



Advance Fuzzy Programming Approach to Bi-Level Quadratic Fractional Programming Problems by Genetic Algorithm

Dr. Savita Mishra

Assistant Professor, Department of Mathematics,
LBSM College Jamshedpur, Kolhan University, Jharkhand, INDIA.

Indrani Dey

Research Scholar, Department of Mathematics,
Kolhan University, Jharkhand, INDIA.

Abstract: In the real world, we often encounter decision making situations involving multiple decision makers (DMs). Especially in industrial or governmental decision making situations, those DMs have different interest and decision priority. In this paper, an advance fuzzy programming technique through genetic algorithms for Bi-level quadratic programming problem (BLQFPP) is proposed by constructing the fitness function of the upper-level programming problem based on the definition of the feasible degree and will obtain a satisfactory solution for both the DMs. Finally, an illustrative numerical example is provided to clarify the proposed approach.

Key words: Bi-level programming problem, Fuzzy programming, Linear membership function, Satisfactory solution, Optimal solution, Fitness Function, Genetic algorithm.

I. INTRODUCTION

Bi-level programming (BLP) is a powerful technique for solving hierarchical decision-making problems. BLP is a tool for modeling decentralized decisions that consists of the objective of the leader at its first level and that of the follower at the second level. A BLPP is formulated for a problem in which two decision makers (DMs) makes decisions successively. For example, in a decentralized firm, top management or an executive of headquarters makes a decision such as budget of the firm, and then each division determines production plan in the full knowledge of the budget. In this section, an advance fuzzy programming technique through genetic algorithms for BLQFPPs is proposed by constructing the fitness function of the upper-level programming problem based on the definition of the feasible degree and will obtain a satisfactory solution for both the DMs. This GA avoids the use of penalty function to deal with the constraints, by changing the randomly generated initial population into an initial population satisfying the constraints in order to improve the ability of the GA to deal with the constraints. The method has no special requirement for the characters of the function and overcome the difficulty discussing the conditions and the algorithms of the optimal solution with the definition of the differentiability of the function. The concept of BLPP was first introduced by Candler and Townsley [7], as well as Fortuny –Amat and McCarl [9] presented the formal formulation of BLPP. The formulation and different version of BLPP are given by Bard [3], Candler [7], Herminia and Carmen [13] and Bialas and Karwan [6]. Bialas and Karwan [5] are the pioneers for linear BLPP who presented vertex enumeration method, called Kth- best solution. Anandalingam [1] proposed Stakelberg solution concept to MLPP as well as Bi-level decentralized programming problem. These were solved by simplex method. Zimmermann [22] first applied Fuzzy set theory in decision making problems with several conflicting objectives. Lai [15] first incorporated the concept of tolerance membership function of Fuzzy set theory to MLPP for obtaining satisfactory solution. Shin et al, Shin and Lee [18] extended Lai's satisfactory solution concept and theory, and proposed a supervised search approach by using non-compensatory max-min aggregation operator for MLPP. In the present era number of work has done in Fuzzy set theory and Fuzzy approach. In this study, we deal with the bi-level Quadratic fractional programming problem. Here we studies and designs the genetic algorithms of BLQFPP based on fuzzy approach. GA was first introduced by Holland [11], he introduced the concept of Adaptation of natural and artificial systems and since then it has been applied by different authors in many OR field. De Jong studied the behavior of a class of genetic adaptation system [8]; Baker introduced Adaptive selective method for GA [2]; Goldberg [10] and Michalewicz [17] have introduced the concept of GA. Wang [20] studied and designed an genetic algorithm (GA) of the bilevel linear programming problem (BLPP) by constructing the fitness function of the upper-level programming problem based on the

definition of the feasible degree. This GA avoids the use of penalty function to deal with the constraints, by changing the randomly generated initial population into an initial population satisfying the constraints in order to improve the ability of the GA to deal with the constraints. Wang Guangmin et al.[21] has introduced Genetic algorithm based on simplex method for solving linear-quadratic bilevel programming problem. Hecheng and Wang [12] proposed genetic algorithm for nonlinear bi-level programming problems etc. M.Sakawa presented [19] a Interactive fuzzy programming for Multi-level Non linear Integer programming problems through genetic algorithm . Mishra and Verma [16] presents an algorithm based on the fitness function for solving bi-level linear fractional programming problems.

Genetic Algorithm, which is a population based search technique it has been widely studied, experimented and applied in many fields in engineering worlds. Not only does GAs provide an alternate method to solving problem, it consistently outperforms other traditional methods in the most of the problem link. In general, GAs performs directed random searches through a given set of alternatives with the aim of finding the best alternative with respect to given criteria of goodness. These criteria are required to be expressed in terms of an objective function, which is usually referred to as fitness function. GA search for the best alternative (in the sense of a given fitness function) through ‘chromosomes’ evolution. In this paper we consider the solution of a bi-level quadratic fractional programming problems (BLQFPP) based on fuzzy approach by constructing the fitness function of the upper-level programming problem based on the definition of the feasible degree . This GA approach avoids the use of penalty function to deal with the constraints, by changing the randomly generated initial population into an initial population satisfying the constraints in order to improve the ability of the GA to deal with the constraints. In this paper we deal with the BLQFPPs with the essentially cooperative DMs and propose a solution procedure using a genetic algorithm for the problem.

II. BI-LEVEL QUADRATIC FRACTIONAL PROGRAMMING PROBLEM

A bi-level quadratic fractional programming problem (BLQFPP) consists of two levels , namely, the first level and the second level and each has quadratic fractional objective function. For instance, by adopting a criterion with respect to finance or corporate planning as an objective function at the upper level and employing a criterion regarding production planning as an objective function at the lower level, a bi-level linear or non-linear fractional programming problem can be formulated for hierarchical decision problems in firms.

III. FORMULATION OF BI-LEVEL QUADRATIC FRACTIONAL PROGRAMMING PROBLEMS

A bi-level quadratic fractional programming problem is mathematically formulated as:

$$\begin{aligned} \underset{x_1}{\text{Minimize}} \quad & z_1(x_1, x_2) \quad \text{Where } x_2 \text{ solves} \\ \underset{x_2}{\text{Minimize}} \quad & z_2(x_1, x_2) \quad \text{subject to} \quad A_1 x_1 + A_2 x_2 \leq b, \quad x_1 \geq 0, x_2 \geq 0. \end{aligned}$$

Where objective functions $z_i(x_1, x_2), (i = 1, 2)$ are represented by a quadratic fractional function (1)

$$z_i(x_1, x_2) = \frac{p_i(x_1, x_2)}{q_i(x_1, x_2)} = \frac{x_1' Q_{i1} x_1 + x_2' Q_{i2} x_2 + c_{i1} x_1 + c_{i2} x_2 + c_{i3}}{x_1' R_{i1} x_1 + x_2' R_{i2} x_2 + d_{i1} x_1 + d_{i2} x_2 + d_{i3}} \quad (2)$$

$x_i, (i = 1, 2)$ is an n_i - dimensional decision variables;

Q_{i1} and $R_{i1}, (i = 1, 2)$ are $n_1 \times n_1$ positive definite matrix.

Q_{i2} and $R_{i2}, (i = 1, 2)$ are $n_2 \times n_2$ positive definite matrix;

c_{i1} and $d_{i1}, (i = 1, 2)$ are n_1 —dimensional row vectors.

c_{i2} and $d_{i2}, (i = 1, 2)$ are n_2 —dimensional row vectors;

c_{i3} and $d_{i3}, (i = 1, 2)$ are constants.

b is an m -dimensional constant column vector; $A_i, (i = 1, 2)$ is an $m \times n_i$ constant matrix; and it is assumed that the denominators are positive i.e. $q_i(x_1, x_2) > 0, (i = 1, 2)$. For the sake of simplicity, we use the following notations:-

$x = (x_1, x_2) \in R^{n_1+n_2}, \quad c_i = (Q_{i1}, Q_{i2}, c_{i1}, c_{i2}, c_{i3}), \quad d_i = (R_{i1}, R_{i2}, d_{i1}, d_{i2}, d_{i3}), (i = 1, 2)$.

Also let DM1 denote the DM at the upper level and DM2 denote the DM at the lower level .In the bi-level quadratic fractional programming problem (2A.2.1.1), $z_1(x_1, x_2)$ and $z_2(x_1, x_2)$ respectively represent objective functions of DM1 and DM2 , and x_1 and x_2 represent decision variables of DM1 and DM2 respectively.

IV. FUZZY PROGRAMMING APPROACH TO BI-LEVEL QUADRATIC FRACTIONAL PROGRAMMING PROBLEM.

In contrast to formulation (1), in proposed dissertation, we employ the following representation in order to deal with a bi-level quadratic fractional problem with cooperative DMs. The decision maker first solve the upper level problem then the Lower level problem .

$$\text{Max}_{x_1} \quad z_1(x) = \frac{Q_{11}x_1^2 + Q_{12}x_2^2 + c_{11}x_1 + c_{12}x_2 + c_{13}}{R_{11}x_1^2 + R_{12}x_2^2 + d_{11}x_1 + d_{12}x_2 + d_{13}}$$

$$\text{subject to} \quad A_1x_1 + A_2x_2 \leq b, \quad x_1 \geq 0, \quad x_2 \geq 0, \quad (3)$$

$$\text{Max}_{x_2} \quad z_2(x) = \frac{Q_{21}x_1^2 + Q_{22}x_2^2 + c_{21}x_1 + c_{22}x_2 + c_{23}}{R_{21}x_1^2 + R_{22}x_2^2 + d_{21}x_1 + d_{22}x_2 + d_{23}}$$

$$\text{subject to} \quad A_1x_1 + A_2x_2 \leq b, \quad x_1 \geq 0, \quad x_2 \geq 0.$$

Let (x_1^u, x_2^u, z_1^u) be the solution for the upper level problem and (x_1^L, x_2^L, z_2^L) be the solution for the lower level problem. Then the decision maker at upper level gives some tolerance to the decision vector x_1^u and the objective function z_1 . Let $\pm p_1$ be the tolerance on x_1^u .

Let the membership function [Zimmermann (1991)][29] for decision vector x_1 be defined as:

$$\mu_{x_1}(x_1) = \begin{cases} [x_1 - (x_1^u - p_1)] / p_1, & (x_1^u - p_1) \leq x_1 \leq x_1^u \\ [(x_1^u + p_1) - x_1] / p_1, & x_1^u < x_1 \leq x_1^u + p_1 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Let the membership function [Zimmermann (1991)] for the objective function z_1 be defined as:

$$\begin{aligned} &1, z_1(x) \geq z_1^u \\ \mu_{z_1}[z_1(x)] &= [z_1(x) - z_1^L] / [z_1^u - z_1^L], z_1^L < z_1(x) \leq z_1^u \\ &0, z_1(x) < z_1^L \end{aligned} \quad (5)$$

The above information which is suitably represented mathematically is sent to the decision maker at lower level as his additional constraints. To rate the satisfaction of each of his potential solution, the decision maker at lower level builds the membership function for the objective function z_2 as:

$$\begin{aligned} &1, z_2(x) \geq z_2^u \\ \mu_{z_2}[z_2(x)] &= [z_2(x) - z_2^L] / [z_2^u - z_2^L], z_2^L < z_2(x) \leq z_2^u \\ &0, z_2(x) < z_2^L \end{aligned} \quad (6)$$

The auxiliary problem for the decision for the decision maker at lower level is

$$\text{Max: } \delta = \mu_{z_2}[z_2(x)] = [z_2(x) - z_2^L] / [z_2^u - z_2^L] \quad (7)$$

$$\text{Subject to} \quad A_1x_1 + A_2x_2 \leq b;$$

$$\mu_{x_1}(x_1) \geq \alpha I; \mu_{z_1}[z_1(x)] \geq \beta; \quad (8)$$

$$x_1, x_2 \geq 0; \alpha \in [0,1]; \beta, \delta \in [0,1];$$

Where α, β and δ are the minimum acceptable degree of satisfaction for decision vector x_1 , objective $z_1(x)$ and $z_2(x)$ respectively. I is a column vector whose elements are all '1' and is of the same dimension as x_1 . To resolve the conflict between the two decision makers and to avoid rejection of solution by the decision maker at

upper level, the decision maker at the lower level should try $\max \alpha, \beta$ and δ simultaneously. Let $\lambda = \min\{\delta, \alpha, \beta\}$, Then the lower level problem (8) is formulated as:

$$\text{Max: } \lambda \quad (9)$$

Subject to

$$\begin{aligned} A_1 x_1 + A_2 x_2 &\leq b; \\ [x_1 - (x_1^u - p_1)] / p_1 &\geq \lambda I; \\ [(x_1^u + p_1) - x_1] / p_1 &\geq \lambda I; \\ \mu_{z_1} [z_1(x)] &= [z_1(x) - z_1^L] / [z_1^u - z_1^L] \geq \lambda; \\ \mu_{z_2} [z_2(x)] &= [z_2(x) - z_2^L] / [z_2^u - z_2^L] \geq \lambda; \quad x_1, x_2 \geq 0; \lambda \in [0,1] \end{aligned} \quad (10)$$

The problem stated in equation (9) and (10) is a fuzzy programming problem (FPP). It is solved by the decision maker at lower level.

V. GENETIC ALGORITHM FOR BI-LEVEL QUADRATIC FRACTIONAL PROGRAMMING PROBLEM (BLQFPP)

The basic idea solving BLQFPP by GA is: firstly, choose the initial population satisfying the constraints, then the lower-level decision maker makes the corresponding optimal reaction and evaluate the individuals according to the fitness function constructed by the feasible degree, until the optimal solution is searched by the genetic operation over and over.

Genetic algorithm (GA) is search algorithms based on the mechanism of natural selection and natural genetics.. The primary concept behind the use of GAs is the representation of solutions to a problem in an encoded format. These encoded parameters (alleles) are referred to as genes and these are joined to build strings, which represent a potential solution to the problem. These strings of variables are called the chromosomes. Each gene can be represented by a binary string or a real value. The fitness of a chromosome as a candidate solution to a problem is an expression of the objective function represented by it. We concerned the GA is a numerical algorithm compatible for the optimization problem since it has no special requirements for the differentiability of the function. Hence the paper solves BLQFFPP based on fuzzy approach by GA.

(i) CODING AND CONSTRAINTS

There are many ways to represent a chromosome, in a GA. The first step is to devise a suitable representation scheme in designing a genetic algorithm for a particular problem. At present, the coding often used are binary vector coding and floating vector coding. But the latter is more near the space of the problem compared with the former and experiments show the latter converges faster and has higher computing precision [26]. The paper adopts the floating vector coding.

Hence the individual is expressed by:

$$V_k = (v_{k1}, v_{k2}, \dots, v_{km}).$$

The individuals of the initial population are generally randomly generated in GA, which tends to generate off-springs who are not in the constraint region. Hence, we must deal with them. Here, we deal with the constraints as follows: generate a group of individuals randomly, then retain the individuals satisfying the constraints $A_1 x_1 + A_2 x_2 \leq b$ as the initial population and drop out the ones not satisfying the constraints. The individuals generated by this way all satisfy the constraints. And, the off-springs satisfy the constraints by corresponding crossover and mutation operators.

(ii) DESIGN OF THE FITNESS FUNCTION

The selection strategy to breed a new generation from the proportion of existing population is one of the most important factors in the genetic search. Individual solutions are selected through a fitness-based process.. To solve the problem (P3) by GA, the definition of the feasible degree is firstly introduced and the fitness function is constructed to solve the problem by GA. Let d denote the large enough penalty interval of the feasible region for each $(x_1, x_2) \in X$:

Definition : Let $\Theta \in [0,1]$ denotes the feasible degree of satisfying the feasible region, and describe it by the following function.

$$\Theta = \begin{cases} 1 & \text{if } \|x_2 - Y(x_1)\| = 0 \\ 1 - \frac{\|x_2 - Y(x_1)\|}{d}, & \text{if } 0 < \|x_2 - Y(x_1)\| \leq d \\ 0 & \text{if } \|x_2 - Y(x_1)\| > d \end{cases} \quad (11)$$

Where, $\|\cdot\|$ denotes the norm.

Further, the fitness function of the GA can be stated as:

$$eval(v_k) = (z_1(x_1, y) - z_{1\min}) * \Theta$$

Where, $z_{1\min}$ is the minimal value of $z_1(x_1, x_2)$ on X .

(iii) GENETIC OPERATORS

In the optimization problem with continuous variable, many genetic operators appeared, the crossover operator is one of the important genetic operators. Simple crossover, heuristic crossover and arithmetical crossover are the most important operators. Among them, arithmetical crossover has the most popular application. The paper uses arithmetical crossover which can ensure the off-springs are still in the constraint region and moreover the system is more stable and the variance of the best solution is smaller. The arithmetical crossover can generate two off-springs which are totally linear combined by the father individuals. If v_1 and v_2 crossovers, then the final off-springs are:

$$\begin{aligned} v_1' &= \alpha * v_1 + (1 - \alpha) * v_2 \\ v_2' &= \alpha * v_2 + (1 - \alpha) * v_1 \end{aligned}$$

where $\alpha \in [0,1]$ is a random number. The arithmetical crossover can ensure closure (that is, $v_1, v_2 \in X$).

The mutation operator is another important genetic operator in GA. Mutation operator performs changes in a single individual. It randomly searches in the neighborhood of a particular solution. Its role is very important to guarantee that the whole search space is reachable. Many mutation operators appeared such as [5,26]: uniform mutation, non-uniform mutation and boundary mutation. We adopt the boundary mutation, which is constructed for the problem whose optimal solution is at or near the bound of the constraint search space. And for the problem with constraints, it is proved to be very useful. If the individual v_k mutates, then

$$v_k' = (v_{k1}', v_{k2}', \dots, v_{km}')$$

Where v_{ki}' is either left (v_{ki}^{left}) or right (v_{ki}^{right}) with same probability. where, left (v_{ki}^{left}), right (v_{ki}^{right}) denote the left, right bound of v_{ki}' , respectively.

Selection abides by the principle: the efficient ones will prosper and the inefficient will be eliminated, searching for the best in the population. Consequently the number of the superior individuals increases gradually and the evolutionary course goes along the more optimization. There are many selection operators. We adopt roulette wheel selection since it is the simplest selection.

(iv) COMPUTATIONAL PROCEDURES OF GENETIC ALGORITHM (GA)

The algorithm process using the GA to decide when to stop computing and return the result is as follows:

Step 1: Initialization: Set the parameters the population size M , crossover probability P_c , mutation probability P_m , the maximal generation of termination the algorithm T (maximal iteration generation MAXGEN), and then set the counter of generation $t=0$;

Step 2: Generating the initial population $P(0)$: The initial population $P(0)$ consists of a set of feasible chromosomes. Initialization of the initial population, M individuals are randomly generated in S , making up of the initial population. After generating sufficient chromosomes, go to next step;

Step 3: Computation of the fitness function: Evaluate the fitness value of the population according to the formula (1).

Step 4: Generate the next generation by genetic operators. Select the individual by roulette wheel selection, crossover according to the formula (2) and mutate according to the formula (3) to generate the next generation.

Step 5: Judge the condition of the termination. When t is larger than the maximal iteration number, stop the GA and output the optimal solution. Otherwise, let $t = t + 1$, turn to Step 3.

VI. THE NUMERICAL RESULT

In this section we present numerical example to demonstrate the solution procedures by proposed Fuzzy GA to solve bi-level quadratic fractional programming problem (BLQFPP). The following example considered by Mishra and Ghosh [21] is again used to demonstrate the solution procedures and clarify the effectiveness of the proposed approach:

Consider the following BLQFPP

$$\max_{x_1} z_1 = \frac{10x_1^2 + 15x_2^2 + 5}{x_1^2 + 2x_2^2 + 1} \quad \text{where } x_2 \text{ solves}$$

$$\max_{x_2} z_2 = \frac{25x_1^2 + 9x_2^2}{2x_1^2 + x_2^2 + 1}$$

subject to

$$4x_1 - 5x_2 \leq 15; \quad 3x_1 - x_2 \leq 21; \quad 2x_1 + x_2 \leq 27; \quad 3x_1 + 4x_2 \leq 45; \quad x_1 + 3x_2 \leq 30;$$

$$x_1 \geq 0, \quad x_2 \geq 0.$$

The compare of the results through 500 generations by the algorithms in the paper and the results in the references is shown in Table1.

Table1: Solution by proposed FGA approach

Parameters			Results			
M	Pc	Pm	X ₁	X ₂	Z ₁	Z ₂
50	0.7	0.15	5.34	1.27	9.6	12.20

M denotes the population scale; Pc denotes crossover probability; Pm denotes mutation probability; Z₁ and Z₂ are the objective function value of the upper-level and lower-level programming problem, respectively.

VII. CONCLUSION.

In this paper we consider the solution of a BLQFPP based on fuzzy approach by GA. In this approach optimal solution of the lower-level problem is dependent on the upper-level problem and considers the solution of each DM by randomly pairing up the decision maker (their solutions). Each pair of DMs (solution) give birth to new feasible trial solutions whose features are a random mixture of the features of the solutions of each decision makers. This is in accordance with a hierarchical system where the upper level DM is the main decision maker.

Compared with the traditional methods, the method has the following features:

- 1) The method has no special requirement for the characters of the function and overcome the difficulty discussing the conditions and the algorithms of the optimal solution with the definition of the differentiability of the function.
- 2) This GA avoids the use of penalty function to deal with the constraints, by changing the randomly generated initial population into an initial population satisfying the constraints in order to improve the ability of the GA to deal with the constraints.
- 3) In addition, in order to evaluate each individual, a fitness function was presented, the fitness evaluation, as a sub-procedure of optimization, can partly improve the leader's objective. Since the fittest members of the population are likely to become parents than others, a genetic algorithm tends to generate improving populations of trial solutions as it proceeds
- 4) Mutations occasionally occur so that certain children also can acquire features (sometimes desirable features) that are not possessed by either parent. This helps a genetic algorithm to explore a new, perhaps better part of the feasible region than previously considered.

The numerical result shows the proposed algorithm is feasible and efficient, can find global optimal solutions with less computational burden.

References

- [1] Anandilingam G, Apprey V (1991) Multi-level programming and conflicting resolution. European Journal of Operational Research 51:233-247
- [2] Baker J.E, Adaptive selection method for genetic algorithms and their applications. Proceedings of the International Conference on genetic Algorithms and their applications. (1985): 101-111.
- [3] Bard J.F. An Algorithm for Solving the Bi-level Programming Problem. Math and O. R. 8(2)(1983); 260-270.

- [4] Bard J.F. Optimality Conditions for the Bi-level Programming Problem. *Naval Res. Logist. Quart.*, 31(1984):13-26.
- [5] Biales W.F, Karwan M.H. Two-level linear programming, *Management Science*, 30(80), (1984): 1001-1020.
- [6] Bialas W.F, Karwan M.H. On two-level Optimization, *IEEE Trans. Automatic Control*, AC-27(1982): 211-214.
- [7] Candler, W., and Townsley, R., A linear bilevel programming problems, *Computers and Operations Research* 9 (1) (1982), 59-76.
- [8] De Jong.K.A An analysis of the behavior of a class of genetic adaptive systems. *Dissertation Abstracts International*. 86(1975): 5140B
- [9] Fortuni-Amat, J., and McCarl, B., A representation and economic interpretation of a two-level programming problem, *Journal of Operational Research Society* 32 (9) (1981), 783-792.
- [10] Goldberg D.E. Genetic algorithm in search, optimization and machine learning. Addison Wesley publishing company, 1989.
- [11] Holland J.H. Adaptation in natural and artificial systems. University of Michigan Press, MI, 1975.
- [12] Hecheng Li and Yuping wang. A genetic algorithm based on optimality conditions for nonlinear bi-level programming problems. *J. Appl. Math. & Informatics*. 28(3)(2010):597-610
- [13] Herminia I. Calvete, Carmen Galé, Theory and Methodology- The bilevel linear/linear fractional programming problem, *European Journal of Operational Research*, 114(1), (1999):188-197
- [14] Ketabchi S. and H. S Moosaei H.S, etl. Optimal Correction of Infeasible System in Linear Equality via Genetic Algorithm., *App. and Appl. Math: Int. Jr. (AAM)*, 5(2)(2010):488 – 494.
- [15] Lai YJ (1996) Hierarchical optimization: a satisfactory solution. *Fuzzy Sets and Systems* 77:321–335
- [16] Mishra S , Verma A , An Algorithm Based on the Fitness Function for Solving Bi-Level Linear Fractional Programming Problems. *International Journal of Modern Mathematical Sciences*, 2015.
- [17] Michalewicz. Z. Genetic algorithms + data structure = evolutionary programs. Springer, New York, 1992.
- [18] Shih, H. S., and Lee, E. S., Compensatory fuzzy multiple level decision making, *Fuzzy Sets and Systems* 114 (1) (2000), 71–87.
- [19] Sakawa M., Azad M.A.K. Interactive Fuzzy Programming For Multi-level Non linear Integer Programming Problems through Genetic Algorithms. *Scientiae Mathematicae Japonicae online*, e-2006, 223-236.
- [20] Wang G., Wan Z and Wang X. Solving Method for a Class of Bi-level Linear Programming based on Genetic Algorithms. *optimization-online.org*. DB_FILE/2003/03/617.
- [21] Wanga G. and Wan Zhongping. etl. Genetic algorithm based on simplex method for solving linear-quadratic bilevel programming problem. *Comp. and Math. with App.*, 56(2008): 2550–2555.
- [22] H. -J. Zimmermann , Fuzzy programming and linear programming with several objective functions, *Fuzzy Sets and Systems*, vol. 1, pp. 45-55, 1978.