American International Journal of Research in Science, Technology, Engineering & Mathematics

Available online at http://www.iasir.net

ISSN (Print): 2328-3491, ISSN (Online): 2328-3580, ISSN (CD-ROM): 2328-3629

AIJRSTEM is a refereed, indexed, peer-reviewed, multidisciplinary and open access journal published by International Association of Scientific Innovation and Research (IASIR), USA

(An Association Unifying the Sciences, Engineering, and Applied Research)

Solution of Multi-Level Linear Fractional Programming Problems by Interactive Fuzzy Programming through Genetic Algorithms with Double Strings

Savita Mishra
Assistant Professor, Department of Mathematics,
LBSM College Jamshedpur, Kolhan University, Chaibasa, Jharkhand, India.

Ranjeet Kumar Singh
Professor & Pro Vice-Chancellor
Kolhan University,Chaibasa, Jharkhand, India.

Abstract: *Hierarchical optimization or multi-level programming techniques are extension of Stackelberg games for solving decentralized planning problem with multiple DMs in a hierarchical organization. This paper presents an interactive fuzzy programming through genetic algorithm with double strings using continuous relaxation based on reference solution updating to solve multi-level linear fractional programming problems.Assuming the cooperative relationship among the DMs at all levels the proposed approach gives satisfactory solution for all DMs.Furthermore, the feasibility and efficiency of the proposed approach is shown by applying it to illustrative numerical example.*

Keywords: *Multi-level Programming Problems, Fractional Programming Problems, Fuzzy Membership Function, Genetic Algorithm.*

I. Introduction

Since fuzzy logic has proven to be a very useful tool for representing human knowledge by means of mathematical expressions, the optimization of the involved parameters has been one of the most investigated problems in the theory of fuzzy expert systems. Genetic algorithms are optimization methods which are based on the mechanisms of natural evolution, such as selection, mutation, or sexual reproduction. Genetic algorithms were introduced approximately 25 years ago and turned out to be a very promising approach to the solution of many problems in artificial intelligence. During the last years the combination of fuzzy logic and GAs [1] has come into fashion. Nevertheless, or better, for exactly that reason it is necessary to investigate this combination critically and to expose the advantages and weaknesses objectively. This paper is intended to provide a profound introduction to both fuzzy logic and genetic algorithms and to explore the possibilities to combine the two paradigms to solve multi-level linear fractional programming problems.In this context, the constraint region is implicitly determined by a series of optimization problems which must be solved in a predetermined sequence. Alternatively, the problem can be viewed as an n-person, nonzero-sum game with perfect information[1] where the order of play is specified at the outset and the players' strategy sets are no longer assumed to be disjoint. A basic assumption is that each player (Decision Maker) knows the entire structure of the game in this form and that all players are governed in their behavior by an inflexible desire to maximize their expected payoff.

In contrast to it is usually assumed that the strategy sets for all are independent or disjoint and that all players move simultaneously. A further consideration is one of cooperation among the players. While this may work out to everyone's advantage, instances arise where the rules of the game or the realities of the situation strictly forbid any type of agreements (e.g. anti-trust laws or the inability to communicate).Two cases must therefore be distinguished:

(1) The noncooperative case, in which any type of collision, such as correlated strategies and side payments, is prohibited.
(2) The cooperative case, in which all such agreements are permitted.

The noncooperative case most accurately reflects the assumption implicit in the multi-level programming problem.In the reminder of this section, we will highlight the structure and properties of multi-level Programming problems.

In this paper, for MLLFPPs, focusing on the case of cooperative relation among all DMs, we present a new interactive fuzzy programming method through genetic algorithms with double strings using continuous relaxation based on reference solution updating (GADSCRRSU) in order to derive a satisfactory solution for the DMs.

II. Multi-level linear fractional programming problems (MLLFPP)

Multi- level linear fractional programming problem (MLLFPP) is identified as a mathematical programming that solves decentralized planning problems with multiple executors in a multi-level or hierarchical organization and each (or at least one) decision maker(DM) has fractional objective functions. Let the MLLFPP in a decision situation be such that the DM of each level takes overall satisfactory balance and tries to maximize his/her own objective function paying serious attention to the preferences of the others. MLLFPP can be defined as a p-person, nonzero sum game with perfect information in which each player moves sequentially from top to bottom and having linear fractional objective functions. Consider a p-level programming problem of maximization-type objective function at each level.

Mathematically, MLLFPP can be formulated as follows:

$$\max_{x_1} z_1(x) = \max_{x_1} imize \qquad z_1(x_1, x_2, ..., x_p) = \max_{x_1} \frac{a_1 x + r_1}{b_1 x + s_1}$$

$$\max_{x_2} z_1(x) = \max_{x_2} imize \qquad z_2(x_1, x_2, ..., x_p) = \max_{x_2} \frac{a_2 x + r_2}{b_2 x + s_2} \qquad (1)$$

$$\vdots$$

$$\max_{x_p} z_p(x) = \max_{x_p} imize \qquad z_p(x_1, x_2, ..., x_p) = \max_{x_p} \frac{a_p x + r_p}{b_p x + s_p}$$

Subject to

$$x \in S = \{A_1 x_1 + A_2 x_2 + ... + A_p x_p \ (\geq, =, \leq) B, x \geq 0\}.$$

Where,

$$z_i(x_1, x_{2,.......,} x_p) = \frac{a_i x + r_i}{b_i x + s_i} = \frac{a_{i1} x_1 + a_{i2} x_2 + ... + a_{ip} x_p + r_i}{b_{i1} x_1 + b_{i2} x_2 + ... + b_{ip} x_p + s_i}, \ (i = 1, 2, ..., \ p)$$

Here, S is the non empty convex constraint set; M is the total number of constraints.

A_i is the $M \times N_i$ matrix $(i = 1, 2, ..., \ p)$;

B is M component column vector;

$a_{i1}, b_{i1} \in R^{N_1}$, $a_{i2}, b_{i2} \in R^{N_2}$,……….. $a_{ip}, b_{ip} \in R^{N_p}$; $r_i, s_i, (i = 1, 2, ..., \ p)$ are scalars.

We take, $x = x_1 \cup x_2 \cup ...... \cup x_p$; $N_1 + N_2 + ... + N_p = N$.

We also assume that $b_{i1} x_1 + b_{i2} x_2 + ... + b_{ip} x_p + s_i > o$, $(i = 1, 2, ..., \ p)$ for all $x \in S$.

This system has interacting decision making units within a hierarchical structure where each level performs its policies after knowing completely the decisions of superior levels. For example, consider a project selection problem in an administrative office at the upper level and several autonomous divisions of a company. In this case, the situation that all the DMs can cooperate with each other seems natural rather than one that all the DMs do not have motivation to cooperate mutually. Under the hypothesis of cooperation among all DMs, Sakawa et al. [12, 13, 14] proposed interactive fuzzy programming for multi-level linear programming problems in order to derive satisfactory solutions for the DMs through interactions with the DM at the upper level by introducing fuzzy goals to consider the imprecise nature of DMs' judgements for objective functions.

III. Interactive Fuzzy Programming for Multi-Level Linear Fractional Programming Problems (MLLFPPs) through Genetic Algorithms

In this section, we describe a new interactive fuzzy programming method through genetic algorithms based on literatures by Sakawa et al. is summarized as follows. It is natural that the DMs have fuzzy goal of their objective functions when they take fuzziness of human judgments into consideration**.** For each of the objective functions

$Z_i(x), i = 1,2.......,p$, in (1), it seems natural to introduce such fuzzy goals for objective functions as " $Z_i(x)$ should be subjectively less than or equal to a certain value".
First, we solve problems to obtain the individual minimum

$$Z_i^{\min} = \min_{x \in X} Z_i(x), (i = 1,2,.....p) \quad (2)$$

and the individual maximum

$$Z_i^{\max} = \max_{x \in X} Z_i(x), (i = 1,2,.........p) \quad (3)$$

of each of the objective functions which are referred to when the DMs elicit membership functions prescribing the fuzzy goals for the objective functions $Z_i(x), (i = 1,2,....p)$, Since these problems are single-objective linear fractional programming problems and it is difficult to obtain optimal solutions to them, we use genetic algorithms with double strings using continuous relaxation based on reference solution updating (GADSCRRSU) which is an extension of genetic algorithms with double strings based on reference solution updating (GADSRSU).
The DMs determine the membership functions $\mu_i(Z_i(x)) = 0, (i = 1,2,....p)$ which are strictly monotone decreasing for $Z_i(x)$, consulting the variation ratio of degree of satisfaction in the interval between the individual minimum of problem (2) and the individual maximum of problem (3). The domain of the membership functions is in the interval $[Z_i^{\min}, Z_i^{\max}], (i = 1,2,........p)$ and the DM specifies the value Z_i^0 of the objective function for which the degree of satisfaction is 0 and the value Z_i^1 of the objective function for which the degree of satisfaction is 1. For the value undesired (larger) than Z_i^0 ,it is defined that $\mu_i(Z_i(x)) = 0, (i = 1,2,......,p)$ and for the value desired (smaller) than Z_i^1, it is defined that $\mu_i(Z_i(x)) = 1, (i = 1,2,......p)$. Here a linear membership function [1] is used, which characterizes the fuzzy goal of the DM at each level.

The corresponding linear membership function $\mu_i(Z_i(x))$ is defined as:

$$\mu_{i_i}(Z_i(x)) = \begin{cases} 1 & , \quad Z_i(x) < Z_i^1 \\ \dfrac{Z_i - Z_i^0(x)}{Z_i^1 - Z_i^0} & , \quad Z_i^1 \le Z_i(x) < Z_i^0 \\ 0 & , \quad Z_i(X) \ge Z_i^0 \end{cases} \quad (4)$$

Where, Z_i^0 and Z_i^1 denote the value of objective function $Z_i(x)$ such that the degrees of the membership function are 0 and 1, respectively, and it is assumed that the DMs subjectively assess Z_i^0 and Z_i^1 .Zimmermann [19] proposed a method for determining the parameters Z_i^0 and Z_i^1 of the eliciting membership function in the following way. That is, using the individual minimum, they are defined as:

$$Z_i^1 = Z_i^{\min} = Z_i(x^{i0}) = \min\{Z_i(x)\} \quad (5)$$

Together with

$$Z_i^0 = \max(Z_i(x^{i0}),.........Z_i(x^{(i-1)0}, Z_i(x^{(i+1)0},.......Z_i(x^{i0})) \quad (6)$$

Having elicited membership functions

$\mu_i(Z_i(x))$ for $Z_i(x)$, i=1,2,………p by the DM at each level, then the original MLLFPP defined by :

$$\begin{aligned} &\max_{DM1} imize \quad \mu_1(Z_1(x)) \\ &\max_{DM2} imize \quad \mu_2(Z_2(x)) \\ &\dots \ \dots \ \dots \ \dots \\ &\dots \ \dots \ \dots \ \dots \ \dots \end{aligned} \quad (7)$$

$$\max_{DMi} imize \quad \mu_i(Z_i(x))$$

$$\text{Subject to} \quad g_j(x) \le 0, \quad j = 1,2\ldots\ldots m$$

$$x_{ik} \in (0,1,\ldots\ldots\ldots,v_{ik}), \; i = 1,2,\ldots,p., \quad k = 1,\ldots\ldots n_l.$$

Since (7) is a multi-level membership maximization problem, in general, a complete optimal solution that simultaneously maximizes all the DMs' degree of satisfaction of their objective functions does not always exist when the objective functions conflict with each other. Thus, a satisfactory solution is expected to be obtained from among M-Pareto optimal solution set which is defined for multi-objective programming problems [10, 11]. For deriving an overall satisfactory solution to the formulated problem (7), first the maximizing decision of the fuzzy decision proposed by Bellman and Zadeh [3] is found. Namely, the following problem is solved for obtaining a solution which maximizes the smallest degree of satisfaction among the *p* DMs:

$$\max imize \min_{i=1,2,\ldots\ldots p} \quad \mu_i(Z_i(x))$$

$$\text{Subject to} \quad g_j(x) \le 0, \quad j = 1,2\ldots\ldots m \tag{8}$$

$$x_{ik} \in (0,1,\ldots\ldots\ldots,v_{ik}), \; i = 1,2,\ldots,p., \quad k = 1,2,3,\ldots\ldots n_i.$$

In order to guarantee the M-Pareto optimality of the obtained solution, the following augmented maximin problem (9) instead of the maximin problem (8) is solved.

$$\max_{x\in X} imize \min_{i=1,2,\ldots\ldots p} \left\{ \mu_i(Z_i(x) + \rho \sum_{q=1}^{p} \mu_q(Z_q(x)) \right\} \tag{9}$$

The term augmented is adopted here because the term $\rho \sum_{q=1}^{p} \mu_q(Z_q(x))$ is added to the standard maximin problem (8), where ρ is a sufficiently small positive number. By solving problem (9), we can obtain a solution which maximizes the smallest degree of satisfaction among all of the *p* DMs. This problem can also be solved by GADSCRRSU. Let us denote an optimal solution to the problem (9) by x^*. Then, we define the satisfactory degree of the p DMs under the constraints as:

$$\lambda = \min\left\{ \mu_1(Z_1(x^*)\ldots\ldots\ldots\mu_p(Z_p(x^*)) \right\} \tag{10}$$

and the ratio of satisfactory degrees between adjacent two levels as:

$$\Delta_i = \frac{\mu_{i+1}(z_{i+1}(x))}{\mu_i(z_i(x))}, \quad i = 1,2,\ldots.,p-1. \tag{11}$$

If the optimal solution x^* to the problem (9) cannot satisfy the DMs, problems concerning a part of the DMs are solved one after another from the last two levels in order to coordinate the satisfactory degrees of the DMs and finally obtain a satisfactory solution of all the DMs.

At an interaction l, let $\mu_i(z_i^l)$ and λ^l denote a satisfactory degree of DMs, $i = 1,2,\ldots,p.,$ and a satisfactory degree of all the levels, respectively, and let $\Delta_i^l = \frac{\mu_{i+1}(z_{i+1}^l)}{\mu_i(z_i^l)}$ denote a ratio of satisfactory degrees of the ith and the $(i+1)th$ levels.

Let x^l denote a solution at the interaction l. For all $i = 1,2,\ldots,p.,$ DMi is proposed a solution by $DM(i+1)$. Then the DMs at all the levels except for the pth level obtain the satisfactory solutions and the interactive process terminates if the following two conditions are satisfied.

IV. Termination conditions of the interactive process

Condition 1: For all $i=1,2,\ldots,i-1,$ DMi 's satisfactory degree is larger than or equal to the minimal satisfactory level $\hat{\delta}_i$ specified by DMi, i.e., $\mu_i(z_i^l)\geq\hat{\delta}_i$, $i=1,2,\ldots,p-1.,$

Condition 2: For all $i=1,2,\ldots,p-1.,$ the ratio of Δ_i^l of satisfactory degrees lies in the interval between the lower and the upper bounds specified by DMi ,i.e., $\Delta_i^l\in[\Delta_i^{\min},\Delta_i^{\max}]$.

Condition 1 is DMi 's required condition for solutions proposed by $DM(i+1)$, and condition 2 is provided in order to keep overall satisfactory balance among all the levels.

Unless the conditions are satisfied simultaneously, some of DMi, $i=1,2,\ldots,p-1.,$ needs to update his minimal satisfactory level $\hat{\delta}_i$. Suppose that the DMs from at the (u + 1)th level to at the (p− 1)th level, i.e., DM(u + 1), DM(u+ 2), . . . , DM(p − 1) are satisfied with the proposed solution but DMu is not satisfied with it. Then DMu, DM (u+ 1), . . . , and DM(p−1) need to update their minimal satisfactory levels $\hat{\delta}_i$, $i=u,u+1,\ldots,p-1.$ For any two levels adjacent to each other, giving the DM at an upper level serious consideration, the DM at a lower level should update his minimal satisfactory level.

Procedure for updating minimal satisfactory level $\hat{\delta}_i$:

Case 1: If the condition 1 with respect to a DM at a level is not satisfied, then the DM decreases the minimal satisfactory level $\hat{\delta}_i$.

Case 2: If the ratio Δ_i^l exceeds its upper bound, then DMi increases the minimal satisfactory level $\hat{\delta}_i$. Conversely, if Δ_i^l is less than its lower bound, then DMi decreases the minimal satisfactory level $\hat{\delta}_i$.

Case 3: Although conditions 1 and 2 are satisfied, if a DM at a level is not satisfied with the obtained solution and judges that it is desirable to increase the satisfactory degree of the DM at the expense of the satisfactory degree of the DM at his lower level, then the first DM increases the minimal satisfactory level $\hat{\delta}_i$ and vice versa.

Let $\hat{\delta}_i{}'$, $i=1,2,\ldots,p-1,$ denote the updated minimal satisfactory level, then the following problem (12) is solved which maximizes the smallest degree of satisfaction among the DMs from 1 to u − 1 and pth levels.

$$\underset{x\in X}{\max imize}\ \underset{i=1,2,\ldots\ldots u,p}{mim}\ \left\{\mu_i(Z_i(x)+\rho\sum_{q=1,2,\ldots.u,p}^{p}\mu_q(Z_q(x))\right\}$$

$$\text{Subject to}\quad g_j(x)\leq 0,\quad j=1,2\ldots\ldots\ldots m \tag{12}$$

$$\mu_i(z_i^k)\geq\hat{\delta}_i{}',\ i=u,\ldots\ldots,p-1.,$$

$$x_{ik}\in\left(0,1,\ldots\ldots\ldots\ldots,v_{ik}\right),\ i=1,2,\ldots,p.,\quad k=1,2,3,\ldots\ldots n_i.$$

It should be noted that GADSCRRSU is applicable to (12).

V. Algorithm of the Interactive Fuzzy Programming through GADSCRRSU

We are now ready to present an interactive algorithm for deriving an overall satisfactory solution to multi-level linear fractional programming problems (1) through genetic algorithms with double strings using continuous relaxation based on reference solution updating (GADSCRRSU), which is summarized as follows:

Step 1: Solve (5) through GADSCRRSU for individual minimum and by using equation (6) calculate z_i^0, for each objective function of all the DMs and ask the DMs to identify their membership functions $\mu_i(z)$, $i=1,2,\ldots,p.,$ of the fuzzy goals for their own objective functions. Also ask the DMi, $i=1,2,\ldots,p-1.,$ to specify subjectively the lower and the upper bounds of the ratio of satisfactory degrees Δ_i.

Step 2: Set interaction $l := 1$ and solve the problem (9) through GADSCRSSU, in which the smallest degree among all the DMs is maximized. The solution is proposed to all the DMs except the pth level.

Step 3: If the solution proposed to DMs at all the levels except for the pth level satisfies the termination conditions, they conclude the solution as a satisfactory one and the algorithm stops. Otherwise, let $l := l + 1$ and go to step 4.

Step 4: If the DMs from at the (u + 1)th level to at the (p − 1)th level, i.e., DM(u + 1), DM(u + 2), . . . , and DM(p − 1) is satisfied with the proposed solution but DMu is not satisfied with it, $DMi\ , i = u,\dots,p-1.$, update the minimal satisfactory levels $\hat{\delta}_i$, $i = u,\dots,p-1.$, in according to the procedure of updating minimal satisfactory level.

Step 5: Solve (12) through GADSCRRSU, in which smallest degree of satisfaction among the DMs from 1 to p − 1 and pth levels is maximized and propose the solution to the DMs from 1 to u− 1 levels. Return to step 3.

VI. Genetic Algorithms with Double Strings using Continuous Relaxation based on Reference Solution Updating (GADSCRRSU)

In this section, we mention GADSCRRSU proposed as a general solution method for nonlinear integer programming problems defined as:

$$\min imize \quad Z(x)$$

$$\text{Subject to} \quad g_j(x) \le 0, \quad j = 1,2.........m \tag{13}$$

$$x_k \in (0,1,...........,v_k), \quad k = 1,2,3,......n_i .$$

In (13), $\boldsymbol{x}$ is an n dimensional integer decision variable vector, $Z(x)$, $g_j(x) \le 0$, $j = 1,2.........m$ are linear fractional functions and v_k , $k = 1,2,3,......n$ is the upper bound of each decision variable.

A. Individual Representation: The individual representation [10, 11] by double strings shown in table 2 is adopted in GADSCRRSU. In the figure, each of $s(k)$, $k = 1,2,3,......n$ is

Indices	$S(1)$	$S(2)$	$\cdots$	$S(k)$	$\cdots$	$S(n)$
Values	$y_s(1)$	$y_s(2)$		$y_s(k)$		$y_s(n)$

Table 2: Double Strings Representation

the index of an element in a solution vector and each of $y_s(k) \in \{0,1,.....,v_k\}$ $k = 1,2,3,......n_i$. is the value of the element, respectively.

B. Decoding Algorithm: Let N be the total number of population (*pop size*). The individuals $\boldsymbol{s}$ with the dimensions n are generated randomly. Unfortunately, however, the direct mapping of the individual can generate infeasible solutions [10, 11]. To eliminate such solutions, a decoding algorithm of double strings for nonlinear integer programming problems (13) using a reference solution x^0, which is a feasible solution and used as the origin of decoding, is constructed as follows.

Decoding algorithm using reference solution:

In the algorithm, it is assumed that a feasible solution x^0 is obtained in advance. Let n, and N be the number of variables and number of individuals in the population, respectively.

Step 1: *If the index of an individual to be decoded is in* $\{1,2,.......[N/2]\}$, *go to stop2. Otherwise, go to step 8.*

Step 2: *Let* $k := 1, \quad x := \{0,....0\}, \quad i := 1,$

Step 3: *Let* $x_s(j) := y_s(j).$

Step 4: *If* $g_i(x) \le 0, \quad j = 1,....,m$

let $i := k, \quad k := k+1,$ *and go to step 5. Otherwise, let* $k := k+1,$ *and go to step 5.*

Step 5: *If* $k \le n$ *, go to step 3. Otherwise, go to step 6.*

Step 6: *If* $i > 0$ *, go to step 7. Otherwise, go to step 8.*

Step 7: *By substituting* $x_s(k) := y_s(k)$, $1 \le k \le i$ *and* $x_s(k) := 0$, $1 \le k \le n$, *we obtain a feasible solution* $\boldsymbol{x}$ *corresponding to the individual* $\boldsymbol{s}$ *and stop.*

Step 8: *Let* $k := 1, \quad x = x^0$

Step 9: *Let* $x_s(k) := y_s(k)$. *If* $y_s(k) := x_s^0(k)$ *let* $k := k+1,$ *let* $k := k+1,$ *, and go to step 11. If* $y_s(k) \neq x_s^0(k)$ *go to step 10.*

Step 10: *If* $g_j(x) \leq 0,$ $j = 1,...,m$ *let j := j + 1, and go to step 11. Otherwise, let* $x_s(k) := x_s^0(k),$ $k := k+1,$ *and go to step 11.*

Step 11: *If* $k \leq n$*, go to step 9. Otherwise, we obtain a feasible solution* ***x*** *corresponding to the individual* ***s*** *and stop.*

This decoding algorithm enables us to decode each of the individuals represented by the double strings to the corresponding feasible solution. However, the diversity of the solution ***x*** greatly depends on the reference solution, because solutions obtained by the decoding algorithm using reference solution tend to concentrate around the reference solution. To overcome such situations, the reference solution updating procedure [10, 11] is adopted here.

C. Fitness: Nature obeys the principle of Darwinian "survival of the fittest", the individuals with high fitness values will, on average, reproduce more often than those low fitness values. For obtaining satisfactory solution for all the DMs to MLLFPPs multi-level nonlinear integer programming problems (1) through GADSCRRSU, the objective function value is used as the fitness value z of an individual ***s***. When the variance of fitness in a population is small, it is often observed that the ordinary roulette wheel selection does not work well because there is little difference between the probability of a good individual surviving and that of a bad one surviving [10, 11]. In order to overcome this problem, the linear scaling [10, 11] is adopted here. The new fitness $z_i^{'}(s),\ i = 1,2,...,p.,$ of the DMi is obtained by using the following linear scaling

$$z_i^{'}(s) := a_i z_i(s) + b_i \tag{14}$$

Where, $z_i(s),$ $i = 1,2,...,p.,$ are the fitness values of the DMs at all levels with respect to each decoded individual ***s***.

D. Genetic Operators: For obtaining satisfactory solution for all the DMs to MLLFPPs (1) through GADSCRRSU, four genetic operators such as reproduction, partially matched crossover (PMX), bit reverse mutation and inversion [10, 11] are adopted here.

E. Usage of Continuous Relaxation: In order to find an approximate optimal solution with high accuracy in reasonable time, we need some schemes such as the restriction of the search space to a promising region, the generation of individuals near the optimal solution and so forth. From the point of view, the information about an optimal solution to the corresponding continuous relaxation problem is used in the generation of the initial population and the bit reverse mutation.

$$\min imize \quad Z(x)$$
$$\text{Subject to} \quad g_j(x) \leq 0, \quad j = 1,2.........m \tag{15}$$
$$0 \leq x_k \leq v_k, \quad k = 1,2,3,......n$$

When this problem is convex, we can obtain a global optimal solution by some convex programming technique, e.g., the sequential quadratic programming. Otherwise, i.e., when it is nonconvex, because it is difficult to find a global optimal solution, we search an approximate optimal solution by some approximate solution method such as genetic algorithms or simulated annealing. Here GENOCOP V [5] is used to find the solution of corresponding continuous relaxation problem (15).

F. Computational Procedures of GADSCRRSU: Now the genetic algorithms with double strings using continuous relaxation based on reference solution updating (GADSCRRSU) for solving nonlinear integer programming problems (13) are summarized in the following:

Step 0: Determine values of the parameters used in GADSCRRSU: the population size *N*, the minimal search generation $I_{\min}$, the maximal search generation $I_{\max} > I_{\min}$, the convergence criterion $\in$, the degree of use of information about solutions to nonlinear programming relaxation problem R , the parameter for feasible solution θ, the parameter for reference solution updating η, the parameter for augmented maxmin problem ρ, the upper bound of each decision variable *v*, the scaling constant *c*mult, the probability of crossover *pc*, the generation gap *G*, the probability of mutation p_m , the probability of inversion p_i and set the generation counter *r* at 0.

Step 1: Generate the initial population consisting of *N* individuals based on the information of a solution to the continuous relaxation problem (15).

Step 2: Decode each individual (genotype) in the current population and calculate its fitness based on the corresponding solution (phenotype).

Step 3: If the termination condition is fulfilled, stop. Otherwise, let *r* := *r* + 1 and go to step 4.
Step 4: Apply the reproduction operator based on the elitist expected value selection, after carrying out linear scaling.
Step 5: Apply the crossover operator, called PMX (Partially Matched Crossover) for double trings.
Step 6: Apply the mutation operator based on the information of an optimal solution to the continuous relaxation problem (15).
Step 7: Apply the inversion operator. Go to step 2.

VIII. Numerical Example

In this section we present numerical example to demonstrate the solution procedures by proposed GADSCRRSU to solve MLLFPP.The following example considered by Dey and Pramanik et al.[4] is again used to demonstrate the solution procedures and clarify the effectiveness of the proposed approach**:**

Consider the following Tri-Level Linear Fractional Programming Problem (TLLFPP)[4]:

$$\max_{x_1} z_1(x) = \frac{2x_1 + x_2 + 3x_3 + 3}{x_1 + x_2 + x_3} \quad \text{(First upper level)}$$

$$\max_{x_2} z_2(x) = \frac{x_1 + 4x_2 - 2x_3 + 1}{2x_1 + 2x_2 + x_3 + 3} \quad \text{(Second upper level)}$$

$$\max_{x_3} z_3(x) = \frac{3x_1 + x_2 - x_3 + 2}{4x_1 + 3x_2 - x_3} \quad \text{(Loewr level)}$$

subject to

$$2x_1 + x_2 + x_3 \le 7; \quad x_1 - 2x_2 + 3x_3 \le 4; \quad -x_1 + 2x_2 + 2x_3 \ge 1; \quad x_1 + 2x_2 \ge 3; \quad x_1 \ge 0, \quad x_2 \ge 0, \quad x_3 \ge 0$$

.

Solution of the above problem obtained Dey and Pramanik et al.[4] using fuzzy mathematical programming is:

Approach	Decision variables			Objective values		
FGP model(1)	$x_1 = 0$	$x_2 = 2$	$x_3 = 0$	$z_1 = 2.5$	$z_2 = 1.286$	$z_3 = 0.667$
FGP model(2)	$x_1 = 0$	$x_2 = 2$	$x_3 = 0$	$z_1 = 2.5$	$z_2 = 1.286$	$z_3 = 0.667$

The proposed GADSCRRSU approach gives the solution as:

Approach	Decision variables			Objective values		
GADSCRRSU Solution Approach	$x_1 = 2.60$	$x_2 = 1.23$	$x_3 = 0.57$	$z_1 = 2.53$	$z_2 = 0.78$	$z_3 = 0.77$

Which is very close or improved to the results obtained by the existing methods Dey and Pramanik et al [4].The solution by this approach is more reasonable and modified as each control variable has some non-zero value.

IX. Conclusion

Since the emergence of multi-level optimization problems at the beginning of the second decade of the last century, it has become a necessary requirement and has an important role to all areas and fields in the real world. From its early stages, it evolved systematically and scientifically through the genius of scientists and professionals in this field. It had passed through several stages, and it has branched more into various specialized disciplines in the real world. The GA approach to MLLFP problem proposed by us can produce results which are very close or improved to the results obtained by the existing methods. This approach 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 .Unique characteristic of a MLPP is reflected by including objective or solutions of each DM. One may use a random process that is biased towards the more fit members. Whenever the random mixture of features and any mutations result in an infeasible solution, this is a miscarriage, so the process of attempting to give birth then is repeated until a child is born that corresponds to a feasible solution.

References

[1] M. A. K. Azad, M. Sakawa, K. Kato and H. Katagiri, Interactive fuzzy programming for two-level nonlinear integer programming problems through genetic algorithms, Asia Pacific Management Review: An International Journal, vol. 10, no. 1, pp. 70-77, 2005 .

[2] Azad, M.A.K., Sakawa, M and Katagiri, H, Interactive fuzzy programming for multi-level nonlinear intger programming problems through genetic algorithms, Scientiae Mathematicae Japonicae online, pp. 223-236, e-2006.

[3] R. E. Bellman, L. A. Zadeh, Decision making in a fuzzy environment, Management Science, vol. 17, pp. 141-164, 1970.

[4] W. F. Bialas and M. H. Karwan, Two-Level Linear Programming, Management Science, vol.30, no. 8, pp. 1004-1020, 1984. 236 M.A.K. AZAD, M. SAKAWA, K. KATO AND H. KATAGIRI
[5] GENOCOP V, ftp://ftp.uncc.edu/coe/evol/zbyszek/genocop5.0.tar.Z
[6] R. Jan and M. Chern, Nonlinear Integer Bilevel Programming, European Journal of Operational Research, vol. 72, pp. 574-587, 1994.
[7] Y. J. Lai, Hierarchical optimization: a satisfactory solution, Fuzzy Sets and Systems, vol. 77, pp. 321-335, 1996.
[8] I. Nishizaki, M. Sakawa and K. Kato, Computational methods through Genetic Algorithms for obtaining Stackelberg solutions to Two-level Zero-One Programming Problems, Proceedings of 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation,pp. 2750-2755, 2000.
[9] I. Nishizaki and M. Sakawa, Computational Methods through Genetic Algorithms for obtaining Stackelberg solutions to Two-Level Mixed Zero-One Programming Problems, Cybernetics and Systems: An International Journal, Vol. 31, pp. 203-221, 2000.
[10] M. Sakawa, Genetic Algorithms and Fuzzy Multiobjective Optimization, Kluwer Academic Publishers, Boston, USA, 2002.
[11] M. Sakawa and K. Kato, Genetic algorithms with double strings for 0-1 programming problems, European Journal of Operational Research, vol. 144, pp. 581-597, 2003.
[12] M. Sakawa, I. Nishizaki and Y. Uemura, Interactive fuzzy programming for multi-level linear programming problems, Computers and Mathematics with Applications, vol. 36, pp. 71-86, 1998.
[13] M. Sakawa, I. Nishizaki and Y. Uemura, Interactive fuzzy programming for multi-level linear programming problems with fuzzy parameters, Fuzzy Sets and Systems, vol. 109, pp. 3-19, 2000.
[14] M. Sakawa, I. Nishizaki and M. Hitaka, Interactive fuzzy programming for multi-level 0-1 programming problems through genetic algorithms, European Journal of Operational Research, vol. 114, pp. 580-588, 1999.
[15] H. S. Shih, Y. J. Lai and E. S. Lee, Fuzzy approach for multi-level programming problems, Computers and Operations Research, vol. 23, pp. 73-91, 1996.
[16] K. Shimizu, Y. Ishizuka and J.F. Bard, Nondifferentiable and Two-Level Mathematical Programming, Kluwer Academic Publishers, Boston, USA, 1997.
[17] Ulrich Bodenhofer, Erich Peter Klement, Tuning Of Fuzzy Systems Using Genetic Algorithm, A thesis submitted to Institute fur Mathematics Johannes Kepler Universitt , Austria,1996.
[18] U. -P. Wen and W. F. Bialas, The hybrid algorithm for solving the three-level linear programming problem, Conputers and Operations Research, vol. 13, pp. 367-377, 1986.
[19] H. -J. Zimmermann, Fuzzy programming and linear programming with several objective functions, Fuzzy Sets and Systems, vol. 1, pp. 45-55, 1978.
[20] Partha Pratim Dey and Surapati Pramanik and C. Bibhas, Multi-Level Linear Fractional Programming Problem Based on Fuzzy Goal Programming Approach, International Journal of Innovative Research in Technology and Science(IJIRTS),4(2),2014, 17-26.