RESEARCH ARTICLE

Genetic algorithmic based approach for multiple-objective probabilistic fractional programming problem involving discrete random variables

Berhanu Belay*, Adane Abebaw

Department of Mathematics, Collage of Natural and Computational Sciences, Debre Tabor University, Debre Tabor, Ethiopia

* These authors contributed equally to this work.
* berhanubelay2@gmail.com

Abstract

This manuscript presents a technique for solving a multiple-objective probabilistic fractional programming problem with discrete random variables. A multiple-objective probabilistic mathematical model is constructed with fractional objectives. In the model, some parameters of coefficients and right hand side parameters of restrictions are assumed as random variables having Pascal and Hyper geometric distributions. The feasibility of probabilistic constraints is checked by means of stochastic simulation. Genetic algorithm approach method is used to obtain the Pareto optimal solution of the proposed model without finding the deterministic model. Genetic algorithm parameters are fixed in all generation. The proposed method is coded by C++ programming language. To illustrate the method, a numerical example and practical example on the case of supply chain management are presented. The result shows that the values of the objective functions are conflicting each other.

Introduction

Fractional mathematical programming is a programming problem where the cost function is the quotient of functions. It is applied when the constraints and cost functions are certain in nature. In other way some real life programming problems have unknown parameters due to uncertain nature. In this case, the uncertainty is handled by using probabilistic programming problem or fuzzy programming problem. A probabilistic fractional programming problems is a mathematical model with some or all data are assumed as random variables.

Optimizing many conflicting fractional objective with respect to the given restrictions is considered as multiple-objective fractional programming (MOFP) problem. When the data of (MOFP) problem are uncertain due to randomness, then it is termed as multiple-objective probabilistic fractional programming (MOPFP) problem.

In multiple-objective programming problem, there exists a set of solutions unlike single objective programming problem, which are superior to the rest of the solutions. The superiority is decided by comparing the values of objective functions. Since multiple-objective
programming problems are always in commensurable and conflicting with each other, the best solution is decided by their dominance.

The solution $X_1$ dominates $X_2$ if

- In all objective functions $X_1$ is not worse than $X_2$.
- At least for one objective function, $X_1$ is strictly better than $X_2$.

In other way, non-dominated solution set is the collection of solutions, which are not dominated by other solution set.

Genetic Algorithm (GA) is the heuristic method that works on the concept of survival of the fittest, which is created by John Holland in 1975. The algorithm performs on strings known as individuals of a given population. GA is applied according to four operators. These operators are known as selection (choice), crossover (hybrid), mutation (change), and elitism.

Using a fixed iteration $n$, collection of population $P(n)$ will be produced. The strings of $P(n)$ are a solution to the mathematical problem. Utilizing the choice or selection operator, new population is obtained from the old population. The fitter string has a big likelihood of being chosen to the modern population. Then the population takes place the method of hybrid and change. In hybrid, two parent strings are selected randomly using fixed crossover likelihood (probability) that makes best property of original strings. Using mutation, modern offsprings can be made by a few alter within the quality of the chromosome according to mutation likelihood (probability).

After the method of choice, crossover and changes are completed, the other operator, which is called elitism can be connected for advancement of the coming generation. Elitism helps to handle better strings and remove bad strings. There are different elitism operators. Some of these are: alternate, partial and, complete elitism. In this manuscript, complete elitism has been used, which works with the following steps.

1. Merge the population strings before selection and after mutation so that the population is doubled.
2. Sort the resulting population in increasing or decreasing order according to the value of fitness function.
3. Then the first best half strings are selected for next generation. i.e, the strings having good fitness function remain in the system and go to another generation where as the poor strings are discarded.

When we increment the generation, the leading fitted string survives. If the ending criteria reaches, calculation stops and gives the best individual solution for the proposed problem.

The best advantages and practicability of genetic algorithm are:

- GA search the solution from the population of points, not a single point.
- GA uses objective function information, not derivatives
- GA can solve multiple objectives
- GA uses probabilistic orators, not deterministic transitions.
- The concept of GA is easy to understand and use.
- GA is suitable for low computational fitness functions.

Even if GA has the above advantages, it has its own limitation. Some of the limitations are for high computational fitness function or high dimensional problems it takes more
computation time (expensive), it needs statistical convergent solution with many simulation, the convergence of the solution depends on initial solution (strings).

We use GA instead of other Heuristic algorithm for multiple-objective cases due to the following reasons:

- the probability of exploration of the search area in GA is very high comparative other heuristic algorithms.
- Simplicity and ease of implementation.
- Large and wide solution space search ability
- Easy to discover global optimum and avoid trapping in local optima.
- The concept of GA is easy to understand and use.
- GA is Parallelism, easily modified and adaptable to different problems
- Genetic representations using chromosomes.

Most real life mathematical programming problems involve more than one fractional objective functions involving uncertain parameters which are conflicting each other. Unlike single objective programming problems, which give unique optimal solution, there may be many non dominated solutions in multiple-objective fractional probabilistic programming problem. Finding the deterministic equivalent of such multiple objective fractional programming under discrete random variable is difficult. As a result, there must be a method that is used to solve such type of problem without finding the deterministic equivalent.

There is lack of studies which solve multiple-objective fractional programming involving discrete random variables using genetic algorithm approach. In addition, the problem has many real life applications. All the above cases, motivate me to conduct and consider the proposed mathematical model as multiple-objective probabilistic fractional programming problem.

This paper has been organized into 6 Sections and at last references do appear. Section 1 focuses on introduction to multiple-objective probabilistic fractional programming and GA. Section 2 presents review of literature. Section 3 is the mathematical model of MOPFP problem. The solution procedure of MOPFP problem is stated in Section 4. Examples for the the proposed model are given in Section 5. At the last section, the conclusion is provided.

**Literature survey**

Many authors published the solution methodology for multiple-objective fractional mathematical problems. Such as [1] presented a method by converting into multiple-objective mathematical problem. The challenge of programming problem in real world application is deciding the precise values of available data. The programming problem under randomness is probabilistic model. To get the optimal value of these programming model, it’s deterministic equivalent is found by taking the data as continuous random variable [2, 3]. Some other researchers [4–7] proposed a method for stochastic programming problems having multiple-objectives. But, getting the deterministic of probabilistic model is time wastage and even difficult. To overcome this difficulty different researchers suggested GA for mathematical problem involving continuous random variable [8–11]. If few of the data of fractional mathematical problems are described by random variable, it is referred to as stochastic fractional programming. Zhu and Huang [12] stated the use of fractional mathematical problem involving random variables in the use of electric power. Das and Mandel [13] solved stochastic fractional programming problems by transforming to deterministic fractional problem. [14] showed the application of
probabilistic fractional goal programming involving multiple-objectives for the distribution of water resource in industries. GA is proposed to find chance constraint fractional programming by handling the probabilistic limitations using stochastic simulation [15–17]. In some real life circumstances, all or some data of the probabilistic programming problem are expressed by discrete random variables. Though, the solution method of MOPFP problem involving discrete distribution does not appear in the literature. In stochastic programming problem where it is difficult to find its deterministic, [9] have estimated the probabilistic constraints by using random variable generation and stochastic simulation. Different researchers solved single objective probabilistic programming having random variables with discrete distributions [18–20].

From the literature that we observed, many authors tried to solve multiple-objective probabilistic programming problem by finding the deterministic equivalent. In addition some authors solved multiple-objective fractional programming problems by using classical method of fractional programming by transforming into equivalent multiple-objective programming problem. All these methods take time and even difficult to find the deterministic equivalence when the parameters involve some discrete random variables like hyper geometric and pascal distributions.

Recently, [21] presented an optimization model for software quality prediction with case study analysis using MATLA. [22] proposed Consensus based combining method for classifier ensembles. [23] studied efficient malware detection approach with feature weighting based on Harris Hawks optimization. [24] Eccentric methodology with optimization to unearth hidden facts of search engine result. [25] studied a new insight on solving fuzzy linear fractional programming in material aspects. [26] proposed fuzzy nonlinear programming approach for multi-objective sum of linear and linear fractional programming problem. [27] proposed a numerical approach for solving linear fractional programming problem in a fuzzy environment. [5] presented multi-objective probabilistic fractional programming problem involving two parameters Cauchy distribution. [28] Studied on probabilistic multi-objective linear fractional programming problems under fuzziness. [29] proposed elite artificial bees’ colony algorithm to solve robot’s fuzzy constrained routing problem [30] studied application of multi-objective probabilistic fractional programming problem in production planning. [31] studied application of fuzzy random-based multi-objective linear fractional programming to inventory management problem.

So in this manuscript, we have made an attempt for solving MOPFP problem having discrete random variables using stochastic simulation based genetic algorithm.

**Formulation of MOPFP problem**

MOPFP Problem is expressed as:

$$\max : Z_q = \frac{N_q(x_j)}{D_q(x_j)} \quad q = 1, 2, \ldots, Q$$  \hspace{1cm} (1)

subject to

$$P\left(\sum_{j=1}^{n} a_j x_j \leq b_i\right) \geq \alpha_i, \quad i = 1, 2, \ldots, m$$  \hspace{1cm} (2)

$$0 < x_i < 1, \quad i = 1, 2, \ldots, m$$  \hspace{1cm} (3)

$$x_j \geq 0, \quad j = 1, 2, \ldots, n$$  \hspace{1cm} (4)
where $N_q(x_j)$ and $D_q(x_j)$ are function of $x_j$. “P” is the probability, $\alpha$ is the number in between 0 and 1 which is the violation.

In this manuscript, we consider the parameter in the coefficient of constraint and the right hand side are hyper-geometric and Pascal random variables respectively.

Suppose that $x$ is a random variable:

(a): $x$ has Pascal distribution with known parameters $n$ and $p$, whose probability mass function is equal to:

$$f(x) = \begin{cases} 
\frac{(x-1)!}{(n-1)!} p^n (1-p)^{x-n}, & \text{if } x = n, n+1, n+2, \ldots, (0 < p < 1) \\
0, & \text{otherwise}
\end{cases}$$

it is denoted by $PS(n, p)$

(b): $x$ has Hyper geometric distribution, if the probability mass function is expressed:

$$f(x) = \binom{n_1}{x} \binom{N-n_1}{m-x} \binom{N}{m}, \max(0, n_1 + m - N) \leq x \leq \min(n_1, m)$$

and it is denoted by $H(N, m, n_1)$. It is a discrete distribution of sampling without replacement. The parameters $N, m, n_1, N-n_1$ represents total population size in both classes, sample size, first class population size and second class population size respectively.

**Solution procedure**

The compromise solution of the proposed mathematical programming is obtained by stochastic simulation together with GA. In this case:

- Finding deterministic equivalence of chance constraint programming problem is not required
- Transforming MOFP problem into non fractional mathematical problem is also not required.

The algorithms is described by the following procedure:

**step 0**: Specify the parameters such as crossover probability ($pc$), mutation probability ($pm$), number of generation ($t$) and distribution parameters.

**step 1**: Randomly initialize the population $P(t)$ of GA.

**step 2**: Based on the choice making circumstance, distinguish the two bounds the chosen variables.

**step 3**: Calculate the functions to be optimized.

**step 4**: Test the likelihood condition of the limitations. In the event that the condition is satisfied at that point go to following step else go to step 1.

**step 5**: Apply selection operator to select the best chromosomes from the population.

**step 6**: Apply crossover operator to produce new pair of chromosomes.

**step 7**: Mutation operation is applied to the existing population of chromosomes to have variety in the present population.
**step 8:** Evaluate the functions to be optimized again.

**step 9:** Check the likelihood criteria of the limitations once more. In the event that the criteria is fulfilled at that point go to following step else go to step 1.

**step 10:** Apply Elitism to the present population.

**step 11:** Check whether max number of generation is achieved or not, if not go to step-3.

**step 12:** When the end condition is come to the show, the present populace gives the leading solution.

The pseudo-code for stochastic simulation based GA is described by:

```
Begin
  t = 1;
  initialize population P(t);
  calculate fitness function;
  while(termination criteria is not reached)
    do
      t = t+1;
      check probability criteria;
      Choose P(t) from p(t-1);
      applying crossover;
      applying mutation;
      evaluate fitness function p(t+1);
      check probability criteria;
      apply elitism;
    end while
  end begin
```

The stream chart for stochastic simulation together with GA is described by Fig 1.

**Examples**

One numerical example and one real life application will be discussed here as follows.

**Numerical example**

Consider the multiple-objective probabilistic fractional programming.

\[
\max \quad Z_1 = \frac{x_1 + 6x_2 - 3x_3}{3x_1 + 8x_2 + x_3 + 4}
\]

\[
\max \quad Z_2 = \frac{-6x_1 + 7x_2 + 2x_3}{2x_1 + 3x_2 + 6}
\]

subject to

\[
P(a_1x_1 - 4x_2 - a_2x_3 \leq b_1) \geq 0.75
\]

\[
P(5x_1 - a_3x_2 - a_4x_3 \leq b_2) \geq 0.55
\]

\[
P(a_5x_1 - a_6x_2 + 3x_3 \leq b_3) \geq 0.65
\]

\[0 \leq x_i \leq 5, \quad i = 1, 2, 3.\]
Fig 1. Stream chart of GA.

https://doi.org/10.1371/journal.pone.0274619.g001
where (i): \(a_1, a_2, a_3, a_4, a_5, a_6\) are Pascal distribution with two parameters \(n\) and \(p\): 4 and 0.6, 5 and 0.7, 3 and 0.8, 5 and 0.9, 2 and 0.75, 3 and 0.85 respectively.

(ii): \(b_1, b_2, b_3\) are hyper-geometric random variables with three parameters \(N, m, n_1\): 40, 17, 15 and 10, 8, 4 and 50, 24, 20 respectively.

To discover the Pareto ideal solutions, we utilize stochastic recreation based GA. The stochastic reenactment GA is coded by utilizing C++ programming dialect. Within the calculation, we utilize competition determination, one point cross over and bit wise transformation administrators and the greatest generation as stopping criteria. The other parameter are given as: size of population \(p(t) = 100\), \(p_m = 0.001\) and \(p_c = 0.7\), and maximum generation \(t\) taken as 200.

In this manuscript the GA parameters are fixed in all cases. The population size should not be very large as a huge computational time is required and which causes GA to slow down, while smaller population might not be enough for a good mating pool. Therefore, an optimal population size needs to be decided by trial and error. That is why we choose the population size \(p(t) = 100\). Crossover has a higher probability, typically 0.7–0.95 is recommended. The experiments we did with crossover showed that a rate of 80% is good enough. This means that \(p_c = 0.7\) is sufficient for our problem. On the other hand, mutation is carried out by flipping some digits of a string, which generates new solutions. For our problem, Mutation rate is in the range of [0.001, 0.05]. In general, this mutation probability is typically low, from 0.001 to 0.05 is recommended for GA. In our cases \(pm = 0.001\) is good choice. Number of generation refers to the number of cycles before termination. In some cases 100 loops are sufficient, but in the other cases more generation is needed. This depends on the problem type and its complexity. So in our problem \(t = 200\) generation is sufficient.

Solving the problem using stochastic simulation GA, we got non dominated solutions which is given by Table 1 with 200 generations. The set of non dominated points are expressed by Fig 2.

As we see the results, the non dominated or compromise solutions are conflicting each other. This means that when \(Z_1\) is best then \(Z_2\) is worst and vice versa. In this case the decision maker can choose one of the best solution depending on his/her interest.

The uncertain coefficients of the constraints are approximated by stochastic simulation based GA method. The method generates infinity many solutions, among these solution the 14 solutions are none dominated solutions.

Table 1. Non dominated solutions and values of \(Z_1\) and \(Z_2\).

| \(x_1\) | \(x_2\) | \(x_3\) | \(Z_1\) | \(Z_2\) |
|-------|-------|-------|-------|-------|
| 0     | 4.98045 | 0.909091 | 0.60678 | 1.75162 |
| 0     | 4.98045 | 1.22678 | 0.58136 | 1.78196 |
| 0     | 4.99511 | 4.73118 | 0.32299 | 2.11665 |
| 0     | 4.98045 | 2.31672 | 0.49680 | 2.71665 |
| 0     | 5      | 2.45357 | 0.48735 | 1.90034 |
| 0     | 4.9609 | 4.15934 | 0.36130 | 2.06127 |
| 0     | 5      | 0.410557 | 0.64634 | 1.86806 |
| 0     | 5      | 2.70772 | 0.46837 | 1.92454 |
| 0     | 5      | 2.12121 | 0.51248 | 1.86869 |
| 0     | 5      | 0.293255 | 0.67936 | 1.66946 |
| 0     | 5      | 2.51222 | 0.482956 | 1.90593 |
| 0     | 5      | 3.75367 | 0.39241 | 2.02416 |
| 0     | 5      | 22.6784 | 0.470556 | 1.92175 |
| 0     | 5     | 4.3695 | 0.349218 | 2.08281 |

https://doi.org/10.1371/journal.pone.0274619.t001
Real life example of supplier selection in supply chain problem

Suppliers have a great role in manufacture or purchasing to satisfy the need of the customers. The main strategy of the decision maker in supply chain management is selecting the best suppliers that archive the targeted goal. In practical supplier selection is affected by conflicting criteria. The most important criteria are quality, cost and delivery time. In addition to this some parameters like total demand and maximum capacity of suppliers are affected by uncertainty. This uncertainty can be characterized by random variables having known distributions. Hence probability theory is the best tool to handle uncertainty. In this example, we develop a multi-objective fractional programming problem for supplier selection problem subject to probabilistic constraints with uncertain demand and maximum capacity of suppliers.

Suppose that 4 suppliers are needed for supplying new products to a market. The market manager designed criteria for selecting suppliers based on cost of product, quality of product and delivery time. It is assumed that the total demand of the market and maximum capacity of each supplier are random variables following Pascal distribution and hyper-geometric distribution. The manager wants to select suppliers which minimize cost per late time deliver ratio and maximize quality of product per on time delivery ratio subject to probabilistic demand and supplier capacity. The supplier quantities are given by Table 2.

Model development. The supplier selection model is multiple-objective fractional programming problem with probabilistic constraints.

The cost functions are:

1. the quality of item per on time delivery i.e
2. cost per late time delivery
the cost functions are evaluated under the following limitations.

1. the total product supplied by suppliers is not less than the total demand of the customer.
2. the capacity of each supplier cannot exceed the maximum capacity.

Therefore, the general mathematical model of suppliers selection is expressed by:

\[
\begin{align*}
\max Z_1 &= \frac{\sum_{i=1}^{4} q_i x_i}{\sum_{i=1}^{4} o_i x_i} \\
\min Z_2 &= \frac{\sum_{i=1}^{4} c_i x_i}{\sum_{i=1}^{4} l_i x_i}
\end{align*}
\]

subject to

\[
\begin{align*}
P(\sum_{i=1}^{4} x_i \geq D) &\geq \beta \\
P(x_i \geq C_i) &\geq \alpha_i \\
x_i &\geq 0, i = 1, 2, 3, 4.
\end{align*}
\]

where:
- \(q_i\): percentage of quality of product of supplier \(i\).
- \(o_i\): percentage of on time delivery of supplier \(i\).
- \(l_i\): percentage of late time delivery of supplier \(i\).
- \(c_i\): cost of product supplied by supplier \(i\).
- \(D\): uncertain demand of the customer which follows Paskal distribution with two parameters \(n\) and \(p\).
- \(C_i\): uncertain maximum capacity of supplier \(i\) which follows beta distribution with three parameters \(N, m, n_1\).
- \(x_i\): quantity of supplier \(i\).
- \(P\): is the probability.
- \(\alpha_i\) and \(\beta\) are the given probabilities at which the constraint violations are admitted.

Substituting all the data values given in Table 2, we obtain the following multiple-objective fractional probabilistic programming problem:

\[
\begin{align*}
\max Z_1 &= \frac{0.75x_1 + 0.80x_2 + 0.95x_3 + 0.65x_4}{0.65x_1 + 0.75x_2 + 0.80x_3 + 0.9x_4} \\
\min Z_2 &= \frac{4x_1 + 2x_2 + 3x_3 + 5x_4}{0.35x_1 + 0.25x_2 + 0.2x_3 + 0.1x_4}
\end{align*}
\]

Table 2. Supplier quantity data.

| supplier | \(x_i\) | \(q_i\) | \(o_i\) | \(l_i\) | \(nC_i\) | \(mC_i\) | \(n_1C_i\) |
|----------|--------|--------|--------|--------|--------|--------|--------|
| 1        | 4      | 75     | 65     | 35     | 100    | 75     | 50     |
| 2        | 2      | 80     | 75     | 25     | 80     | 60     | 40     |
| 3        | 3      | 95     | 80     | 20     | 90     | 65     | 50     |
| 4        | 5      | 65     | 90     | 10     | 120    | 80     | 65     |
subject to
\[ P(x_1 + x_2 + x_3 + x_4 \geq D) \geq 0.7 \]  \hspace{1cm} (12)
\[ P(x_1 \geq C_1) \geq 0.65 \]  \hspace{1cm} (13)
\[ P(x_2 \geq C_2) \geq 0.85 \]  \hspace{1cm} (14)
\[ P(x_3 \geq C_3) \geq 0.45 \]  \hspace{1cm} (15)
\[ P(x_4 \geq C_4) \geq 0.35 \]  \hspace{1cm} (16)
\[ x_1, x_2, x_3, x_4 \geq 0. \]  \hspace{1cm} (17)

where (i): \( D \) is random variable following Pascal distribution with two parameters \( n = 5 \) and \( p = 0.01 \):

(ii): \( C_1, C_2, C_3, C_4 \) are random variables following hyper-geometric distribution with three parameters \( N, m, n_1 \) having value given in Table 2.

Results and discussion. The non dominated solutions are gotten utilizing stochastic recreation based GA by utilizing \( p(t) = 100, p_c = 0.7, p_m = 0.001 \) and \( t = 200 \) and dissemination data stated above. Coding stochastic reenactment based GA utilizing C++, we get the taking after non dominated ideal solutions of supplier determination demonstrate. Table 3 appears the dominated solutions and ideal values of the two objective functions.

The pareto frontier points the supplier selection problem are expressed by Fig 3.

Conclusion

In this manuscript, MOPFP problem has been solved using stochastic simulation together with GA by waving the deterministic identical of the problem. The method doesn’t require any

| \( x_1 \) | \( x_2 \) | \( x_3 \) | \( x_4 \) | \( Z_1 \) | \( Z_2 \) |
|---|---|---|---|---|---|
| 1.17302 | 14.9756 | 9.83382 | 0 | 1.11788 | 10.5805 |
| 3.75367 | 19.9609 | 12.4927 | 0 | 1.11849 | 10.7273 |
| 2.73705 | 17.4976 | 9.52102 | 0 | 1.11442 | 10.4943 |
| 1.54448 | 14.4477 | 7.46823 | 0 | 1.1121 | 10.3212 |
| 0.938416 | 17.6344 | 9.05181 | 0 | 1.1107 | 10.1805 |
| 2.24829 | 10.7918 | 8.79765 | 0 | 1.1256 | 11.101 |
| 16.9697 | 13.001 | 7.46823 | 0 | 1.12959 | 11.8239 |
| 16.6569 | 19.7263 | 19.824 | 0.0391007 | 1.13527 | 11.9266 |
| 16.2659 | 18.7488 | 20 | 0 | 1.13693 | 11.9821 |
| 0 | 20 | 20 | 0 | 1.12903 | 11.1111 |
| 9.97067 | 8.11339 | 12.45367 | 0.0195503 | 1.4485 | 12.4554 |
| 14.1349 | 5.8651 | 19.2571 | 0 | 1.1585 | 13.1867 |
| 16.0899 | 7.46823 | 19.5112 | 0 | 1.15501 | 13.0074 |
| 1.19257 | 5.45455 | 19.5699 | 0 | 1.162141 | 13.2003 |
| 0.449658 | 4.69208 | 15.5425 | 0 | 1.16072 | 13.0899 |
| 1.75953 | 3.57771 | 16.0117 | 0 | 1.1657 | 13.4557 |
| 8.97361 | 3.42131 | 9.16911 | 0 | 1.15332 | 13.0536 |

https://doi.org/10.1371/journal.pone.0274619.t003
classical method of fractional programming to convert the multiple-objective fractional programming problem into identical multiple-objective programming problem. The method is simple and easy to use which is coded by C++ programming language. Moreover, GA helps to generate many solutions at a time which are non dominated solutions. This is important for decision makers to make a good decision by considering all the possible directions. In the proposed method, when the number of iteration increases the probability of getting best non dominated solution is very high.

Since most real life problems are uncertain due to randomness, stochastic simulation can easily handle the probabilistic constraints. For such type of mathematical problem the proposed method is used to get the non dominated solution for multiple-objective fractional uncertain problems for which the decision maker can select best supplier among many suppliers.

We recommend that the proposed problem can be solved by other heuristic algorithms like particle swarm algorithm and ant colony algorithm. In addition the problem can extended to fully fuzzy multi-objective probabilistic fractional programming problem and multi-level multi objective probabilistic fractional programming problem that involves both discrete and continuous random variables.

Author Contributions

Conceptualization: Berhanu Belay.
Methodology: Berhanu Belay.
Software: Adane Abebaw.
Writing – original draft: Berhanu Belay.
Writing – review & editing: Berhanu Belay.
References

1. Güzel N. A proposal to the solution of multiobjective linear fractional programming problem. In: Abstract and applied analysis. vol. 2013. Hindawi; 2013

2. Charnes A, Cooper W. Chance constraints and normal deviates. Journal of the American Statistical Association. 1962; 57(297):134–148. https://doi.org/10.1080/01621459.1962.10482155

3. Hulsurkar S, Biswal MP, Sinha SB. Fuzzy programming approach to multi-objective stochastic linear programming problems. Fuzzy Sets and Systems. 1997; 88(2):173–181. https://doi.org/10.1016/S0165-0114(96)00056-5

4. Charles V, Ansari S, Khalid M. Multi-objective stochastic linear programming with general form of distributions. Int J Oper Res Optim. 2011; 2(2):261–278.

5. Acharya S, Ranarahu N, Dash J, Acharya M. Solving multi-objective probabilistic programming problem. Journal of Intelligent & Fuzzy Systems. 2014; 26(2):935–948. https://doi.org/10.3233/IFS-130784

6. Javaid S, Ansari S, Anwar Z. Multi-objective stochastic linear programming problem when b’s follow Weibull distribution. Opsearch. 2013; 50(2):250–259. https://doi.org/10.1007/s12597-012-0101-6

7. Zhou J, Yang F, Wang K. Multi-objective optimization in uncertain random environments. Fuzzy Optimization and Decision Making. 2014; 13(4):397–413. https://doi.org/10.1007/s10700-014-9183-3

8. Iwamura K, Liu B. A genetic algorithm for chance constrained programming. Journal of Information and Optimization sciences. 1996; 17(2):409–422. https://doi.org/10.1080/02522667.1996.10699291

9. Jana R, Biswal M. Stochastic simulation-based genetic algorithm for chance constraint programming problems with continuous random variables. International Journal of Computer Mathematics. 2004; 81(9):1069–1076. https://doi.org/10.1080/03057920412331272144

10. Poojari CA, Varghese B. Genetic algorithm based technique for solving chance constrained problems. European journal of operational research. 2008; 185(3):1128–1154. https://doi.org/10.1016/j.ejor.2006.06.045

11. Dutta S, Acharya S, Mishra R. Genetic algorithm approach for solving multi-objective fuzzy stochastic programming problem. International Journal of Mathematics in Operational Research. 2017; 11(1):1–28. https://doi.org/10.1504/IJMOR.2017.085377

12. Zhu H, Huang G. Dynamic stochastic fractional programming for sustainable management of electric power systems. International Journal of Electrical Power & Energy Systems. 2013; 53:553–563. https://doi.org/10.1016/j.ijepes.2013.05.022

13. Das SK, Mandal T. A single stage single constraints linear fractional programming problem: An approach. Operations Research and Applications: An International Journal (ORAJ). 2015; 2(1):9–14.

14. Ren CF, Li RH, Zhang LD, Guo P. Multiobjective stochastic fractional goal programming model for water resources optimal allocation among industries. Journal of Water Resources Planning and Management. 2016; 142(10):04016036. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000681

15. Udhayakumar A, Charies V, Uthieranaj VR. Stochastic simulation-based genetic algorithm for chance constrained fractional programming problem. International Journal of Operational Research. 2010; 9(1):23–38. https://doi.org/10.1504/IJOR.2010.034359

16. Charles V, Dutta D. A parametric approach to linear probabilistic fractional programming problems. Mathematical & Computational Models. 2005;(13):171–183.

17. Saad OM, Biltagy MS, Farag TB. An algorithm for multi objective integer nonlinear fractional programming problem under fuzziness. Gen. 2011; 2(1):1–17.

18. Ruszczyński A. Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. Mathematical Programming. 2002; 93(2):195–215. https://doi.org/10.1007/s10107-002-0337-7

19. Prekopa A, Vizvari B, Badics T. Programming under probabilistic constraint with discrete random variable. In: New trends in mathematical programming. Springer; 1998. p. 235–255.

20. Dentcheva D, Prékopa A, Ruszczynski A. Concavity and efficient points of discrete distributions in probabilistic programming. Mathematical Programming. 2000; 89(1):55–77. https://doi.org/10.1007/PL00011393

21. Gheisari M, Panwar D, Tomar P, Harsh H, Zhang X, Solanki A, et al. An optimization model for software quality prediction with case study analysis using MATLAB. 7:85123–85138.

22. Alzubi OA, Alzubi JAA, Tedmon S, Rashaideh H, Alomani O. Consensus-based combining method for classifier ensembles.; 15(1):76–86.

23. Alzubi OA, Alzubi JA, Al-Zoubi A, Hassonah MA, Kose U. An efficient malware detection approach with feature weighting based on Harris Hawks optimization; p. 1–19.
24. Sethuraman J, Alzubi JA, Manikandan R, Gheisari M, Kumar A. Eccentric methodology with optimization to unearth hidden facts of search engine result pages; 12(2):110–119.
25. Das SK, Edalatpanah SA. New insight on solving fuzzy linear fractional programming in material aspects. Fuzzy Optimization and Modeling Journal. 2020 Oct 1; 1(2):1–7.
26. Veeramani C, Sharanya S, Ebrahimnejad A. Optimization for multi-objective sum of linear and linear fractional programming problem: fuzzy nonlinear programming approach; 14(3):219–233.
27. Ebrahimnejad A, Ghomi SJ, Mirhosseini-Alizamini SM. A revisit of numerical approach for solving linear fractional programming problem in a fuzzy environment. Applied Mathematical Modelling. 2018 May 1; 57:459–73. https://doi.org/10.1016/j.apm.2018.01.008
28. Khalifa H, Ammar E. Study on probabilistic multi-objective linear fractional programming problems under fuzziness; 31(1):1–12.
29. Abbaszadeh Sori A, Ebrahimnejad A, Motameni H. Elite artificial bees’ colony algorithm to solve robot’s fuzzy constrained routing problem; 36(2):659–681.
30. Belay B, Acharya S, Mishra R. Application of multi-objective probabilistic fractional programming problem in production planning; 43(1-2):150–167.
31. Abd El-Wahed Khalifa H, Kumar P, Alodhaibi SS. Application of fuzzy random-based multi-objective linear fractional programming to inventory management problem. Systems Science & Control Engineering. 2022 Dec 31; 10(1):90–103. https://doi.org/10.1080/21642583.2022.2040030