Solving fuzzy systems of linear equations by a nonlinear programming method

H. Reynaerts[1] and S. Muzzioli[2]

[1] Department of Applied Mathematics and Computer Science,
Ghent University,
Krijgslaan 281, building S9, 9000 Gent, Belgium
(e-mail: huguette.reynaerts@UGent.be)

[2] Department of Economics,
University of Modena and Reggia Emilia,
V.le Berengario 51, 41100 Modena, Italy
(e-mail: muzzioli.silvia@unimore.it)

Abstract. Linear systems of equations, with uncertainty on the parameters, play a major role in various problems in economics and finance. In this paper fuzzy linear systems of the general form $A_1x + b_1 = A_2x + b_2$, with A_1, A_2, b_1 and b_2 matrices with fuzzy elements, are solved by means of a nonlinear programming method. The relation between this methodology and the algorithm proposed in [Muzzioli and Reynaerts, 2004] is highlighted. The methodology is finally applied to an economic and a financial problem.
Keywords: Fuzzy linear systems, fuzzy vector, nonlinear programming.

1 Introduction

Several problems in economics and finance boil down to the solution of a system of linear equations. When we only have some vague knowledge about the actual value of the parameters, it may be convenient to represent some or all of them with a fuzzy number. For such a fuzzy linear system $Ax = b$, where the elements a_{ij} of the $n{*}n$ matrix A and the elements b_i of the n-vector b are fuzzy numbers, the following solutions have been proposed: the classical solution X_C, the vector solution X_J and the marginal solutions X_E and X_I (see [Buckley and Qu, 1991]). In [Muzzioli and Reynaerts, 2004] this method is extended to the more general fuzzy system of equations $A_1x+b_1 = A_2x+b_2$, with A_1, A_2, b_1 and b_2 matrices with fuzzy elements. Further it is proved that the systems $Ax = b$ and $A_1x + b_1 = A_2x + b_2$ have the same vector solution if $A_1 - A_2 = A$ and $b_2 - b_1$=b. Finally an algorithm to find the vector solution, is introduced.

The aim of this paper is to investigate the solution of the fuzzy linear system by means of a nonlinear programming method and to highlight the relation between this methodology and the algorithm proposed in [Muzzioli and Reynaerts, 2004].

The plan of the paper is the following: in section 2 we recall the vector solution X_J and the algorithm in order to get this solution. In section 3 we show that the algorithm boils down to a nonlinear programming problem and we work out the Kuhn-Tucker conditions. In section 4 we apply the method to several examples. The last section concludes.

2 The vector solution of the fuzzy system $\boldsymbol{A_1x + b_1 = A_2x + b_2}$

A (triangular) fuzzy number f is defined by three numbers (f_1, f_2, f_3). An α-cut, $\alpha \in [0,1]$, of f is the interval $[\underline{f}(\alpha), \overline{f}(\alpha)]$, with:

$$\underline{f}(\alpha) = (1-\alpha)f_1 + \alpha f_2 \qquad \overline{f}(\alpha) = (1-\alpha)f_2 + \alpha f_3$$

In [Muzzioli and Reynaerts, 2004] we prove that the system $Ax = b$ (where A is a $n*n$- matrix of fuzzy numbers and b a n-vector of fuzzy numbers) and all linear systems $A_1x + b_1 = A_2x + b_2$, where $A_1 - A_2 = A$ and $b_2 - b_1 = b$, have the same vector solution X_J, as defined by [Buckley and Qu, 1991] if all matrices $A(0)$ with $A(0)_{ij} \in a_{ij}(0)$ are nonsingular.

The α-cuts of X_J are the following sets:

$$X_J(\alpha) = \{x \in \mathbb{R}^n \mid A(\alpha)x = b(\alpha), A(\alpha)_{ij} \in a_{ij}(\alpha), b(\alpha)_i \in b_i(\alpha)\}$$

The marginals of $X_{Jj}, j = 1, 2, \dots, n$, are obtained by projecting X_J on the coordinate axes. In the same paper we consider the following simple algorithm which finds directly the marginals of the vector solution X_J for each unknown. One solves $2^{n(n+1)}$ systems, for each α-cut, where each element of the extended coefficient matrix of those systems is either the lower or the upper bound of the α-cut of the corresponding element of the original fuzzy extended coefficient matrix. The final solution for each unknown, is investigated by taking the minimum and the maximum of the solutions found in each system for this unknown. Since for all parameters a_{ij}, b_i, $[\underline{a}_{ij}(\alpha_1), \underline{a}_{ij}(\alpha_1)] \subset [\underline{a}_{ij}(\alpha_1), \underline{a}_{ij}(\alpha_1)]$ and $[\underline{b}_i(\alpha_1), \underline{b}_i(\alpha_1)] \subset [\underline{b}_i(\alpha_1), \underline{b}_i(\alpha_1)]$ if $\alpha_1 > \alpha_2$, the minimal (resp. maximal) value of $x_k^*(\alpha_1)$ is always greater (resp. smaller) then the minimal value of $x_k^*(\alpha_2)$ and thus $[\underline{x}_k^*(\alpha_1), \overline{x}_k(\alpha_1)^*] \subset [\underline{x}_k^*(\alpha_2), \overline{x}_k(\alpha_2)^*]$.

This has as concequence that $x_k^*(1) \in [\underline{x}_k^*(0), \overline{x}_k^*(0)]$, for all k and thus the solution of the algorithm is always a fuzzy number. This procedure ensures that all possible solutions, consistent with the parameters of the system, are taken. A simplification of the previous method is to find the solutions for $\alpha = 1$ and $\alpha = 0$ and impose ex post a triangular form on the solution, whenever $x_j(1) \in [\underline{x}_j(0), \overline{x}_j(0)]$, for all j. In order to find $x_j(1)$, for all j, one just solves the crisp system, substituting $\alpha = 1$ in the fuzzy system. In order to find $[\underline{x}_j(0), \overline{x}_j(0)]$, for all j, one applies the algorithm for $\alpha = 0$. If $x_j(1) \in [\underline{x}_j(0), \overline{x}_j(0)]$, for all j, then one takes as solution the triangular fuzzy numbers $(\underline{x}_j(0), x_j(1), \overline{x}_j(0))$.

3 The nonlinear programming method

The algorithm can be considered as n nonlinear programming problems where:

- the object functions, $x_k^*(b_1,\ldots,b_n,a_{1,1},\ldots,a_{nn}), k=1,2,\ldots,n$ are the solutions of the system of equations considered as functions of the coefficients,
- with constraints:

$$\underline{b}_1(\alpha) \leq b_1 \leq \overline{b}_1(\alpha) \ldots \underline{b}_n(\alpha) \leq b_n \leq \overline{b}_n(\alpha)$$
$$\underline{a}_{1,1}(\alpha) \leq b_1 \leq \overline{a}_{1,1}(\alpha) \ldots \underline{a}_{n,n}(\alpha) \leq a_{n,n} \leq \overline{a}_{n,n}(\alpha)$$

The object functions should as well be minimized as maximized to find the extremes of the α-cuts of the solution.

The Kuhn-Tucker conditions should be verified for extrema. The Lagrange functions are the following for all $k=1,2,\ldots n$:

$$\begin{aligned} L_k(b_1,\ldots,b_n,a_{1,1},\ldots,a_{nn}) = {} & x_k^*(b_1,\ldots,b_n,a_{1,1},\ldots,a_{nn} \\ & - \lambda_1(b_1-\overline{b}_1(\alpha)) - \ldots - \lambda_n(b_n-\overline{b}_n(\alpha)) \\ & - \lambda_{n+1}(a_{1,1}-\overline{a}_{1,1}(\alpha)) - \ldots - \lambda_{n(n+1)}(a_{nn}-\overline{a}_{nn}(\alpha)) \end{aligned}$$

The (necessary) Kuhn-Tucker conditions for a maximum (resp. minimum) are:

$$\begin{aligned} & \underline{b}_i(\alpha) \leq b_i \leq \overline{b}_i(\alpha) \quad (b_i-\underline{b}_i(\alpha))\frac{\partial L_k}{\partial b_i}=0, \qquad \frac{\partial L_k}{\partial b_i} \leq 0 \quad (resp. \geq 0), \\ & \lambda_i \frac{\partial L_k}{\partial \lambda_i}=0, \qquad \lambda_i \geq 0 \qquad (resp. \leq 0), \quad \forall i=1,2,\ldots,n \end{aligned}$$

$$\begin{aligned} & \underline{a}_{ij}(\alpha) \leq a_{ij} \leq \overline{a}_{ij}(\alpha) \quad (a_{ij}-\underline{a}_{ij}(\alpha))\frac{\partial L_k}{\partial a_{ij}}=0, \qquad \frac{\partial L_k}{\partial a_{ij}} \leq 0 \qquad (resp. \geq 0), \\ & \lambda_{i*n+j}\frac{\partial L_k}{\partial \lambda_{i*n+j}}=0, \qquad \lambda_{i*n+j} \geq 0 \quad (resp. \leq 0), \qquad \forall i,j=1,2,\ldots,n \end{aligned}$$

Since the partial derivatives are:

$$\frac{\partial L_k}{\partial b_i} = \frac{\partial x_k^*}{\partial b_i} - \lambda_i \quad \forall i, \qquad \frac{\partial L_k}{\partial a_{ij}} = \frac{\partial x_k^*}{\partial a_{ij}} - \lambda_{i*n+j} \quad \forall i,j$$

$$\frac{\partial L_k}{\partial \lambda_i} = -(b_i-\overline{b}_i(\alpha)) \quad \forall i, \qquad \frac{\partial L_k}{\partial \lambda_{i*n+j}} = -(a_{ij}-\overline{a}_{ij}(\alpha)) \quad \forall i,j$$

the Kuhn-Tucker conditions for a maximum (resp. minimum) are:

$$\underline{b}_i(\alpha) \leq b_i \leq \overline{b}_i(\alpha) \quad \lambda_i \geq 0 \quad (resp. \leq 0)$$

$$(b_i - \underline{b}_i(\alpha))(\frac{\partial x_k^*}{\partial b_i} - \lambda_i) = 0 \tag{1}$$

$$\frac{\partial x_k^*}{\partial b_i} - \lambda_i \leq 0 \quad (resp. \geq 0) \tag{2}$$

$$\lambda_i(b_i - \overline{b}_i(\alpha)) = 0, \forall i = 1, 2, \ldots, n \tag{3}$$

$$\underline{a}_{ij}(\alpha) \leq a_{ij} \leq \overline{a}_{ij}(\alpha) \quad \lambda_{i*n+j} \geq 0 \quad (resp. \leq 0)$$

$$(a_{ij} - \underline{a}_{ij}(\alpha))(\frac{\partial x_k^*}{\partial a_{ij}} - \lambda_{i*n+j}) = 0 \tag{4}$$

$$\frac{\partial x_k^*}{\partial a_{ij}} - \lambda_{i*n+j} \leq 0 \quad (resp. \geq 0) \tag{5}$$

$$\lambda_{i*n+j}(a_{ij} - \overline{a}_{ij}(\alpha)) = 0, \forall i, j = 1, 2, \ldots, n \tag{6}$$

For a maximum (resp. minimum) the following cases can occur:

- Suppose that $\frac{\partial x_k}{\partial b_i} > 0$ (resp. < 0) then from (2) it follows that $\frac{\partial x_k}{\partial b_i} \leq$ (resp. $\geq$)λ_i and thus $\lambda_i \neq 0$. Then from (3) one concludes that $b_i^* = \overline{b}_i(\alpha)$.
- Suppose that $\frac{\partial x_k}{\partial b_i} < 0$ (resp. > 0) then, since $\lambda_i \geq$ (resp. $\leq$) 0 it follows that $\frac{\partial x_k}{\partial b_i} \neq \lambda_i$ and thus from (1) one concludes that $b_i^* = \underline{b}_i(\alpha)$.
- Suppose that $\frac{\partial x_k}{\partial b_i} = 0$ then the Kuhn-Tucker conditions are the following:

  $$(b_i - \underline{b}_i(\alpha))\lambda_i = 0, \qquad \lambda_i \geq 0, \qquad \lambda_i(b_i - \overline{b}_i(\alpha)) = 0$$

  and thus the necessary conditions hold for all $b_i \in [\underline{b}_i(\alpha), \overline{b}_i(\alpha)]$.

The same cases, with analogous conclusions, occur for the coefficients a_{ij}.

4 Economic examples

(1) The market price of a good and the quantity produced are determined by the equality between supply and demand. Demand is the amount of a good that consumers are willing and able to buy at a given price. Supply is the amount of a good producers are willing and able to sell at a given price. Suppose that demand and supply are linear functions of the price:

$$\begin{cases} q_d = a * p + b \\ q_s = c * p + d \end{cases},$$

where q_s is the quantity supplied, that is required to be equal to q_d, the quantity demanded, p is the price and a, b, c and d are coefficients to be estimated. Suppose that we have only some imprecise data on the relation between the quantity supplied and demanded at a given price, then we can naturally describe the parameters by fuzzy numbers. Due to the equilibrium conditions, the following fuzzy linear system should be solved:

$$\begin{cases} x_1 = a * x_2 + b \\ x_1 = c * x_2 + d \end{cases}$$

This corresponds (see [Muzzioli and Reynaerts, 2004]) to find the vector solution of the fuzzy system:

$$\begin{cases} x_1 - a * x_2 = b \\ x_1 - c * x_2 = d \end{cases}$$

If one applies the nonlinear programming method, the following object functions should be maximized (resp. minimized):

$$x_1(a,b,c,d) = \frac{bc - ad}{c - a} \qquad x_2(a,b,c,d) = \frac{b - d}{c - a}$$

with constraints:

$$\begin{array}{cc} \underline{a} \leq a \leq \overline{a}(< 0) & (0 <)\underline{b} \leq b \leq \overline{b} \\ (0 <)\underline{c} \leq c \leq \overline{c} & \underline{d} \leq d \leq \overline{d}(< 0) \end{array}$$

First of all we calculate the partial derivatives of the object functions:

$$\frac{\partial x_1}{\partial a} = \frac{c(b-d)}{(c-a)^2} \qquad \frac{\partial x_1}{\partial b} = \frac{c}{(c-a)}$$

$$\frac{\partial x_1}{\partial c} = \frac{-a(b-d)}{(c-a)^2} \qquad \frac{\partial x_1}{\partial d} = \frac{-a}{(c-a)}$$

$$\frac{\partial x_2}{\partial a} = \frac{(b-d)}{(c-a)^2} \qquad \frac{\partial x_2}{\partial b} = \frac{1}{(c-a)}$$

$$\frac{\partial x_2}{\partial c} = \frac{-(b-d)}{(c-a)^2} \qquad \frac{\partial x_2}{\partial d} = \frac{-1}{(c-a)}$$

Since $\frac{\partial x_1}{\partial a} > 0$ one obtains the maximum of x_1 for $a^{max} = \overline{a}$ and the minimum for $a^{min} = \underline{a}$.
Since $\frac{\partial x_1}{\partial b} > 0$ one obtains the maximum of x_1 for $b^{max} = \overline{b}$ and the minimum for $b^{min} = \underline{b}$.
Since $\frac{\partial x_1}{\partial c} > 0$ one obtains the maximum of x_1 for $c^{max} = \overline{c}$ and the minimum for $c^{min} = \underline{c}$.
Since $\frac{\partial x_1}{\partial d} > 0$ one obtains the maximum of x_1 for $d^{max} = \overline{d}$ and the minimum

for $d^{min} = \underline{d}$.
Since $\frac{\partial x_2}{\partial a} > 0$ one obtains the maximum of x_1 for $a^{max} = \overline{a}$ and the minimum for $a^{min} = \underline{a}$.
Since $\frac{\partial x_2}{\partial b} > 0$ one obtains the maximum of x_1 for $b^{max} = \overline{b}$ and the minimum for $b^{min} = \underline{b}$.
Since $\frac{\partial x_2}{\partial c} < 0$ one obtains the maximum of x_1 for $c^{max} = \underline{c}$ and the minimum for $c^{min} = \overline{c}$.
Since $\frac{\partial x_2}{\partial d} < 0$ one obtains the maximum of x_1 for $d^{max} = \underline{d}$ and the minimum for $d^{min} = \overline{d}$.
The solution to the system is:

$$([\frac{\underline{bc} - \underline{ad}}{\underline{c} - \underline{a}}, \frac{\overline{b}\overline{c} - \overline{a}\overline{d}}{\overline{c} - \overline{a}}], [\frac{\underline{b} - \overline{d}}{\overline{c} - \underline{a}}, \frac{\overline{b} - \underline{d}}{\underline{c} - \overline{a}}])$$

(2) The binary tree model of Cox *et al.* (1979) is used to price options and other derivative securities. A European call option is a financial security that provides its holder, in exchange for the payment of a premium, the right but not the obligation to buy a certain underlying asset at a certain date in the future for a specified price K. In the binary tree model of [Cox *et al.*, 1979] the following assumptions are made: (A1) the markets have no transaction costs, no taxes, no restrictions on short sales, and assets are infinitely divisible; (A2) the lifetime T of the option is divided into N time steps of length T/N; (A3) the market is complete; (A4) no arbitrage opportunities are allowed, which implies for the risk-free interest factor, $1 + r$, over one step of length T/N, that $d < 1 + r < u$, where u is the up and d the down factor. The European call option price at time zero, has a well-known formula in this model,

$$EC(K,T) = \frac{1}{(1+r)^N} \sum_{j=0}^{N} \binom{N}{j} p_u^j p_d^{N-j} \left(S(0)u^j d^{N-j} - K\right)_+ ,$$

where K is the exercise price, $S(0)$ is the price of the underlying asset at time the contract begins, p_u and p_d are the resp. up and down risk-neutral transition probabilities. Fundamental for the option valuation is the derivation of the risk neutral probabilities, which are obtained from the following system:

$$\begin{cases} p_u + p_d = 1 \\ up_u + dp_d = 1 + r. \end{cases} \tag{7}$$

The solution is given by:

$$p_u = \frac{(1+r) - d}{u - d} \qquad p_d = \frac{u - (1+r)}{u - d}.$$

In order to estimate the up and down jump factors from market data, the standard methodology (see Cox *et al.* (1979)) leads to set:

$$u = e^{\sigma\sqrt{T/N}}, d = e^{-\sigma\sqrt{T/N}},$$

where σ is the volatility of the underlying asset.

If there is some uncertainty about the value of the volatility, then it is also impossible to precisely estimate the up and down factors. [Muzzioli and Reynaerts, 2004] suggest to model the up and down jump factors by triangular fuzzy numbers.

A fuzzy version of the two equations of the system (7) should now be introduced. This can be done (for each equation) in two different ways, since for an arbitrary fuzzy number f there exists no fuzzy number g such that $f + g = 0$ and for all non-crisp fuzzy numbers $f + (-f) \neq 0$:

$$\begin{aligned} p_u + p_d &= (1,1,1) \\ p_u &= (1,1,1) - p_d \end{aligned}$$

respectively

$$\begin{aligned} up_u + dp_d &= (1+r, 1+r, 1+r) \\ up_u &= (1+r, 1+r, 1+r) - dp_d \end{aligned}$$

where p_u and p_d are the fuzzy up and down probabilities and u and d are triangular fuzzy numbers.

Therefore the linear system (7) can be rewritten in four different ways:

$$\begin{cases} p_u + p_d = 1 \\ up_u + dp_d = 1 + r, \end{cases} \tag{8}$$

$$\begin{cases} p_u = 1 - p_d \\ up_u + dp_d = 1 + r, \end{cases} \tag{9}$$

$$\begin{cases} p_u = 1 - p_d \\ dp_d = (1 + r) - up_u, \end{cases} \tag{10}$$

and

$$\begin{cases} p_u + p_d = 1 \\ dp_d = (1 + r) - up_u. \end{cases} \tag{11}$$

Different solutions to the same fuzzy linear system have been found in Muzzioli and Torricelli (2001), and in [Reynaerts and Vanmaele, 2003], by solving system (8) and system (9), respectively.

It is easy to see that the four systems have no classical solution, therefore we investigate the vector solution.

If one applies this algorithm to the financial example, one should solve the following systems:

$$\begin{cases} p_u + p_d = 1 \\ \underline{u}p_u + \underline{d}p_d = 1 + r. \end{cases}$$

$$\begin{cases} p_u + p_d = 1 \\ \overline{u}p_u + \underline{d}p_d = 1 + r. \end{cases}$$

$$\begin{cases} p_u + p_d = 1 \\ \underline{u}p_u + \overline{d}p_d = 1 + r. \end{cases}$$

$$\begin{cases} p_u + p_d = 1 \\ \overline{u}p_u + \overline{d}p_d = 1 + r. \end{cases}$$

The solutions to those systems are resp.:

$$\begin{cases} p_u = \frac{(1+r)-\underline{d}}{\underline{u}-\underline{d}} \\ p_d = \frac{\underline{u}-(1+r)}{\underline{u}-\underline{d}}. \end{cases}$$

$$\begin{cases} p_u = \frac{(1+r)-\underline{d}}{\overline{u}-\underline{d}} \\ p_d = \frac{\overline{u}-(1+r)}{\overline{u}-\underline{d}}. \end{cases}$$

$$\begin{cases} p_u = \frac{(1+r)-\overline{d}}{\underline{u}-\overline{d}} \\ p_d = \frac{\underline{u}-(1+r)}{\underline{u}-\overline{d}}. \end{cases}$$

$$\begin{cases} p_u = \frac{(1+r)-\overline{d}}{\overline{u}-\overline{d}} \\ p_d = \frac{\overline{u}-(1+r)}{\overline{u}-\overline{d}}. \end{cases}$$

The final solution is obtained by taking the minimum and maximum for each unknown:

$$\begin{cases} \underline{p_u} = min(\frac{(1+r)-\underline{d}}{\underline{u}-\underline{d}}, \frac{(1+r)-\underline{d}}{\overline{u}-\underline{d}}, \frac{(1+r)-\overline{d}}{\underline{u}-\overline{d}}, \frac{(1+r)-\overline{d}}{\overline{u}-\overline{d}}) \\ \overline{p_u} = max(\frac{(1+r)-\underline{d}}{\underline{u}-\underline{d}}, \frac{(1+r)-\underline{d}}{\overline{u}-\underline{d}}, \frac{(1+r)-\overline{d}}{\underline{u}-\overline{d}}, \frac{(1+r)-\overline{d}}{\overline{u}-\overline{d}}) \\ \underline{p_d} = min(\frac{\underline{u}-(1+r)}{\underline{u}-\underline{d}}, \frac{\overline{u}-(1+r)}{\overline{u}-\underline{d}}, \frac{\underline{u}-(1+r)}{\underline{u}-\overline{d}}, \frac{\overline{u}-(1+r)}{\overline{u}-\overline{d}} \\ \overline{p_d} = max(\frac{\underline{u}-(1+r)}{\underline{u}-\underline{d}}, \frac{\overline{u}-(1+r)}{\overline{u}-\underline{d}}, \frac{\underline{u}-(1+r)}{\underline{u}-\overline{d}}, \frac{\overline{u}-(1+r)}{\overline{u}-\overline{d}}). \end{cases}$$

Therefore, the vector of fuzzy numbers:

$$\begin{pmatrix} [\frac{(1+r)-\overline{d}}{\overline{u}-\overline{d}}, \frac{(1+r)-\underline{d}}{\underline{u}-\underline{d}}] \\ [\frac{\underline{u}-(1+r)}{\underline{u}-\underline{d}}, \frac{\overline{u}-(1+r)}{\overline{u}-\overline{d}}] \end{pmatrix},$$

is a solution to the system.

Note that the algorithm boils down to the following nonlinear programming problems (for each α):

$$\begin{aligned} & max_{u,d} \quad (\text{resp.}min_{u,d}) \quad \frac{1+r-d}{u-d} \\ & \text{where} \quad (1+r \leq)\underline{u} \leq u \leq \overline{u} \\ & \quad \text{and} \quad \underline{d} \leq d \leq \underline{d}(\leq 1+r) \end{aligned}$$

$$\begin{aligned} &max_{u,d}(\text{resp.}min_{u,d})\frac{u-(1+r)}{u-d} \\ &\text{where} \quad (1+r \leq)\underline{u} \leq u \leq \overline{u} \\ &\quad\text{and} \quad \underline{d} \leq d \leq \overline{d}(\leq 1+r) \end{aligned}$$

Since $\frac{\partial p_u}{\partial u} = \frac{d-(1+r)}{(u-d)^2} < 0$ the maximum of p_u is obtained for $u^{max} = \underline{u}$ and the minimum for $u^{min} = \overline{u}$.
Since $\frac{\partial p_u}{\partial d} = \frac{(1+r)-u}{(u-d)^2} < 0$ the maximum of p_u is obtained for $d^{max} = \underline{d}$ and the minimum for $d^{min} = \overline{d}$.
Since $\frac{\partial p_d}{\partial u} = \frac{(1+r)-d}{(u-d)^2} > 0$ the maximum of p_d is obtained for $u^{max} = \overline{u}$ and the minimum for $u^{min} = \underline{u}$.
Since $\frac{\partial p_d}{\partial d} = \frac{u-(1+r)}{(u-d)^2} > 0$ the maximum of p_d is obtained for $d^{max} = \overline{d}$ and the minimum for $d^{min} = \underline{d}$

5 CONCLUSIONS

In this paper we have investigated the solution of a fuzzy linear system of equations by resorting to a non-linear programming methodology.

We have applied the methodology proposed to two important economic applications.

Acknowledgements

H. Reynaerts would like to acknowledge the financial support by the BOF-project 001104599 of the Ghent University.

References

[Buckley and Qu, 1991]J.J. Buckley and Y. Qu. Solving systems of linear fuzzy equations. *Fuzzy sets and systems*, pages 33–43, 1991.

[Cox *et al.*, 1979]J.C. Cox, S.A. Ross, and S. Rubinstein. Option pricing, a simplified approach. *Journal of Financial Economics*, pages 229–263, 1979.

[Muzzioli and Reynaerts, 2004]S. Muzzioli and H. Reynaerts. Fuzzy binary tree model for european-style vanilla options. In I. Batyrshin, J. Kacprzyk, and L. Sheremetov, editors, *Fuzzy Sets and Soft Computing in Economics and Finance*, pages 222–229, 2004.

[Reynaerts and Vanmaele, 2003]H. Reynaerts and M. Vanmaele. A sensitivity analysis for the pricing of european call options in a binary tree model. In J. Bernard, T. Seidenfeld, and M. Zaffalon, editors, *Imprecise Probabilities and their Applications*, pages 467–481, 2003.