SOLUTION PROCEDURES FOR SOME MULTIOBJECTIVE NETWORK PROBLEMS

by

A.TARIK MENLİOĞLU

B.S.in I.E. Boğaziçi University, 1978

Submitted to the Faculty of Engineering
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
INDUSTRIAL ENGINEERING

Boğaziçi University
September 1981

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor Dr.Gündüz Ulusoy for his invaluable guidance and patience for the long discussions throughout the course of this study. I am also deeply grateful to Dr. Gülseren Kızıltan for her generous help. I would like to thank Dr.İlhan Or for his comments and suggestions on the composition of the final text.

I am also pleased to express my thanks to Davit Pinhas for his sincere help and suggestions.

My special thanks are due to the members of the Operations Research Division of Marmara Scientific and Industrial Research Institute for their encouragement and understanding.

ABSTRACT

In this thesis, solution procedures for multiobjective linear network problems and bicriteria fixed charge network problems are presented.

A labeling algorithm for multiobjective linear network simplex method is developed by incorporating a method of multiobjective LP and extended to the case with lower and upper bounds on the arc flows.

A branch and bound algorithm is given for the bicriteria fixed charge problem in order to generate the efficient extreme points.

ÖZET

Bu tezde, çok amaçlı doğrusal serim problemleri ve iki amaçlı değişmez maliyetli serim problemleri için çözüm yöntemleri önerilmektedir.

Çok amaçlı doğrusal serim problemlerinin baskın uç noktalarını serim simpleks yöntemi kullanarak bulan bir algoritma verilmiştir. Aynı algoritma ayrıt üzerinde alt ve üst sınırlar olduğu durum için de genişletilmiştir.

İki amaçlı değişmez maliyetli problemlerin baskın uç noktalarının bulunması için bir dallandırma sınırlandırma algoritması önerilmektedir.

TABLE OF CONTENTS

CHAPTER I
INTRODUCTION

Most of the decision making problems are multiple objective in nature. In some of the decision problems an important objective can be selected as the single objective to be optimized, however in most cases there may be more than one conflicting and noncommensurable objectives. Especially, in public decision making problems, the decision maker has to make the best decision based on the general duties and objectives. The models developed by using a single objective can be optimized by the powerful techniques of optimization, but may fall short of expressing the real decision problem. When the other objectives which are as important as the objective considered are incorporated, a more expressive model of the decision problem may be formulated but it requires somewhat different solution techniques than the traditional ones.

During the last decade, mathematicians, operations researchers, and researchers in related fields have contributed extensively to research in multiple objective optimization.

The solution to a multiobjective optimization problem, generally, is not a single point but a set of efficient points. The set of efficient points is defined as the set of points where an improvement in one of the objectives can only be gained at the expense of at least one of the other objectives.

The methods developed for finding the preferred solution fall into two classes: (i) finding the efficient solution set and then selecting the preferred solution according to the preferences of the decision maker, (ii) taking the preferences of the decision maker into account through the use of an interactive procedure.

The methods which have been developed for multiobjective analysis have found applications mostly in strategical planning problems such as sectoral economy planning, energy planning, water resources planning, pipeline network planning, facility location planning and firm planning [5,21].

Network problems form a class of the most established problems of operations research. Maximum flow, minimum cost flow and the shortest path problems are among the best known network problems. There may be more complex decision problems which may require simultaneous consideration of maximization of flow with minimization of one or more measures of cost as multiple objectives. These three network problems can be formulated as LP problems, but there are more efficient algorithms for solving such problems than the general simplex method[3]. These algorithms have been developed based on the speicial structure of the network problems One such algorithm is the network simplex algorithm.

In this thesis, solution procedures for generating the set of efficient extreme points for the multiobjective linear network problems and multiobjective fixed charge network problems are analyzed. Throughout the thesis, the enumeration of only the efficient extreme points is attempted.

In chapter 2, a survey of multiobjective optimization, especially, multiobjective linear programming in some detail is made, some major network problems are stated and reports of research and applications on multiobjective network problems are reviewed. Then some possible applications of the models tackled in the third and fourth chapters are proposed.

In chapter 3, a labeling algorithm for multiobjective linear network flow problems is given. This algorithm is a modified labeling algorithm of network simplex method which includes the efficiency check procedure and the concept of monotone connectedness properity of the set of efficient extreme points developed recently[19]. In order to incorporate the maximum flow objective, the algorithm is extended to handle the upper and lower bounds on the arc flows.

In chapter 4, a branch and bound algorithm is given to handle the bicriteria fixed charge network problem. The algorithm is based on the branch and bound algorithm for the fixed charge problem developed in [27]. The algorithm finds the set of efficient extreme points when the fixed charge objective and the linear objective are handled as separate objectives. It is possible to extend this solution procedure to the case with more than two objectives.

CHAPTER II
LITERATURE SURVEY

II.1. MULTIPLE OBJECTIVE OPTIMIZATION

II.1.1. INTRODUCTION

In the literature, generally, the multiple criteria optimization problem has been formulated as a vector maximization problem and all of the efficiency check procedures are given with respect to the maximization problem. Although the problems considered in this thesis are minimization problems, to be consistent with the literature, the survey will be developed on the maximization problem.

The general multiobjective decision making problem may be formulated as follows:

$$\text{(II.1.1.1)} \qquad \begin{array}{l} \text{Maximize } F(x) \\ \quad \text{s.t.} \\ \qquad x \in X = \{x \in R^n;\ G(x) \leqq 0\} \end{array}$$

F(x) is an ℓ-vector of objectives and G(x) is also a vector valued function of dimension m.

Methods involving multiple objective decision making problems range from finding the "best" (i.e the most preferred) solution by using the known utility function of the decision maker, to finding the set of all efficient solutions without having a priori preference weights of the decision maker. There are methods in between these extremes being closer to one end such as progressive determination of the preferences of the decision maker by interactive methods[32] and generating a relevant subset of efficient solutions corresponding to the preference interval specified by the decision maker[19,28].

However, in this thesis the methods of finding the set of all efficient extreme points without having a priori preferences of the decision maker will be examined. Therefore, the solution to the multiobjective optimization problem is defined to be the set of efficient points. The efficient solution is defined as follows:

<u>Definition</u> II.1.1. The vector $x^o \varepsilon X$ is efficient if there exists no $x \varepsilon X$ such that $F(x) \geq F(x^o)$*.

Two basic approaches have been developed for the solution of the problem stated above. One is the parametric approach by which, if the objective space is convex, the set of all efficient, solutions can be found by solving the

---

* The convention employed using inequalities is as below: Let $x=(x_1,\ldots,x_n)$ and $y=(y_1,\ldots,y_n)$ then

i) $x=y$ if and only if $x_j=y_j$ for all $j=1,\ldots,n$.

ii) $x \geqq y$ if and only if $x_j \geqq y_j$ for all $j=1,\ldots,n$

iii) $x \geq y$ if and only if $x_j \geqq y_j$ for all $j=1,\ldots,n$ and $x \neq y$.

parametric optimization problem[6].

$$
\text{(II.1.1.2)} \qquad \begin{array}{ll} \text{Max} & \lambda^T F(x) \\ \text{s.t} & \\ & x \in X \\ & \lambda \geq 0 \end{array}
$$

The other approach is the constraint approach where one of the objectives is taken as a primary objective and the others are added into the constraint set to be satisfied at the values which have been specified[16]. The constraint approach problem is formulated as:

$$
\text{(II.1.1.3)} \qquad \begin{array}{lll} \text{Max} & f_1(x) & \\ \text{s.t} & & \\ & x \in X & \\ & f_j\ (x) \geq \alpha_j & j=2,\ldots,\ell \end{array}
$$

Where the right hand side α_j will be changed parametrically. Here the solutions to the problem with all nonzero lagrange variables associated with constraints corresponding to the objectives are efficient. The constraint approach has an advantage over the parametric approach in generating the efficient solution set even when the objective space is nonconvex, where the parametric approach fails to generate all of the efficient points.

II.1.2. MULTIOBJECTIVE LINEAR PROGRAMMING

A multiobjective linear programming (MOLP) problem is formulated as follows

$$\text{(II.1.2.1)} \qquad \text{Max} \begin{bmatrix} C^1 & x \\ \vdots & \\ C^\ell & x \end{bmatrix}$$

$$x \in X = \{x \mid Ax \leq b, \ x \geq 0\}$$

where C^i is an row vector of dimension n being the coefficients of ith objective function. A is an mxn matrix of the coefficients of the constraints, b is an m-vector of the right hand sides and x is an n-vector of the decision variables. Since the generation of all efficient extreme points is aimed in this thesis, only the generating approaches related with the efficient extreme points will be surveyed.

The methods for generating the efficient extreme points of a MOLP problem are based on the theory of linear programming, the simplex method and the parametric linear programming.

The set of efficient extreme points is a connected set that is, there is either only one point in the set or there is path of adjacent efficient extreme points between any two points in the set. Therefore, it is possible to enumerate all efficient extreme points by starting at an efficient extreme point and going through only the efficient extreme points.

The methods reported in the literature differ in the efficiency check procedures and in the enumerating paths through the efficient vertices of the convex polyhedron. Several authors have proposed procedures for the efficiency check and enumeration of the efficient extreme points. A comparative survey of these procedures including the generation of the non extreme efficient points is given in Kızıltan[19]. Here, some of these procedures will be briefly mentioned and two of them will be discussed in more detail.

The first efficiency check subproblems are proposed by Philip[24]. He gives two different LP subproblems to check the efficiency of a given point. He also indicates how to find another efficient extreme point once an initial efficient point is found. Steuer and Evans[11] give a revised simplex method for MOLP problems. Ecker and Kouada[8] give a method for finding all eficient extreme points by determination of the efficiency of an edge incident to an efficient extreme point.

Zeleny[31] gives an efficiency check subproblem. The basic feasible solution x^o is efficient if and only if the maximum objective value of the subproblem below is zero.

$$\text{(II.1.2.2)} \qquad \begin{array}{l} \text{Max} \quad e^T s \\ \text{s.t} \\ \quad x \in \bar{X} = \{(x,s) \mid x \in X,\ Cx - s \geqq Cx^o,\ s \geqq 0\} \end{array}$$

Since by the definition of efficiency there can not be $x \in X$ such that $Cx \geq Cx^o$.

Starting at an efficient extreme point, this subproblem is solved for all adjacent extreme points which are not obviously dominated and the adjacent efficient extreme points are determined. Then going to one of them which is not explored already and performing the efficiency check procedure for all adjacent extreme points to that extreme point and repeating this process all efficient extreme points are enumerated.

Isermann [17] states the equivalence of multiobjective linear programming problem (II.1.2.1) and the linear multiparametric problem below.

$$\text{(II.1.2.3)} \qquad \begin{aligned} &\text{Max} \quad \lambda^T Cx \\ &\text{s.t} \\ &\qquad x \in X \\ &\qquad \lambda > 0 \end{aligned}$$

<u>Theorem:</u> II.1.2.1. x^o is an efficient point for the multiobjective linear problem (II.1.2.1) if and only if there exists a $\lambda > 0$ such that x^o is an optimal solution for the multiparametric linear problem (II.1.2.3). Isermann gives the following adjacency definition for the efficient extreme points.

<u>Definition:</u> II.1.2.1. Let x' and x'' be efficient basic solutions. x' and x'' are said to be adjacent if and only if,

(i) x' and x'' have (m-1) basic variables in common (i.e they are adjacent vertices) and

(ii) each $\bar{x} = \alpha x' + (1-\alpha)x''$, $0 \leq \alpha \leq 1$ is efficient.

Furthermore, he gives the definition of dual feasibility as follows:

Definition: II.1.2.2. An efficient basic feasible solution x' is said to be dual feasible if and only if the system

$$R^T\lambda \leq 0, \ \lambda > 0$$

has no solution, where R is the reduced cost matrix associated with the basic solution x'.

Dual feasibility of a basic feasible solution is a sufficient condition for the efficiency of a basic feasible solution of the multiobjective linear programming problem. It is also necessary for the efficiency of a nondegenerate basic feasible solution. A degenerate basic feasible solution may be efficient without being dual feasible, but at least one of the degenerate efficient basic solutions which represent the same extreme point in the convex polyhedron is dual feasible. Thus, in order to obtain all efficient extreme points determination of all dual feasible bases is adequate. Isermann gives the following definition of the solution graphs.

Definition: II.1.2.3. Let E be the set of the dual feasible bases and $L = \{(x^i, x^j)\ x^i$ and x^j are adjacent dual feasible basic solutions$\}$. The undirected graph $G = (E,L)$ is the solution graph.

He gives the proof that the solution graph G is finite and connected. Based on this fact he gives an algorithm to enumerate all dual feasible bases. Kıziltan [19] gives an algorithm to enumerate all efficient extereme points which is based on the dual feasibility concept. The monotone connectedness properity which is stated in the following theorem provides an efficient procedure for enumerating all dual feasible bases through a path of efficient edges along which the value of a specific objective is nonincreasing.

__Theorem:__ II.1.2.2. Each dual feasible basis, except the one where objective k attains its maximum, has at least one adjacent dual feasible basis obtained by introducing a non-basic variable x_s with the associated reduced cost coefficient of the k th objective is strictly less than zero.

Kıziltan gives the following LP subproblem for the efficiency check.

$$\text{(II.1.2.4)} \qquad \begin{array}{ll} \text{Min} & s_q \\ \text{s.t} & \\ & -R^{jT} v + s = R^{jT} e, \ \forall \ j \in P \\ & v \geqq 0 \ , \ s \geqq 0 \end{array}$$

where R^j is the j th column of the reduced cost matrix, and $P = \{j/R^j \not\leq 0\}$. If the minimum of s_q is zero then entering x_q will lead to an efficient basic solution.

Starting at the dual feasible basis where the k th objective is at its maximum it is only required for the nonbasic variable x_q such that $R_{qk} \geq 0$ is to be checked whether they lead to an efficient basis or not. Employing some additional tests eliminates the need for solving the sub-problem for each x_q such that $R_{qk} \geq 0$.

II.2. SOME MAJOR NETWORK PROBLEMS

II.2.1. INTRODUCTION

The theory on networks is closely related with the graph theory and the same terminology is used. The following definitions are employed both in graph theory and networks.

A graph $G(N,E,\Phi)$ is a collection of nodes denoted by the set N and edges denoted by the set E. Φ is the incidence relationships between the nodes and the edges. If the links have directions then the graph is called as a directed graph.

A chain is a sequence of links connecting any two nodes, when the direction along, the chain is specified, then it is called a path. If the initial and terminal nodes of a chain is the same then it is called as a cycle, similarly if the initial and terminal nodes of a path is the same then it is called a circuit.

A graph is said to be connected if there is at least one chain connecting every pair of nodes. A tree is a connected graph with no cycles. A spanning tree of a graph G is a subgraph of G which forms a tree including every node of G. If there is an one ended arc outward from a node of a spanning tree then it is called as rooted spanning tree. The one ended are is called a root.

A network is a graph which has weights associated with the arcs. A source in a network is a node with all of its arcs are directed outwards and a sink is a node with all of its arcs are directed in.

The theory on networks can be applied to a wide range of problems from diverse areas such as electrical networks, communication systems, transportation systems, information theory and data structures, project planning and production scheduling. The problems which have the special properities of the network structure, not necessarily representing a physical network, can be formulated as network models. For such problems, special algorithms which are more efficient than the general methods have been developed by exploiting the network structure.

Three main network problems are shortest path, maximum flow and the minimum cost flow problems. These problems will be reviewed briefly in the following sections. By using logarithmic transformation the network problems with multiplicative objective function can be formulated as shortest path problems. That simple transformation technique employed in chapter 4. Maximum flow and minimum cost flow objectives can be handled in the same multiobjective minimization formulation. The solution procedure proposed in chapter 3 has been derived from a solution method for the minimum cost flow network problem.

II.2.2. THE SHORTEST PATH PROBLEM

Given a directed network with the distances assigned on the arcs, the problem is to find the minimum distance path from one specific node to another.

The shortest path from a node to another node (or all other nodes which are determined jointly) can be found when the network has no negative cycles. One of the most efficient shortest path algorithm for networks with positive arcs is Dijkstra's algorithm with $O(m^2)$ computations and comparisons, where m is the number of nodes[20].

II.2.3. THE MAXIMUM FLOW PROBLEM

Given a directed network with arc capacities, the problem is to maximize the flow from a source to a sink. The concept and algorithm is due to Ford and Fulkerson[13]. They give the theorem of max-flow min-cut stating the maximum flow through a capacitated network from the source to the sink is equal to the capacity of the minimal cut. The constructive proof of the theorem leads to the maximum flow algorithm[12]. Ford and Fulkerson's max-flow algorithm is the first labeling algorithm in the field.

The maximum flow problem may be formulated as follows:

$$\max \quad v$$

s.t.

$$\sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = \begin{cases} v, & \text{if } i=1 \\ 0, & \text{if } i\neq 1 \text{ or } i\neq m \\ -v, & \text{if } i=m \end{cases}$$

$$0 \leq x_{ij} \leq u_{ij}$$

The max-flow problem can be generalized to a multi-source multi-sink problem by adding a pair of nodes being the super source and the super sink and arcs connecting the super source to the sources and the sinks to the super sink as in figure II.1.

Figure II.1

If there are capacitated nodes in the network, they can be handled by creating two vertices instead of the capacitated node, which are connected by an arc with the capacity of the node, as in Figures II.2.a and II.2.b.

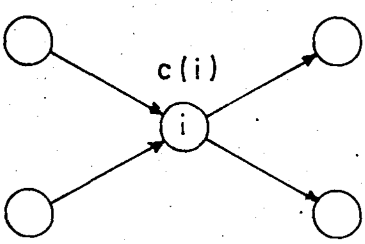


Figure II.2.a.

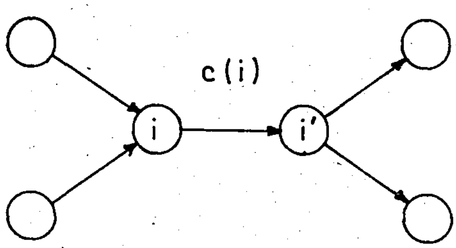


Figure II.2.b.

II.2.4. THE MINIMUM COST FLOW PROBLEM

Given a directed network with unit shipping costs on the arcs, the problem is to find the flow vector, composed of arc flows, which satisfies the supply and demand contraints and minimizes the cost.

The well known transportation and transshipment problems are minimum cost flow problems. The most general of such problems is the minimum cost circulation problem.

II.2.4.1. The Transportation Problem

The problem is to find the minimum cost flow in a bipartite graph, where the two kinds of nodes are origins and destinations. There can be arcs only from origins to destinations. The problem can be formulated as follows:

$$
\text{(II.2.4.1)} \qquad
\begin{aligned}
\min \quad & \sum_{(i,j)} c_{ij} \, x_{ij} \\
\text{s.t} \quad & \sum_{j=1}^{n} x_{ij} = s_i \;, \quad i=1,\ldots,m \\
& \sum_{i=1}^{m} x_{ij} = d_i \;, \quad j=1,\ldots,n \\
& x \geqq 0
\end{aligned}
$$

In order to have a feasible solution to the problem above the total demand must be equal to the total supply, that is

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_i$$

If the total supply exceeds the total demand or vice versa a dummy destination or a dummy source must be added with the difference as its demand or supply.

The problem may be solved by the efficient transportation algorithm rather than the general simplex method.

II.2.4.2. The Transshipment Problem

If there are nodes which are neither supply nodes or demand nodes in the network, then the problem is the transshipment problem. The transshipment problem can be converted into a transportation problem and solved by the transportation algorithm or by the network simplex method of linear programming[3].

II.2.4.3. The Minimum Cost Flow Problem

If it is possible to send the flow through the other sources and sinks, that is if there can be transshipment from a source or a sink, then the problem is a minimum cost flow problem and can be formulated as follows:

$$\min \sum_{(i,j)} c_{ij} x_{ij}$$

(II.2.4.2) s.t.

$$\sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = b_i \ , \ i=1,\ldots,m$$

$$x_{ij} \geq 0 \ , \ \forall (ij)$$

Nodes with $b_i > 0$ are called source nodes, with $b_i < 0$ sinks and with $b_i = 0$ are transshipment nodes.

If there are upper and lower bounds on the arc flows the nonnegativity constraints are replaced by the bound constraints such as $x_{ij} \geq \ell_{ij}$ and $x_{ij} \leq u_{ij}$, where ℓ_{ij} and u_{ij} are the lower and the upper bounds on the flow of arc (i,j).

The minimum cost network flow problem can be solved by the network simplex method which will be described and used in the next chapter.

The problems stated above can be reformulated as a minimum circulation problem by adding a return flow from the sink to the sources. This formulation is the most general of the network flow problems and can be used in multiple objective network programming.

The out-of-kilter algorithm requires this formulation. This algorithm is an another efficient network flow algorithm which is due to Fulkerson[10].

II.3. LITERATURE SURVEY ON MULTIOBJECTIVE NETWORK PROBLEMS

In literature, there are very few papers dealing with developing efficient methods for multiobjective network analysis, however, in the field of application, multiobjective methods are required.

In this section, six papers will be surveyed. Two of them are proposing methods for bicriterion transportation problems; one is an adaptation of a method developed for general multiobjective linear programming problems to the transportation problem; the other one proposes multiobjective shortest path algorithms and the remaining two report models and solutions for real life multiobjective transshipment and shortest path problems.

Aneja and Nair[1] give an algorithm for bicriterion transportation problem which finds the efficient extreme points in the objective space. Since the objective space is convex it is possible to find the set of the efficient extreme points by parametric search in the objective space. The algorithm starts with finding the minimum of both objectives, and generates other efficient extreme points in the objective space by repeatedly solving the transportation problem which minimizes a positively weighted average of the objective functions. Choosing two points r and s in the objective space from previously stored pairs of efficient extreme points, new weigths for the weighted objective function are calculated as $a_1^{(r,s)} = |z_2^s - z_2^r|$ and $a_2^{(r,s)} = |z_1^s - z_1^r|$ where z_i^r denotes the value of the ith objective at point r. Minimization of weighted objective function with these weights either generates a new efficient extreme point in the objective space or ends up at one of these points.

If weighted minimization results the point either r or s a new efficient extreme point is not generated but the pair (r,s) is excluded from further consideration. If a new efficient extreme point k is generated the two pairs (r,k) and (k,s) are added to the set of pairs of efficient point to be considered. Since the set of efficient extreme points is finite the algorithm terminates in finite number of iterations. It is reported that the algorithm terminates exactly at 2k-3 iterations, if there are k (k>2) efficient extreme points in the objective space. A third objective of minimizing the maximum time is incorporated by an outer loop using the bicriterion algorithm.

Srinivasan and Thompson[26] give another parametric bicriterion algorithm for finding cost verses average time trade-off curve for multimodal transportation problem. They state the problem as follows:

$$\text{(II.3.1)} \qquad \text{Min} \begin{bmatrix} \sum_{i,j,k} c_{ijk} \, x_{ijk} \\ \sum_{i,j,k} t_{ijk} \, x_{ijk} \end{bmatrix}$$

$$\text{s.t.} \quad \sum_{jk} x_{ijk} = s_i \,, \text{ for } \forall_i$$

$$\sum_{ik} x_{ijk} = d_j \,, \text{ for } \forall_j$$

$$x_{ijk} \geq 0$$

where c_{ijk} is the unit cost of transportation, t_{ijk} the time of shipping and x_{ijk} the amount of shipment from origin i to destination j with mode k. The right hand sides s_i and d_j are respectively the supply at i and demand at j.

In this formulation the average shipment time is expressed as weighted average by the shipments x_{ijk} as below:

$$\sum_{i,j,k} t_{ijk} \, x_{ijk} \, / \, \sum_i s_i$$

Since $\sum_i s_i$ is constant they use the second objective function in the formulation. They solve the bicriteria linear transportation problem by an parametric algorithm they have developed.

Thuente[30] gives two algorithms for the multicriteria shortest path problems. The first method is a dynamic programming approach for acyclic networks. When there exists an ℓ-vector of lengths associated with each arc the problem is to find the efficient paths from the source to the sink.

The nodes are renumbered such that $i < j$ if there is arc (i,j). Since the network is assumed to be acyclic this can be done. Thus, dynamic programming equations are given as:

$$f_1 = 0$$

$$f_i = \underset{j<i}{\text{eff}} \{f_j + d_{ji}\} \qquad i = 2,\dots,n$$

$$d_{ji} = \infty \qquad \text{if } (j,i) \notin A$$

Using operator "eff", he means all the efficient paths to node i from node 1 is stored. The nodes have labels $L(i)$ with $\ell+2$ entries which denote each efficient path from node 1 to each node i. The first and the second entries indicate the preceding node and the number of the efficient path coming from the previous node respectively and the remaining ℓ entries have the values of the ℓ criteria of the efficient path specified by the label.

The second approach proposed by Thuente uses interval criterion weights concept which is due to Steuer. He obtains $\ell-1$ combinations of lower and upper bounds of the weight interval such that $\sum_{b}^{\ell} w_k = 1$. It is proposed to find the weighted shortest path for each feasible combination of the weights.

These concepts about two proposed algorithms are given very briefly and are not developed into complete algorithms.

Isermann [18] constructs a multiobjective transportation problem in order to enumerate all efficient solutions of a linear multiple objective transportation problem based on the previous results he has reported in [17] for efficiency check and enumeration of the efficient solutions which are reviewed in the first section of this chapter.

Moore, Taylor and Lee [23] report a multiobjective transshipment model which is solved by goal programming. The problem has two parties involved, one is the management with the objective of cost minimization, the other is the labor union with a set of objectives such as shipping traffic from a specific plant to a specific warehause to be minimized, or maximum of 50 percent of total supply will be transshipped through specific warehauses. The problem is formulated into a goal programming model. The objective of weighted deviations is optimized and parametric analysis performed on the weights. The solution procedure is general and does not use the special structure of the transshipment problems.

Egberg, Cohon and ReVelle [9] report an initial work on a multiobjective analysis on the location planning of gas pipeline system from off shore platform to an onshore platform. The main objectives of the system stem from the high cost and potential environmental impact caused. They report that no explicitly multiple objective models have been adressed to gas pipeline network system as a whole. Environmental agencies have developed policies to minimize the number of pipelines and to confine them to a few specific pipeline corridors. The problem defined as a shortest path problem with multiple objectives but formulated as an LP problem. Two reasons have been reported for the LP formulation. The first is the difficulty of the post-optimality analysis in shortest path problems, second is the intention to include the network gathering and processing facilities into the model. The parametric objective weighing method is used to handle the minimization the corridor length, wetlands area, forrested area and developed area in the corridor.

CHAPTER III
MULTIOBJECTIVE LINEAR NETWORK FLOW PROBLEMS

III.1. INTRODUCTION

An important subclass of multiobjective network problems are multiobjective linear network flow problems. The cost of sending the flow from some supply nodes to some demand nodes is, generally, a linear function of the flow. Someother linear measures, for instance, an environmental cost measure, or the cost of deterioration of the flow as indicated in[1,9] can be incorporated. Sometimes, maximization or minimization of flow on some arcs is required. One other relevant problem can be to maximize the flow through the network simultaneously with the minimization of the other cost objectives.

These type of problems can be formulated as follows:

$$\text{(III.1.1)} \qquad \text{Min} \begin{bmatrix} c^1_{ij} & x_{ij} \\ \vdots & \\ c^\ell_{ij} & x_{ij} \end{bmatrix}$$

s.t.

$$\sum_{i=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = b_i \;, \quad i=1,2,\ldots,m$$

$$x_{ij} \geq 0$$

We have ℓ linear objectives to be minimized, where c_{ij}^{k}, $\{k=1,\ldots,\ell\}$ is the unit shipping cost due to objective k on the arc (i,j). The constraints of the problem are the flow conservation equations of the network. The difference between the total flow into and out of a node is equal to b_i $\{i=1,\ldots,m\}$. The right hand side of the constraints, b_i, is equal to the supply generated at node i. If the node i is a source node $b_i > 0$. If node i is an intermediate node $b_i = 0$ and if it is a sink then $b_i < 0$. The total supply is equal to the total demand, that is $\sum_{i=1}^{m} b_i = 0$.

The problem posed above is a MOLP problem and it can be solved by one of the methods developed for MOLP problems. Although the linear network flow problems can be solved by the simplex method, a network simplex method has been developed by exploiting the special structure of the network problems which is more efficient than the other approaches [15]. Also, in the multiobjective linear network problems a network simplex method provides an efficient procedure for finding the efficient extreme points.

II.2. REVIEW OF THE NETWORK SIMPLEX METHOD

The coefficient matrix of the constraint set of (III.1.1) is the node-arc incidence matrix of the network. The coefficient matrix does not have full rank. If we select an (m-1) by (m-1) submatrix we form a nonsingular matrix. There are m nodes with flow constraints and the rank of the coefficient matrix is (m-1). Since the simplex method requires a full rank constraint matrix, an artificial variable corresponding to a node should be added to form a matrix of rank m.

Every basis for the minimal cost network flow problem corresponds to a roted spanning tree, and every rooted spanning tree is lower triangular; thus every basis is triangular.

A matrix is totally unimodular if the determinant of every square submatrix formed from it has value -1, 0 or +1. In the network problem any column of the coefficient matrix corresponding to arc (i,j) contains exactly two nonzero elements a "1" in row i and a "-1" in row j. The coefficient matrix of a network flow problem is totally unimodular. Thus the basis matrix B is unimodular which implies that B^{-1} is an integer matrix, for arbitrary integer right hand side b, every basic solution formed as $(X_B, X_N) = (B^{-1} b, 0)$ is integer. Proofs can be obtained from Bazara and Jarvis[3] and Garfinkel and Nemhauser[14]. The triangularity of the basis matrix permits an efficient determination of the dual variables and the values of the basic variables directly on the network through a labeling procedure.

If lower and upper bounds are present on the arcs an algorithm for the bounded network can be used. In this case the arcs at either their upper or lower bounds are considered nonbasic.

A Labeling algorithm for the network Simplex Method with lower and upper bounds is reported in [3]. It is stated as follows:

Initialization Step

Start with an initial feasible basis, and set the basic variables to their required values.

Main Steps

1) Compute the dual variables w_i $i=1,\ldots,m$ set $w_m=0$. Compute w_i,s going from the root to the other nodes through the basic arcs.

2) For each nonbasic arc compute $z_{ij} - c_{ij} = w_i - w_j - c_{ij}$. If $z_{ij} - c_{ij} \leq 0$ for all nonbasic variables at their lower bound and $z_{ij} - c_{ij} \geq 0$ for all nonbasic variables at their upper bound. Stop, the present basis is optimal.

If for a nonbasic variable x_{ij} at its lower bound $z_{ij} - c_{ij} > 0$, or if for a nonbasic variable at its upper bound the reduced cost coefficient is less than zero, then such an arc is a candidate for an entering arc.

3) Entering one arc forms only one cycle with the spanning tree associated with the present basis. Through a labeling process determine the leaving arc as the first one which will go to its either

bound and also determine the maximum flow change Δx.

4) Perform the flow change by backtracking through the labels on the cycle formed in step 3.

5) If the entering variable goes from its lower bound to its upper bound or vice versa. The leaving variable and the entering variable are the same, and the basis is preserved return to step 2. Otherwise, remove the leaving variable from the basis and add the entering variable to the basis. Return to Step 1.

III.3. THE MULTIOBJECTIVE NETWORK SIMPLEX METHOD

For finding all efficient extreme points of a multi-objective linear network problem network simplex frame can be used instead of standard simplex format.

The basic modifications of the single objective algorithm are:

1) An $m \times \ell$ matrix of dual variables w_i^a are calculated
2) An $\ell \times n$ matrix of the reduced costs are calculated
3) An efficiency check subproblem is embedded
4) The calculation of the values of the basic arcs, for any given basis, is added.

The fourth modification performs the generation of a required basic solution solution which may require several piroting operations in the simplex tableau format through a labeling and backtracking process. This convenience is due to the network structure of the problem.

Since the set of efficient basic solutions is connected, starting at an efficient basic solution all of the efficient extreme solutions of the convex polyhedron defined by the flow conservation conservation constraints can be enumerated.

III.3.1. FINDING AN INITIAL EFFICIENT BASIC SOLUTION

Although there are other ways of finding an initial efficient basic feasible solution, a method which can be used is to find the basic solution which minimizes any one of the objectives, say the first objective. If there is no alternative solution to the optimal basic solution to the first objective that solution is an efficient solution. Otherwise, alternative optimal bases are enumerated until a dual feasible basis is found.

The method described above for finding an initial efficient basic solution is analogous to Phase I of the single objective simplex method.

III.4. A LABELING ALGORITHM FOR THE MULTIOBJECTIVE NETWORK SIMPLEX METHOD

III.4.1. DEVELOPMENT OF THE ALGORITHM

The algorithm is developed for enumeration of all efficient extreme points of a linear multiobjective network flow problem. The efficiency check and the concept of monotone connectedness is applied from Kızıltan[19]. Since the problem is a vector minimization problem here, the LP subproblem for the efficiency check is changed as:

$$\min \quad s_{ef}$$

$$\text{s.t} \quad v^T R + s = -eR$$

$$v \geqq 0, \quad s \geqq 0$$

where s_{ef} is the slack associated with the nonbasic arc (e,f) and R is the reduced cost matrix.

III.4.2. THE ALGORITHM

Initial Step

Start with an efficient basic solution J_1 which minimizes the 1st objective.

Form the set of generated efficient basic solutions $N \leftarrow \{J_1\}$

Form the set of efficient basic solutions to be generated $M \leftarrow \emptyset$

Form the set J_1' as the set of nonbasic arcs.

Set $k \leftarrow 1$

Main Steps

1) Set $w_m^a = 0$, $a = 1,\ldots,\ell$, ℓ = the number of objectives

- If w_i^a has been computed, w_i^a has not been computed and arc (i,j) is a basic arc, then set $w_j^a = w_i^a - c_{ij}^a$.

- If w_i^a has been computed, w_j^a has not been computed and arc (j,i) is a basic arc, then set $w_j^a = w_i^a + c_{ij}^a$.
- Repeat Step 1 until all w_i^a's have been computed.

2) For each arc $\varepsilon\ J_k'$ compute

$$z_{ij}^a - c_{ij}^a \leftarrow w_i^a - w_i^a - c_{ij}^a \ , \quad a = 1,\ldots,\ell$$

3) i) Apply the prelimenary check to each $(e,f)\ \varepsilon\ J_k'$. If there exist an arc $(e,f)\ \varepsilon\ J_k'$ such that $(Z-c)^{(e,f)} \leqq 0$, then J_{ef} is a dominated basis, where $(z-c)^{(e,f)}$ is the reduced cost column of the nonbasic arc (e,f) and J_{ef} is the new basis which will be obtained by introducing the arc $(e,f)\ \varepsilon\ J_k'$ into the basis.

ii) Form B_k as the set of remaining nonbasic arcs. Form the sets $P_k \leftarrow \{(i,j)\ z_{ij}^1 - c_{ij}^1 \leqq 0,\ (i,j)\ \varepsilon\ B_k\}$ $EV \leftarrow \emptyset$. If $P_k = \emptyset$ go to step 10, otherwise continue.

4) Choose an arc $(e,f)\ \varepsilon\ P_k$ and apply the efficiency check by solving the LP subproblem:

$$\begin{aligned} \text{Min} \quad & s_{ef} \\ \text{s.t} \quad & \\ & v^T R + s = -e^T R \\ & v \geqq 0 \ , \ s \geqq 0 \end{aligned}$$

where R is the reduced cost matrix associated with the arcs $\varepsilon\ B_k$.

Make the following observations during the solution of the subproblem

i) If any s_{ij} associated with $(i,j) \in P_k$ is nonbasic in any simplex tableau or basic at zero level update the sets as $EV \leftarrow EV + \{(i,j)\}$ and $P_k \leftarrow P_k - \{(i,j)\}$.

ii) If all coefficients of the corresponding row to a basic s_{ij} $((i,j) \in P_k)$ arc nonpositive update P_k as $P_k \leftarrow P_k - \{(i,j)\}$

iii) If the optimal value of s_{ef} is zero $EV \leftarrow EV + \{(e,f)\}$ and $P_k \leftarrow P_k - \{(e,f)\}$

If $P_k \neq \emptyset$ repeat step 4. Otherwise continue.

5) If $EV = \emptyset$ go to Step 8. Otherwise choose $(e,f) \in EV$, set $s = f$, $t = e$ and $L(s) = (+t,\infty)$

6- a) If node i has a label, node j has no label and arc (i,j) is basic, set $L(j) = (i,\Delta j)$ where $\Delta j = \Delta i$.

b) If node i has a label, node j has no label and arc (j,i) is basic set $L(j) = (-i,\Delta_j)$, where $\Delta_j = \min\{\Delta_i, x_{ji}\}$, if $x_{ji} < \Delta_i$ set $(g,h) = (j,i)$.

c) Repeat step 6 until node t is labeled.

7) Leaving arc is (g,h) and entering arc is (e,f). Set $EV \leftarrow EV - \{(e,f)\}$, and $J_{ef} \leftarrow J_k - \{(g,h)\} + \{(e,f)\}$. If $J_{ef} \in N$ go to Step 5. Otherwise set $M \leftarrow M + J_{ef}$ and go to Step 5.

8) If $J_{ef} \in N$ (i,e the last basis which has been obtained by entering (e,f) is explored before) go to step 10. Otherwise continue.

9) Set $\Delta = \Delta_t$. If the first entry of L(t) is i, then add Δ to x_{kt}. Otherwise, if the first entry in L(t) is -i, subtract Δ from x_{it}. Backtrack to node i and repeat the process until node t is reached in the backtracking process.

Update the sets $M \leftarrow M - J_{ef}$ and $N \leftarrow N + J_{ef}$.
Set $k \leftarrow k + 1$ and $J_k \leftarrow J_{ef}$, go to step 1.

10) If $M = \emptyset$ stop. N is equal to the set of all efficient bases. Otherwise, select a basis $J \varepsilon M$ and obtain that basic solution through a labeling and backtracking process. Set $k \leftarrow k + 1$, $J_k \leftarrow J$. Update the sets $M \leftarrow M - J_k$ and $N \leftarrow N + J_k$. go to step 1.

III.4.3. EXAMPLE PROBLEM

To illustrate the algorithm the example problem below is solved.

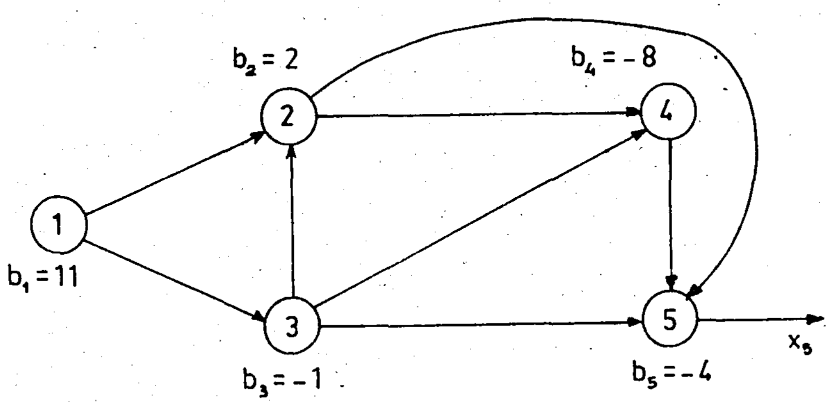


Figure III.4.1

Given the network in Fig.II.4.1 and the cost matrix

$$
\begin{array}{cc}
 & \begin{array}{ccccccccc} x_{12} & x_{13} & x_{24} & x_{25} & x_{32} & x_{34} & x_{35} & x_{45} & x_{5} \end{array} \\
C = & \begin{bmatrix} 1 & 2 & 5 & 4 & 1 & 3 & 1 & 2 & 0 \\ 2 & 4 & 2 & 2 & 2 & 1 & 3 & 1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{array}
$$

find all the efficient extreme points.

The solution procedure:

Initial Step

The initial efficient basic solution is obtained by maximizing the first objective,

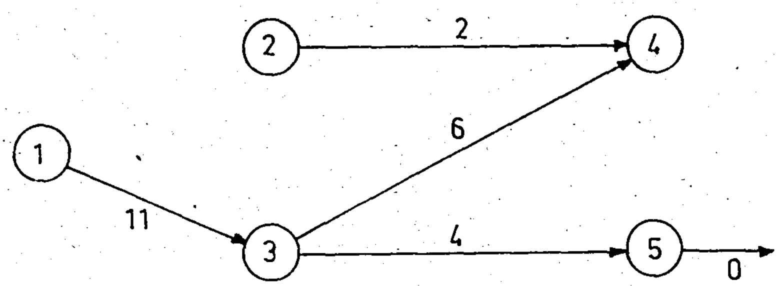


Figure III.4.2

$k \leftarrow 1$

$J_k \leftarrow \{(1,3),(2,4),(3,4),(3,5)\}$

$N \leftarrow J_k = \{|(1,3),(2,4),(3,4),(3,5)|\}$

$J'_k \leftarrow \{(1,2),(2,5),(3,2),(4,5)\}$

Step 1) $w_5^1 = 0$, $w_5^2 = 0$, $w_5^3 = 0$

$w_3^a = w_5^a + c_{35}^a$

$w_3^1 = 1$, $w_3^2 = 3$, $w_3^3 = 0$

$w_1^a = w_3^a + c_{13}^a$

$w_1^1 = 3$, $w_1^2 = 7$, $w_1^3 = -1$

$w_4^a = w_3^a - v_{43}^a$

$w_4^1 = -2$, $w_4^2 = 2$, $w_4^3 = 0$

$w_2^a = w_4^a + v_{24}^a$

$w_2^1 = 3$, $w_2^2 = 4$, $w_2^3 = 0$

Step 2) for each (ij) J_k^1 calculate;

$$(Z-C)^{(1,2)} = \begin{bmatrix} 3 - 3 - 1 \\ 7 - 4 - 2 \\ -1 - 0 - 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

$$(Z-C)^{(2,5)} = \begin{bmatrix} 3 - 0 - 4 \\ 4 - 0 - 2 \\ 0 - 0 - 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix}$$

$$(Z-C)^{(3,2)} = \begin{bmatrix} 1 - 3 - 1 \\ 3 - 4 - 2 \\ 0 - 0 + 1 \end{bmatrix} = \begin{bmatrix} -3 \\ -3 \\ 1 \end{bmatrix}$$

$$(Z-C)^{(4,5)} = \begin{bmatrix} -2 - 0 - 2 \\ 2 - 0 - 1 \\ 0 - 0 - 0 \end{bmatrix} = \begin{bmatrix} -4 \\ 1 \\ 0 \end{bmatrix}$$

Step 3)

i) There is no nonbasic arc (e,f) such that

$(Z-C)^{(e,f)} \leq 0$

ii) $B_1 = \{(1,2),(2,5),(3,2),(4,5)\}$

Step 4)

For each $(e,f) \in P_k = \{(i,j) \mid z^1_{ij} - c^1_{ij} \leq 0 , (i,j) \in B_k\}$, $P_k = B_k$ perform the efficiency check through the subproblem.

Initial basic feasible solution to the subproblem.

	v_1	v_2	v_3	s_{12}	s_{25}	s_{32}	s_{45}	
s_{12}	0	-1	-1	1	-1	0	0	2
v_1	1	-2	0	0	-1	0	0	1
s_{32}	0	-9	1	0	-3	1	0	8
s_{45}	0	-7	0	0	-4	0	1	7

Table III.4.1

Observations on the tableau:

1) s_{25} is nonbasic therefore entering x_{25} will lead to an efficient basic solution.

2) s_{12} and s_{45} can not be less than 2 and 7 respectively because all of the coefficients of the corresponding rows are nonpositive. Thus entering x_{12} or x_{45} will not lead to efficient bases.

Only s_{32} remains to be checked by solving the subproblem.

which minimizes s_{32}. It becomes nonbasic at the first iteration therefore x_{32} will lead to an efficient basic solution.

Form the set EV the set of nonbasic variables which lead to efficient bases

$EV = \{(2,5),(3,2)\}$

$P_k = \emptyset$

Step 5)

$(e,f) = (2,5)$, $s = 5$, $t = 2$, $L(5) = (+2,\infty)$

Step 6)

$(3,5)$ basic 3 has no label $L(3) = (-5,4),(g,h) = (3,5)$

$(3,4) \to L(4) = (+3,4)$

$(2,4) \to L(2) = (-4,2),(g,h) = (2,4)$

Step 7)

$$J_{25} = \{(1,3),(2,4),(3,4),(3,5)\} + \{(2,5)\} - \{(2,4)\}$$
$$= \{(1,3),(3,4),(3,5),(2,5)\}$$

$J_{25} \notin N$, $M \leftarrow M + J_{25} = \{[(1,3),(3,4),(3,5),(2,5)]\}$

$EV \leftarrow EV - \{(2,5)\} = \{(3,2)\}$

Step 5) $(e,f) = (3,2)$, $s = 2$, $t = 3$, $L(2) = (+3,\infty)$

Step 6) $(2,4)$ basic 4 has no label $L(4) = (+2,\infty)$

$(3,4) \to L(3) = (-4,6),(g,h) = (3,4)$

Step 7) $J_{32} = \{(1,3)\ ,\ (2,4)\ ,\ (3,5)\ ,\ (3,2)\}\ ,\ J_{32} \not\in N$

$M \leftarrow M + J_{32} = \{[(1,3),(3,4),(3,5),(2,5)],[(1,3),(2,4),(3,5),(3,2)]\}$

$EV \leftarrow \emptyset$

Step 5 $EV = \emptyset$ go to step 8

Step 8) $(e,f) = (3,2)\ ,\ J_{32} \not\in N$

$\Delta \leftarrow \Delta_2 = 6$

Step 9) $L(3) = (-4,6) \rightarrow x_{34} \rightarrow x_{34} - 6 = 0$

$L(4) = (2,\infty) \rightarrow x_{24} \leftarrow x_{24} + 6 = 8$

$L(2) = (3,\infty) \rightarrow x_{32} \leftarrow x_{32} + 6 = 6$

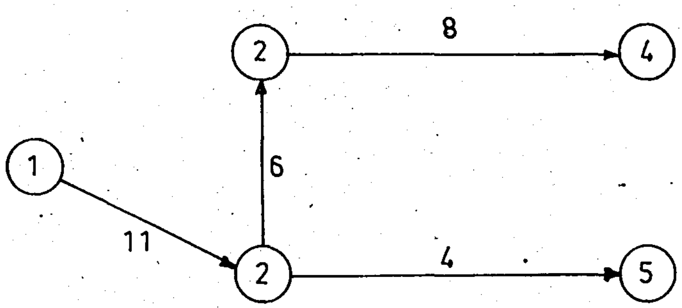


Figure II.4.2. The new efficient basic solution adjacent to J_1.

$M \leftarrow M - J_{32} = \{[(1,3),(3,4),(3,5),(2,5)]\}$

$N \leftarrow N + J_{32} = \{[(1,3),(2,4),(3,4),(3,5)],[(1,3),(2,4),(3,5),(3,2)]\}$

$k \leftarrow k + 1 = 2\ ,\ J_2 \leftarrow J_{32}$, go to step 1.

Step 1)

$$w_5^1 = 0\ ,\ w_5^2 = 0\ ,\ w_5^3 = 0$$
$$w_3^1 = 1\ ,\ w_3^2 = 3\ ,\ w_3^3 = 0$$
$$w_1^1 = 3\ ,\ w_1^2 = 7\ ,\ w_1^3 = -1$$
$$w_2^1 = 0\ ,\ w_2^2 = 1\ ,\ w_2^3 = 1$$
$$w_4^1 = -5,\ w_4^2 = -1,\ w_4^3 = 1$$

Step 2)

$$(Z-C)^{(1,2)} = \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix},\ (Z-C)^{(2,5)} = \begin{bmatrix} -4 \\ -1 \\ 1 \end{bmatrix},\ (Z-C)^{(3,4)} = \begin{bmatrix} 3 \\ 3 \\ -1 \end{bmatrix},$$

$$(Z-C)^{(4,5)} = \begin{bmatrix} -7 \\ -2 \\ 1 \end{bmatrix}$$

Step 3)

i) There is no nonbasic arc (e,f) such that $(Z-C)^{(e,f)} \leq 0$

ii) $B_k = \{(1,3),(2,5),(3,4),(4,5)\}$

$P_k = \{(2,5),(4,5)\}$

Step 4)

	v_1	v_2	v_3	s_{12}	s_{25}	s_{34}	s_{45}	
v_3	0	-13	1	0	-3	-4	0	21/2
s_{12}	0	-16	0	1	-4	-6	0	13
v_1	1	- 3	0	0	-1	-1	0	2
s_{45}	0	-10	0	0	-4	-3	1	11

Table III.4.2.
Initial basic feasible solution to the subproblem.

Observations on the initial basic feasible solution

1) s_{25} is nonbasic therefore entering x_{25} will lead to an efficient basic solution.

2) All of the coefficients of the row associated with s_{45} is nonpositive and $s_{45} > 11$, therefore, entering x_{45} will not lead to an efficient basis.

$EV = \{(2,5)\}$,

$P_k = \emptyset$.

Step 5) $(e,f) = (2,5)$, $s = 5$, $t = 2$, $L(5) = (+2,\infty)$

Step 6) $(g,h) = (3,5)$

Step 7) $J_{25} = \{(1,3),(2,4),(3,2),(2,5)\}$, $J_{25} \not\in N$

$M \leftarrow M + J_{25} = \{[(1,3),(3,4),(3,5),(2,5)],[(1,3),(2,4),(3,2),(2,5)]\}$

$EV \leftarrow EV - \{(2,5)\} = \emptyset$

Step 5) $EV = \emptyset$, go to step 8

Step 8) $(e,f) = (2,5)$, $J_{25} \not\in N$

Step 9) $\Delta \leftarrow \Delta_t = 4$

Through the flow change process the following basic solution is obtained.

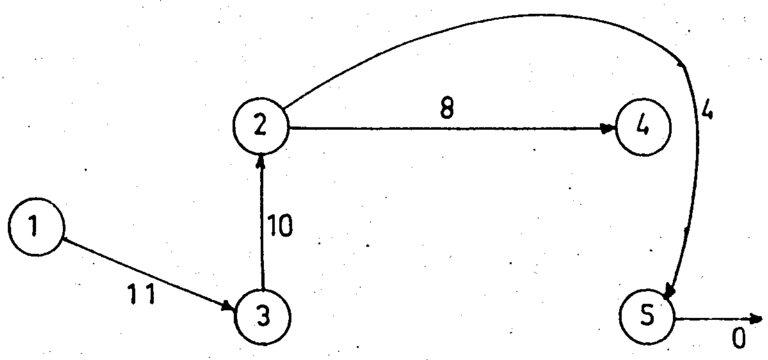


Figure III.4.3. New efficient basic solution adjacent to J_2.

$M \leftarrow M - J_{25} = \{[(1,3),(3,4),(3,5),(2,5)]\}$

$N \leftarrow N + J_{25} = \{[(1,3),(2,4),(3,4),(3,5)],[(1,3),(2,4),(3,5),(3,2)],[(1,3),(2,4),(3,2),(2,5)]\}$

$k \leftarrow k + 1 = 3$

$J_3 \leftarrow J_{25}$, go to step 1.

Step 1) w_i^a , $a = 1,2,3$, $i=1,\ldots,5$ are calculated

Step 2) $(Z-C)^{(1,2)} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$, $(Z-C)^{(3,4)} = \begin{bmatrix} 3 \\ 3 \\ -1 \end{bmatrix}$, $(Z-C)^{(3,5)} = \begin{bmatrix} 4 \\ 1 \\ -1 \end{bmatrix}$,

$(Z-C)^{(4,5)} = \begin{bmatrix} -3 \\ -1 \\ 0 \end{bmatrix}$

Step 3)

i) $(Z-C)^{(4,5)} \leq 0$, therefore, J_{45} is dominated by the present basis.

ii) $B_3 = \{(1,2),(3,4),(3,5)\}$

$P_k = \emptyset$

Step 4) $P_k = \emptyset$ i.e there is no arc $(e,f) \in B_3$ such that $z_{ef}^1 - c_{ef}^1 \leq 0$, then go to step 10.

Step 10)

$k \leftarrow k + 1 = 4$

Select the last element of M

$J_4 = \{(1,3),(2,5),(3,4),(3,5)\}$

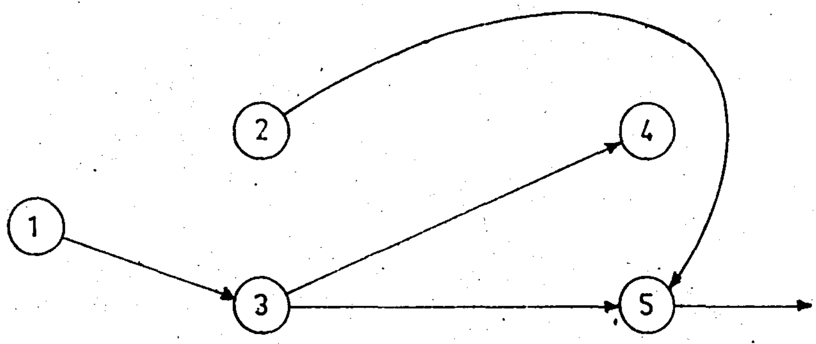


Figure III.4.4. The selected efficient basis from the set M to which a move will be done.

$x_{13} = 11$, $x_{25} = 2$, $x_{34} = 8$

$x_{35} = 4 - 2 = 2$

$M \leftarrow M - J_4 = \emptyset$

$N \leftarrow N + J_4$

go to step 1.

Performing steps 1 to 4, the set $EV = \{(1,2),(3,2)\}$ is obtained.

Performing steps 5 to 7 for $(i,j) \in EV$

$J_{12} = \{(1,3),(2,5),(3,4),(1,2)\}$, $J_{12} \notin N$

$J_{32} = \{(1,3),(2,5),(3,4),(3,2)\}$, $J_{32} \notin N$

are obtained and the set M is updated. At steps 8 and 9 the arc (3,2) is entered, The efficient basic solution is obtained.

The following updates are made $M \leftarrow M - J_{32}$, $N \leftarrow N + J_{32}$, $k \leftarrow k + 1 = 5$, $J_5 \leftarrow J_{32}$, and return to step 1.

Performing steps 1 to 4 on the basis J_5 it is detected that $EV = \emptyset$ and at step 10 k is updated as $k \leftarrow 6$, a basis ε M is selected as $J_6 = \{(1,3),(2,5),(3,4),(1,2)\}$ and the corresponding basic feasible solution is obtained. The sets M and N are updated, and returned to step 1.

Performing steps 1 to 4 on the basis J_6 the set $EV = \{(2,4),(3,2)\}$ is obtained.

Performing steps 5-7 for $(i,j) \varepsilon EV$

$$J_{24} = \{(1,3),(1,2),(2,5),(2,4)\} \text{ , } J_{24} \notin N$$
$$M \leftarrow M + J_{24} = \{[(1,3),(1,2),(2,5),(2,4)]\} \text{ and}$$
$$J_{32} = \{(1,3),(2,5),(3,4),(3,2)\} \text{ , } J_{32} \in N$$

are obtained.

Step 8) $(e,f) = (3,2)$

$J_{32} \varepsilon N$, i.e $J_{32} \notin M$, $EV = EV - \{(3,2)\}$

Select $(e,f) \varepsilon EV$, $(e,f) \leftarrow (2,4)$,

perform labeling and flow change process

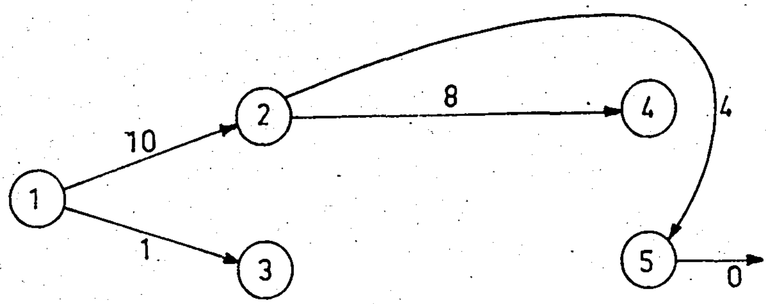


Figure III.4.5. The new efficient basic feasible solution adjacent to J_6.

$M \leftarrow M - J_{24} = \emptyset$

$N \leftarrow N + J_{24}$

$k \leftarrow k + 1 = 7$

$J_7 \leftarrow J_{24} = \{(1,3),(1,2),(2,5),(2,4)\}$

go to step 1

Performing steps 1 to 4 on the basis J_7 the set $EV = \{(3,4)\}$ is obtained

Performing steps 5,6,7, $J_{34} = \{(1,3),(1,2),(2,5),(3,4)\}$ is obtained, $J_{34} \varepsilon N$, go to step 10.

Step 10)

$M = \emptyset$

The set N is the set of all efficient basic solutions.

III.5. A LABELING ALGORITHM FOR THE MULTIOBJECTIVE NETWORK SIMPLEX METHOD WITH LOWER AND UPPER BOUNDS

III.5.1. DEVELOPMENT OF THE ALGORITHM

In this section, the multiobjective network simplex algorithm will be extended to the case with lower and upper bounds. When there exists bounds on the arc flows the efficiency check subproblem is modified as follows:

$$\text{Min} \quad s_{ef}$$

$$\text{s.t.}$$

$$v^T(Z-C)^{(i,j)} + s_{ij} = -e^T(Z-C)^{(i,j)} \quad , \forall \ (i,j) \ \varepsilon \ LB$$

$$-v^T(Z-C)^{(b,d)} + s_{bd} = e^T(Z-C)^{(b,d)} \quad , \forall \ (b,d) \ \varepsilon \ UB$$

$$v \geq 0 \ , \ s \geq 0$$

where $(Z-C)^{ij}$ is the reduced cost column associated with arc (i,j), LB and UB are the set of nonbasic arcs at their lower and upper bounds respectively such that introducing any one will not yield in decrease in the 1st objective.

Also the labeling routine is extended accordingly. For the sake of completeness the steps which are identical with the previous algorithm will be repeated here.

III.5.2. THE ALGORITHM WITH BOUNDS

Initial Step

Start with an efficient basic solution J_1 which minimizes the 1st objective

Here, an efficient basic solution is identified both by its basic arcs and also with its nonbasic arcs indicating whether these ares are their upper or lower bounds. Form the set of generated efficient basic solutions $N \leftarrow \{J_1\}$. Form the set of efficient basic solutions to be generated $M \leftarrow \emptyset$. Form the set J_1' as the set of nonbasic arcs.

Set $k \leftarrow 1$

Main Steps

1) Set $w_m^a = 0$, $a = 1,\ldots,\ell$, ℓ = the number of objectives

- If w_i^a has been computed, w_j^a has not been computed and arc (i,j) is a basic arc, then set $w_j^a = w_i^a - c_{ij}^a$.

- If w_i^a has been computed, w_j^a has not been computed and arc (j,i) is a basic arc, then set $w_j^a = w_i^a + c_{ij}^a$.

Repeat step 1 until all w_i^a s have been computed.

2) For each nonbasic arc compute.

$$z_{ij}^a - c_{ij}^a = w_i^a - w_j^a - c_{ij}^a \ , \ a = 1,\ldots,\ell$$

3) Apply the prelimenary check to each $(e,f) \in J_k'$

i-a) If there exists an arc $(e,f) \in J_k^1$ and x_{ef} ℓ_{ef}, where ℓ_{ef} is the lower bound on arc (e,f), such that $(Z-C)^{(e,f)} \leq 0$, then J_{ef} is a dominated arc (e,f), and J_{ef} is the new basis which will be obtained by introducing (e,f) into the basis.

i-b) If there exists $(e,f) \in J_k'$ and $x_{ef} = u_{ef}$. Where u_{ef} is the upper bound on arc (e,f), such that $(Z-C)^{(e,f)} \geq 0$, then J_{ef} is dominated.

ii) Form B_k as the set of remaining nonbasic arcs. Form the sets:

$P_k \leftarrow \{(i,j)\ z_{ij}^1 - c_{ij}^1 \leq 0$ if $x_{ij} = \ell_{ij}$, or $z_{ij}^1 - c_{ij}^1 \geq 0$ if $x_{ij} = u_{ij}$, $(i,j) \in J_k'\}$ and

$EV \leftarrow \emptyset$

If $P_k = \emptyset$ go to step 10.

4) Choose an arc $(e,f) \in P_k$ and apply the efficiency check by solving the LP subproblem.

$$\min \quad s_{ef}$$

s.t.

$$v^T(Z-C)^{(i,j)} + s_{ij} = -e^T(Z-C)^{(i,j)} \quad , \forall\ (i,j) \in LB$$

$$-v^T(Z-C)^{(b,d)} + s_{bd} = e^T(Z-C)^{(b,d)} \quad , \forall\ (b,d) \in UB$$

$$v \geqq 0 \ , \ s \geqq 0$$

where $LB = \{(i,j) \mid x_{ij} = \ell_{ij} \text{ and } (i,j) \in B_k\}$ and $UB = \{(i,j) \mid x_{ij} = u_{ij} \text{ and } (i,j) \in B_k\}$.

Make the following observations during the solution of the subproblem.

i) If any s_{ij} associated with $(i,j) \in P_k$ is nonbasic in any simplex tableau or basic at zero level update the sets $EV \leftarrow EV + \{(i,j)\}$ and $P_k \leftarrow P_k - \{(i,j)\}$

ii) If all coefficients of the corresponding row to a basic s_{ij} ($(i,j) \in P_k$ are nonpositive update P_k as $P_k \leftarrow P_k - \{(i,j)\}$

iii) If $(e,f) \in P_k$ make one iteration on the tableau if $s_{ef} = 0$ update $EV \leftarrow EV + \{(e,f)\}$ and $P_k \leftarrow P_k - \{(e,f)\}$ make observations (i) and (ii)

- If $P_k \neq \emptyset$ repeat step 4. Otherwise continue

5) If $EV = \emptyset$ go to Step 8. Otherwise choose $(e,f) \in EV$

- If $x_{ef} = u_{ef}$ set $s = e$, $t = f$, $(g,h) = (e,f)$
  set $L(s) = (-t,\ x_{ef} - \ell_{ef})$

- If $x_{ef} = \ell_{ef}$ set $s = f$, $t = e$, $(g,h) = (e,f)$
set $L(s) = (+t, u_{ef} - x_{ef})$

6) The labeling process

a) if node i has a label, node j has no label arc (i,j) is basic, set $L(j) = (+i, \Delta_j)$ where $\Delta_j = \min \{\Delta_i , u_{ij} - x_{ij}\}$, if $u_{ij} - x_{ij} < \Delta_i$ set $(g,h) = (i,j)$

b) if node i has a label, node j has no label arc (j,i) is basic, set $L(j) = (-i, \Delta_j)$ where $\Delta_j = \min \{\Delta_i, x_{ji} - \ell_{ji}\}$. If $x_{ji} - \ell_{ji} < \Delta_i$ set $(g,h) = (j,i)$

c) Repeat step 6 until node t is labeled.

7) Leaving arc is (g,h) and entering arc is (e,f).

Set $EV = EV - \{(e,f)\}$ and $J_{ef} \leftarrow J_k - \{(g,h)\} + \{(e,f)\}$. and adjust the upper and lower bound indicators accordingly. If $J_{ef} \in N$ go to Step 5. Otherwise set $M \leftarrow M + J_{ef}$ and go to step 5.

8) If $J_{ef} \in N$ go to step 10. Otherwise continue.

9) Execute the flow change process:

Set $\Delta = \Delta_t$, if the first entry of $L(t)$ is $+i$ then add Δ to x_{it}; otherwise, if the first entry in $L(t)$ is $-i$ subtract from x_{it}. Backtract to node i and repeat the process until node t is reached in the backtracking process.

Update the sets $M \leftarrow M - J_{ef}$ and $N \leftarrow N + J_{ef}$ Set $k \leftarrow k+1$ and $J_k \leftarrow J_{ef}$, go to Step 1.

10) If $M = \emptyset$ Stop. N is the set of all efficient bases. Otherwise, select a basis $J \in M$ and obtain that basic solution through a labeling and backtracking process Set $k \leftarrow k + 1$, $J_k \leftarrow J$. Update the sets $M \leftarrow M - J_k$ and $N \leftarrow N + J_k$ go to step 1.

CHAPTER IV

A SOLUTION PROCEDURE FOR MULTIOBJECTIVE PROBLEMS WITH A COMBINATION OF ADDITIVE AND A SPECIAL TYPE OF MULTIPLICATIVE OBJECTIVES

IV.1. INTRODUCTION

When a problem is given to maximize the reliability and minimize a linear cost function simultaneously, the reliability function can be formulated as a type of multiplicative function. $\Pi P_{ij}^{y_{ij}}$. The variables y_{ij} are zero-one variables. The working probability P_{ij} of the component (i,j) is independent of the others. This multiplicative function is transformed into an additive function by a simple logarithmic transformation $\Sigma - \ln(P_{ij})y_{ij}$. Using such transformation the most reliable path problem can be formulated and solved as a shortest path problem [10].

Given a network, the problem is stated as to send a flow which satisfies the flow conservation constraints, and simultaneously maximizing the reliability of sending the required amount of flow and minimizing the associated cost. The problem can be formulated as follows:

$$\max \prod_{i,j} p_{ij}^{y_{ij}}$$

$$\min \sum_{i,j} c_{ij}\, x_{ij}$$

s.t

(IV.1.1)
$$\sum_{j=1}^{n} X_{ij} - \sum_{k=1}^{n} X_{ki} = bi \quad , \text{ for } i= 1,\ldots,n$$

$$X_{ij} \geq 0 \quad , \text{ for } \forall\ (i,j)$$

$$y_{ij} = \begin{cases} 1 & , \text{ if } x_{ij}>0 \\ 0 & , \text{ otherwise} \end{cases}$$

Transforming the multiplicative objective to be maximized into $\Sigma - \ln(p_{ij}) y_{ij}$ results the following objectives

$$\min \begin{bmatrix} \sum_{i,j} - \ln(p_{ij})\ y_{ij} \\ \\ \sum_{i,j} c_{ij}\ x_{ij} \end{bmatrix}$$

subject to the same constraint set in the above formulation.

One can observe that the sum of a single additive objective function and a single multiplicative function results in a fixed charge objective function.

The problem above is stated as a bicriteria problem. In fact, there can be more than one multiplicative and additive objectives. For simplicity in statement and less computational work, the algorithm and its related example in sections 4 and 5 respectively of this chapter will be given only for the bicriteria case.

The costs associated with the arcs in a network can be handled as fixed charges. The other fixed charges, aside from the reliability measure, can be the construction costs, set up times or distances related with the arcs of the network. When the fixed charges are noncommensurable with the cost of flow a multiobjective fixed charge problem can be formulated.

Since we are dealing with vector comparisons in the multiobjective case, the solution of the fixed charge problem is not necessarily going to be the solution of the fixed charge problem, however, it will prove useful to investigate the solution procedures proposed for the fixed charge problem.

IV.2. THE FIXED CHARGE PROBLEM

IV.2.1. GENERAL PROPERTIES OF THE FIXED CHARGE PROBLEM

The fixed charge problem may be formulated as follows:

$$\text{Min} \quad \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n} d_j y_j$$

s.t.

(IV.2.1)
$$\sum_{j=1}^{n} a_{ij} x_j = b_i \qquad i = 1,2,\ldots,m$$

$$x_j \geq 0 \qquad y_j = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{if } x_j = 0 \end{cases}, \; j=1,2,\ldots,n$$

The problem stated above requires the minimization of a concave function over a convex polyhedron defined by $X = \{x | \Sigma a_{ij} x_j \leq b_i \ , \ i=1,2,...,m\}$. The concavity of the objective function has been proven in[7].

Since the global minimum of a concave function over a convex polyhedron is at one or more of the extreme points of the convex polyhedron, the optimum of the fixed charge problem is attained at an extreme point of the convex polyhedron defined by the constraints. Hirsch and Dantzig have shown that this is true for the fixed charge problem. It is also shown that for a nondegenerate problem with all equality constraints and all positive fixed charges, all extreme points are local minima.

The fixed charge problem can also be formulated as a mixed integer linear programming problem:

$$
\text{(IV.2.2)} \qquad
\begin{aligned}
\text{Min} \quad & \sum_{j=1}^{n} c_j x_j + d_j y_j \\
\text{s.t.} \quad & \\
& Ax = b \\
& x_j - M_j y_i \leq 0 \ , \quad j = 1,...,n \\
& y_i = (0,1) \qquad , \quad j = 1,...,n \\
& x \geq 0 \ .
\end{aligned}
$$

where M_j is an upper bound on x_j.

There are several approximate and exact algorithms for solution of the fixed charge problem.

Utilization of the fact that the optimal solution to the fixed charge problem is at an extreme point of the convex polyheron of the constraints results inn some approximate solution procedures by means of some heuristics or approximations. Balinski[2], Cooper and Drebes[7], Steinberg[27] have developed heuristic algorithms for the fixed charge problem.

The exact solution procedures are mixed integer solution, vertex ranking solution proposed by Murty[24] and improved by McKeown[22] and Taha[29] and branch and bound algorithms proposed by Bod[4] and Steinberg[27].

IV.2.2. EXACT SOLUTION PROCEDURES

Exact solution procedures for the fixed charge problem require partial or implicit enumeration of the vertices of the convex polyhedron. Thus they require much more computation time than the approximate algorithms. An appropriate mixed integer algorithm may be used to solve the fixed charge problem when formulated as a mixed integer problem. Algorithms specially designed for the fixed charge problem may generally be more efficient. In this section two such algorithms will be reviewed.

Murty[24] proposes a vertex ranking algorithm for the fixed charge problem. The algorithm is based on ranking the vertices of the polyhedron in nondecreasing order according to the linear objective function values and then adding the fixed charges to determine the optimal solution. Murty's procedure is based on two facts.

First: Let $s_1, s_2, \ldots, s_k$ be the k vertices of the convex polyhedron of the constraints rank in the nondecreasing order of their linear objective function values then the next vertex in the rank is adjacent to one of the k previous vertices. Secondly the ranking procedure can be bounded. Suppose some vertex s_r is determined in the rank the next vertex r+1 to be ranked should be such that $Z_{r+1}+D_o \leq Z_r+D_r$, where Z_r and Z_{r+1} are the linear objective values to the vertices r and r+1 respectively, D_r is the fixed charge value of the vertex r and D_o is a lower bound on the fixed charge. Thus when a vertex k+1 with $Z_{k+1}+D_o>Z_k+D_k$ is reached the ranking procedure stops, because vertices with the linear objective value greater than or equal to Z_k will yield greater objective value to the fixed charge problem than the k th vertex in the rank. The optimal solution can be obtained from the set of the vertices which were ranked so far. As Murty reports the efficiency of the algorithm improves with the nearness of D_o to the greatest lower bound of the fixed charge component of the objective function. As a numerical example, he solves a fixed charge transportation problem and determines D_o by summing the smallest ℓ fixed charges where ℓ is the number of destinations, since there should be at least ℓ positive variables to satisfy the demands. In general, in a problem with m constraints, it is required to sum the smallest m-v fixed charges where v is the highest degree of degenaracy of the problem.

Degeneracy causes difficulty in ranking vertices, therefore if a degenerate basic solution is encountered it is required to determine all the bases which represent the same extreme point.

McKeown improves Murty's algorithm by finding a better lower bound D_o. He generates a set covering problem from the fixed charge problem with the fixed charge objective function, and solves the set covering problem by relaxing the integrality of the variables as $0 \leq y_i \leq 1$.

McKeown has shown that the minimum of the relaxed version of the set covering problem is a lower bound on the fixed charges of the fixed charge problem.

Steinberg[27] proposes an exact branch and bound algorithm for the fixed charge problem. The algorithm finds the global optimum solution without having to enumerate all basic feasible solutions. The algorithm generates an enumeration tree by branching at each node assigning the variable $x_j > 0$ or $x_j = 0$. In otherwords at each node one more constraint either $x_j > 0$ or $x_j = 0$ is added to the primary convex polyhedron of the fixed charge problem. A path will be terminated when there are m additional $x_j > 0$ constraints or (n-m) additional $x_j = 0$ constraints are imposed on the initial constraints set, or no feasible solution exists when the current additional constraints are imposed. Terminating the paths when one of the three conditions stated above occurs will lead to enumeration of all extreme points of the convex polyhedron. Thus a bounding procedure to reduce the number of vertices enumerated is employed. The algorithm starts with an upper bound. A good upper bound is proposed such as an heuristic objective value. The path is fathomed when a lower bound computed is greater than the present upper bound and whenever a path is terminated with a unique basic feasible solution if the corresponding value of the objective function is less than the current upper bound it replaces the present upper bound when there is no live vertices the present upper bound and the corresponding solution is the optimal solution. The computation of the lower bound at each node requires solution of an LP problem each time with one more constraint added to the present constraint set when going down the tree.

The maximum level of degeneracy of the problem must be determined in order to find the lower bound on the fixed charges. Before starting the solution n linear programming problems are solved as below.

$$\text{(IV.2.3)} \quad \begin{aligned} &\text{Min} \quad z_j = x_j \\ &\text{s.t.} \\ &\qquad Ax = b \\ &\qquad x \geqq 0 \\ &\qquad x_j \text{ in basis} \end{aligned}$$

The number of solutions with zero optimum value determines the maximum level of degeneracy.

IV.3. BICRITERIA FIXED CHARGE NETWORK PROBLEM

A bicriterion fixed charge problem may be formulated as follows

$$\text{(IV.3.1)} \quad \text{Min} \begin{bmatrix} \sum_{ij} c_{ij} \; x_{ij} \\ \\ \sum_{ij} d_{ij} \; y_{ij} \end{bmatrix}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = b_i \qquad i=1,\ldots,m$$

$$x_{ij} \geqq 0 \qquad \forall \; (i,j)$$

$$y_{ij} = \begin{cases} 1 & x_{ij} > 0 \\ 0 & x_{ij} = 0 \end{cases}$$

we have two objectives, c_{ij} is the unit shipping cost on the arc (i,j) and d_{ij} is the fixed charge associated with the presence of the arc (i,j). It is aimed to find the efficient extreme points of the convex polyhedron defined by the flow conservation constraints. In the literature, there is no known procedure for solving such a problem. In the next section, a branch and bound algorithm for solving the fixed charge problem which is due to Steinberg[27] will be extended to two objectives. Enumeration of all efficient basic solutions of the fixed charge problem is possible by using the bicriterion branch and bound algorithm for the fixed charge problem.

Since we are dealing with fixed charge network problem, at each vertex of the branch and bound algorithm the LP subproblem can be solved by network simplex method.

IV.4. A BRANCH AND BOUND ALGORITHM FOR BICRITERIA FIXED CHARGE NETWORK PROBLEM

Initilization

Step 1)

Start at a live vertex 0 and set the initial upper bound vector $UB^o = \binom{\infty}{\infty}$ and set $q=0$. Go to Step 3.

Branching

Step 2)

· If no live vertices go to Step 7, otherwise select a live vertex (Depth first branching rule is applied) Branch to $x_{ij} = 0$ and go to Step 4.

Separation

Step 3)

Select an unassigned arc (i,j) with the maximum d_{ij} and branch to $x_{ij} > 0$.

Bounding

Step 4)

Compute the vector of lower bounds $\underline{Z}^k$. The first component $\underline{z}_1^k$ is the lower bound on the linear objective which can be found by solving the LP problem;

$$\underline{z}_1^k = \min \quad cx$$

$$Ax = b$$

$$x_{ij} \geq 1 \quad \text{for } (i,j) \in S_1^k$$

$$x_{ij} = 0 \quad \text{for } (i,j) \in S_2^k$$

$$x \geq 0$$

$$x \text{ is an extreme point of } Ax = b$$

S_1^k is the set of ares (i,j) which must be positive at vertex k and S_2^k is the set of arcs (i,j) which must be zero at vertex k.

If the LP problem has no feasible solution fathom, and go to Step 2.

The second component of $\underline{Z}^k$ which corresponds to the fixed charge objective can be calculated as follows:

Let $S_3^k = \{(i,j) \mid (i,j) \notin S_1^k \cup S_2^k\}$

$P = \{(i,j) \mid (i,j)$ can be in the basis at zero level$\}$

$Q = P \cap (S_2^k \cup S_3^k)$

N_1^k = the number elements in S_1^k

N_Q^k = the number of elements in Q

S_4^k = the set of last $m-N_1^k - N_Q^k$ arcs

Then

$$\underline{z}_2^k = \sum_{(i,j)\,\varepsilon\, S_1^k} d_{ij} + \sum_{(i,j)\,\varepsilon\, S_4^k} d_{ij}$$

Fathoming

Step 5

a) If one of the following holds fathom

i) If there arc M constraints of the form "$x_{ij} > 0$"

ii) If there are n-m constraints of the form "$x_{ij} = 0$"

iii) If $d_{ij} = 0$ for all $(i,j)\ \varepsilon\ S_3^k$

Set $\bar{Z}^k \leftarrow \underline{Z}^k$ and go to step 2.

b) If $\underline{Z}^k \geq UB^p$ for some p fathom and go to step 2, otherwise compute $\bar{Z}_k$, the first component of $\bar{Z}^k$ is $\bar{z}_1^k = z_1^k$ and the second component is the sum of d_{ij} with $x_{ij} > 0$ in the optimal solution corresponding to z_1^k, If $\bar{Z}^k = \underline{Z}^k$ fathom and go to step 2. Otherwise go to step 6.

Step 6)

If $\bar{Z}^k \geq UB^p$ for any p go to step 3, otherwise set $q \leftarrow q + 1$ and $UB^q \leftarrow \bar{Z}^k$ and store corresponding solution x^q, if $UB^p \geq \bar{Z}^k$ for any $p < q$, drop UB^p, and condense the set, $q \leftarrow q - 1$ go to Step 3.

Termination

Step 7)

If $q = 0$ no feasible solution,

If $q \geq 1$, UB^p, $p = 1,\ldots,q$ and the corresponding solutions are the efficient extreme solutions.

IV.4.1. FINDING AN INITIAL BASIC FEASIBLE SOLUTION FOR THE LP SUBPROBLEMS FOR COMPUTING THE LOWER BOUNDS

In the branch and bound algorithm at each node one more constraint of the form $x_{ij} = 0$ or $x_{ij} > 0$ is added. Thus initial basic feasible solution is required for the LP sub-problem for the computation of the lower bound on the linear objective. The following LP problem is solved to obtain an initial basic feasible solution if an additional constraint $x_{ij} = 0$ is imposed at node k.

$$\begin{aligned} \min \quad & x_{ij} \\ \text{s.t.} \quad & \\ & Ax = b \\ & x \geqq 0 \end{aligned}$$

with the additional constraints on variables $x_{ij} > 0$ for the variables made basic and $x_{ij} = 0$ for those variables which are assigned to zero. Also, x must be an extreme point of the polyhedron $Ax = b$.

When the optimal objective value for this problem is zero, either x_{ij} is removed from the basis or it is basic at zero level. Otherwise, there is no feasible solution.

However, when x_{ij} is not basic in the previous basis then the present basis remains feasible when the additional constraint $x_{ij} = 0$ is added.

If an additional constraint $x_{ij} > 0$ is imposed at node k then an LP subproblem with the same constraint set as the above problem but with the objective function as maximize x_{ij} must be solved

If there is an basic feasible solution with a positive objective value at any stage then it is not necessary to optimize since an initial basic feasible solution is obtained. If the optimum objective function value is zero then there is no feasible solution.

If the variable x_{ij} is basic in the present basic feasible solution that solution is still basic feasible when the additional constraint $x_{ij} > 0$ is added.

IV.4.2. THE SOLUTION PROCEDURE FOR THE RESTRICTED LP SUBPROBLEM

It may be required, at any node k of the branch and bound algorithm, to solve the following LP subproblem:

$$
\begin{aligned}
&\min \quad cx \\
&\text{s.t.} \\
&\qquad Ax = b \\
&\qquad x_{ij} > 0 \quad , \quad \text{for } (i,j) \in S_1^k \\
&\qquad x_{ij} = 0 \quad , \quad \text{for } (i,j) \in S_2^k \\
&\quad x \text{ is an extreme point of } Ax = b \\
&\qquad x \geqq 0
\end{aligned}
$$

The regular pivoting operations of the simplex method are performed until a stage where there are only the candidate entering variables whose corresponding leaving variables are elements of S_1^k. At this stage the current value of the objective function $\hat{z}_1$ has been obtained, but there may be an extreme point which has a better objective function value but not adjacent to the present basis. In order to determine if there is a better objective value the following check, procedure is employed.

For each $(i,j) \notin S_2^k$ or not basic in the present basis solve the following problem

$$\begin{aligned} &\max \quad x_{ij} \\ &\text{s.t.} \\ &\qquad Ax = b \\ &\qquad cx + s_1 = \hat{z}_1 \\ &\qquad x_{ij} - s_2 = \hat{x}_i \\ &\qquad x \, , \, s \geqq 0 \end{aligned}$$

and

$$\begin{aligned} &x_{ij} = 0 \, , \ (i,j) \in S_1^k \\ &x_{ij} = 0 \, , \ (i,j) \in S_2^k \\ &x \text{ is an extreme point to } Ax = b \end{aligned}$$

where $\hat{x}_{ij}$ is the current value of x_{ij}, initially zero.

At each step the minimum increase in x_{ij} is taken and if at any stage $s_1 > 0$, the procedure is terminated and a new extreme point is obtained. The regular simplex iterations can start at this basis.

IV.4.3. COMPUTING THE LOWER BOUNDS

When proceeding down the tree it is not always required to solve the LP subproblem in order to check whether the lower bound at node k exceeds the one of the present upper bounds (i.e UB^p s). Any one of the following cases may occur at node k.

i) If $x_{ij} > 0$ is the additional constraint and x_{ij} is already basic in the present basis $\underline{z}_1^k$ the first component of the lower bound vector is the same as the objective function value of the present basis. The second component $\underline{z}_2^k$ can be calculated as follows:

$$\underline{z}_2^k = \sum_{(i,j) s_1^k} d_{ij} + \sum_{(i,j) \varepsilon s_4^k} d_{ij}$$

ii) If $x_{ij} > 0$ is the additional constraint and x_{ij} is not basic in the present basis compute $\underline{z}_2^k$ the second component of the lower bound vector. Set $\underline{z}_1^k$ as the same as the objective value of the present basis. If this $\underline{z}^k \geqq UB^p$ for some p fathome and go to step 2. Otherwise, solve the LP subproblem.

iii) If $x_{ij} = 0$ is the additional constraint and x_{ij} is basic in the present basis apply the same procedure of the step 4 of the algorithm.

iv) If $x_{ij} = 0$ is the additional constraint and it is not basic in the present basis the first component of the lower bound vector is the same as the objective function value of the present basic solution. The second component needs to be computed as in i) above.

IV.5. EXAMPLE PROBLEM

To illustrate the branch and bound algorithm the example problem belov is solved.

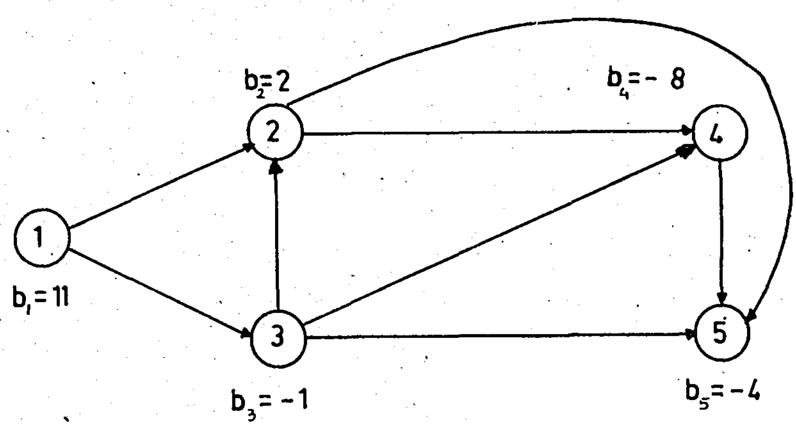


Figure IV.1

$d_{ij} = -100 \times \ell n\ (P_{ij})$ is substituted

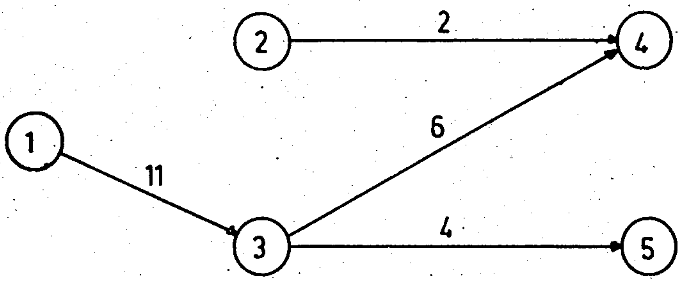
$$\begin{array}{cc} & \begin{array}{cccccccc} (1,2) & (1,3) & (2,4) & (2,5) & (3,2) & (3,4) & (3,5) & (4,5) \end{array} \\ c = & \begin{bmatrix} 1 & 2 & 5 & 4 & 1 & 3 & 1 & 2 \end{bmatrix} \\ d = & \begin{bmatrix} 3.04 & 2.02 & 12.78 & 4.08 & 8.33 & 18.63 & 24.86 & 5.12 \end{bmatrix} \end{array}$$

Given the network in Fig.III.2, the unit cost vector c, and the fixed charge vector d the problem is to find the set of all efficient extreme points.

The solution procedure

$k = 1$

$x_{35} > 0$ is the additional constraint



$$\left.\begin{aligned} Z_{12}-C_{12} &= -1 \\ Z_{25}-C_{25} &= -1 \\ Z_{32}-C_{32} &= -3 \\ Z_{45}-C_{45} &= -4 \end{aligned}\right\} \Rightarrow \text{optimal}$$

Figure IV.2.

$\underline{z}_1^1 = 54$

$\underline{z}_2^1 = 24.86 + (4.08 + 3.04 + 2.02) = 34$

$\bar{z}_1^1 = 54$

$\bar{z}_2^1 = 2.02 + 12.78 + 18.63 + 24.86 = 58.29$

$UB^1 = \binom{54}{58.29}$, $UB^0 \geq \bar{z}^1$, Drop UB^0.

$k = 2$

$(x_{35} > 0)$, $x_{34} > 0$ are additional constraints,

x_{34} is basic.

$\underline{z}_1^2 = 54$

$\underline{z}_2^2 = 24.86 + 18.63 + (3.04 + 2.02) = 48.55$

$\bar{z}_1^2 = 54$

$\bar{z}_2^2 = 58.29$

$k = 3$

$(x_{35} > 0,\ x_{34} > 0)$ and $x_{24} > 0$ are additional constraints

x_{24} is basic,

$\underline{z}_1^3 = 54$

$\underline{z}_2^3 = 24.86 + 18.63 + 12.78 + (2.02) = 58.29$

$\bar{z}_1^3 = 54$

$\bar{z}_2^3 = 58.29$

$\bar{Z}^3 = \underline{Z}^3$, fathom.

$k = 4$

$(x_{35} > 0,\ x_{34} > 0)$ and $x_{24} = 0$ are additional constraints, x_{24} is basic in the present basis.

To make x_{24} nonbasic solve

$$\begin{array}{ll} \min & x_{24} \\ \text{s.t.} & Ax = b \\ & x_{ij} > 0 \ (i,j) \in S_1^4 \\ & x_{ij} = 0 \ (i,j) \in S_2^4 \\ & x \text{ is an extreme point.} \end{array}$$

$w_5 = w_4 = w_3 = w_1 = 0,\ w_2 = 1$

$z_{12} - c_{12} = -1$

$z_{25} - c_{25} = 1 \rightarrow$ enter $(2,5) \rightarrow$

$z_{32} - c_{32} = -1$

$z_{45} - c_{45} = 0$

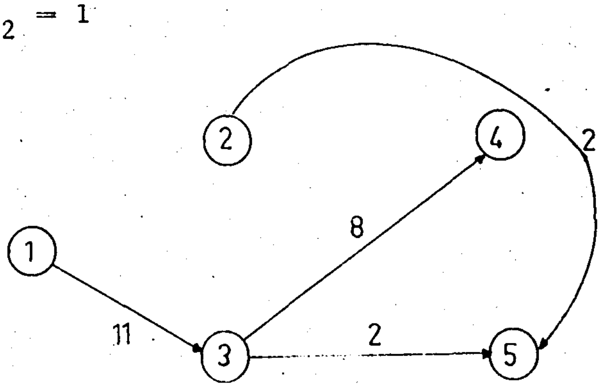


Figure IV.3

Now, solve the LP $\quad \min \quad cx$

$$\text{s.t}$$

$$Ax = b$$

$$x_{ij} > 0 \quad (i,j) \in S_1^4$$

$$x_{ij} = 0 \quad (i,j) \in S_2^4$$

x is an extreme point.

$z_{12} - c_{12} = 3 - 4 - 1 = -2$

$z_{24} - c_{24} = 4 \quad 2 - 5 = 1 \quad \rightarrow \quad x_{24}$ should remain nonbasic

$z_{32} - c_{32} = 1 - 4 - 1 = -4$

$z_{45} - c_{45} = -2 - 0 - 2 = -4$

$\underline{z}_1^4 = cx = 56$

$\underline{z}_2^4 = 24.86 + 18.63 + (3.04 + 2.02) = 48.55$

$\bar{z}_1^4 = 56$

$\bar{z}_2^4 = 2.02 + 4.08 + 18.63 + 24.86 = 49.59$

$\bar{Z}^4 \not\leq UB^p$ for any $p \rightarrow UB^2 \leftarrow \bar{Z}^4 = \binom{56}{49.59}$

$k = 5$

$(x_{35} > 0 \ , \ x_{34} > 0 \ , \ x_{24} = 0)$ and $x_{32} > 0$ are additional constraints.

x_{32} is not basic in the present basis.

$\underline{z}_2^5 = 24.86 + 18.63 + 8.33 + (2.02) = 53.83$

$\underline{z}_2^5 \geq 49.59$, since $\underline{z}_1^5 \geq 56$ fathom.

$k = 6$

$(x_{35} \geq 1 , x_{34} \geq 1 , x_{24} = 0)$ and $x_{32} = 0$ are additional constraints

x_{32} is not basic $\rightarrow$ no change in the lower bounds

$k = 7$

$(x_{35} > 0, x_{34} > 0 , x_{24} = 0 , x_{32} > 0)$ and $x_{45} > 0$ are additional constraints.

In order to make x_{45} basic x_{35} is has to leave, therefore, fathom.

$k = 8$

$(x_{35} > 0 , x_{34} > 0 , x_{24} = 0 , x_{32} = 0)$ and $x_{45} = 0$ are additional constraints.

x_{45} is not basic in the present basis.

$\underline{Z}^8 = \binom{56}{48.55}$, $\bar{Z}^8 = \binom{56}{49.59}$

$k = 9$

$(x_{35} > 0 , x_{34} > 0,, x_{24} = 0 , x_{32} = 0 , x_{45} = 0)$ and $x_{25} >' 0$ are additional constraints.

x_{25} is basic in the present basis;

$\underline{z}_1^9 = 56$

$\underline{z}_2^9 = 24.86 + 18.63 + 4.08 + (2.02) = 49.59$

$\bar{Z}^9 = \binom{56}{49.59}$

$\bar{Z}^9 = \underline{Z}^9$, fathom.

$k = 10$

$(x_{35} > 0\ ,\ x_{34} > 0\ ,\ x_{24} = 0\ ,\ x_{32} = 0\ ,\ x_{45} = 0)$ and $x_{25} = 0$ are additional constraints.

x_{25} is basic in the present basis.

Solve the LP problem to minimize x_{25} in order to make it nonbasic.

$$w_5 = w_4 = w_3 = w_1 = 0\ ,\ w_2 = 1 \rightarrow z_{24} - c_{24} = 1$$

Infeasible, fathom, infact there are 4 constraints of the form $x_{ij} = 0$.

$k = 11$

$(x_{35} > 1)$ and $x_{34} = 0$ are the additional constraints present basis :

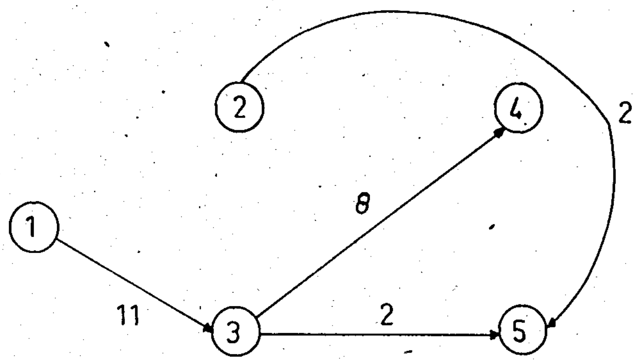


Figure IV.4

x_{34} is basic make it nonbasic by solving LP to minimize x_{34}.

$x_{34} = 8$, $w_5 = w_2 = w_3 = w_1 = 0$, $w_4 = -1$

$z_{24} - c_{24} = 0 + 1 - 0 = 1 \rightarrow$ enter (2,4)

$\Delta = \min \{8,\infty,2\} = 2$

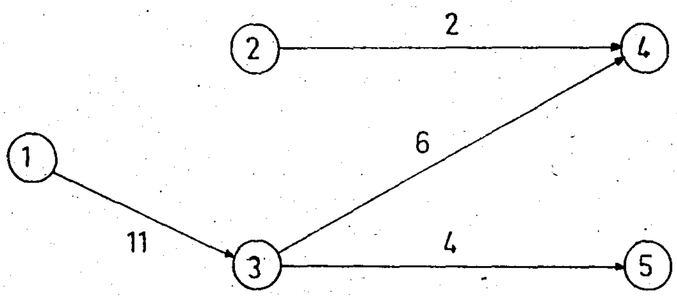


Figure IV.5

$x_{34} = 6$, $w_5 = w_3 = w_1 = 0$, $w_4 = w_2 = -1$, $z_{32} - c_{32} = 1$

enter (3.2) $\Delta = \min \{\infty,6\}$

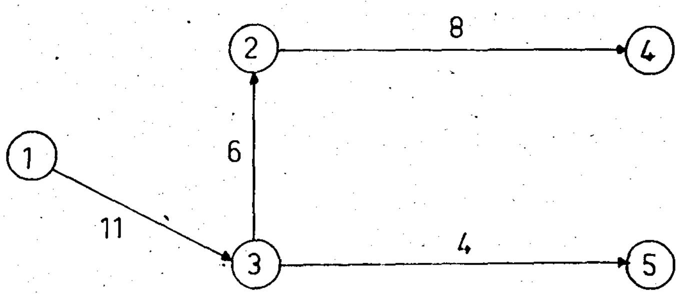


Figure IV.6

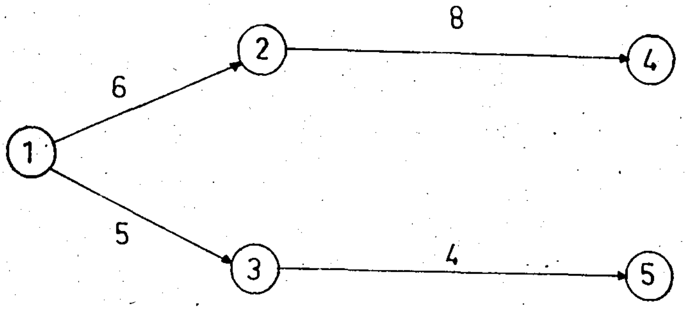
Initial feasible basis is obtained, solve LP to minimize cx.

$z_{12} - c_{12} = 3 - 0 - 1 = 2 \rightarrow$ enter (1,2), $\Delta = \min \{6,11\} = 6$

$z_{25} - c_{25} = 0 - 0 - 4 = -4$

$z_{35} - c_{35} = 1 + 5 - 3 = 3$

$z_{45} - c_{45} = -5 - 0 - 2 = -7$



$cx = 72 - 12 = 60$

Figure IV.7

$z_{25} - c_{25} = 1 - 0 - 4 = -3$

$z_{32} - c_{32} = 0 - 1 - 1 = -2$

$z_{34} - c_{34} = 0 \quad 4 - 3 = 1$

$z_{45} - c_{45} = -4 - 0 - 2 = -6$

$\underline{z}_1^{11} = 60$

$\underline{z}_2^{11} = 24.86 + (4.08 + 3.04 + 2.02) = 34$

$\bar{Z}^{11} = \binom{60}{42.7}$

$\bar{Z}^{11} \not\leq UB^p$ for any p , $UB^3 = \binom{60}{42.7}$

$UB^p \not\leq \bar{Z}^{11}$ for any $p < q$.

$k = 12$

$(x_{35} > 0 \; , \; 0_{34} = 0)$ and $x_{24} > 0$ are additional constraints.

$x_{24} > 0$ is basic in the present solution

$\underline{z}_1^{12} = 60$

$\underline{z}_2^{12} = 24.86 + 12.78 + (3.04 + 2.02) = 42.7$

$\bar{Z}^{12} = \binom{60}{42.7}$

$\bar{Z}^{12} = \underline{Z}^{12}$, fathom.

$k = 13$

$(x_{35} > 0 \ , \ x_{34} = 0)$ and $x_{24} = 0$ are additional constraints.

x_{24} is basic, in order to make it non basic; solve LP to minimize x_{24}.

$w_5 = w_3 = w_1 = w_2 = 0 \ , \ w_4 = -1 \ , \ z_{34} - c_{34} = 0 + 1 - 0 = 1$

x_{34} should enter but $x_{34} = 0$ is an additional constraint, fathom because of infesibility.

$k = 14$

$x_{35} = 0$ is the only additional constraint.

The present basis is; x_{35} is basic.

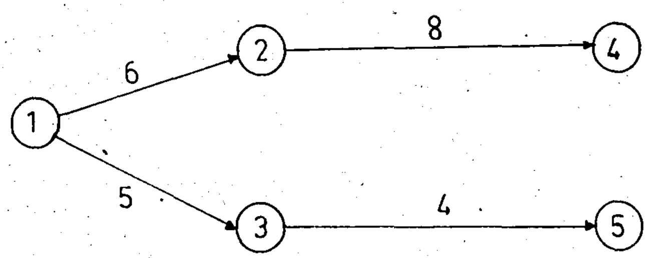


Figure IV.8

Solve LP to minimize x_{35}.

$w_5 = 0 \ , \ w_3 = w_1 = w_2 = w_4 = 1$

$z_{25} - c_{25} = 1 \rightarrow$ enter $(2,5) \rightarrow \Delta = \{4,5,\infty\} = 4$

$z_{45} - c_{45} = 1$

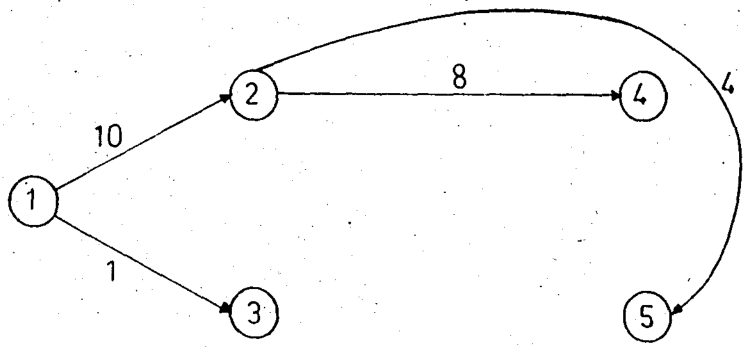


Figure IV.9

In order to find the lower bound z_1 minimize cx.

$z_{32} - c_{32} = 3 - 4 - 1 = -2$

$z_{34} - c_{34} = 3 \quad 1 - 3 = 1 \rightarrow$ enter (3,4)

$z_{35} - c_{35} = 3 - 0 - 1 = 2$

$z_{45} - c_{45} = -1 - 0 - 2 = -3$

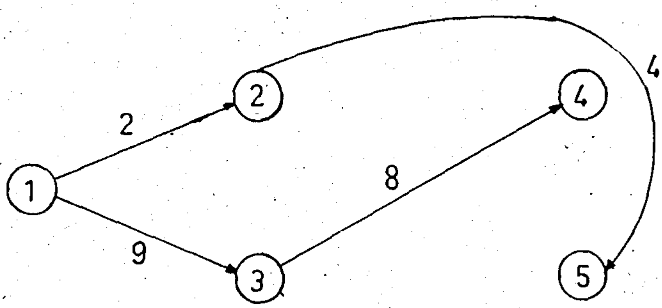


Figure IV.10

$cx = 60$

$z_{24} - c_{24} = 4 - 0 - 5 = -1$

$z_{32} - c_{32} = 3 - 4 - 1 = -2$

$z_{35} - c_{35} = 3 - 0 - 1 = 2$

$z_{45} - c_{45} = 0 - 0 - 2 = -2$

$\underline{z}_1^{14} = 60$

$\underline{z}_2^{14} = (5.12 + 4.08 + 3.04 + 2.02) = 14.26$

$\bar{z}_1^{14} = 60$

$\bar{z}_2^{14} = 3.04 + 2.02 + 18.63 + 4.08 = 27.77$

$\bar{Z}^{14} \not\leq UB^p$ for any p

$UB^3 \geq \bar{Z}^{14}$, drop UB^3 , $UB^3 \leftarrow \binom{60}{27.77}$

$k = 15$

$(x_{35} = 0)$ and $x_{34} > 0$ are additional constraints.

x_{34} is basic in the present basis.

$\underline{z}_1^{15} = 60$

$\underline{z}_2^{15} = 18.63 + (4.08 + 3.04 + 2.02) = 27.77$

$\bar{z}_1^{15} = 60$

$\bar{z}_1^{15} = 27.77$

$\bar{Z}^{15} = \underline{Z}^{15}$, fathom.

$k = 16$

$(x_{35} = 0)$ and $x_{34} = 0$ are additional constraints

x_{34} is basic in the present basis when it is made nonbasic the following basis is obtained.

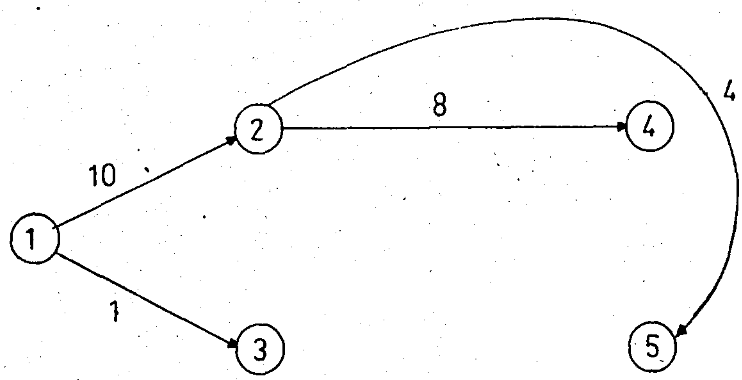


Figure IV.11

$\underline{z}_1^{16} = 68$

$\underline{z}_2^{16} = 14.26$

$\bar{Z}^{16} = \binom{68}{21.92}$, $\bar{Z}^{16} \ngeq UB^p$ for any p , $UB^4 = \binom{68}{21,92}$

$UB^p \nleq \bar{Z}^{16}$ for any $p < q$

$k = 17$

$(x_{35} = 0\ ,\ x_{34} = 0)$ are additional constraints, x_{24} is basic in the present basis.

$\underline{Z}^{17} = \binom{68}{21.92}\ ,\ \bar{Z}^{17} = \binom{68}{21.92}$

$\bar{Z}^{17} = \underline{Z}^{17}$, fathom.

$k = 18$

$(x_{35} = 0\ ,\ x_{34} = 0)$ and $x_{24} = 0$ are additional constraints.

x_{24} basic, solve LP to minimize x_{24} in order to make it nonbasic.

x_{34} is entering variable but $x_{34} = 0$ then fathom because of infeasibility.

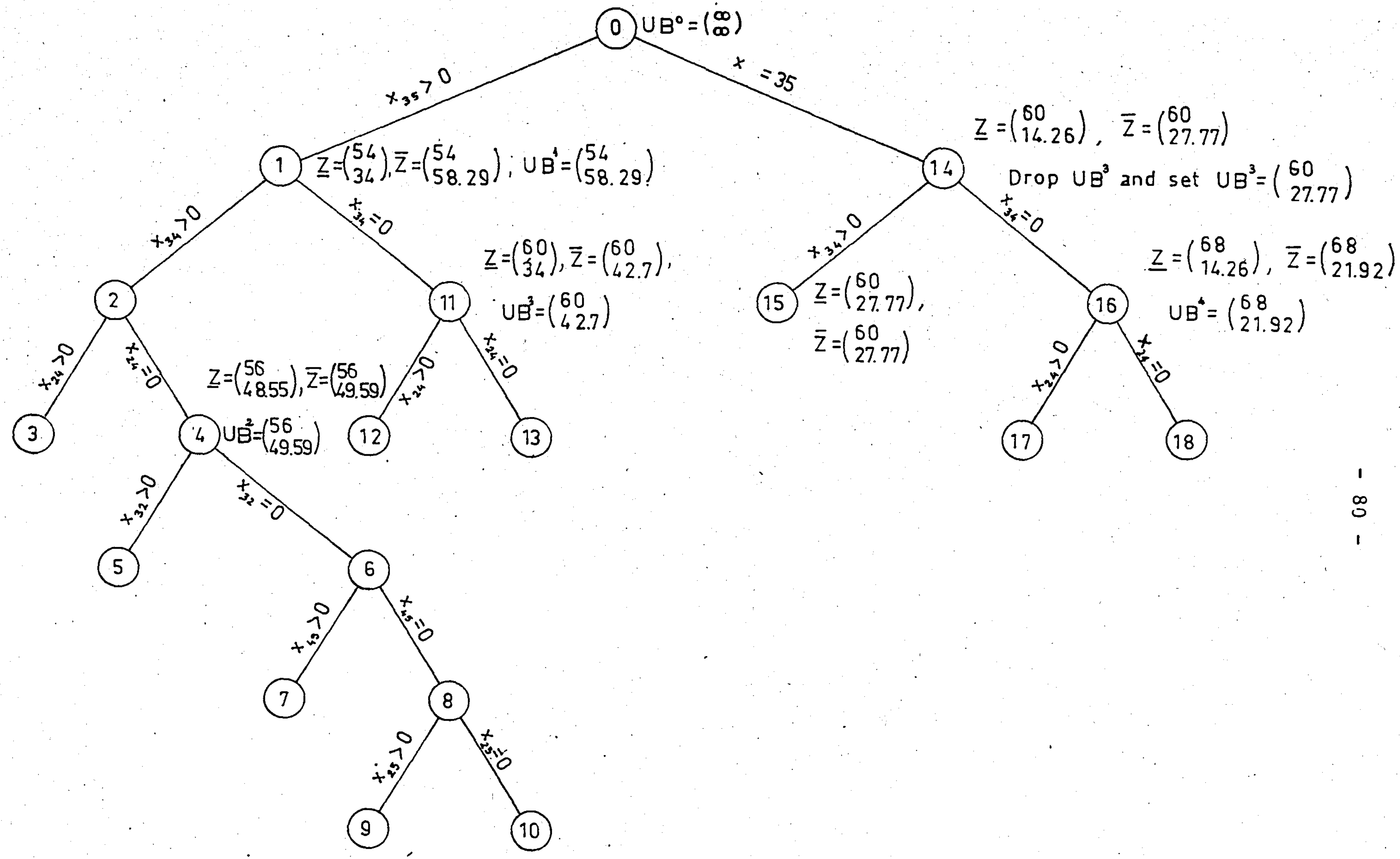


Figure IV.12. The solution tree for the example problem.

IV.6. SOME ADDITIONAL CONSIDERATIONS

Some additional considerations on the branch and bound algorithm can be listed as follows:

1. THE BRANCHING RULE

The depth first branching rule is selected in the algorithm. This rule provides a more efficient use of the present basis than the breath first branching rule. The breadth first rule may generate more and diverse upper bounds and increase the probability of fathoming in the subsequent vertices. Therefore, initially calculating the upper bound for the first two branches provides two different upper bounds for the following vertices when the depth first rule is applied.

2. A PROCEDURE FOR CALCULATION OF THE LOWER BOUND ON THE FIXED CHARGES

The method proposed by McKeown[22] can be employed at each vertex by solving the following LP problem:

$$\min \sum_{(i,j)} d_{ij}\, y_{ij}$$

(IV.6.1)

$$\sum_{(i,j)} \delta_{i,(i,j)}\, y_{ij} \geq \beta_i$$

$$0 \leq y_{ij} \leq 1$$

where

$$\delta = \begin{cases} 1 & \text{if} \quad a_{ij} > 0 \\ 0 & \text{if} \quad a_{ij} \leq 0 \end{cases} \quad \text{and} \quad \beta_i = \begin{cases} 1 & \text{if} \quad b_i > 0 \\ 0 & \text{if} \quad b_i = 0 \end{cases}$$

The minimum value of this problem gives the lower bound on the fixed charge function. Since the feasibility requirement is added then tighter lower bounds than the method used in the algorithm can be found.

3. ALTERNATIVE EFFICIENT SOLUTIONS

The fathoming condition $\overline{Z}^k = \underline{Z}^k$ allows to find only one efficient solution with the same objective values. If it is required to generate the alternative efficient solutions it requires more computational time. When the example in the previous section is solved without this fathoming rule the solution tree in figure IV.13 is obtained. The number of vertices in the branch and bound tree increase from 18 to 42 and in fact, there is no alternative efficient solutions.

4. A DIFFERENT SOLUTION PROCEDURE FOR SOLVING THE RESTRICTED LP SUBPROBLEM

The network simplex method with lower and upper bounds can be used. Since the solutions to the network problem are integer we can set lower value 1 for each variable which appears as $x_{ij} > 0$ in the branch and bound process. If the only candidate entering variables require any x_{ij} which is assigned as $x_{ij} > 0$, it is allowed to be nonbasic at it is lower bound and the lower bound may be calculated. But this lower bound is looser than the one which may be found by the method described in section IV. Also there will be additional storage requirements and difficulties associated with finding an initial basic feasible solution at each vertex k.

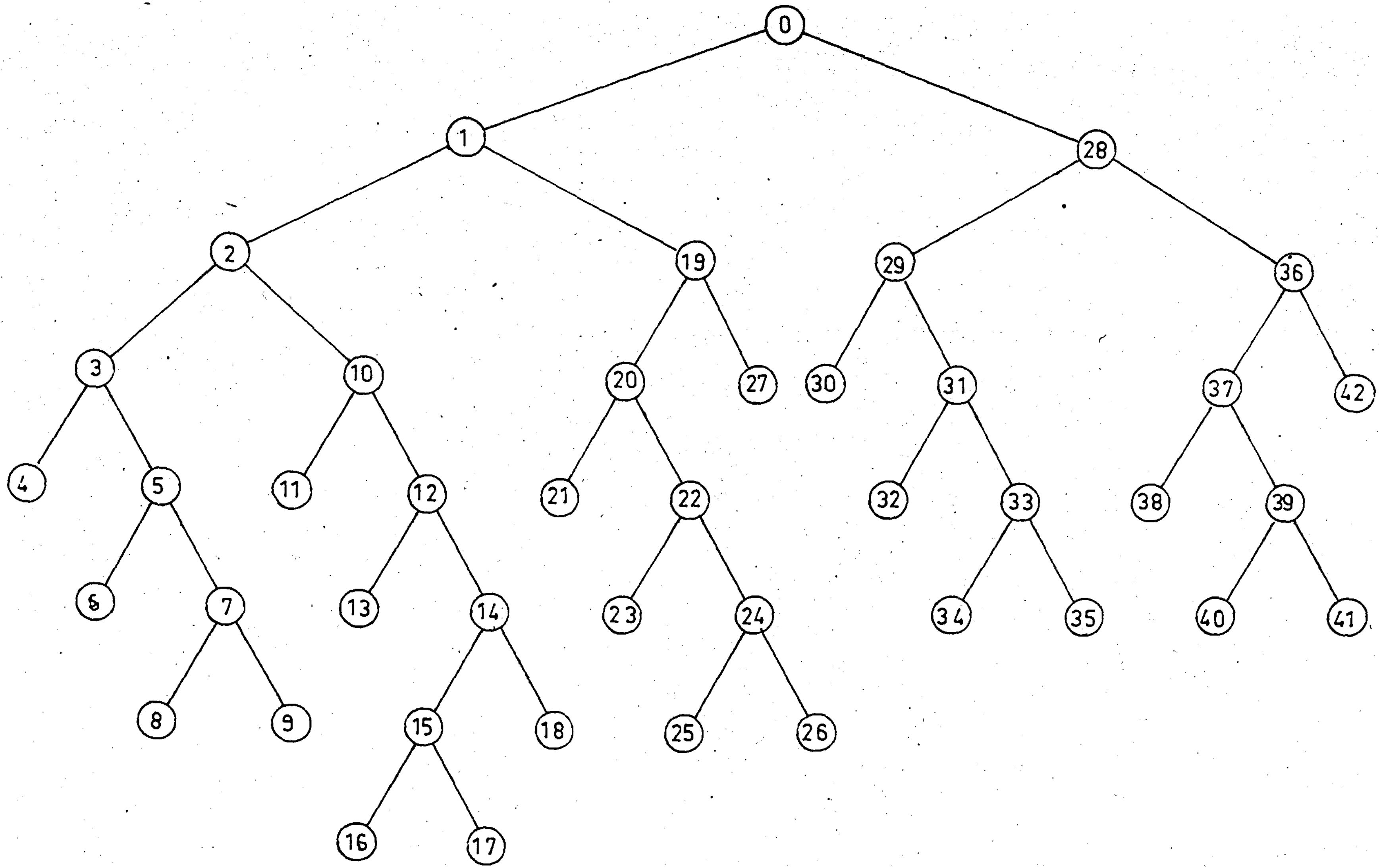


Figure IV.13. The solution tree for the example problem when the alternative efficient solutions are not ignored.

CHAPTER V
SUMMARY AND EXTENSIONS

In this thesis, solution procedures for multiobjective network problems with linear objectives and fixed charge objectives are developed.

For the multiobjective linear network problems the network simplex algorithm is extended into a multiobjective algorithm which includes an efficiency check procedure. This algorithm is also extended to an algorithm with lower and upper bounds. Only the efficient extreme points are generated by using the labeling algorithm for the multiobjective linear network problems. The algorithm can be extended in order to generate the efficient edges and faces of the convex polyhedron.

A branch and bound algorithm is given for finding the efficient extreme points to the bicriteria fixed charge problem. It seems that in the general case where all fixed charges are greater than zero, the set of efficient points will consist only of extreme points. A point on an edge or face will include fixed charges of all variables at a positive level and will be dominated with respect to the fixed charge objective by any one of the extreme points of the face. Then the extreme point of the face with the minimum value of the first objective will dominate all interior points. Only if the value of the first objective remains constant over the face and the fixed charges associated with variables which are not common in the bases representing the extreme points of the face are zero, the points on the face will be efficient. It would be worthwhile to develop a formal proof and to extend the algorithm to incorporate this consideration.

In the branch and bound procedure the computational testing must be done. Different branching rules may be applied, the efficiency of the breadth first rule may be checked. A better method for determination of the maximum level of degeneracy may be employed. Also improvements on the computation of the lower bounds both on the linear objective and the fixed charges can be developed. The algorithm may also be extended to the multiobjective fixed charge problems without increasing the number of subproblems to be solved at each vertex, i.e. instead of solving one subproblem for each linear objective, it could be possible to extent to the multiobjective case by solving one MOLP.

REFERENCES

1- Aneja,Y.P. and K.P.K.Nair, "Bicriteria Transportation Problem", Management Science, Vol.25, 73-78, 1979.

2- Balinski,M.L., "Fixed Cost Transportation Problem", Naval Research Logistics Quarterly, Vol.8, 41-54, 1961.

3- Bazaraa,M.S. and J.J.Jarvis, Linear Programming and Network Flows, John Wiley and Sons, 1977.

4- Bod,P., "Solution of a Fixed Charge Linear Programming Problem", Proceedings Symp.Math.Programming, Princeton, 1967, (H.W.Kuhn, ed), pp.367-375, Princeton Univ.Press, Princeton, New Jersey.

5- Cochrane J.L. and M.Zeleny (Eds), Multiple Criteria Decision Making, University of South Carolina Press, Columbia, South Carolina, 1973.

6- Cohon,J.L. and D.H.Marks, "Multiobjective Screening Models and Water Resources Investment", Water Resources Research, Vol.9, 826-836, 1973.

7- Cooper L. and C.Drebes, "An Approximate Solution Method for the Fixed Charge Problem", Naval Research Logistics Quarterly, Vol.14, 101-113, 1967.

8- Ecker,J.G. and I.A.Kouada, "Finding All Efficient Extreme Points for Multiple Objective Linear Programs", Mathematical Programming, Vol.14. 249-261, 1978.

9- Egberg,D.J., J.L.Cohon, and C.S.ReVelle, "Multiobjective Sitting Model of A Natural Gas Pipeline System" Modeling and Simulation Vol.11, Part 4: Socio-Economics Proceedings of the Elevent Annual Pittsburg Conference, School of Engineering, Univ.of Pittsburg, 1980.

10- Elmagraphy,S.E., Some Network Models in Management Science, Springer-Verlag, Berlin-Heidelberg, 1970.

11- Evans,J.P. and R.E.Steuer, "A Revised Simplex Method for Linear Multiple Objective Programs", Mathematical Programming, Vol.5, 54-72, 1973.

12- Frank,H. and I.T.Frisch, Cummunication, Transmission and Transportation Networks, Addison-Wesley, New York, 1971.

13- Ford,L.R. and D.R.Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey, 1962.

14- Garfinkel,R.S. and G.L.Nemhauser, Integer Programming, John Wiley and Sons, New York, 1972.

15- Glover,F., D.Klingman and J.Stutz, "An Augmented Threaded Index Method for Network Optimization", INFOR, Vol.12, 293-298, 1974.

16- Haimes and Hall, "Multiobjectives in Water Resource Systems Analysis: The Surrogate Worth Trade-off Method", Water Resources Research, Vol.10, 615-624, 1974.

17- Isermann,H., "The Enumeration of the Set of All Efficient Solutions for a Linear Multiple Objective Program", Operational Research Quarterly, Vol.28, 711-725, 1977.

18- ————————, "The Enumeration of All Efficient Solutions For a Linear Multiple-Objective Transportation Problem", Naval Research Logistics Quarterly, Vol.26, 123-139, 1979.

19- Kızıltan,G., Multiobjective Linear Programming and Multiobjective Zero-One Programming, PhD.Dissertation, Boğaziçi University, 1981.

20- Lawler,E., Combinatorial Optimization Networks and Matroids, Holt, Rinehart and Winston, 1976.

21- Lee,S.M., I.G.Green, and C.S.Kim, "A Multiple Criteria Model for the Location-Allocation Problem", Computers and Operations Research, Vol.8, 1-8, 1981.

22- McKeown,P., "A Vertex Ranking Procedure for Solving the Linear Fixed Charge Problem", Operations Research, Vol. 23, 1183-1191, 1975.

23- Moore,B., W.Taylor, III, and S.M.Lee, "Analyses of a Transshipment Problem with Multiple Objectives", Computers and Operations Research, Vol.15, 39-46, 1978.

24- Murty,K.G., "Solving the Fixed Charge Problem by Ranking the Extreme Points", Operations Research, Vol.16, 268-279, 1968.

25- Philip,J., "Algorithms for the Vector Maximum Problem", Mathematical Programming, Vol.2, 207-229, 1972.

26- Srinivasan,V. and Thompson,G.L., "Determining Cost vs. Time Pareto-Optimal Frontiers in Multi-Modal Transportation Problems", Transportation Science,Vol.11, 1-19, 1977.

27- Steinberg,D., "The Fixed Charge Problem", Naval Research Logistics Quarterly, Vol.17, 217-236, 1970.

28- Steuer,R.E., "Multiple Objective Linear Programming with Interval Criterion Weights", Management Science, Vol.23, 305-316, 1976.

29- Taha,H.A., "Concave Minimization Over a Convex Polyhedron" Naval Research Logistic Quarterly, Vol.20, 533-548, 1973.

30- Thuente,D.J., "Two Algorithms for Shortest Paths Through Multiple Criteria Networks" Presented at the Atlanta ORSA/TIMS Meeting, 1979.

31- Zeleny,M., Linear Multiobjective Programming, Springer-Verlag, Berlin, 1974.

32- Zionts,S., "Multiple Criteria Decision Making: An Overview and Several Approaches", Working Paper. No 454, School of Management, State University of New York at Buffalo, 1980.