro de um intervalo limitado pelos valores superior e inferior fornecidos pela função $\mu_G(\mathbf{x})$, sendo

$$\begin{aligned} m &= g_0(\mathbf{x}^*(0)) = \min_{\mathbf{x} \in C_0} g_0(\mathbf{x}) \\ M &= g_0(\mathbf{x}^*(1)) = \min_{\mathbf{x} \in C_1} g_0(\mathbf{x}), \end{aligned}$$

onde, C_0, C_1 são os níveis de corte para α igual a 0 e 1, respectivamente, do conjunto de restrições *fuzzy* C .

Em alguns problemas de otimização *fuzzy*, para achar mínimos de uma função objetivo sujeita a determinadas restrições, pode-se estabelecer a seguinte função:

$$\mu_G(\mathbf{x}) = \frac{m}{g_0(\mathbf{x})}. \quad (2.10)$$

Claramente, os limites superior e inferior da função objetivo são dados por:

$$\begin{aligned} \mu_G^u &= 1 \\ \mu_G^l &= \frac{m}{M}. \end{aligned}$$

Substituindo a Equação (2.10) na Equação (2.9), obtemos:

$$\mu_G(\mathbf{x}^*) = \frac{1}{m} \min_{\mathbf{x} \in C_{\alpha^*}} g_0(\mathbf{x}) \quad (2.11)$$

Desta maneira, a metodologia empregada tem capacidade de otimizar um problema *fuzzy* pelo método de busca nos limites do conjunto de soluções factíveis.

2.1.3 Relação entre os métodos

Nesta sub-seção é apresentada a relação entre os métodos descritos nas sub-seções 2.1.1 e 2.1.2. O desenvolvimento da demonstração abaixo mostra que as soluções obtidas pelos métodos acima convergem para uma única solução, mediante um intervalo definido chamado de nível de satisfação.

Lema 2.1 *Sejam b_0 , T_0 parâmetros aleatórios do primeiro método, e m , M parâmetros obtidos na resolução do segundo método. Então quando $b_0 = m$ e $T_0 = M$, os limitantes inferior e superior de $\mu_0(g_0(\mathbf{x}))$ são os mesmos de $\mu_G(\mathbf{x})$, $\forall \mathbf{x}$.*

Prova: Reescrevendo a equação (2.2), podemos expressar a restrição (a) como $t_j \leq T_j(1 - S)$, $\forall j = 0, 1, \dots, m$. Assim, substituindo na restrição (b), teremos $g_j(\mathbf{x}) - T_j(1 - S) \leq b_j \Rightarrow g_j(\mathbf{x}) \leq b_j + T_j(1 - S)$. Desta forma, a restrição (c) fica inoperante, pois o conjunto de restrições não depende mais de t_i , $\forall j$. Logo, a nova formulação fica

$$\begin{aligned} \max \quad & S \\ \text{s.a} \quad & g_j(\mathbf{x}) \leq b_j + T_j(1 - S) \\ & S \geq 0, \quad \mathbf{x} \geq \mathbf{0}, \quad j = 0, 1, \dots, m \end{aligned} \quad (2.12)$$

Atualizando as funções de pertinência de cada restrição, temos

$$\mu_j(g_j(\mathbf{x})) = \begin{cases} 1, & g_j(\mathbf{x}) \leq b_j \\ \frac{(b_j + T_j) - g_j(\mathbf{x})}{T_j}, & b_j < g_j(\mathbf{x}) < b_j + T_j \\ 0, & g_j(\mathbf{x}) \geq b_j + T_j \end{cases}$$

Sendo $j = 0$, $b_0 = m$, $T_0 = M$, temos

$$\mu_0(g_0(\mathbf{x})) = \begin{cases} 1, & g_0(\mathbf{x}) \leq m \\ \frac{(m + M) - g_0(\mathbf{x})}{M}, & m < g_0(\mathbf{x}) < m + M \\ 0, & g_0(\mathbf{x}) \geq m + M \end{cases}$$

Como $g_0(\mathbf{x}) = M$ no caso clássico, e este é o maior valor que $g_0(\mathbf{x})$ pode assumir, $\forall \mathbf{x} \geq \mathbf{0}$, então

$$\mu_0(g_0(\mathbf{x})) = \frac{(m + M) - g_0(\mathbf{x})}{M}, \quad m \leq g_0(\mathbf{x}) \leq M$$

Portanto, no máximo $\mu_0(g_0(\mathbf{x})) = 1$ e no mínimo $\mu_0(g_0(\mathbf{x})) = \frac{m}{M}$. ■

Teorema 2.1 *Sejam as condições do Lema 2.1 satisfeitas, então, para ambos os métodos, no ponto ótimo $S = \mu_G \cap \alpha$.*

Prova: (1) Se $S = 1$, temos $\min_{0 \leq j \leq m} \mu_j(g_j(\mathbf{x})) = 1$. Logo, $\exists \in \mathbb{R}^n$ tal que $\mu_j(g_j(\mathbf{x})) = 1, \forall j$. Se $\mu_D = 1$, temos $\mu_G \cap \mu_C = 1$. Assim, $\mu_G = 1$ e $\mu_C = 1$, para algum $x \in \mathbb{R}^n$. Logo, $\mu_i(g_i(\mathbf{x})) = 1, \forall i$, então $g_i(\mathbf{x}) = b_i \Rightarrow \alpha = 1$.

(2) Se $S = 0$, temos $\min_{0 \leq j \leq m} \mu_j(g_j(\mathbf{x})) = 0$. Logo, $\exists x \in \mathbb{R}^n$ tal que $\mu_j(g_j(\mathbf{x})) = 0$, para algum j . Logo, pelo Lema 2.1, temos $\mu_0(g_0(\mathbf{x})) = 0$, se, e somente se, $m = 0$. Se $\mu_D = 0$, temos $\mu_G \cap \mu_C = 0$. Assim, $\mu_G = 0$ ou $\mu_C = 0$, para algum $x \in \mathbb{R}^n$. No primeiro caso $\mu_G = 0$, se, e somente se, $m = 0$. No segundo caso $\mu_C = 0 \Rightarrow \mu_i(g_i(\mathbf{x})) = 0$, para algum i , então $g_i(\mathbf{x}) = b_i + T_i(1 - \alpha) \Rightarrow \alpha = 0$.

(3) Seja $S = \lambda, \lambda \in (0, 1)$. Então $\min_{0 \leq j \leq m} \mu_j(g_j(\mathbf{x})) = \lambda$. Logo, $\exists x \in \mathbb{R}^n$ tal que $\mu_j(g_j(\mathbf{x})) = \lambda$, para algum j . Logo, pelo Lema 2.1, temos $\mu_0(g_0(\mathbf{x})) = \lambda$, se, e somente se $\lambda \in [\frac{m}{M}, 1]$. Seja $\mu_D = \lambda$, então $\mu_G \cap \mu_C = \lambda$. Assim, $\mu_G = \lambda$ ou $\mu_C = \lambda$, para algum $x \in \mathbb{R}^n$. No primeiro caso $\mu_G = \lambda$, se, e somente se $\lambda \in [\frac{m}{M}, 1]$. No segundo caso $\mu_C = \lambda \Rightarrow \mu_i(g_i(\mathbf{x})) = \lambda$, para algum i , então

$$\begin{cases} g_i(\mathbf{x}) > b_i & \Rightarrow \alpha < 1 \\ g_i(\mathbf{x}) < b_i + T_i & \Rightarrow \alpha > 0 \end{cases}$$

Portanto, $S = \mu_G \cap \alpha$. ■

Os resultados computacionais usando os conceitos demonstrados nessa sub-seção estão apresentados na Seção 5.2 do Capítulo 5. Os problemas utilizados para confirmar essa relação estão formulados na Seção 5.1 do mesmo capítulo.

2.2 Parâmetros fuzzy na função objetivo

A Sub-seção 2.2.1 apresenta os métodos clássicos de problemas de programação não-linear irrestritos adaptados para obtenção da solução de um PPNLF. Tais métodos foram desenvolvidos modificando os métodos iterativos clássicos.

Os métodos adaptados que solucionam problemas de programação não-linear restritos são apresentados na Sub-seção 2.2.2. Estes métodos também foram desenvolvidos através de modificações realizadas nos métodos iterativos que resolvem problemas clássicos.

Nesta seção, denotamos $Q_\epsilon(x)$ o conjunto de pontos que pertencem a vizinhança de $\bar{x}$ com raio ϵ , ou seja, $Q_\epsilon(x) = \{x \text{ tal que } \|x - \bar{x}\| < \epsilon, \forall x \in \mathbb{R}^n\}$.

2.2.1 Programação não-linear irrestrita

Um problema de otimização irrestrita pode ser formulado da seguinte maneira:

$$\begin{aligned} \min \quad & f(\tilde{\mathbf{a}}; \mathbf{x}) \\ & \mathbf{x} \in \Omega \end{aligned} \tag{2.13}$$

onde $f(\tilde{\mathbf{a}}; \mathbf{x}) : \mathbb{F}(\mathbb{R}^m)\mathbb{R}^n \rightarrow \mathbb{R}$ e $\Omega \subseteq \mathbb{R}^n$.

Os métodos de otimização irrestrita podem ser divididos em duas classes, o UNIDIMENSIONAL e o MULTIDIMENSIONAL. Estes métodos podem ser resolvidos usando buscas sem ou com o uso de derivadas, através dos quais a otimalidade da solução obtida decorre da verificação de condições de otimalidade.

A teoria descrita nesta sub-seção apresenta as condições de otimalidade para as duas classes de métodos descritas acima, quando os parâmetros da função objetivo são números *fuzzy*.

2.2.1.1 Condições de otimalidade

As CONDIÇÕES NECESSÁRIAS E SUFICIENTES DE OTIMALIDADE foram baseadas em (BAZARAA et al. 1993, LUENBERGER 1989), as quais foram estendidas para problemas com parâmetros *fuzzy* na função objetivo. A comparação em x será clássica, ao passo que as comparações baseadas nas funções objetivos serão *fuzzy*, para as quais serão utilizados os índices de ordenação de Yager.

Dado um ponto $x \in \mathbb{R}^n$, pretendemos caracterizá-los como um ponto de mínimo, supondo que a função *fuzzy* associada seja diferenciável. O Corolário 2.1 do Teorema 2.2 nos fornece as CONDIÇÕES DE PRIMEIRA ORDEM para $\bar{x}$ ser ótimo local e o Teorema 2.3 as CONDIÇÕES DE SEGUNDA ORDEM usando a matriz Hessiana.

Teorema 2.2 (Condições Necessárias de Primeira Ordem) *Suponha que $f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) : \mathbb{F}(\mathbb{R}^m) \times \mathbb{R}^n \rightarrow \mathbb{F}(\mathbb{R})$, $\tilde{\mathbf{a}} \in \mathbb{F}(\mathbb{R}^m)$ e $\bar{\mathbf{x}} \in \mathbb{R}^n$, seja diferenciável em $\bar{\mathbf{x}}$. Se existe um vetor $\mathbf{d} \in \mathbb{R}^n$ tal que $\nabla f(\tilde{\mathbf{a}}; \bar{\mathbf{x}})^T \mathbf{d} \lesssim \tilde{0}$, então existe um $\delta > 0$ tal que $f(\tilde{\mathbf{a}}; \bar{\mathbf{x}} + \lambda \mathbf{d}) \lesssim f(\tilde{\mathbf{a}}; \bar{\mathbf{x}})$ para cada $\lambda \in (0, \delta)$, onde $\mathbf{d}$ é uma DIREÇÃO DE DESCIDA de $f(\tilde{\mathbf{a}}; \bar{\mathbf{x}})$.*

Prova: vide (CANTÃO 2003) ■

Corolário 2.1 *Suponha que $f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) : \mathbb{F}(\mathbb{R}^m) \times \mathbb{R}^n \rightarrow \mathbb{F}(\mathbb{R})$, $\tilde{\mathbf{a}} \in \mathbb{F}(\mathbb{R}^m)$ e $\bar{\mathbf{x}} \in \mathbb{R}^n$, seja diferenciável em $\bar{\mathbf{x}}$. Se $\bar{\mathbf{x}}$ é um mínimo local, então $\nabla f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) \simeq \tilde{\mathbf{0}}$.*

Prova: vide (CANTÃO 2003) ■

Teorema 2.3 (Condições Necessárias de Segunda Ordem) *Suponha que $f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) : \mathbb{F}(\mathbb{R}^m) \times \mathbb{R}^n \rightarrow \mathbb{F}(\mathbb{R})$, $\tilde{\mathbf{a}} \in \mathbb{F}(\mathbb{R}^m)$ e $\bar{\mathbf{x}} \in \mathbb{R}^n$, seja duas vezes diferenciável em $\bar{\mathbf{x}}$. Se $\bar{\mathbf{x}}$ é um ponto de mínimo local, então $\nabla f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) \simeq \tilde{\mathbf{0}}$ e $\nabla^2 f(\tilde{\mathbf{a}}; \bar{\mathbf{x}})$ é uma matriz semi-definida positiva com parâmetros fuzzy.*

Prova: vide (CANTÃO 2003) ■

As condições de otimalidade discutidas acima são condições necessárias. Por outro lado, um ponto satisfazendo estas condições não é necessariamente um mínimo local. O Teorema 2.4 expressa uma condição suficiente para um mínimo local.

Teorema 2.4 (Condições Suficientes de Segunda Ordem) *Seja $f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) : \mathbb{F}(\mathbb{R}^m) \times \mathbb{R}^n \rightarrow \mathbb{R}$ uma função PSEUDOCONVEXA FUZZY em $\bar{\mathbf{x}}$. Então, $\bar{\mathbf{x}}$ é um ponto de mínimo local se, e somente se, $\nabla f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) \simeq \tilde{\mathbf{0}}$.*

Prova: vide (CANTÃO 2003) ■

2.2.1.2 Método de busca unidimensional com parâmetros fuzzy

Apresentamos aqui alguns métodos de minimização de funções unidimensionais. Estes métodos estão divididos entre buscas sem o uso de derivadas, tais como busca dicotômica, seção áurea e Fibonacci, e buscas com o uso de derivadas, tais como bissecção, Newton, falsa posição e interpolação cúbica. Estes métodos são apresentados por CANTÃO (2003) com todos os detalhes sobre os seus conceitos e explicações, e em pseudo-código dos mesmos.

Métodos de busca sem uso de derivadas: Os métodos que não usam derivadas para minimizar uma função f de uma variável sobre um intervalo limitado e fechado resolvem estes problemas estreitando o intervalo de incerteza. O intervalo de incerteza é definido por $[a, b]$, com $a < b$, que contém o ponto ótimo da função a ser minimizada, o qual é desconhecido.

No método de busca dicotômica, divide-se o intervalo de incerteza em duas partes iguais. Centrando no ponto médio, traçamos uma circunferência de raio δ , o qual determina dois pontos que, depois de avaliados na função a ser minimizada geram o novo intervalo de incerteza. Este procedimento é realizado enquanto o tamanho do intervalo de incerteza for maior que a tolerância determinada.

O método da seção áurea utiliza um valor como razão que divide o intervalo de incerteza em três partes. Avalia-se os dois pontos calculados acima na função a ser minimizada e, em seguida, é determinado o novo intervalo, em que um dos pontos calculados torna-se um limitante do novo intervalo e o outro ponto passa a ser um dos divisores deste intervalo de incerteza, restando então calcular o outro divisor e avaliá-lo. Ao atingir a tolerância, o ponto ótimo será o ponto médio do último intervalo de incerteza.

O método de Fibonacci gera uma seqüência, baseada na seqüência de Fibonacci, usando os limitantes do intervalo. A partir desta seqüência gera-se uma razão pela qual divide-se o intervalo de incerteza em três partes. Deste ponto em diante, este método usa os mesmos conceitos do método da seção áurea.

Métodos de busca com uso de derivadas: Os métodos que usam derivadas para minimizar uma função unidimensional determinam direções que apontam para o ponto ótimo do problema. Estas direções são determinadas pela avaliação de um ponto baseada na derivada primeira da função a ser minimizada. Estes métodos podem precisar ou não do intervalo de incerteza.

O método de bissecção divide o intervalo de incerteza em duas partes iguais. Avalia-se o ponto médio do intervalo baseado na derivada primeira da função a ser minimizada, que é um número *fuzzy*. Se o valor “defuzzyficado”, em módulo, for menor que o critério de parada definida, o método termina. Caso contrário, o ponto médio transforma-se em limitante do intervalo, e o método continua enquanto o critério de parada não for satisfeito ou o tamanho do intervalo de incerteza for maior que a tolerância definida.

No método de Newton utiliza-se as derivadas de primeira e segunda ordem da função com parâmetros *fuzzy* a ser minimizada. Este método está baseado na exploração de uma aproximação quadrática. A motivação deste método está em atingir uma região em que a derivada segunda seja definida positiva, fornecendo uma boa convergência. Este método termina quando o módulo da diferença entre pontos de iterações for menor que o critério de

parada determinado.

O método da falsa posição usa uma aproximação quadrática parecida com a usada pelo método de Newton. A principal diferença entre estes dois métodos está na utilização da derivada primeira da função, enquanto o método de Newton usa a derivada de segunda ordem.

No método de interpolação cúbica necessita-se da derivada primeira da função e de um intervalo de incerteza, os quais permitem interpolar uma função cúbica *fuzzy*. Uma ferramenta importante para acelerar a convergência deste método baseia-se numa busca linear, quando pode ser aplicada a **Condição de Wolfe**.

2.2.1.3 Método de busca multidimensional com parâmetros *fuzzy*

Os métodos apresentados acima evoluem para os métodos multidimensionais. Estes métodos também são divididos em buscas sem o uso de derivadas, tal como o método de Rosenbrock, e buscas com uso de derivadas, tais como a máxima descida, Newton e gradiente conjugado. Pode-se encontrar todos os detalhes destes métodos em (CANTÃO 2003).

Métodos de busca sem uso de derivadas: O método multidimensional de busca sem o uso de derivadas apresentado aqui determina um vetor $\mathbf{d}$ de direção de busca, que minimiza a função. Este vetor de direção é determinado a cada iteração, quando um novo ponto $\mathbf{x}$ é obtido.

O método de Rosenbrock determina direções de busca, os quais são linearmente independentes e ortonormais entre si. Para gerar este vetor de direções utiliza-se o PROCESSO DE ORTOGONALIZAÇÃO DE GRAM-SCHMIDT. Este método realiza buscas lineares a cada direção, assim solucionando o problema multidimensional em alguns passos.

Métodos de busca com uso de derivadas: Estes métodos usam a derivada da função para obter a direção de busca. A complicação extra associada à busca de direção *fuzzy* está na propagação desta característica, o que pode ser controlado com a discretização da direção antes do cálculo do ponto da próxima iteração, aproximando-se do valor ótimo da função objetivo. Discretizando cada elemento da direção *fuzzy* em J fatores, determina-se os extremos deste número *fuzzy*, expresso por intervalos de confiança.

O método de máxima descida, ou também conhecido como método do gradiente, determina uma direção de descida *fuzzy*, que é determinada pelo valor simétrico da derivada primeira avaliada no ponto da atual iteração. Considera-se uma direção de descida quando o valor da função avaliada no ponto atual é menor que o valor calculado no ponto anterior.

O método de Newton para problemas multidimensionais utiliza a mesma teoria dos problemas unidimensionais. A diferença existente está na derivada segunda, que agora é uma matriz quadrada com a mesma dimensão do problema. Esta matriz gerada pela derivada de segunda ordem é denominada matriz hessiana. Os elementos da matriz hessiana são números *fuzzy*, o que torna a resolução do problema instável, pois não existem uma definição concreta para determinar a inversa de uma matriz com elementos *fuzzy*.

No método do gradiente conjugado, determina-se a primeira direção de busca como sendo o valor da derivada primeira aplicada no ponto inicial. As demais direções serão determinadas usando a equação desenvolvida por FLETCHER E REEVES, apresentada em (BAZARAA et al. 1993), gerando assim direções ortogonais. Sendo as direções com elementos *fuzzy*, necessita-se utilizar a discretização expressa acima.

2.2.2 Programação não-linear restrita

Considera-se o seguinte problema de programação não-linear

$$\begin{aligned} \min \quad & f(\tilde{\mathbf{a}}; \mathbf{x}) \\ \text{s.a} \quad & \mathbf{g}(\mathbf{x}) \leq 0, \\ & \mathbf{h}(\mathbf{x}) = 0, \\ & \mathbf{x} \in \Omega \end{aligned} \tag{2.14}$$

$$\begin{aligned} f &: \mathbb{F}(\mathbb{R}^m) \times \mathbb{R}^n \rightarrow \mathbb{F}(\mathbb{R}) \\ \mathbf{g} &: \mathbb{R}^n \rightarrow \mathbb{R}^k, & \mathbf{g} &:= (g_1, g_2, \dots, g_k) \\ \mathbf{h} &: \mathbb{R}^n \rightarrow \mathbb{R}^l, & \mathbf{h} &:= (h_1, h_2, \dots, h_l), \end{aligned}$$

onde $\mathbf{h}(\mathbf{x}) = 0$ e $\mathbf{g}(\mathbf{x}) \leq 0$ são restrições funcionais, e $x \in \Omega$ é uma restrição de conjunto. Se $\mathbf{x}$ satisfaz todas as restrições, então $\mathbf{x}$ é um ponto factível. Sendo $f(\tilde{\mathbf{a}}; \mathbf{x})$ a função *fuzzy* a ser minimizada, se $\mathbf{x}^*$ é a solução factível que determina o menor valor de f , então é denominada de solução ótima.

Como na Sub-Seção anterior, apresentamos as condições de otimalidade e os métodos desenvolvidos em (CANTÃO 2003) para resolver os problemas de programação não-linear restritos.

2.2.2.1 Condições de otimalidade

As condições de otimalidade para os problemas de programação não-linear restritas *fuzzy* são as condições clássicas de KKT estendidas, apresentados em (BAZARAA et al. 1993, LUENBERGER 1989).

Em (CANTÃO 2003) são fornecidas definições essenciais para garantir as CONDIÇÕES NECESSÁRIAS E SUFICIENTES de otimalidade.

Teorema 2.5 (Condições Necessárias de Primeira Ordem) *Seja $\bar{\mathbf{x}}$ um ponto regular para o conjunto de restrições do problema (2.14) e suponha que as funções $f(\tilde{\mathbf{a}}; \bar{\mathbf{x}})$, $g_i(\mathbf{x})$, para $i = 1, 2, \dots, k$ e $h_j(\mathbf{x})$, para $j = 1, 2, \dots, l$ sejam uma vez diferenciáveis. Se $\bar{\mathbf{x}}$ é um ponto de mínimo local para o problema (2.14), então existem multiplicadores de Lagrange $\mathbf{u} \geq \mathbf{0}$ e $\mathbf{v}$, tais que:*

$$\nabla L(\tilde{\mathbf{a}}; \bar{\mathbf{x}}, \mathbf{u}, \mathbf{v}) = \nabla f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) + \sum_{i \in I} u_i \nabla g_i(\bar{\mathbf{x}}) + \sum_{j=1}^l v_j \nabla h_j(\bar{\mathbf{x}}) \simeq \bar{\mathbf{0}}, \quad (2.15a)$$

$$u_i g_i(\bar{\mathbf{x}}) = 0, \quad i = 1, 2, \dots, k. \quad (2.15b)$$

onde $I = \{i : g_i(\bar{\mathbf{x}}) = 0\}$, o conjunto de restrições ativas.

Prova: vide (CANTÃO 2003). ■

Teorema 2.6 (Condições Necessárias de Segunda Ordem) *Suponha que a função $f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) : \mathbb{F}(\mathbb{R}^m) \times \mathbb{R}^n \rightarrow \mathbb{F}(\mathbb{R})$ e as funções $g_i(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$, para $i = 1, 2, \dots, k$ e $h_j(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$, para $j = 1, 2, \dots, l$, sejam duas vezes diferenciáveis, e seja $\bar{\mathbf{x}}$ um ponto regular de $g_i(\mathbf{x})$ e $h_j(\mathbf{x})$. Se $\bar{\mathbf{x}}$ é um ponto de mínimo local para o problema (2.14), então existem multiplicadores de Lagrange $\mathbf{u} \geq \mathbf{0}$ e $\mathbf{v}$, tais que as condições (2.15) se verificam e a matriz hessiana com parâmetros fuzzy*

$$\nabla^2 L(\tilde{\mathbf{a}}; \bar{\mathbf{x}}, \mathbf{u}, \mathbf{v}) = \nabla^2 f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) + \sum_{i=1}^k u_i \nabla^2 g_i(\bar{\mathbf{x}}) + \sum_{j=1}^l v_j \nabla^2 h_j(\bar{\mathbf{x}}) \quad (2.16)$$

é semi-definida positiva fuzzy no subespaço tangente às restrições ativas em $\bar{\mathbf{x}}$.

Logo, $\mathbf{d}^T \nabla^2 L(\tilde{\mathbf{a}}; \bar{\mathbf{x}}, \mathbf{u}, \mathbf{v}) \mathbf{d} \succeq \tilde{0}$, $\forall \mathbf{d} \in M' = \{\mathbf{d} \neq \mathbf{0} : \nabla g_i(\mathbf{x})^T \mathbf{d} \leq 0, \forall i \in I, \nabla h_j(\mathbf{x})^T \mathbf{d} = 0, j = 1, 2, \dots, l\}$.

Prova: vide (CANTÃO 2003). ■

Teorema 2.7 (Condições suficientes de segunda ordem) *Seja $\bar{\mathbf{x}}$ um ponto regular para o problema (2.14). Supondo que existam multiplicadores de Lagrange $\mathbf{u} \geq \mathbf{0}$ e $\mathbf{v}$ tais que as condições em (2.15) se verificam, e a matriz Hessiana com parâmetros fuzzy (2.16) seja semi-definida positiva fuzzy. Então $\bar{\mathbf{x}}$ é um ponto de mínimo local.*

Prova: vide (CANTÃO 2003). ■

2.2.2.2 Métodos não-lineares restritos com parâmetros fuzzy

Os métodos apresentados por CANTÃO (2003) para solucionar problemas desta natureza, são uma extensão dos métodos abordados em (BAZARAA et al. 1993, FERREIRA 2004, LUENBERGER 1989). Estes métodos são os de função penalidade e do lagrangeano aumentado, os quais aproximam problemas restritos por problemas irrestritos.

Método de função penalidade: A motivação do método de função penalidade está em adicionar o resultado das restrições, obtido da avaliação do ponto atual do problema, como uma penalidade na função objetivo, transformando o problema restrito em um problema irrestrito. A função penalidade, $P(\mathbf{x})$, tende a direcionar o método a encontrar um ponto que adicione penalidade nula e um valor de função objetivo mínimo. A $P(\mathbf{x})$ é formada por funções contínuas satisfazendo as condições de otimalidade vistas acima. O nível de penalização é controlado por um parâmetro, denotado por ρ . Quando $\rho \rightarrow \infty$, a solução do problema irrestrito tende para a solução do problema restrito, porém o condicionamento do problema irrestrito piora, à medida que ρ aumenta.

Método do lagrangeano aumentado: Esse método tem a mesma motivação que o método de função penalidade, o qual resolve um problema restrito através de uma seqüência de problemas irrestritos. A diferença entre o método de Penalização e o método do Lagrangeano

Aumentado está na introdução dos multiplicadores de Lagrange, que permite uma perturbação no termo de penalização original. A aposta deste método está na busca em evitar o mau condicionamento do nível de penalização.

As definições apresentadas nesta seção são um resumo das teorias demonstradas por CANTÃO (2003), nos Capítulos 4 e 5. Os resultados computacionais que confirmam esta teoria estão no Capítulo 6 desta mesma referência.

A contribuição que esta dissertação propõe é a união das teorias apresentadas neste capítulo em um único tipo de problema. Os próximos dois capítulos mostrarão métodos adaptados baseados nos de otimização clássicos.

Capítulo 3

Programação matemática *fuzzy*

Introdução

A proposta principal desta dissertação está na construção de métodos que solucionem problemas de programação matemática com parâmetros *fuzzy* na função objetivo e no conjunto de restrições. Neste capítulo apresentamos três métodos adaptados, indiretamente, de modelos desenvolvidos para resolver problemas de programação matemática clássicos. Assim, o Problema (1.1) será tratado daqui por diante da seguinte forma:

$$\begin{aligned} \min \quad & f(\tilde{\mathbf{a}}; \bar{\mathbf{x}}) \\ \text{s.a} \quad & g_k(\mathbf{x}) \lesssim b_k, \quad k = 1, \dots, m \\ & h_l(\mathbf{x}) \cong c_l, \quad l = 1, \dots, n \\ & x \geq 0 \end{aligned} \tag{3.1}$$

onde $\tilde{\mathbf{a}}$ representa coeficientes *fuzzy* do problema a ser otimizado, e $\lesssim$ e $\cong$ representam a incerteza presente no conjunto de restrições.

O sinal de igualdade *fuzzy* no Problema (3.1) representa um problema, pois, para solucionar problemas com este tipo de restrição, a literatura recomenda o desmembramento em duas restrições de desigualdade. Na Seção 3.1 são apresentados mais detalhes referentes a este problema computacional, pois com esta implementação o esforço computacional aumenta consideravelmente.

Na Seção 3.2 estão apresentados os três métodos adaptados para solucionar os problemas de programação matemática com parâmetros *fuzzy* na função objetivo e com o conjunto de restrições *fuzzy*.

3.1 Conjunto de restrições *fuzzy*

Existem três tipos de problemas de programação matemática com restrições: (a) com conjunto de restrições de igualdade; (b) com conjunto de restrições de desigualdade e (c) com conjunto de restrições mistas. Nos problemas abordados neste trabalho, a natureza *fuzzy* está inserida nos símbolos, ou seja, nos sinais de igualdade e de desigualdade de cada restrição.

As funções de pertinência associadas as restrições de igualdade são apresentadas da seguinte forma:

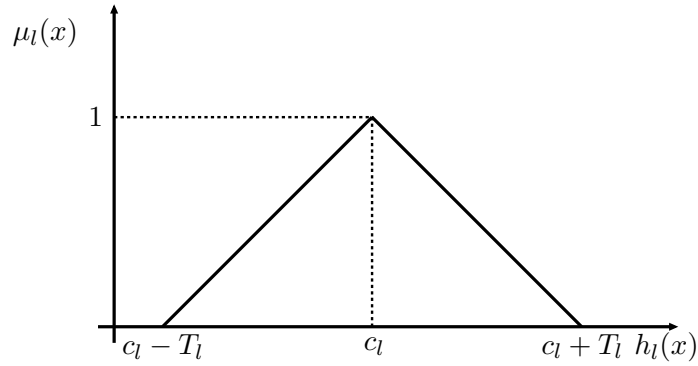


Figura 3.1: Função de pertinência da restrição de igualdade $h_l(\mathbf{x}) = c_l$.

Já as funções de pertinência associadas às restrições de desigualdade podem ser escritas de duas formas que estão representadas graficamente nas Figuras 3.2 e 3.3.

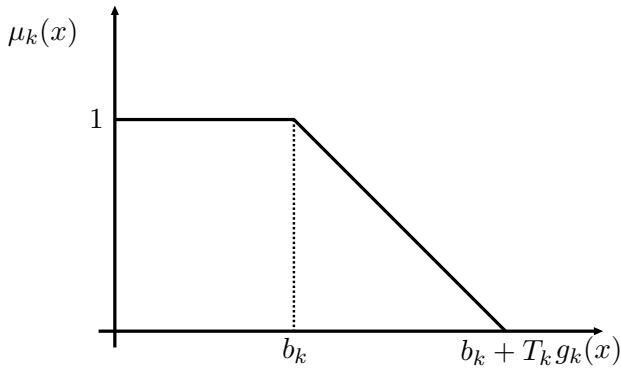


Figura 3.2: Função de pertinência da restrição de desigualdade $g_k(\mathbf{x}) \lesssim b_k$.

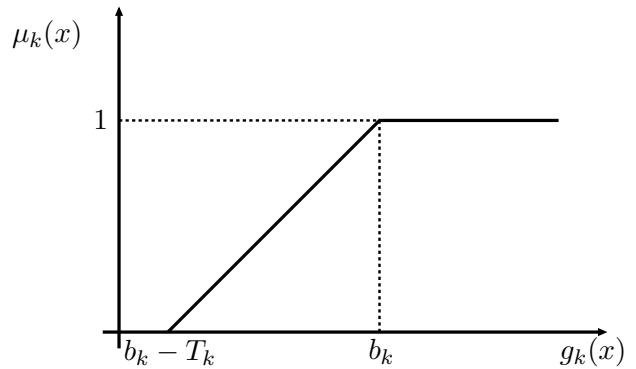


Figura 3.3: Função de pertinência da restrição de desigualdade $g_k(\mathbf{x}) \gtrsim b_k$.

Os métodos adaptados apresentados no decorrer deste capítulo solucionam somente problemas com restrições de desigualdade. Assim, cada restrição de igualdade é re-escrita em duas restrições de desigualdade. Portanto, a Figura 3.1 é dividida nas Figuras 3.2 e 3.3, com $c_l = b_k$ e $T_l = T_k$. Devido à decomposição das restrições de igualdade, o número de

restrições do Problema (3.1) passa a ter $m + 2n$ restrições, onde m é o número de restrições de desigualdade do problema original e $2n$ representa as restrições de desigualdade geradas pelas n restrições de igualdade.

3.2 Métodos adaptados para programação matemática *fuzzy*

Os métodos apresentados nesta seção são propostas para solucionar problemas de programação matemática com incertezas, tanto na função objetivo quanto no conjunto de restrições. Estes métodos são adaptações de métodos desenvolvidos para solucionar problemas com incertezas no conjunto de restrições, descritos em (LEE et al. 1999, TRAPPEY et al. 1988, XU 1989), os quais adaptam métodos clássicos que resolvem problemas de programação matemática clássica.

Em (ZIMMERMANN 1983), Zimmermann desenvolveu uma adaptação de métodos clássicos para resolver problemas de programação linear *fuzzy*, enquanto (LEE et al. 1999, TRAPPEY et al. 1988) expandiram este método para solucionar problemas de programação não-linear *fuzzy*. Baseando-nos nesta expansão, na Sub-Seção 3.2.1 propomos algumas modificações para resolver problemas do tipo descrito em (3.1).

Outro método otimizador de problemas de programação não-linear *fuzzy* foi descrito em (XU 1989), e na Sub-seção 3.2.2 é apresentada uma proposta para solucionar problemas de programação matemática com parâmetros *fuzzy* na função objetivo e com incertezas no conjunto de restrições.

A proposta descrita na Sub-seção 3.2.3 refere-se ao acréscimo de um termo de penalização na função objetivo. Este método foi baseado na técnica de inserir variáveis auxiliares no problema original com o intuito de facilitar a resolução do mesmo.

3.2.1 Método adaptado de Zimmermann

Um problema de programação matemática com uma função objetivo e restrições *fuzzy* pode ser escrito genericamente, utilizando como base o Problema (3.1), da seguinte forma:

$$\begin{aligned} \min \quad & g_0(\tilde{\mathbf{a}}; \mathbf{x}) \\ \text{s.a} \quad & g_i(\mathbf{x}) \lesssim b_i, \quad i = 1, \dots, m + 2n \\ & x \in \Omega, \end{aligned} \tag{3.2}$$

onde $\tilde{\mathbf{a}}$ representam os parâmetros *fuzzy* na função objetivo e $\lesssim$ representam as características *fuzzy* no conjunto de restrição do problema.

A função objetivo com parâmetros *fuzzy* pode ser interpretada como

$$\mathbb{Df}(g_0(\tilde{a}; x)) \lesssim \mathbb{Df}(\tilde{b}_0)$$

A transformação das restrições com incertezas em restrições clássicas de desigualdade é feita acrescentando uma variável t_j , com $j = 0, 1, \dots, m + 2n$, que indica o nível de violação das restrições *fuzzy*; então, pode-se modificar o problema (3.2) da seguinte forma:

$$\begin{aligned} \mathbb{Df}(g_0(\tilde{\mathbf{a}}; \mathbf{x})) &\leq \mathbb{Df}(\tilde{b}_0) + \mathbb{Df}(\tilde{t}_0) \\ g_i(\mathbf{x}) &\leq b_i + t_i, \quad i = 1, \dots, m + 2n \\ \mathbf{x} &\geq \mathbf{0} \\ 0 &\leq \mathbb{Df}(\tilde{t}_0) \leq \mathbb{Df}(\tilde{T}_0) \\ 0 &\leq t_i \leq T_i \end{aligned}$$

onde $t_j = [\mathbb{Df}(\tilde{t}_0), t_1, \dots, t_{m+2n}]$, com $j = 0, 1, \dots, m + 2n$, representa a variação de violação da restrição j , e $T_j = [\mathbb{Df}(\tilde{T}_0), T_1, \dots, T_{m+2n}]$ o valor máximo de violação admissível da restrição j .

As funções de pertinência $\mu_0(g_0(\tilde{\mathbf{a}}; \mathbf{x}))$ e $\mu_i(g_i(\mathbf{x}))$, com $i = 1, \dots, m + 2n$, estão descritas abaixo:

$$\mu_0(g_0(\tilde{\mathbf{a}}, \mathbf{x})) = \begin{cases} 0, & \text{se } \mathbb{Df}(\tilde{t}_0) \geq \mathbb{Df}(\tilde{T}_0) \\ 1 - \frac{\tilde{t}_0}{\tilde{T}_0}, & \text{se } 0 < \mathbb{Df}(\tilde{t}_0) < \mathbb{Df}(\tilde{T}_0) \\ 1, & \text{se } \mathbb{Df}(\tilde{t}_0) \leq 0 \end{cases}$$

$$\mu_i(g_i(\mathbf{x})) = \begin{cases} 0, & \text{se } t_i \geq T_i \\ 1 - \frac{t_i}{T_i}, & \text{se } 0 < t_i < T_i \\ 1, & \text{se } t_i \leq 0 \end{cases}$$

onde cada uma representa o grau de satisfação que a solução $\mathbf{x}$ fornece para a restrição j , com $j = 0, 1, \dots, m + 2n$. Contudo, pode-se definir uma função de agregação de todas as restrições como

$$\bigcap_{0 \leq j \leq m+2n} \bigcap_{\mathbf{x}} \mu_j(g_j(\mathbf{x})) \in [0, 1]$$

onde $\bigcap$ é o símbolo de intersecção.

Usando a regra de minimização para a intersecção de múltiplos conjuntos *fuzzy*, o método tenta encontrar a intersecção satisfatória das pertinências dentro de cada restrição. Então, o método tenta maximizar o grau de todas as restrições satisfeitas. Fazendo $S = \min_{0 \leq j \leq m+2n} \mu_j(g_j(\mathbf{x}))$, o modelo não-linear *fuzzy* pode ser formulado como:

$$\begin{aligned} \max \quad & \tilde{S} \\ \text{s.a} \quad & (a) \quad \text{IDf}(g_0(\tilde{\mathbf{a}}; \mathbf{x}) - \tilde{b}_0 - \tilde{T}_0 * (1 - \tilde{S})) \leq 0 \\ & (b) \quad g_i(\mathbf{x}) - b_i - T_i * (1 - \text{IDf}(\tilde{S})) \leq 0 \\ & \tilde{S} \in [0, 1], \quad \mathbf{x} \in \Omega, \\ & \tilde{T}_0 \geq \tilde{0}, \quad T_i \geq 0, \quad i = 1, \dots, m + 2n \end{aligned} \tag{3.3}$$

Nota-se que a restrição (a) é a função objetivo do problema original, e as restrições sumarizadas em (b) representam o conjunto de restrições do problema original. O objetivo $\tilde{S}$ é representado como número *fuzzy* porque a função de pertinência da função objetivo original fornece um valor *fuzzy*.

3.2.2 Método adaptado de Xu

O Método Duas Fases começa transformando um problema *fuzzy* em um problema clássico equivalente. Então, podemos escrever o Problema (3.2) como

$$\begin{aligned} \min \quad & g_0(\tilde{\mathbf{a}}; \mathbf{x}) \\ \text{s.a} \quad & g_i(\mathbf{x}) \leq b_i + T_i, \quad i = 1, \dots, m + 2n \\ & \mathbf{x} \in \Omega \end{aligned} \tag{3.4}$$

Contudo, utilizando uma função $\mu_i(\mathbf{x}) : \mathbb{R}^n \rightarrow [0, 1]$, descrito na Equação (2.4), teremos graus diferentes de satisfação dentro do intervalo real unitário $[0, 1]$.

Para tanto, precisamos parametrizar as restrições usando os níveis α -cortes propostos na teoria de conjuntos *fuzzy* (ZADEH 1965), apresentados pela Equação (2.5) no capítulo

anterior, porque, depois de realizadas as transformações e os cortes, faz-se uma intersecção de todas as restrições com um operador de agregação, da seguinte forma:

$$\mu_C(\mathbf{x}) = \min_{i=1}^{m+2n} \mu_i(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^n$$

Logo, utiliza-se o mínimo, em (2.6), como operador de agregação.

Por fim, consegue-se escrever o problema *fuzzy* em um problema clássico paramétrico. Contudo, existem vários valores da nossa função objetivo, dependendo do valor de α . Logo, a Equação (3.4) é reescrita como

$$\begin{aligned} \min \quad & g_0(\tilde{\mathbf{a}}; \mathbf{x}) \\ \text{s.a} \quad & g_i(\mathbf{x}) \leq b_i + T_i(1 - \alpha), \\ & \mathbf{x} \in \Omega, \quad \alpha \in [0, 1], \quad i = 1, \dots, m + 2n \end{aligned} \quad (3.5)$$

A solução do Problema (3.5) resulta em um conjunto de soluções, uma para cada valor de α diferente, e assim termina a primeira fase desse método.

A segunda fase consiste em fazer a intersecção da função objetivo com a intersecção de todas as restrições da fase anterior. Assim,

$$\mu_D = \mu_C \bigcap \mu_{\tilde{G}},$$

onde $\mu_D, \mu_{\tilde{G}}, \mu_C : \mathbb{F}(\mathbb{R}^n) \rightarrow [0, 1]$.

A decisão ótima no espaço factível é obtida com a função de pertinência

$$\mu_D(\mathbf{x}^*) = \max_{\mathbf{x} \in \mathbb{R}^n} \mu_D(\mathbf{x}). \quad (3.6)$$

Este método de resolução é proposto por Bellman e Zadeh (BELLMAN & ZADEH 1970).

A Equação (3.6), pode fornecer um nível ótimo α^* e o ponto ótimo $\mathbf{x}^*$ tal que

$$\mu_{\tilde{G}}(\mathbf{x}^*) = \max_{\mathbf{x} \in C_{\alpha^*}} \mu_{\tilde{G}}(\mathbf{x}), \quad (3.7)$$

onde C_{α^*} é o nível α^* -corte do conjunto de restrições *fuzzy* C .

A solução *fuzzy* é dada dentro de um intervalo limitado pelos valores superior e inferior fornecidos pela função $\mu_G(\mathbf{x})$, sendo

$$\begin{aligned} \tilde{m} &= g_0(\tilde{\mathbf{a}}; \mathbf{x}^*(0)) = \min_{\mathbf{x} \in C_0} g_0(\tilde{\mathbf{a}}; \mathbf{x}) \\ \tilde{M} &= g_0(\tilde{\mathbf{a}}; \mathbf{x}^*(1)) = \min_{\mathbf{x} \in C_1} g_0(\tilde{\mathbf{a}}; \mathbf{x}), \end{aligned}$$

onde, C_0 , C_1 são os níveis de corte para α igual a 0 e 1, respectivamente, do conjunto de restrições *fuzzy* C .

Em alguns problemas de otimização *fuzzy*, para achar mínimos de uma função objetivo sujeita a determinadas restrições, podemos estabelecer a seguinte função:

$$\mu_{\tilde{G}}(\mathbf{x}) = \frac{\tilde{m}}{g_0(\tilde{\mathbf{a}}; \mathbf{x})}. \quad (3.8)$$

Claramente, os limites superior e inferior da função objetivo são dados por

$$\begin{aligned} \mu_{\tilde{G}}^u &= 1 \\ \mu_{\tilde{G}}^l &= \text{Idf} \left(\frac{\tilde{m}}{\tilde{M}} \right). \end{aligned} \quad (3.9)$$

Substituindo a Equação (3.8) na Equação (3.7), obtemos

$$\mu_{\tilde{G}}(\mathbf{x}^*) = \frac{1}{\tilde{m}} \min_{\mathbf{x} \in C_{\alpha}^*} g_0(\tilde{\mathbf{a}}; \mathbf{x}) \quad (3.10)$$

Desta maneira, a metodologia empregada tem capacidade de otimizar um problema *fuzzy* pelo método de busca nos limites do conjunto de soluções factíveis.

3.2.3 Método com penalização na função objetivo

O terceiro algoritmo adaptado para resolver Problemas de Programação Não-Linear com incertezas na Função Objetivo e no Conjunto de Restrições, descrito no Problema (3.1), transforma o parâmetro α , do Problema (3.5), numa variável de decisão do problema. Assim,

$$\begin{aligned} \min \quad & g_0(\tilde{\mathbf{a}}; \mathbf{x}) + \frac{\tilde{m}}{\tilde{M}}(1 - \alpha) \\ \text{s.a} \quad & g_i(\mathbf{x}) - T_i(1 - \alpha) \leq b_i, \\ & \mathbf{x} \in \Omega, \quad \alpha \in [0, 1], \quad i = 1, \dots, m + 2n. \end{aligned} \quad (3.11)$$

Sendo α uma variável de decisão, temos de introduzi-la na função objetivo com algum fator de penalização. Este valor de penalização foi escolhido mediante o seu uso nas adaptações descritas anteriormente, onde os valores $\tilde{m}$ e $\tilde{M}$ são adotados, respectivamente, como iguais a $\tilde{b}_0$ e $\tilde{T}_0$.

No próximo capítulo descreveremos dois modelos de algoritmos genéticos para solucionar problemas de programação matemática com incertezas na função objetivo e no conjunto

de restrições, formalizados matematicamente em (3.1). Será apresentado uma forma mais simplificada de lidar com restrições de igualdade, em comparação com a forma apresentada neste Capítulo 3.

Capítulo 4

Meta-heurística fuzzy

Introdução

O algoritmo genético é um método de busca para problemas de otimização baseados no mecanismo de seleção natural e genética natural, isto é, no princípio da evolução. Estes algoritmos têm demonstrado sucessos consideráveis fornecendo boas soluções para muitos problemas complexos de otimização e recebeu bastante atenção durante as últimas décadas. A vantagem dos algoritmos genéticos está em obter, de forma estocástica, várias soluções que permitem uma maior análise do espaço de restrições. Os algoritmos genéticos, incluindo programas e estratégias evolucionárias, são discutidos em vários livros, como (GOLDBERG 1989, MICHALEWICZ 1996), e aplicados a uma grande variedade de problemas, tais como problemas de controle ótimo, de transporte, escalonamento entre outros. As publicações de Liu e Iwamura (LIU & IWAMURA 1998a, LIU & IWAMURA 1998b) serviram de base para a criação dos algoritmos expostos aqui, porém eles diferem em alguns aspectos.

Na Seção 4.1 são mostradas as restrições e particularidades que os algoritmos genéticos conseguem resolver. Nesta seção apresenta-se também a forma como se calcula o *fitness* e o nível de satisfação de cada indivíduo da população. O nível de satisfação é usado para determinar a factibilidade de cada problema.

Na Seção 4.2 são apresentadas duas versões de Algoritmos Genéticos para resolver problemas de programação não-linear com incertezas na função objetivo e nas restrições, mais os passos utilizados para a construção destes algoritmos.

4.1 Critérios de decisão

Os algoritmos de otimização necessitam de critérios de decisão para a definição do momento em que o ótimo do problema é alcançado, isto é, para se encontrar a solução do problema que atenda a todos os critério desejados. Estes critérios estão apresentados nesta seção de três formas: (i) satisfação do conjunto de restrições (factibilidade); (ii) valor de *fitness* e (iii) nível de satisfação.

Na Sub-seção 4.1.1 são mostrados os tipos de conjuntos de restrições que os algoritmos genéticos descritos neste trabalho levam em consideração. Na Sub-seção 4.1.2 apresenta-se a forma de calcular o valor do *fitness* de cada indivíduo da população. Por fim, na Sub-seção 4.1.3 apresenta-se o cálculo do nível de satisfação do indivíduo em relação ao ótimo do problema a ser otimizado.

4.1.1 Conjunto de restrições

Os problemas de programação matemática podem ser divididos em 3 (três) tipos de conjuntos de restrições, porém os métodos de otimização que usam conceitos estocásticos resolvem somente problemas de programação matemática com um conjunto de restrições de desigualdade, pois o conjunto de restrições de igualdade torna difícil manter a factibilidade. Contudo, a condição *fuzzy* no conjunto de restrições permite, aos algoritmos genéticos descritos neste trabalho, resolver problemas com qualquer tipo de conjunto de restrições.

A condição *fuzzy* acrescentada nas restrições de igualdade pode ser representada graficamente como na Figura 3.1, enquanto as Figuras 3.2 e 3.3 representam a condição *fuzzy* acrescentada nas duas formas de restrições de desigualdade.

A literatura defende a decomposição de cada restrição de igualdade em duas restrições de desigualdade, como apresentadas nas Figuras 3.2 e 3.3. Assim, o esforço computacional aumenta proporcionalmente ao aumento do número de restrições de igualdade.

A proposta descrita neste trabalho refere-se o uso da função módulo para facilitar a implementação dos algoritmos genéticos. Seja $h_l(\mathbf{x}) \cong c$, descrito na Figura 3.1, então $h_l(\mathbf{x}) - c \cong 0$. Logo, usando a função módulo temos $|h_l(\mathbf{x}) - c| \lesssim 0 \Rightarrow |h_l(\mathbf{x}) - c| \leq T_l \bullet (1 - \alpha)$. Assim, com esta transformação a Figura 3.1 pode ser representada graficamente como na Figura 4.1 a seguir.

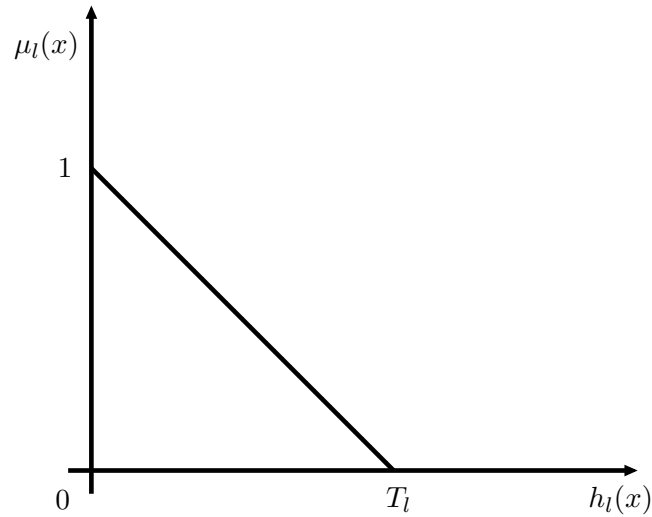


Figura 4.1: Função de pertinência da restrição de igualdade transformada

Logo, transforma-se a restrição de igualdade numa única restrição de desigualdade, porém esta nova restrição não permite um relaxamento da restrição primitiva, isto é, esta restrição transformada ainda mantém as características de uma restrição de igualdade. Porém, o esforço computacional permanece, praticamente, sem alteração.

4.1.2 Valor de *fitness*

A forma de determinar a qualidade de cada indivíduo da população durante toda uma evolução é chamada de *fitness*. O *fitness* pode ser avaliado de várias formas, porém a mais utilizada é o valor da função objetivo quando calculado para o cromossomo do indivíduo. A melhor forma de determinar o tipo de qualidade a ser avaliada no indivíduo depende da representação estrutural usada no algoritmo genético. Em (GOLDBERG 1989, MICHALEWICZ 1996) pode-se encontrar outras formas de calcular o *fitness*.

Neste trabalho, a forma de avaliação de *fitness* escolhida foi o valor da função objetivo, pois a representação estrutural, apresentada na próxima seção, tornou-se propícia a avaliação desta qualidade.

Sendo o valor de *fitness* de cada indivíduo um número *fuzzy*, então necessita-se encontrar um número clássico que melhor represente a distribuição do *fitness*. Existem, na literatura, alguns métodos de representação, dos quais alguns podem ser encontrados em (CANTÃO 2003, KLIR & FOLGER 1998, LIU & IWAMURA 1998a, PEDRYCZ & GOMIDE 1998).

O método de representação escolhido para o valor de *fitness* foi o primeiro índice de Yager, também chamado de centro de gravidade ou centro de massa.

4.1.3 Nível de satisfação

O uso de números *fuzzy* no critério de decisão nos fornece um conjunto de soluções com uma variedade de valores de pertinência. Assim, o método usado para otimização pode encontrar melhores soluções que as encontradas por métodos clássicos, porém com um valor de pertinência baixo demais. O valor de pertinência da solução apresenta 2 (dois) pontos a serem avaliados: (a) valor de pertinência baixo com valor de *fitness* baixo, portanto, com uma confiabilidade baixa; (b) valor de pertinência alto com um *fitness* mais elevado, portanto, com uma confiabilidade alta. A forma de tratar esta situação é criar um nível de satisfação, o qual aceita soluções ótimas acima de um determinado valor de pertinência. O nível de satisfação pretende criar uma relação de *fitness* e confiabilidade.

O nível mínimo de satisfação aceitável está determinado na Seção 3.2.2 do Capítulo 3 deste trabalho. O cálculo deste nível mínimo de satisfação aceitável está na razão entre a solução ótima do problema a ser otimizado, na qual as restrições estão totalmente violadas, com a solução ótima deste mesmo problema, em que as restrições estão totalmente satisfeitas. Esta razão está dentro do intervalo aberto $(0, 1)$. Sendo as soluções encontradas números *fuzzy*, então a razão entre as soluções determinará um outro número *fuzzy*. Este procedimento determina uma relação *fuzzy* no qual cada índice de Yager não forneceu bons resultados. Portanto, o número “defuzzificado” que representa o valor da razão escolhido neste trabalho é o valor modal da relação das duas soluções *fuzzy* calculadas.

4.2 Algoritmos genéticos adaptados para programação matemática fuzzy

Nesta seção serão mostradas duas versões de algoritmos genéticos. Estes algoritmos foram desenvolvidos tendo como base uma simulação *fuzzy*, descrita em (LIU & IWAMURA 1998a), baseada em algoritmo genético básico.

Na Sub-seção 4.2.1 descreve-se um algoritmo genético puro, isto é, este algoritmo usa

estratégias evolucionárias básicas para obter uma solução para o problema de programação matemática.

Na Sub-seção 4.2.2, um algoritmo genético é descrito usando as estratégias evolucionárias mencionadas na Sub-seção 4.2.1 e mais um método de busca local. Alguns autores chamam este algoritmo de "memético", pois utiliza-se de mecanismos que não pertencem ao conceito genético padrão.

4.2.1 Algoritmo genético puro

O algoritmo genético puro utilizado neste trabalho enfatiza somente o uso de estratégias evolucionárias básicas. Estas estratégias estão centradas no processo de seleção, na operação de reprodução e na operação de mutação. Ao final desta seção estão sumarizados os passos de um algoritmo genético puro.

4.2.1.1 Representação estrutural

No conceito básico de algoritmo genético, cada cromossomo é dividido em genes, os quais são representados por elementos binários, e o tamanho dos cromossomos depende da precisão requerida. Neste trabalho usamos uma representação alternativa na qual os genes de cada cromossomo são implementados como pontos flutuantes, isto é, elementos reais. Nesta representação, cada cromossomo tem o mesmo número de genes que a dimensão da variável do problema a ser otimizado, e cada cromossomo representa uma solução deste problema. Assim, seja $\mathbf{V}^i = (x_1^i, x_2^i, \dots, x_n^i)$ um vetor solução do problema a ser otimizado, onde n é a dimensão da variável de decisão, i o tamanho da população em cada geração e $x_j^i \in \mathbb{R}$ para cada i e j . Então cada indivíduo da população terá cromossomo com a mesma representação do $\mathbf{V}^i$ com n genes.

4.2.1.2 Processo de inicialização

A geração de uma população inicial depende da dimensão da variável de decisão do problema a ser otimizado, pois o tamanho da população é feito igual a 10 vezes esta dimensão, isto é, $pop_size = 10 \bullet n$. A população inicial é gerada de forma aleatória, na qual o **decisor** determina um ponto interior, denotado por $\mathbf{V}^0$, que respeite todas as restrições deste problema. A geração aleatória consiste em selecionar uma direção, $\mathbf{b}^i \in \mathbb{R}^n$, de forma randô-

mica. Em seguida, multiplica-se esta direção por um número positivo, $\Gamma \in \mathbb{R}$, suficientemente grande. Este número Γ determina o passo que será dado na direção aleatória determinada, e ele não é usado somente para o processo de inicialização, mas também na operação de mutação. A forma de determinar a população inicial é definir um cromossomo $\mathbf{V}^i$ como $\mathbf{V}^0 + \Gamma \bullet \mathbf{d}^i$, onde $i = 1, 2, \dots, pop_size$, se este for factível para as restrições de desigualdade e igualdade. Caso contrário, o valor Γ será atualizado por um número aleatório entre 0 e Γ até $\mathbf{V}^0 + \Gamma \bullet \mathbf{d}^i$ ser factível. Na implementação deste algoritmo, cada posição do vetor $\mathbf{d}^i$, para cada i , encontra um valor real no intervalo $[-1, 1]$ que pode abranger todas as direções possíveis. O valor de Γ torna a ser igual a 0 caso o número aleatório encontrado seja menor que 10^{-4} , pois nesta situação o ponto inicial encontra-se limítrofe à fronteira do conjunto de soluções factíveis. Este processo é repetido pop_size vezes e fornece pop_size soluções iniciais factíveis $\mathbf{V}^1, \mathbf{V}^2, \dots, \mathbf{V}^{pop_size}$.

4.2.1.3 Função de avaliação

A função de avaliação, denotada por $eval(\mathbf{V}^i)$, com $i = 1, 2, \dots, pop_size$, é realizada para determinar a probabilidade da reprodução de cada indivíduo da população, isto é, para cada cromossomo $\mathbf{V}^i$. A possibilidade que um indivíduo seja selecionado é proporcional ao seu valor de *fitness* em relação aos outros indivíduo na população. Assim, os indivíduos que possuírem os melhores *fitness* terão uma chance maior, usando a seleção da roleta, de serem selecionados a produzir filhos utilizando a operação de reprodução.

Sejam $\mathbf{V}^1, \mathbf{V}^2, \dots, \mathbf{V}^{pop_size}$ os indivíduos da população na geração corrente. Um método interessante é aquele baseado na tentativa de reproduzir os indivíduos de acordo com a sua classificação, segundo o valor objetivo atual. É possível assumir o uso da relação ordenada entre os pop_size indivíduos da população, tal que os pop_size indivíduos possam ser reorganizados dos melhores para os piores, isto é, do melhor indivíduo, de acordo com o valor de *fitness*, para o pior. Seja um parâmetro $a \in (0, 1)$ num sistema genético dado, então pode-se definir uma função de avaliação baseada na ordenação dos indivíduos de uma população, que pode ser escrita como:

$$eval(\mathbf{V}^i) = a(1 - a)^{i-1}, \quad i = 1, 2, \dots, pop_size.$$

Como mencionado acima, para $i = 1$ está armazenado o melhor indivíduo, para $i =$

pop_size está o pior indivíduo, e

$$\sum_{j=1}^{pop_size} eval(\mathbf{V}^j) \approx 1.$$

4.2.1.4 Processo de seleção

O processo de seleção é baseado no uso de uma técnica chamada de roleta para selecionar os indivíduos da população. A roleta é girada pop_size vezes, de forma que, em cada giro, seleciona-se um único indivíduo para uma população auxiliar. As operações de reprodução e de mutação utilizam-se desta população auxiliar para realizar os cálculos necessários, que contribuem à evolução da população durante o decorrer das gerações.

A seleção pelo giro da roleta, usada neste processo, pode ser descrita da seguinte forma:

1. Calcular a probabilidade acumulativa q_i para cada indivíduo $\mathbf{V}^i$ da população,

$$q_0 = 0,$$

$$q_i = \sum_{j=1}^i eval(\mathbf{V}^j), \quad i = 1, 2, \dots, pop_size;$$

2. Gerar um número real aleatório r entre $[0, q_{pop_size}]$;
3. Selecionar o i -ésimo indivíduo $\mathbf{V}^i$ ($1 \leq i \leq pop_size$), tal que, $q_{i-1} < r \leq q_i$;
4. Repetir os itens 2 e 3 pop_size vezes e obtendo pop_size cópias dos indivíduos.

Assim, é constituída uma população auxiliar, com o mesmo tamanho da população original, gerada por indivíduos selecionados de forma aleatória, em que alguns são prestigiados por terem melhor valor de *fitness*.

4.2.1.5 Operação de reprodução

Na Operação de Reprodução define-se um parâmetro, P_r , que determina quantos indivíduos são selecionados para iniciar o processo de geração de indivíduos filhos. Este parâmetro de reprodução é um valor real fixado pelo **decisor** no intervalo aberto de $(0, 1)$. O número de indivíduos que participam da operação de reprodução é igual ao número inteiro positivo mais próximo do valor $P_r \bullet pop_size$, de forma que, se este número inteiro for ímpar, então o resultado é acrescido de uma unidade. Um valor inteiro par é forçado para que a cada indivíduo selecionado para a reprodução possa ser garantido um indivíduo parceiro. Neste

trabalho, o valor da probabilidade de reprodução é fixado em $P_r = 0.2$; então, podemos concluir que o número de indivíduos que são selecionados para a operação de reprodução é expresso pela seguinte equação:

$$P_r \bullet pop_size \Rightarrow 0.2 \bullet (10 \bullet n) \Rightarrow 2 \bullet n$$

Logo, o número de indivíduos selecionados, de forma aleatória para a operação de reprodução é igual a $2 \bullet n$, para qualquer dimensão do problema a ser otimizado.

A forma mecânica usada no operador de reprodução é a variação linear entre dois pontos, os quais são os indivíduos selecionados para serem os pais. Em primeiro lugar, gera-se um número aleatório c dentro do intervalo aberto $(0, 1)$, depois opera-se com os dois indivíduos pais, P_1 e P_2 , que fornecerão dois filhos F_1 e F_2 da seguinte forma:

$$\mathbf{Filho}_1 = c \bullet \mathbf{Pai}_1 + (1 - c) \bullet \mathbf{P}_2 \quad \text{e} \quad \mathbf{Filho}_2 = (1 - c) \bullet \mathbf{P}_1 + c \bullet \mathbf{P}_2$$

Se a região factível é um conjunto convexo, então ambos os indivíduos filhos são factíveis se os indivíduos pais também o forem. Contudo, na maioria dos problemas de programação matemática, a região factível não é necessariamente convexa ou é difícil de verificar a sua convexidade. A solução para este conflito é checar a factibilidade de cada indivíduo filho. Se ambos os indivíduos filho forem factíveis, então os mesmos substituirão os indivíduos pai. Caso contrário, tenta-se factibilizar o indivíduo filho infactível, pois refaz-se o operador de reprodução com uma nova geração de um número aleatório c até encontrar um cromossomo factível ou termina-se com um número pré-definido de ciclos. Neste caso, os indivíduos pai são mudados somente se os indivíduos filho forem factíveis.

4.2.1.6 Operação de mutação

Um parâmetro P_m foi definido com o intuito de determinar quantos indivíduos são selecionados para iniciar a operação de mutação, que introduz na solução vigente uma perturbação que é suposto influenciar na diversificação de busca da região factível. O número de indivíduos que são selecionados é igual ao número inteiro mais próximo de $P_m \bullet pop_size$. Este valor P_m é escolhido dentro do intervalo real aberto $(0, 1)$, no qual o **decisor** determina o seu valor. Neste trabalho, o valor da probabilidade de mutação é fixado em $P_m = 0.1$; então,

pode-se concluir que o número de indivíduos que é selecionado para a operação de mutação é expresso pela seguinte equação:

$$P_m \bullet pop_size \Rightarrow 0.1 \bullet (10 \bullet n) \Rightarrow n$$

Logo, o número de indivíduos selecionados para a operação de mutação é igual a n , para qualquer dimensão do problema a ser otimizado.

Para cada indivíduo selecionado, $\mathbf{V}$, a operação é realizada escolhendo uma direção $\mathbf{d} \in \mathbb{R}^n$ por meio aleatório. Se $\mathbf{V} + \Gamma \bullet \mathbf{d}$ não for factível para as restrições do problema, então o valor Γ é atualizado por um número aleatório entre 0 e Γ , até que a operação anterior seja factível, onde Γ é um número positivo suficientemente grande definido no item **processo de inicialização**. Caso o processo não encontre uma solução factível para valor de $\Gamma \leq 10^{-4}$, então define-se $\Gamma = 0$.

4.2.1.7 Procedimento do algoritmo genético puro

Depois da seleção, reprodução e mutação, a nova população é modificada para a próxima geração. O algoritmo genético termina quando atingir um número pré-definido de repetições destes passos. Pode-se descrever cada passo deste algoritmo genético idealizado para resolver problemas de programação matemática com parâmetros *fuzzy* na função objetivo e no conjunto de restrições da seguinte forma:

Algoritmo 4.2.1: Algoritmo Genético Puro

Entrada: Introduzir parâmetros pop_size , P_r e P_m .

Saída: Retornar o melhor indivíduo como solução do problema.

Passo 1 $\Rightarrow$ Inicializar pop_size indivíduos;

Passo 2 $\Rightarrow$ Checar a factibilidade dos indivíduos e calcular os seus *fitness*;

Passo 3 $\Rightarrow$ Ordenar a população, calcular a função de avaliação;

Passo 4 $\Rightarrow$ Realizar o processo de seleção;

Passo 5 $\Rightarrow$ Aplicar a operação de reprodução e checar a factibilidade dos filhos;

Passo 6 $\Rightarrow$ Utilizar a operação de mutação e checar a factibilidade dos mutantes;

Passo 7 $\Rightarrow$ Selecionar os indivíduos que passaram para a próxima geração;

Passo 8 $\Rightarrow$ Calcular o *fitness* de cada indivíduo;

Passo 9 $\Rightarrow$ Repetir os passos $P3$ a $P7$ por um número pré-definido de gerações.

4.2.2 Algoritmo genético com busca local

O algoritmo genético descrito abaixo apresenta uma estratégia chamada de "operação de busca local". A implementação de um algoritmo genético com a estratégia da operação de busca local é chamada, por alguns autores, de ALGORITMO MEMÉTICO. A operação de busca local é baseada no MÉTODO DE MÁXIMA DESCIDA. Ao final desta seção estão sumarizados os passos necessários para concluir um algoritmo genético com busca local.

4.2.2.1 Operação de busca local

Esta operação de busca local tem por finalidade realizar uma busca na vizinhança e tentar encontrar solução melhor que a atual. Assim, usa-se essa estratégia em alguns indivíduos com a intenção de evoluir a população em número de gerações menor. Alguns cuidados devem ser tomados com a estratégia de busca local, pois quando aplicada em toda a população, gera-se dois problemas:

1. esforço computacional mais elevado;
2. convergência prematura para um mínimo local (principalmente).

A operação de busca local baseia-se no método de máxima descida, usado para otimizar problemas de programação matemática irrestritos. Algumas mudanças foram feitas para atender as necessidades deste algoritmo genético. A principal mudança realizada no método de máxima descida, descrito em (CANTÃO 2003), está na retirada do critério de parada. Assim, o cálculo do passo pode ser feito de duas formas:

$$\lambda_k = \arg \min_{\lambda > 0} f(\tilde{\mathbf{a}}; \mathbf{x} + \lambda \mathbf{d}) + pen * \sum_{i=0}^R \max[0, g(\mathbf{x} + \lambda \mathbf{d})]$$

$$\lambda_k = \arg \min_{\lambda > 0} f(\tilde{\mathbf{a}}; \mathbf{x} + \lambda \mathbf{d}) - bar * \sum_{i=0}^R \ln[-g(\mathbf{x} + \lambda \mathbf{d})]$$

Esta estratégia define somente uma direção de descida, pois, com o decorrer das gerações, essa direção tende a convergir para a solução ótima do problema a ser otimizado. A garantia de fornecer o melhor passo na direção definida está em avaliar a função objetivo acrescida do somatório do valor de todas as restrições infactíveis. Este procedimento foi idealizado com base nos métodos de penalidade e de barreira, porém não é resolvido nenhum destes métodos na estratégia de busca local.

4.2.2.2 Procedimento do algoritmo genético com busca local

Depois da seleção, reprodução e mutação, a nova população é modificada para a próxima geração. O algoritmo genérico termina quando atingir um número pré-definido de repetições destes passos. Pode-se descrever cada passo deste algoritmo genético idealizado para resolver problemas de programação matemática com parâmetros *fuzzy* na função objetivo e no conjunto de restrições da seguinte forma:

Algoritmo 4.2.2: Algoritmo Genético com Busca Local

Entrada: Introduzir parâmetros pop_size , P_r e P_m .

Saída: Retornar o melhor indivíduo como solução do problema.

Passo 1 $\Rightarrow$ Inicializar pop_size indivíduos;

Passo 2 $\Rightarrow$ Checar a factibilidade dos indivíduos e calcular o seus *fitness*;

Passo 3 $\Rightarrow$ Ordenar a população, calcular a função de avaliação;

Passo 4 $\Rightarrow$ Realizar o processo de seleção;

Passo 5 $\Rightarrow$ Aplicar a operação de reprodução e checar a factibilidade dos filhos;

Passo 6 $\Rightarrow$ Utilizar a operação de mutação e checar a factibilidade dos mutantes;

Passo 7 $\Rightarrow$ Usar a operação de busca local e checar a factibilidade dos modificados;

Passo 8 $\Rightarrow$ Selecionar os indivíduos que passaram para a próxima geração;

Passo 9 $\Rightarrow$ Calcular o *fitness* de cada indivíduo;

Passo 10 $\Rightarrow$ Repetir os passos $P3$ a $P8$ por um número pré-definido de gerações.

Mostramos, neste capítulo, todas as estratégias utilizadas para a construção de dois algoritmos genéticos para solucionar problemas de programação matemática com parâmetros *fuzzy* na função objetivo e com incertezas no conjunto de restrições. No próximo capítulo são apresentados os resultados computacionais referentes à execução destes algoritmos genéticos.

Capítulo 5

Resultados Numéricos

Introdução

Neste capítulo estão alguns problemas utilizados para verificar a eficiência da teoria demonstrada nos Capítulos 2, 3 e 4. Os problemas foram divididos em duas categorias, as quais representam as diferentes abordagens realizadas na primeira seção do Capítulo 2, em que as incertezas encontram-se no conjunto de restrições, e nos Capítulos 3 e 4, em que as incertezas aparecem tanto na função objetivo quanto no conjunto de restrições.

A primeira seção deste capítulo mostra a formulação dos problemas. Esta formulação dos problemas referentes a primeira seção do Capítulo 2 trata de três problemas que podem ocorrer em ambientes da vida real. Os problemas usados para avaliar a teoria dos Capítulos 3 e 4 são formulações hipotéticas, porém suficientes para validar o estudo realizado.

Os resultados computacionais, junto com uma breve análise, estão apresentados na Seção 5.2. A análise dos resultados confronta a solução determinada por métodos clássicos com a solução encontrada por métodos adaptados para solucionar problemas com incerteza.

A implementação da metodologia apresentada na Seção 2.1 foi desenvolvida utilizando o programa MATLAB® 6.1, mais especificamente, a função **fmincon** do *ToolBox Optimization*, para problemas de programação restrita, enquanto a implementação dos métodos iterativos e dos algoritmos genéticos foram desenvolvidos em C++. A máquina que usamos para simular todos os casos foi um Pentium 4, 2.53GHz, com 512Mb de memória RAM.

5.1 Formulação dos problemas

Uma formulação matemática tenta representar, através de funções, as relações existentes em um problema específico, o qual pode representar um caso real ou hipotético. As formulações apresentadas na Sub-Seção 5.1.1 representam problemas reais, enquanto os problemas da Sub-Seção 5.1.2 são formulações de casos hipotéticos.

5.1.1 Incertezas no conjunto de restrições

Apresentamos nesta sub-seção três problemas de programação não-linear com incertezas no conjunto de restrições. Estes problemas foram retirados de (AL-AHMARI 2001, XU 1989).

5.1.1.1 Problema 1: Suporte de teto com três barras

A estrutura de três barras, da Figura a seguir, é um problema de suporte comum usado por alguns pesquisadores na literatura para demonstrar a aplicação do desempenho do algoritmo desenvolvido por eles.

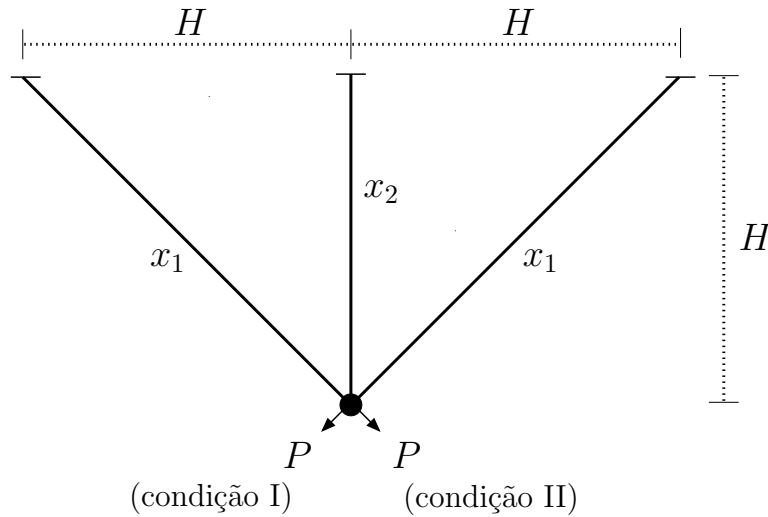


Figura 5.1: Suporte de Teto com Três Barras

Seja $P = 2000\text{kgf}$, $H = 100\text{cm}$, e a densidade $\gamma = 0.01\text{kgf/cm}^3$. Os limites e tolerâncias admissíveis de várias variáveis físicas são dadas na Tabela 5.1 a seguir.

$\sigma^u = 2000\text{ kgf/cm}^2$	$\sigma^l = -1500\text{ kgf/cm}^2$	$u^u = \frac{10^5}{E}\text{ cm}$	$A^l = 0.1\text{ cm}^2$
$T_\sigma^u = 400\text{ kgf/cm}^2$	$T_\sigma^l = 300\text{ kgf/cm}^2$	$T_u^u = \frac{0.2 \times 10^5}{E}\text{ cm}$	$T_A^l = 0.02\text{ cm}^2$

Tabela 5.1: Valores dos limites e tolerâncias para as restrições

O σ representa a força, u é o deslocamento vertical do nó, E representa o módulo elástico, A é a área de secção transversal e T representa a tolerância de cada variável.

O problema de otimização *fuzzy* do suporte de teto com três barras pode ser formulado da seguinte forma:

$$\widetilde{\min} \quad W = H(2\gamma\sqrt{2}x_1 + \gamma x_2) \quad (5.1a)$$

$$s.a \quad \frac{2000(\sqrt{2}x_1 + x_2)}{\sqrt{2}x_1^2 + 2x_1x_2} \lesssim 2000 \quad (5.1b)$$

$$\frac{2000\sqrt{2}x_1}{\sqrt{2}x_1^2 + 2x_1x_2} \lesssim 2000 \quad (5.1c)$$

$$\frac{-2000x_2}{\sqrt{2}x_1^2 + 2x_1x_2} \gtrsim -1500 \quad (5.1d)$$

$$\frac{2 \times 10^5}{(x_1 + \sqrt{2}x_2)E} \lesssim \frac{10^5}{E} \quad (5.1e)$$

$$x_1, x_2 \gtrsim \frac{1}{10} \quad (5.1f)$$

Neste problema, as variáveis de decisão também são *fuzzy*. Portanto, estas variáveis necessitam de níveis de tolerâncias.

5.1.1.2 Problema 2: Anteparo ondulado

O segundo problema refere-se a um anteparo ondulado de casco de navio.

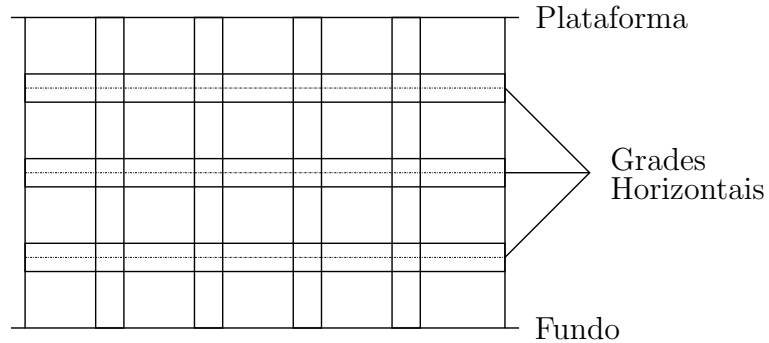


Figura 5.2: Visão lateral do anteparo ondulado

A Figura 5.2 mostra um anteparo transversal verticalmente ondulado para um tanque de 65.000 *dwt* com uma visão frontal, que está construído de acordo com as regras das

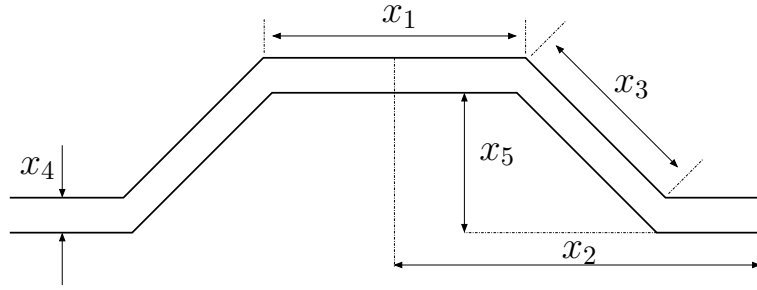


Figura 5.3: Especificação das variáveis do anteparo ondulado

especificações relevantes ao projeto de navios. Os dados do projeto são: largura do anteparo $B = 21.12m$, altura do painel inferior $H = 4.5m$, e densidade $7.85 \times 10^{-3}t/mm - m^2$. As posições das colunas horizontais e a largura do casco são fixadas de modo que a forma da corrugação dependa do painel inferior.

As variáveis do projeto, apresentadas na Figura 5.3, são: x_1 = largura do aro (mm); x_2 = semi-largura da corrugação (mm); x_3 = comprimento da armação (mm); x_4 = espessura da placa no painel inferior (mm); e x_5 = profundidade da corrugação (mm).

A função objetivo examina o peso do painel inferior e está sujeita a 6 (seis) restrições *fuzzy* no módulo da secção (5.2b), o momento de inércia (5.2c), a espessura mínima da chapa do painel inferior (5.2d, 5.2e e 5.2f), uma restrição geométrica (5.2g) e uma restrição clássica que relaciona as dimensões da secção de corrugação (5.2h).

Logo, o problema de otimização fuzzy do casco ondulado pode ser formulado como segue:

$$\widetilde{\min} \quad W = 0.746064(x_1 + x_3)x_4x_2^{-1} \quad (5.2a)$$

$$s.a \quad x_5x_4(3x_1 + x_3)x_2^{-1} \gtrsim 18871.4 \quad (5.2b)$$

$$x_5x_4(3x_1 + x_3)(x_5 + 2x_4)x_2^{-1} \gtrsim 540408 \quad (5.2c)$$

$$x_4 - 0.017297x_1 \gtrsim 2.5 \quad (5.2d)$$

$$x_4 - 0.017297x_3 \gtrsim 2.5 \quad (5.2e)$$

$$x_4 - 5.5 \gtrsim 6.5 \quad (5.2f)$$

$$\arcsen(x_5 x_3^{-1}) \gtrsim 40^\circ \quad (5.2g)$$

$$x_5^2 \leq x_3^2 - (x_2 - x_1)^2 \quad (5.2h)$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Os níveis máximos de tolerância admissíveis para as restrições, 5.2b a 5.2g, do Problema (5.2) estão apresentados, respectivamente, na Tabela 5.2.

$\mathbf{T}_1$	$\mathbf{T}_2$	$\mathbf{T}_3$	$\mathbf{T}_4$	$\mathbf{T}_5$	$\mathbf{T}_6$
3774.28	108081.6	0.5	0.5	1.3	0.14

Tabela 5.2: Níveis de tolerância para as restrições

5.1.1.3 Problema 3: Modelo de processo industrial

O terceiro problema refere-se a um modelo de processo industrial para otimizar os parâmetros e subdivisões da profundidade de corte de uma indústria de manufatura. Um problema de otimização típico de uma operação de máquinas compreende um ou múltiplos objetivos econômicos com várias restrições. O objetivo econômico pode incluir:

- (1) taxa de produção máxima ou tempo de produção mínima;
- (2) custo de produção mínima;
- (3) taxa de lucro máximo.

Nos artigos (AL-AHMARI 2001, LEE et al. 1999, TRAPPEY et al. 1988), o objetivo é minimizar o custo total de produção, sujeito a restrições paramétrica (velocidade de rotação, alimentação e profundidade do corte) e operacionais (rugosidade da superfície, força do corte e potência da máquina). No auxílio a formulação do problema, descrevemos na Tabela 5.3 a notação que é utilizada.

A formulação deste problema, usando o custo mínimo da produção total como critério de otimização, pode ser:

$$\min U = A_2 + A_1 f_s^{\left(\frac{q}{p}-1\right)} d_s^{\left(\frac{r}{p}\right)} + \sum_{i=1}^{i=nor} A_2 + A_1 f_{ri}^{\left(\frac{q}{p}-1\right)} d_{ri}^{\left(\frac{r}{p}\right)} + k_0 t_p \quad (5.3a)$$

A_1, A_2, D, ζ, T_s	Constantes;
C_0	Constante da vida útil da ferramenta de Taylor;
d_s	Profundidade do corte para finalização (mm);
d_{min}	Mínima profundidade do corte permissível (mm);
d_{max}	Máxima profundidade do corte permissível (mm);
d_{ri}	Profundidade do corte do i -ésimo passo da rugosidade (<i>rough pass</i>) (mm);
d_t	Profundidade total do metal para ser removido (mm);
f_s	Alimentação para finalizar ($mmpr$);
f_{ri}	Alimentação para o i -ésimo passo da rugosidade ($mmpr$);
f_{min}	Mínima alimentação permissível ($mmpr$);
f_{max}	Máxima alimentação permissível ($mmpr$);
F_{max}	Máxima força de corte (kgf);
h_1, h_2	Constantes que pertencem ao transporte da ferramenta e tempo de aproximação/parte;
k_0	Custo superior ($\$/min$);
k_1	Constante na equação de força do corte;
k_t	Custo de uma borda de corte ($\$/cutting\ edge$);
L	Comprimento da peça de trabalho (<i>workpiece</i>) (mm);
nor	Número ótimo de passos da rugosidade;
p, q, r	Expoentes de velocidade, alimentação e profundidade do corte na equação da vida útil da ferramenta;
P_{max}	Potência máxima de corte disponível (hp);
R	Raio frontal da ferramenta de corte (mm);
R_{max}	Altura pico-vale da superfície áspera para a passagem final (μm);
R_r	Altura pico-vale da superfície áspera para o passo de rugosidade (μm);
t_e	Tempo requerido para mudar a ferramenta (min);
t_p	Tempo de preparação ($min/piece$);
T_p	Intervalo fixo de tempo após o qual parte da ferramenta é trocada (min);
U	Custo total da produção por unidade ($\$/piece$);
ν_{min}	Velocidade mínima de corte (mpm);
ν_{max}	Velocidade máxima de corte (mpm);
μ, ν	Expoentes de alimentação e profundidade do corte na equação da potência de corte;
η	Eficiência do poder de corte da ferramenta.

Tabela 5.3: Descrição detalhada dos dados do problema 5.3

onde

$$A_1 = \frac{\pi D L k_0}{1000 T_p} \left[\frac{T_s}{C_0} \right]^{\frac{1}{p}} \left[T_s + \zeta \left(t_e + \frac{k_t}{k_0} \right) \right] \quad A_2 = k_0 (h_1 L + h_2) \quad T_p = \frac{T_s}{\zeta}$$

Sujeito às seguintes restrições:

(1) Velocidade mínima

$$f_s^{(\frac{q}{p})} d_s^{(\frac{r}{p})} \gtrsim \frac{C_0^{(\frac{1}{p})}}{T_p^{(\frac{1}{p})} v_{max}} \quad f_{ri}^{(\frac{q}{p})} d_{ri}^{(\frac{r}{p})} \gtrsim \frac{C_0^{(\frac{1}{p})}}{T_p^{(\frac{1}{p})} v_{max}} \quad (5.3b)$$

(2) Velocidade máxima

$$f_s^{(\frac{q}{p})} d_s^{(\frac{r}{p})} \lesssim \frac{C_0^{(\frac{1}{p})}}{T_p^{(\frac{1}{p})} v_{min}} \quad f_{ri}^{(\frac{q}{p})} d_{ri}^{(\frac{r}{p})} \lesssim \frac{C_0^{(\frac{1}{p})}}{T_p^{(\frac{1}{p})} v_{min}} \quad (5.3c)$$

(3) Alimentação e superfície mínima

$$f_s \gtrsim f_{min} \quad f_{ri} \gtrsim f_{min} \quad (5.3d)$$

(4) Alimentação e superfície máxima

$$f_s \lesssim \min(f_{max}, \sqrt{8RR_{max}}) \quad f_{ri} \lesssim \min(f_{max}, \sqrt{8RR_r}) \quad (5.3e)$$

(5) Força de corte

$$k_1 f_s^\mu d_s^\mu \lesssim F_{max} \quad k_1 f_{ri}^\mu d_{ri}^\mu \lesssim F_{max} \quad (5.3f)$$

(6) Potência do corte

$$f_s^{(\mu - \frac{q}{p})} d_s^{(\nu - \frac{r}{p})} \lesssim \frac{6120 \eta T_p^{(\frac{1}{p})} P_{max}}{k_1 C_0^{(\frac{1}{p})}} \quad f_{ri}^{(\mu - \frac{q}{p})} d_{ri}^{(\nu - \frac{r}{p})} \lesssim \frac{6120 \eta T_p^{(\frac{1}{p})} P_{max}}{k_1 C_0^{(\frac{1}{p})}} \quad (5.3g)$$

(7) Profundidade do corte mínimo

$$d_s \gtrsim d_{min} \quad d_{ri} \gtrsim d_{min} \quad (5.3h)$$

(8) Profundidade do corte máximo

$$d_s \lesssim d_{max} \quad d_{ri} \lesssim d_{max} \quad (5.3i)$$

(9) Profundidade total

$$d_t = d_s + \sum_{i=1}^{nor} d_{ri} \quad (5.3j)$$

$D = 50 \text{ mm}$	$p = 5$	$\zeta = 1$	$h_1 = 7 \times 10^{-4}$
$d_t = 60 \text{ mm}$	$q = 1.75$	$\mu = 0.75$	$h_2 = 0.3$
$d_{min} = 1.0 \text{ mm}$	$r = 0.75$	$\nu = 0.95$	$C_0 = 6 \times 10^{11}$
$d_{max} = 3.0 \text{ mm}$	$k_0 = \$0.5/min$	$P_{max} = 5 \text{ kW}$	$F_{max} = 200 \text{ kgf}$
$f_{min} = 0.1 \text{ mm/rev}$	$k_1 = 108$	$R = 1.2 \text{ mm}$	$T_{min} = 25 \text{ min}$
$f_{max} = 0.9 \text{ mm/rev}$	$k_t = \$2.5/cutting \text{ edge}$	$R_r = 100 \text{ }\mu\text{m}$	$T_{max} = 45 \text{ min}$
$\nu_{min} = 5 \text{ m/min}$	$t_p = 0.75 \text{ min/piece}$	$R_{max} = 10 \text{ }\mu\text{m}$	$T_s = 25 \text{ min}$
$\nu_{max} = 500 \text{ m/min}$	$t_e = 1.5 \text{ mm/edge}$	$L = 300 \text{ mm}$	

Restrições	Tolerâncias		Restrições	Tolerâncias	
5.3b	T_1	4	5.3f	T_5	10
5.3c	T_2	200	5.3g	T_6	4
5.3d	T_3	0	5.3h	T_7	0
5.3e	T_4	1	5.3i	T_8	0

Tabela 5.4: Níveis máximos de tolerância

Os dados do problema são os seguintes:

As especificações da violação máxima de tolerância das incertezas de cada restrição deste exemplo estão sumarizadas na Tabela 5.4.

As idéias para a formulação dos dados de incerteza deste exemplo numéricos podem ser encontradas em (LEE et al. 1999, TRAPPEY et al. 1988).

5.1.2 Incertezas na função objetivo e no conjunto de restrições

Nesta sub-seção apresentamos 6 (seis) problemas de programação matemática, dos quais três pertencem ao conjunto dos problemas com restrições somente de desigualdade, um ao grupo com restrição somente de igualdade e os demais são com restrições de igualdade e de desigualdade, isto é, restrições mistas.

Nestes problemas as incertezas nos dados foram inseridas nas constantes da função objetivo com uma variação de 10% em torno do valor modal, por exemplo, o número $\tilde{2}$ pode variar em 0.2 unidades para mais e/ou para menos. A solução ótima dos problemas, expostos aqui, sem inserir incertezas em seus dados estão apresentados nas colunas $\bar{\mathbf{x}}^T$ e $f(\bar{\mathbf{x}}^T)$.

5.1.2.1 Problemas com restrições de desigualdade

Os problemas com restrições de desigualdade, descritos na Tabela 5.5, foram retirados de SCHITTKOWSKI (1987).

5.1.2.2 Problemas com restrições de igualdade

O problema com restrição de igualdade, mostrado na Tabela 5.6, foi retirado da referência (BAZARAA et al. 1993).

5.1.2.3 Problemas com restrições de igualdade e desigualdade

Os problemas com conjunto de restrições mistas, apresentados na Tabela 5.7 estão referenciados em HIMMELBLAU (1972).

5.2 Resultados e análise

As soluções dos problemas descritos nas Tabelas 5.5, 5.6 e 5.7 estão apresentados nesta seção, em que os mesmos serão resolvidos pelos métodos adaptados do Capítulo 3 e dos algoritmos genéticos desenvolvidos no Capítulo 4, que estão vinculados a uma breve análise dos mesmos. De acordo com a discussão feita no Capítulo 4, podemos comparar os resultados dos algoritmos adaptados, desenvolvidos neste trabalho, com os resultados obtidos pelos métodos clássicos sem a inserção dos parâmetros *fuzzy*.

5.2.1 Incertezas no conjunto de restrições

Nesta sub-seção estão discriminados os resultados dos problemas reais de programação matemática descritos na Sub-seção 5.1.1, resolvidos utilizando os métodos apresentados no

Prob.	$f(\tilde{a}; \mathbf{x})$	Variação <i>Fuzzy</i>	$x_{inicial}$	Restrições	Violação	Solução clássica	
						$\bar{\mathbf{x}}^T$	$f(\bar{\mathbf{x}}^T)$
PG1	$(x_1 - \tilde{2})^2 + (x_2 - \tilde{1})^2$	10%	$[0.5, 0.5]^T$	$g_1(x) = x_1^2 - x_2 \lesssim 0$ $g_2(x) = x_2^2 - x_1 \lesssim 0$	$T_1 = 0.35$ $T_2 = 0.35$	$[1, 1]^T$	1
PG2	$x_1^2 + x_2^2 - \tilde{4}x_1 + \tilde{4}$	10%	$[1.0, 2.5]^T$	$g_1(x) = x_2 - x_1 - 2 \lesssim 0$ $g_2(x) = x_1^2 - x_2 + 1 \lesssim 0$ $g_3(x) = -x_1 \leq 0$ $g_4(x) = -x_2 \leq 0$	$T_1 = 1.0$ $T_2 = 0.5$ $T_3 = 0.0$ $T_4 = 0.0$	$[0.5536, 1.306]^T$	3.79894
PG3	$\tilde{9}x_1^2 + x_2^2 + \tilde{9}x_3^2$	10%	$[1.0, 1.0, 1.0]^T$	$g_1(x) = 1 - x_1x_2 \lesssim 0$ $g_2(x) = -x_2 \leq 0$ $g_3(x) = x_3 \leq 0$	$T_1 = 0.5$ $T_2 = 0.0$ $T_3 = 0.0$	$[0.5774, 1.732, -0.2 \times 10^{-5}]^T$	6

Tabela 5.5: Problemas com restrições somente de desigualdade

Prob.	$f(\tilde{a}; \mathbf{x})$	Variação <i>Fuzzy</i>	$x_{inicial}$	Restrições	Violação	Solução clássica	
						$\bar{\mathbf{x}}^T$	$f(\bar{\mathbf{x}}^T)$
PH1	$(x_1 - \tilde{2})^4 + (x_1 - \tilde{2}x_2)^2$	10%	$[0.0, 0.0]^T$	$h_1(x) = x_1^2 - x_1 \cong 0$	$T_1 = 0.5$	$[0.94611, 0.89344]^T$	1.9433

Tabela 5.6: Problema com restrição somente de igualdade

Prob.	$f(\tilde{a}; \mathbf{x})$	Variação <i>Fuzzy</i>	$x_{inicial}$	Restrições	Violação	Solução clássica	
						$\bar{\mathbf{x}}^T$	$f(\bar{\mathbf{x}}^T)$
PM1	$1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$	10%	$[3.0, 0.3, 4.0]^T$	$h_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 \cong 0$ $h_2(x) = 8x_1 + 14x_2 + 7x_3 - 56 \cong 0$ $g_1(x) = -x_1 \leq 0$ $g_2(x) = -x_2 \leq 0$ $g_3(x) = -x_3 \leq 0$	$T_1 = 2.5$ $T_2 = 5.6$ $T_3 = 0.0$ $T_4 = 0.0$ $T_5 = 0.0$	$[3.512, 0.217, 3.552]^T$	961.715
PM2	$(x_1 - 2)^2 + (x_2 - 1)^2$	10%	$[0.0, 0.5]^T$	$h_1(x) = x_1 - 2x_2 + 1 \cong 0$ $g_1(x) = x_1^2/4 - x_1^2 - 1 \lesssim 0$	$T_1 = 0.5$ $T_2 = 0.5$	$[0.823, 0.911]^T$	1.393

Tabela 5.7: Problemas com restrições de igualdade e desigualdade

Capítulo 2. Junto aos resultados está uma análise que confronta a solução dos métodos descritos no Capítulo 2, que resolvem problemas com incertezas somente no conjunto de restrições, com a solução dos problemas sem relaxar nenhuma das restrições.

5.2.1.1 Problema 1: Suporte de teto com três barras

Na Tabela 5.8 são apresentadas as soluções para os diferentes valores de b_0 , utilizando o método da Seção 2.1.1. A análise mais superficial que podemos estabelecer refere-se à sequência crescente dos valores de b_0 , em que os valores da função objetivo, referente a um único b_0 , também formam uma sequência crescente. Outro ponto importante é o número de iterações das simulações nas quais obteve-se o menor número quando $b_0 = T_0$, isto é, a solução ótima do problema não-linear clássico. O tempo de processamento para cada valor de b_0 foi em média 0.04 segundos.

b_0	2.10	2.20	2.30	2.3570	2.40	2.50	2.60	2.70	2.80	2.8284
S	0.7839	0.8137	0.8435	0.8605	0.8733	0.9029	0.9325	0.9621	0.9916	1.0000
$\mathbf{x}_1$	0.6390	0.6427	0.6464	0.6486	0.6502	0.6540	0.6578	0.6617	0.6656	0.6667
$\mathbf{x}_2$	0.9037	0.9089	0.9142	0.9172	0.9195	0.9249	0.9303	0.9357	0.9412	0.9428
FunObj	2.7112	2.7268	2.7426	2.7516	2.7585	2.7746	2.7908	2.8072	2.8237	2.8284
Iter.	6	6	6	6	6	5	5	5	5	5

Tabela 5.8: Resultados da implementação do método da Seção 2.1.1

A Tabela 5.9 apresenta o resultado da parametrização do α , que nos fornece os limitantes inferior, $m = 2.3570$, e superior, $M = 2.8284$, do problema quando fazemos α iguais a 0 (zero) e 1 (um), respectivamente, concluindo, desta forma, a primeira fase do método da Seção 2.1.2. O processamento da primeira fase desse método durou 0.35 segundos. Logo, encontramos um conjunto de soluções factíveis que será utilizada na segunda fase para determinar a solução do problema, que é apresentada na Tabela 5.10. Contudo, neste exemplo (5.1) temos como avaliação do valor da função objetivo, que é linear, com restrições não lineares, e portanto os valores encontrados são muito próximos, pois a função é estritamente convexa.

α	x_1	x_2	FunObj	Iter.
0.0	0.5556	0.7857	2.3570	6
0.1	0.5650	0.7990	2.3970	6
0.2	0.5747	0.8128	2.4383	6
0.3	0.5848	0.8270	2.4811	6
0.4	0.5952	0.8418	2.5254	5
0.5	0.6061	0.8571	2.5713	5
0.6	0.6173	0.8730	2.6189	5
0.7	0.6289	0.8894	2.6683	5
0.8	0.6410	0.9065	2.7196	5
0.9	0.6536	0.9243	2.7730	5
1.0	0.6667	0.9428	2.8284	5

Tabela 5.9: Resultados da implementação do método da Seção 2.1.2 - Primeira fase

De acordo com a Seção 2.1.3, para obter os resultados da Tabela 5.8, o valor T_0 é igual ao **FunObj** da Tabela 5.9, para $\alpha = 1.0$. Nas colunas com o valor de b_0 iguais a 2.3570 e 2.8284 da Tabela, pode-se notar a semelhança com o resultado ótimo da Tabela 5.10 e o resultado ótimo da formulação clássica deste exemplo, respectivamente.

α	x_1	x_2	FunObj	μ_G	Tol	Iter	Tempo
0.8614	0.6487	0.9174	2.7522	0.8564	0.0050	12	17.0s

Tabela 5.10: Resultados da implementação do método da Seção 2.1.2 - Segunda fase

5.2.1.2 Problema 2: Anteparo ondulado

Na Tabela 5.11 são apresentadas as soluções para os diferentes valores de b_0 , utilizando o método da Seção 2.1.1. Os valores da função objetivo, referente a um único b_0 , formam uma seqüência crescente, exceto quando $b_0 = 11.00$. O motivo para tal situação deve-se ao método de resolução não ser globalmente convergente, e este ter encontrado mínimo local, e concluído esta simulação. Outro ponto importante é o número de iterações das simulações, em que o segundo valor de iterações é menor que o primeiro e o terceiro é o maior valor, confirmando a explicação do mínimo local. As outras simulações mostram, gradativamente, números decrescentes de iterações até obter o menor número, quando $b_0 = T_0$.

A solução da primeira fase é apresentada na Tabela 5.12, em que os limitantes inferior, $m = 11.6025$, e superior, $M = 13.1114$, para valores de α iguais a 0(zero) e 1(um), formam o conjunto solução que é utilizado na segunda fase. Estes limitantes são elementos fundamentais para a resolução do método duas fases. Agora é utilizada uma busca por bissecções pelos

$\mathbf{b}_0$	10.50	11.00	11.50	11.6025	12.00	12.50	13.00	13.1114
$\mathbf{S}$	0.8207	0.8550	0.8893	0.8964	0.9237	0.9580	0.9923	1.0000
$\mathbf{x}_1$	636.7076	637.4339	636.4198	637.8867	638.4580	639.1737	639.8861	640.0448
$\mathbf{x}_2$	992.3444	991.8308	986.4230	990.0337	989.6224	989.1047	988.5861	988.4701
$\mathbf{x}_3$	636.7076	637.4339	636.4198	637.8867	638.4580	639.1737	639.8861	640.0448
$\mathbf{x}_4$	13.4235	13.4532	13.4528	13.4817	13.5052	13.5348	13.5643	13.5709
$\mathbf{x}_5$	528.1279	529.8348	531.5335	531.8758	533.2093	534.8751	536.5282	536.8958
FunObj	12.8513	13.0367	12.9509	12.9612	13.0008	13.0507	13.1006	13.1117
Iter.	10	9	9	9	8	10	6	6

Tabela 5.11: Resultados da implementação do método da Seção 2.1.1

limites do conjunto dado e depois de 4 iterações consegue-se chegar ao ponto ótimo com um valor de μ igual a 0.8949, mostrado na Tabela 5.13.

α	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	FunObj	Iter.
0.0	579.0948	894.9252	579.0948	12.0166	485	11.6025	15
0.1	585.8749	905.7204	585.8749	12.1839	491	11.7599	14
0.2	592.5631	916.3764	592.5631	12.3496	496	11.9157	14
0.3	599.1656	926.9055	599.1656	12.5138	502	12.0700	14
0.4	605.6791	937.2951	605.6791	12.6764	507	12.2228	14
0.5	612.1142	947.5706	612.1142	12.8377	512	12.3742	12
0.6	618.4655	957.7135	618.4655	12.9976	517	12.5311	7
0.7	624.7403	967.7411	624.7403	13.1561	522	12.6762	7
0.8	630.9446	977.6670	630.9446	13.3134	527	12.8213	7
0.9	637.0794	987.4907	637.0794	13.4696	532	12.9665	7
1.0	643.1377	997.1899	643.1377	13.6244	537	13.1114	16

Tabela 5.12: Resultados da implementação do método da Seção 2.1.2 - Primeira fase

Nas colunas com o valor de b_0 iguais a 11.6025 e 13.1114 da Tabela 2.1.1, pode-se notar a semelhança com o resultado ótimo da Tabela 5.13 e o resultado ótimo da formulação clássica deste exemplo, respectivamente. Obtém-se os resultados da Tabela 5.11, da forma como foi provado na seção anterior, com o valor T_0 igual ao **FunObj** da Tabela 5.12, para $\alpha = 1.0$.

α	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	FunObj	μ_G	Tol	Iter
0.8996	637.0512	987.4424	637.0512	13.4689	532.0340	12.9658	0.8949	0.0047	36

Tabela 5.13: Resultados da implementação do método da Seção 2.1.2 - Segunda Fase

5.2.1.3 Problema 3: Modelo de processo industrial

A Tabela 5.14 mostra uma solução ótima encontrada quando as restrições do Exemplo Numérico (5.3) da seção anterior estão com as tolerâncias totalmente violadas, e o número de iterações necessárias para a sua convergência, utilizando o método da Seção 2.1.1.

Modelo	d_t (mm)	f_s (mm)	f_1 (mm)	f_2 (mm)	f_3 (mm)	d_s (mm)	d_1 (mm)	d_2 (mm)	d_3 (mm)	U (\$/piece)	Iter.
Modelo	6	1.31	1.9	-	-	2.999	3.001	-	-	1.325	5
Fuzzy para	8	1.31	1.9	1.9	-	1.999	3.000	3.001	-	1.759	28
os casos das	8.5	1.31	1.9	1.9	-	2.500	3.000	3.000	-	1.767	19
restrições	9	1.31	1.9	1.9	-	3.000	3.000	3.000	-	1.773	5
totalmente	9.5	1.31	1.9	1.9	1.9	1.000	2.833	2.833	2.833	2.179	39
violadas	10	1.31	1.9	1.9	1.9	1.000	3.000	3.000	3.000	2.184	41

Tabela 5.14: Resultados do exemplo numérico 5.3 com as restrições totalmente violadas

A solução ótima do modelo de Al-Ahmari, apresentado na Tabela 5.15, foi retirado de (AL-AHMARI 2001). Esta solução foi obtida utilizando o programa LINGO, contudo esta referência não relata as iterações necessárias para a convergência do problema em todos parâmetros d_t avaliados. Assim, este modelo está exposto somente para confrontar o ponto ótimo e o custo com os demais modelos.

Modelo	d_t (mm)	f_s (mm)	f_1 (mm)	f_2 (mm)	f_3 (mm)	d_s (mm)	d_1 (mm)	d_2 (mm)	d_3 (mm)	nor	U (\$/piece)
Modelo de Al-Ahmari	6	0.310	0.566	-	-	3.000	3.000	-	-	1	1.939
	8	0.310	0.900	0.585	-	3.000	2.079	2.921	-	2	2.481
	8.5	0.310	0.713	0.566	-	3.000	2.500	3.000	-	2	2.550
	9	0.310	0.566	0.566	-	3.000	3.000	3.000	-	2	2.619
	9.5	0.310	0.774	0.900	0.900	3.000	2.342	2.079	2.079	3	2.953
	10	0.310	0.606	0.900	0.900	3.000	2.842	2.079	2.079	3	3.022
Modelo Clássico	6	0.310	0.566	-	-	3.000	3.000	-	-	1	1.939
	8	0.310	0.713	0.712	-	3.000	2.500	2.500	-	2	2.481
	8.5	0.310	0.632	0.632	-	3.000	2.750	2.750	-	2	2.550
	9	0.310	0.566	0.566	-	3.000	3.000	3.000	-	2	2.619
	9.5	0.310	0.854	0.854	0.854	3.000	2.167	2.167	2.167	3	2.953
	10	0.310	0.777	0.777	0.777	3.000	2.334	2.334	2.332	3	3.022
Modelo Fuzzy de Trappey et. al.	6	0.474	1.064	-	-	3.000	3.000	-	-	1	1.644
	8	0.470	1.060	1.060	-	2.000	3.000	3.000	-	2	2.156
	8.5	0.469	1.060	1.060	-	2.500	3.000	3.000	-	2	2.173
	9	0.468	1.058	1.058	-	3.000	3.000	3.000	-	2	2.187
	9.5	0.468	1.058	1.058	1.058	1.000	2.833	2.833	2.833	3	2.645
	10	0.466	1.056	1.056	1.056	1.000	3.000	3.000	3.000	3	2.654
Modelo Fuzzy de Xu	6	0.493	1.083	-	-	3.000	3.000	-	-	1	1.629
	8	0.485	1.075	1.075	-	2.000	3.000	3.000	-	2	2.143
	8.5	0.487	1.077	1.077	-	2.500	3.000	3.000	-	2	2.156
	9	0.485	1.075	1.075	-	3.000	3.000	3.000	-	2	2.170
	9.5	0.477	1.067	1.067	1.067	1.000	2.833	2.833	2.833	3	2.635
	10	0.478	1.068	1.068	1.068	1.000	3.000	3.000	3.000	3	2.641

Tabela 5.15: Resultado do exemplo numérico 5.3 dos modelos para comparação

Na Tabela 5.16 é mostrado o nível de satisfação alcançado pelos modelos *fuzzy* juntamente com o número de restrições dos mesmos. Nesta tabela ainda apresentamos o número de iterações necessários para solucionar o modelo clássico.

Na análise destes resultados, pode-se constatar que os modelos *fuzzy* precisam de um

Modelo	d_t	μ	iterações
Modelo Clássico	6	100%	3
	8	100%	5
	8.5	100%	5
	9	100%	3
	9.5	100%	6
	10	100%	6
Modelo <i>Fuzzy</i> de Trappey et. al.	6	83.56%	13
	8	83.97%	51
	8.5	84.07%	34
	9	84.20%	14
	9.5	84.23%	56
	10	84.44%	58
Modelo <i>Fuzzy</i> de Xu	6	81.34%	14
	8	82.08%	43
	8.5	81.95%	33
	9	81.70%	12
	9.5	82.70%	56
	10	82.70%	58

Tabela 5.16: Análise dos resultados do exemplo numérico 5.3

número de iterações superior aos modelos clássicos, porém o valor da função objetivo de ambos os modelos *fuzzy* são menores. Logo, a análise destes resultados podem ser importantes para a escolha do modelo a ser utilizado.

5.2.2 Incertezas na função objetivo e no conjunto de restrições

Nesta sub-seção mostramos os resultados obtidos nos problemas apresentados na Sub-seção 5.1.2, os quais foram solucionados pelos métodos adaptados apresentados nos Capítulos 3 e 4. Na Tabela 5.17 estão descritos os resultados ótimos dos problemas de duas formas: (i) restrições totalmente satisfeitas, que solucionam problemas atendendo rigorosamente as restrições, isto é, não permitindo o relaxamento de nenhuma delas; (ii) restrições totalmente violadas, que representa o oposto da forma acima mencionada, isto é, todas as restrições permitem relaxamento mediante um nível de tolerância inserido para representar a condição *fuzzy* presente nas mesmas. Os dados da Tabela 5.17 mostram os resultados do problema (3.5) impondo o parâmetro $\alpha = 1$, no caso sem permitir violação, e $\alpha = 0$, no caso totalmente violado.

Mediante os resultados apresentados na Tabela 5.17 podemos calcular o nível de satisfação mínimo para cada um dos problemas de programação matemática *fuzzy* com incertezas na função objetivo e no conjunto de restrições apresentados nestes trabalhos. O intervalo de

Prob.	Restrição	Mínimo de $f(\tilde{a}; x^*)$			Tempo
		x^*	$f(\tilde{x}; x^*)$	$F(f(\tilde{a}; x^*))$	
PG1	Totalmente Satisfeitas	$[1.0005, 1.0004]^T$	$[0.80655, 0.99905, 1.2115]$	1.0041	14s
	Totalmente Violadas	$[1.2074, 1.2073]^T$	$[0.50443, 0.67117, 0.86292]$	0.67739	8s
PG2	Totalmente Satisfeitas	$[0.55304, 1.3052]^T$	$[3.1761, 3.7973, 4.4185]$	3.7973	18s
	Totalmente Violadas	$[0.57687, 1.2322]^T$	$[2.9128, 3.5436, 4.1743]$	3.5436	70s
PG3	Totalmente Satisfeitas	$[0.57802, 1.7287, -0.8 \times 10^{-12}]^T$	$[5.6945, 5.9952, 6.2959]$	5.9952	37s
	Totalmente Violadas	$[0.54934, 1.6369, -0.1 \times 10^{-11}]^T$	$[5.1237, 5.3953, 5.6669]$	5.3953	53s
PH1	Totalmente Satisfeitas	$[0.9506, 0.90276]$	$[0.9753, 1.9436, 3.5089]$	2.0926	1s
	Totalmente Violadas	$[1.0333, 0.81706]$	$[0.53695, 1.2344, 2.4371]$	1.3606	2s
PM1	Totalmente Satisfeitas	$[3.6735, 0.2074, 3.3868]^T$	$[951.75, 961.75, 971.75]$	961.75	-
	Totalmente Violadas	-	$[720.0, 800.0, 880.0]$	800.0	-
PM2	Totalmente Satisfeitas	$[0.8238, 0.9116]^T$	$[0.9508, 1.3912, 1.9294]$	1.416	-
	Totalmente Violadas	-	$[1.045, 1.15, 1.265]$	1.15	-

Tabela 5.17: Níveis máximos e mínimos de tolerância para o conjunto de restrições

satisfação é determinado pelas equações descritas em (3.9). Os traços que estão nessa tabela informam que os resultados não foram simulados. Os resultados das situações totalmente satisfeitas dos problemas PM1 e PM2 foram retirados da mesma fonte, enquanto que as situações totalmente violadas foram supostas.

A principal análise a ser feita diante de todos os resultados aqui apresentados está na escolha da relação entre o valor da função objetivo e o nível de satisfação calculado. Esta escolha depende do **decisor**, pois o conhecimento prévio do objetivo principal a ser alcançado lhe guiará a uma escolha apropriada.

Método	Mínimo de $f(\tilde{a}; x^*)$				Tempo
	x^*	$f(\tilde{x}; x^*)$	$\mathbb{D}f(f(\tilde{a}; x^*))$	μ	
Adaptação de Zimmermann	$[1.0535, 1.0534]^T$	$[0.71657, 0.89871, 1.1059]$	0.90494	0.77643	39s
Adaptação de Xu	$[1.058, 1.0578]^T$	$[0.70907, 0.89076, 1.0974]$	0.89698	0.75749	4s
Terceira Adaptação	$[1.0005, 1.0004]^T$	$[0.80655, 0.99895, 1.2114]$	1.004	1.0000	3s
Genético Puro	$[1.06081, 1.04544]^T$	$[0.53847, 0.88415, 1.3189]$	0.90706	0.759117	12s
Genético com Busca Local	$[1.03289, 1.00209]^T$	$[0.57862, 0.93551, 1.3728]$	0.95563	0.717547	79s

Tabela 5.18: Resultado para o problema PG1

A Tabela 5.18 mostra que o algoritmo genético com busca local foi o mais demorado, seguido pelo método adaptado de Zimmermann. A terceira adaptação, referente ao método de penalização descrito no Capítulo 3, apresenta o maior valor defuzzyficado da função objetivo, porém o nível de satisfação é máximo, isto é, não permite violação nas restrições, e conseguiu resolver o problema PG1 de forma mais rápida. O método adaptado de Xu fornece o menor valor defuzzyficado de função objetivo e tanto o nível de satisfação quanto a velocidade de processamento são admissíveis.

A Tabela 5.19 pode-se constatar que os resultados do valor defuzzyficado da função objetivo e do nível de satisfação são muito parecidos entre si. A terceira adaptação ainda continua

Método	Mínimo de $f(\tilde{a}; x^*)$				Tempo
	x^*	$f(\tilde{x}; x^*)$	$\mathbb{D}f(f(\tilde{a}; x^*))$	μ	
Adaptação de Zimmermann	$[0.55094, 1.2965]^T$	$[3.1603, 3.7807, 4.4011]$	3.7807	0.93768	2s
Adaptação de Xu	$[0.55473, 1.3015]^T$	$[3.1607, 3.7826, 4.4045]$	3.7826	0.9361	11s
Terceira Adaptação	$[0.55177, 1.3038]^T$	$[3.1765, 3.7972, 4.4179]$	3.7972	1.0000	5s
Genético Puro	$[0.522019, 1.2306]^T$	$[2.9113, 3.6988, 4.4863]$	3.6988	0.958039	11s
Genético com Busca Local	$[0.54986, 1.29312]^T$	$[2.9584, 3.7757, 4.5931]$	3.7757	0.938688	14s

Tabela 5.19: Resultado para o problema PG2

sendo o método que prioriza estritamente o nível de satisfação, porém o valor do objetivo sofre pouca modificação entre todos.

Método	Mínimo de $f(\tilde{a}; x^*)$				Tempo
	x^*	$f(\tilde{x}; x^*)$	$\mathbb{D}f(f(\tilde{a}; x^*))$	μ	
Adaptação de Zimmermann	$[0.57749, 1.6907, 0.9 \times 10^{11}]^T$	$[5.5598, 5.8599, 6.1601]$	5.8599	0.92252	3s
Adaptação de Xu	$[0.5764, 1.7191, -0.2 \times 10^{18}]^T$	$[5.6464, 5.9454, 6.2444]$	5.9454	0.91687	5s
Terceira Adaptação	$[0.57935, 1.7247, -0.2 \times 10^{24}]^T$	$[5.6932, 5.9953, 6.2974]$	5.9953	1.0000	9s
Genético Puro	$[0.55586, 1.73454, 0.082679]^T$	$[5.2659, 5.851, 6.4361]$	5.8663	0.919717	8s
Genético com Busca Local	$[1.0, 1.0, 1.0]^T$	$[17.1, 19, 20.9]$	5.851	0.92212	16s

Tabela 5.20: Resultado para o problema PG3

A Tabela 5.20 mostra que o nível de satisfação de todos os métodos estão acima de 90%. O tempo de processamento do genético com busca local foi o mais demorado, porém este forneceu o menor valor defuzzyficado da função objetivo. Observando os tempos, os níveis de satisfação e o valor de objetivo, a adaptação de Zimmermann foi a que forneceu melhores resultados.

Método	Mínimo de $f(\tilde{a}; x^*)$				Tempo
	x^*	$f(\tilde{x}; x^*)$	$\mathbb{D}f(f(\tilde{a}; x^*))$	μ	
PH1	$[0.946748, 0.828629]^T$	$[0.82683, 1.7355, 3.2347]$	1.8829	0.828629	12s
PM1	$[3.0582, 0.321026, 3.99534]^T$	$[861.26, 961.28, 1061.3]$	961.28	0.832225	18s
PM2	$[0.782109, 0.893693]^T$	$[0.83529, 1.2714, 1.8075]$	1.2963	0.90452	12s

Tabela 5.21: Resultado usando o algoritmo genético puro

A Tabela 5.21 apresenta os resultados obtidos dos problemas com restrição somente de igualdade e os com restrições de igualdade e desigualdade usando o algoritmo genético puro. Os resultados encontrados dos problemas PH1, PM1 e PM2 obtiveram um nível de satisfação acima de 80%. Os valores defuzzyficados da função objetivo de cada problema foi inferior ao apresentado na Tabela 5.17.

Capítulo 6

Considerações Finais

6.1 Conclusões

O tratamento do problema (3.1) foi a proposta principal para o desenvolvimento deste trabalho. Para tanto, iniciamos um estudo mais detalhado sobre problemas de programação não-linear *fuzzy*, nos quais a condição *fuzzy* está presente tanto na função objetivo como no conjunto de restrições, descrito em (LEE et al. 1999, TRAPPEY et al. 1988, XU 1989). Mediante o estudo apresentado nas Seções 2.1.1 e 2.1.2, podemos concluir que existe uma relação de igualdade, descrita em 2.1.3, entre a variável $\mathbf{S}$, descrita nos métodos de (LEE et al. 1999, TRAPPEY et al. 1988), e o parâmetro α , do método descrito em (XU 1989). Esse resultado nos fornece uma ferramenta fundamental de análise na determinação da solução ótima desejada. Esse estudo gerou duas publicações submetidas e aceitas em congressos nacionais da área, (SILVA & CANTÃO 2004, SILVA & YAMAKAMI 2004). Porém, ainda não foi proposto um método eficiente e, ao mesmo tempo, de fácil implementação para resolver diversos tipos destes problemas não-lineares. Estudamos a base teórica desenvolvida por CANTÃO (2003) e os critérios de otimalidade de programação não-linear com parâmetros *fuzzy* na função objetivo. Esta referência ainda nos apresenta métodos que solucionam problemas do tipo (2.14). Portanto, unindo estas duas características de incertezas estudadas, conseguimos desenvolver 3 (três) métodos iterativos, descritos no Capítulo 3, e 2 (dois) algoritmos genéticos, descritos no Capítulo 4, que solucionam os problemas propostos.

Os 3 (três) métodos iterativos adaptados que usam a derivada da função objetivo como direção ótima e os 2 (dois) algoritmos genéticos, tanto o puro quanto o com busca local, desenvolvidos neste trabalho, apresentaram bons resultados para problemas hipotéticos. Os resultados obtidos, pelos métodos e algoritmos desenvolvidos neste trabalho, foram melhores

que os ótimos clássicos fornecidos pela literatura, porém com um nível de satisfação inferior a 100%, isto é, a solução encontrada permite violação em uma ou algumas restrições do problema a ser otimizado.

Um ponto principal na construção dos algoritmos genéticos foi a forma de tratar os problemas matemáticos com restrições de igualdade. O uso da função módulo para transformar uma restrição de igualdade numa restrição de desigualdade, de forma que esta última herde todas as características da restrição de igualdade, propiciou um esforço computacional equivalente ao problema original. Contudo, este artifício não pôde ser usado nos métodos iterativos porque a função módulo não é diferenciável.

6.2 Trabalhos futuros

Apesar de todo o conhecimento que foi tratado e desenvolvido neste trabalho sobre problemas de programação matemática *fuzzy*, ainda faltam alguns pontos a serem estudados com mais riqueza de detalhes. Entre estes está o tratamento de problemas matemáticos, nos quais a inserção de parâmetros *fuzzy* seja de forma completa. Assim, o Problema (1.1) pode ser reescrito da seguinte forma:

$$\begin{aligned}
 \min \quad & f(\tilde{\mathbf{a}}_1; \tilde{\mathbf{x}}) \\
 s.a \quad & \mathbf{g}(\tilde{\mathbf{a}}_2; \tilde{\mathbf{x}}) \lesssim \tilde{\mathbf{b}} \\
 & \mathbf{h}(\tilde{\mathbf{a}}_3; \tilde{\mathbf{x}}) \cong \tilde{\mathbf{c}}, \\
 & \tilde{\mathbf{x}} \in \Omega
 \end{aligned} \tag{6.1}$$

onde $\Omega \subseteq \mathbb{F}(\mathbb{R}^n)$, $\mathbf{f} : \mathbb{F}(\mathbb{R}^n) \rightarrow \mathbb{F}(\mathbb{R})$, $\mathbf{g} : \mathbb{F}(\mathbb{R}^n) \rightarrow \mathbb{F}(\mathbb{R}^k)$, $\tilde{\mathbf{b}} \in \mathbb{F}(\mathbb{R}^k)$, $\mathbf{h} : \mathbb{F}(\mathbb{R}^n) \rightarrow \mathbb{F}(\mathbb{R}^l)$ e $\tilde{\mathbf{c}} \in \mathbb{F}(\mathbb{R}^l)$

Uma vez desenvolvida esta linha de pesquisa da expansão dos problemas de programação matemática com parâmetros *fuzzy* formalizados matematicamente em (6.1), podemos explorar a dualidade dos mesmos.

Um outro ponto onde a programação matemática *fuzzy* pode auxiliar na resolução de problemas clássicos não-convexos está na possibilidade de se trabalhar com a casca convecxa da região factível, obtida pela relaxação das restrições. Logo, o conhecimento das características presentes em problemas clássicos não-convexos nos quais, ao inserir parâmetros *fuzzy*, podem tornar-se convexos, é essencial para encontrar a solução ótima do problema “fuzzyficado”.

Outra proposta está no estudo de problemas de programação matemática multiobjetivo, pois estes problemas aproximam-se mais dos problemas reais. O estudo de problemas lineares multiobjetivos com parâmetros *fuzzy* está bem avançado, porém na área de problemas não-lineares multiobjetivos *fuzzy*, estes estudos ainda não se intensificaram.

Referências Bibliográficas

- AL-AHMARI, A. M. A. (2001). Mathematica model for determining machining parameters in multipass turning operations with constraints, *International Journal Production Research* **39**(15): 3397–3376.
- BAZARAA, M. S., SHERALI, H. D. & SHETTY, C. M. (1993). *Nonlinear Programming - Theory and Algorithms*, second edn, Jonh Wiley & Sons, New York.
- BELLMAN, R. E. & ZADEH, L. A. (1970). Decision-marking in a fuzzy environment, *Management Science* **17**(4): B141–B164.
- CANTÃO, L. A. P. (2003). *Programação Não-Linear com Parâmetros Fuzzy*, PhD thesis, FEEC - UNICAMP, Campinas.
- FERREIRA, P. A. V. (2004). Otimização não-linear. URL at <http://www.dt.fee.unicamp.br:80/~valente/ia543.html>.
- GOLDBERG, D. E. (1989). *Genetic Algorithms in Search, Optimization ans Machi Learning*, Addison-Wesley, Reading, MA.
- HIMMELBLAU, D. M. (1972). *Applied Nonlinear Programming*, McGraw-Hill Book Company, Reading, MA.
- KAUFMANN, A. & GUPTA, M. M. (1984). *Introduction to Fuzzy Arithmetic: Theory and Applications*, Van Nostrand Reinhold, New York.
- KLIR, G. J. & FOLGER, T. A. (1998). *Fuzzy Sets, Uncertainty and Information*, Englewood Cliffs, Prentice Hall, New Jersey.
- LEE, Y. H., YANG, B. H. & MOON, K. S. (1999). An economic machining process model using fuzzy non-linear programming and neural network, *International Journal Production Research* **37**(4): 835–847.
- LIU, B. & IWAMURA, K. (1998a). Chance constrained programming with fuzzy parameters, *Fuzzy Sets and Systems* **94**: 227–237.
- LIU, B. & IWAMURA, K. (1998b). A note on chance constrained programming with fuzzy coefficients, *Fuzzy Sets and Systems* **100**: 229–233.

- LUENBERGER, D. G. (1989). *Linear and Nonlinear Programming*, second edn, Addison-Wesley, Massachusetts.
- MICHALEWICZ, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, third edn, Springer, New York.
- PEDRYCZ, W. & GOMIDE, F. (1998). *An Introduction of Fuzzy Sets: Analysis and Design*, A Bardford Book.
- SCHITTKOWSKI, K. (1987). *More Test Examples for Nonlinear Programming Codes*, Spring-Verlag.
- SILVA, R. C. & CANTÃO, L. A. P. (2004). Comparação entre modelos clássicos e fuzzy para problemas de programação não-linear, *XXXVI Simpósio Brasileiro de Pesquisa Operacional*, pp. 1969–1978.
- SILVA, R. C. & YAMAKAMI, A. (2004). Uma contribuição para programação não-linear fuzzy, *XV Congresso Brasileiro de Automática*.
- TRAPPEY, J.-F. C., LIU, C. R. & CHANG, T.-C. (1988). Fuzzy non-linear programming: Theory and application in manufacturing, *International Journal Production Research* **26**(5): 975–985.
- XU, C. (1989). Fuzzy optimization of structures by the two-phase method, *Computers & Structures* **31**(4): 575–580.
- ZADEH, L. A. (1965). Fuzzy sets, *Information and Control* **8**: 338–353.
- ZIMMERMANN, H. J. (1983). Fuzzy mathematical programming, *Computer & Operation Research* **10**(4): 291–298.