APPLYING GENETIC ALGORITHMS TO SOLVE SOME FUZZY EQUATIONS

Feng-Tse Lin
Department of Applied Mathematics
Chinese Culture University, Yangminshan, Taipei, Taiwan, ROC
Email: ftlin@staff.pccu.edu.tw

Jing-Shing Yao
Department of Mathematics
National Taiwan University, Taipei, Taiwan, ROC

ABSTRACT

This study investigated a genetic approach to solve some fuzzy equations for optimization problems. We argue that the classical methods of solving fuzzy equations are too restrictive and that an alternative of the solution concept is possibly needed. By applying genetic algorithms to solve fuzzy equations, the use of membership functions for fuzzy numbers is no longer needed. We give some illustrative numerical examples to demonstrate both feasibility and efficiency of the proposed genetic algorithms while solving fuzzy equations.

1. INTRODUCTION

Solving fuzzy equations is one of the fundamental problems in fuzzy set theory [1, 5, 9]. In many practical applications, uncertainty will be modeled using fuzzy sets so that the parameters of the problem will be fuzzy numbers producing fuzzy equations to be solved [2, 3].

Consider a quadratic algebraic expression $ax^2 + bx$ for a, b real parameters and x a real variable. Let $y = N(a, b, x)$ = $ax^2 + bx$. After substituting triangular fuzzy numbers $\tilde{A}$, $\tilde{B}$, and $\tilde{X}$ into $ax^2 + bx$ for a, b, and x, respectively, we obtain the fuzzy set $\tilde{A}\tilde{X}^2 + \tilde{B}\tilde{X}^2$. The triangular fuzzy numbers are represented by $\tilde{A} = (a - \Delta_1, a, a + \Delta_2)$, $\tilde{B} = (b - w_1, b, b + w_2)$, and $\tilde{X} = (x - d_1, x, x + d_2)$ together with their membership functions $m(a \mid \tilde{A})$, $m(b \mid \tilde{B})$, and $m(x \mid \tilde{X})$, respectively. In the literature [3, 5], there are two ways of classical methods to evaluate the above fuzzy expression. The first method of finding the value of $\tilde{A}\tilde{X}^2 + \tilde{B}\tilde{X}^2$ is by using the extension principle. Let $Đ(a, b, x)$ denote the minimum of $m(a \mid \tilde{A})$, $m(b \mid \tilde{B})$, and $m(x \mid \tilde{X})$. Assume that we set $\tilde{Y}$ equal to $\tilde{A}\tilde{X}^2 + \tilde{B}\tilde{X}^2$. Then the membership function for $\tilde{Y}$ is defined by $m(x \mid \tilde{Y}) = sup\{Đ(a, b, x) \mid N(a, b, x) = y\}$. To find **a**-cuts of $\tilde{Y}$, we let $F(a) = \{N(a, b, x) \mid a \in \tilde{A}(a), b \in \tilde{B}(a), x \in \tilde{X}(a)\}$, $0 \le a \le 1$. Then we have $F(a) = \tilde{Y}(a)$. Alternately, the second method is that we evaluate $\tilde{A}\tilde{X}^2 + \tilde{B}\tilde{X}^2$ using **a**-cuts and interval arithmetic [3]. For any triangular fuzzy numbers $\tilde{A}$ and $\tilde{B}$, the following operations hold.

(1) $(\tilde{A}\ \tilde{B})(a) = \tilde{A}(a)\ \tilde{B}(a)$, and

(2) $(\tilde{A} \pm \tilde{B})(a) = \tilde{A}(a) \pm \tilde{B}(a)$.

The terms $\tilde{A}\ \tilde{B}$ and $\tilde{A} \pm \tilde{B}$ are computed using the extension principle and the terms $\tilde{A}(a)\ \tilde{B}(a)$, $\tilde{A}(a) \pm \tilde{B}(a)$ are computed using interval arithmetic. Let $\tilde{Z}(a) = \tilde{A}(a)\ \tilde{X}(a)\ \tilde{X}(a) + \tilde{B}(a)\ \tilde{X}(a)$, for $0 \le a \le 1$. We can see that $\tilde{Y}(a)$ is a subset of $\tilde{Z}(a)$. Obviously, a fuzzy algebraic expression using **a**-cuts and interval arithmetic for evaluation can produce a larger fuzzy set then the use of the extension principle.

In this paper, however, we propose a genetic approach to evaluate fuzzy algebraic expressions but without the need to define membership functions while using fuzzy numbers. By applying Genetic Algorithms (GAs) to solve fuzzy equations, the computation of fuzzy equations needs neither the extension principle nor **a**-cuts and interval arithmetic, but the usual evolution. Assume that $\tilde{X}$ is an arbitrary fuzzy set on the interval $[0, \frac{a}{b}]$ and $N(\tilde{X})$ is a fuzzy equation. In the genetic approach, we do not need to define the membership functions of $N(\tilde{X})$. Instead, the interval $[0, \frac{a}{b}]$ is equally divided into M pieces. Let $x_j = j\frac{a}{bM}$, $0 \le j \le M$. Let $\tilde{X}(x_j) = m_j \in [0,1]$,

$0 \le j \le M$, be the membership grade of x_j in $\tilde{\tilde{X}}$. Thus we obtain a discrete fuzzy set $\tilde{\tilde{X}}$ = $(m_0, m_1, ..., m_M)$, where m_j, $0 \le j \le M$, is a random number in [0, 1]. In other words, we wish to find $\tilde{\tilde{X}}$ in $[0, \frac{a}{b}]$ that maximizing $N(\tilde{\tilde{X}})$ via GAs. However, we cannot maximize $N(\tilde{\tilde{X}})$ directly since it is a fuzzy set. Instead, we can compute the centroid of fuzzy equations for the maximization problem. The centroid can be used as the fitness value for the evolution of GAs. Therefore all we need to do is to find a vector $\tilde{\tilde{X}}$ in $[0, 1]^{M+1}$ that maximizes the centroid.

2. STATEMENTS OF THE PROBLEM

The problem considered in this paper is stated as follows. We have three equations of the problem, the price, the cost, and the profit functions. The price function is

$$P(x) = a - bx, \; 0 \le x \le a/b, \tag{1}$$

where x is a demand, and the cost function is

$$C(x) = e + gx + kx^2, \; x \ge 0, \tag{2}$$

where a, b, e, g, k are all known positive numbers, and $b(a-g) < 2a(b+k)$, $g < a$. The objective of the problem is to obtain the maximization of the profit function *N(x)*, which is defined by

$$N(x) = xP(x) - C(x) = -e + (a-g)x - (b+k)x^2, \tag{3}$$

$0 \le x \le a/b$, and also to obtain the optimal price for the demand x. Since $N'(x) = a - g - 2(b+k)x$, we have $x = \frac{a-g}{2(b+k)}$ $(= x^*, say)$, $x \in (0, \frac{a}{b}]$, and hence, the maximum profit is

$$N(x^*) = -e + \frac{(a-g)^2}{4(b+k)}. \tag{4}$$

Now, consider the fuzzy case of the problem. Assume that the demand x is vague even for the same price $P(x)$ (see Fig. 1). Hence, the demand x can be represented as the following triangular fuzzy number,

$$\tilde{x} = (x - \Delta_1, x, x + \Delta_2), \; 0 < \Delta_1 < x, \; \Delta_2 > 0 \tag{5}$$

Clearly, the fuzzy profit is $N(\tilde{x})$. Let $N(x) = y$, then (3) becomes

$$(b+k)x^2 - (a-g)x + e + y = 0. \tag{6}$$

From (6), we can see that $D(y) = (a-g)^2 - 4(b+k)(e+y) \ge 0$ when $0 \le y \le N(x)$. If $D(y) \ge 0$, then (6) obviously has two roots which are given by the quadratic formula,

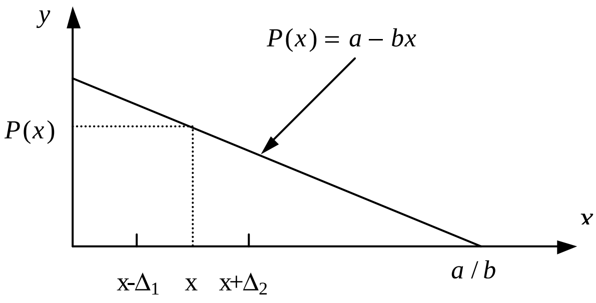


Fig. 1 The fuzzy demand *x*

$$r_1(y) = \frac{a - g - \sqrt{D(y)}}{2(b+k)},$$

$$r_2(y) = \frac{a - g + \sqrt{D(y)}}{2(b+k)}. \tag{7}$$

After applying the extension principle and doing some calculations, we obtain the membership function of fuzzy profit $N(\tilde{x})$ as follows, $N(\tilde{x})(y) =$

$$\sup_{y=N(x)} \tilde{x}(x) = \begin{Bmatrix} \max[\tilde{x}(r_1(y)), \tilde{x}(r_2(y))], \; if\, 0 \le y \le N \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad ,\text{otherwise} \end{Bmatrix} \tag{8}$$

where the membership function of the triangular fuzzy number $\tilde{x}$ is $\tilde{x}(x)$ and the membership function of the fuzzy profit $N(\tilde{x})$ is $N(\tilde{x})(y)$. However, the finding of $N(\tilde{x})(y)$ is not an easy task at all. Therefore we propose the genetic approach for evaluating fuzzy expressions but without the need to define membership functions for those fuzzy numbers. By applying GAs to solve fuzzy equations, the computation of fuzzy equations needs neither the extension principle nor α-cuts and interval arithmetic, but the usual evolution.

3. THE GENETIC APPROACH

The concept of proposed genetic approach is stated as follows. We first discretize triangular fuzzy number $\tilde{\tilde{X}}$ = (x - Δ_1, x, x + Δ_2) of (5) in an interval [0, a/b] yielding an arbitrary fuzzy set $\tilde{\tilde{X}}$ (see Fig. 2). Then we divide the interval [0, a/b] into M pieces, $x_j = j\frac{a}{bM}, 0 \le j \le M$.

Let the membership grade of $\tilde{\tilde{X}}$ at x_j be

$$\tilde{\tilde{X}}(x_j) = m_j, \; 0 \le j \le M \text{ and } m_j \in [0,1]. \tag{9}$$

Thus we obtain a discrete fuzzy set

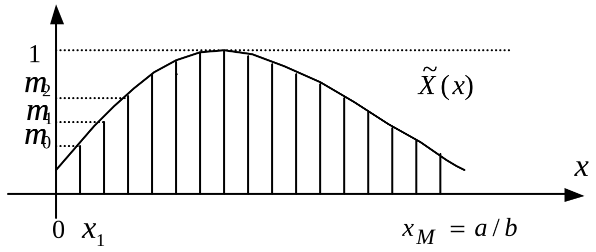


Fig. 2 Fuzzy set $\tilde{X}$

$\tilde{X} = (m_0, m_1, ..., m_M)$ or

$$\tilde{X} = \frac{m_0}{x_0} + \frac{m_1}{x_1} + ... + \frac{m_M}{x_M}. \qquad (10)$$

From $N(x) = y$ (in (3)) and (10), if each $N(x_k)$, $0 \le k \le M$, is different, the fuzzy profit function is defined by $N(\tilde{X}) = N(m_0, m_1, ..., m_M)$

$$= \frac{m_0}{N(x_0)} + \frac{m_1}{N(x_1)} + ... + \frac{m_M}{N(x_M)} \qquad (11)$$

and the centroid is defined by

$$\boldsymbol{q}(N(\tilde{X})) = \frac{\sum_{j=0}^{M} N(x_j) m_j}{\sum_{j=0}^{M} m_j}. \qquad (12)$$

From (6) and (7), if $0 \le y \le N$ and $N(x) = y$, the equations yield two solutions, i.e. $x = r_1(y), x = r_2(y)$. Therefore, in (11), if $N(x_i) = N(x_k) = y_0$ (say), $i \ne k$, $\frac{m_i}{N(x_i)} + \frac{m_k}{N(x_k)}$ becomes $\frac{\max(m_i, m_k)}{N(x_i)}$, and the centroid of $N(\tilde{X})$ becomes $\boldsymbol{q}(N(\tilde{X})) = \frac{R}{P}$, where

$$P = \sum_{j \ne i,k} m_j + \max(m_i, m_k) \text{ and}$$

$$R = \sum_{j \ne i,k} N(x_j) m_j + N(x_i)(\max[m_i, m_k]). \qquad (13)$$

Similarly, the optimal price for $\tilde{X}$ is

$$E(P(\tilde{X})) = \frac{\sum_{j=0}^{M} P(x_j) m_j}{\sum_{j=0}^{M} m_j}.$$

Obviously, with the proposed genetic approach, all we need to do is to find a vector $\tilde{X}$ in $[0, 1]^{M+1}$ (real numbers) that maximizing $q(N(\tilde{X}))$.

4. GENETIC ALGORITHMS

Genetic algorithms (GAs) are stochastic search technique based on the principles and mechanisms of natural genetics and natural selection [8]. The basic concepts of GAs are they start with a population of randomly generated candidates and evolve towards better solutions by applying genetic operators such as crossover and mutation, modeled on natural genetic inheritance and Darwinian survival-of-the-fittest [6, 7]. The proposed GAs for solving fuzzy equations is given as follows, which is different from the fuzzy genetic algorithms to solve fuzzy optimization problem in [4]. The general structure of the proposed GAs is given in the Appendix.

Step1. *Generate initial population.*
Initial population of size n is randomly generated from $[0, 1]^{M+1}$ according to uniform distribution in the closed interval [0,1]. Let the population be

$$\tilde{X}_j = (m_{j0}, m_{j_1}, ..., m_{j_M}) = \frac{m_{j0}}{x_0} + \frac{m_{j1}}{x_1} + ... + \frac{m_{jM}}{x_M} \qquad (14)$$

where $1 \le j \le n$ and m_{ji} is a real number in [0, 1], i = 0, 1, 2, ..., M. Each individual $\tilde{X}_j$ in the population is a chromosome.

Step 2. *Calculate the fitness value for each chromosome*.
For each chromosome $\tilde{X}_j$, $1 \le j \le n$, the centroid $\boldsymbol{q}(N(\tilde{X}_j))$ is calculated as the fitness. Then chromosomes can be rated in terms of their fitness. Let the total fitness for the population be $T = \sum_{j=1}^{n} \boldsymbol{q}(N(\tilde{X}_j))$. The cumulative fitness (partial sum) for each chromosome, $S_k = \sum_{j=1}^{k} \boldsymbol{q}(N(\tilde{X}_j))$, $1 \le k \le n$, is calculated. Next, intervals $I_1 = [0, S_1]$, $I_j = [S_{j-1}, S_j]$, j = 2, 3, .., n-1, and $I_n = [S_{n-1}, S_n]$ are constructed for the purpose of selection. In our implementation, a roulette wheel strategy is adopted as the selection procedure.

Step 3. *Selection and reproduction*.
The selection process begins by spinning the roulette wheel n times; each time, a single chromosome is selected for a new population in the following way. Each time a random number r from the range [0, T] is generated. If $r \in I_1$, then select first chromosome $\tilde{X}_1$; otherwise, select the

kth chromosome $\tilde{X}_k$, $2 \le k \le n$, if $r \in I_k$.. This selection process is continued until the new population has been created. Finally, rename the new population into $\tilde{Z}_1, \tilde{Z}_2, \tilde{Z}_3, \ldots$ in the order they were picked. The probability of a selection for each chromosome $\tilde{X}_j$ to appear in the new population is $\frac{q(N(\tilde{X}_j))}{T}$ and its expected value is $n\frac{q(N(\tilde{X}_j))}{T}$. Thus, this procedure tends to choose more $\tilde{X}_j$ with higher $q(N(\tilde{X}_j))$ fitness to go on into the next population.

Step 4. *Perform crossover.*

Crossover used here is the one-point method, which randomly selects one cut-point and exchanges the right parts of two parents to generate offspring. Each pair ($\tilde{Z}_1$, $\tilde{Z}_2$), ($\tilde{Z}_3$, $\tilde{Z}_4$), …, ($\tilde{Z}_{n-1}$, $\tilde{Z}_n$) produces two children via crossover. Let p be the probability of a crossover, $0 \le p \le 1$. Usually p is between 0.6 and 0.8, so we expect that, on average, 60% to 80% of chromosomes undergo crossover. Consider two chromosomes $\tilde{Z}_1$ and $\tilde{Z}_2$ for crossover. We generate a random number r in the interval [0, 1]. If $r \le p$, then perform crossover on $\tilde{Z}_1$ and $\tilde{Z}_2$. Otherwise, if $r > p$, the two children $\tilde{Z}_1'$ and $\tilde{Z}_2'$ are identical to their parents $\tilde{Z}_1$ and $\tilde{Z}_2$. Suppose we are to perform crossover. We generate another random integer number u from the range [0, M-1]. Assume that u equals 3, the two chromosomes $\tilde{Z}_1$ and $\tilde{Z}_2$ are cut after the fourth position, and the offspring $\tilde{Z}_1'$ and $\tilde{Z}_2'$ are generated by exchanging the right parts of them.

Step 5. *Perform mutation.*

Mutation resets one position to zero with a probability equal to the mutation rate. Let q be the probability of a mutation, $0 \le q \le 1$. Usually q is a very small value around 0.003 to 0.03, so we expect that, on average, 0.3% to 3% of total position of population would undergo mutation. There are $n \times m$ positions in the whole population; we expect 0.003 $n \times m$ to 0.03 $n \times m$ mutations per generation. For example, consider chromosome $\tilde{Z}_1'$. Let w_i be a random number in [0,1], $0 \le i \le M$. If $w_i < q$ then reset the $i+1$th position in $\tilde{Z}_1'$ to zero. After the mutation is completed on the whole population, we let $\tilde{X}_j = \tilde{Z}_j'$, $j = 1,2,\ldots, n$. Now we just completed one iteration of GAs. Step 2 through step 5 is done K times, where K is the maximum number of iterations.

Finally, the algorithm is terminated after K generations. Let the last population be $\tilde{X}_1^*, \tilde{X}_2^*, \ldots, \tilde{X}_n^*$. Now, we need to calculate the centroid of each chromosome $\tilde{X}_j^*$, $1 \le j \le n$, in the last population. The maximum value of $q(N(\tilde{X}_k^*))$ is the best chromosome in the population, i.e. $\max_{1\le j\le n} q(N(\tilde{X}_j^*)) = q(N(\tilde{X}_k^*))$ for some $k \in \{1, 2, .., n\}$. The best chromosome is definitely the optimal solution of the problem. Let the best chromosome be

$$\tilde{X}_k^* = (m_{k0}^*, m_{k1}^*, \ldots, m_{kM}^*) = \frac{m_{k0}^*}{x_0} + \frac{m_{k1}^*}{x_1} + \ldots + \frac{m_{kM}^*}{x_M}.$$

According to (1), the fuzzy price is

$$P(\tilde{X}_k^*) = a - b\tilde{X}_k^* = \frac{m_{k0}^*}{a - bx_0} + \frac{m_{k1}^*}{a - bx_1} + \ldots + \frac{m_{kM}^*}{a - bx_M},$$

and the centroid of $P(\tilde{X}_k^*)$ is

$$E(P(\tilde{X}_k^*)) = \frac{\sum_{j=0}^{M} (a - bx_j) m_{kj}^*}{\sum_{j=0}^{M} m_{kj}^*}. \qquad (15)$$

5. ILLUSTRATIVE EXAMPLES

In this section, we give three illustrative numerical examples to demonstrate both feasibility and efficiency of the genetic algorithms while solving fuzzy equations

.

Example 1. The problem instance is: the price function, $P(x) = 100 - 2x$, $0 \le x \le 50$; the cost function, $C(x) = 10 + x$; and the profit function, $N(x) = xP(x) - C(x) = -10 + 99x - 2x^2$, $0 \le x \le 50$. The optimal solution of the crisp case is obtained as follows. Since $N'(x) = 99 - 4x = 0$, we obtain x = 24.75. Thus the optimal solutions are P(24.75) = 50.5 and N(24.75) = 1215.125.

Now, consider the fuzzy case, i.e. the demand x is vague. Let M = 10. We have $x_k = k\frac{50}{10} = 5k$, $0 \le k \le 10$, as shown in Table 1. Then we obtain $N(x_0) = -10$, $N(x_1) = 435$, $N(x_2) = 780$, $N(x_3) = 1025$, $N(x_4) = 1170$, $N(x_5) = 1215$, $N(x_6) = 1160$, $N(x_7) = 1005$, $N(x_8) = 750$, $N(x_9) = 395$, and $N(x_{10}) = -60$. From (12) and (13), we have $N(\tilde{X}) = N(m_0, m_1, \ldots, m_{10})$ and the centroid

$q(N(\tilde{X})) = \frac{\sum_{j=0}^{10} N(x_j)m_j}{\sum_{j=0}^{10} m_j}$. The optimal price in the fuzzy sense is given by $E(P(\tilde{X})) = \frac{\sum_{j=0}^{10} P(x_j)m_j}{\sum_{j=0}^{10} m_j}$.

The parameters for genetic algorithms are: (1) population size n = 100; (2) the probability of a crossover p = 0.6; (3) the probability of a mutation q = 0.003; (4) the maximum number of iterations K = 4000. The result obtained from GAs for $\tilde{X}$ over 4000 generations is shown in Table 1. The count of crossover is 120,263 and the count of mutation is 1,225. We obtain $\tilde{X}$ = (0.00, 0.00, 0.00, 0.00, 0.00, 0.80, 0.00, …, 0.00) with the largest value of $q(N(\tilde{X}))$, 1215, and the optimal price in the fuzzy sense $E(P(\tilde{X}))$, 50.

Table 1 Best value for $\tilde{X}$ in the example1 over 4000 generations

j	0	1	2	3	4	5	6	7	8	9	10
x_j	0.0	5.0	10.0	15.0	20.0	25.0	30.0	35.0	40.0	45.0	50.0
m_j	0.00	0.00	0.00	0.00	0.00	0.80	0.00	0.00	0.00	0.00	0.00

Note: crossover : 120,263, mutation : 1,225, $q(N(\tilde{X}))$ = 1215, $E(P(\tilde{X}))$ = 50.

Example 2. The problem instance is: the price function is a quadratic, $P(x) = 101 - 10x + x^2$, $0 \le x \le 5$; the cost function, $C(x) = 10 + x$, $x \ge 0$; and the profit function is $N(x) = -10 + 100x - 10x^2 + x^3$, $0 \le x \le 5$. Since $b^2 - 4ac = 100 - 404 < 0$, then we have $x = 10/2 = 5$ and $N'(x) = 100 - 20x + 2x^2 = 2(x-5)^2 + 50 > 0$. In the crisp case of the problem, when $x = 5$, the optimal solution are $P(5) = 76$ and $N(5) = 365$.

Consider when x is vague. Let $M = 10$, we have $x_k = k\frac{5}{10} = 0.5k$, $0 \le k \le 10$, as shown in Table 2. We obtain $N(x_0) = -10$, $N(x_1) = 37.625$, $N(x_2) = 81$, $N(x_3) = 120.875$, $N(x_4) = 158$, $N(x_5) = 193.125$, $N(x_6) = 227$, $N(x_7) = 260.375$, $N(x_8) = 294$, $N(x_9) = 328.625$, and $N(x_{10}) = 365$.

From (13), the centroid is $q(N(\tilde{X})) = \frac{\sum_{j=0}^{10} N(x_j)m_j}{\sum_{j=0}^{10} m_j}$ and the optimal price in the fuzzy sense is $E(P(\tilde{X})) = \frac{\sum_{j=0}^{10} P(x_j)m_j}{\sum_{j=0}^{10} m_j}$. The parameters for genetic algorithms are:

(1) population size n = 100; (2) the probability of a crossover p = 0.6; (3) the probability of a mutation q = 0.003; (4) the maximum number of iterations K = 3,000. The result obtained from GAs for $\tilde{X}$ over 3,000 generations is shown in Table 2. The count of crossover is 89,732 and the count of mutation is 918. We obtain $\tilde{X}$ = (0.00, 0.00, 0.00, …, 0.00, 0.915) with the largest value of $q(N(\tilde{X}))$, 365, and the optimal price in the fuzzy sense $E(P(\tilde{X}))$, 76.

Table 2 Best value for $\tilde{X}$ in the examp le2 over 3000 generations

j	0	1	2	3	4	5	6	7	8	9	10
x_j	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
m_j	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.915

Note: crossover : 89,732, mutation : 918, $q(N(\tilde{X}))$ = 365, $E(P(\tilde{X})) = 76$.

Example 3. The problem instance is: the demand function is $p(x) = 30 - 2x$; the cost function, $C(x) = 10 + 6x$; and the profit function, $N(x) = -10 + 24x - 2x^2$, $0 \le x \le 15$. The optimal solutions of the crisp case are $N(6) = 62$ and $P(6) = 18$.

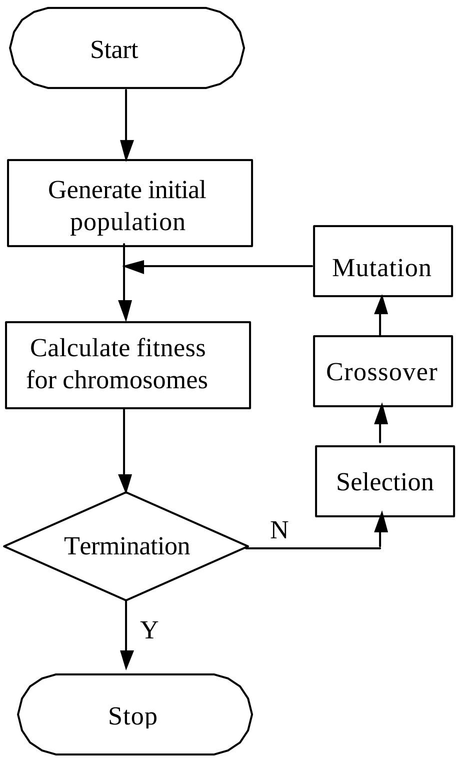
Now when x is vague. Let $M = 20$. We have $x_k = \frac{15}{20}k = 0.75k$, $0 \le k \le 20$. We obtain $N(x_0) = N(x_{16}) = -10$, $N(x_1) = N(x_{15}) = 6.875$, $N(x_2) = N(x_{14}) = 21.5$, $N(x_3) = N(x_{13}) = 33.875$, $N(x_4) = N(x_{12}) = 44$, $N(x_5) = N(x_{11}) = 51.875$, $N(x_6) = N(x_{10}) = 57.5$, $N(x_7) = N(x_9) = 60.875$, $N(x_8) = 62$, $N(x_{17}) = -29.125$, $N(x_{18}) = -50.5$, $N(x_{19}) = -74.125$, $N(x_{20}) = -100$. The parameters for genetic algorithms are: (1) population size n = 100; (2) the probability of a crossover p = 0.8; (3) the probability of a mutation q = 0.03; (4) the maximum number of iterations K = 6,000. The result obtained from GAs for $\tilde{X}$ over 6,000 generations is m_8 = 0.75 while the others are 0.00. The count of crossover is 240,271 and the count of mutation is 18,102. We obtain the largest value of $q(N(\tilde{X}))$, 62, and the optimal price in the fuzzy sense $E(P(\tilde{X}))$, 18.

6. CONCLUSION

In this study we proposed the genetic approach to solve fuzzy equations, but without the need to define the membership functions, using fuzzy numbers. Since the optimal demand x^*, e.g. $x^* = \frac{a-g}{2(b+k)}$, is in $(0, \frac{a}{b}]$, we can take M a suitable value to divide the interval [0, a/b] into small pieces, i.e. $x_j = j\frac{a}{bM}$, $0 \le j \le M$. Let $x_i = x^*$ for some $i \in \{0,1,2,\ldots,M\}$. From (9), we can see that the random numbers $m_1, m_2, \ldots, m_M$, which generated from [0, 1], should have a possibility of $m_j = 0, \forall j \neq i$ and $m_i \neq 0$. Thus (10) will become $\tilde{X} = (0,\ldots,0,m_i,0,\ldots,0) = \frac{m_i}{x_i} = \frac{m_i}{x^*}$. From (11), we obtain $N(0,\ldots,0,m_i,0,\ldots,0) = \frac{m_i}{N(x^*)}$, where $N(x^*)$ is the centroid and also is the maximum profit. Therefore, we conclude that the search space of GAs includes the crisp optimal solution and near-optimal solutions for the fuzzy case of the problem.

APPENDIX

The general structure of genetic algorithms.



REFERENCES

[1] J. J. Buckley and Y. Qu, “On using α-cuts to evaluate fuzzy equations”, Fuzzy Sets and Systems, 38, pp. 309-312, 1990.

[2] J. J. Buckley and Y. Qu, “Solving linear and quadratic fuzzy equations”, Fuzzy Sets and Systems, 38, pp. 43-59,1991.

[3] J. J. Buckley and Y. Qu, “Solving systems of linear fuzzy equations”, Fuzzy Sets and Systems, 43, pp. 33-43, 1991.

[4] J. J. Buckley and Y. Hayashi, “Fuzzy genetic algorithm and applications”, Fuzzy Sets and Systems, 61, pp. 129-136, 1994.

[5] J. J. Buckley, “Fuzzify”, Fuzzy Sets and Systems, 73, pp. 245-248, 1995.

[6] L. Davis (ed.), Genetic Algorithms and Simulated Annealing, Morgan Kaufmann, San Mateo, CA, 1987.

[7] M. Gen and R. Cheng, Genetic Algorithms & Engineering Design, John Wiley & Sons, Inc., New York, 1997.

[8] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine learning, Addison-Wesley, Reading, MA. 1989.

[9] M. Sakawa, K. Kato, H. Sunada, T. Shibano, “Fuzzy programming for multiobjective 0-1 programming problems through revised genetic algorithms”, European Journal of Operational Research, 97, pp. 149-158, 1997.