Int. J. Contemp. Math. Sciences, Vol. 5, 2010, no. 16, 791 - 803

Solving a Fuzzy Nonlinear Optimization Problem by Genetic Algorithm

N. Ravi Shankar [1], G. Ananda Rao, J. Madhu Latha and V. Sireesha

Dept.of Applied Mathematics, GIS
GITAM University, Visakhapatnam, India
[1] e-mail: drravi68@gmail.com

Abstract

We present a new method to optimize the nonlinear objective function with fuzzy coefficients and fuzzy constraints using Genetic Algorithm (GA). We use GA to solve this fuzzy problem with defining the membership for fuzzy numbers. The proposed approach simulates every fuzzy number by distributing it into certain partition points. The final values obtained after the evolutionary process represent the membership grade of the fuzzy number. The computation of fuzzy equations by GA does not require the conventional extension principle or interval arithmetic and α-cuts for solving fuzzy nonlinear programming. The empirical results show that the proposed approach obtains very good solutions within the given bounds of each fuzzy coefficient compared with other fuzzy methods. The fuzzy concept of GA approach is different but gives better results than other traditional fuzzy methods.

Keywords: Fuzzy nonlinear programming, Genetic algorithm , Fuzzy number, Fuzzy ranking

1. Introduction

Fuzzy set theory has been applied to many disciplines such as control theory and operations research, mathematical modeling and industrial applications. The concept of fuzzy mathematical programming on general level was first proposed by Tanaka et al. [3]. The first formulation of fuzzy linear programming is proposed by Zimmermann [2]. Afterwards, many authors considered various types of fuzzy linear programming problems and proposed several approaches for solving these problems [4-10].

In many actual problems especially in the complexity industrial systems, there exist many kinds of fuzzy nonlinear production planning and scheduling problems and they can not be described and solved by traditional production planning and scheduling model, so, the research on modeling and optimization method for nonlinear programming under fuzzy environment is not only important in the fuzzy optimization theory but also great and wide value in application to the production planning and scheduling problems.

In this paper, we present a new method to optimize the nonlinear objective function with fuzzy coefficients and fuzzy constraints using Genetic Algorithm (GA). We use GA to solve this fuzzy problem with defining the membership for fuzzy numbers. The proposed approach simulates every fuzzy number by distributing it into certain partition points. The final values obtained after the evolutionary process represent the membership grade of the fuzzy number. The computation of fuzzy equations by GA does not require the conventional extension principle or interval arithmetic and α-cuts for solving fuzzy nonlinear programming.

The remainder of this paper is organized in the following way. In section 2, we discuss about fuzzy optimization, basic definitions and fuzzy conversion. Section 3 presents the proposed approach using GA. In section 4, we present a numerical example and compared results with crisp solution presented by Zimmerman. Finally we draw conclusions from the results in Section 5.

2. Fuzzy concepts

Definition 1 : If X is a collection of objects denoted generically by x then a fuzzy set Ã in X is a set of ordered pairs:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)), x \in X\} \tag{1}$$

$\mu_{\tilde{A}}(x)$ is called the membership function or grade of membership of x in Ã which maps X to the membership space M .The range of the membership function is a subset of the nonnegative real numbers whose supremum is finite. Elements with a zero degree of membership are normally not listed.

Example :

A realtor wants to classify the house he offers to his clients. One indicator of comfort of these houses is the number of bedrooms in it. Let X= {1,2,3 …10} be the set of available types of houses described by x = number of bedrooms in a house .Then the fuzzy set "comfortable type of house for a 4-person family" may be described as

Ã = { (1, .2),(2, .5),(3, .8),(4, 1),(5 , .7), (6, .3)}

Definition 2 : A fuzzy number Ã is a convex normalized fuzzy set Ã of the real line R such that

(i) there exists exactly one $x_0 \in R$ with $\mu_{\tilde{A}}(x_0) = 1$.

(ii) $\mu_A(x)$ is piecewise continuous.

Example :

The following fuzzy sets are fuzzy numbers:

Approximately 5 ={(3, 0.2), (4, 0.6), (5, 1), (6, 0.7) , (7, 0.1)}

Approximately 10 ={ (8, 0.3), (9, .7), (10, 1), (11, .7), (12, .3)}

Operations on fuzzy Number:

If Ã and B̃ are fuzzy numbers whose membership functions are continuous and onto from R to [0, 1].

Addition: Ã ⊕ B̃

$$\mu_{A\oplus B} = \sup_{z=x+y}\{\min\{\mu_A(x), \mu_B(y)\}\}$$

Example :

$$A = \{(2, 1), (3, 0.5)\}, B = \{(3, 1), (4, 0.5)\}$$

$$A\oplus B = \{(5, 1), (6, 0.5), (7, 0.5)\}$$

Subtraction: Ã Θ B̃

$$\mu_{A\Theta B} = \sup_{z=x-y}\{\min\{\mu_A(x), \mu_B(y)\}\}$$

Example :

$$A = \{(2, 1), (3, 0.5)\}, B = \{(3, 1), (4, 0.5)\}$$

$$A\Theta\, B = \{(-2, 0.5), (-1, 1), (0, 0.5)\}$$

Multiplication: Ã ⊗ B̃

$$\mu_{A\otimes B} = \sup_{z=x.y}\{\min\{\mu_A(x), \mu_B(y)\}\}$$

Example :

$$A = \{(2, 1), (3, 0.5)\}, B = \{(3, 1), (4, 0.5)\}$$

$$A \otimes B = \{ (6,1),(8,.5),(9,.5),(12,.5) \}$$

Definition 3 : A triangular membership function is specified by three parameters {a,b,c} as follows:

$$\text{Triangle}(x : a, b, c) = \begin{cases} 0 & , \quad x < a \\ (x-a)/(b-a) & , \quad a \le x \le b \\ (c-x)/(c-b) & , \quad b \le x \le c \\ 0 & , \quad x > c \end{cases}$$

Fuzzy conversion

We take 3 numbers one is actual number, remaining two are lower and upper limits respectively. Using the triangular membership function we convert the given number into discrete fuzzy number.

Let the triangular fuzzy number $\tilde{A} = (a,b,c)$ be replaced by an arbitrary fuzzy set in a given interval $[a,c]$.we divide the interval $[a,c]$ into t partitions, then $\Delta t = (c-a)/t$. Here we increase Δt from a. the grade of membership function $\mu(x)$ is defined as

$$\mu(x) = \begin{cases} 0 , & x < a \\ (x-a)/(b-a) , & a \le x \le b \\ (c-x)/(c-b) , & b \le x \le c \\ 0 , & x > c \end{cases}$$

Here, a, b, c are lower, actual and upper numbers.
If a= 10, b= 15, c=20.
Let t =10 then Δt =1.
If x=12, here $a \le x \le b$ so $\mu(x) = (x-a)/(b-a)$ = (12-10)/ (15-10) =0.4.
If x=16, here $b \le x \le c$ so $\mu(x) = (c-x)/(c-b)$ = (20-16)/ (20-15) =0.8.

Fuzzy number for $\tilde{A}$={(10,0),(11,0.2),(12,0.4),(13,0.6),(14,0.8),(15,1), (16,0.8),(17,0.6),(18,0.4),(19,0.2),(20,0)}.

Comparison of two fuzzy numbers using distance based method[1]

A fuzzy number $\tilde{A}$=(a,b,c ;1) is described as any fuzzy subset of the real line R with membership function $f_{\tilde{A}}$ which has the following properties:

(i) $f_{\tilde{A}}$ is a continuous mapping from R to the closed interval [0,1],

(ii) $f_{\tilde{A}}(x) = 0$, for all $x \in (-\infty, a]$,

(iii) $f_{\tilde{A}}$ is strictly increasing on $[a,b]$,

(iv) $f_{\tilde{A}}(x) = 1$, for $x = b$,

(v) $f_{\tilde{A}}(x)$ is strictly decreasing on $[b,c]$,

(vi) $f_{\tilde{A}}(x) = 0$ for all $x \in [c, +\infty)$, where a, b, c are real numbers.

The membership function $f_{\tilde{A}}$ of $\tilde{A}$ can be expressed as

$$f_{\tilde{A}}(x) = \begin{cases} f^L_{\tilde{A}}(x), & a \le x \le b, \\ 1 & x = b, \\ f^R_{\tilde{A}}(x), & b \le x \le c \;, \\ 0 & \text{otherwise} \end{cases}$$

where $f^L_{\tilde{A}}(x):[a,b] \to [0,1]$ and $f^R_{\tilde{A}}(x):[b,c] \to [0,1]$.

Since $f^L_{\tilde{A}}(x):[a,b] \to [0,1]$ is continuous and strictly increasing, the inverse function of $f^L_{\tilde{A}}(x)$ exists. Similarly, since $f^R_{\tilde{A}}(x):[b,c] \to [0,1]$ is continuous and strictly increasing, the inverse functions of $f^R_{\tilde{A}}(x)$ also exists.

The inverse functions of $f^L_{\tilde{A}}(x)$ and $f^R_{\tilde{A}}(x)$ can be denoted by $g^L_{\tilde{A}}(x)$ and $g^R_{\tilde{A}}(x)$ respectively. $g^L_{\tilde{A}}(x)$ and $g^R_{\tilde{A}}(x)$ are continuous on [0,1] that means both $\int g^L_{\tilde{A}}(x)$ and $\int g^R_{\tilde{A}}(x)$ over 0 to 1 exist.

The centroid point of a fuzzy number corresponds to a $\tilde{x}$ on the horizontal axis and a $\tilde{y}$ on the vertical axis. The centroid point ($\tilde{x}$, $\tilde{y}$) for a fuzzy number $\tilde{A}$ is defined as

$$\tilde{x}(\tilde{A}) = \frac{\int_a^b (x f_{\tilde{A}}^L)dx + \int_b^c (x f_{\tilde{A}}^R)dx}{\int_a^b f_{\tilde{A}}^L dx + \int_b^c f_{\tilde{A}}^R dx}$$

$$\tilde{y}(\tilde{A}) = \frac{\int_0^1 (y g_{\tilde{A}}^L)dy + \int_0^1 (y g_{\tilde{A}}^R)dy}{\int_0^1 g_{\tilde{A}}^L dy + \int_0^1 g_{\tilde{A}}^R dy}$$

Cheng proposed the distance based method[1] for ranking fuzzy numbers

$$R(\tilde{A}) = \sqrt{(\tilde{x})^2 + (\tilde{y})^2}$$

for any two fuzzy numbers $\tilde{A}_i$ and $\tilde{A}_j$,

If $R(\tilde{A}_i) < R(\tilde{A}_j)$, then $\tilde{A}_i < \tilde{A}_j$.

If $R(\tilde{A}_i) = R(\tilde{A}_j)$, then $\tilde{A}_i = \tilde{A}_j$.

If $R(\tilde{A}_i) > R(\tilde{A}_j)$, then $A_i > A_j$.

3. Proposed Method

A constrained optimization problem can be mathematically formulated as

$$\begin{aligned} &\text{Minimize} \quad f(x) \\ &\text{Subject to} \quad g_j(x) \le 0, \ j = 1,\dots, m \\ &\qquad x \in X \end{aligned}$$

where $f(x)$, the objective function, and $g_j(x)$, j = 1,……,m, the constraints , are defined on R^n , usually called solution space or feasible set, and X is a vector of n components $x_1, x_2, ..., x_n$.

This problem must be solved for the values $x_1, x_2, ..., x_n$ belonging to the solution space that satisfy the constraints and minimize the function f . This is to find a feasible point x^*, that is a point $x^* \in X$ satisfying all the constraints, such that $f(x^*) \le f(x)$ for each feasible point x. Then x^* is an optimal solution. Various optimization problems can be categorized based on the characteristics of X, $f(x)$ and $g_j(x)$, j=1,…,m. Thus if $f(x)$ and $g_j(x)$ are linear functions the model above describes a linear optimization problem otherwise the model becomes a nonlinear optimization problem which, in contrast to the linear case, in general is difficult to solve, and usually deterministic algorithms are not applicable.

Consider a nonlinear programming problem with fuzzy coefficients. From a mathematical point of view the problem can be addressed as:

$$\text{Minimize } g(\tilde{a}_1, \tilde{a}_2, \tilde{a}_3, \ldots, \tilde{a}_k, x)$$
$$\text{subject to } f_i(\tilde{b}_{i1}, \tilde{b}_{i2}, \tilde{b}_{i3}, \ldots, \tilde{b}_{ij}, x) \leq \tilde{c}_i, \quad i=1\ldots m,$$
$$x_i \in [l_j, u_j], \quad j = 1\ldots\ldots n, \quad l_i \geq 0$$

Where $x = (x_1,...,x_n) \in R^n$ is a n dimensional real-valued parameter vector, $[l_i, u_i] \subset R(i = 1,...,n)$, $g(x), f_i(x)$ are continuous arbitrary functions.

Proposed Approach using Genetic algorithm

Step 1 : Convert the coefficients of objective function and the constraints into fuzzy number using membership function. Choose population size p_s, crossover probability p_c, and mutation probability p_m. Choose a maximum allowable generation number $t_{\max}$. Set t=0.
Step 2 : Randomly select initial population.
Step 3: Check whether the members of initial population are satisfy or not of all constraints, if all constraints are satisfy then go to step 4 other wise go to step 2.
Step 4 : Evaluate the object function and select min/max according to our problem.
Step 5 : Calculate the fitness value of function for every member of population.
Step 6 : Check that the number of generations are less than that of fixed maximum allowable generations, if not terminate.
Step 7 : Select the some members from the present generation which has fitness value greater than average fitness value.
Step 8 : Apply crossover and mutation operators on selected members from the old generation.
Step 9 : Check whether the new population satisfy the constraints or not, if all the constraints are satisfy then go to step 4 other wise again apply the crossover and mutation operator on old population up to satisfy the all constraints.

We use binary coding for the variable $x_i \in [a_i, b_i]$,where a_i is the lower limit and b_i is the upper limit of x_i. To generate solution, we randomly take a string X_i of length l_i for every x_i whose characteristics are either 0 or 1.using this X_i we calculate x_i for every i.

$$n_i = \sum_{k=1}^{l_i} (2^k * X_k)$$

$$x_i = a_i + (b_i - a_i) * n_i / (2^{l_i} - 1).$$

Let $[a_i, b_i]$= [0, 6], string length l_i=10

X=1100100000

$$n = 2^9 * 1 + 2^8 * 1 + 2^7 * 0 + 2^6 * 0 + 2^5 * 1 + 2^4 * 0 + 2^3 * 0 + 2^2 * 0 + 2^1 * 0 + 2^0 * 0$$
$$n = 800$$
$$x_i = 0 + (6-0) * 800/1023 = 4.692$$

In step 3, using fuzzy addition and scalar multiplication we calculate left side of the constrains, we get one fuzzy number now we compare the two fuzzy numbers(left & right of relation) using the distance based method[1]. In step 4, Using the fuzzy addition and fuzzy scalar multiplication calculate the objective function value. Finally we get a fuzzy number. According to our problem select either maximum or minimum in that fuzzy number. In step 5, we calculate the fitness value of function for all population, if f_i is the objective value of population i.

For maximization, fitness value F_i of population i is $F_i = f_i$

For minimization, $F_i = \dfrac{1}{(1+f_i)}$

In step 7, After calculating the fitness value of each member, we give ranking based on fitness value. After that calculate the average fitness value, using this value select the some members from the present generation which has fitness value grater than average fitness value.

4. Numerical Example

A general nonlinear programming problem as follows:

$$\text{Minimize} \quad x_1^3 + x_2^3 - 30x_1^2 - 60x_2^2 + 300x_1 + 1200x_2 - 9000 \ \text{ subject to}$$
$$x_1^2 + x_2^2 - 10x_1 - 10x_2 > 50,$$
$$x_1^2 + x_2^2 - 12x_1 - 10x_2 < 21.81,$$
$$13 \le x_1 \le 100,$$
$$0 \le x_2 \le 100.$$

The known global optimal solution is $x^* = (14.095, 0.84296)$ and $f(x^*) = -6961.81$. In this example we consider coefficients of objective functions are fuzzy numbers,

$$\text{Minimize} \quad \tilde{1}x_1^3 + \tilde{1}x_2^3 - \tilde{30}x_1^2 - \tilde{60}x_2^2 + \tilde{300}x_1 + \tilde{1200}x_2 - \tilde{9000} \ \text{ subject to}$$
$$x_1^2 + x_2^2 - 10x_1 - 10x_2 > 50$$
$$x_1^2 + x_2^2 - 12x_1 - 10x_2 < 21.81$$
$$13 \le x_1 \le 100$$
$$0 \le x_2 \le 100$$

Lower, actual and upper numbers of given coefficients of objective function are presented in table I.

Table I : Fuzzy number representation

Fuzzy number	Lower number	Actual number	Upper number
$\tilde{1}$	0	1	2
$-\tilde{30}$	-31	-30	-29
$-\tilde{60}$	-61	-60	-59
$\tilde{300}$	299	300	301
$\tilde{1200}$	1199	1200	1201
$-\tilde{9000}$	-9001	-9000	-8999

After fuzzy conversion we get, the number of partitions t = 10

$\tilde{1}$ ={(0,0),(0.2,0.2),(0.4,0.4),(0.6,0.6),(0.8,0.8)(1,1),(1.2,0.8),(1.4,.6),
(1.6,.4),(1.8,.2),(2,0)}
$-\tilde{30}$={(-31,0),(-30.8,0.2),(-30.6,0.4),(-30.4,0.6),(-30.2,0.8),(-30,1),
(-29.8,0.8),(-29.6,0.6),(-29.4,0.4),(-29.2,0.2),(-29,0)}
$-\tilde{60}$={(-61,0),(-60.8,0.2),(-60.6,0.4),(-60.4,0.6),(-60.2,0.8),(-60,1),(-59.8,0.8),
(-59.6,0.6),(-59.4,0.4),(-59.2,0.2),(-59,0)}
$\tilde{300}$={(299,0),(299.2,0.2),(299.4,0.4),(299.6,0.6),(299.8,0.8),(300,1),(300.2,0.8),
(300.4,0.6),(300.6,0.4),(300.8,0.2),(301,0)}
$\tilde{1200}$={(1199,0),(1199.2,0.2),(1199.4,0.4),(1199.6,0.6),(1199.8,0.8),(1200,1),
(1200.2,0.8),(1200.4,0.6),(1200.6,0.4),(1200.8,0.2),(1201,0)}
$-\tilde{9000}$={(-9001,0),(-9000.8,0.2),(-9000.6,0.4),(-9000.4,0.6),(-9000.2,0.2),
(-9000,1),(-8999.8,0.8),(-8999.6,0.6),(-8999.4,0.4),(-8999.2,0.2),
(-8999,0)}

We use binary substrings of length 10 for both x_1 and x_2. Further, we select roulette-wheel selection, a single-point crossover operator, and a bit-wise mutation operator. The crossover probability p_c is set at 0.8 while the mutation probability p_m is set at 0.05. Population size p_s is set at 20 while the algorithm terminates at t_{max} =1000. The initial randomly generated population is presented in table II.

Table II : The initial randomly generated population

Substring1(X_1)	Substring2(X_2)	x_1	x_2
1001111001	1010001100	14.86	3.18
1010100001	1011011110	14.97	3.58
0111111001	0101011110	14.48	1.71
1000101101	0111001010	14.63	2.23
1010100000	1100010101	14.97	4.18
1010101000	1101011000	14.99	3.85
1000010110	0110010011	14.56	1.96
1001111101	1010000000	14.86	3.13
1000001111	0110001001	14.54	1.92
1010011101	1011000100	14.96	3.46
1011001000	1110111000	15.08	4.65
1001100110	1000110101	14.80	2.76
1010100101	1100000110	14.98	3.78
1000111011	0111100110	14.67	2.37
1010101010	1110100101	15.00	4.56
0111110100	0101101010	14.46	1.77
1001111111	1001100110	14.87	3.00
0111111011	0101101110	14.48	1.79
1010100111	1110100010	14.99	4.54
1001011111	1000110001	14.78	2.74

After substituting in objective function we get best fuzzy number among the populations is $x_1 = 14.480938$, $x_2 = 1.710655$

$f(x_1,x_2)$ = { (-7352.85,0),(-7087.85,0.2),(-6822.84,0.4),(-6557.83,0.6), (-6292.82,0.8),(-6027.82,1),(-6027.68,0.2),(-6027.47,0) }

After this we calculate the fitness value for every individual of the population (here we are taking f for calculating fitness value at μ=1).In selection process average fitness of the population is calculated, after this we form mating pool by selecting members from the old population which have fitness value grater than average fitness value. After selection we apply crossover operator on mating pool. Population after crossover and mutation are presented in table III and table IV respectively.

Table III : Population after crossover

Substring1(X_1)	Substring2(X_2)	x_1	x_2
1000111011	0001100110	14.67	0.49
1001111111	0001011000	14.87	0.43
1010100111	0001111010	14.99	0.59
1010100101	1111001001	14.99	4.73
1001111101	1011010110	14.87	3.54
1010100101	1100111000	14.99	4.03
0111111001	0111100000	14.48	2.34
1000111011	1100100101	14.67	3.93
1000010110	0110100000	14.56	2.03
1011001000	0010000010	15.09	0.63
1010100101	0000111101	14.99	0.29
1010100000	0001101101	14.97	0.53
0111110100	1100101100	14.46	3.96
1000001111	1001010101	14.54	2.91
1001111001	0111001011	14.85	2.24
1000111011	1001111001	14.67	3.09
1001111101	0010100010	14.86	0.79
1010101010	1010101000	15.00	3.32
1010101010	1110101000	15.00	4.57
1010011101	0001011011	14.96	0.44

Table IV : Population after Mutation

Substring1(X_1)	Substring2(X_2)	x_1	x_2
1001111001	1110001001	14.85	4.42
1001111111	0010100100	14.87	0.80
1010110111	0111101100	15.03	2.40
1011100000	0011010110	15.15	1.05
1111111101	0001000010	15.99	0.32
1000000011	0010110101	14.51	0.88
0000110001	1010010001	13.14	3.21
1000111001	0011011000	14.66	1.05
1000000110	0001000111	14.51	0.34
1000101000	0011010010	14.61	1.02
1110100101	0001101001	15.73	0.51
1110100000	0011001010	15.72	0.98
0111110100	1101000010	14.46	4.07
1001001010	0100011111	14.71	1.40
1001111000	0100000101	14.85	1.27
1000111011	1001100101	14.67	2.99
1011111011	1001000101	15.23	2.83
1011001010	1011011011	15.09	3.57
1000100111	1011001001	14.61	3.48
1010010101	1111101111	14.93	4.92

Two different methods were taken in this study for the purpose of analyzing the effectiveness of the algorithm. The first method was based on using the same population size but with different numbers of generations. The second method was based on using the grade of membership function. Each method had five runs. In the first method the fixed population size was 20 and the number of generations for each run was 100, 200, 300, 400 and 500, respectively. The best results of Z obtained in five runs at grade of membership value $\mu = 1$ are -6849.58, -6882.18, -6915.03, -6931.41 and -6936.928, respectively. Which are shown in Fig. 1. In the graph at 0 on x–axis we consider crisp value. we compare the crisp value with our proposed results.

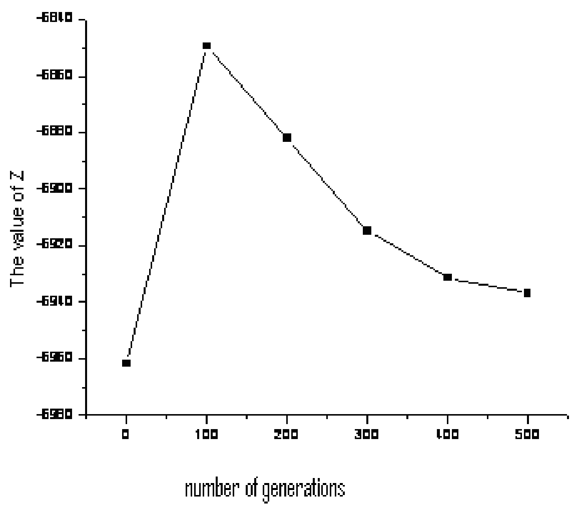

Fig. 1 : Objective function value at μ = 1.

The best results of Z obtained in five runs at grade of membership value μ=0.4 are -6979.747,-7012.0927,-7044.628,-7060.885,-7066.342. Which are shown in Fig. 2. In the graph at 0 on x–axis we consider crisp value. We compare the crisp value with our proposed results.

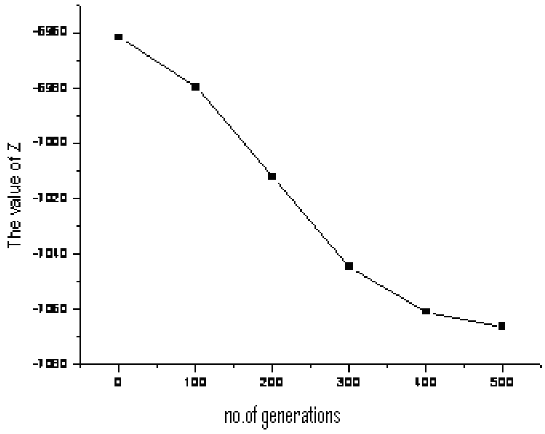

Fig. 2 : Objective function value at μ = 0.4

5. Conclusion

This study has investigated the genetic algorithm approach to solving fuzzy nonlinear programming problem with fuzzy coefficients and fuzzy constraints. This shows that the genetic algorithm is a good candidate tool for solving fuzzy nonlinear optimization problems without requiring mathematical reductions or transformations. The computation needs not to use the extension principle nor the interval arithmetic and α-cuts. The empirical results show that the proposed approach can obtain very good solutions within the given bound for each fuzzy coefficient that accomplishing flexible nonlinear programming. The results of this study may lead to the development of effective genetic algorithms for solving other model of fuzzy nonlinear programming .

References

[1] Y. Deng, Z. Zhenfu and Liu Qi , Ranking Fuzzy Numbers with an Area Method using radius of Gyration , Computers and mathematics with Applications **51** (2006) 1127-1136.

[2] H.J. Zimmermann, Fuzzy Programming and Linear Programming with Several Objective Functions , Fuzzy sets and systems, **1**(1978) 45-55.

[3] H. Tanaka, T. Okuda and K. Asai, On fuzzy mathematical programming, The Journal of Cybernetics **3** (1974) 37-46.

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

[5] S. Chanas , The use of parametric programming in fuzzy linear programming,Fuzzy Sets and Systems **11** (1983) 243-251.

[6] M. Delgado, J.L.Verdegay and M.A. Vila , A general model for fuzzy linear programming, Fuzzy sets and Systems **29** (1989) 21-29.

[7] K. Ganesan, P.Veeramani , Fuzzy linear programming with trapezoidal fuzzy numbers, Ann. Oper. Res. **143** (2006) 305-315.

[8] N. Mahdavi-Amiri, S.H. Nasseri ,Duality in fuzzy number linear Programming by use of a certain linear ranking function,Applied Mathematics and Computation **180** (2006) 206-216.

[9] N. Mahdavi-Amiri, S.H. Nasseri , Duality results and a dual simplex method for linear programming with trapezoidal fuzzy variables, Fuzzy Sets and Systems **158** (2007) 1961-1978.

[10] H.R. Maleki, M.Tata and M. Mashinchi, Linear programming with fuzzy variables, Fuzzy Sets and Systems **109** (2000) 21-33.

Received: September, 2009