



Solving Method for a Class of Bi-Level Quadratic Fractional Programming Problems based on Genetic Algorithm

Savita Mishra

Department of Mathematics, L.B.S.M. College, Jamshedpur, Kolhan University, Jharkhand, INDIA.

Arun Bihari Verma

Department of Mathematics, Mahila College, Chaibasa, Kolhan University, Jharkhand, INDIA.

Abstract: We present genetic algorithm (GA) for solving bi-level quadratic fractional programming problem (BLQFPP) 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. 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. Finally, the feasibility and effectiveness of the proposed approach is demonstrated by the numerical example.

Keywords: Bi-Level Linear Quadratic Programming Problems, Genetic Algorithm, Fitness Function, Satisfactory Solution.

I. Introduction

A bi-level programming problem (BLPP) consists of two levels, namely, the first level and the second level. The first level decision maker (DM) is called the centre. The second level DM called follower, executes its policies after the decision of higher level DM called leader (centre) and then the leader optimizes its objective independently but may be affected by the reaction of the follower i.e. BLPP is a sequence of two optimization problems in which the constraints region of one is determined by the solution of second.

There are many methods to solve BLPPs. The formulation and different version of BLPP are given by Bard [8, 9], Candler [24], Bard and Falk [7] and Bialas and Karwan [23]. Bialas and Karwan [23] are the pioneers for linear BLPP who presented vertex enumeration method, called Kth- best solution. These were solved by simplex method. To solve the non-linear problem that arises due to the K-T conditions, Bialas and Karwan [23] proposed a parametric complementary pivot (PCP) algorithm which interactively solves a slight perturbation of the system. Bard and Falk [7] proposed the grid search algorithm. Based on Bard and Falk's algorithm, Unlu [4] proposed a technique of bi-criteria programming. In the frame work of fuzzy decision of Bellman and Zadeh [16] presented a fuzzy programming approach for solving multi-objective linear fractional programming problem by the combined use of the bi-section method and the phase one of simplex method of linear programming. Mishra and Ghosh [21] presented fuzzy programming approach to solve bi-level linear fractional programming problems. Again, Mishra and Ghosh [19] proposed interactive fuzzy programming approach to solve bi-level quadratic fractional programming problems. Also, Mishra [20] presented weighting method for bi-level linear fractional programming problems.

Fractional programming (FP) [1,2,12,17,19,20,21] is generally used for modelling real life problem which has one or more than one objective(s) as a ratio of two functions such as profit/loss, inventory/sales, actual cost/standard cost, output/employee etc. Fractional programs arise in various contexts such as, in investment problems, the firm wants to select a number of projects on which money is to be invested so that the ratio of the profits to the capital invested is maximum subject to the total capital available and other economic requirements which may be assumed to be linear or non-linear

Bi-level quadratic fractional programming problems (BLQFPPs) are studied by a few [19,22]. 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. GAs was first introduced by Holland [11] and since then it has been applied to many OR field such as: De Jong [13]; Baker [10]; Goldberg [3]; Wang [5]; Wanga Guangmin et al. [6]; Narang and Arora [18]; Hecheng and Wang [14] etc.

In this paper we consider the solution of a BLQFPP 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. Perhaps the most creative task in making a decision is to choose the factors that are important for that decision. The efficiency of the techniques depends to a great extent on the nature of the mathematical formulation of the problem. Genetic Algorithm, which is a population – based search technique Goldberg [3] 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. This paper demonstrates the merit of this technique in deciding optimal solution of bi-level linear fractional decision-making problem taking into consideration the various constraints and complexities representing the real situation.

II. Bi-level Quadratic Fractional Programming Problems

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. These divisions are independent. A multi-level programming problem (MLPP) can be defined as a p -person, nonzero sum game with perfect information in which each player moves sequentially from top to bottom. This problem is a nested hierarchical structure. When $p = 2$, we call the system a bi-level programming problem.

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

$$\max_{x_1} z_1(x_1, x_2) \quad \text{where } x_2 \text{ solves} \quad (P1)$$

$$\max_{x_2} z_2(x_1, x_2) \quad (P2)$$

$$\text{subject to } Ax_1 + Bx_2 \leq r$$

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

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

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

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:-

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

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, $z_1(x_1, x_2)$ 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.

In the bi-level quadratic fractional programming problem (P1)-(P2), $z_1(x, y)$ and $z_2(x, y)$ respectively represent objective functions of DM1 and DM2, and x_1 and x_2 represent decision variables under the control of DM1 and DM2 respectively.

Once x_1 is fixed, the term containing x_1 in the objective function of the lower-level problem is a constant. So the objective function of the lower-level problem is simply denoted as: $\bar{z}_2 = \phi(x_2)$.

Let $S = \{(x_1, x_2) : Ax_1 + Bx_2 \leq r\}$ denote the constraint region of BLQFPP. Here, we assume S is nonempty and bounded. Let $Q(x_1) = \{x_2 : Bx_2 \leq r - Ax_1, x_2 \geq 0\}$ nonempty and bounded. Let $Y(x_1)$ denote the optimal solution set of the problem $\max_{x_2 \in Q(x_1)} z_2 = \phi(x_2)$. We assume the element of the set $Y(x_1)$ exists and is unique, then the

inducible region is: $\psi_{z_2}(S) = \{(x_1, x_2) : (x_1, x_2) \in S, x_2 = Y(x_1)\}$

Hence, the problem (P1) can be changed into:

$$\text{Max } z_1(x) = \frac{x_1' Q_{11} x_1 + x_2' Q_{12} x_2 + c_{11} x_1 + c_{12} x_2 + c_{13}}{x_1' R_{11} x_1 + x_2' R_{12} x_2 + d_{11} x_1 + d_{12} x_2 + d_{13}} \quad \text{P(3)}$$

subject to $Ax_1 + Bx_2 \leq r, \quad x_1 \geq 0, \quad x_2 \geq 0.$

$$x_2 = Y(x_1)$$

Then, if the point (x_1, x_2) is the solution of the following problem

$$\text{Max } z_1(x) = \frac{x_1' Q_{11} x_1 + x_2' Q_{12} x_2 + c_{11} x_1 + c_{12} x_2 + c_{13}}{x_1' R_{11} x_1 + x_2' R_{12} x_2 + d_{11} x_1 + d_{12} x_2 + d_{13}} \quad \text{P(3)}$$

subject to $Ax_1 + Bx_2 \leq r, \quad x_1 \geq 0, \quad x_2 \geq 0.$

and $x_2 = Y(x_1)$, then (x_1, x_2) is the solution of the following problem (P1).

The paper will discuss the numerical method of BLQFPP under the definition.

Definition 1: The point (x_1, x_2) is feasible if $(x_1, x_2) \in \psi_{z_1}(S)$.

Definition 2: The feasible point $(x_1^*, x_2^*) \in \psi_{z_2}(S)$ is the optimal solution of the BLQFPP if $z_1(x_1^*, x_2^*) \geq z_1(x_1, x_2)$ for each point $(x_1, x_2) \in \psi_{z_1}(S)$.

III. Design of the GA for Bi-level Quadratic Fractional Programming Problem (BLQFPP)

Genetic algorithm (GA) is search algorithms based on the mechanism of natural selection and natural genetics. GA is a stochastic heuristic optimization search technique designed following the natural selection process in biological evolution to arrive at optimal or near optimal solutions to complex decision problems. 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. 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. It is not easy to know the upper-level objective function of BLPP has no explicit formulation, since it is compounded by the lower-level solution function which has no explicit formulation. Thus, it is hard to express the definition of the derivation of the function in common sense. And it is difficult to discuss the conditions and the algorithms of the optimal solution with the definition. We concerned the GA [5] 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 BLQFPP by GA.

3.1. Coding and Constraints

The first step in designing a genetic algorithm for a particular problem is to devise a suitable representation scheme. 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 $Ax_1 + Bx_2 \leq r$ 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.

3.2. Design of the Fitness Function

During each successive epoch, a proportion of existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process. The selection strategy is one of the most important factors in the genetic search. 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 $(x_1, x_2) \in S$:

Definition 3: Let $\Theta \in [0,1]$ denotes the feasible degree of satisfying the feasible region, and describe it by the following function:

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

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

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

$$eval(v_k) = (z_1(x_1, x_2) - z_{1\min}) * \Theta$$

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

3.3. Genetic Operators

The crossover operator is one of the important genetic operators. 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 crossover, then the final off-springs are:

$$v_1' = \alpha * v_1 + (1 - \alpha) * v_2$$

$$v_2' = \alpha * v_2 + (1 - \alpha) * v_1$$

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

is, $v_1, v_2 \in S$).

The mutation operator is another important genetic operator in GA. Mutation operator performs changes in a single individual. It randomly searches in the neighbourhood 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]: 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

$$v_k' = (v_{k1}', v_{k2}', \dots, v_{km}')^T$$

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. We adopt roulette wheel selection since it is the simplest selection.

3.4. Termination Criteria

The judgment of the termination is used to decide when to stop computing and return the result. We adopt the maximal iteration number [5,6] as the judgment of the termination. The algorithm process using the GA 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.

IV. The Numerical Result

In this section we present numerical example to demonstrate the solution procedures by proposed GA to solve BLQFPP. The following example considered by Mishra [22] is again used to demonstrate the solution procedures and clarify the effectiveness of the proposed approach:

Consider the following bi-level quadratic fractional programming problem (BLQFPP):

$$\text{Max}_{x_1} z_1 = \frac{10x_1^2 + 15x_2^2 + 5}{x_1^2 + 2x_2^2 + 1}$$

$$\text{Max}_{x_2} z_2 = \frac{25x_1^2 + 9x_2^2}{2x_1^2 + x_2^2 + 1}$$

subject to

$$4x_1 - 5x_2 \leq 15;$$

$$3x_1 - x_2 \leq 21;$$

$$2x_1 + x_2 \leq 27;$$

$$3x_1 + 4x_2 \leq 45;$$

$$x_1 + 3x_2 \leq 30;$$

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

The compare of the results through 500 generations by the algorithms in the paper and the results in the references is shown in Table 1.

Table 1: Solution by proposed GA approach

Parameters			Result in the paper				Result in the references			
M	Pc	Pm	X ₁	X ₂	Z ₁	Z ₂	X ₁	X ₂	Z ₁	Z ₂
50	0.7	0.15	5.34	1.27	9.6	12.20	5.34	1.27	9.6	12.20

M denotes the population scale; P_c denotes crossover probability; P_m denotes mutation probability; Z_1 and Z_2 are the objective function value of the upper-level and lower-level programming problem, respectively.

V. Conclusion

In this paper we consider the solution of a bi-level quadratic fractional programming problems 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. The proposed approach work with an entire population of trial solution and the population is used

to create linking paths between its members and to re-launch the search along these paths. The numerical result shows the proposed algorithm is feasible and efficient, can find global optimal solutions with less computational burden.

References

- [1] A. Charnes and W.W.Cooper. Programming with linear fractional functions. *Naval Res. Logist. Quart.*9 (1962):181-186.
- [2] B. Martos, Hyperbolic Programming. *Naval Res. Logist. Quart.* 11(1964):135- 155.
- [3] D.E Goldberg. *Genetic algorithm in search, optimization and machine learning*. Addison Wesley publishing company, 1989.
- [4] G. A. Unlu. Linear Bi-level Programming Algorithm Based on Bi-criteria Programming. *Comp and O R.* 14(1987):173-179.
- [5] G. Wang, Z. Wan and X Wang. Solving Method for a Class of Bi-level Linear Programming based on Genetic Algorithms. *optimization-online.org*. DB_FILE/2003/03/617.
- [6] Guangmin Wanga and Zhongping Wan etl. Genetic algorithm based on simplex method for solving linear-quadratic bilevel programming problem. *Comp. and Math. with App.*, 56(2008): 2550–2555.
- [7] J.F. Bard and J.E. Falk. An Explicit Solution to the Multilevel Programming Problem, *Comp. and Oper. Res.* 9(1)(1982):77-100.
- [8] J.F. Bard. An Algorithm for Solving the Bi-level Programming Problem. *Math and O. R.* 8(2)(1983); 260-270.
- [9] J.F. Bard. Optimality Conditions for the Bi-level Programming Problem. *Naval Res. Logist. Quart.*, 31(1984):13-26.
- [10] J.E Baker, Adaptive selection method for genetic algorithms and their applications. *Proceedings of the International Conference on genetic Algorithms and their applications*. (1985): 101-111.
- [11] J.H Holland. Adaptation in natural and artificial systems. University of Michigan Press, MI, 1975.
- [12] J.R Isbell and W.H Marlow. Attrition Games. *Naval Res. Logist. Quart.* 3(1956):71-93.
- [13] K.A De Jong. An analysis of the behavior of a class of genetic adaptive systems. *Dissertation Abstracts International*. 86(1975): 5140B.
- [14] Li Hecheng and Wang Yuping. A genetic algorithm based on optimality conditions for nonlinear bi-level programming problems. *J. Appl. Math. & Informatics*. 28(3)(2010):597-610.
- [15] P.C Gilmore and R.E Gomory. A Linear Programming Approach to the Cutting Stock Problem-Part2. *Operational Research*. 11(1963):863-867.
- [16] R.E Bellman and L.A Zadeh. Decision Making in Fuzzy Environment. *Management Science*, 17(1970):141-164.
- [17] R. Jagannathan. On Some Properties of Programming Problems in Parametric Form Pertaining to Fractional Programming. *Management Science*, 12(1966): 609-615.
- [18] Ritu Narang and S.R Arora. An enumerative algorithm for non-linear multi-level integer programming problem. *Yugoslav J. of Oper. Res.* 19(2)(2009): 263-279.
- [19] Savita Mishra and Ajit Ghosh. Interactive Fuzzy Programming Approach to Bi-level Quadratic Fractional Programming Problems. *Annals of Oper. Res.*143(2006): 249-261.
- [20] Savita Mishra. Weighting Method for Bi-level Linear Fractional Programming Problems. *European Journal of Oper. Res.* 183(2007): 296-302.
- [21] Savita Mishra and Ajit Ghosh. Fuzzy Programming Approach to Bi-level Linear Fractional Programming Problems. *Int. Jr. of Comp. Math. Sc. and Appl.* 1(2-4)(2007):181-189.
- [22] Savita Mishra, “ Goal Programming Approach to Bi-Level Quadratic Fractional Programming Problems, American International Journal of Research in Formal, Applied & Natural Sciences, (AIJRFANS), 13(1) (2016): pp.45-54.
- [23] W.F Bialas and M.H Karwan. Two Level Linear Programming. *Management Science*, 30(1984): 1004-1020.
- [24] W. Candler and M.H Karwan. A Linear Two Level Programming Problem. *Comp. and Oper. Res.* 9(1)(1982): 59-76.