TR/01/94 **01/02/94**

NETWORK PROBLEMS

and Algorithms

Farhad Djannaty

Dr K Darby Dowman

w9253493

Contents

Abstract

Special structure linear programming problems have received considerable attention during the last two decades and among them network problems are of particular importance and have found numerous applications in management science and technology.

The mathematical models of the shortest route, maximal flow, and pure minimum cost flow problems are presented and various interrelationships among them are investigated. Finally three algorithms due to Dijkstra and Ford and Fulkerson which deal with the solution of the above three network problems are discussed.

1 Introduction

One of the important techniques of mathematical programming which has been developed and extended during recent years is network optimization. This powerful method of optimization can be applied to numerous real life problems in many areas such as production, distribution, project planning, facility location, resource management, and financial planning. Many classes of network problems can be modeled as linear programming problems with a special structure. By exploiting this structure, a number of special purpose algorithms have been proposed which can solve large problems more efficiently than would be the case if the general purpose simplex method is used. In addition a general algorithm is available which deals with the minimum cost flow problem and it provides a framework within which a number of special types of the network problems can be solved. Reseaech into developing efficialgorithms to solve network problems of greater complexity is ongoing. Such problems although they may not be pure network problems either contain network sub-structures or are such that a network representation is a potentially useful relaxation.

1.1 Definition of a network

A network G = (N,A) comprises two sets N = {1,2,...n} and A = {(i,j);i,j $\in$ N}. where N is called the set of *nodes, vertices, or points* and A is called the set of *arcs, edges, or links.* In addition if an orientation is specified on each arc of a network then it is called a directed network otherwise it is termed an undirected network.

A pictorial representation is often used to represent a network in which a circle denotes a node and a line indicates an arc and if the network is directed the orientation is shown by an arrowhead on each arc. An ordered pair (i,j) is

used to define the arc emanating at node i and terminating at node j. Figure 11.is a small directed network which consists of 6 nodes and 9 arcs. In this example N = {1,2,3,4,5,6} and A = {(1,2),(1,3),(2,4),(2,5),(3,4),(3,5),(4,5),(4,6),(5,6)}.

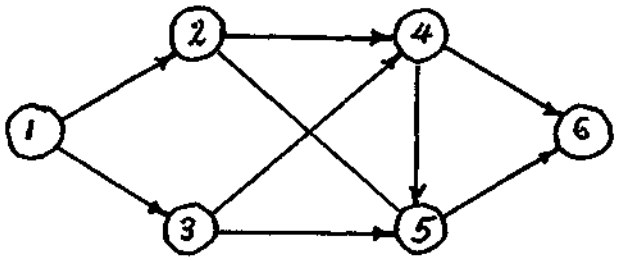


Figure 1.1

1.2 Outline of the structure of the report

In section 2 a number of special cases of network problems are discussed. These cases are known as *Transportation problem, Transshipment problem, Assignment problem, Shortest route problem, Maximal flow problem, Minimal spanning tree problem, and the General pure minimum cost flow problem* . The interrelationship between these well known standard problems, and the property of unimodularity are presented in section 3. Two special purpose algorithms for the shortest route and maximal flow are presented in section 4.A general algorithm called the Out of Kilter algorithm which deals with the general pure minimum cost flow problem is presented in section 5.

2 Network problems

2.1 Transportation problem

Transportation is an important part of each society in which many decisions must be made. In the search for optimal decisions, mathematical modeling for transport network optimization can be of great help.

A common problem within the application area is to find a transportation plan which minimizes the total cost of shipping goods from a specified set of source points(such as plants, warehouses, cities) to a predetermined number of destination points(such as customers, warehouses, cities),subject to the supply availabilies and demand requirements. In this special kind of network problem it is assumed that all flows are sent directly from sources to destinations.

To formulate the mathematical model of the transportation problem assume there are m sources and n destination points. Let s_i be the amount of commodity supplied at source i (i=l,2,...,m) ,and d_j is the amount demanded at destination j (j=l,2,...n).The per unit cost of transporting the commodity from source i to the destination j is assumed to be c_{ij} Let x_{ij} be the decision variable indicating the number of units to be supplied from the source i to destination j. It is further assumed that the total amount of commodity supplied at all source points is equal to the total amount demanded at all destination points, that is $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$ .This condition is known as the *balance condition* for the transportation problem.If this condition is not satisfied by the problem the mathematical model of the problem must be modified, by introducing a dummy source or a dummy destination.

The mathematical model of the transportation problem can be stated as below:

$$Min\, z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

(model2.1)

$$\sum_{j=1}^{n} x_{ij} = s_i \qquad (i = 1,2,...n)$$

$$\sum_{i=1}^{m} x_{ij} = d_j \qquad (j = 1,2,...n)$$

$$x_{ij} \geq 0, (i = 1,2,...m \quad j = 1,2,...n)$$

2.2 Transshipment problem

One of the assumptions of the transportation problem is that there are no intermediate points between the set of sources and the set of destination points and that the supply and demand points can not act like intermediate warehouses. If the indirect transportation of goods through intermediate points such as warehouses is permitted, it may be cheaper, rather than directly going from point i to point j to go from i to j via an intermediate point k. This extension of the transportation problem is called the *transshipment problem.*

Consider a given network with n nodes with at least one source and at least one demand node. Let f_{ij} be the flow or goods sent along arc (i,j) and C_{ij} be the per unit cost of this flow. Let s_i the amount of flow or commodity supplied at the souce node i and d_i be the amount of flow demanded at the destination i. Provided that the summations are taken only over the existing arcs, the mathematical model of the problem can be stated as follows:

$$Minimize \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} f_{ij}$$

s.t. (model 2.2)

$$\sum_{j=1}^{n} f_{ij} - \sum_{k=1}^{n} f_{ki} = \left\{ \begin{array}{ll} s_i & \text{if i is a source node} \\ -d_i & \text{if i is a destination node} \\ 0 & \text{otherwise} \end{array} \right\} \; i=1,2\ldots n.$$

$$f_{ij} \geq \forall\, i,j$$

2.3 The assignment problem

The assignment problem deals with the allocation of resources(e.g. employees, machines) to activities (e.g. jobs, locations)on the basis of a one to one cor-

respondence between resources and activities. The number of resources (assignees) and activities (assignments) are assumed to be equal. If this balance equation does not hold for the assignment problem it can be restored by adding suitable dummy assignees or assignments. Let x_{ij} (i=l,2,...,n),(j=l,2,...,n) be a decision variable such that if x_{ij} =1 resource i is assigned to activity j and x_{ij}=0 otherwise. Let c_{ij} denote the cost of this assignment. Then the mathematical model stated as follows:

$$Min \quad z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \qquad \text{(model 2.3)}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, 2, ..., n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, 2, ..., n$$

$x_{ij} \geq 0$ for all i and j

x_{ij} is binary for all i and j

It can be seen that this model is a special case of the transportation problem in which the number of sources equals the number of destinations and with unit supplies and demands. Although the problem can be solved as in transportation problem, more specialized algorithms which take advantage of the special structure of the assignment problem have been developed.

2.4 Minimal spanning tree problem

To provide for stating the definition of the spanning tree some network terminology is necessary.

A Path between two nodes is a sequence of arcs which connects these two nodes. A path can be directed or undirected.

A *cycle* is a path which emanates from a node and ends at the same node. Two nodes are said to be *connected* if the network contains at least one undirected path which joins them. A network is called a connected network if each pair of nodes is connected.
A *Tree* is a network which contains no cycle.

A *spanning tree* is a tree which includes all nodes of the network. In a network with n nodes a spanning tree must contain exactly n-1 arcs.

Let each arc of a network have a parameter such as time, cost, or length associated with it. The *weight* of a tree is defined to be the sum of the parameters associated with all arcs in the tree. Therefore a *minimal spanning tree* is one which has the minimum weight.

The concept of minimal spanning tree is very useful and has numerous applications in transportation, electrical engineering, communication systems and as a subproblem within more complex problems. This problem can be solved efficiently by a special purpose algorithm of low complexity.

2.5 The shortest route problem

This problem deals with finding the shortest path from a source node to any other node including the sink of a network. Many important Industrial problems can be formulated as the shortest route problem including *equipment replacement, capital investment, project scheduling, inventory and planning.* The shortest route problem often occurs as a subproblem in more complex problems. Let c_{ij} be the distance, time, or cost between nodes i and j and let x_{ij}, be a binary decision variable indicating the selection of the arc i-j in the shortest route then the mathematical model of the problem is as follow:

$$Minimize\, z = \sum_i \sum_j c_{ij} x_{ij}$$

s.t (model 2.4)

$$\sum_j x_{ij} - \sum_k x_{ki} = \begin{cases} 1 & \text{if node i is the source node} \\ -1 & \text{if node i is the sink node} \\ 0 & \text{for all nodes other than source and sink} \end{cases} \quad (i=1,2,...n)$$

$$x_{ij} \geq 0 \; \forall \; i,j$$

Note that the above summations are over the arcs such that (i,j) is an arc in the network. Although the problem is one of finding a path it can be interpreted as of finding a flow of one unit from source to sink. The first constraint guarrantees that one unit of flow is sent from the origin and the second constraint makes sure that this unit of flow is received at the sink.

2.6 The maximal flow problem

Assume that there is a positive number attached with each arc of a given network to indicate the maximum amount of flow which is allowed to be

passed through it. It is further assumed that the network is oriented from a single source toward a single sink. This problem is subject to the *conservation of flow* property. This property requires all nodes other than source or sink to pass all the flow leading into that node to be equal to the amount of flow leaving from it. The objective of the maximal flow problem is to find a schedule which sends the maximum possible amount of flow from source to the sink of a network and not to violate the upper bounds on the capacity of all arcs.

A useful concept which could help to find out when optimality is achieved is the definition of a *cut*. A cut is a set of arcs whose removal from a network will disconnect flow from source to the sink.

In order to describe the mathematical model of the maximal flow problem assume that there is no limit on the amount of flow supplied at the source. Let i=l, and i=n denote the corresponding indices of the source and sink, f_{ij} represents the amount of flow sent along the arc (i,j).The maximal flow problem formulation is as below:

$$Maximize \sum_{i=n} f_{in}$$

st (model 2.5)

$$\sum_j f_{ij} - \sum_j f_{ji} = 0, i \neq 1, and\ i \neq n$$

$$0 \leq f_{ij} \leq u_{ij} \quad \text{if (i,j) is an arc in the network}$$

The above summations are taken only over those arcs which exist in the network.

2.7 General pure minimum cost flow problem

The minimum cost flow problem plays a central role among all network problems ranging from the special cases like the assignment problem to the general

one such as the pure minimum cost flow problem with gains and losses.

To define the mathematical model of the problem, let the following parameters be associated with each arc (i,j):

1. A linear cost coefficient $c_{ij.}$

2. A real valued function f_{ij} which represents the amount of flow sent along the arc (i,j)

3. A lower bound l_{ij} on f_{ij}.

4. An upper bound u_{ij} on f_{ij}

Let a known flow s_i- be sent from each source node i and a known flow d_j be received by each sink node j. Besides the conservation of flow at each intermediary node it is assumed that $\sum_i s_i = \sum_j d_j$.

$$\text{Minimize} \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} c_{ij} f_{ij}$$

st (model 2.6)

$$\sum_j f_{ij} - \sum_k f_{ki} = \begin{cases} s_i & \text{if i is a source} \\ -d_i & \text{if i is a sink} \\ 0 & \text{otherwise} \end{cases} \qquad (i = 1,2,\ldots n)$$

$$l_{ij} \le f_{ij} \le u_{ij} \forall i, j$$

2.8 Generalized network problem (Network with gains and losses)

In all the previously described network problems conservation of flow is assumed, i. e. the amount of flow entering an arc is equal to the amount leaving it. In a number of applications it is required to modify the flow values during its passage along an arc. These applications include water reser-

voir models (evaporation,rainfall),investment models(gains and losses of capital),production and distribution models(quality control rejects, order cancellation) crew scheduling problems(absenteeism) and many more. A network that provides for gains and losses of flow across its arcs is called a generalized network. The flow modification is made by gain/loss factors. Note that the conservation of flow in intermediate nodes are still maintained.

The generalized network problem can be viewed as a minimum cost flow problem with gains and losses. The objective is

(1) to supply F units of flow to the sink ,t, at a minimum cost or

(2) to dispatch a flow of $\overline{F}$ units from the source,s,to the sink,t,at a minimum cost.

In order to formulate the mathematical model of the problem as a linear programming problem the following conventions are necessary:

Let u_{ij}, l_{ij}, and have the same definitions as in the minimum cost flow problem. Let α_{ij}, be the arc multiplier or gain/loss factor for arc (i,j). Note that if $\alpha_{ij} > 1$, the flow is increased (gains),if $\alpha_{ij} < 1$ the flow is reduced(losses),and if $\alpha_{ij} = 1$,a pure minimum cost flow problem is obtained. It is apparent that the flow values for this problem are not necessarily integer and can be any positive real number. The mathematical model of the network problem with gains and losses using the first objective function is summarized as below:

$$\text{Minimize} \sum_i \sum_j c_{ij} f_{ij}$$

st (model 2.7)

$$\sum_j f_{sj} - \sum_j \alpha_{ij} f_{js} \le \overline{F}$$

$$\sum_j f_{ij} - \sum_j \alpha_{ij} f_{ji} = 0 \quad i \ne s,t$$

$$\sum_j f_{tj} - \sum_j \alpha_{jt} f_{jt} = -F$$

$l_{ij} \le f_{ij} \le u_{ij}$ for all (i,j)

The above formulation assumes that $\alpha_{ij} \ge 0 \forall i, j$ and that all sources are combined into one source s and all sinks are combined into one sink t. It is also evident that $\overline{F} = F$ may not be true as in pure minimum cost flow problem. The input flows are not known until the final solution is found but it can be assumed to be less than or equal to $\overline{F}$ in which case it is possible that no feasible solution to the problem exist. However if $\overline{F}$ is sufficiently large, no feasible solution will erase.

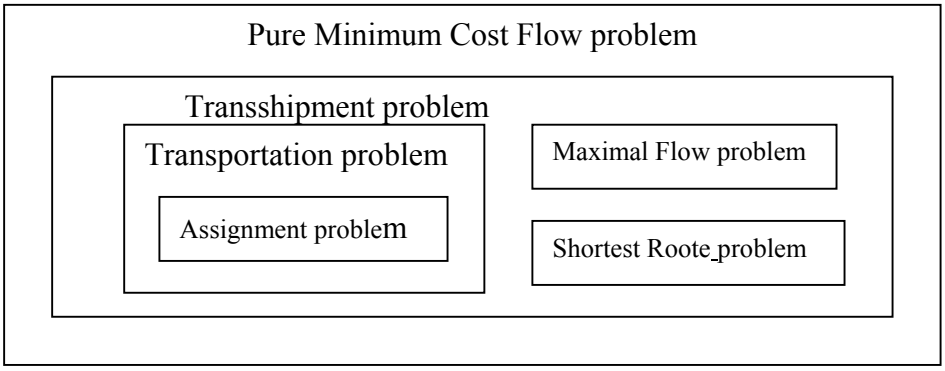
3 Interrelationship among network problems

The pure minimum cost flow problem ,although not the most general problem, can be interpreted as a general meeting point for a number of network problems. By suitable modifications in the minimum cost flow problem it can be converted into other network problems. For example if the flow capacity restrictions of model 2.6 are removed, the transshipment model (2.2) is obtained. The transportation model 2.1 can be obtained by restricting all nodes in the transshipment model 2.3 to be either source or destination. The assignment model 2.3 is obviously a special case of the model 2.1 in which all supplies and demands are equal to one.

Let the number of sources and destinations in model 2.6 be equal to 1 and $c_{ij} = -1 \forall i, j$ and excluding the source and sink the conservation of flow holds at intermediary points. Let the capacity restrictions of the arcs be maintained, then the modified model is a maximal flow problem. Now let the number of sources and sinks in model 2.6 be 1 and $s_1 = d_1 = 1$ then the shortest route model 2.4 is obtained.

We can find some more interrelationships among models other than the pure minimum cost model 2.6.By letting the number of sources and sinks and the total flow equal to one in the transshipment model 2.2 the shortest route model 2.4 will be obtained. The maximal flow model 2.5 is proved to be a special case of the transshipment problem. By incorporating the gain/loss factor in model 2.6 and relaxing the condition that the amount of flow supplied at all sources should be equal to the total flow demanded by all sinks, the generalized network problem model 2.7 is obtained.

Figure 3.1 might be helpful to understand the above mentioned interrelationships among network problems.



Fiqure 3.1

In the previous section a linear programming model was stated for each network problem. These LP models have special structure that can be exploited in developing highly efficient algorithms which deal with these models.

When formulating a network problem as a linear programming model the following correspondence holds between the elements of the network and it's linear programming formulation.

l. Each variable corresponds to an arc.

2. Each constraint corresponds to a node.
Figure 3.2 shows an example of a minimum cost flow problem with five nodes and seven arcs (for simplicity the arc capacities are ignored).

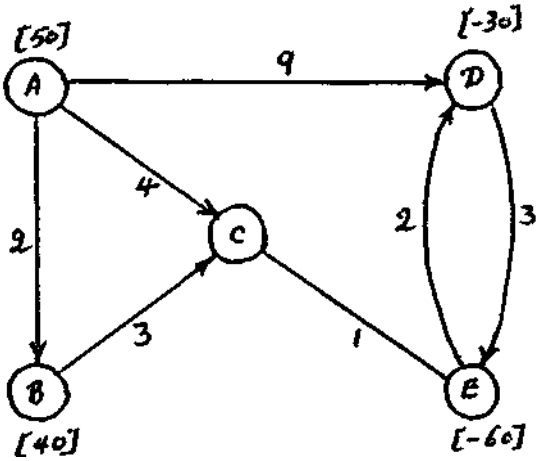


Figure 3.2

The LP formulation of the above problem has five constraints and seven variables

Table 3.1 is the initial tableau of the example.

	x_{AB}	x_{AC}	x_{AD}	x_{BC}	x_{CE}	x_{DE}	x_{ED}	*RHS*
Node A	1	1	1					50
Node B	-1			1				40
Node C		-1		-1	1			0
Node D			-1			1	-1	-30
Node E					-1	-1	1	-60
O.B.F	2	4	9	3	1	3	2	

The special structure of the network problems is even reflected in the tableau for this small example. The decision variables have the coefficient of 0,l, or -1 in all rows. Further each variable appears in exactly two equations one with coefficient 1 for a node from which the arc emanates and -1 for a node that the arc is leading into it.

The corresponding LP formulation of the problem is as below:

$$\text{Maximize} = 2x_{AB} + 4x_{AC} + 9x_{AD} + 3x_{BC} + x_{CE} + 3x_{DE} + 2x_{ED}$$

s.t.

$$x_{AB} + x_{AC} + x_{AD} = 50$$

$$-x_{AB} + x_{BC} = 40$$

$$-x_{AC} - x_{BC} + x_{CE} = 0$$

$$-x_{AD} + x_{DE} - x_{ED} = -30$$

$$-x_{CE} - x_{DE} + x_{ED} = -60$$

3.1 Unimodularity

If B is a basis for a LP with constraints A X=b, X $\geq$ 0 where A is partitioned into (B,N) then a basic solution is $X_B = B^{-1}$ .Provided that all coefficients of the LP model are integer then a sufficient condition for this solution to be integer is that B^{-1} is an integer matrix. The above discussion leads to the concept of unimodularity.A square matrix B is called unimodular if | det B |=1. An integer m by n matrix A is totally unimodular if every square nonsingular submatrix of A is unimodular. The following theorems concern the relationships between ILP and unimodularity.

Theorem 1.If A is totally unimodular then every basic solution of the following LP is integer:

Max cx

st Ax $\leq$ b where b is integer

x $\geq$ 0

Total unimodularity is not a necessary condition for the solution to be integer.

Theorem 2.Let A be an integer matrix. The following three statements are equivalent:

1. A is totally unimodular.

2.The extreme points of S(b)={x | Ax $\leq$ *b,x* $\geq$ 0 } are integer for arbitrary integer b.

3.Every square nonsingular submatrix of A has an integer inverse.

Theorem 3.An integer matrix A with a_{ij} = 0 1,-1 for all i and j, is totally unimodular if

l. No more than two nonzero elements appear in each column.

2. The rows can be partitioned into two subsets Q_1, Q_2 s.t.

(a) If a column contains two nonzero elements with the same sign, one element is in each of the subsets.

(b) If a column contains two nonzero elements of opposite sign, both elements are in the same subset.

The general minimum cost flow problem has an A matrix which is totally unimodular and if capacities are integer then an integer solution the LP formulation of the network problem is guaranteed.

4 Special purpose algorithms for network problems

4.1 The shortest route algorithm

A wellknown special purpose algorithm which deals with this problem is due to *Dijkstra*[4].The special structure of the shortest route problem is exploited and it is efficient in solving large scale problems.

To provide for describing the algorithm let us assume that the distance

between nodes i and j is c_{ij}. Each node is assigned a label which either *temporary or permanent*. Initially a temporary label indicates the direct distance between the source and any other node including the sink. Those nodes which are not directly connected to the source are assigned a temporary label of ∞.Those nodes which are determined to be in a shortest path are considered to be permanent. The source node is always labeled permanent. This algorithm operates on a simple logic that if a shortest path from source to another node is found then all other nodes located on it are labeled permanent. This means that the shortest path for all of them are found. Dijkstra's algorithm is an iterative one and it is described as below:

The shortest path from the source to node j is initially estimated as N_j and the algorithm tries to improve upon this solution. When it is found that no more improvement is possible, node j is labeled permanent and it can be represented by a symbol such as $\boxed{N_j}$ The source node receives the permanent label of $\boxed{0}$ while all other nodes receive the temporary label of $\boxed{\infty}$ The next permanent node is determined by finding the minimum of all temporary labels. After these initial preparations the algorithm iterates between the following steps and it stops when the sink is labeled permanent:

l. The sum of the last permanent label and it's direct distance to any temporary node is found and compared with the temporary label of that node. The minimum of the two quantities is defined as the new temporary label of that node.

2. The new permanent label is determined by finding the minimum of all temporary labels. If this new permanent label applies to the sink then the algorithm will terminate, otherwise it will return to step l.

4.2 The maximum flow algorithm

The maximal flow algorithm due to Ford and Fulkerson [5] is one which repeatedly finds paths of positive flow capacity between source and sink. The flow allocated to each path is accumulated and the maximal flow is obtained if no further paths of positive flow capacity can be found. To implement the above logic two concepts of labeling and flow augmenting are used.

The purpose of labeling is to specify both the origin of a shipment and whether this shipment causes the current flow to be increased or decreased. A label is an ordered pair of a node and a value. The value is the amount by which the current flow is to be changed and the node is the predecessor node of the arc along which the flow is sent to the labeled node. Let q_j be the amount of flow sent to node j from node i, then node j is labeled $[+q_j,i]$ which means an increase in the flow value. If the flow is to be decreased by q_j then node j is labeled $[-q_j, i]$ In both cases we say that *node j is labeled from i.*

Figure 4.1 further explains this procedure by two examples (a) and (b).In example (a) node j is labeled [+3,i] which means the current flow from i to j is increased by 3 and in example (b) node j receives the label [-3,i] which reflects a reduction of 3 in the flow value.

When node j is labeled from node i, the current amount of flow and the remaining unused capacity which is called the *residual capacity* of arc (i,j) should be updated. This algorithm only considers arcs with strictly positive

residual capacity.

Figure 4.1

A flow-augmenting path is one that connects source and sink such that a positive amount of flow could be sent along it. The purpose of flow-augmentation is to find a new path which can increase the total amount of flow leading into the sink. If a path is found then the residual capacity of all arcs in the path should be modified. The mathematical details of flow-augmentation is as described below.

Let (i,j) be a directed arc from i to j and j is labeled from i with $[+q_j, i]$then it is possible to increase the flow of the arc by q_j. Assume that the current flow in arc (i,j) is f_{ij} and the upper bound of the flow is u_{ij} then q_j is less than or equal to the residual capacity of $u_{ij} - f_{ij}$ Increasing the flow by this amount is not always possible, because the amount of flow supplied at node i may be less than the residual capacity, therefore q_j is determined by the equation $q_j = min[q_i, u_{ij} - f_{ij},]$. If the flow in the arc (i,j) is to be reduced the reduction is determined by $q_j = min[q_i, f_{ij}]$.

The algorithm proceeds as follows. The source node is initially labeled [∞, −] to show that the amount of flow supplied at the source is not limited, and all other nodes are unlabeled. We now move towards the sink while labeling nodes and seek a flow augmenting path through labeled nodes. To avoid being trapped in a local maximum the algorithm has the ability to

remove previously assigned flows to provide for a new flow augmenting path. From now on one of the following events will take place:

1. Once the flow augmenting path from source to sink is found, the sink node t is labeled $[q_t, k]$ where k is the predecessor node of the sink on the path which deliver q_t units of flow and each arc flow on the path will be increased or decreased by the amount q_t Then the residual capacities are updated as well and the current labels are erased and the procedure will be repeated.
2. If the sink node t cannot be labeled, that is, no more flow augmenting paths can be found, the current total flow is optimal and algorithm is terminated.

5 The minimum cost flow algorithm

An algorithm which deals with the pure general min cost flow problem is the *Out-of-Kilter* algorithm. Linear programming, dualty theory, and complementary slackness conditions are simulaneously used in the development of this algorithm. The arcs of the network are assumed to be directed and capacitated. A *circulation* is an assignment of flow to the arcs of the network such that the conservation of flow property is held at all nodes. In order to provide for circulations it is sometimes necessary to add an arc and modify the original network. For example a circulation might be made by adding an arc which establishes a direct connection between source and sink .This arc is called a return arc. The out-of-kilter algorithm is an iterative procedure which finds a circulation that minimizes the total cost of passing a feasible flow through the network.

Given a network (N, A) let assume that the per unit cost of shipping a flow along arc (i, j) is c_{ij} .The mathematical model of the problem can be

stated as the following LP problem:

$$\text{Minimize } \sum_{(i,j)\in A} c_{ij} f_{ij}$$

s.t.

$$f_{ij} \le u_{ij} \; \forall (i,j) \in A$$
$$f_{ij} \ge l_{ij} \; \forall (i,j) \in A$$

$$\sum_{j\in N} f_{ij} - \sum_{j\in N} f_{ji} = 0 \; \forall \, i \in N \; s.t. i \ne j$$

In order to write the dual of this LP problem, we will rewrite it in the following form:

$$\text{Maximize } \sum_{(i,j)\in A} -c_{ij} f_{ij}$$

s.t.

$$\sum_{j\in N} f_{ij} - \sum_{j\in N} f_{ji} = 0 \; \forall \, i \in N \, s.t. i \ne j$$

$$f_{ij} \le u_{ij} \; \forall (i,j) \in A$$

$$-f_{ij} \le -l_{ij} \; \forall (i,j) \in A$$

$$f_{ij} \ge 0$$

The corresponding dual problem is

$$\text{Minimize } \sum \left(u_{ij} \alpha_{ij} - l_{ij} \delta_{ij} \right)$$

s.t.

$$\pi_i - \pi_j + \alpha_{ij} - \delta_{ij} \ge -c_{ij} \; \forall (i,j) \in A$$

$$\pi_i \text{ unrestricted } \forall_i$$

$$\alpha_{ij} \ge 0 \; \forall (i,j) \in A$$

$$\delta_{ij} \ge 0 \; \forall (i,j) \in A$$

The π variables are associated with the conservation of flow property constraints, the a variables correspond to the upper bound constraints, and the δ variables correspond to the lower bound constraints. The optimality conditions can be derived from the dual problem. The necessary and sufficient conditions for the optimality are as follow:

Primal feasibility

P_1: $\sum_j f_{ij} - \sum_j f_{ji} = 0$ (conservation of flow) $\forall i \in N$

P_2: $l_{ij} \le f_{ij} \le u_{ij} =$ (capacity constraints) $\forall (i,j) \in A$

Dual feasibility

D_1: $\pi_i - \pi_j + \alpha_{ij} \ge -\delta_{ij} \ge -c_{ij} \forall (i,j) \in A$

D_2: $\alpha_{ij} \ge 0 \qquad \forall (i,j) \in A$

D_3: $\delta_{ij} \ge 0 \qquad \forall (i,j) \in A$

Complementary slackness

C_1: $if\ \pi_i - \pi_j + \alpha_{ij} - \delta_{ij} > -c_{ij} \qquad then\ f_{ij} = 0$

C_2: $if\ \alpha_{ij} > 0 \qquad then\ f_{ij} = u_{ij}$

C_3: $if\ \delta_{ij} > 0 \qquad then\ f_{ij} = l_{ij}$

If we assume the following conventions:

1. $\alpha_{ij} = max\{0; \pi_j - \pi_i - c_{ij}\}$

2. $\delta_{ij} = max\{0; \pi_j + \pi_i + c_{ij}\}$

An equivalent formulation of the optimality conditions can be stated as below:

3. If $\pi_i - \pi_j > c_{ij}\ then\ \alpha_{ij} > 0\ and\ f_{ij} = u_{ij}$
4. If $\pi_i - \pi_j > c_{ij}\ then\ \delta_{ij} > 0\ and\ f_{ij} = l_{ij}$
5. If $\pi_i - \pi_j > c_{ij}\ then\ u_{ij} \ge f_{ij} \ge l_{ij}$

and

6. $\sum_j f_{ij} - \sum_j f_{ji} = 0 \; \forall i$

Provided that the conditions 1 and 2 are satisfied and $\bar{c}_{ij} = c_{ij} + \pi_i - \pi_j$ the optimality conditions can be summarized as below:

k_1 : $If \bar{c}_{ij} < 0 \; then \; f_{ij} = u_{ij}$

k_2 : $if \; \bar{c}_{ij} > 0 \; then \; f_{ij} = l_{ij}$

k_3 : $if \; \bar{c}_{ij} = 0 \; then \; l_{ij} \le f_{ij} \le u_{ij}$

k_3 : Conservation of flow is satisfied

If an arc (i, j) satisfies conditions $k_1 k_2 k_3$ the arc is said to be in-kilter and if any one of these conditions does not hold the arc is said to be out-of kilter. An optimal solution is one for which all arcs are in-kilter and the conservation of flow constraints are satisfied.

The out-of-kilter algorithm tries to find the values of π_i and f_{ij} such that the above optimality conditions are met. The algorithm initializes f_{ij} with values such that the conservation of flow constraints are satisfied and it assigns arbitrary values to π_j .In terms of the optimality conditions ,nine mutually exclusive combinations can be recognized for each arc. Table 4.2 lists these combinations. The values of $\bar{c}_{ij}$ uniquely determine whether an arc is in-kilter or out-of-kilter.

POSSIBLE STATES OF AN ARC.

State	$\bar{c}_{ij}$	f_{ij}	*In kilter*
α	$\bar{c} > 0$	f=1	Yes
β	$\bar{c} = 0$	$l \leq f \leq u$	Yes
δ	$\bar{c} < 0$	$f = u$	Yes
α_1	$\bar{c} > 0$	$f < l$	No
β_1	$\bar{c} = 0$	$f < L$	No
δ_1	$\bar{c} < 0$	$f < u$	No
α_2	$\bar{c} > 0$	$f > l$	No
β_2	$\bar{c} = 0$	$f > u$	No
δ_2	$\bar{c} < 0$	$f > u$	No

This value also indicates whether an increase or a decrease in the arc flow will bring that arc in-kilter or not. This change in the amount of flow in arc (i,j) may violate the conservation of flow property at the connecting nodes. In order to restore this property another path from j to i is necessary. The arc (i,j) and the path from j to i form a *cycle.* Changing the flow along this cycle must be made in such a way that (i) no in-kilter arc is thrown out-of-kilter and (ii) no out-of-kilter arc is thrown further out-of-kilter. Assuming that a path from j to i is found there should be a functional notation which tells us how to change the flow of the arcs involved and which path to choose. This is called the *labeling procedure* which is discussed below:

5.1 The labeling procedure

l. If an arc is found in which the flow is to be increased to bring it in-kilter, the arc must be in either $\beta_1, \delta_1, \alpha_1,$. Label node j with $[q_j \ , \ i^+]$ which means that node j may receive q_j additional units from node i. If the arc is in state α_1 then define q_j to be $min\left[q_i, l_{ij} - f_{ij}\right]$ and if the arc is in state β_1 or δ_1 then

define q_j to be $min\ [q_i, u_{ij} - f_{ij}]$

2.1f the flow along arc (i,j) is to be decreased the arc must be in state $\beta_2, \delta_2,$ or $\alpha_2,$ label node i as $[q_i, j^{-1}]$ This indicates that the flow leaving node i and entering node j can be reduced by q_i. If the arc is in state $\alpha_2,$ or $\beta_2,$ define q_i *as min* $[q_j, f_{ij} - l_{ij}]$ and if the arc is in state $\delta_2,$ then define q_i *as min* [q_j ,f_{ij}-u_{ij}]

3. If an arc (i,j) is found to be in state $\alpha, \beta,$ or δ it is in kilter and it's flow should not be altered. The only exception is state β in which the flow might be increased or decreased without violating any in kilter condition.

The above modifications will bring the arc in kilter. In order to preserve the conservation of flow an alternative path from node i to j (or from j to i) must be found. To keep track of all changes made for the arcs along the alternative path, each intermediate node should also be labeled.

Consider an arbitrary arc (x,y) on an alternative path. This intermediate arc will fall in one of the nine mutually exclusive states. Now suppose that an observer is standing on an arbitrary labeled node x and wants to traverse the arc from x(labeled) to node y(unlabeled). From the labeled node ,the observer seeks to traverse all forward and reverse arcs incident to that node. If an arc is in the proper state, the node at the other end can be labeled. Several nodes might be labeled, although only one label is necessary to proceed. Once the labeling has been completed from a given node, that node is marked *scanned.* The question that must be answered as unscanned nodes on incident arcs are labeled is: "should we increase or decrease the flow along this particular node"?. At this point we may proceed from a scanned node to an unscanned but labeled node. The amount by which the flow in arc (x,y) can be changed is determined from the unique state of the arc, and the decision rule by which the change is made is given in the following table.

LABELING PROCESS THROUGH A FORWARD ARC

$(x,y) \in S$

π_x	c_{xy}	π_y
x	f_{xy}	y

$\bar{c}_{xy} = c_{xy} + \pi_x - \pi_y$

Label y, $[q_y, x^+]$ *if* $\bar{c}_{xy} > 0$ *and* $f_{xy} < l_{xy}$;

$$q_y = min[q_x, l_{xy} - f_{xy}]$$

$\bar{c}_{xy} \le 0$ *and* $f_{xy} < u_{xy}$;

$$q_y = min[q_x, u_{xy} - f_{xy}]$$

To preserve conservation of flow ,an alternative path from node j to node i(or from i to j) is sought ,and once found ,the flow along the path is adjusted according to final label of q_i(*or* q_j).Consider a wandering walker that starts a journey from node j through scanned nodes toward node i.

LABELING PROCESS THROUGH A REVERSE ARC

$(x,y) \in S$

π_x	c_{xy}	π_y
x	f_{yx}	y

$\bar{c}_{yx} = c_{yx} + \pi_y - \pi_x$

label y, $[q_y, x^{-1}]$ *if* $\bar{c}_{yx} \ge 0$ *and* $f_{yx} > l_{yx}$;

$$q_y = min[q_x, f_{yx} - l_{yx}]$$

$\bar{c}_{yx} < 0$ *and* $f_{yx} > u_{yx}$;

$$q_y = min[q_x, f_{yx} - u_{yx}]$$

Recall that no change is allowed in the flow across an arc if that change either(l)forces an in-kilter arc out of kilter or (2)drives an out-of-kilter arc further out-of-kilter. Therefore it is possible that the walker reaches a node at which he cannot continue. It seems that the walk is over and no path from j to i is available. Such an event is called *non-breakthrough.* Fortunately, if non-breakthrough occurs there is one more alternative available in the search for an optimal network flow. Recall that the state of an arc is uniquely deter-

mined by $\bar{c}_{ij} = c_{ij} + \pi_i - \pi_j$ so that a change in the π value affects the nine possible states of an arc. Each node has a π value associated with it, therefore there are exactly n dual variables for a network with n nodes. Assuming that non-breakthrough has occurred; which π variables should we change to provide a possible path from node j to node i(or from i to j if the flow is reduced)?There are two mutually exclusive sets of nodes:(l)scanned labeled nodes and (2)unlabeled nodes. The only nodes of interest with respect to non-breakthrough are nodes that will enables us to complete a walk between nodes j and i. Logically speaking ,to continue we must walk from a scanned labeled node to an unlabeled node. Therefore, the only π values that need to be considered are those associated with the arcs connecting labeled nodes to unlabeled nodes. *Define the set of all labeled nodes to be A, and the set of all unlabeled nodes to be* $\bar{A}$.

If we are standing on an arbitrary scanned node x and looking toward an unlabeled node y;we will either be looking at a forward arc or a reverse arc. If the arc is forward ,flow can pass from A to $\bar{A}$,and if the arc is reverse flow can pass from $\bar{A}$ into A.

CASE1: Let B be the set of all arcs originating at a node in A and terminating at a node in $\bar{A}$ with $\bar{c} > 0$ and flow less than or equal to the upper bound

CASE2: Let B be the set of all arcs originating at a node in $\bar{A}$ and terminating at a node in A with $\bar{c} > 0$ and flow greater than or equal to the lower bound.

Since $\bar{c}$ can be calculated for every arc in sets B and $\bar{B}$, we should proceed as follow:

1. Case 1: For any $\bar{c} > 0$ Define $\xi_1 = min\, B\left[\bar{c}_{xy}\right]$ *if* $B \neq \phi$; *otherwise* $\xi_1 = \infty$

2. Case 2: For any $\bar{c} < 0$ Define $\xi_2 = min\, \bar{B}\left[-\bar{c}_{xy}\right]$ *if* $\bar{B} \neq \phi$; *otherwise* $\xi_2 = \infty$

3. Let $\xi = min\left[\xi_1, \xi_2\right]$

4. Change all node numbers (π values) in the set $\bar{A}$ by adding ξ to π_k, where

k is a member of the set $\overline{A}$.

5. Do not erase any previous labels.

The process then continues by returning to the labeling procedure.

If the preceding steps change the state of at least one arc leading from the set of all labeled nodes to the set of all unlabeled nodes, the labeling procedure is continued until (l)breakthrough occurs (2) non-breakthrough occurs again. If the preceding steps do not change the state of at least one arc leading from A to $\overline{A}$, a feasible solution can not be found. When a breakthrough occurs, an alternative path from j to i (or from i to j) has been found, and the necessary action is to retrace this path and change the flow of all arcs along the path. At this point all labels are erased, another out-of-kilter arc is chosen and the procedure begins again. An optimal solution is found when all arcs are in-kilter.

In the event that breakthrough does not occur, sets B and $\overline{B}$,are reestablished and the node numbers are once again changed according to the previous rule. The labeling procedure is repeated until either arc (i j) is in-kilter or until a non-breakthrough occurs at which $\xi=\infty$ In the case where $\xi=\infty$, there is no optimal solution to the problem and the algorithm terminate. Note that if $\bar{c}=0$ for each arc ,all forward arc flows are at u_{ij} ,and all reverse arc flows are at l_{ij} then no path can be found.

The out-of-kilter algorithm can now be summaraized as follows: An initial circulation is chosen which satisfies the conservation of flow equations. A circulation of zero will always satisfy this condition. Next an arbitrary set of π values is assigned to the nodes. By using the labeling procedure, flow arcs are adjusted when breakthrough occurs, otherwise new node numbers are found and the procedure is repeated.

5.2 Algorithmic Steps

Step 1: Find an out-of-kilter arc (i,j).If none exists, stop.

Step 2: Determine if the flow in the arc should be increased or decreased to bring the arc in kilter. If it should be increased, go to step 3.1f it should be decreased, go to step 4.

Step 3: Using the labeling algorithm, find a path in the network from j to i along which the flow can be passed without causing any arcs on the path to become further out of kilter. If a path is found, adjust the flow in the path and increase the flow in (i,j).If (i,j) is in kilter, go to step l. If it is still out of kilter, repeat step 3. If no path can be found, go to step 5.

Step 4: Find a path from i to j along which the flow can be passed without causing any arc to become further out of kilter. If a path is found, adjust the flow in the path and decrease the flow in (i,j).If (i,j) is in kilter, go to step l. If (i,j) is still out of kilter, repeat step 4.1f no path is found, go to step 5.

Step 5: Change the π values and repeat step 2 for arc (i,j) keeping the same labels on all arcs already labeled. If the node numbers become ∞ ,stop since no feasible flow exists.

6 Summary

The purpose of this short report is to give the reader a general idea of networks and four wellknown problems in this area .Network optimization has found many applications in industry and management. The mathematic-cal formulation of these problems is of great help in discovering a number of interrelationships among them. Some network problems can be interpreted as special cases of more general ones. All network problems listed in this report can be formulated as linear programming problems. Highly efficient algorithms which take advantage of the special structure of network prob-

lems were presented. There are other algorithms which deal with the general pure minimum cost flow problem but the out-of-kilter algorithm is the most general and widely used one. A labeling procedure is utilized in all these algorithms to store information regarding the status of a node. Other network problems such as covering, and matching will be discussed in a forthcoming report.

References

[1] Bradley, S.P., Hax, A. C., and Magnanti, T.L.(1977) Applied Mathematical Programming: Addison-Wesley&Sons

[2] Bazarraa,M.S.;(1990) Linear Programming and Network Flows; John Wiley&Sons

[3] Cooper and Steinberg(1974) Methods and Applications of Linear Programming: W.B.Saunder Company.

[4] Dijkstra,E.W.;(1959),"Anote on two problems in connections with graphs", Numerishe Mathematik, 1,269-271.

[5] Ford, L., and Fulkerson (1962),Flows in Networks; Princeton,N.J.: Princeton University Press.

[6] Garfinkel,R. and Nemhauser,G.L;(1972) Integer Programming:John Wiley&Sons.

[7] Hillier,F.S. and Lieberman,G.J.(1990) Introduction to Operation Research (Fifth Edition),Mc Graw.Hill.

[8] Phillips,D.T. and Garcia-Diaz,A. (1981), Fundamentals of Network Analysis-Hall;Prentice Hall,Inc.