A Genetic Algorithm Approach for Solving Fuzzy Linear and Quadratic Equations

M. Hadi Mashinchi, M. Reza Mashinchi, and Siti Mariyam H. J. Shamsuddin

***Abstract*—**In this paper a genetic algorithms approach for solving the linear and quadratic fuzzy equations $\widetilde{A}\widetilde{x} = \widetilde{B}$ and $\widetilde{A}\widetilde{x}^2 + \widetilde{B}\widetilde{x} = \widetilde{C}$, where $\widetilde{A}$, $\widetilde{B}$, $\widetilde{C}$ and $\widetilde{x}$ are fuzzy numbers is proposed by genetic algorithms. Our genetic based method initially starts with a set of random fuzzy solutions. Then in each generation of genetic algorithms, the solution candidates converge more to better fuzzy solution $\widetilde{x}_b$. In this proposed method the final reached $\widetilde{x}_b$ is not only restricted to fuzzy triangular and it can be fuzzy number.

***Keywords*—**Fuzzy coefficient, fuzzy equation, genetic algorithms.

I. Introduction

If any environments can be modeled as an equation then the first and primary request is to find a solution for it. In uncertain environments, the modeled equation can be a fuzzy equation. The solution for fuzzy equations plays an important role in uncertain decision making.

Recenly, great efforts have been done to solve fuzzy equations. In literature, standard analytical techniques are proposed by Buckley and Qu in [1]. According to [2] standard analitical techniques are not suitable for equations such as $\widetilde{A}\widetilde{x}^5 + \widetilde{B}\widetilde{x}^4 + \widetilde{C}\widetilde{x}^3 + \widetilde{D}\widetilde{x}^1 = \widetilde{E}$. Buckley and Eslami proposed neural networks solution for triangular quadratic fuzzy equations in [3]. Fuzzy neural networks are applied to solve fuzzy linear systems with fuzzy triangular coefficient to find the real roots [4]. Recently, a fuzzy Monte Carlo method has been proposed for triangular fuzzy linear and quadratic equations [5].

In this paper, we apply genetic algorithms to find a solution $\widetilde{x}_b$ to satisfy the fuzzy linear and quadratic equations (if there exists any solution). There are two differences between our proposed method and most of other ones. First, it is a generalized form for [1, 3, 4, 5], since in our proposed method, the coefficients are not just restricted to triangular fuzzy numbers. Second, in contrast to [4] which can find only crisp solutions, our method can find both crisp and fuzzy ones.

Our new approach is confined for solving the linear and quadratic fuzzy equations [1] with genetic algorithms. A fuzzy linear equation is in form of $\widetilde{A}\widetilde{x} = \widetilde{B}$ and a fuzzy quadratic equation have the form of $\widetilde{A}\widetilde{x}^2 + \widetilde{B}\widetilde{x} = \widetilde{C}$, where $\widetilde{A}$, $\widetilde{B}$, $\widetilde{C}$, and $\widetilde{x}$ are not restricted to triangular fuzzy numbers and they can be any fuzzy numbers. A fuzzy set $\widetilde{A}$ on $\Re$, the set of real numbers, is defined in (1).

$$\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x)) \mid x \in \Re, \mu_{\widetilde{A}} : \Re \to [0,1]\}. \qquad (1)$$

$\widetilde{A}$ is fuzzy number if $\mu_{\widetilde{A}} : \Re \to [0,1]$ and it satisfies the following conditions [2]:

1. $\mu_{\widetilde{A}}(x)$ is upper semicontinous,
2. There exists real numbers a, b, c and d such that $a \le b \le c \le d$ and
   - 2.1 $\mu_{\widetilde{A}}(x) = 0$ outside some interval $[a,d]$,
   - 2.2 $\mu_{\widetilde{A}}(x)$ is monotonic increasing on $[a,b]$,
   - 2.3 $\mu_{\widetilde{A}}(x)$ is monotonic decreasing on $[c,d]$,
   - 2.4 $\mu_{\widetilde{A}}(x) = 1$ on $[b,c]$.

$\widetilde{A}$ is triangular fuzzy number [6], if

$$\mu_{\widetilde{A}}(x) = \begin{cases} \dfrac{x-m}{\alpha} + 1, & m - \alpha \le x \le m \\ \dfrac{m-x}{\beta} + 1, & m \le x \le m + \beta \\ 0, & otherwise \end{cases}$$

where α, β and m are real numbers standing for the left and right spreads and the peak of $\widetilde{A}$, respectively.

II. Genetic Algorithms

Genetic algorithms which are inspired from principles of natural evolution are widely used for optimization problems

Authors are with the Department of Multimedia and Computer Graphics, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, Johor, Malaysia (correspondence author's e-mail: h_mashinchi@yahoo.com).

in computer science. They works based on three operations, selection, crossover and mutation.

The primary step to use genetic algorithms for solving any optimization problem is that how to model the answers for problem as the chromosomes (individuals, solutions). Then initially, genetic algorithms produce random solutions for the problem in the first generation. Selection operator chooses a set of solutions from current generation according to their merits and their satisfactory value to be a good answer for the problem. Then, Crossover operator is applied on the selected solutions and produce off-springs from them. The mutation operator just changes some portions of the off-springs.

The selection operator works based on the fitness function which returns the satisfactory value for each individual to be a potential solution. One of the most important factors for convergence in genetic algorithm is choosing proper fitness function [7].

A. Modeling of Fuzzy Equation by Genetic Algorithms

Genetic algorithms are shown their good performance when there exists many variables that are needed to be optimized. Although, in linear and quadratic fuzzy equations there exists just $\tilde{x}_b$ to be optimized, but in case we need to find the best smooth shape for $\tilde{x}_b$, then the number of variables growth dramatically as we are required to apply many level sets. By α^{th} level set or $\alpha-cut$ for fuzzy set $\tilde{A}$ we mean the set in (2).

$$\tilde{A}_\alpha = \{x \in \Re \mid \mu(x) \geq \alpha\}; \alpha \in [0,1]. \tag{2}$$

Thus, to find the very smooth solution for equations such as $\tilde{A}\tilde{x} = \tilde{B}$ or $\tilde{A}\tilde{x}^2 + \tilde{B}\tilde{x} = \tilde{C}$, genetic algorithms are very potential to be applied. In Fig. 1, a general view of solving fuzzy equations by genetic algorithm is shown.

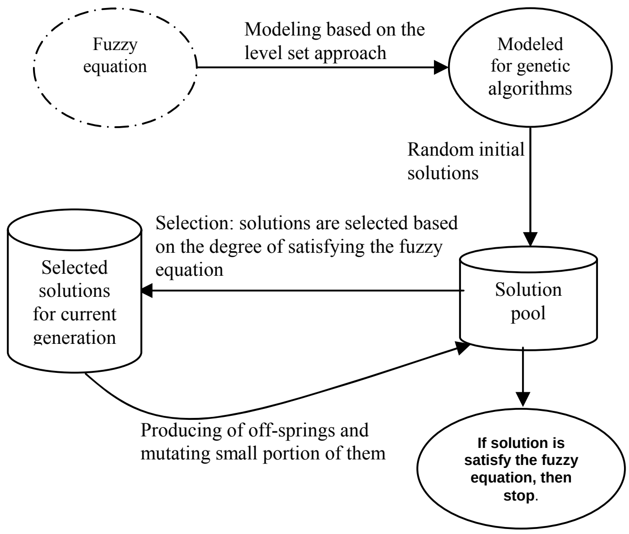

Fig. 1 Genetic algorithms process

In Section 3, it has been defined that how genetic algorithms are applicable to solve fuzzy equations.

III. Solving Fuzzy Quadratic and Linear Equations with Genetic Algorithms

In the proposed method, to find almost the best fuzzy solution for a fuzzy equation, first a solution is needed to be converted to its chromosome representation. The solution candidates for the fuzzy equations are converted to their level sets representation to be solvable by genetic algorithms. Fig. 2 shows a level set structure of the chromosome for a potential solution.

x_0^L	• • •	x_α^L	• • •	x_1^L	x_1^U	• • •	x_α^U	• • •	x_0^U

Fig. 2 Chromosome representation of a potential solution $\tilde{x}$ for fuzzy equation

In Fig. 2, by x_α^L and x_α^R, we mean the lower and upper bands of closed intervals for $\tilde{x}_\alpha = [x_\alpha^L, x_\alpha^R]$, respectively.

Then a proper fitness function is needed to be applied. The best solution for the fuzzy equation is the one which after substitution in the equation, the fuzzy value of the left hand side of the equation is very close to the fuzzy value of in the right hand side of the fuzzy equation. Therefore, a fitness function (or we may call it error function) based on the level sets is defined in Equation (3).

$$f(\tilde{x}) = \int_{\alpha=0}^{1} \left(| L_\alpha^L(x_\alpha^L, x_\alpha^U) - R_\alpha^L | + | L_\alpha^U(x_\alpha^L, x_\alpha^U) - R_\alpha^U | \right), \tag{3}$$

where $L_\alpha^L(x_\alpha^L, x_\alpha^U)$ and R_α^L are the lower bounds of α^{th} level sets for left side of the equation substituted with α^{th} level sets of the $\tilde{x}$ and fuzzy constant in the right side of the fuzzy equation, respectively. Similarly, $L_\alpha^U(x_\alpha^L, x_\alpha^U)$ and R_α^U are the upper bounds of α^{th} level sets for left hand side of the equation substituted with α^{th} level sets of the $\tilde{x}$ and fuzzy constant in the right side of the fuzzy equation respectively.

The aim is to find $\tilde{x}_b$ where $f(\tilde{x}_b) = 0$. Thus, for any $\tilde{x}_b$ if $f(\tilde{x}_b)$ is close to zero (the meaning of closeness is defined in section IV), then $\tilde{x}_b$ can be recognized as a good approximated solution of the fuzzy equation.

A. Fuzzy Linear Equations

Standardization is the first step for solving any fuzzy linear equation in the proposed method. Any fuzzy linear equation is required to be converted into standard form of $\tilde{A}\tilde{x} = \tilde{B}$. Then, assuming that $\tilde{x}$ is the potential solution for the equation, the fitness function is defined in Equation (4).

$$f(\tilde{x}) = \int_{\alpha=0}^{1} \left(| L_\alpha^L(x_\alpha^L, x_\alpha^U) - R_\alpha^L | + | L_\alpha^U(x_\alpha^L, x_\alpha^U) - R_\alpha^U | \right), \qquad (4)$$

where,

-
- $L_\alpha^L(x_\alpha^L, x_\alpha^U) = \min(A_\alpha^L x_\alpha^L, A_\alpha^U x_\alpha^L, A_\alpha^L x_\alpha^U, A_\alpha^U x_\alpha^U)$,
- $R_\alpha^L = B_\alpha^L$,
- $L_\alpha^U(x_\alpha^L, x_\alpha^U) = \max(A_\alpha^L x_\alpha^L, A_\alpha^U x_\alpha^L, A_\alpha^L x_\alpha^U, A_\alpha^U x_\alpha^U)$,
- $R_\alpha^U = B_\alpha^U$.

If $f(\tilde{x})$ is very close to zero, then $\tilde{x}_b = \tilde{x}$ could be a very good approximation of the solution. But, if after many generations in genetic algorithms, there is not any $\tilde{x}_b$ such that $f(\tilde{x}_b)$ is close to zero then the equation may not have any solution. However, in non- existence solution case, the captured $\tilde{x}_b$ which leads to the lowest value for $f(\tilde{x}_b)$ is almost the best fuzzy number which satisfies the fuzzy equation.

B. *Quadratic Fuzzy Equations*

To make the fitness function of a fuzzy quadratic equation, any form of fuzzy quadratic equations is required to be converted to standard form of $\tilde{A}\tilde{x}^2 + \tilde{B}\tilde{x} = \tilde{C}$. Then, the fitness function as defined in Equation (5) can be used.

$$f(\tilde{x}) = \int_{\alpha=0}^{1} \left(| L_\alpha^L(x_\alpha^L, x_\alpha^U) - R_\alpha^L | + | L_\alpha^U(x_\alpha^L, x_\alpha^U) - R_\alpha^U | \right), \qquad (5)$$

where,

- $L_\alpha^L(x_\alpha^L, x_\alpha^U) = \min(A_\alpha^L x_\alpha^L x_\alpha^L, A_\alpha^L x_\alpha^L x_\alpha^U, A_\alpha^L x_\alpha^U x_\alpha^U, A_\alpha^R x_\alpha^L x_\alpha^L, A_\alpha^R x_\alpha^L x_\alpha^U, A_\alpha^R x_\alpha^U x_\alpha^U) + \min(B_\alpha^L x_\alpha^L, B_\alpha^U x_\alpha^L, B_\alpha^L x_\alpha^U, B_\alpha^U x_\alpha^U)$,
- $R_\alpha^L = C_\alpha^L$,
- $L_\alpha^u(x_\alpha^L, x_\alpha^U) = \max(A_\alpha^L x_\alpha^L x_\alpha^L, A_\alpha^L x_\alpha^L x_\alpha^U, A_\alpha^L x_\alpha^U x_\alpha^U, A_\alpha^R x_\alpha^L x_\alpha^L, A_\alpha^R x_\alpha^L x_\alpha^U, A_\alpha^R x_\alpha^U x_\alpha^U) + \max(B_\alpha^L x_\alpha^L, B_\alpha^U x_\alpha^L, B_\alpha^L x_\alpha^U, B_\alpha^U x_\alpha^U)$,
- $R_\alpha^U = C_\alpha^U$.

Similarly with fuzzy linear equations, If $f(\tilde{x})$ is very close to zero, then $\tilde{x}_b = \tilde{x}$ could be a very good approximation of the solution. Proposed method is able to converge to only one solution each time. To find the other solutions (if there exists any other), it is needed to re-begin the genetic algorithm. If for all executions, the stopping criteria reached with same solution then the equation may have only one solution. But, if different solutions from each time of running are captured then there is set of solutions.

Some quadratic equations do not have any solution. Thus, if $f(\tilde{x}_b)$ does not converge to zero, then the equation may have not any solution. But the best $\tilde{x}_b$ which will be captured, can be recognized as the best fuzzy number for the equality $\tilde{A}\tilde{x}^2 + \tilde{B}\tilde{x} = \tilde{C}$.

IV. Simulations and Results

Different fuzzy linear and quadratic equations and are simulated with proposed method. For the stopping condition in genetic algorithms, maximum generation of 200 with population size of 50 chromosomes is considered. Similarly with [5], it is supposed that if the value of fitness function (error value) is less than 0.5, then a potential solution has been reached.

The later simulation examples are not solvable by methods in [1, 3, 4, 5], since [1, 3, 5] can only solve fuzzy equations with triangular fuzzy numbers and [4] is applicable to find just crisp roots.

Example 4.1

Equation $\tilde{A}\tilde{x} + \tilde{C} = \tilde{D}$, where $\tilde{A} = \tilde{C}$ and $\tilde{D}$ are fuzzy triangular numbers as in Fig. 3. After conversion to its standard form $\tilde{A}\tilde{x} = \tilde{B}$ where $\tilde{B} = (\tilde{D} - \tilde{C})$, then the solution $\tilde{x}_b$ is reached after 92 generations of genetic algorithms. The captured $\tilde{x}_b$ is shown in Fig. 4.

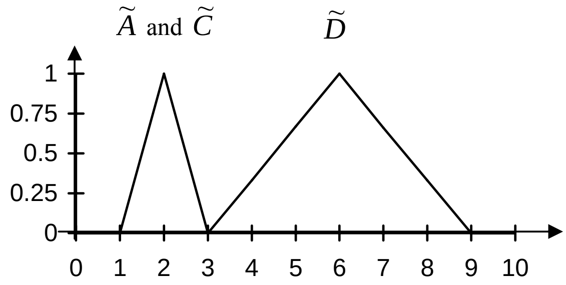

Fig. 3 Fuzzy coefficients $\tilde{A}$, $\tilde{C}$ and $\tilde{D}$ for fuzzy linear equation in Example 4.1

The solution for this fuzzy linear equation should be the crisp value $x_b = 2$. As it can be seen from Fig. 4, our method has found almost very good approximation for $\tilde{x}_b$. In this example, in the 60th generation, error is 0.3646 which is less than the error stopping condition. Better result can be reached, if less error condition is considered.

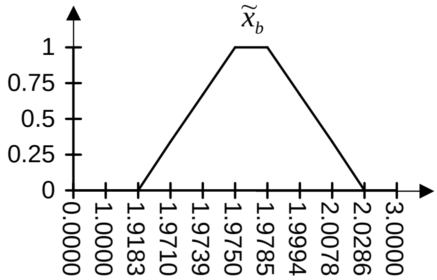

Fig. 4 Solution $\widetilde{x}_b$ which can satisfy the equation in Example 4.1

Example 4.2

In this example equation, we solve $\widetilde{A}\widetilde{x}+\widetilde{C}=\widetilde{D}$, where $\widetilde{A}$, $\widetilde{C}$ and $\widetilde{D}$ are fuzzy numbers as illustrated in Fig. 5. Similarly with Example 4.1, it is converted to standard form of $\widetilde{A}\widetilde{x}=\widetilde{B}$.

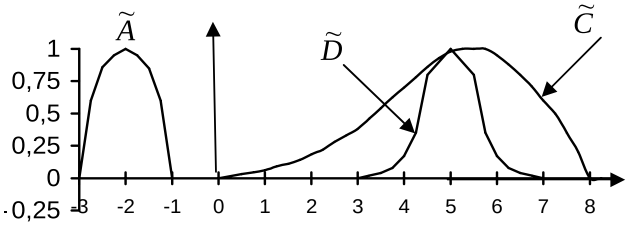

Fig. 5 Fuzzy coefficients $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{C}$ for fuzzy linear equation in Example 4.2

The level set intervals of captured $\widetilde{x}_b$, and also $\widetilde{A}$, $\widetilde{C}$ and $\widetilde{D}$ are shown in Table I. The approximation shape of $\widetilde{x}_b$ is illustrated in Fig. 6.

TABLE I
LEVEL SET INTERVALS OF CAPTURED $\widetilde{x}_b$, $\widetilde{A}$, $\widetilde{B}$ AND $\widetilde{C}$ WHICH CAN SATISFY THE EQUATION IN EXAMPLE 4.2

Level set	$\widetilde{x}_b$	$\widetilde{A}$	$\widetilde{C}$	$\widetilde{D}$
α=0	[-0.641,1.028]	[-3,-1]	[3,7]	[0,9]
α=0.34	[-0.516,0.361]	[-2.9,-1.1]	[4,6]	[2.9,7.5]
α=0.67	[-0.513,0.358]	[-2.7,-1.3]	[4.5,5.5]	[3.5,7]
α=1	[-0.461,-0.010]	[-2,-1]	[5,5]	[5,6]

The error convergence during the generations for the proposed method in this example is shown in Fig. 7. Here we have shown that our method is able to solve a fuzzy linear equations that have fuzzy number coefficients.

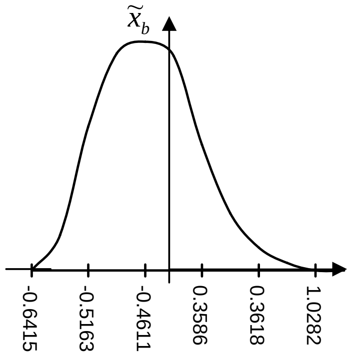

Fig. 6 Solution $\widetilde{x}_b$ which satisfies the equation in Example 4.2

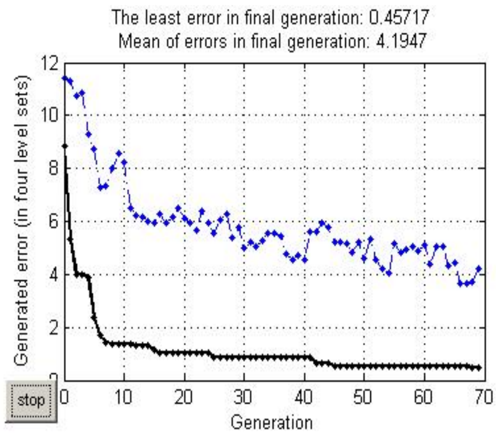

Fig. 7 Error Convergence of $\widetilde{x}_b$ to its optimized level

In Fig. 7 the points connected with thick line shows the error of the best solution in each generation and points connected with dashed line shows the mean of the error for 50 cchromosomes in each generation. In this example, after 69 generations error is 0.4517. Since this is the overall error for four level sets, the captured solution has average error of 0.11 in each level set.

Example 4.3

In this example equation $\widetilde{A}\widetilde{x}^2+\widetilde{B}\widetilde{x}=\widetilde{C}$, where $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{C}$ are fuzzy numbers shown in Fig. 8, is considered for solving.

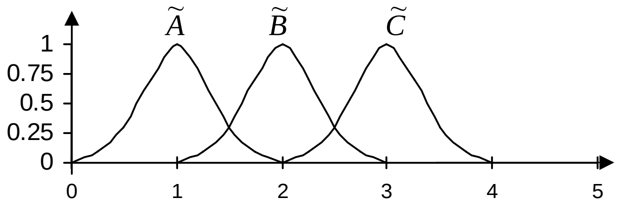

Fig. 8 Fuzzy coefficients $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{C}$ for fuzzy linear equation in Example 4.2

Here, for simplicity, four level sets are considered. After 86 generations an almost good approximation for $\tilde{x}_b$ is captured with overall error of 0.4263.

Table II shows the details of level sets for $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ and the final solution $\tilde{x}_b$. The shape of $\tilde{x}_b$ is shown in Fig. 9.

TABLE II
LEVEL SETS OF, $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ AND FINAL $\tilde{x}_b$ WHICH CAN SATISFIES THE EQUATION IN EXAMPLE 4.3

Level set	$\tilde{x}_b$	$\tilde{A}$	$\tilde{B}$	$\tilde{C}$
α=0	[0.006,0.425]	[2,4]	[1,3]	[0,2]
α=0.34	[0.078,0.419]	[2.1,3.9]	[1.1,2.9]	[0.1,1.9]
α=0.67	[0.170, 0.395]	[2.3,3.7]	[1.3,2.7]	[0.3,1.7]
α=1	[0.331,0.331]	[3,3]	[2,2]	[1,1]

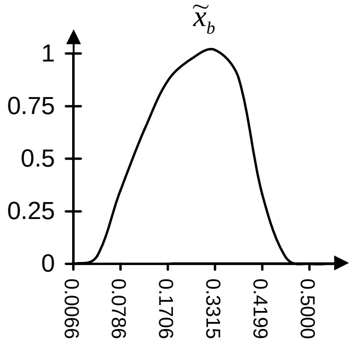

Fig. 9 Solution $\tilde{x}_b$ which satisfies the equation in Example 4.3

V. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a genetic algorithm approach to find a solution (if there exists) of fuzzy linear and quadratic equations. Our method is not restricted to just triangular coefficients as in [1, 3, 4, 5]. Also it is not confined to find crisp solution of the fuzzy equation as in [4]. The proposed method is a generalized form for solving fuzzy linear and quadratic equations, since it can solve fuzzy linear and quadratic equations with fuzzy numbers as coefficients and it can find both crisp and fuzzy solutions.

For the future work, we are going to apply our proposed method to solve general form of polynomial fuzzy equations with higher degrees and fuzzy matrix equations. Fuzzy matrix equations are in the form $\tilde{A}\tilde{x} = \tilde{B}$, where $\tilde{A}$ and $\tilde{B}$ are $n \times n$ and $n \times 1$ fuzzy matrixes. The goal is to find proper solution $\tilde{x}$ which is $n \times 1$ fuzzy matrix that can satisfy the mentioned fuzzy matrix equation. Solving fuzzy matrix equation has many applications in lots of disciplines such as fuzzy optimization area.

ACKNOWLEDGMENT

Authors would like to thank Research Management Center (RMC) of Universiti Teknologi Malaysia for its supports.

REFERENCES

[1] J.J. Buckley, Y. Qu, Solving fuzzy equations: a new solution concept, Fuzzy Sets and Systems 39 (1991) 291–301.
[2] S. Abbasbandy and B. Asady, "Newton's method for solving fuzzy nonlinear equations", Applied Mathematics and Computation, Vol. 159, pp. 349–356, 2004.
[3] J.J. Buckley and E. Eslami, "Neural net solutions to fuzzy problems: the quadratic equation", Fuzzy Sets and Systems, Vol. 86, pp. 289-298, 1997.
[4] S. Abbasbandy and M. Otadi, "Numerical solution of fuzzy polynomials by fuzzy neural network", Applied Mathematics and Computation, Vol. 181, pp. 1084-1089, 2006.
[5] J.J. Buckley and E. Eslami, "Solving Fuzzy Equations Using Monte Carlo Methods", Proceedings of Asian Fuzzy System Society International Conference, September 17- 20, China, pp-133-135, 2006.
[6] Hassan Mishmast Nehi, "Fuzzy linear programming, single and multiobjective functions", Ph.D of Mathematics Thesis, University of Kerman, 2003.
[7] R. A. Aliev, B. Fazlollahi, and R. M. Vahidov, "Genetic algorithm-based learning of fuzzy neural network, Part 1: feed-forward fuzzy neural networks", Fuzzy Sets and Systems, Vol. 118, No. 2, pp. 351-358, 2001.