2– PROGRAMAÇÃO LINEAR FUZZY

Neste capítulo serão abordadas a parte conceitual e a importância dos modelos de programação linear *fuzzy*.

2.1. PROGRAMAÇÃO LINEAR

A Programação Matemática constitui uma das mais importantes variedades dos modelos quantitativos ligados à Pesquisa Operacional (Goldbarg, 2000). Dentro da Programação Matemática encontra-se a Programação Linear (PL), a qual é um ramo da Otimização que estuda a maximização (ou minimização) de uma função linear sujeita a restrições que podem ser equações ou inequações lineares. Com o advento de computadores mais rápidos e poderosos, a programação linear tem se tornado de grande utilidade em aplicações práticas em diversas áreas, tais como Economia, Ciência da Computação, Medicina, Finanças, Matemática, assim como em diversos ramos da Engenharia. É importante salientar que o termo programação está associado a planejamento, e não à programação do ponto de vista computacional.

Na Programação Linear o uso de modelos faz parte de sua própria essência. Estudos desse tipo envolvem a construção de um modelo, que simplifica a realidade, porém preservando as relações essenciais de causa e efeito sobre o problema. Cria-se, portanto, um modelo simbólico, usando-se relações matemáticas que descrevem o problema em estudo. A Figura 2.1 ilustra o procedimento.

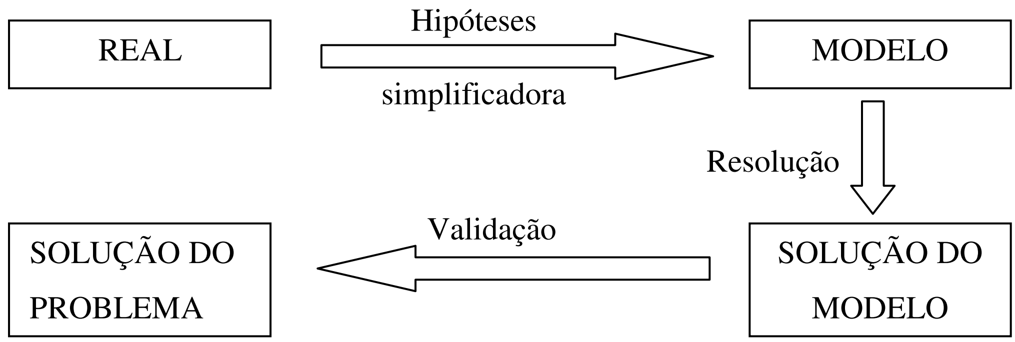

Figura 2.1 -Procedimento

Pode-se dizer que, ao se construir um modelo, passa-se do mundo real ao virtual, mediante simplificações que viabilizam uma elaboração conceitual e sua avaliação subseqüente.

Desta forma, obedecendo ao esquema sugerido pela Figura 2.1, um projeto de Programação Linear pode ser decomposto em cinco estágios ou etapas, a saber:

1. Identificação do problema;
2. Construção do modelo;
3. Determinação da solução do modelo;
4. Teste e validação da solução proposta;
5. Implementação da solução.

A seguir serão apresentados os principais conceitos de programação linear clássica necessários para o bom entendimento dos modelos apresentados nos próximos capítulos.

2.2. CONCEITOS

Modelos de programação linear fazem uso da terminologia descrita a seguir:

1. *Variáveis de Decisão:* representam o que realmente se quer determinar com a solução do problema, como, por exemplo, o volume, ou a quantidade, de cada produto final a ser produzido. Para se descrever um problema de programação linear, deve-se ter pelo menos 2 variáveis de decisão; caso contrário não é caracterizado um problema.
2. *Função Objetivo:* expressa, exatamente, a meta (ou objetivo) que se deseja alcançar com a solução do problema em questão; nela estão relacionadas, de forma linear, as variáveis de decisão.
3. *Restrições:* estabelecem limites ou possíveis valores das variáveis de decisão. Por exemplo, a disponibilidade limitada de certas matérias primas é expressa em forma de uma equação (ou inequação) linear.
4. *Região Viável:* Região no espaço em que todas as restrições são atendidas simultaneamente; todos os pontos dentro dessa região viável são possíveis, mas somente um ponto (ou ocasionalmente um pequeno conjunto de pontos) é ótimo.

5. *Linearidade:* as contribuições (coeficientes tecnológicos) de cada variável para o resultado final são valores fixos. Desta forma, as restrições para o problema de programação linear representam relações lineares.

Os modelos de programação linear são constituídos por uma função (linear) que descreve o objetivo a ser alcançado (função objetivo) e pelas restrições, todas elas lineares:

- ✓ A função objetivo é aquela que deve ser maximizada ou minimizada: por exemplo, lucro ou custo total;
- ✓ As restrições são as limitações existentes: por exemplo, disponibilidade de recursos ou necessidades a serem atingidas. Estas restrições representam relações de interdependências entre as variáveis de decisão.

A partir disto pode-se formular o seguinte problema de programação linear com n variáveis de decisão e m restrições:

$$
\begin{array}{ll}
\text{Max/Min} & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{sujeito a} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
& \vdots \qquad \vdots \qquad\qquad \vdots \quad \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\
& x_1 \ ; x_2 \ ; \cdots \ ; x_n \geq 0
\end{array}
$$

Os coeficientes c_j, b_i e a_{ij}, $i = 1,2, \ldots, m$ e $j = 1,2, \ldots, n$ são constantes reais; supõe-se ainda que cada equação seja multiplicada por (-1), caso necessário, de modo a garantir que todo $b_i \geq 0$.

Na apresentação matricial, a forma padrão da programação linear pode ser escrita:

$$
\begin{array}{ll}
\text{Max/Min} & Z = \boldsymbol{cx} \\
\text{Sujeito a} & \mathbf{A}\boldsymbol{x} = \boldsymbol{b} \quad (\boldsymbol{b} \geq \mathbf{0}) \\
& \boldsymbol{x} \geq \mathbf{0}
\end{array}
\qquad (2.1)
$$

onde: $\mathbf{A}$ = matriz m x n;

$\boldsymbol{b}$ = vetor m x 1;

$\boldsymbol{x}$ = vetor n x 1;

$\boldsymbol{c}$ = vetor 1 x n.

Comparando-se as dimensões m x n da matriz **A**, nota-se que o caso de interesse prático ocorre para $n > m$, quando o sistema deve oferecer uma infinidade de soluções e o método simplex busca identificar a solução ótima. Quando $n = m$, a matriz dos coeficientes é quadrada e, caso esta matriz seja não singular, o sistema terá uma solução única, não existindo alternativas de escolha. O último caso $(n < m)$ não faz sentido prático, indicando restrições redundantes no sistema, que devem ser descartadas, reduzindo o sistema para os dois casos anteriores.

Utilizando-se a forma matricial (2.1), e supondo que a função objetivo consista em maximizar $Z = \boldsymbol{cx}$, as seguintes definições são essenciais:

1. Uma solução viável é um vetor não negativo, $\boldsymbol{x} \geq 0$, satisfazendo $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$;
2. Região viável, S, é o conjunto de todas as soluções viáveis:

$$S = \{\boldsymbol{x} \mid \mathbf{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$$

3. Solução ótima é um vetor $\boldsymbol{x}^*$ tal que:

$$\boldsymbol{x}^* \in S \text{ e } \boldsymbol{cx}^* \geq \boldsymbol{cx}, \text{ para qualquer outro } \boldsymbol{x} \in S$$

4. Valor ótimo de um programa linear, Z^*, é o valor da função objetivo correspondente à solução ótima x^*:

$$Z^* = \boldsymbol{cx}^*$$

5. Ótimo alternativo é um vetor $\boldsymbol{x}^0 \in S$, sendo $\boldsymbol{x}^0 \neq \boldsymbol{x}^*$, tal que:

$$\boldsymbol{c}\mathbf{x}^0 = \boldsymbol{cx}^*$$

6. Solução ilimitada é uma seqüência de valores $\boldsymbol{x}^k$, $k = 1,2,...$, tais que $\boldsymbol{x}^k \in S$ e $\boldsymbol{cx}^k \geq \boldsymbol{cx}^{k\text{-}1}$ para todo k, de modo que $\lim\limits_{k \to \infty} cx^k = \infty$.

Em seguida são apresentados alguns conceitos relevantes para este trabalho (Bazaraa, 1990).

1. *Conjunto Convexo*

Um conjunto X no espaço E^m é chamado de *Conjunto Convexo* se, dado dois pontos quaisquer x_1 e x_2 em X, então $\lambda x_1 + (1-\lambda)x_2 \in X$ para cada $\lambda \in [0,1]$.

2. *Direções Extremas de um Conjunto Convexo*

Uma direção extrema de um conjunto convexo é uma direção do conjunto que não pode ser representada como uma combinação positiva de duas direções distintas do conjunto. Dois vetores, d_1 e d_2 são ditos serem distintos (não equivalentes) se d_1 não pode ser representado como múltiplo de d_2.

Qualquer outra direção do conjunto que não é um múltiplo ou sub-múltiplo de d_1 ou d_2 pode ser representada como $\lambda_1 d_1 + \lambda_2 d_2$ onde $\lambda_1, \lambda_2 \geq 0$.

3. *Cone Convexo*

Os cones convexos representam uma classe importante de conjuntos convexos. Seja C um cone convexo; ele é dito ser um conjunto convexo com a propriedade adicional $\lambda x \in$ C se o raio $\{\lambda x : \lambda \geq 0\}$ pertence a C.

Portanto, um cone é um conjunto convexo que consiste totalmente de raios emanando da origem.

Como um cone convexo é formado pelos seus raios, então ele pode ser totalmente caracterizado por suas direções. De fato, nem todas as direções são necessárias, desde que uma direção não extrema pode ser representada como uma combinação positiva de direções extremas. Portanto, um cone convexo é caracterizado por suas direções extremas.

Dado um conjunto de vetores a_1, a_2,..., a_k, é possível formar um cone convexo C com estes vetores. Este cone consiste de todas as combinações de a_1, a_2,..., a_k, isto é:

$$C = \left\{ \sum_{j=1}^{k} \lambda_j a_j : \lambda_j \geq 0, para\ j = 1, 2, \ldots, k \right\}$$

Na próxima seção serão apresentados os conceitos de um problema de programação linear *fuzzy*.

2.3.MODELOS DE PROGRAMAÇÃO LINEAR FUZZY

Ao longo dos anos, muito tem sido feito com a intenção de modelar problemas reais em termos de modelos de programação linear, já que, devido à facilidade de obtenção da solução e interpretação de resultados, esta é uma das técnicas mais utilizadas. Entretanto, dado o seu potencial, poder-se-ia esperar um número ainda maior de aplicações em problemas reais. Porém, para a sua utilização é necessário que os dados utilizados dados sejam obtidos de forma precisa e bem definida, o que implica muitas vezes em custos excessivos de medição e aferição dos medidores. Além disso, as informações provenientes do mundo real não são precisas; em geral, os problemas reais possuem grande complexidade e sua representação precisa e correta torna-se difícil.

Os coeficientes da função objetivo e das restrições dos modelos de programação matemática, em especial a programação linear, são, por definição, fixos, o que não corresponde à realidade em muitos problemas de cunho prático. Portanto, nos casos onde os dados representam demandas, capacidades disponíveis, taxas de custo etc., existe uma necessidade de se modelar tais problemas de uma forma mais flexível, de modo a contemplar incertezas intrínsecas aos problemas.

Com a finalidade de colocar estas incertezas no modelo matemático, no período de 1955 a 1960, alguns pesquisadores como Dantzig, Beale, Tinter, Charnes e Cooper (George L. Nemhauser, 1998) tentaram estender os métodos de programação linear para lidar com problemas de otimização com restrições cujos coeficientes estão sujeitas a variações aleatórias (modelos de programação estocástica). Esses primeiros modelos foram aplicados a problemas de diferentes setores, tais como teoria do estoque, microeconomia e manutenção de sistemas.

Os modelos de programação estocástica (Bazaraa, 1993) podem ser vistos como uma extensão dos modelos de programação linear e não linear para modelos de decisão onde os coeficientes têm uma representação probabilística. As situações onde dúvidas advêm da exatidão de conceitos, certeza de definições, graus de credibilidade tem pouco a ver com a ocorrência de eventos, que é a parte mais importante da teoria da probabilidade (Luhandjula, 1989).

A teoria dos conjuntos *fuzzy* oferece uma poderosa forma de modelar a incerteza dos dados sem necessidade de recorrer aos conceitos estocásticos

(Rommelfanger, 1996). Assim, a teoria dos conjuntos *fuzzy* aparece como sendo ideal para lidar com problemas que são formulados como modelos de programação linear onde os parâmetros de entrada possuem imprecisão.

Deste modo, modelos de programação linear capazes de lidar com incertezas nos dados constituem-se em uma importante ferramenta de apoio a tomadas de decisão em problemas gerenciais. Neste contexto, Bellman e Zadeh (Bellman and Zadeh, 1970) propuseram o conceito de tomada de decisão em ambientes *fuzzy*, apresentando uma metodologia para muitos problemas de otimização *fuzzy* onde tanto as restrições quanto a função objetivo podem ser representadas por conjuntos *fuzzy*.

A programação linear *fuzzy* pode ser vista como uma classe de problemas de otimização onde os parâmetros do modelo não estão bem definidos, ou seja, os coeficientes da função objetivo ou das restrições não são precisamente conhecidos, e alguma das inequações envolvidas no modelo podem estar sujeitas a limites não muito precisos (Pedrycz and Gomide, 1998).

A diferença entre a programação linear *fuzzy* e a programação linear convencional, onde os coeficientes são determinísticos (*crisp*), é que o modelo *fuzzy* se situa no limite entre o modelo convencional, que está no plano matemático e o problema real (Inuiguchi, 1997).

Dependendo da informação disponível para a construção do modelo de otimização, várias classes de problemas podem aparecer: restrições *fuzzy*, funções objetivo ou metas *fuzzy*, e coeficientes *fuzzy* na função objetivo, nas restrições, ou em ambos (Pedrycz and Gomide, 1998).

Assim, pode-se descrever um modelo geral de otimização *fuzzy* como (Cadenas and Verdegay 1995):

(i) *Problemas com restrições fuzzy*: o tomador de decisões permite uma satisfação flexível das restrições, isto é, por não poder definir exatamente as metas e as restrições do problema, são permitidas pequenas violações das restrições(b_i);

(ii) *Problemas com objetivo fuzzy*: as informações com relação aos coeficientes da função objetivo(c_j) são incertas; estes coeficientes podem ser definidos como números *fuzzy*. O conjunto de números *fuzzy* reais é denotado como F(R);

(iii) *Problemas com coeficientes fuzzy*: podem ser utilizados números *fuzzy* tanto nos coeficientes da matriz tecnológica(a_{ij}) como nas constantes do lado digareito (b_i), sendo, neste caso, um problema próximo ao do item (i). Estes casos podem ocorrer quando os dados são demandas, capacidades disponíveis, taxas de retorno etc, que envolvem quantidades imprecisas. Tais modelos são adequados para situações práticas, onde os parâmetros são obtidos a partir de especialistas.

Cabe ressaltar que toda combinação das três possibilidades acima é permitida, podendo ser enquadrada em um caso geral, conforme segue:

$$
\begin{array}{ll}
\text{Max} & \tilde{c}_1 x_1 + \tilde{c}_2 x_2 + \cdots + \tilde{c}_n x_n \\
\text{sujeito a} & \tilde{a}_{11} x_1 + \tilde{a}_{12} x_2 + \cdots + \tilde{a}_{1n} x_n \ \tilde{\leqq}\ \tilde{b}_1 \\
& \tilde{a}_{21} x_1 + \tilde{a}_{22} x_2 + \cdots + \tilde{a}_{2n} x_n \ \tilde{\leqq}\ \tilde{b}_2 \\
& \vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \quad\ \vdots \\
& \tilde{a}_{m1} x_1 + \tilde{a}_{m2} x_2 + \cdots + \tilde{a}_{mn} x_n \ \tilde{\leqq}\ \tilde{b}_m \\
& x_1 \ ; x_2 \ ; \cdots \ ; x_n \geq 0
\end{array}
\qquad (2.2)
$$

onde os seguintes elementos são considerados:

- ✓ Para cada linha (restrição) existe uma função de pertinência $\{\exists \mu_i \in F(R)$ tal que $\mu_i: R \rightarrow [0,1]\}$ definindo o número *fuzzy* do lado direito da equação;
- ✓ Para cada $i = 1,..., m$ e $j = 1,..., n$, $\exists\ \mu_{ij} \in F(R)$ tal que $\mu_{ij}: R \rightarrow [0,1]$, definindo os números *fuzzy* da matriz tecnológica;
- ✓ Para cada $j = 1,..., n$, $\exists\ \mu_j \in F(R)$ tal que $\mu_j: R \rightarrow [0,1]$, definindo os números *fuzzy* da função objetivo;
- ✓ Para cada linha, $\exists\ \mu_j \in F(R)$ tal que $\mu_j: R \rightarrow [0,1]$, fornecendo, $\forall x \in \Re^n$, o grau de acompanhamento do número *fuzzy* $\tilde{a}_{i1} x_1 + \tilde{a}_{i2} x_2 + ... + \tilde{a}_{in} x_n$, $i = 1,..., m$, com respeito à i-ésima restrição, isto é, a adequação entre este número *fuzzy* e o correspondente $\tilde{b}_i$ com relação à i-ésima restrição; e
- ✓ Os símbolos $\tilde{\leqq}$ ou $\tilde{\geqq}$ quando colocados nas inequações do modelo de programação linear *fuzzy* representam uma flexibilidade, ou seja, o tomador de decisões aceita que algumas restrições sejam parcialmente

violadas. Note-se que no caso convencional este tipo de violação não é permitido, representando uma infactibilidade no problema.

O sistema geral (2.2) inclui casos especiais, onde:

1. A função objetivo é *crisp*, ou seja, z (x) = $c_1x_1 + c_2x_2 + \ldots + c_nx_n$
2. Alguma ou todas as restrições são *crisp*, isto é, g_i (x) = $a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i$
3. Alguma ou todas as restrições estão na forma *soft*, isto é, $g_i(x) = a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \tilde{\leq} \tilde{b}_i$
4. Estes casos especiais apresentados nos itens 1 a 3 acima podem ser combinados.

A partir do descrito neste tópico, serão apresentadas na próxima seção, de forma resumida, as principais metodologias para a resolução dos modelos descritos.

2.4. TÉCNICAS DE RESOLUÇÃO DE MODELOS DE PROGRAMAÇÃO LINEAR FUZZY

De forma a melhor organizar esta seção, os modelos existentes foram separados em cinco partes distintas. Na Figura 2.2 é apresentada uma taxonomia de programação linear *fuzzy*, onde é possível identificar melhor a organização feita entre os modelos descritos nas próximas seções. Esta taxonomia foi feita considerando qual conjunto de coeficientes possui incerteza, e por este motivo é modelado como número *fuzzy*. Na primeira seção (2.4.1), aborda-se a programação flexível, ou programação linear com restrições *soft*, onde apenas os coeficientes do lado direito (b_i, i = 1, 2,..., m) das restrições do modelo de programação linear são tratados como coeficientes *fuzzy*.

Na segunda parte (seção 2.4.2), são descritos os modelos de programação linear com relações de desigualdade, onde os coeficientes do lado direito (b_i, i = 1, 2,..., m) e do lado esquerdo (a_j, j = 1, 2,..., n) das restrições não podem ser definidos com precisão, sendo então considerados coeficientes *fuzzy*.

Na terceira parte (seção 2.4.3) são apresentados os modelos de programação linear com imprecisão nos coeficientes do vetor de custos (c_j, j = 1, 2,..., *n*). Neste caso apenas os coeficientes da função objetivo não tem precisão nos seus dados de entrada, sendo modelados como coeficientes *fuzzy*. Primeiramente são relacionados trabalhos que propõem a transformação do problema *fuzzy* em um problema de programação linear multi-objetivo, resolvido então, pelas técnicas já desenvolvidas. Em seguida são descritos os demais trabalhos que descrevem outras metodologias para a resolução de tais problemas.

Na seção 2.4.4 são descritos os casos gerais i.e., combinações dos três anteriores. Os trabalhos descritos nesta seção são divididos em três grupos. O primeiro apresenta trabalhos que usam a técnica de intervalos para descrever os coeficientes *fuzzy* e, em seguida resolver o modelo. No segundo grupo estão os artigos que utilizam a rede neural como uma ferramenta para encontrar uma solução para o problema de programação linear *fuzzy*. Por último, são descritos os trabalhos que mostram aplicações práticas de modelos de programação linear *fuzzy*.

Por último, na seção 2.4.5, abordam-se artigos que utilizam a programação possibilística para descrever as incertezas.

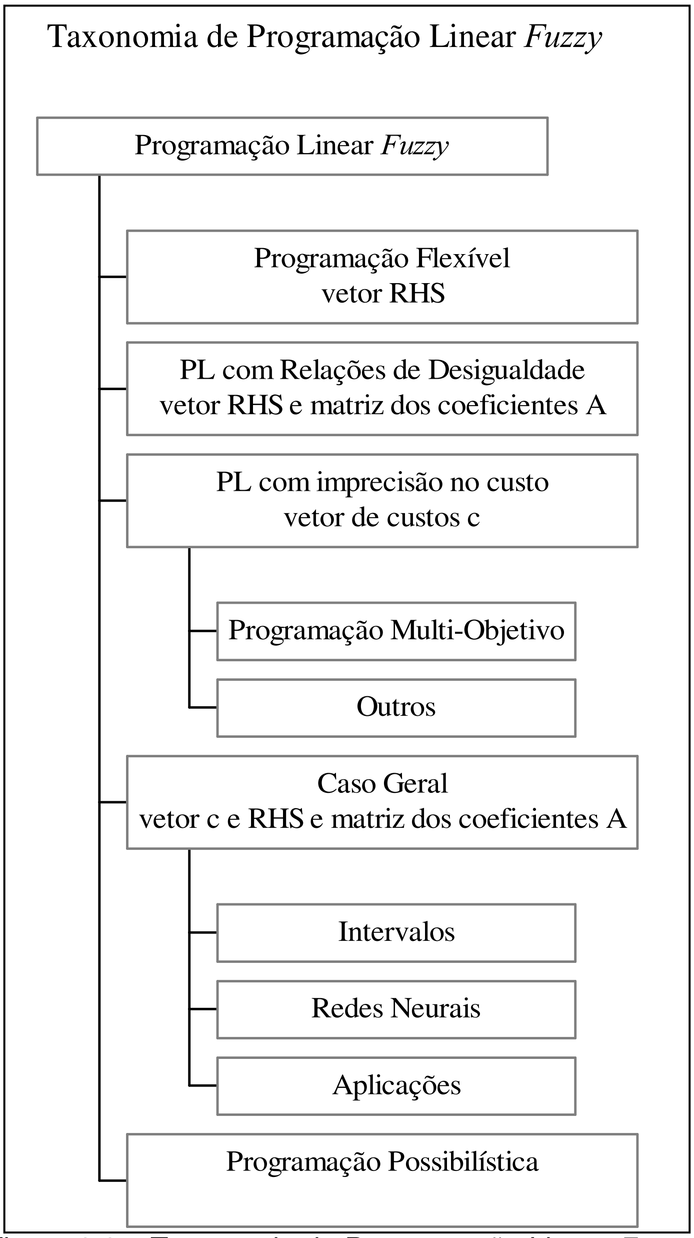

Figura 2.2 – Taxonomia de Programação Linear *Fuzzy*

2.4.1.Programação linear com restrições soft

Neste caso o tomador de decisões é capaz de especificar todos os coeficientes do modelo descrito em (2.2), porém nem todas as constantes do lado direito das restrições podem ser claramente determinadas como números *crisp*. Assim, as restrições que não podem ser descritas em termos *crisp* são chamadas de restrições *soft* e podem ser escritas da seguinte forma:

$$g_i(x) = g_i(x_1, x_2, \ldots, x_n) = a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \;\tilde{\leqq}\; \tilde{b}_i$$

Este tipo de modelo descreve a imprecisão da constante do lado direito, $\tilde{b}_i$ a partir de um conjunto *fuzzy* com suporte $[b_i, b_i+d_i] \subseteq \Re$, $d_i \geq 0$, e uma função de pertinência decrescente μ_{b_i}.

A função de pertinência μ_{b_i} deve ser especificada de forma a expressar a satisfação individual do tomador de decisões em relação à restrição *soft* $g_i(x)$.

Na literatura o coeficiente do lado direito flexível é modelado da seguinte forma(Rommelfanger, 1996):

$$\mu_{D_i}(g_i) = \begin{cases} 1 & \text{se} \quad g_i < b_i \\ \mu_{b_i}(g_i) & \text{se} \quad b_i \leq g_i \leq b_i + d_i \\ 0 & \text{se} \quad b_i + d_i < g_i \end{cases}$$

O número μ_{b_i} pode ser interpretado da seguinte forma, conforme pode ser observado na Figura 2.3:

- ✓ $\alpha = 1$: significa que g_i pertence com certeza ao conjunto de valores disponíveis;
- ✓ $\alpha = \lambda_A$: significa que o tomador de decisão está desejando aceitar g_i como um valor disponível.
- ✓ $\alpha = \varepsilon$: o tomador de decisão deseja negligenciar valores g_i tais que $\alpha < \varepsilon$.

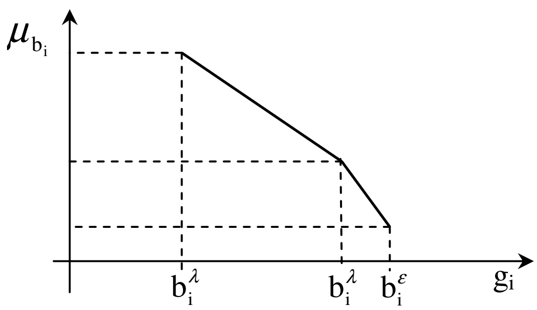

Figura 2.3 – Função de Pertinência de b_i

De acordo com Zimmermann (Rommelfanger, 1996), a relação $\tilde{\leqq}$ nas restrições *soft* pode ser interpretada como:

$$g_i(x) \;\tilde{\leqq}\; \tilde{B}_i \qquad \Leftrightarrow \qquad \begin{cases} g_i(x) \leq b_i + d_i & \Rightarrow \text{atender à restrição} \\ \mu_{D_i}(x) \to \text{Max} & \Rightarrow \text{obter máxima satisfação} \end{cases}$$

Assim, cada restrição *soft* pode ser substituída por uma restrição *crisp*, que representa a flexibilidade da restrição *soft* e um objetivo, chamado objetivo *fuzzy*.

Este objetivo *fuzzy*, que é obter a máxima satisfação, é acrescentado à função objetivo original.

Logo, supondo m_1 restrições *soft* e $m - m_1$ restrições *crisp*, a função objetivo do novo problema pode ser descrita mais precisamente como um sistema de otimização multiobjetivo do tipo:

$$\underset{x \in X_u}{\text{Max}} \; (z(x), \mu_1(x), \ldots, \mu_{m_1}(x))$$

onde,

$$X_u = \{x \in \Re_0^n \mid a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i + d_i, \forall i = 1,2,\ldots,m_1,$$
$$\text{e } a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i, \forall i = m_1 + 1,\ldots,m\}$$

Diferentes autores propuseram outros tipos de modelagem para a função de pertinência do lado direito das restrições.

- ✓ Forma linear: Zimmerman (1975), Sommer (1978) e Wernes (1984);
- ✓ Forma côncava: Sakawa (1983), Rommelfanger (1978) com funções exponenciais e, Hannan (1981), Rommelfanger (1984) Nakamura (1984) e Sakawa & Yano (1990) com funções lineares por partes;
- ✓ Forma de S: Hannan (1981) e Rommelfanger (1984) com funções lineares por partes, Leberling (1981) e Sakawa & Yano (1990) com funções hiperbólicas inversas, Zimmerman e Zysno (1982) com funções logísticas e Schwab (1983) com funções cúbicas.
- ✓ Forma Trapezoidal: (Chanas, 1984) analisa o problema de transportes com valores de suprimento *fuzzy* dos fornecedores e com valores de demanda *fuzzy* dos clientes. Para a solução do problema é utilizada a técnica de programação paramétrica. (Felizari, 2003) apresenta um estudo da aplicação de otimização *fuzzy* em problemas de programação matemática linear, particularmente num sistema de *blending* presente na indústria petroquímica.

Guu & Wu (1999) propuseram um método de duas fases para obter uma solução melhor do que a apresentada ao se utilizar um operador 'max-min' para um problema de programação linear *fuzzy* onde os recursos são *fuzzy*.

Em Liu (2001) é proposto um novo tipo de método para resolução de problemas de programação linear *fuzzy* baseado no grau de satisfação das restrições. A partir de um novo método de ordenamento das restrições, é possível determinar um índice para cada uma das restrições. Com o uso deste índice, o responsável pela tomada de decisões pode fazer as restrições apertadas ou frouxas com base em sua atitude otimista ou pessimista e obter uma solução ótima do espaço de soluções *fuzzy*.

2.4.2. Programação linear com relações de desigualdade

Umas das questões principais na obtenção de uma solução para modelos de programação linear *fuzzy* é a interpretação das relações de desigualdade nas restrições *fuzzy*, ou seja, $\tilde{A}_i(x) \,\tilde{\leqq}\, \tilde{B}_i$ (Rommelfanger, 1996), onde '~' indica que o tomador de decisões não é capaz de especificar claramente as restrições do problema.

Na literatura podem ser encontrados diversos conceitos e interpretações para a relação de desigualdade $\tilde{\leqq}$. Em muitas destas metodologias as restrições *fuzzy* são trocadas por uma ou duas restrições *crisp*. Uma das interpretações mais flexíveis é de Rommelfanger (1996), que considera dois índices: um pessimista, utilizado também por Slowinsk (1986), entre outros, e uma meta, denominada *meta fuzzy*:

$$\tilde{A}_i(x) \,\tilde{\leqq}_R\, \tilde{B}_i \;\Leftrightarrow \begin{cases} \sum_{j=1}^{n} \left(\overline{a}_{ij} + \overline{\alpha}_{ij}^{\varepsilon}\right) x_j \leq b_i + \beta_i^{\varepsilon} & \rightarrow \text{índice pessimista} \\ \mu_i(x) = \mu_{D_i}(\overline{a}_i(x)) \rightarrow \text{Max} & \rightarrow \text{meta fuzzy} \end{cases}$$

A relação $\tilde{\leqq}_R$ coincide com a interpretação usual da relação de desigualdade, apresentada nas restrições *soft,* no caso das inequações serem determinísticas. Então, pode-se dizer que a relação $\tilde{\leqq}_R$ representa uma definição geral para as relações de desigualdade em modelos de otimização.

Em 1976, Negoita & Sularia (Rommelfanger, 1996) consideraram incertezas, representadas por conjuntos *fuzzy*, nos coeficientes tecnológicos e nas constantes do lado direito das restrições do modelo de programação linear. Ainda nesta linha de raciocínio podem ser citados os trabalhos de Tanaka & Asai (1984), Ramik & Rimanek (1985), Slowinsk (1986), Carlson e Korhonem (1986), Luhandjula (1987), Rommelfanger (1988) e Buckley (1988 e 1989). Em geral, muito dos métodos apresentados consistem na troca das restrições *fuzzy* por uma ou duas restrições *crisp*.

Ainda nesta linha de problemas *fuzzy*, Fang, Hu et al. (1999) mostraram, com base em uma organização dos números *fuzzy*, que problemas de programação linear *fuzzy* podem ser reduzidos a problemas de programação linear semi-infinita, onde a cada iteração é introduzido um conjunto de restrições *crisp*.

2.4.3. Maximizando a função objetivo

Quando os coeficientes na função objetivo são descritos como números *fuzzy*, o problema pode ser modelado como:

$$\begin{aligned} &\text{Max} \quad \sum_{j=1}^{n} \tilde{c}_j x_j \\ &\text{sujeito a} \quad \sum a_{ij} x_j \leq b_i,\ i = 1,\ldots,m \\ &\qquad\qquad x_j \geq 0,\ j = 1,\ldots,n \end{aligned} \tag{2.3}$$

onde, para cada j = 1,..., n, existe uma função de pertinência $\mu_j \in F(\Re)$ associada, definindo os números *fuzzy* na função objetivo (Cadenas and Verdegay 1995).

Seja a função objetivo do sistema de otimização (2.3):

Max $\tilde{Z}(x) = \tilde{c}_1 x_1 \oplus \ldots \oplus \tilde{c}_n x_n$,

onde o símbolo $\oplus$ representa a adição extendida, melhor descrita em Rommelfanger (1996).

Supõe-se o caso mais simples, onde os coeficientes $\tilde{c}_j$ têm a forma de um número *fuzzy* trapezoidal $\tilde{c}_j = (\underline{c}_j; \bar{c}_j; \underline{\gamma}_j; \bar{\gamma}_j)$, conforme pode ser observado na Figura 2.4.

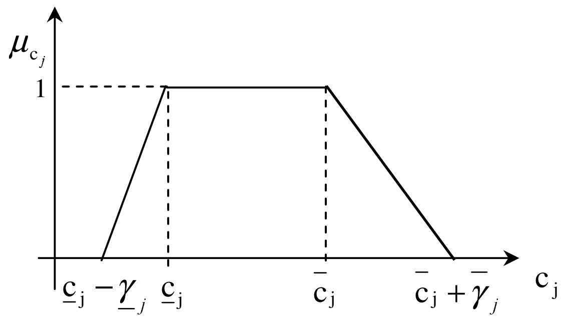

Figura 2.4 – Função de Pertinência de $\tilde{c}_j$

Assim, $\tilde{Z}(x)$ pode ser escrito como:

$$\tilde{z} = \left(\underline{c}(x);\, \overline{c}(x);\, \underline{\gamma}(x);\, \overline{\gamma}(x)\right),$$

onde,

$\underline{c}(x) = \sum_{j=1}^{n} \underline{c}_j x_j$, $\underline{\gamma}(x) = \sum_{j=1}^{n} \underline{\gamma}_j x_j$, $\overline{c}(x) = \sum_{j=1}^{n} \overline{c}_j x_j$, $\overline{\gamma}(x) = \sum_{j=1}^{n} \overline{\gamma}_j x_j$, conforme representado na Figura 2.5.

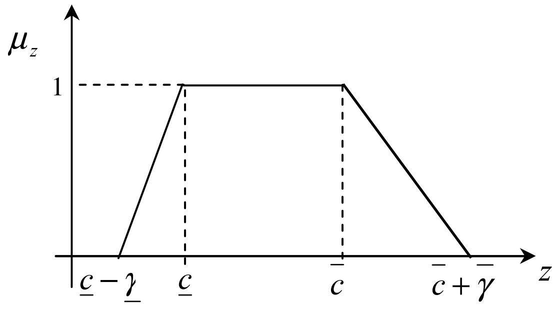

Figura 2.5 – Função de Pertinência de $\tilde{z}$

Desta forma, a função objetivo *fuzzy* do sistema (2.3) implica em 4 metas que devem ser satisfeitas simultaneamente no conjunto das soluções factíveis *X*:

- ✓ Max → $\underline{c}(x)$
- ✓ Max → $\underline{c}(x) - \underline{\gamma}(x)$
- ✓ Max → $\overline{c}(x)$
- ✓ Max → $\overline{c}(x) + \overline{\gamma}(x)$

Este tipo de problema torna-se agora um problema de programação linear multi-objetivo e, como tal, pode ser resolvido pelas técnicas desenvolvidas e apresentadas, por exemplo, em Ignizio (1982).

Tanaka et al, em 1984, (Rommelfanger, 1996), apresentaram um tratamento para os problemas de programação linear *fuzzy* onde apenas os coeficientes da função objetivo não eram bem definidos. Os autores procuraram resolver o problema inicial trocando a função objetivo *fuzzy* por uma solução compromisso. Esta nova função objetivo pode ser descrita como uma média ponderada, conforme pode ser observado em (2.4). Esta nova função foi chamada de "objetivo compromisso".

$$z(x) = \frac{1}{6} \sum \left(2\underline{c}_j + 2\overline{c}_j + \underline{\gamma}_j + \overline{\gamma}_j\right) x_j \qquad (2.4)$$

Ainda nesta linha de transformar o problema de programação linear fuzzy em um problema de programação linear multi-objetivo, Sakawa & Yano, em 1989

(Rommelfanger, 1996), propuseram que se calculasse uma solução Pareto α, ou solução não dominada α, onde os coeficientes dos custos fossem restritos por um $\alpha-\text{cut}$. Algo similar também foi proposto por Luhandjula (1987), com o conceito de "solução eficiente β possível". Numa solução Pareto ótima, ou solução não dominada, ou ainda, solução eficiente, busca-se gerar o conjunto total de soluções eficientes que não são dominadas por qualquer outra solução (Ignizio, 1982).

Em Zhang (Zhang, 2003) também transforma-se, através de propriedades e conceitos de soluções ótimas para problemas de programação linear *fuzzy*, este tipo de problema em um problema de programação linear multi-objetivo. Para isso, alguns teoremas são desenvolvidos com o intuito de converter problemas de programação linear *fuzzy* em problemas de otimização com quatro funções objetivo. Por último, os autores apresentam dois exemplos ilustrativos para demonstrar o procedimento de solução. Os exemplos apresentados permitem mostrar que o método de solução apresentado neste artigo inclui um método já existente, determinado em Maeda (Maeda, 2001), como um caso especial, sendo a metodologia desenvolvida mais abrangente.

Inuiguchi, Ichihashi et al. (1990) propuseram uma técnica para a encontrar uma solução aproximada para problemas de programação linear *fuzzy* onde as funções de pertinência que descrevem o conjunto *fuzzy* que representa a função objetivo são funções contínuas lineares por partes. A metodologia descrita pode ser utilizada também para aproximar as funções de pertinência não lineares por funções de pertinência lineares por partes. A vantagem apresentada neste método é que ele não necessita do uso repetitivo das técnicas de programação linear para a sua utilização.

Em Inuiguchi (2004) é mostrado que uma solução básica ótima possível pode ser representada por uma combinação convexa de vértices ótimos (pontos extremos) possíveis. A partir desta representação, e sabendo-se que os pontos extremos viáveis estão conectados, é desenvolvido um método para enumerar todos os vértices ótimos possíveis, junto com os graus de pertinência associados. O algoritmo de enumeração é dado e exemplificado por um simples exemplo numérico.

2.4.4. Caso Geral

Yaguang Yang (1991) apresentou um novo método para resolução de problemas de programação linear com incertezas. Em contraste ao problema de programação linear estocástica e muitos modelos de programação linear *fuzzy*, os parâmetros do problema discutido neste artigo não são nem variáveis aleatórias com distribuições de probabilidade conhecidas nem parâmetros com distribuições de possibilidade conhecidas. Tudo o que se sabe a respeito dos parâmetros são os seus intervalos de existência. Portanto, como estes parâmetros não possuem média e nem distribuição conhecida, não se pode utilizar programação estocástica ou programação linear *fuzzy*. Neste artigo, o conceito de função de confiança é introduzido e, baseados nesta definição, os autores resolvem o problema considerando apenas os intervalos de cada coeficiente. Este algoritmo tem sua convergência comprovada, sendo os graus de risco da decisão para diferentes funções de confiança então discutidos.

Shaocheng (1994) abordou o problema de programação linear com incertezas de duas formas. A primeira é a programação linear com números *fuzzy* e a segunda a programação linear com números dentro de intervalos. Os problemas com coeficientes que são números *fuzzy* são aproximados de duas formas: "*fuzzy* decisive set approach" e "metodologia de programação linear com número dentro de intervalo para vários níveis de funções de pertinência". A primeira metodologia ("*fuzzy decisive set approach*") está baseada no ordenamento entre números *fuzzy*, tomando as pertinências do conjunto decisivo como parâmetros desconhecidos, sendo reduzido a um problema de programação não linear. Neste caso é combinado o método da biseção e a fase 1 do método *simplex* para se obter uma solução possível.

Os problemas de programação linear com coeficientes que são números que variam dentro de intervalos são aproximados tomando o valor máximo e o valor mínimo da faixa de valores das inequações como condições das restrições, reduzindo em dois problemas de programação linear clássicos e, a partir destes problemas, encontra-se a faixa da solução ótima para o problema inicial.

Em Cadenas and Verdegay (1995) apresenta-se a análise e o desenvolvimento de um sistema interativo de suporte à tomada de decisão para problemas de programação linear *fuzzy*. Este sistema resolve problemas de programação linear sem incertezas (crisp ou clássico), e os problemas com incertezas nos coeficientes $\tilde{c}_j$, nos coeficientes do lado direito das restrições $\tilde{b}_i$ e o caso geral, onde todos os coeficientes possuem incertezas. Em cada um destes casos as incertezas são modeladas como números *fuzzy*. O sistema é interativo, buscando informações com o usuário sobre os diferentes elementos de cada problema para poder, através de cortes nas funções de pertinência, resolver cada um dos problemas.

De forma semelhante, Cadenas (2004) considera um problema de determinação de uma dieta para a pecuária em fazendas na Argentina como um problema de programação linear *fuzzy*. O problema assim modelado é resolvido por um sistema de suporte à decisão (DSS) chamado SACRA, baseado no PROBO (Cadenas, 1995), que é altamente amigável, interativo e não requer conhecimento sobre programação linear *fuzzy*.

Em Vansant (2004) os autores tratam de um problema de seleção de *mix* de produção em uma fábrica de chocolate. Devido às imprecisões em cada matéria prima, o problema é tratado como um problema de programação linear *fuzzy*. Os coeficientes são modelados como números *fuzzy*, onde a função de pertinência é uma função de pertinência do tipo S.

Ainda na linha dos artigos em que são feitas aplicações de programação linear *fuzzy* tem-se Eshwar (2004). Neste artigo são aplicados os conhecimentos de conjuntos e números *fuzzy* para introduzir as incertezas e imprecisões em um modelo de programação linear utilizado nas indústrias da construção civil. Como resultado da aplicação de conceitos *fuzzy* ao modelo de programação linear tradicional é identificado o número ótimo de peças de equipamentos necessários para completar o projeto no período determinado. Foi considerado um estudo de caso realístico para a otimização e o software LINGO versão 6 foi utilizado para resolver as várias equações não lineares.

Iung (2000) também faz uma aplicação das técnicas de programação linear fuzzy. Neste caso, o autor modela os problemas de planejamento da expansão de sistemas de transmissão de energia elétrica com incertezas como um problema de programação linear *fuzzy*

Canos, Ivorra et al. (1999) propuseram uma versão *fuzzy* para o problema clássico da p-mediana. Este tipo de abordagem para o problema clássico da p-mediana tem a vantagem de permitir ao tomador de decisão levar em conta soluções que tenham menor custo, apesar de deixarem parte da demanda não atendida. Os autores consideraram, dentro das restrições iniciais, que uma parte das restrições é *fuzzy*, associando uma função de pertinência. Além disso, foi considerada uma meta para o valor do custo, sendo também modelado como número *fuzzy*. Desta forma, o tomador de decisões poderá escolher soluções parcialmente factíveis que cubram de modo parcial as demandas, implicando numa grande redução no custo. Este tipo de abordagem traz um grau de liberdade maior para poder modificar a cobertura das demandas, visto que será oferecido o acesso a diferentes cenários.

O algoritmo descrito em Canos, Ivorra et al. (1999) considerou pequenas modificações na demanda total que deve ser coberta e no custo de transporte associado a esta demanda. Entretanto, continuando com o problema da p-mediana, Canos and C. Ivorra (2001) mostraram que a mesma metodologia pode ser aplicada num sentido mais global, ou seja, para estudar o comportamento de custos ótimos quando ocorrem violações arbitrárias das mesmas restrições *fuzzy*, fornecendo também uma informação útil no caso de uma análise de sensibilidade.

A metodologia descrita por Jamison and Lodwick (2001) consiste em transformar um problema de programação linear *fuzzy* em um problema de otimização irrestrito. Neste problema irrestrito a função a ser otimizada contém a função objetivo original e as restrições do problema original multiplicadas por constantes, formando assim uma penalidade para violações. Os autores mostram que a nova função objetivo é uma função côncava, podendo ser maximizada globalmente.

Em Ekel (1998) é apresentada e formalizada uma metodologia para resolver uma larga classe de problemas de otimização onde os coeficientes da função objetivo e das restrições são *fuzzy*. Esta metodologia está associada a uma modificação nos métodos tradicionais de programação matemática e permite cortar as alternativas dominadas acima e abaixo. A subseqüente contração da região incerta de decisão está associada à redução do problema de tomada de decisão multicritério em um ambiente *fuzzy*. Além disso, também é proposta uma metodologia geral aplicada dentro do contexto de um modelo de otimização *fuzzy* discreta baseada na modificação de algoritmos de otimização discreta. Antes da aplicação desses algoritmos existe uma transição do modelo com coeficientes *fuzzy* nas funções objetivo e nas restrições para um equivalente análogo com coeficientes *fuzzy* somente na função objetivo. Os resultados apresentados são de caráter universal e já estão sendo utilizados para resolver problemas de engenharia de potência. Este artigo contribuiu na definição da região de factibilidade do problema de programação linear *fuzzy*, onde os coeficientes das restrições e da função objetivo são modelados como números *fuzzy*, conforme visto no item 4.3.1.

Ramik & Rommelfanger (1996) trataram de problemas de programação matemática onde as inequações são lineares; as funções de pertinência que as caracterizam como *fuzzy* não são necessariamente lineares.

Buckley, Feuring et al. (1999), Li e Da (2000) e Chong, Hui et al. (1999) utilizaram uma rede neural para obter soluções aproximadas para problemas de programação linear *fuzzy*. O primeiro artigo utiliza o algoritmo evolucionário para substituir o algoritmo *backpropagation*. O segundo demonstra que a flexibilidade das redes neurais pode ser utilizada na modelagem e resolução de diversos problemas de programação matemática. O terceiro artigo modela as redes neurais por sistemas de gradiente dinâmico, construídos a partir de uma família paramétrica de funções de penalidade (não diferenciáveis) exatas, provando em seguida que para um dado problema de programação linear e parâmetros de penalidade suficientemente grandes, algumas trajetórias da rede neural convergem num tempo finito para seu conjunto solução.

Em Buckley (2000) os autores desejam encontrar soluções para um problema de programação linear onde todos os parâmetros e variáveis são números *fuzzy*. Primeiramente os autores modificaram o problema de maximização de um número *fuzzy*, o valor da função objetivo, em um problema de programação linear multi-objetivo *fuzzy*. É provado que a programação flexível pode ser utilizada para explorar o conjunto não dominado total para o programa linear *fuzzy* multi-objetivo. Um algoritmo evolucionário é desenvolvido para resolver a programação linear flexível, sendo aplicado em um problema exemplo, gerando boas soluções.

Em Maleki (2000), os autores apresentam a proposta de um novo método para resolução de problemas de programação linear com variáveis *fuzzy*. Esta metodologia usa o conceito de comparação de números *fuzzy*.

Em Ekel (2001) são apresentados resultados de pesquisa no uso de conjuntos *fuzzy* para manusear várias formas de incerteza em problemas de otimização relacionados ao projeto e controle de sistemas complexos. É dada muita atenção para considerar a incerteza de metas associadas com o caráter multicritério de alguns problemas de otimização.

Em Baykasoglu (2003) os autores propõem analisar e classificar os procedimentos existentes para resolução de uma variedade de problemas de otimização com múltiplos objetivos *fuzzy* incluindo programas com metas *fuzzy*. Uma outra proposta deste artigo é guiar o tomador de decisões para encontrar um método de solução para resolver seu próprio modelo *fuzzy*.

2.4.5.Programação Possibilística

Um problema de programação linear possibilística é definido como:

$$\begin{aligned} &\text{Max/Min} \quad Z = \sum_{j=1}^{n} \tilde{c}_j x_j \\ &\text{sujeito a} \qquad \sum \tilde{a}_{ij} x_j * \tilde{b}_i, \quad i = 1, \ldots, m \qquad (2.4) \\ &\qquad\qquad\qquad x_j \ge 0, \quad j = 1, \ldots, n \end{aligned}$$

onde:

- ✓ "*" pode denotar $\subset, \subseteq, =, \supset, \supseteq$, para cada restrição *i*;
- ✓ Os coeficientes $\tilde{c}_j, \tilde{a}_{ij}, \tilde{b}_i$ representam conjuntos *fuzzy*.

Esses conjuntos *fuzzy* podem ser especificados como trapezoidais:

- ✓ $\tilde{c}_j = [c_{j1} \mid c_{j2}, c_{j3} \mid c_{j4}], \tilde{a}_{ij} = [a_{j1} \mid a_{j2}, a_{j3} \mid a_{j4}], \tilde{b}_i = [b_{j1} \mid b_{j2}, b_{j3} \mid b_{j4}]$

Note-se que os números *fuzzy* são distribuições de possibilidade associadas às variáveis *fuzzy* e, portanto, determinam uma restrição aos possíveis valores que as variáveis podem assumir.

Para os conjuntos *fuzzy* descritos, tem-se:

- ✓ $\text{Poss}[\tilde{c}_j = \gamma] = \mu(\gamma \mid \tilde{c}_j)$: possibilidade de $\tilde{c}_j$ ser igual a γ
- ✓ $\text{Poss}[\tilde{a}_{ji} = \alpha] = \mu(\alpha \mid \tilde{a}_{ji})$: possibilidade de $\tilde{a}_{ij}$ ser igual a α
- ✓ $\text{Poss}[\tilde{b}_i = \beta] = \mu(\beta \mid \tilde{b}_i)$: possibilidade de $\tilde{b}_i$ ser igual a β
- ✓ $\text{Poss}[Z = z]$ é a distribuição de possibilidade da função objetivo *Z*.

Além disso, pode-se definir a possibilidade com que x satisfaz a i-ésima restrição, da seguinte forma:

$$\Pi(a_{ij}, b_i) = \min(\mu(a_{i1}|\tilde{a}_{i1}), \ldots, \mu(a_{in}|\tilde{a}_{in}), \mu(b_i|\tilde{b}_i)),$$

onde $a_i = (a_{i1}, \ldots, a_{in})$ é a distribuição conjunta de $\tilde{a}_{ij}$ e $\tilde{b}_i$.

Ainda, define-se a possibilidade de que x é factível com respeito a i-ésima restrição como:

$$\text{Poss}[x \in \Im_i] = \sup_{a_i, b_i}\{\Pi(a_i, b_i) \mid a_i x * b_i\},$$

onde $\Im_I$ região de viabilidade da i-ésima restrição

Portanto, para $x \geq 0$, tem-se a possibilidade que x é factível, ou seja,

$$\text{Poss}[x \in \Im] = \min_{l \leq i \leq m} (\text{Poss}[x \in \Im_i]),$$

onde $\Im$ é a região factível do problema (2.3).

Em Buckley (1989) a programação linear possibilística utiliza conjuntos *fuzzy* triangulares; no trabalho é descrito um procedimento de solução para problemas de programação linear possibilística que não estão na forma padrão (onde todas as restrições de (2.4) são equações lineares de igualdade e todas as constantes do lado direito são não negativas). A solução envolve a distribuição de possibilidade Poss[Z = z] para a função objetivo e, então, é determinada a solução compromisso para as variáveis de decisão. Além disso, mostra-se que a solução para o problema de programação linear proporciona valores corretos para z em Poss[Z = z] = α para cada α entre 0 e 1.

Em Inuiguchi (1997) e Inuiguchi and Ramik (2000) são apresentadas, por meio de exemplos, uma metodologia de resolução do problema de programação linear possibilística – utilizando conjuntos *fuzzy* triangulares simétricos –, onde os coeficientes da função objetivo e do lado esquerdo das restrições possuem incertezas. É feita uma comparação com metodologias de programação estocástica, analisando-se as suas vantagens e desvantagens. É considerado um problema de seleção de *portfolio* de ações ótimo.

Da mesma forma, em Inuiguchi and Tanino (2000) abordou-se o problema de seleção de *portfolio* de ações ótimo, tradicionalmente tratado por um modelo de programação estocástica ("modelo de Markowitz"). Utilizando a semelhança entre a programação estocástica e a programação possibilística os autores propuseram uma nova metodologia de resolução baseada na programação possibilística.

Gandolpho, Vellasco et al. (2002) modelou um problema de mistura de carvões para siderúrgicas a coque, onde as incertezas são inerentes a cada matéria prima, utilizando a programação linear *fuzzy*. Em seguida foi utilizada a programação possíbilísitica para obter uma solução, sendo então comparada ao resolvido no âmbito programação linear *crisp*.

Em Tanaka, Guo et al. (2000) são estudados vários tipos de distribuição de possibilidade das variáveis *fuzzy* aplicadas a problemas de programação linear possibilística. Mostra-se que um problema de programação linear possibilística pode ser diretamente reduzido a um problema de programação linear convencional quando são utilizadas distribuições de possibilidade triangulares ou intervalos. Quando são consideradas as distribuições de possibilidade exponenciais, os problemas de programação possibilística tornam-se problemas de otimização não linear.

Em Inuiguchi (2002) os autores propõem uma metodologia de decomposição de cenário para tratamento de números *fuzzy* interativos, que são definidos por regras do tipo *if-then fuzzy*. As propriedades dos números *fuzzy* decompostos em cenários são investigadas e apresentadas. Além disso, os autores aplicam esta metodologia na programação linear possibilística. Neste caso, os coeficientes da matriz tecnológica e os coeficientes da função objetivo no modelo de programação linear possibilística são definidos como vetores de variáveis possibilísticas, cujas faixas de possibilidades são dadas pelos números *fuzzy* decompostos em cenários.

2.5. A ANÁLISE DE SENSIBILIDADE

Em muitas aplicações práticas alguns dados utilizados não são conhecidos a priori e, assim, faz-se uso de estimativas. É importante ser capaz de encontrar uma nova solução ótima do problema quando os dados estiverem efetivamente disponíveis, sem resolver o problema novamente (e, eventualmente, inúmeras vezes). Além disso, existem diversos casos em que se deseja determinar a sensibilidade da solução ótima em face de mudanças nos dados de entrada. Em certas situações, as restrições podem não ser muito rígidas, como por exemplo uma restrição que reflita a disponibilidade de algum recurso. Essa disponibilidade pode aumentar com a compra de algum equipamento novo, horas extras ou, ainda, com a contratação de mão-de-obra. Dessa forma, é desejável examinar o efeito de se relaxar alguma(s) da(s) restrição(ões) no valor da função objetivo, bem como na estrutura da solução, sem resolver o problema original novamente.

A análise de sensibilidade é executada após uma solução ótima corrente de um modelo de programação linear ter sido encontrada. A meta é determinar se as mudanças nos coeficientes do modelo, conforme citado anteriormente, não afetam a solução atual e, caso afetem, como pode ser encontrada de forma eficiente a nova solução, caso ela exista.

Em geral, as mudanças nos dados de entrada do modelo podem resultar em quatro casos distintos:

1. A solução encontrada permanece ótima;
2. A solução atual torna-se *infactível*;
3. A solução atual torna-se não ótima;
4. A solução corrente torna-se *infactível*.

O primeiro caso é o mais simples, pois as mudanças nos dados de entrada não afetam a solução do problema, permanecendo assim a solução ótima encontrada anteriormente. O caso 2 indica que a solução corrente tornou-se *infactível* e, portanto, deve-se aplicar o método dual *simplex* para se retomar a *factibilidade* do problema. No caso 3 pode-se aplicar o método primal *simplex* para se restaurar a sua otimalidade. No último caso são utilizados ambos os métodos, primal e dual *simplex*, para se restabelecer uma solução para o problema.

No Apêndice A é feita a abordagem teórica necessária para um bom entendimento das metodologias utilizadas na análise de sensibilidade em programação linear.

A seguir será apresentada uma breve descrição histórica sobre o desenvolvimento da análise de sensibilidade, ou análise pós-ótima, em programação linear.

2.6. DESENVOLVIMENTO HISTÓRICO

Logo após o surgimento do método *Simplex*, colocaram-se certas questões como “o que ocorre quando há mudanças nos dados originais?”, “que efeito terão na solução ótima algumas influências a posteriori?", "é possível encontrar um conjunto de soluções (para as mudanças nos dados iniciais) para o dado problema tal que todas sejam ótimas?". Todas essas perguntas são motivadas pelo fato de que os dados de entrada muitas vezes não são precisos, ou seja, eles costumam conter incertezas. Essas, por sua vez, só se tornavam conhecidas após o problema ter sido otimizado. Esses tipos de perguntas estimularam a procura por uma solução para as possíveis mudanças nos dados iniciais dos problemas, sem a necessidade de resolvê-los novamente. As primeiras pesquisas nesse sentido foram dirigidas para pequenas mudanças no termo do lado direito e para uma parametrização no coeficiente da função objetivo (Gal & Greenberg, 1997).

Seguindo essa tendência de desenvolvimento, apareceu em 1977 um dos primeiros trabalhos de *Análise de Sensibilidade* (AS) em programação linear inteira (Geoffrion & Nauss, 1977), apresentando a teoria básica para possíveis análises pós-ótimas em problemas de programação inteira, com o intuito de estendê-las para a programação linear. Em (Wolsey, 1981) e (Schrage & Wolsey, 1985) apresenta-se de maneira formal, e por meio de exemplos, como podem ser obtidas as análises pós ótimas no caso de programação inteira (PI).

Em Gal (1979) apresenta-se uma concatenação de boa parte da teoria até então desenvolvida nessa área. Motivados pelo desenvolvimento da parte teórica, e pela falta de aplicações práticas desta ferramenta poderosa, surge no início da década de 80 o primeiro esforço para tornar mais automática a aplicação das metodologias de análise de sensibilidade: o software *Analyze* (Greenberg, 1983) – desenvolvido para acompanhar a análise de sensibilidade em modelos de programação linear. Inicialmente foi utilizado para um trabalho específico, servindo a posteriori para explicar de forma mais clara a solução de modelos de programação linear. Com esse trabalho, tornou-se possível utilizar a análise de sensibilidade em programação linear de forma mais automática, sem a necessidade de um especialista em matemática.

Na segunda metade da década de 80 aparecem trabalhos importantes explicando os possíveis problemas na análise de sensibilidade quando da presença de degenerescência (Greenberg 1986, Gal 1986, Gilford 1994), a qual é examinada com respeito às perturbações no vetor de custo e das constantes do lado direito, bem como os problemas algorítmicos que aparecem (como a ciclagem). No caso da ciclagem, é discutida e modificada (Gilford, 1994) uma nova metodologia (método TNP – *Transitions Node Pivoting Procedure*), de forma a ajudar na análise de sensibilidade. Ainda seguindo essa linha de pesquisa, pode se citar o trabalho de Knolmayer (Knolmayer, 1984), onde é considerado o problema da determinação de preços duais e uma análise de sensibilidade para propósitos gerenciais. Este trabalho tem continuidade na década de 90 (Jansen, Jong et al., 1997), onde são apresentados os erros de interpretação que ocorrem quando do uso de *softwares* de otimização na análise de sensibilidade. São sugeridos três métodos para se realizar a análise, evitando-se problemas com a interpretação.

Nos trabalhos apresentados nesta seção é possível observar que, apesar do desenvolvimento teórico, pouco foi apresentado em termos práticos. O uso da AS permite que o usuário tenha acesso a faixas possíveis de valores onde os dados de entrada podem ser modificados sem que a solução ótima encontrada seja modificada. Entretanto, não fica claro para o tomador de decisões, que nada entende de programação linear, o que ocorre quando algum(ns) dos limites apresentados na AS não são respeitados.

No próximo capítulo é apresentada de maneira formal o modelo geral de misturas, bem como o modelo de mistura de carvões para obtenção do coque metalúrgico.