Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 5, N. 1, 2017.

Trabalho apresentado no CNMAC, Gramado - RS, 2016.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Um Algoritmo de Pontos Interiores na Resolução de Problemas de Programação Linear com Coeficientes Fuzzy

Wesley V. I. Shirabayashi[1]
Departamento de Matemática, UEM, Maringá, PR, Pós-doutorando na FEEC-UNICAMP
Akebo Yamakami[2]
Departamento de Sistemas e Energia, FEEC, UNICAMP, Campinas, SP
Juliana Verga[3]
Campus Avançado de Jandaia do Sul, UFPR, Jandaia do Sul, PR

Resumo. Existe uma grande variedade de trabalhos que tratam do problema de programação linear *fuzzy*, uma parte aborda condições de otimalidade e outra parte aborda modos de obter solução. Dentre estes, são poucos os trabalhos que utilizam técnicas de pontos interiores. Neste trabalho, apresentamos um algoritmo de pontos interiores para resolução do problema de programação linear com coeficientes fuzzy na função objetivo. Tal algoritmo é baseado no clássico algoritmo afim-escala e é do tipo factível, ou seja, precisa de um ponto interior factível para iniciar as iterações.

Palavras-chave. Pontos interiores, Programação linear *fuzzy*, Primal afim-escala.

1 Introdução

Muitos problemas reais envolvem algum nível de incerteza nos dados ou parâmetros, e até mesmo nas soluções. Em geral, isto não é levado em consideração na hora da modelagem do problema de programação linear convencional. Nesse sentido, baseados na teoria dos conjuntos *fuzzy* introduzida por Zadeh [14] e nos trabalhos subsequentes [2, 13, 16], surgiram os problemas de programação linear *fuzzy* (PLF).

É grande o número de trabalhos onde os autores estudam problemas PLF, suas propriedades e mostram estratégias para sua resolução [3, 4, 6, 10, 11, 15, 16]. Métodos de pontos interiores estão entre as técnicas mais utilizadas e efetivas na resolução de problemas de programação linear, e no entanto são poucos os trabalhos que utilizam esse tipo de abordagem em problemas de programação linear *fuzzy*.

Neste trabalho, apresentamos um método de pontos interiores para resolução do problema de programação linear *fuzzy* com coeficientes *fuzzy* na função objetivo. Essa abordagem é uma modificação do clássico algoritmo primal afim-escala [1].

[1]wvishirabayashi@uem.br
[2]akebo@dt.fee.unicamp.br
[3]juliana.verga@ufpr.br

2 Conceitos básicos sobre números *fuzzy*

Nesta seção apresentamos os conceitos básicos da teoria *fuzzy* e como realizar operações com números *fuzzy*. Para maiores detalhes, consultar Dubois e Prade [7] e Pedrycz e Gomide [12].

Definição 2.1. *Um conjunto* fuzzy A *é descrito por uma função de pertinência que mapeia os elementos de um universo* X *no intervalo unitário* $[0, 1]$*:*

$$\mu_A : X \to [0, 1].$$

As formas mais comuns de funções de pertinência são triangular e trapezoidal, outras formas incluem funções gaussiana, exponencial, etc.

Definição 2.2. *Um número triangular fuzzy é definido por* $\tilde{a} = (\underline{a}, a, \overline{a})$ *onde* a *é o valor modal (elemento do universo com grau de pertinência igual a* 1*),* $\underline{a}$ *e* $\overline{a}$ *são os limitantes, inferior e superior, respectivamente. Um número triangular fuzzy também pode ser denotado por* $\tilde{a} = (a, \alpha, \beta)$*, onde* $\alpha = a - \underline{a}$ *é o espalhamento à esquerda e* $\beta = \overline{a} - a$ *o espalhamento à direita (*$\alpha, \beta \neq 0$*).*

Definição 2.3. *Um número trapezoidal fuzzy ou um intervalo fuzzy é definido por* $\tilde{a} = (\underline{a}, a_1, a_2, \overline{a})$*onde* a_1 *é o extremo inferior do valor modal,* a_2 *o extremo superior do valor modal,* $\underline{a}$ *e* $\overline{a}$ *são os limitantes, inferior e superior, respectivamente. Um número trapezoidal fuzzy também pode ser denotado por* $\tilde{a} = (a_1, a_2, \alpha, \beta)$*, onde* $\alpha = a_1 - \underline{a}$ *é o espalhamento à esquerda e* $\beta = \overline{a} - a_2$ *o espalhamento à direita.*

Vejamos algumas operações com números triangulares *fuzzy* utilizando a representação com espalhamentos à esquerda e à direita.

Definição 2.4. *Sejam* $\tilde{a} = (a, \alpha_1, \beta_1)$ *e* $\tilde{b} = (b, \alpha_2, \beta_2)$ *dois números triangulares fuzzy e* $k \in \mathbb{R}$*. Definem-se as operações:*

- *Soma:* $\tilde{a} + \tilde{b} = (a + b, \alpha_1 + \alpha_2, \beta_1 + \beta_2)$;
- *Multiplicação por escalar:* $k\tilde{a} = (ka, k\alpha_1, k\beta_1)$, *se* $k \geq 0$
  $k\tilde{a} = (ka, -k\beta_1, -k\alpha_1)$, *se* $k < 0$.

As operações com números trapezoidais *fuzzy* são realizadas de modo análogo.

3 Método de pontos interiores primal afim-escala

O problema padrão de programação linear (PL) é definido por:

$$\begin{array}{ll} \min & c^T x \\ s.a & \left\{ \begin{array}{l} Ax = b \\ x \geq 0 \end{array} \right. , \end{array} \tag{1}$$

onde $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ e $c \in \mathbb{R}^n$.

DOI: 10.5540/03.2017.005.01.0457

O método de pontos interiores primal afim-escala para resolução de um problema de PL foi proposto por Barnes em [1] como uma modificação mais eficiente do método de Karmarkar, [9]. A idéia do método consiste em iniciar com x^0 factível e interior, isto é, $Ax^0 = b,\ x^0 > 0$, utilizar $Pd = P(-\nabla f(x)) = -Pc$ como direção de descida para manter a factibilidade, onde P é o projetor no núcleo de A, $P = I - A(AA^T)^{-1}A$. Não é só isso, para manter os iterandos equidistantes da fronteira das restrições é feito um reescalamento no espaço de busca, através de uma transformação afim.

A seguir apresentamos os passos do método primal afim-escala.

Algoritmo Primal Afim-Escala:

Dados x^0 interior factível ($Ax^0 = b,\ x^0 > 0$), $\lambda \in (0,1)$ e $\epsilon > 0$ (pequeno).

Para $k = 0, 1, 2, \ldots$, faça:

1: Reescalamento.

$X_k = diag(x^k)$

$D = AX_k$

$\bar{c} = X_k c$

2: Cálculo da direção.

$P = I - D(DD^T)^{-1}D$

$p = -P\bar{c}$

$d = X_k p$

3: Cálculo do tamanho do passo (mantendo a interioridade).

$\alpha_k = min\left\{\frac{-(x_j)^k}{d_j} | d_j < 0\right\}$

4: Atualização dos iterandos.

$x^{k+1} = x^k + \alpha_k \lambda d$

Até que $\frac{\|x^{k+1} - x^k\|}{\|x^k\|} < \epsilon$.

4 Programação linear *fuzzy*

Existe uma grande variedade de formulações para os problemas de programação linear *fuzzy* (PLF). Por exemplo, Buckley e Feuring, em [3], consideram um PLF onde todos os coeficientes e variáveis são números triangulares *fuzzy*. Já em [11], os autores abordam um PLF onde as variáveis são números *fuzzy*. As técnicas de resolução também são variadas, por exemplo, utilizando algoritmos evolutivos, variações de algoritmos Simplex, métodos de penalidade, etc. Também é comum a utilização de funções *ranking* para induzir uma ordem no conjunto de números *fuzzy*, pela ordem natural do números reais, [4, 10].

Existem poucos trabalhos que tratam de métodos de pontos interiores na resolução de problemas de programação linear *fuzzy*. Em [11], os autores adaptam o algoritmo de pontos interiores primal afim-escala para resolver um PLF onde as variáveis são números triangulares fuzzy. Em [15] os autores aplicam o algoritmo de Karmarkar, [9], ao problema de programação linear *fuzzy* onde os coeficientes da função objetivo são números *fuzzy* trapezoidais. Neste caso, eles utilizam uma função *ranking* para defuzzificar os componentes *fuzzy* que aparecem nos passos algorítmicos e para garantir que a sequência de pontos, gerada pelo algoritmo, reduza o valor da função objetivo com relação à essa função *ranking* linear.

DOI: 10.5540/03.2017.005.01.0457

A função *ranking* utilizada em [15], para um número trapezoidal fuzzy $\tilde{a} = (a_1, a_2, \alpha, \beta)$ é dada por:

$$R(\tilde{a}) = \frac{a_1 + a_2}{2} + \frac{1}{4}(\beta - \alpha). \tag{2}$$

Definição 4.1. *Dados dois números fuzzy $\tilde{a}$ e $\tilde{b}$, define-se que $\tilde{a} \prec (\preceq)\tilde{b}$, se e somente se $R(\tilde{a}) < (\leq)R(\tilde{b})$.*

Para o desenvolvimento deste trabalho adotaremos a seguinte formulação do problema de programação linear com coeficientes *fuzzy* na função objetivo:

$$\begin{array}{ll} \min & \tilde{c}^T x \\ s.a & \left\{ \begin{array}{l} Ax = b \\ x \geq 0 \end{array} \right. , \end{array} \tag{3}$$

onde $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ e $\tilde{c}$ é um vetor com n componentes *fuzzy*.

Nossa idéia é utilizar a técnica apresentada por Cantão em [5], onde a direção de busca *fuzzy* é discretizada, ou seja, são obtidas várias direções e é escolhida uma que de o menor valor da função objetivo, com relação a alguma função *ranking*, ou com relação a alguma ordem definida para números *fuzzy*. Neste caso, discretizaremos $\tilde{c}$ e utilizaremos essas discretizações no algoritmo primal afim-escala

4.1 Discretização de uma direção *fuzzy*

Dado $J \in \mathbb{N}$ e $\tilde{d}$ um vetor com n componentes *fuzzy*, sejam $(d_1)_i$ e $(d_2)_i$ os limitantes inferior e superior, respectivamente, de $\tilde{d}_i$ (i-ésimo componente *fuzzy* de $\tilde{d}$).

Algoritmo de Discretização:

1: Para cada $i = 1, \ldots, n$, faça:

$\delta_i = \dfrac{(d_2)_i - (d_1)_i}{J - 1}$.

2: Para cada $j = 1, \ldots, J$, faça:

$d^{(j)} = d^{(1)} + (j-1)\delta$,

onde $d^{(1)}$ é o vetor com os limitantes inferiores de cada $\tilde{d}_i$, e δ é o vetor com os δ_i's.

4.2 Algoritmo primal afim-escala modificado com direções discretizadas

Dados x^0 interior factível ($Ax^0 = b,\ x^0 > 0$), $J \in \mathbb{N}$, $\lambda \in (0,1)$ e $\epsilon > 0$.

Discretizar $\tilde{c}$ conforme Subseção (4.1).

Para $k = 0, 1, 2, \ldots$, faça:

1: Reescalamento.

$X_k = diag(x^k)$

$D = AX_k$

2: Cálculo da direção.

Para cada vetor c^j da discretização de $\tilde{c}$, faça: $\bar{c} = X_k c^j$

$P = I - D(DD^T)^{-1}D$

$p = -P\bar{c}$

$d = X_k p$

DOI: 10.5540/03.2017.005.01.0457

3: Cálculo do tamanho do passo (mantendo a interioridade).

$\alpha_j = min\left\{\frac{-(x_i)^k}{d_i}|d_i < 0\right\}$

4: Atualização dos iterandos.

$x^j = x^k + \alpha_j \lambda d$

x^{k+1} será o x^j que minimiza $\{(c^j)^T x^k | j = 1, 2, \ldots, J\}$ com relação a alguma ordem pré-definida.

Até que $\frac{\|x^{k+1}-x^k\|}{\|x^k\|} < \epsilon$.

4.3 Comparação de números *fuzzy*

Como já citado anteriormente existe uma grande variedade de métodos para comparar números *fuzzy*. Neste trabalho, para efeito de comparação de números *fuzzy*, utilizamos a função *ranking* (2) e também a chamada ordem lexicográfica, definida por Farhadinia em [8] e utilizada em [10]. Esta ordem definida em [8] utiliza quatro parâmetros para comparação de números *fuzzy*, e é uma ordem total, no sentido que quaisquer dois números *fuzzy* $\tilde{a}$ e $\tilde{b}$ são comparáveis, isto é, ou $\tilde{a} \prec \tilde{b}$, ou $\tilde{a} \succ \tilde{b}$ ou $\tilde{a} \approx \tilde{b}$. A seguir, definimos os parâmetros de comparação e a ordem lexicográfica para números *fuzzy* trapezoidais, conforme [8].

Definição 4.2. *Dado um número fuzzy trapezoidal* $\tilde{a} = (a_1, a_2, \alpha, \beta)$, *seja* $V(\tilde{a}) = (C(\tilde{a}), L(\tilde{a}), W(\tilde{a}), S(\tilde{a}))$, *onde:*

$$C(\tilde{a}) = a_1; \quad L(\tilde{a}) = a_1 - \alpha; \quad W(\tilde{a}) = a_2 - a_1 + \alpha + \beta; \quad S(\tilde{a}) = a_2 - a_1 + \tfrac{1}{2}(\alpha + \beta);$$

Em particular, pode-se comparar dois números *fuzzy* trapezoidais $\tilde{a}$ e $\tilde{b}$, pela seguinte sequência de passos:

1. Se $C(\tilde{a}) = C(\tilde{b})$ vá a 2. Se $C(\tilde{a}) < C(\tilde{b})$, então $\tilde{a} \prec \tilde{b}$, senão $\tilde{b} \prec \tilde{a}$;
2. Se $L(\tilde{a}) = L(\tilde{b})$ vá a 3. Se $L(\tilde{a}) < L(\tilde{b})$, então $\tilde{a} \prec \tilde{b}$, senão $\tilde{b} \prec \tilde{a}$;
3. Se $W(\tilde{a}) = W(\tilde{b})$ vá a 4. Se $W(\tilde{a}) < W(\tilde{b})$, então $\tilde{a} \prec \tilde{b}$, senão $\tilde{b} \prec \tilde{a}$;
4. Se $S(\tilde{a}) = S(\tilde{b})$, então $\tilde{a} \approx \tilde{b}$. Se $S(\tilde{a}) < S(\tilde{b})$, então $\tilde{a} \prec \tilde{b}$, senão $\tilde{b} \prec \tilde{a}$.

5 Exemplo

Nesta seção apresentamos um exemplo para ilustrar o funcionamento do algoritmo apresentado. O Algoritmo 1 foi implementado em Matlab, a letra A indica que foi utilizada a função *ranking* (2) para a comparação de números *fuzzy* e a letra B indica a utilização da ordem de lexicográfica. Também foram utilizados os seguintes parâmetros $\epsilon = 10^{-3}$, $J = 10$, $\lambda = 0.95$.

DOI: 10.5540/03.2017.005.01.0457

5.1 Exemplo 5.2 de [15]

Neste exemplo, os coeficientes são números trapezoidais *fuzzy* dados na forma $\tilde{a} = (a_1, a_2, \alpha, \beta)$ e consiste no seguinte:

$$\begin{aligned} \max \quad & (8,10,2,6)x_1 + (10,12,1,17)x_2 + (3,5,1,5)x_3 + (4,6,2,6)x_4 \\ & +(6,8,1,5)x_5 + (9,11,1,5)x_6 + (2,4,2,6)x_7 + (4,7,1,3)x_8 \end{aligned}$$

$$\begin{aligned} \text{suj. a} \quad & x_1 + x_2 + x_5 + x_6 \leq 5; \quad x_3 + x_4 + x_7 + x_8 \leq 10; \\ & 5x_1 + 3x_3 \leq 15; \quad 5x_2 + 3x_4 \leq 15; \quad 15x_5 + 8x_7 \leq 60; \quad 15x_6 + 8x_8 \leq 60; \\ & x_1, x_2 \leq 3; \quad x_3, x_4 \leq 5 \quad x_5, x_6 \leq 4; \quad x_7, x_8 \leq 7.5 \\ & x_i \geq 0, \ \ i = 1, 2, \ldots, 8. \end{aligned}$$

Colocando o problema na forma (3), utilizamos o seguinte ponto inicial:

$$x^0 = (1,1,1,1,1,1,1,1,1,6,2,2,4,4,3,3,6.5,6.5,7,7,37,37)^T.$$

Na Tabela 1 temos as soluções aproximadas, desconsideradas as variáveis de folga, obtidas pelo algoritmo com os dois modos de comparar números *fuzzy*, e também são apresentados os valores *fuzzy* de $\tilde{c}^T\overline{x}$ para cada umas das aproximações.

Tabela 1: Soluções aproximadas.

Algoritmo	$\overline{x}$
Alg. 1 - A	$(1.9939, 2.9999, 1.6762, 0.0001, 0.0006, 0.0056, 0.8342, 7.4894)$
Alg. 1 - B	$(1.9995, 3.0000, 1.6674, 0.0000, 0.0003, 0.0003, 0.8331, 7.4994)$
	$\tilde{c}^T\overline{x}$
Alg. 1 - A	$(82.6588, 120.1479, 17.8281, 98.8470)$
Alg. 1 - B	$(82.6659, 120.1652, 17.8327, 98.8330)$

As soluções obtidas pelos Algoritmos 1-A e 1-B foram levemente diferentes mas compatíveis com a obtida em [15], mesmo utilizando pontos iniciais diferentes. Solução obtida em [15]:

$$x^* = (1.9980, 2.9950, 1.6694, 0.0007, 0.0017, 0.0008, 0.8314, 7.4984)^T$$

$$c^T x^* = (82.6633, 120.1615, 17.8300, 98.8263).$$

6 Conclusões

Apresentamos um algoritmo de pontos interiores para resolver o problema de programação linear com coeficientes *fuzzy* na função objetivo. Através de um exemplo foi mostrado que tal algoritmo obtem solução para o problema, mas são necessários mais testes para atestar a eficiência e robustez do algoritmo.

DOI: 10.5540/03.2017.005.01.0457

Referências

[1] E.R. Barnes, A variation on Karmarkar's algorithm for solving linear programming problems, *Math. Program.*, 36:174–182, 1986.

[2] R. E. Bellman and L. A. Zadeh, Decision-making in a fuzzy environment, *Management Sciences*, 17:B141–B164, 1970/71.

[3] J. J. Buckley and T. Feuring, Evolutionary algorithm solution to fuzzy problems: Fuzzy linear programming, *Fuzzy Sets and Systems*, 109:35–53, 2000.

[4] L. M. Campos and J. L. Verdegay, Linear programming problems and ranking of fuzzy numbers, *Fuzzy Sets and Systems*, 32:1–11, 1989.

[5] L. A. P. Cantão, Programação Não-Linear com Parâmetros Fuzzy: Teoria e Algoritmos. Tese de Doutorado, FEEC/UNICAMP, 2003.

[6] M. Delgado, J. L. Verdegay and M. A. Vila, A general model for fuzzy linear programming, *Fuzzy Sets and Systems*, 29:21–29, 1989.

[7] D. Dubois and H. Prade. *Fuzzy sets and systems: theory and applications*. Academic Press, London, 1980.

[8] B. Farhadinia, Ranking fuzzy numbers based on lexicographical ordering, *Int. J. Math. Comput. Sci.*, 5:220–223, 2009.

[9] N. K. Karmarkar, A new polinomial time algorithm for linear programming problem, *Combinatorica*, 4:373–395, 1984.

[10] J. Kaur and A. Kumar, A new method to find the unique fuzzy optimal value of fuzzy linear programming problems, *J. Optim. Theory Appl.*, 156:529–534, 2013.

[11] A. Nagoor Gani, S. N. Mohamed Assarudeen. An approximate optimal solution for the fuzzy variable linear programming problem using interior point technique. *Int. J. of Pure and Applied Math.*, 85:395–404, 2013.

[12] W. Pedrycz and F. Gomide. *Fuzzy systems engineering: toward human-centric computing*. John Wiley & Sons, Hoboken, NJ, 2007.

[13] H. Tanaka, T. Okuda and K. Asai. On fuzzy-mathematical programming, *Journal of Cybernetics*, 1:37–46, 1974.

[14] L. A. Zadeh, Fuzzy sets, *Information and Control*, 8:338–353, 1965.

[15] Y. H. Zhong, Y. L. Jia, D. Chen and Y. Yang. Interior point method for solving fuzzy number linear programming problems using linear ranking function, *Journal of Appl. Mathematics*, vol. 2013, Art. ID 795098, 9 pages, 2013. DOI: 10.1155/2013/795098.

[16] H. J. Zimmermann, Fuzzy programming and linear programming with several objective functions, *Fuzzy Sets and Systems*, 1:45–55, 1978.

