CHAPTER 1

INTRODUCTION

1.1 Introduction

Optimization refers to finding a feasible solution which correspond to extremum value for one or more objective functions. The need for finding such an optimal solution to a problem comes mostly for the purpose of either designing a solution with minimum possible cost, or with maximum possible reliability, or others.

When an optimization problem modeling of a physical system involves only one objective function, the task of finding the optimal solution is called single-objective optimization. Since the time of the Second World War, most efforts have been made to understand, develop and apply single-objective optimization methods.

In the world around us it is rare for any problem to concern only a single value or objective. Generally multiple objectives or parameters have to be met or optimized before any solution is considered adequate. Multi objective optimization methods are of great importance in practice, particularly in engineering design, scientific experiments and business decision making.

When an optimization problem involves more than one objective function, the task of finding optimum solution is known as multi-objective optimization. Most real world search and optimization problems naturally involve multiple objectives. The extremism cannot be applied to only one objective, when the rest of the objectives are also important. Different solutions may produce trade-offs
among different objectives. A solution that is extreme with respect to one objective requires a compromise in other objectives.

The problem of optimizing one or several ratios of functions is called **fractional programming**. In applications of single-ratio fractional programming, numerator and denominator may be representing output, input, profit, cost, capital, risk or time. In decision making, when several ratios are to be optimized simultaneously and a compromise is sought which optimizes a weighted sum of these ratios. This describes situations where a compromise is sought between absolute and relative terms like profit and return on investment (profit/capital) or return and return/risk.

The analysis of fractional programs with only one ratio has largely dominated the literature until about 1980. A series of international conferences were held which demonstrates a shift of interest from the single-ratio to the multi-ratio case[9]. The sum-of-ratios fractional program has important applications in several areas such as production, transportation, finance, engineering, statistics etc. One of the most difficult fractional programs encountered so far is the sum-of-ratios problem[60]. It proves to be essentially NP-complete.

### 1.2 Formulation of Linear Programming Problem

In formulating a mathematical model[2,42], there should be an excellent correlation between the answers to the problem predicted by the model and what would happen in the real world. The important aspect of formulating the model is the construction of an objective function which requires a quantitative measure of effectiveness. If there are many objectives, it is necessary to combine the respective measures into a composite measure of effectiveness.
The linear programming model involves a linear function\([12,14,18,54]\) of several variables to be optimized (maximized or minimized) subject to a set of linear constraints and non-negativity restrictions\([31,62]\) on the variables.

A Linear Programming problem\([10]\) in its general form is shown below

Extremize Objective Function  
\[ Z = c_1 x_1 + c_2 x_2^+ + \ldots + c_n x_n + c_0 \]

Subject to the Constraints:

\[ a_{11} x_1 + a_{12} x_2^+ + \ldots + a_{1n} x_n \quad (\ast) b_1 \]
\[ a_{21} x_1 + a_{22} x_2^+ + \ldots + a_{2n} x_n \quad (\ast) b_2 \]
\[ \ldots \ldots \ldots \ldots \]
\[ a_{m1} x_1 + a_{m2} x_2^+ + \ldots + a_{mn} x_n \quad (\ast) b_m \]
\[ X_j \geq 0 \]

Where,

- \(C_j\) is the cost coefficient of \(X_j\)
- \(X_j\) is a decision variable
- \(a_{ij}\) is a known constant
- \(\ast\) stands for \(\geq\) or \(=\) or \(\leq\) for each constraint

### 1.3 Linear Fractional Programming Problem

Problems concerning optimization of an objective function which is a ratio of two functions subject to a set of constraints and non-negativity constraints are called fractional programming problems\([36]\). When the numerator and denominator of the objective function and the constraints are linear, the problem is called linear fractional programming problem. Mathematically it can be represented as

Extremize  
\[ Z = \frac{C^T X + c_0}{D^T X + d_0} \]
Subject to \( AX \leq p_0 \)

Where \( X \geq 0 \)

i) \( X \) is the decision vector of order \( n \times 1 \).

ii) \( A \) is the constraint matrix of order \( m \times n \).

iii) \( C \) and \( D \) are contribution coefficient vectors of order \( n \times 1 \).

iv) \( p_0 \) is the resource vector of order \( m \times 1 \).

v) \( c_o \) and \( d_o \) are scalars.

vi) \( n \) and \( m \) are number of decision variables and constraints respectively.

### 1.4 Solution to Linear Fractional Programming Problem

Few solution methods to solve linear fractional programming problems available in the literature are outlined in the following sections.

#### 1.4.1 Ratio Algorithm

In the ratio algorithm\[48\], the ratio of the respective contribution coefficients in the numerator and denominator of the objective function is calculated for each of the promising variables. These ratios are used to select an entering variable. Once a variable is considered to enter into the basis, the variable is entered into the basis as per the simplex procedure used for the solution of linear programming problems. It is also assumed that the value of the denominator of the objective function will always be positive.

#### 1.4.2 Modified Ratio Algorithm

Modified ratio algorithm\[37\] used for the solution of linear fractional programming problems first arranges the promising variables by computing the objective value due to each of the decision variables. Then, the arranged variables are made to enter into the basis one by one after checking if they are still promising. The computational efficiency of the revised simplex algorithm lies in utilizing the inverse of the current basic matrix to generate the next inverse. For
this, the revised simplex procedure is a univariate search technique and suffers the drawback of slow convergence with the tendency of variables popping in and popping out of the basis.

1.5 Optimizing the Sum of Linear Fractional Functions

The problem of optimizing the sum of \( m \) linear fractional functions subject to \( m \) linear constraints, arises in a number of theoretical and applied areas. The problem of optimizing the sum of linear fractional is defined as follows:

\[
\text{max/min } f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{m} \frac{n_i(x_1, \ldots, x_n)}{d_i(x_1, \ldots, x_n)}
\]

such that for each \( i = 1, 2, \ldots, m \), \( n_i(x_1, x_2, \ldots, x_n)\) and \( d_i(x_1, x_2, \ldots, x_n) \neq 0 \) and \( \in S \), the feasible domain \( S \) is defined by \( m \) linear constraints. Each linear fractional term \( \frac{n_i(x_1, \ldots, x_n)}{d_i(x_1, \ldots, x_n)} \) is called a ratio. Many economic applications (maximization of productivity, return on investment, and return/risk) can be reduced to solving the sum of linear fractional functions. Solution methods that solve the problem involving fractional functions are outlined in the following sections.

1.5.1 Fractional Programming: Sum of ratios case

One of the most difficult fractional programs encountered so far is sum-of-ratios problem. The research paper [15] provides a recent survey of applications, theoretical results and various algorithmic approaches for this challenging problem.

Y. Almogy and Levin tried to extend Dinkelbach’s method to the sum-of-ratios. The algorithm is based on decoupling numerators and denominators.

Cambini, Martein and Schaible show that a linear sum-of-ratios problem can be reduced to a linear function by applying the Charnes-Cooper
transformation used in a single-ratio problem. Also they have proposed an algorithm[8] for maximizing the sum of a linear function and a linear ratio.

The existing methods work for problems with smaller number of ratios. Some of the applications call for methods which can handle a larger number of ratios (upto fifty ratios). Currently such methods are not available. D.Z. Chen et. al.[26] proposed a method to handle larger number of ratios but with fewer number of decision variables.

1.6 Multi Objective Optimization Problem

A variation of the linear programming problem formulation wherein more than one linear objective function is specified. With a multiobjective linear program[47], no prioritization of the multiple objectives is expressed or implied. A key issue for multiobjective linear programs is that there is typically no solution that simultaneously optimizes all objectives. Multiobjective linear programming seeks a list of "non-dominated solutions." A non-dominated solution is one from which it is impossible to improve performance on one objective without some sacrifice in at least one other objective. In these cases the decision makers are looking for the “most preferred” solution. In multi objective optimization, the concept of optimality is replaced with that of efficiency or Pareto optimality. The efficient solutions (or Pareto optimal, non-dominated, non-inferior) are the solutions that cannot be improved in one objective function without deteriorating their performance in at least one of the rest.

1.6.1 Example of Multi Objective Linear Programming Problem

Power generation from energy resources is a simple example for multi objective optimization problem. There are four type of power generation units in a region, namely, lignite fired, oil fired, natural gas fired and renewable energy sources (RES). The characteristics of these units are shown in the following table:
### Power Generation Characteristics

<table>
<thead>
<tr>
<th></th>
<th>Lignite (L)</th>
<th>Oil (O)</th>
<th>Natural Gas (N)</th>
<th>RES (R)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Maximum production per year (GWh)</strong></td>
<td>31000</td>
<td>15000</td>
<td>22000</td>
<td>10000</td>
</tr>
<tr>
<td><strong>Cost of production (Rs./MWh)</strong></td>
<td>30</td>
<td>75</td>
<td>60</td>
<td>90</td>
</tr>
<tr>
<td><strong>CO2 emission coefficient (t/MWh)</strong></td>
<td>1.44</td>
<td>0.72</td>
<td>0.45</td>
<td>0</td>
</tr>
</tbody>
</table>

The yearly demand is 64000 GWh and is characterized by three type of loads: base load (60%), medium load (30%) and peak load (10%). The lignite fired units can be used only for base and middle load, the oil fired units for middle and peak load, the RES units for base and peak load and the natural gas fired units for all type of loads. The endogenous sources are lignite and RES. There are three objective functions: the minimization of production cost, the minimization of CO$_2$ emissions and the minimization of external dependence (i.e. oil and natural gas) and we want to generate the relative efficient solutions of the problem. The multi objective model is as follows:

Minimize $Z_1 = 30L + 75O + 60N + 90R$

Minimize $Z_2 = 1.44L + 0.72O + 0.45N$

Minimize $Z_3 = O + N$

Subject to the constraints:

$L - L_1 - L_2 = 0$

$O - O_2 - O_3 = 0$

$N - N_1 - N_2 - N_3 = 0$

$R - R_1 - R_3 = 0$

$L \leq 31000$

$O \leq 15000$
N <= 22000
R <= 10000
L1 + N1 + R1 >= 38400
L2 + O2 + N2 >= 19200
O3 + N3 + R3 >= 6400

1.6.2 Formulation of Multi Objective Linear Programming Problem

A Linear Programming problem in its general form[2] is shown below

Extremise Objective Functions

\[ Z_1 = c_{11}x_1 + c_{12}x_2 + ... + c_{1n}x_n + c_{10} \]
\[ Z_2 = c_{21}x_1 + c_{22}x_2 + ... + c_{2n}x_n + c_{20} \]
\[ \vdots \]
\[ Z_t = c_{t1}x_1 + c_{t2}x_2 + ... + c_{tn}x_n + c_{t0} \]

Subject to the Constraints

\[ a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n (*)b_1 \]
\[ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n (*)b_2 \]
\[ \vdots \]
\[ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n (*)b_m \]
\[ x_j >= 0 \]

Where,

- \( t \) is the number of objective functions
- \( C_j \) is the cost coefficient of \( X_j \)
- \( X_j \) is a decision variable
- \( a_{ij} \) is a known constant
- \( b_i \) is a constant that determines upper or lower limit of a resource
- \(^* \) stands for \( >= \) or \( = \) or \( <= \) for each constraint
1.7 Overview of Existing Multi-Objective Optimization Methods

In a multi-objective optimization problem, a set of values for the decision variables which optimizes a set of objective function is to be found out. Various methods[47, 63] that are available to solve multi-objective linear programming problems are briefly outlined hereunder.

1.7.1 Pareto Optimum Method

When there are a set of solutions such that we can't improve any objective further without at the same time worsening another then we have what is called the 'Pareto-optimal set' or 'Pareto front' of objective vectors. In such a case all the other lesser solutions are said to be 'dominated' by these better ones and can be discarded. For example, the set of objective values 5,3,4 dominates the set 4,3,4 (the former improves the first objective without reducing the others), the set 4,8,2 doesn't dominate 4,7,3 however, but is an alternative trade off between the last two objectives.

The Pareto optimal set yields an infinite set of solutions, from which the desired solution can be chosen. In most cases, the Pareto optimal set is on the boundary of the feasible region.

1.7.2 Weighting Objectives Method

This method takes each objective function and multiplies it by a "weighting coefficient"[14] which is represented by wi. The modified functions are then added together to obtain a single cost function, which can easily be solved using any single objective method. The multi objective optimization problem with t number of objectives is stated as follows:

\[
\max (w_1 z_1 + w_2 z_2 + \ldots + w_t z_t)
\]
subject to a set of linear constraints and non-negativity restrictions on the variables.

\[ \Sigma w_i = 1 \]

By varying the weights \( w_i \) we obtain different efficient solutions.

1.7.3 Hierarchical Optimization Method

This method ranks the objectives in descending order of importance, from 1 to \( k \). Each objective function is then minimized individually subject to a constraint that does not allow the minimum for the new function to exceed a prescribed fraction of a minimum of the previous function.

1.7.4 Trade-Off Method

This method is also known as the constraint method or the \( \varepsilon \) - constraint method. This method allows us to determine the complete Pareto set of optimal points, but only if all possible values of \( \varepsilon_i \) are used.

1.7.5 Global Criterion Method

This method finds the vector of decision variables which is the minimum of a global criterion. The global criterion is usually defined as seeing how close the user can get to the ideal vector \( \bar{f}^\circ \). The quantity \( \bar{f}^\circ \) is the ideal solution, which is sometimes replaced with a so-called demand level \( \tilde{f} \) that is specified by the user (if the ideal solution is unknown). The scalar objective function for this method is usually written as

\[
f(\bar{x}) = \sum_{i=1}^{ik} \left( \frac{f_i^\circ - f_i(\bar{x})}{f_i^\circ} \right)^p
\]

\( P \) can take any value chosen by the user.
1.7.6 Method of Distance Functions

Another form of the global function is the method of distance function. Here, the distance between the ideal solution and the present solution is minimized. A family of functions are defined. There are two disadvantages to this method:

(1) The ideal solution should be known, otherwise a demand level is assumed.

(2) If the wrong demand level is chosen, the result will be non-Pareto solutions.

The two demand levels would lead to a non-Pareto solution using this method. For this reason it is very important to exercise great care in picking the demand level.

1.7.7 Goal Programming Method

This is perhaps the most well known method of solving Multi-Objective optimization problems. This method was originally developed by Charnes and Cooper (1961) and Ijiri (1965). For this method, the user must construct a set of goals (which may or may not be realistic) that should be obtained (if possible) for the objective functions. The user then assigns weighting factors to rank the goals in order of importance. Finally a single objective function is written as the minimization of the deviations from the above stated goals.

A "goal constraint" is slightly different than a "real constraint" in goal programming problems. A "goal constraint" is a constraint that the user would like to be satisfied, but a slight deviation above or below this constraint is acceptable.
1.7.8 Min-Max Optimum method

If one solves for the optimization of each of the objective functions individually, the min-max optimum is the set of points which will give the smallest values of the relative deviations from the individual objective functions. This optimum assumes that each of the objective functions is equally important. Before the min-max optimum can be defined mathematically, a number of functions must be defined first.

1.7.9 Multiple Linear Fractional Objectives Solution Method

In a research paper[39], the authors present a new technique to compute non-dominated solutions, with an error that can be made as low as the user wants, in multiple objective linear fractional programming (MOLFP), using reference points. The basic idea is to divide, by the approximate ‘middle’, the non-dominate region into two sub-regions and to analyze each of them in order to try to discard one. The process is repeated with the remaining region and it ends when the regions are so little that the differences among their non-dominated solutions are lower than a predefined error. This method was tested with a financial planning problem with 10 objectives, 45 decision variables and 38 constraints.

1.7.10 Multiobjective Problem as Sum of Linear Fractions

In this research work, a multi-objective optimization linear programming problem is converted to a sum of linear fractions[14, 25, 43] as explained in the following paragraphs.

Let $z_1, z_2, z_3, ..., z_t$ be the objective functions and $g_1, g_2, ..., g_m$ be constraints. The problem is solved as a linear fractional programming problem taking different ratio permutations of these objective functions. There will be $t!$ number of orderings. For example, with four objectives ($t=4$), there will be 12 different orderings of objectives in permutation set (PN_set) as depicted below:
The objective fractions are formed as follows:

\[ Z = \frac{z_1}{z_2} + \frac{z_3}{z_4}, \quad Z = \frac{z_1}{z_2} + \frac{z_4}{z_3}, \quad Z = \frac{z_1}{z_3} + \frac{z_2}{z_4} \quad \ldots \text{etc.} \]

1.8 Genetic Algorithm

There is no single optimum solution, for problems with more than one objective function. There exist a number of solutions which are all optimal. In a multi-objective optimization problems many such optimal solutions are important.

Classical way to solve multi-objective optimization problems is to follow the preference-based approach, where a relative preference vector is used to scalarize multiple objectives[24].

One of the most striking differences to classical search and optimization algorithms is that genetic algorithms[3,28] use a population of solution in each iteration, instead of single solution. This ability of genetic algorithm to find multiple optimal solutions in one single simulation[7] run makes genetic algorithms unique in solving multi-objective optimization problems[6,17].

1.9 Motivation

The first international conference with an emphasis on fractional programming by NATO Advanced Study Institute on Generalized Convexity in Optimization and Economics was held in 1980. In single-ratio fractional
programming, the maximization of a ratio of a (nonnegative) concave and a
(positive) convex function is of particular interest in applications. Such a function
is semi-strictly quasi-concave, and hence a local is a global maximum [58]. In the
differentiable case the ratio is a pseudo-concave function. Thus a Krush-Kuhn-
Tucker point is a global maximum. Furthermore a single-ratio concave-convex
fractional program can be related to a concave maximization problem with the
help of the generalized Charnes-Cooper transformation of variables. In a max-min
fractional program, if all ratios are concave-convex, then most of the properties of
single-ratio problems still hold in the multi-ratio case. Hence properties of
concave-convex single-ratio fractional programs essentially extend to multi ratio
concave-convex max-min fractional programs.

Solution methods proposed earlier can handle fewer number of ratios and
decision variables. Some of these have been computationally tested. Typically
execution times grow very rapidly with the number of ratios. Also some of the
applications call for methods which can handle a larger number of ratios. Currently such methods are not available.

The primary objective of any multi-objective optimization method is to
extremize all the objective functions simultaneously with respect to a set of
constraints.

The majority of the classical methods avoid the complexities involved in a
ture multi-objective optimization problem and transform multiple objective
optimization problem into a single objective function by using some user-defined
parameters.

In this research work, an attempt is made to convert a linear multi-objective
optimization problem into linear fractional programming problem. The amount of
profit per unit of investment cost should be considered in finding the objective function value. Fractional programming helps to find such an optimized value.

A new genetic algorithm employing evolutionary principles is proposed to solve linear programming problems with multiple objectives.

In this thesis work, it is aimed

1) to develop a new algorithm to solve linear fractional programming problems with ‘t’ ratios, ‘n’ variables and ‘m’ constraints.

2) to develop a new algorithm by suitably modifying ratio algorithm for single objective optimization to solve multi objective optimization linear fractional programming problems with ‘t’ number of objectives, ‘m’ number of constraints and ‘n’ number of decision variables.

3) to develop an evolutionary algorithm to solve multi-objective optimization problems.

4) to develop software package for the implementation of three algorithms given above.

5) to establish mathematically, the computational efficiency of the proposed algorithms.

Multi-objective optimization has been studied extensively. There exists many algorithms and application case studies involving multiple objectives. Most studies in classical multi-objective optimization do not treat multi-objective optimization any differently than single objective optimization. The studies seem to concentrate on various means of converting multiple objectives into a single objective. Many studies involve comparing different schemes of such conversions, provide reasons in favor of one conversion over another.

The multi-objective optimization method proposed in this thesis work does not convert multi objective problem into single objective problem. The given
objective functions are treated as they are. A new type of transformation known as sum of ratios of objective functions is adopted. The contribution of decision variables to all the objective functions are determined in each iteration of the proposed algorithms. Solution that is extreme to all the objectives are determined. Hence, it approaches the multi-objective optimization linear programming problems in a new pattern and solves them using linear fractional programming concepts.

According to G. Hadley any good procedure to solve a linear programming problem should take a maximum of $m$ iterations to obtain the optimum solution, where $m$ is the total number of constraints. The proposed methods possess this characteristic.

1.10 Conclusion

In this chapter, an overview of existing methods for linear fractional programming, multi objective optimization and genetic algorithm are presented. The drawbacks of earlier methods are stated. The motivation and an overview of proposed research work is outlined. Next chapter presents the details of existing methods for solving multi objective linear programming problems.