Chapter-1

Introduction
Chapter 1

Introduction

The Mathematical Programming became a scientific field in its own, right during late forties and has experienced tremendous growth since then. It is now regarded as one of the most important areas of applied mathematics with extensive applications in engineering, economics and natural sciences. It has become a standard tool for improving business efficiency around the world. It is being used in solving maximum profit, minimum cost, minimum overtime, minimum pollution, maximum production and other similar problems. Major credit for a significant part of continuing improvements in the economic productivity of the industrialized nations, can be attributed to the use of Mathematical Programming.

Linear Programming is a part of mathematical programming. In the linear programming, there is a linear objective function and some linear constraints with non negativity restrictions of variables.

Advanced linear programming problems are some special cases of linear programming. Such as fractional programming problems are Advanced L.P.P. in which objective function is fractional and rest conditions are same as in L.P.P. Multiobjective programming problems in which there are many objectives is also an advanced L.P.P. Travelling salesman problem, in which the non negative variables can take only 0 or 1 values and rest is same as in L.P.P., is also an advanced L.P.P. Transportation problem is also and advanced L.P.P. These advanced L.P.P are more practical than ordinary L.P.P. as for example in university
admissions— one object maximum student acceptance, second object minimum new appointments, third object maximum class occupancy and so on … Similarly in hospital planning, one object minimum staff, second object maximum patient treated, third object maximum bed occupancy ….. Both the cases are of multiple objective programming. And for other instance in university planning we may have the objective function as student—teacher ratio, tenured to non tenured faculty ratio, on the other hand in the hospital planning we may have objective function as cost to bed ratio, nurse to doctor ratio, doctor to patient ratio. These are the cases of fractional programming problems. That is why we are interested in the field of advanced linear programming problems.

In scientific computations, it is found that computer codes on Mathematical Programming and in particular, practically saying linear programming are among the most commonly used softwares. Before a computer can be intelligently employed, however a model must be formulated and good algorithms must be developed. To build a model, one requires to put the subject matter axiomatically.

To this effect, in the thesis, efforts are made to develop algorithms and solution methodologies for various advanced linear programming problems viz. travelling salesman problem, transportation problem, fractional programming problems, multi-objective programming problem.

Brief survey of related problems such as travelling salesman problem, transportation problem fractional programming problem and multi-objective programming problems, is being presented here.

This chapter is divided in two parts, part 1A and 1B, part 1A contains brief survey of the related work and part 1B contains summary of the thesis.
Part 1A Brief Survey of the Related Work.

1A.1 Travelling Salesman Problem (TSP)

Travelling salesman problem is a special case of linear programming, in which the variables can take only 0 and 1 values. In the n city, travelling salesman problem, a salesman starts from home city and wishes to visit other n-1 cities once and only once. And after visiting n-1 cities, he returns to home city again. The distance (cost or time) between the cities has been given to the salesman. In what order should he visit the cities, to minimize the total distance (time, cost) travelled, is the salesman problem that we want to study and is said to be ordinary salesman problem and has been extensively considered in the literature.

The travelling salesman problem can be mathematically stated as

Minimize \[ Z = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \]

\[ \sum_{j=1}^{n} x_{ij} = 1 \quad (j = 1, 2, \ldots, n) \]
\[ \sum_{i=1}^{n} x_{ij} = 1 \quad (i = 1, 2, \ldots, n) \] \[ \ldots \ldots \text{(P 1.1.1)} \]

Subjected to \( x_{ij} = 0 \text{ or } 1 \quad \forall \quad i, j = 1, 2, \ldots, n \)

\[ x_{ij} = \begin{cases} 1 & \text{if he travels from city } i \text{ to } j \\ 0 & \text{otherwise} \end{cases} \]

Here \( d_{ij} \) are defined to be large positive numbers to ensure that all \( x_{ii} (i = 1, 2, \ldots, n) \) are 0.

The constraint

\[ \sum_{i=1}^{n} x_{ij} = 1 \]
Specifies that, starting from any city other than city \( j \), he visits city \( j \), only once. Similarly the constraint.

\[
\sum_{j=1}^{n} x_{ij} = 1
\]

Specifies the restriction that, from city \( i \), he goes to only one of the cities other than \( i \).

1A.1.1 Some types of travelling salesman problem.

1A.1.1 (a) Asymmetric travelling salesman problem [154]

Given \( n \) cities \{1,2,\ldots,n\} and a cost matrix \( (C_{ij}) \) which defines the cost between each pair of cities, the travelling salesman problem (TSP) is to find a minimum-cost tour that visits each city once and returns to the starting city. When the cost \( C_{ij} \) from city "i" to city "j" is not necessarily equal to that cost \( C_{ji} \) from city "j" to city "i". The problem is said asymmetric salesman problems.

1A.1.1 (b) Ordered cluster travelling salesman problem [10]

In this problem a vehicle starting and ending at a given depot must visit a set of \( n \) points. The points are partitioned into \( K \) disjoint clusters so that points of cluster \( K \), are visited before \((K+1)\)th cluster is visited. The vehicle must first visit the points in cluster 1, then the points in cluster 2, and finally the points in cluster \( K \), so that the distance travelled is minimized.

1A.1.1 (c) Acyclic travelling salesman problem [40]

The ordinary salesman problem was defined as, a salesman starts from a depot and travels each station only and only once, to the path of
minimum distance (cost) and returns to depot. This motion path is cyclic path if an additional condition, in the ordinary salesman problem, that the salesman does not return to the depot, is imposed then the travelling salesman problem is said to be acyclic salesman problem.

1A.1.1 (d) **Travelling salesman problem with precedence constraints [26]**

In the travelling salesman problem if there is a condition for the salesman that his tour must follow \( a_i < a_j \) i.e. \( a_i \) city should be visited before \( a_j \) is visited. Then \( a_i < a_j \) is said to be precedence constraint. There may be more than one precedence constraints in a travelling salesman problem.

1A.1.2 **Some definitions related to travelling salesman problem**

1A.1.2(a) **Alphabet table.**

It is the table that is very important in finding tour of the salesman. In this table, columns represent S-D (where S= station number, D= distance) and rows represent station number. We write the distance of a particular station in a row (in front of the station number) in increasing order by using S-D.

<table>
<thead>
<tr>
<th>Station Number</th>
<th>S-D</th>
<th>S-D</th>
<th>S-D</th>
<th>S-D</th>
<th>S-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2-0</td>
<td>3-2</td>
<td>5-3</td>
<td>4-5</td>
<td>1-999</td>
</tr>
<tr>
<td>2</td>
<td>1-0</td>
<td>4-2</td>
<td>5-3</td>
<td>3-4</td>
<td>2-999</td>
</tr>
<tr>
<td>3</td>
<td>2-0</td>
<td>4-0</td>
<td>5-1</td>
<td>1-2</td>
<td>3-999</td>
</tr>
<tr>
<td>4</td>
<td>1-1</td>
<td>3-2</td>
<td>2-3</td>
<td>5-4</td>
<td>4-999</td>
</tr>
<tr>
<td>5</td>
<td>3-0</td>
<td>4-1</td>
<td>2-20</td>
<td>1-27</td>
<td>5-999</td>
</tr>
</tbody>
</table>

So with the help of alphabet table we can find feasible tours and hence optimal tour.
1A.1.2 (b) Lexicographic search [73]

Lexicographic search resembles with the word searching in a dictionary or say lexicographic search is same to the procedure used in searching a word in a dictionary.

Some definitions regarding lexicographic optimization given by Hamacher et al. [73] are given below.

(i) Lexicographically 'better' (smaller) vector: - Let a and b be two vectors in \( \mathbb{R}^n \). \( a \) is said to be lexicographically 'Better' (smaller) than b, is denoted by \( a \prec b \), if and only if for \( r \geq 1 \), we have

\[
a_k = b_k, \quad k = 1, 2, 3 \quad \ldots \ldots \ldots \ldots, \quad r-1
\]

and \( a_r < b_r \),

for example if

\[
a = \begin{pmatrix}
1 \\
-2 \\
5
\end{pmatrix}, \quad b = \begin{pmatrix}
1 \\
1
\end{pmatrix}
\]

then \( a \prec b \) as \( a_1 = b_1 \) and \( a_2 < b_2 \), the third component \( a_3 \) and \( b_3 \) are significant.

(ii) Lexicographic minimum of a set:-

Let S be a set of vectors in \( \mathbb{R}^n \). Then the lexicographic minimum of this set is the vector which is lexicographically 'better' (smaller) than or equal to every other vectors of the set S for example.

\[
S = \begin{pmatrix}
\begin{pmatrix}
1 \\
5
\end{pmatrix}, \quad \begin{pmatrix}
1 \\
2
\end{pmatrix}, \quad \begin{pmatrix}
-1 \\
15
\end{pmatrix}
\end{pmatrix}
\]

\[
\text{Lexmin } S = \begin{pmatrix}
1 \\
-1 \\
15
\end{pmatrix}
\]
1A.1.2 (c) Applications of lexicographic optimization

Lexicographic optimization finds its application in a wide variety of fields of optimization problems and more importantly in transportation problems, assignment problem, travelling salesman problem, multiple objective and goal programming problems. It was used as a 'priority evacuation' in building evacuation models by Hamacher et al [73].

1A.1.2 (d) Path [26]

In the travelling salesman problem the salesman visits from home station to another stations. The path is the linkage between two stations, that salesman visits.

Let salesman starts from 1 then visits, 3 then visits, 2 then visits, 4 and then visits, 5 and then returns back. i.e. 1-3-2-4-5-1. Then any part of this visit is path 1-3 or 3-2, 2-4, 4-5, 5-1.

1A.1.2 (e) Tour [26]

Tour is the whole visit illustration, in which, complete cycle of stations is mentioned i.e. above mentioned cyclic illustration 1-3-2-4-5-1 is a tour.

1A.1.2 (f) Subtour[26]

Subtour is the illustration of the conjunction of the paths travelled by the salesman upto the current station i.e. 1-3-2-4 or 1-3-2.

1A.1.2 (g) Dynamic programming technique

This is the process in which a sequence of interrelated decisions has to be made in different steps, mathematically. A dynamic programming problem is a decision making problem in n-variables, the problem being sub divided into n subproblems (segments). Each subproblem being a decision making problem in one variable only.
1A.1.2 (h) TSP with one job at each station.

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>∞</td>
<td>15</td>
<td>5</td>
<td>22</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>2.</td>
<td>15</td>
<td>∞</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>22</td>
</tr>
<tr>
<td>3.</td>
<td>5</td>
<td>4</td>
<td>∞</td>
<td>1</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>4.</td>
<td>22</td>
<td>3</td>
<td>1</td>
<td>∞</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>5.</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>∞</td>
<td>1</td>
</tr>
<tr>
<td>6.</td>
<td>17</td>
<td>22</td>
<td>12</td>
<td>5</td>
<td>1</td>
<td>∞</td>
</tr>
</tbody>
</table>

1A.1.2 (i) TSP with multiple job facilities at various stations.

<table>
<thead>
<tr>
<th>Job facilities available at various stations</th>
<th>Home stations</th>
<th>A_1</th>
<th>A_2</th>
<th>A_3</th>
<th>A_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Job</td>
<td>-</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>J_1, J_2</td>
<td>A_1</td>
<td>5</td>
<td>--</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>J_1, J_4, J_6</td>
<td>A_2</td>
<td>6</td>
<td>10</td>
<td>--</td>
<td>14</td>
</tr>
<tr>
<td>J_1, J_2, J_3</td>
<td>A_3</td>
<td>7</td>
<td>13</td>
<td>14</td>
<td>--</td>
</tr>
<tr>
<td>J_4, J_5</td>
<td>A_4</td>
<td>8</td>
<td>16</td>
<td>11</td>
<td>8</td>
</tr>
</tbody>
</table>

1A.1.3 Literature surveyed

As the TSP is a special case of programming problems, Dantzing et. al. [49] has solved a large scale salesman problem via 0-1 programming directly. William et.al. [155] presented a relationship between longest path problem and salesman problem.

Later different conditions were imposed on the original salesman problem such as multiple job facilities at various stations. Bansal and
Kumar [17] gave an algorithm to solve TSP with multiple job facilities, using dynamic programming procedure.

Later Shoshana Anily et. al. [136] presented a 5/3 approximation algorithm for ordered clustered salesman problem and it was proved that the algorithm runs in $O(n^3)$ time. In the same paper, they applied the same algorithm for the path version of ordered clustered traveling salesman problem. Previously Arkin et al. [10] presented a 3.5-approximation algorithm of the OCTSP.

O.E. Charles-Owaba [40] gave optimality condition to the acyclic travelling salesman problem in which the salesman does not return to the starting point. Charles O.E. Owaba took n-station acyclic travelling salesman problem consisting of n-1 elements of the distance matrix. He considered lower bounds for elements, on the theoretical basis for identifying and eliminating non-optimal elements. This concept of element elimination was then used to define optimality conditions. Finally an iterative algorithm which converges at the optimal sequence was proposed.

Weixiong Zhang [154] gave a note on the complexity of the asymmetric salesman problem. One of the most efficient approaches known for finding an optimal tour of the asymmetric travelling salesman problem (ATSP) is branch and bound (BnB) subtour elimination. For two decades, expert opinion has been divided over whether the expected complexity of the ATSP under BnB subtour elimination is polynomial or exponential in the number of cities. Wexiong Zhang [154] shown that the argument for polynomial expected complexity does not hold.
Bianco et al. [26] gave exact and heuristic procedures for the salesman problem with precedence constraints, based on dynamic programming. Ravi Kumar M. [120] gave data guided algorithms in combinatorial optimization.

1A.2 Transportation Problem

The transportation problem is one of the sub class of L.P.P's in which the objective is to transport various quantities of a single homogeneous commodity that are initially stored at various origins, to different destinations. To achieve this objective one must know the amount and location of available supplies and quantities demanded.

If the total availability of the goods is equal to total demands required, then this type of transportation problem is said to be balanced transportation problem and if the total availability of the goods is greater to the total demands (requirements) then this type of transportation problem is said to be unbalanced transportation problem.

1A.2.1 Some types of transportation problem

1A.2.1 (a) Cost minimizing transportation problem

(CMTP) In the Transportation problem in which objective function is to minimize the transportation cost is said to be cost minimizing transportation problem. In the CMTP cost per unit from one origin to other destination is given. Mathematically CMTP can be stated as

\[
\begin{align*}
\text{Min.} \quad & Z = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} C_{ij} \\
\text{Subjected to} \quad & \sum_{j=1}^{n} x_{ij} = a_i, i = 1, 2, ..., m \\
& \sum_{i=1}^{m} x_{ij} = b_j, j = 1, 2, ..., n, x_{ij} > 0 \forall i, j \\
& s.t. \quad \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j
\end{align*}
\]

\[\text{........(P 1.2.1)}\]
(i) Flow constrained cost minimizing transportation problem

\[
\begin{align*}
\text{Min.} \quad & Z = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} C_{ij} \\
\text{Subjected to} \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} = P \\
\text{Max} \left( \sum_{i=1}^{m} a_i, \sum_{j=1}^{n} b_j \right) \leq P \leq \text{Min} \left( \sum_{i=1}^{m} a_i, \sum_{j=1}^{n} b_j \right)
\end{align*}
\]

\[\text{--------- (P 1.2.2)}\]

(ii) Cost minimizing transportation problem with mixed constraints.

\[
\begin{align*}
\text{Min} \quad & Z = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} C_{ij} \\
\text{Subjected to} \quad & \sum_{j=1}^{n} x_{ij} \leq a_i, i \in I_1 \\
& = a_i, i \in I_2 \\
& \geq a_i, i \in I_3 \\
& \sum_{i=1}^{m} x_{ij} \leq b_j, j \in J_1 \\
& = b_j, j \in J_2 \\
& \geq b_j, j \in J_3
\end{align*}
\]

\[\text{--------- (P 1.2.3)}\]

where \( U_{i=1}^{3} I_i = I, U_{j=1}^{3} J_j = J \)

\( I_{i_1} \cap I_{i_2} = \phi, \quad \forall i_1, i_2 = 1,2,3, \ i_1 \neq i_2 \)

\( J_{j_1} \cap J_{j_2} = \phi, \quad \forall j_1, j_2 = 1,2,3, \ j_1 \neq j_2 \)

CMTP is very widely studied and has many applications. Appa G.M. [9] studied some useful variants of the CMTP.

1A.2.1 (b) Time minimizing transportation problem (TMTP)

In the time minimizing transportation problem the object of transportation problem is to minimize the maximum time between
different origins and destinations. Mathematically the TMTP can be stated as-

\[
\text{Min } T(X) = \max_{i \in I} \left( \sum_{j \in J} t_{ij} \right)_{x_{ij} > 0}
\]

\[
\sum_{j=1}^{n} x_{ij} = b_j, \quad j = 1,2,...,n
\]

\[
\sum_{i=1}^{n} x_{ij} = a_i, \quad i = 1,2,...,m
\]

\[
x_{ij} > 0
\]

\[\text{ }(P 1.2.4)\]

The variants of TMTP can be stated as defined for CMTP. Time minimizing transportation problem is also called the bottleneck transportation problems. Important studies in (TMTP) have been made by, Bhatia H.L. et. al [24,25]. Gupta A. [66], Ahuja [6], Satya Prakash. [73], Chandra S. et al. [37] have also discussed the lexicographic version of time minimizing transportation problem.

\textbf{1A.2.1 (c) Bulk transportation problem}

A zero one transportation problem is commonly known as bulk transportation problem (BTP). It consists of a set of sources producing a homogenous material with a fixed maximum capacity and a set of destinations whose demand for this material is known and also that this demand is to be met by a single source subject to it's capacity. In the cost minimizing bulk transportation problem (CMBTP) a cost is associated with the shipment of the material from each source to each destination and the problem consists of finding a transportation schedule of flowing the material from the source to the destination that satisfy their demands and minimize the total transportation costs. Herein, the transportation cost is independent of quantity shipped.
Bulk transportation problem (BTP) is a sort of generalized assignment problem (GAP). As BTP and GAP are 0-1 programming problems, these can be solved by the algorithms developed by Balas E. [15]. The utility of these problems is well known in regard of resource allocation problems arising in the areas of production planning and facility location.

1A.2.2 Literature surveyed

Verma Vanita and Puri M.C. [151] gave a branch and bound procedure for CMBTP. Thirwani D. et al. [148] gave an algorithm for such restricted flow transportation problem, having it's objective function of bi-criterion i.e. cost minimizing and time minimizing.

In many problems the appropriate objective is not the sum of the costs of transportation but it is the maximum of the costs of shipments, that are dispatched from the sources to the various destinations or the maximum of the time of transportation from the sources to the destinations. These are known as the bottleneck bulk transportation problem (BBTP). Time cost trade off in bulk transportation problem has been studied by Gupta A. et al. [66].

Bulk transportation problem is a sort of GAP. Min cost-sum version of these problems have many real life applications. For example, it occurs as sub problem in routing problems [58], in fixed charge plant location models in which customer requirements must be satisfied by a single plant, grouping and loading of flexible manufacturing systems, resource scheduling, scheduling of project networks, storage space allocation, in designing communications networks with node capacity constraints. In

Applications of bottleneck (or minimax) version of bulk transportation problem arise in machine loading problems in which the objective function typically represent, job processing time on machine (agents). If job assignments to different machines can be performed in parallel, whereas jobs assigned to each machine can be performed sequentially, it is easy to see that minimizing the time to complete all jobs is equivalent to finding a feasible agents-to-jobs assignments that satisfies the

\[ \min \left( \sum_{j} t_{ij} : \sum_{j} x_{ij} = 1 \right). \]

Bottleneck version also finds it's application in facility lay out involving the assignment of pallet loads to docks, and in many public sector models including the location of emergency public facilities where it is desired to minimize the maximum response (Weighted travel time) to any of the demand paths.

In a standard transportation problem one is interested in minimizing the cost (time) of transporting a homogeneous product produced at, m origins, to n destinations with the given availabilities and requirements, at origins and destinations respectively. But when the problem deals with transporting a set of p different commodities from m origins to n destinations, the multi commodity transportation problem results. Since here three sets of entities have to be matched together to achieve certain objectives so this problem is known as the 3-dimentional transportation problem or solid transportation problem. It was Haley
[69,70] who introduced this 3-D transportation problem, and formulated three special structures of linear programming problem. Haley gave the solution methodology [69,70] for cost minimizing solid transportation problem and also discussed the conversion of an unbalanced problem to a balanced problem. Bhatia et al. in 1976 [24] formulated and developed techniques for the solution of the time minimizing solid transportation problem.

Kim J-U et al. [84] presented heuristic algorithms for a multi period multi-stop transportation planning problem. They considered a multi-period, multi-stop transportation planning problem in a one-warehouse multi-retailer distribution system where a fleet of homogeneous vehicles delivers product from a warehouse to retailers. The objective of the MPMSTP is to minimize the total transportation distance for product delivery over the planning horizon, while satisfying demands of the retailers. G.V. Sharma [130] gave a reduced matrix method to solve the transportation problem. In his paper original cost matrix is reduced to its symmetric transformation by a sequence of operations involving only subtraction and/or addition of positive constants, to all the elements of a row or a column, until a feasible solution is found by making positive assignments to the zero containing cells of the transformed matrix. Dorit S. and Hochbaum [56] gave a linear time algorithm for the bottleneck transportation problem with a fixed number of sources.

1A.3 Fractional Programming Problems

Fractional Programming is an important part of advanced LPP. Fractional programming problem contains the objective function as fractional. It may be linear fraction, linear plus quotient linear, etc.
Linear fractional (ratio) criteria are frequently encountered in finance as illustrated by the following situations.

CORPORATE PLANNING

min (debt-to-equity ratio), max (return on investment), max (output per employee), min (actual cost to standard cost).

BANK BALANCE SHEET MANAGEMENT

min (risk-assets to capital), max (actual capital to required capital), min (foreign loans to total loans), min (residential mortgages to total mortgages)

Besides above, the fractional objectives occur in other areas. As for instance, marine transportation. Instead of minimizing profit from a given unit voyage, a more relevant measure is profit divided by the duration of the unit voyage. In water resources, we may wish to minimize water temperature elevations in a river due to the cooling of power generation plants in the basin. The objective would then be to minimize the BTUs to be dissipated, divided by the volume of flow. In health care, we may have cost to bed, nurse-to-doctor, and doctor to patient ratios. In university planning, we may have student teacher ratios, tenured to no tenured faculty ratios, and so forth.

1A.3.1 Types of linear fractional programming problems

1A.3.1 (a) Single objective linear fractional programming

Let us consider the single objective linear fractional program

\[
\begin{align*}
\text{Max} & \quad \left\{ \frac{c^T x + \alpha}{d^T x + \beta} = Z \right\} \\
\text{s.t.} & \quad x \in S = \{ x \in \mathbb{R}^n \mid Ax = b, \; x \geq 0, \; b \in \mathbb{R}^m \}
\end{align*}
\]

...........(P 1.3.1)

in which the objective function's denominator is positive everywhere in S.
1A.3.1 (b) Multiple objective linear fractional programming problems

\[
\begin{align*}
\text{max} & \quad \left\{ \frac{c^T_j x + \alpha_j}{d^T_j x + \beta_j} = Z_j \right\} \\
\text{max} & \quad \left\{ \frac{c^T_2 x + \alpha_2}{d^T_2 x + \beta_2} = Z_2 \right\} \\
\vdots & \quad \vdots \\
\text{max} & \quad \left\{ \frac{c^T_k x + \alpha_k}{d^T_k x + \beta_k} = Z_k \right\} \\
\text{s.t.} & \quad x \in \mathbb{R}^n, \ Ax = b, \ x \geq 0, \ b \in \mathbb{R}^n \\
\text{...........(P 1.3.2)}
\end{align*}
\]

It is customary to assume that the MOLFP is well posed in the sense that all denominators are positive everywhere in S.

The general linear fractional programming problem is of the form

\[
\begin{align*}
\text{maximize} & \quad Z = \frac{Z_1}{Z_2} = \frac{\sum_{j=1}^{n} c^T_j x_j + c_0}{\sum_{j=1}^{n} d^T_j x_j + d_0} \\
\text{Subject to} & \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i : i = 1, 2, \ldots, m \\
& \quad x_j \geq 0 : j = 1, 2, \ldots, n \\
\text{...........(P 1.3.3)}
\end{align*}
\]

1A.3.2. Literature surveyed

Various methods are available to solve these kind of problems.

Some of them are

(i) Charnes-Cooper method
(ii) Wolfe's parametric method
(iii) Kanti-Swarup method
(iv) Ratio algorithm
Among the methods mentioned above, Charnes-Cooper method, Kanti Swarup method and Ratio Algorithm are amenable for computerisation.

In Charnes-Cooper method [41] the fractional objective function is converted into a linear function by substituting \( y = tx \) where \( t \) is the reciprocal of the denominator and thus the problem is converted into a linear programming problem. The solution is obtained by solving the two linear programming problems derived on the basis of, sign of \( t \).

In Kanti-Swarup method [82] a new row is formed from numerator and denominator coefficients of the objective function and this row is used to select the entering variable. The selection of leaving variable and the iterations are the same as in simplex method. It is assumed that the denominator value is always positive for all feasible solutions.

In ratio algorithm [127] an entering variable is selected by considering the ratios of the contribution coefficients of the variables both in the numerator and denominator of the objective function. And the selection of leaving variable as well as the iterations are the same as in revised simplex method. The optimality of the solution is checked by using the objective function value and the ratios of the contribution coefficients of the variables. Here also, it is assumed that the denominator value of the objective function is always positive for all feasible solutions.

In the above methods, only one variable is selected to enter into the basis at each simplex iteration. Ranger Frisch [119] attempted to select two variables to enter into the basis for the solution of linear programming problems and he concluded that the method was
unsuccessful. Multiplex Algorithm based on multiple column selection method [126,128] had been proposed by S. Sakthivel in order to solve large scale linear programming problems. Nagunathan E. R. and S. S [106] proposed the ratio multiplex algorithm, which brings more than one variable into the basis in each pass, for solving linear fractional programming problems. This algorithm attempts to bring into the basis only such variables which are restricted to leave the basis until optimal solution is obtained.

Above are the some important and basic methods to solve fractional linear programming problems. Besides this, lot of work has been done in fractional linear programming and the author has surveyed extensive literature.


1A.4 Multiobjective Programming

Multiple objective programming is sort of advanced cases of linear programming. The multiobjective programming can be solved by either maximizing (minimizing) objects or giving some goals to each objective. The discussion about this is given as following.

1A.4.1 Multiple objective programming

In the multiple objective programming there are more than one objective functions. But it is rarely possible to optimize all the objective functions simultaneously. It is worth noting that multiple objectives also arise when several individuals have to make a joint decision. The objective functions modeling is the aspirations of each individual. For example, in the production problem, the three objectives may be those of the managing director the sales director and the export director. And similarly there are other examples.

In general we will examine problems having p linear objective functions \( \sum C_kx_j \) for \( k=1,2,\ldots,p \), which we wish to optimize, subject to linear constraints. Typically, these objectives will conflict in that there is no feasible solution which simultaneously maximizes them all. In such a case, some form of conflict resolution must be adopted to arrive at a
solution. One of the simplest and most frequently used approaches will also play an important role in our development of efficiency and the idea is to multiply each objective by a weighting factor and then add the weighted objectives e.g.

Let the MOP is
\[
\begin{align*}
\max & \sum_{j} C_{1j} x_j \\
\max & \sum_{j} C_{2j} x_j \\
\vdots & \\
\vdots & \\
\max & \sum_{j} C_{nj} x_j
\end{align*}
\]
\[\quad s.t. \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \forall i = 1,2,\ldots,m \\
\quad x_j \geq 0 \quad \forall j = 1,2,\ldots,n \]

then by using above strategy the above MOP may be written as

\[
\begin{align*}
\text{LP} \ (W_1 \ldots W_p) : \quad \text{maximize} & \quad \sum_{j} \sum_{k} (W_k C_{kj}) x_k \\
\quad & \sum_{j} a_{ij} x_j \leq b_i \forall i = 1,2,\ldots,m \\
\quad & x_j \geq 0 \forall j = 1,2,\ldots,n
\end{align*}
\]

Where \( W_k \) are the weights of the \( K_{th} \) objective functions.

**1A.4.2 Goal programming**

In the multiple objective programming we discussed that there are many objective functions in MOP and it is not possible to optimize all the objective functions simultaneously, so we defined weighted objective functions in which weight of each objective function represents it's importance. We are to make attention in optimizing the MOP that the order of weights is their priority order.
Actually decision maker's aims are often not to maximize profits (or sales etc.) but to achieve a satisfactory level (or profits). This involves specifying a goal or target value for the objective function. The decision-maker tries to maximize the objective function up to the goal value but is not interested in values exceeding the goal.

In goal programming there are two basic models: The Archimedean model and the Preemptive model. With the Archimedean model, we generate candidate space. In Preemptive (lexicographic) goal programming, the goals are grouped according to priorities. The goals at the highest priority level are considered to be infinitely more important than goals at the second priority and the goals at second priority level are considered to be infinitely more important than goals at the third priority level, and so on.

1A.4.3 Some definitions related to multi objective programming

1A.4.3 (a) Efficient solutions

In the multiobjective programming problem if a feasible solution \((x_1, x_2, \ldots, x_n)\) dominates \((y_1, y_2, \ldots, y_n)\) in the sense that

\[
\sum_j C_{kj} x_j \geq \sum_j C_{kj} y_j \quad \forall k
\]

\[
\text{and} \sum_j C_{kj} x_j > \sum_j C_{kj} y_j, \text{for at least one } k
\]

then we would prefer \((x_1, x_2, \ldots, x_n)\) to \((y_1, y_2, \ldots, y_n)\)

we will say that a feasible solution is efficient if it is dominated by no other feasible solution. Synonyms for 'efficient' include 'admissible', 'Pareto Optimal'.

1A.4.3 (b) Vector maximization problem

When we are studying a multiple objective problem with a view to calculate pareto optimal points, we speak the problem as vector maximization problem (VMP)
Lot of work has been done in the multiple objective programming, the author has surveyed the following literature in the related field.

**1A.4.4 Literature surveyed**

Part 1B Summary of the Thesis

The thesis contains following five chapters.

Chapter 1 : Introduction

Chapter 2 : Single Travelling Salesman Problems: A Lexi Search Approach

Chapter 3 : Lexicographic Optimization of Multiple Travelling Salesman Problem with a truncation

Chapter 4 : Lexicographic Version of Multicommodity Transportation Problem

Chapter 5 : Solution of Linear plus Linear Fractional Objective Functional Programming Problems

First chapter Comprises a comprehensive review of some advanced linear programming problems including introduction, types and related important definitions of advanced linear programming problems and summary of thesis.

Second chapter and Third chapter focused on the application of the lexi-search approach for single and multiple salesman problem. Lexi-search is an implicit enumeration technique applicable in solving a wide class of zero-one discrete programming problems like assignment problem, travelling salesman problem etc. In this technique it is possible to list all the solutions in a structural hierarchy of their values and are treated as words. Words are generated systematically in the decreasing order of their contribution to the objective function.

Second chapter has been divided in two parts. Part first 2A contains travelling salseman problem (TSP) with multiple job facilities
and precedence constraints. In this problem there are \( m \) jobs and \( n \) stations. The distance between each pair of stations and facilities of jobs at each station are known. And we have some precedence constraints \( i_r < j_r : r = 1, 2, \ldots, k \), that is the station \( j_r \) should not be visited unless station \( i_r \) is already visited. A salesman starts from the initial home station and returns to it after completing all the jobs. The salesman in his tour may or may not visit all the stations and should not visit a station more than once, provided no job (s) available at the home station. In the tour the salesman travels according to precedence constraints. The author has made a lexi search algorithm to solve this problem. And has given numerical illustration to implement the algorithm.

Mathematically the problem discussed in 2A may be stated as

\[
\text{Minimize} \quad Z = \sum_i \sum_j d(A_i, A_j)x(A_i, A_j); i, j = 1, 2, \ldots, (n-1) = \sum_i \sum_j d_{ij} x_{ij}
\]

(For simplicity we write \( d(A_i, A_j) = d_{ij} \) and \( x(A_i, A_j) = x_{ij} \))

subject to

\[
\sum_{j=1}^{n-1} x_{1j} = 1, \quad \sum_{j=1}^{n-1} x_{ji} = 1,
\]

(Since the salesman starts from a depot say 1 and goes back to it) \((P\ 2A.1.1)\)

\[
\sum_{i \neq k} x_{ik} = \sum_{k \neq j} x_{kj} = 1
\]

subjected to

\[
i_r < j_r : r = 1, 2, \ldots, k
\]

25
The unwanted sub-tours are eliminated by lexicographic search procedure.

In part 2B the author has presented one generalization of travelling salesman problem, called the 'Time - Dependent Travelling Salesman Problem' which is different from the usual travelling salesman problem. In this problem the third - dimension of the travelling salesman problem represents time (some facility). In part 2B has the following time - dependent travelling salesman problem.

"There are n cities and N = [1,2,.......n]. The cost array C (i, j, k) is the cost of a salesman visiting from city i to city j at time (facility) k, is known (i,j,k =1,2,.........,n). The restriction for the travelling salesman problem is that at a point of time he should not visit more than one pair of cities in his 'tour'. The problem is to find a feasible tour satisfying the above conditions such that the total tour cost is minimum".

The problem discussed in 2B can be mathematically stated as:

\[
\text{Minimize } Z = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} C(i,j,k) X(i,j,k)
\]

Subject to

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} X(i,j,k) = 1 \quad k=1,2,...........,n \quad \ldots \ldots \text{(P 2B.1.1)}
\]

\[
\sum_{j=1}^{n} \sum_{k=1}^{n} X(i,j,k) = 1 \quad i=1,2,...........,n
\]

\[
\sum_{i=1}^{n} \sum_{k=1}^{n} X(i,j,k) = 1 \quad j=1,2,...........,n
\]

\[
X(i, j, k) = 1 \quad \text{or} \quad 0, \text{represents the salesmen's visit, when salesman visits city } j \text{ from city } i \text{ at time } k \text{ or otherwise.}
\]
The author has proposed an algorithm, which is based on lexicographic search. For this an alphabet table is made in order to Lexicographic search. In the table the cost $C(i,j,k)$ is written in ascending order giving corresponding $i$, $j$ and $k$. A numerical illustration is also given at the end of the chapter. The main motivating force for the present study was to find a lexi-search algorithm to the three dimensional - TSP. Lexi-search process resembles with the word searching in dictionary. Thus the author has made an application of lexicographic search.

**Third chapter** comprises multiple travelling salesman problem with a truncation. In this chapter lexi search approach is used for multiple travelling salesman problem. So the author has developed lexi search algorithm which takes care of the simple combinatorial structure of the problem, for finding optimal solution of truncated multiple travelling salesman problem.

Truncated multiple travelling salesman problem is as follows. In this problem there are $n$ cities $N = (1,2,\ldots,n)$, the distance $d(i,j)$ between any pair of cities $(i,j)$ is known. A subset with $n_0$ cities of $n$ cities has to be traveled by the M-Salesmen. The number of cities to be traveled by each salesman 'i' is $n_i$ cities with $\sum n_i = n_0$. A salesman has to visit only given number of cities in his tour. The problem is to find a feasible M-tour for $n_0$ cities for the M-salesmen with minimum total length.

The selection of $n_0$ cities from $n$ cities is the truncation. In multiple travelling salesman problem there are $n_c^{n_0}$ possibilities in selecting $n_0$ cities from $n$ cities. This problem can also be solved by solving M-
travelling salesman problem, in all $n_{c_n}$ possibilities. But for higher values of $n$ and $n_0$ it is difficult to solve $n_{c_n}$ problems, hence not advisable.

"Multiple travelling salesman problem with a truncation" can be stated as:

\[
\begin{align*}
\text{Minimize} \quad Z &= \sum_{i=1}^{n} \sum_{j=1}^{n} d(i, j) x(i, j) \\
\text{Subject} \quad \sum_{i=1}^{n} x(i, j) \Phi(j) &= 0 \quad \text{or} \quad 1, j = 1, 2, \ldots, n \\
\sum_{j=1}^{n} x(i, j) \Phi(i) &= 0 \quad \text{or} \quad 1, i = 1, 2, \ldots, n \\
\sum_{i=1}^{n} \Phi(i) &= n_0 = \sum_{j=1}^{n} \Phi(j) \\
\sum_{j=1}^{m} MS(i) &= n_0, \quad n_0 < n
\end{align*}
\]

Where $x(i, j) = 0$ or $1$ in which $x(i, j) = 1$ indicates that a salesman visits $j$ from city $i$ and if there is no such trip it is indicated by $x(i, j) = 0$. And similarly $\Phi(i) = 0$ or $1$ in which $\Phi(i) = 1$ indicates that the $i^{th}$ city is traveled by a salesman otherwise $\Phi(i) = 0$.

$MS(i)$ and $d(i, j)$ are given, where $MS(i)$ is the number of cities traveled by the $i^{th}$ salesman.

In fourth chapter procedure is developed to find a lexicographic optimal solution of Lexicographic General Optimization Problem (LGOP) for multicommodity transportation problem.

The LGOP can be stated as

\[
\text{LGOP} : \text{Lex } \min_{X \in S} F(X)
\]
Where $S \subset \mathbb{R}^n$ be the given set of the possible solutions of the optimization problem, and $F : S \rightarrow \mathbb{R}^p$ be a p-dimensional real valued vector function. Let $f_k : S \rightarrow \mathbb{R}$ be the $k^{th}$ component of $F$.

The lexicographic general optimization problem (LGOP) is one in which in addition to minimizing $f_1$, one is interested in minimizing $f_2$, if $f_1$ is as small as possible and one is interested in minimizing $f_3$, if $f_1$ and $f_2$ are as small as possible and so on.

The multicommodity transportation problem (MCTP) deals with the minimization of cost of transporting certain types of goods from various origins to various destinations. Thus, if there are $m$ origins, $n$ destinations and $s$ types of goods available at each origin with $A_{jl}$ the requirement of the $l^{th}$ type of goods at the $j^{th}$ destination, $B_{il}$ the availability of the $l^{th}$ type of goods at the $i^{th}$ origin and $E_{ij}$ the total amount of goods to be sent from the $i^{th}$ origin to the $j^{th}$ destination.

In lexicographic version of (MCTP) the objective is to send the quantity as little as possible not only on the costliest routes but also on the second costliest routes (if the shipment on costliest route is as small as possible), the third costliest routes (if the shipment on first two costliest routes is as small as possible) and so on.

Thus, lexicographic version of (MCTP) can be stated as:

$$(LMTP): \min_{x \in S} \left[ \sum_{i=1}^{m} \sum_{(i,j) \in N_i} \sum_{l=1}^{n} x_{ijl}, \sum_{(i,j) \in N_j} \sum_{l=1}^{n} x_{ijl}, \ldots, \sum_{(i,j) \in N_p} \sum_{l=1}^{n} x_{ijl} \right]$$

\[
\begin{align*}
\sum_{i=1}^{m} x_{ijl} &= A_{jl} \quad , \quad j = 1,2,\ldots,n; \quad l = 1,2,\ldots,s \\
\sum_{j=1}^{n} x_{ijl} &= B_{il} \quad , \quad l = 1,2,\ldots,s; \quad i = 1,2,\ldots,m \\
\sum_{l=1}^{s} x_{ijl} &= E_{ij} \quad , \quad i = 1,2,\ldots,m; \quad j = 1,2,\ldots,n \\
x_{ijl} &\geq 0
\end{align*}
\]

.........(P 4.1.1)
where, for feasibility the following conditions are necessary.

\[ \sum_{j=1}^{n} A_{ji} = \sum_{i=1}^{m} B_{ji} \quad \sum_{i=1}^{m} B_{ji} = \sum_{j=1}^{n} E_{ij} \quad \sum_{j=1}^{n} E_{ij} = \sum_{i=1}^{m} A_{ji} \]

\[ \sum_{i} \sum_{j} A_{ji} = \sum_{i} \sum_{j} B_{ji} = \sum_{i} \sum_{j} E_{ij} \]

Let \( C_1, C_2, C_3 \ldots \) be the first, second, third highest cost of transportation i.e., \( C_1 = \max_{(ijl) \in I \times J \times L} \{c_{ijl}\} \), where \( I \) is the index set of \( m \) origins, \( J \) be the index set of \( n \) destinations and \( L \) be the index set of \( s \) goods. Then \( C_k \)'s will be \( C_k = \max \{c_{ijl} : c_{ijl} < C_{k-1}, (i,j,l) \in I \times J \times L\} \).

And let \( \hat{N} = \{i,j,l : (i,j,l) \in I \times J \times L\} \). Then the above mentioned \( \hat{N}_k \) are found by participating \( \hat{N} \) into \( \hat{p} \) disjoint subsets \( \hat{N}_k = \{(i,j,l) : (i,j,l) \in I \times J \times L, c_{ijl} = C_k\}, k = 1,2,\ldots, \hat{p}. \) It is being assumed, without loss of generality, that \( c_{ijl} > 0 \forall (i,j,l) \in I \times J \times L \).

To solve MCTP the author has developed a solution methodology, that give lexicographic optimal solution of the problem. Numerical illustration of Multi commodity transportation problem is also given, at the end of the fourth chapter.

Fifth chapter is focused on optimization of linear plus linear fractional programming problems of the type (T).

\[
\begin{align*}
\max \text{ (min)} \quad & L(X) + \frac{1}{M(X)} \\
\text{Subjected to:} \quad & A \ X \leq b \\
& X \geq 0
\end{align*}
\]

Where \( L(X) \) and \( M(X) \) are linear. \( A \) is a matrix and \( b \) is a vector.
This chapter is further divided in two parts (5A) and (5B). Part (5A) comprises a linear plus linear fractional programming problem with homogeneous constraints.

The problem discussed in part (5A) is

\[
\text{Maximize } Z = (c^T X + p) + \frac{1}{(d^T X + q)} \quad \ldots \ldots (5.1)
\]

subject to

\[
AX = b \quad \ldots ... (5.2) \quad \ldots ...(P5A.1.1)
\]

\[
X \geq 0
\]

with \( a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{in} x_n = 0 \) for some \( i \), in \( AX = b \)

Where \( c \) and \( d \) are \( nx1 \) vectors and \( X \) is an \( nx1 \) vector, \( b \) is an \( mx1 \) vector, where \( X^T \) is the transpose of \( X \).

In this part the intention is to reduce the computing time of the optimization process when a block of constraints are homogenous in nature. The method seems to be beneficial to large class of programming models containing a great number of homogeneous constraints. To reduce constraints a transformation matrix \( T \), which eliminates the homogenous constraints from further consideration, is developed in part (5A.1) part (5A.2) presents the desired relationship between the original problem and the transformed problem developed by using transformation matrix \( T \).

And part (5B) comprises a vector maximisation of the following multiobjective programming problem.

\[
\begin{align*}
\text{Min. } f_1 (X) &= (c_1^T X + p_1) + \frac{1}{(d_1^T X + q_1)} \\
\text{Min. } f_2 (X) &= (c_2^T X + p_2) + \frac{1}{(d_2^T X + q_2)} \\
& \vdots \\
\text{Min. } f_r (X) &= (c_r^T X + p_r) + \frac{1}{(d_r^T X + q_r)} \\
\text{s.t. } AX &\leq b \\
X &\geq 0
\end{align*}
\]

\[\ldots...(P 5B.1.1)\]
Where $A$ is an $m \times n$ matrix, $b$ an $m \times 1$ column vector; $X$ is an $n \times 1$ vector of decision variables, $c_i$ and $d_i$ for all $i = 1, 2, \ldots, r$ are $n \times 1$ vectors. Also $p_1, p_2, \ldots, p_r$ and $q_1, q_2, \ldots, q_r$ are scalar constants.

The author has shown that a pareto optimal solution of this problem can be obtained through a pareto optimal solution of a fractional programming problem with linear fractional objective functions and convex constraints.

References used in thesis are also given at the end of the thesis.

****