Network Models

8

There are several kinds of linear-programming models that exhibit a special structure that can be exploited in the construction of efficient algorithms for their solution. The motivation for taking advantage of their structure usually has been the need to solve larger problems than otherwise would be possible to solve with existing computer technology. Historically, the first of these special structures to be analyzed was the transportation problem, which is a particular type of network problem. The development of an efficient solution procedure for this problem resulted in the first widespread application of linear programming to problems of industrial logistics. More recently, the development of algorithms to efficiently solve particular large-scale systems has become a major concern in applied mathematical programming.

Network models are possibly still the most important of the special structures in linear programming. In this chapter, we examine the characteristics of network models, formulate some examples of these models, and give one approach to their solution. The approach presented here is simply derived from specializing the rules of the simplex method to take advantage of the structure of network models. The resulting algorithms are extremely efficient and permit the solution of network models so large that they would be impossible to solve by ordinary linear-programming procedures. Their efficiency stems from the fact that a pivot operation for the simplex method can be carried out by simple addition and subtraction without the need for maintaining and updating the usual tableau at each iteration. Further, an added benefit of these algorithms is that the optimal solutions generated turn out to be *integer* if the relevant constraint data are integer.

8.1 THE GENERAL NETWORK-FLOW PROBLEM

A common scenario of a network-flow problem arising in industrial logistics concerns the distribution of a single homogeneous product from plants (origins) to consumer markets (destinations). The total number of units produced at each plant and the total number of units required at each market are assumed to be known. The product need not be sent directly from source to destination, but may be routed through intermediary points reflecting warehouses or distribution centers. Further, there may be capacity restrictions that limit some of the shipping links. The objective is to minimize the variable cost of producing and shipping the products to meet the consumer demand.

The sources, destinations, and intermediate points are collectively called *nodes* of the network, and the transportation links connecting nodes are termed *arcs*. Although a production/distribution problem has been given as the motivating scenario, there are many other applications of the general model. Table E8.1 indicates a few of the many possible alternatives.

A numerical example of a network-flow problem is given in Fig 8.1. The nodes are represented by numbered circles and the arcs by arrows. The arcs are assumed to be *directed* so that, for instance, material can be sent from node 1 to node 2, but not from node 2 to node 1. Generic arcs will be denoted by i–j, so that 4–5 means the arc *from* node 4 *to* node 5. Note that some pairs of nodes, for example 1 and 5, are not connected directly by an arc.

Table 8.1 Examples of Network Flow Problems

	Urban transportation	*Communication systems*	*Water resources*
Product	Buses, autos, etc.	Messages	Water
Nodes	Bus stops, street intersections	Communication centers, relay stations	Lakes, reservoirs, pumping stations
Arcs	Streets (lanes)	Communication channels	Pipelines, canals, rivers

Figure 8.1 exhibits several additional characteristics of network flow problems. First, a flow capacity is assigned to each arc, and second, a per-unit cost is specified for shipping along each arc. These characteristics are shown next to each arc. Thus, the flow on arc 2–4 must be between 0 and 4 units, and each unit of flow on this arc costs \$2.00. The ∞'s in the figure have been used to denote unlimited flow capacity on arcs 2–3 and 4–5. Finally, the numbers in parentheses next to the nodes give the material supplied or demanded at that node. In this case, node 1 is an origin or *source node* supplying 20 units, and nodes 4 and 5 are destinations or *sink nodes* requiring 5 and 15 units, respectively, as indicated by the negative signs. The remaining nodes have no net supply or demand; they are intermediate points, often referred to as *transshipment nodes*.

The objective is to find the minimum-cost flow pattern to fulfill demands from the source nodes. Such problems usually are referred to as *minimum-cost flow* or *capacitated transshipment* problems. To transcribe the problem into a formal linear program, let

$$x_{ij} = \text{Number of units shipped from node } i \text{ to } j \text{ using arc } i\text{–}j.$$

Then the tabular form of the linear-programming formulation associated with the network of Fig. 8.1 is as shown in Table 8.2.

The first five equations are flow-balance equations at the nodes. They state the conservation-of-flow law,

$$\begin{pmatrix}\text{Flow out}\\ \text{of a node}\end{pmatrix} - \begin{pmatrix}\text{Flow into}\\ \text{a node}\end{pmatrix} = \begin{pmatrix}\text{Net supply}\\ \text{at a node}\end{pmatrix}.$$

As examples, at nodes 1 and 2 the balance equations are:

$$\begin{aligned} x_{12} + x_{13} \qquad\qquad\qquad &= 20 \\ x_{23} + x_{24} + x_{25} - x_{12} &= \ \ 0. \end{aligned}$$

It is important to recognize the special structure of these balance equations. Note that there is one balance equation for each node in the network. The flow variables x_{ij} have only 0, +1, and −1 coefficients in these

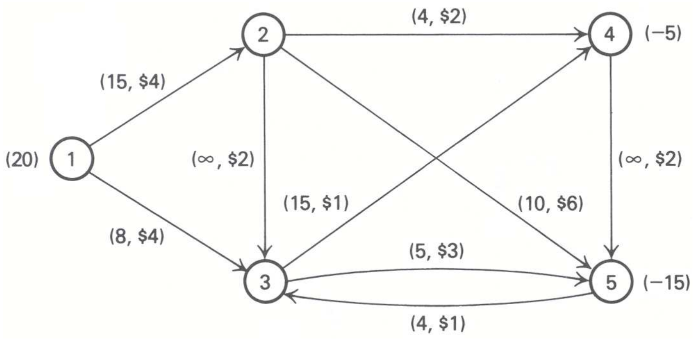


Figure 8.1 Minimum-cost flow problem.

Table 8.2 Tableau for Minimum-Cost Flow Problem

	x_{12}	x_{13}	x_{23}	x_{24}	x_{25}	x_{34}	x_{35}	x_{45}	x_{53}	*Righthand side*
Node 1	1	1								20
Node 2	−1		1	1	1					0
Node 3		−1	−1			1	1		−1	0
Node 4				−1		−1		1		−5
Node 5					−1		−1	−1	1	−15
Capacities	15	8	∞	4	10	15	5	∞	4	
Objective function	4	4	2	2	6	1	3	2	1	(Min)

equations. Further, each variable appears in exactly two balance equations, once with a $+1$ coefficient, corresponding to the node from which the arc emanates; and once with a -1 coefficient, corresponding to the node upon which the arc is incident. This type of tableau is referred to as a *node–arc incidence matrix*; it completely describes the physical layout of the network. It is this particular structure that we shall exploit in developing specialized, efficient algorithms.

The remaining two rows in the table give the upper bounds on the variables and the cost of sending one unit of flow across an arc. For example, x_{12} is constrained by $0 \leq x_{12} \leq 15$ and appears in the objective function as $2x_{12}$. In this example the lower bounds on the variables are taken implicitly to be zero, although in general there may also be nonzero lower bounds.

This example is an illustration of the following general *minimum-cost flow* problem with *n* nodes:

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

subject to:

$$\sum_j x_{ij} - \sum_k x_{ki} = b_i \quad (i = 1, 2, \ldots, n), \qquad \text{[Flow balance]}$$

$$l_{ij} \leq x_{ij} \leq u_{ij}. \qquad \text{[Flow capacities]}$$

The summations are taken only over the arcs in the network. That is, the first summation in the ith flow-balance equation is over all nodes j such that i–j is an arc of the network, and the second summation is over all nodes k such that k–i is an arc of the network. The objective function summation is over arcs i–j that are contained in the network and represents the total cost of sending flow over the network. The ith balance equation is interpreted as above: it states that the flow out of node i minus the flow into i must equal the net supply (demand if b_i is negative) at the node. u_{ij} is the upper bound on arc flow and may be $+\infty$ if the capacity on arc i–j is unlimited. ℓ_{ij} is the lower bound on arc flow and is often taken to be zero, as in the previous example. In the following sections we shall study variations of this general problem in some detail.

8.2 SPECIAL NETWORK MODELS

There are a number of interesting special cases of the minimum-cost flow model that have received a great deal of attention. This section introduces several of these models, since they have had a significant impact on the development of a general network theory. In particular, algorithms designed for these specific models have motivated solution procedures for the more general minimum-cost flow problem.

The Transportation Problem

The transportation problem is a network-flow model without intermediate locations. To formulate the problem, let us define the following terms:

a_i = Number of units available at source i $(i = 1, 2, \ldots, m)$;
b_j = Number of units required at destination j $(j = 1, 2, \ldots, n)$;
c_{ij} = Unit transportation cost from source i to destination j
$(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$.

For the moment, we assume that the total product availability is equal to the total product requirements; that is,

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j.$$

Later we will return to this point, indicating what to do when this supply–demand balance is not satisfied. If we define the decision variables as:

x_{ij} = Number of units to be distributed from source i to destination j
$(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$,

we may then formulate the transportation problem as follows:

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

subject to:

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad (i = 1, 2, \ldots, m), \tag{2}$$

$$\sum_{i=1}^{m} (-x_{ij}) = -b_j \qquad (j = 1, 2, \ldots, n), \tag{3}$$

$$x_{ij} \geq 0 \qquad (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n) \tag{4}$$

Expression (1) represents the minimization of the total distribution cost, assuming a linear cost structure for shipping. Equation (2) states that the amount being shipped from source i to all possible destinations should be equal to the total availability, a_i, at that source. Equation (3) indicates that the amounts being shipped to destination j from all possible sources should be equal to the requirements, b_j, at that destination. Usually Eq. (3) is written with positive coefficients and righthand sides by multiplying through by minus one.

Let us consider a simple example. A compressor company has plants in three locations: Cleveland, Chicago, and Boston. During the past week the total production of a special compressor unit out of each plant has been 35, 50, and 40 units respectively. The company wants to ship 45 units to a distribution center in Dallas, 20 to Atlanta, 30 to San Francisco, and 30 to Philadelphia. The unit production and distribution costs from each plant to each distribution center are given in Table E8.3. What is the best shipping strategy to follow?

The linear-programming formulation of the corresponding transportation problem is:

$$\begin{aligned}\text{Minimize } z = {} & 8x_{11} + 6x_{12} + 10x_{13} + 9x_{14} + 9x_{21} + 12x_{22} + 13x_{23} \\ & + 7x_{24} + 14x_{31} + 9x_{32} + 16x_{33} + 5x_{34},\end{aligned}$$

Table 8.3 Unit Production and Shipping Costs

	Distribution centers				
Plants	Dallas	Atlanta	San Francisco	Phila.	*Availability* (units)
Cleveland	8	6	10	9	35
Chicago	9	12	13	7	50
Boston	14	9	16	5	40
Requirements (units)	45	20	30	30	[125]

subject to:

$$\begin{array}{rrrrrrrrrrrrcr}
x_{11} & +x_{12} & +x_{13} & +x_{14} & & & & & & & & & = & 35, \\
& & & & x_{21} & +x_{22} & +x_{23} & +x_{24} & & & & & = & 50, \\
& & & & & & & & x_{31} & +x_{32} & +x_{33} & +x_{34} & = & 40, \\
-x_{11} & & & & -x_{21} & & & & -x_{31} & & & & = & -45, \\
& -x_{12} & & & & -x_{22} & & & & -x_{32} & & & = & -20, \\
& & -x_{13} & & & & -x_{23} & & & & -x_{33} & & = & -30, \\
& & & -x_{14} & & & & -x_{24} & & & & -x_{34} & = & -30,
\end{array}$$

$$x_{ij} \geq 0 \qquad \text{(i=1,2,3; j=1,2,3,4).}$$

Because there is no ambiguity in this case, the same numbers normally are used to designate the origins and destinations. For example, x_{11} denotes the flow from source 1 to destination 1, although these are two distinct nodes. The network corresponding to this problem is given in Fig. 8.2.

The Assignment Problem

Though the transportation model has been cast in terms of material flow from sources to destinations, the model has a number of additional applications. Suppose, for example, than n people are to be assigned to n jobs and that c_{ij} measures the performance of person i in job j. If we let

$$x_{ij} = \begin{cases} 1 & \text{if person } i \text{ is assigned to job } j, \\ 0 & \text{otherwise,} \end{cases}$$

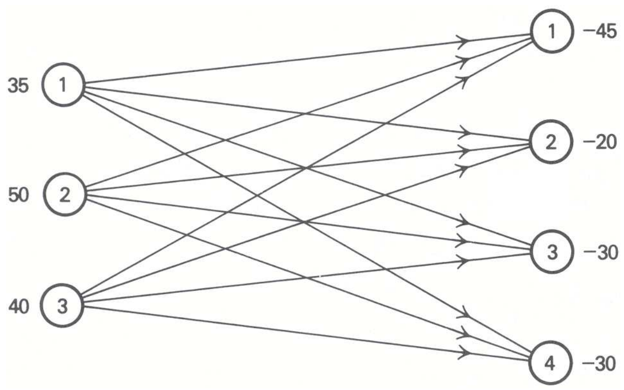


Figure 8.2 Transportation network.

we can find the optimal assignment by solving the optimization problem:

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

subject to:

$$\begin{aligned}
\sum_{j=1}^{n} x_{ij} &= 1 \qquad (i = 1, 2, \ldots, n),\\
\sum_{i=1}^{n} x_{ij} &= 1 \qquad (j = 1, 2, \ldots, n),\\
x_{ij} &= 0 \quad \text{or} \quad 1 \qquad (i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, n).
\end{aligned}$$

The first set of constraints shows that each person is to be assigned to exactly one job and the second set of constraints indicates that each job is to be performed by one person. If the second set of constraints were multiplied by minus one, the equations of the model would have the usual network interpretation.

As stated, this assignment problem is formally an integer program, since the decision variables x_{ij} are restricted to be zero or one. However, if these constraints are replaced by $x_{ij} \geq 0$, the model becomes a special case of the transportation problem, with one unit available at each source (person) and one unit required by each destination (job). As we shall see, network-flow problems have integer solutions, and therefore formal specification of integrality constraints is unnecessary. Consequently, application of the simplex method, or most network-flow algorithms, will solve such integer problems directly.

The Maximal Flow Problem

For the maximal flow problem, we wish to send as much material as possible from a specified node s in a network, called the *source*, to another specified node t, called the *sink*. No costs are associated with flow. If v denotes the amount of material sent from node s to node t and x_{ij} denotes the flow from node i to node j over arc i–j the formulation is:

$$\text{Maximize } v,$$

subject to:

$$\sum_{j} x_{ij} - \sum_{k} x_{ki} = \begin{cases} v & \text{if } i = s \text{ (source)},\\ -v & \text{if } i = t \text{ (sink)},\\ 0 & \text{otherwise}, \end{cases}$$

$$0 \leq x_{ij} \leq u_{ij} \qquad (i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, n).$$

As usual, the summations are taken only over the arcs in the network. Also, the upper bound u_{ij} for the flow on arc i–j is taken to be $+\infty$ if arc i–j has unlimited capacity. The interpretation is that v units are supplied at s and consumed at t.

Let us introduce a fictitious arc t–s with unlimited capacity; that is, $u_{ts} = +\infty$. Now x_{ts} represents the variable v, since x_{ts} simply returns the v units of flow from node t back to node s, and no formal external supply of material occurs. With the introduction of the arc t–s, the problem assumes the following special form of the general network problem:

$$\text{Maximize } x_{ts},$$

subject to:

$$\sum_{j} x_{ij} - \sum_{k} x_{ki} = 0 \qquad (i = 1, 2, \ldots, n),$$

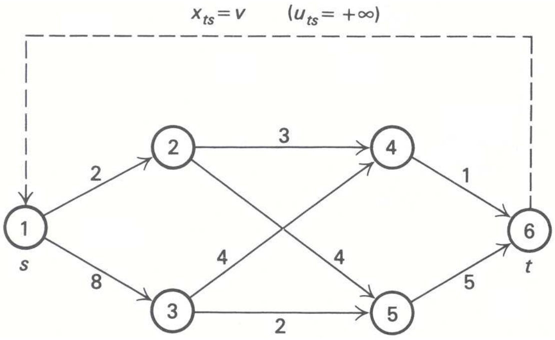


Figure 8.3

$$0 \leq x_{ij} \leq u_{ij} \qquad (i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, n).$$

Let us again consider a simple example. A city has constructed a piping system to route water from a lake to the city reservoir. The system is now underutilized and city planners are interested in its overall capacity. The situation is modeled as finding the maximum flow from node 1, the lake, to node 6, the reservoir, in the network shown in Fig. 8.3.

The numbers next to the arcs indicate the maximum flow capacity (in 100,000 gallons/day) in that section of the pipeline. For example, at most 300,000 gallons/day can be sent from node 2 to node 4. The city now sends 100,000 gallons/day along each of the paths 1–2–4–6 and 1–3–5–6. What is the maximum capacity of the network for shipping water from node 1 to node 6?

The Shortest-Path Problem

The shortest-path problem is a particular network model that has received a great deal of attention for both practical and theoretical reasons. The essence of the problem can be stated as follows: Given a network with distance c_{ij} (or travel time, or cost, etc.) associated with each arc, find a path through the network from a particular origin (source) to a particular destination (sink) that has the shortest total distance. The simplicity of the statement of the problem is somewhat misleading, because a number of important applications can be formulated as shortest- (or longest-) path problems where this formulation is not obvious at the outset. These include problems of equipment replacement, capital investment, project scheduling, and inventory planning. The theoretical interest in the problem is due to the fact that it has a special structure, in addition to being a network, that results in very efficient solution procedures. (In Chapter 11 on dynamic programming, we illustrate some of these other procedures.) Further, the shortest-path problem often occurs as a subproblem in more complex situations, such as the subproblems in applying decomposition to traffic-assignment problems or the group-theory problems that arise in integer programming.

In general, the formulation of the shortest-path problem is as follows:

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

subject to:

$$\sum_j x_{ij} - \sum_k x_{ki} = \begin{cases} 1 & \text{if } i = s \text{ (source)}, \\ 0 & \text{otherwise}, \\ -1 & \text{if } i = t \text{ (sink)} \end{cases}$$

$$x_{ij} \geq 0 \quad \text{for all arcs } i\text{–}j \text{ in the network.}$$

We can interpret the shortest-path problem as a network-flow problem very easily. We simply want to send one unit of flow from the source to the sink at minimum cost. At the source, there is a net supply of one unit; at the sink, there is a net demand of one unit; and at all other nodes there is no net inflow or outflow.

As an elementary illustration, consider the example given in Fig. 8.4, where we wish to find the shortest distance from node 1 to node 8. The numbers next to the arcs are the distance over, or cost of using, that arc. For the network specified in Fig. 8.4, the linear-programming tableau is given in Tableau 1.

Tableau 8.4 Node–Arc Incidence Tableau for a Shortest-PathProblem

	x_{12}	x_{13}	x_{24}	x_{25}	x_{32}	x_{34}	x_{37}	x_{45}	x_{46}	x_{47}	x_{52}	x_{56}	x_{58}	x_{65}	x_{67}	x_{68}	x_{76}	x_{78}	*Rela-tions*	RHS
Node 1	1	1																	=	1
Node 2	−1		1	1	−1						−1								=	0
Node 3		−1			1	1	1												=	0
Node 4			−1			−1		1	1	1									=	0
Node 5				−1				−1			1	1	1	−1					=	0
Node 6									−1			−1		1	1	1	−1		=	0
Node 7							−1			−1					−1		1	1	=	0
Node 8													−1			−1		−1	=	−1
Distance	5.1	3.4	0.5	2.0	1.0	1.5	5.0	2.0	3.0	4.2	1.0	3.0	6.0	1.5	0.5	2.2	2.0	2.4	=	z (min)

8.3 THE CRITICAL-PATH METHOD

The Critical-Path Method (CPM) is a project-management technique that is used widely in both government and industry to analyze, plan, and schedule the various tasks of complex projects. CPM is helpful in identifying which tasks are critical for the execution of the overall project, and in scheduling all the tasks in accordance with their prescribed *precedence relationships* so that the total project completion date is minimized, or a target date is met at minimum cost.

Typically, CPM can be applied successfully in large construction projects, like building an airport or a highway; in large maintenance projects, such as those encountered in nuclear plants or oil refineries; and in complex research-and-development efforts, such as the development, testing, and introduction of a new product. All these projects consist of a well specified collection of tasks that should be executed in a certain prescribed sequence. CPM provides a methodology to define the interrelationships among the tasks, and to determine the most effective way of scheduling their completion.

Although the mathematical formulation of the scheduling problem presents a network structure, this is not obvious from the outset. Let us explore this issue by discussing a simple example.

Suppose we consider the scheduling of tasks involved in building a house on a foundation that already exists. We would like to determine in what sequence the tasks should be performed in order to minimize

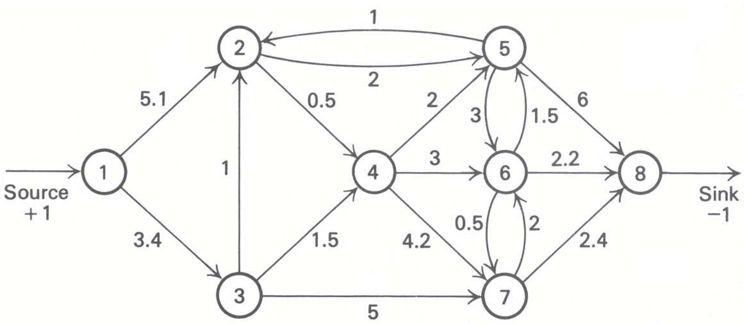


Figure 8.4 Network for a shortest-path problem.

the total time required to execute the project. All we really know is how long it takes to carry out each task and which tasks must be completed before commencing any particular task. In fact, it will be clear that we need only know the tasks that *immediately* precede a particular task, since completion of all *earlier* tasks will be implied by this information. The tasks that need to be performed in building this particular house, their immediate predecessors, and an estimate of their duration are give in Table E8.4.

It is clear that there is no need to indicate that the siding must be put up before the outside painting can begin, since putting up the siding precedes installing the windows, which precedes the outside painting. It is always convenient to identify a ‘‘start’’ task, that is, an immediate predecessor to all tasks, which in itself does not have predecessors; and a ‘‘finish’’ task, which has, as immediate predecessors, *all* tasks that in actuality have no successors.

Table 8.4 Tasks and Precedence Relationships

No.	*Task*	*Immediate predecessors*	*Duration*	*Earliest starting times*
0	Start	—	0	—
1	Framing	0	2	t_1
2	Roofing	1	1	t_2
3	Siding	1	1	t_2
4	Windows	3	2.5	t_3
5	Plumbing	3	1.5	t_3
6	Electricity	2, 4	2	t_4
7	Inside Finishing	5, 6	4	t_5
8	Outside Painting	2, 4	3	t_4
9	Finish	7, 8	0	t_6

Although it is by no means required in order to perform the necessary computations associated with the scheduling problem, often it is useful to represent the interrelations among the tasks of a given project by means of a network diagram. In this diagram, nodes represent the corresponding tasks of the project, and arcs represent the precedence relationships among tasks. The network diagram for our example is shown in Fig. 8.5.

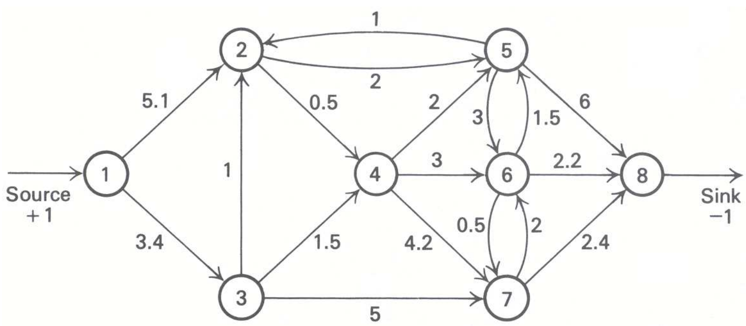


Figure 8.5 Task-oriented network.

As we can see, there are nine nodes in the network, each representing a given task. For this reason, this network representation is called a task- (or activity-) oriented network.

If we assume that our objective is to minimize the elapsed time of the project, we can formulate a linear-programming problem. First, we define the decision variables t_i for $i = 1, 2, \ldots, 6$, as the earliest starting times for each of the tasks. Table 8.4. gives the earliest starting times where the same earliest starting time is assigned to tasks with the same immediate predecessors. For instance, tasks 4 and 5 have task 3 as their immediate predecessor. Obviously, they cannot start until task 3 is finished; therefore, they should have the *same* earliest starting time. Letting t_6 be the earliest completion time of the entire project, out objective is to minimize the project duration given by

$$\text{Minimize } t_6 - t_1,$$

subject to the precedence constraints among tasks. Consider a particular task, say 6, installing the electricity. The earliest starting time of task 6 is t_4, and its immediate predecessors are tasks 2 and 4. The earliest starting times of tasks 2 and 4 are t_2 and t_3, respectively, while their durations are 1 and 2.5 weeks, respectively. Hence, the earliest starting time of task 6 must satisfy:

$$t_4 \geq t_2 + 1,$$
$$t_4 \geq t_3 + 2.5.$$

In general, if t_j is the earliest starting time of a task, t_i is the earliest starting time of an immediate predecessor, and d_{ij} is the duration of the immediate predecessor, then we have:

$$t_j \geq t_i + d_{ij}.$$

For our example, these precedence relationships define the linear program given in Tableau E8.2.

Tableau 8.2

t_1	t_2	t_3	t_4	t_5	t_6	*Relation*	RHS
−1	1					≥	2
	−1	1				≥	3
	−1		1			≥	1
		−1	1			≥	2.5
		−1		1		≥	1.5
			−1	1		≥	2
			−1		1	≥	3
				−1	1	≥	4
−1					1	=	T (min)

We do not yet have a network flow problem; the constraints of (5) do not satisfy our restriction that each column have only a plus-one and a minus-one coefficient in the constraints. However, this *is* true for the rows, so let us look at the dual of (5). Recognizing that the variables of (5) have not been explicitly restricted to the nonnegative, we will have equalityconstraints in the dual. If we let x_{ij} be the dual variable associated with the constraint of (5) that has a minus one as a coefficient for t_i and a plus one as a coefficient of t_j, the dual of (5) is then given in Tableau 3.

Tableau 8.3

x_{12}	x_{23}	x_{24}	x_{34}	x_{35}	x_{45}	x_{46}	x_{56}	*Relation*	RHS
−1								=	−1
1	−1	−1						=	0
	1		−1	−1				=	0
		1	1		−1	−1		=	0
				1	1		−1	=	0
						1	1	=	1
2	3	1	2.5	1.5	2	3	4	=	z (max)

Now we note that each column of (6) has only one plus-one coefficient and one minus-one coefficient, and hence the tableau describes a network. If we multiply each equation through by minus one, we will have the usual sign convention with respect to arcs emanating from or incident to a node. Further, since the righthand side has only a plus one and a minus one, we have flow equations for sending one unit of flow from node 1 to node 6. The network corresponding to these flow equations is given in Fig. 8.6; this network clearly maintains the precedence relationships from Table 8.4. Observe that we have a longest-path problem, since we wish to maximize z (in order to minimize the project completion date T). Note that, in this network, the arcs represent

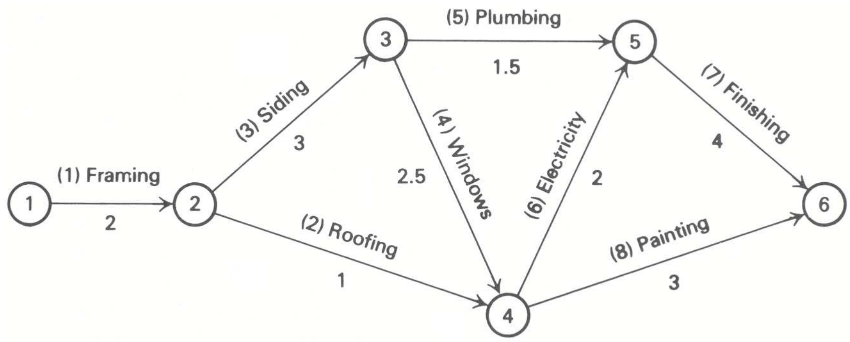


Figure 8.6 Event-oriented network.

the tasks, while the nodes describe the precedence relationships among tasks. This is the opposite of the network representation given in Fig. 8.5. As we can see, the network of Fig. 8.6. contains 6 nodes, which is the number of sequencing constraints prescribed in the task definition of Table 8.4. since only six earliest starting times were required to characterize these constraints. Because the network representation of Fig. 8.6 emphasizes the event associated with the starting of each task, it is commonly referred to as an event-oriented network.

There are several other issues associated with critical-path scheduling that also give rise to network-model formulations. In particular, we can consider allocating funds among the various tasks in order to reduce the total time required to complete the project. The analysis of the cost-*vs*.-time tradeoff for such a change is an important network problem. Broader issues of resource allocation and requirements smoothing can also be interpreted as network models, under appropriate conditions.

8.4 CAPACITATED PRODUCTION—A HIDDEN NETWORK

Network-flow models are more prevalent than one might expect, since many models not cast naturally as networks can be transformed into a network format. Let us illustrate this possibility by recalling the strategic-planning model for aluminum production developed in Chapter 6. In that model,bauxite ore is converted to aluminum products in several smelters, to be shipped to a number of customers. Production and shipment are governed by the following constraints:

$$\sum_{\text{a}} \sum_{\text{p}} Q_{\text{sap}} - M_{\text{s}} \qquad = 0 \qquad (\text{s} = 1, 2, \ldots, 11), \tag{7}$$

$$\sum_{\text{a}} Q_{\text{sap}} \qquad - E_{\text{sp}} = 0 \qquad (\text{s} = 1, 2, \ldots, 11;\ \text{p} = 1, 2, \ldots, 8), \tag{8}$$

$$\sum_{\text{s}} Q_{\text{sap}} \qquad = d_{\text{ap}} \qquad (\text{a} = 1, 2, \ldots, 40;\ \text{p} = 1, 2, \ldots, 8), \tag{9}$$

$$\underline{m}_{\text{s}} \leq M_{\text{s}} \leq \overline{m}_{\text{s}} \qquad (\text{s} = 1, 2, \ldots, 11),$$

$$\underline{e}_{\text{sp}} \leq E_{\text{sp}} \leq \overline{e}_{\text{sp}} \qquad (\text{p} = 1, 2, \ldots, 40).$$

Variable Q_{sap} is the amount of product p to be produced at smelter s and shipped to customer a. The constraints (7) and (8) merely define the amount M_{s} produced at smelter s and the amount E_{sp} of product p (ingots) to be ''cast'' at smelter s. Equations (9) state that the total production from all smelters must satisfy the demand d_{ap} for product p of each customer a. The upper bounds of M_{s} and E_{sp} reflect smelting and casting capacity, whereas the lower bounds indicate minimum economically attractive production levels.

As presented, the model is not in a network format, since it does not satisfy the property that every variable appear in exactly two constraints, once with a $+1$ coefficient and once with a -1 coefficient. It can be stated as a network, however, by making a number of changes. Suppose, first, that we rearrange all the constraints of (7), as

$$\sum_{\text{p}} \left(\sum_{\text{a}} Q_{\text{sap}} \right) - M_{\text{s}} = 0,$$

and substitute, for the term in parenthesis, E_{sp} defined by (8). Let us also multiply the constraints of (9) by (-1). The model is then rewritten as:

$$\sum_{\text{p}} E_{\text{sp}} - M_{\text{s}} = 0 \qquad (\text{s} = 1, 2, \ldots, 11), \tag{10}$$

$$\sum_{\text{a}} Q_{\text{sap}} - E_{\text{sp}} = 0 \qquad (\text{s} = 1, 2, \ldots, 11;\ \text{p} = 1, 2, \ldots, 8), \tag{11}$$

$$\sum_{\text{s}} -Q_{\text{sap}} = -d_{\text{ap}} \quad (\text{a} = 1, 2, \ldots, 40;\ \text{p} = 1, 2, \ldots, 8), \tag{12}$$

$$\underline{m}_{\text{s}} \leq M_{\text{s}} \leq \overline{m}_{\text{s}} \qquad (\text{s} = 1, 2, \ldots, 11),$$
$$\underline{e}_{\text{sp}} \leq E_{\text{sp}} \leq \bar{e}_{\text{sp}} \qquad (\text{p} = 1, 2, \ldots, 8).$$

Each variable E_{sp} appears once in the equations of (10) with a $+1$ coefficient and once in the equations of (11) with a -1 coefficient; each variable Q_{sap} appears once in the equations of (11) with a $+1$ coefficient and once in the equations of (12) with a -1 coefficient. Consequently, except for the variables M_{s}, the problem is in the form of a network. Now, suppose that we *add* all the equations to form one additional redundant constraint. As we have just noted, the terms involving the variables Q_{sap} and E_{sp} will all vanish, so that the resulting equation, when multiplied by minus one, is:

$$\sum_{\text{s}} M_{\text{s}} = \sum_{\text{a}} \sum_{\text{p}} d_{\text{ap}}. \tag{13}$$

Each variable M_{s} now appears once in the equations of (10) with a $+1$ coefficient and once in the equations of (13) with a -1 coefficient, so appending this constraint to the previous formulation gives the desired network formulation.

The network representation is shown Fig. 8.7. As usual, each equation in the model defines a node in the network. The topmost node corresponds to the redundant equation just added to the model; it just collects production from the smelters. The other nodes correspond to the smelters, the casting facilities for products at the smelters, and the customer–product demand combinations. The overall supply to the system, $\sum_{\text{a}} \sum_{\text{p}} d_{\text{ap}}$, as indicated at the topmost node, is the total production at the smelters, and must equal the demand for all products.

In practice, manipulations like these just performed for re-expressing problems can be used frequently to exhibit network structure that might be hidden in a model formulation. They may not always lead to pure network-flow problems, as in this example, but instead might show that the problem has a substantial network component. The network features might then be useful computationally in conjunction with large-scale systems techniques that exploit the network structure.

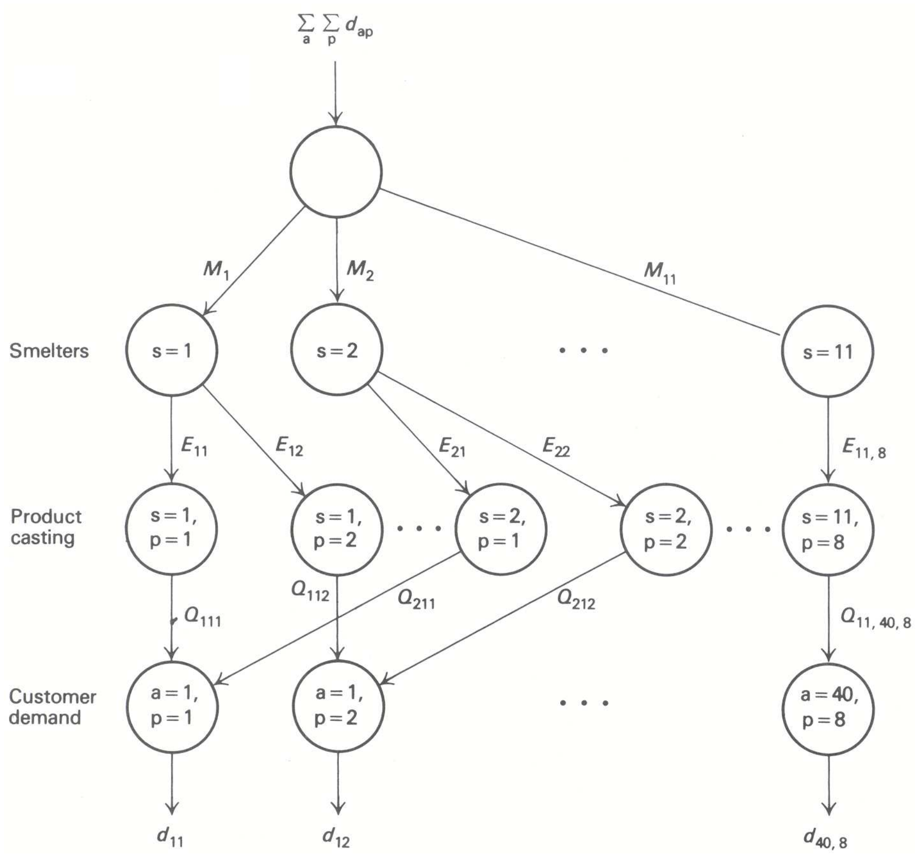


Figure 8.7 Network formulation of the aluminum production-planning model.

Finally, observe that the network in Fig. 8.7 contains only a small percentage of the arcs that could potentially connect the nodes since, for example, the smelters do not connect directly with customer demands. This low density of arcs is common in practice, and aids in both the information storage and the computations for network models.

8.5 SOLVING THE TRANSPORTATION PROBLEM

Ultimately in this chapter we want to develop an efficient algorithm for the general minimum-cost flow problem by specializing the rules of the simplex method to take advantage of the problem structure. However, before taking a somewhat formal approach to the general problem, we will indicate the basic ideas by developing a similar algorithm for the transportation problem. The properties of this algorithm for the transportation problem will then carry over to the more general minimum-cost flow problem in a straightforward manner. Historically, the transportation problem was one of the first special structures of linear programming for which an efficient special-purpose algorithm was developed. In fact, special-purpose algorithms have been developed for all of the network structures presented in Section 8.1, but they will not be developed here.

As we have indicated before, many computational algorithms are characterized by three stages:

1. obtaining an initial solution;
2. checking an optimality criterion that indicates whether or not a termination condition has been met (i.e., in the simplex algorithm, whether the problem is infeasible, the objective is unbounded over the feasible region, or an optimal solution has been found);
3. developing a procedure to improve the current solution if a termination condition has not been met.

After an initial solution is found, the algorithm repetitively applies steps 2 and 3 so that, in most cases, after

Tableau 8.4

Sources	Destinations 1	2	$\cdots$	n	Supply
1	c_{11} x_{11}	c_{12} x_{12}	$\cdots$	c_{1n} x_{1n}	a_1
2	c_{21} x_{21}	c_{22} x_{22}	$\cdots$	c_{2n} x_{2n}	a_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
m	c_{m1} x_{m1}	c_{m2} x_{m2}	$\cdots$	c_{mn} x_{mn}	a_m
Demand	b_1	b_2	$\cdots$	b_n	Total

a finite number of steps, a termination condition arises. The effectiveness of an algorithm depends upon its efficiency in attaining the termination condition.

Since the transportation problem is a linear program, each of the above steps can be performed by the simplex method. Initial solutions can be found very easily in this case, however, so phase I of the simplex method need not be performed. Also, when applying the simplex method in this setting, the last two steps become particularly simple.

The transportation problem is a special network problem, and the steps of any algorithm for its solution can be interpreted in terms of network concepts. However, it also is convenient to consider the transportation problem in purely algebraic terms. In this case, the equations are summarized very nicely by a tabular representation like that in Tableau E8.4.

Each row in the tableau corresponds to a source node and each column to a destination node. The numbers in the final column are the supplies available at the source nodes and those in the bottom row are the demands required at the destination nodes. The entries in cell i–j in the tableau denote the flow allocation x_{ij} from source i to destination j and the corresponding cost per unit of flow is c_{ij}. The sum of x_{ij} across row i must equal a_i in any feasible solution, and the sum of x_{ij} down column j must equal b_j.

Initial Solutions

In order to apply the simplex method to the transportation problem, we must first determine a basic feasible solution. Since there are $(m+n)$ equations in the constraint set of the transportation problem, one might conclude that, in a nondegenerate situation, a basic solution will have $(m+n)$ strictly positive variables. We should note, however, that, since

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij},$$

one of the equations in the constraint set is redundant. In fact, any one of these equations can be eliminated without changing the conditions represented by the original constraints. For instance, in the transportation example in Section 8.2, the last equation can be formed by summing the first three equations and subtracting the next three equations. Thus, the constraint set is composed of $(m+n-1)$ independent equations, and a corresponding nondegenerate basic solution will have exactly $(m+n-1)$ basic variables.

There are several procedures used to generate an initial basic feasible solution, but we will consider only a few of these. The simplest procedure imaginable would be one that ignores the costs altogether and rapidly produces a basic feasible solution. In Fig. 8.8, we have illustrated such a procedure for the transportation problem introduced in Section 8.2. We simply send as much as possible from origin 1 to destination 1, i.e., the minimum of the supply and demand, which is 35. Since the supply at origin 1 is then exhausted but the demand at destination 1 is not, we next fulfill the remaining demand at destination 1 from that available at origin 2. At this point destination 1 is completely supplied, so we send as much as possible (20 units) of the

Table 8.5 Finding an Initial Basis by the Northwest Corner Method

	Distribution centers				
Plants	1. Dallas	2. Atlanta	3. San Fran.	4. Phila.	*Supply*
1. Cleveland	35				~~35~~
2. Chicago	10	20	20		~~50~~ ~~40~~ ~~20~~
3. Boston			10	30	~~40~~ ~~30~~
Demand	~~45~~ ~~10~~	~~20~~	~~30~~ ~~10~~	~~30~~	

remaining supply of 40 at origin 2 to destination 2, exhausting the demand there. Origin 2 still has a supply of 20 and we send as much of this as possible to destination 3, exhausting the supply at origin 2 but leaving a demand of 10 at destination 3. This demand is supplied from origin 3, leaving a supply there of 30, exactly corresponding to the demand of 30 needed at destination 4. The available supply equals the required demand for this final allocation because we have assumed that the total supply equals the total demand, that is,

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j.$$

This procedure is called the *northwest-corner* rule since, when interpreted in terms of the transportation array, it starts with the upper leftmost corner (the northwest corner) and assigns the maximum possible flow allocation to that cell. Then it moves to the right, if there is any remaining supply in the first row, or to the next lower cell, if there is any remaining demand in the first column, and assigns the maximum possible flow allocation to that cell. The procedure repeats itself until one reaches the lowest right corner, at which point we have exhausted all the supply and satisfied all the demand.

Table E8.5 summarizes the steps of the northwest-corner rule, in terms of the transportation tableau, when applied to the transportation example introduced in Section 8.2.

The availabilities and requirements at the margin of the table are updated after each allocation assignment. Although the northwest-corner rule is easy to implement, since it does not take into consideration the cost of using a particular arc, it will not, in general, lead to a cost-effective initial solution.

An alternative procedure, which is cognizant of the costs and straightforward to implement, is the *minimum matrix* method. Using this method, we allocate as much as possible to the available arc with the lowest cost. Figure 8.9 illustrates the procedure for the example that we have been considering. The least-cost arc joins origin 3 and destination 4, at a cost of \$5/unit, so we allocate the maximum possible, 30 units, to this arc,

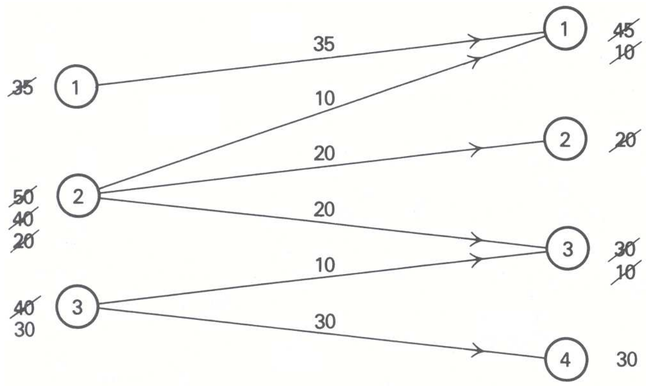


Figure 8.8 Finding an initial basis by the northwest-corner method.

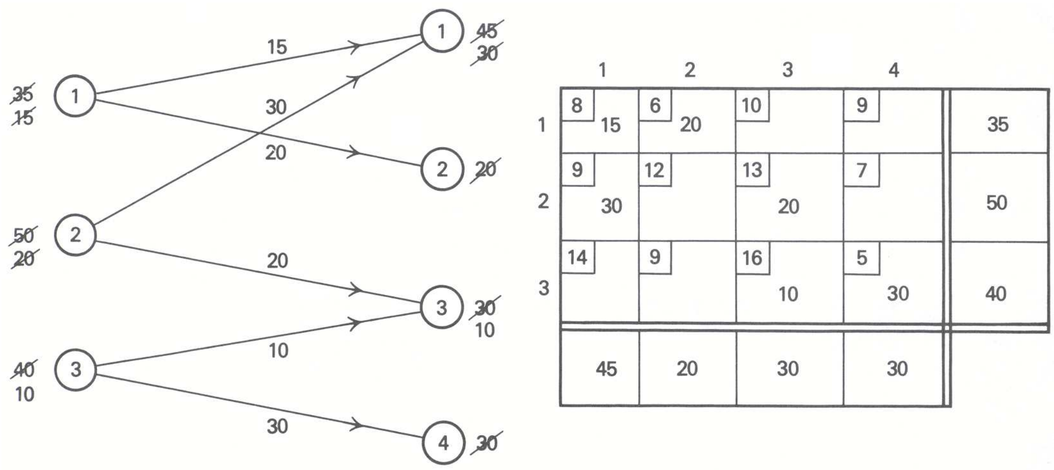


Figure 8.9 Finding an initial basis by the minimum-matrix method.

completely supplying destination 4. Ignoring the arcs entering destination 4, the least-cost remaining arc joins origin 1 and destination 2, at a cost of \$6/unit, so we allocate the maximum possible, 20 units, to this arc, completely supplying destination 2. Then, ignoring the arcs entering either destination 2 or 4, the least-cost remaining arc joins origin 1 and destination 1, at a cost of \$8/unit, so we allocate the maximum possible, 15 units, to this arc, exhausting the supply at origin 1. Note that this arc is the least-cost *remaining* arc but not the next-lowest-cost unused arc in the entire cost array. Then, ignoring the arcs leaving origin 1 or entering destinations 2 or 4, the procedure continues. It should be pointed out that the last two arcs used by this procedure are relatively expensive, costing \$13/unit and \$16/unit. It is often the case that the minimum matrix method produces a basis that simultaneously employs some of the least expensive arcs and some of the most expensive arcs.

There are many other methods that have been used to find an initial basis for starting the transportation method. An obvious variation of the minimum matrix method is the *minimum row* method, which allocates as much as possible to the available arc with least cost *in each row* until the supply at each successive origin is exhausted. There is clearly the analogous *minimum column* method.

It should be pointed out that any comparison among procedures for finding an initial basis should be made only by comparing *solution times*, including both finding the initial basis *and* determining an optimal solution. For example, the northwest-corner method clearly requires fewer operations to determine an initial basis than does the minimum matrix rule, but the latter generally requires fewer iterations of the simplex method to reach optimality.

The two procedures for finding an initial basic feasible solution resulted in different bases; however, both have a number of similarities. Each basis consists of exactly $(m + n - 1)$ arcs, one less than the number of nodes in the network. Further, each basis is a subnetwork that satisfies the following two properties:

1. Every node in the network is connected to every other node by a sequence of arcs from the subnetwork, where the direction of the arcs is ignored.
2. The subnetwork contains no loops, where a loop is a sequence of arcs connecting a node to itself, where again the direction of the arcs is ignored.

A subnetwork that satisfies these two properties is called a *spanning tree*.

It is the fact that a basis corresponds to a spanning tree that makes the solution of these problems by the simplex method very efficient. Suppose you were told that a feasible basis consists of arcs 1–1, 2–1, 2–2, 2–4, 3–2, and 3–3. Then the allocations to each arc can be determined in a straightforward way without algebraic manipulations of tableaus. Start by selecting any *end* (a node with only 1 arc connecting it to the rest of the network) in the subnetwork. The allocation to the arc joining that end must be the supply or demand available at that end. For example, source 1 is an end node. The allocation on arc 1–1 must then be 35, decreasing

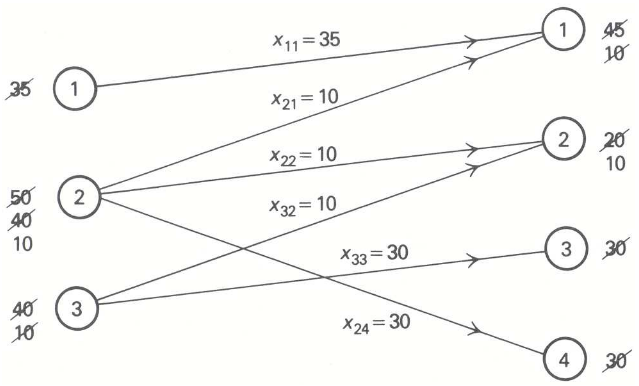


Figure 8.10 Determining the values of the basic variables.

the unfulfilled demand at destination 1 from 45 to 10 units. The end node and its connecting arc are then dropped from the subnetwork and the procedure is repeated. In Fig. 8.10, we use these end-node evaluations to determine the values of the basic variables for our example.

This example also illustrates that the solution of a transportation problem by the simplex method results in *integer* values for the variables. Since any basis corresponds to a spanning tree, as long as the supplies and demands are integers the amount sent across any arc must be an integer. This is true because, at each stage of evaluating the variables of a basis, as long as the remaining supplies and demands are integer the amount sent to any end must also be an integer. Therefore, if the initial supplies and demands are integers, any basic feasible solution will be an integer solution.

Optimization Criterion

Because it is so common to find the transportation problem written with $+1$ coefficients for the variables x_{ij} (i.e., with the demand equations of Section 8.2 multiplied by minus one), we will give the optimality conditions for this form. They can be easily stated in terms of the reduced costs of the simplex method. If x_{ij} for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$ is a feasible solution to the transportation problem:

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

subject to: *Shadow prices*

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad (i = 1, 2, \ldots, m), \qquad u_i$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad (j = 1, 2, \ldots, n), \qquad v_j$$

$$x_{ij} \geq 0,$$

then it is optimal if there exist shadow prices (or simplex multipliers) u_i associated with the origins and v_j associated with the destinations, satisfying:

$$\bar{c}_{ij} = c_{ij} - u_i - v_j \geq 0 \qquad \text{if } x_{ij} = 0, \tag{14}$$

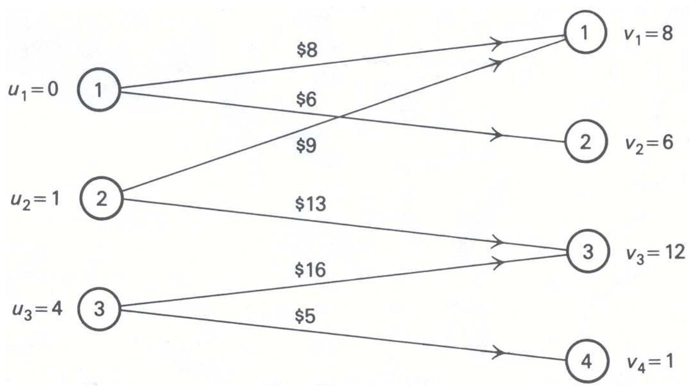


Figure 8.11 Determining the simplex multipliers.

and

$$\bar{c}_{ij} = c_{ij} - u_i - v_j = 0 \qquad \text{if } x_{ij} > 0. \tag{15}$$

The simplex method selects multipliers so that condition (15) holds for all basic variables, even if some basic variable $x_{ij} = 0$ due to degeneracy.

These conditions not only allow us to test whether the optimalsolution has been found or not, but provide us with the necessary foundation to reinterpret the characteristics of the simplex algorithm in order to solve the transportation problem efficiently. The algorithm will proceed as follows: after a basic feasible solution has been found (possibly by applying the northwest-corner method or the minimum matrix method), we choose simplex multipliers u_i and $v_j (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$ that satisfy:

$$u_i + v_j = c_{ij} \tag{16}$$

for basic variables. With these values for the simplex multipliers, we compute the corresponding values of the reduced costs:

$$\bar{c}_{ij} = c_{ij} - u_i - v_j \tag{17}$$

for all nonbasic variables. If every $\bar{c}_{ij}$ is nonnegative, then the optimal solution has been found; otherwise, we attempt to improve the current solution by increasing as much as possible the variable that corresponds to the most negative (since this is a minimization problem) reduced cost.

First, let us indicate the mechanics of determining from (16) the simplex multipliers associated with a particular basis. Conditions (16) consist of $(m + n - 1)$ equations in $(m + n)$ unknowns. However, any one of the simplex multipliers can be given an arbitrary value since, as we have seen, any one of the $(m + n)$ equations of the transportation problem can be considered redundant. Since there are $(m + n - 1)$ equations in (16), once one of the simplex multipliers has been specified, the remaining values of u_i and v_j are determined uniquely. For example, Fig. 8.11 gives the initial basic feasible solution produced by the minimum matrix method. The simplex multipliers associated with this basis are easily determined by first arbitrarily setting $u_1 = 0$. Given $u_1 = 0$, $v_1 = 8$ and $v_2 = 6$ are immediate from (16); then $u_2 = 1$ is immediate from v_1, and so on. It should be emphasized that the set of multipliers is not unique, since any multiplier could have been chosen and set to any finite value, positive or negative, to initiate the determination.

Once we have determined the simplex multipliers, we can then easily find the reduced costs for all nonbasic variables by applying (17). These reduced costs are given in Tableau 5. The —'s indicate the basic variable, which therefore has areduced cost of zero. This symbol is used to distinguish between basic variables and nonbasic variables at zero level.

Since, in Tableau 5, $\bar{c}_{13} = -2$ and $\bar{c}_{32} = -1$, the basis constructed by the minimum matrixmethod does not yield an optimal solution. We, therefore, need to find an improved solution.

Tableau 5 Reduced Costs

8 / —	6 / —	10 / -2	9 / 8	0
9 / —	12 / 5	13 / —	7 / 5	1
14 / 2	9 / -1	16 / —	5 / —	4
8	6	12	1	v_j \ u_i

Improving the Basic Solution

As we indicated, every basic variable has associated with it a value of $\overline{c}_{ij} = 0$. If the current basic solution is not optimal, then there exists at least one nonbasic variable x_{ij} at value zero with $\overline{c}_{ij}$ negative. Let us select, among all those variables, the one with the most negative $\overline{c}_{ij}$ (ties are broken by choosing arbitrarily from those variables that tie); that is,

$$\overline{c}_{st} = \operatorname*{Min}_{ij} \left\{ \overline{c}_{ij} = c_{ij} - u_i - v_j | \overline{c}_{ij} < 0 \right\}.$$

Thus, we would like to increase the corresponding value of x_{st} as much as possible, and adjust the values of the other basic variables to compensate for that increase. In our illustration, $\overline{c}_{st} = \overline{c}_{13} = -2$, so we want to introduce x_{13} into the basis. If we consider Fig. 8.5 we see that adding the arc 1–3 to the spanning tree corresponding to the current basis creates a unique loop O_1–D_3–O_2–D_1–O_1 where O and D refer to origin and destination, respectively.

It is easy to see that if we make $x_{st} = \theta$, maintaining all other nonbasic variables equal to zero, the flows on this loop must be adjusted by plus or minus θ, to maintain the feasibility of the solution. The limit to which θ can be increased corresponds to the smallest value of a flow on this loop from which θ must be subtracted. In this example, θ may be increased to 15, corresponding to x_{11} being reduced to zero and therefore dropping out of the basis. The basis has x_{13} replacing x_{11}, and the corresponding spanning tree has arc 1–3 replacing arc 1–1 in Fig. 8.12. Given the new basis, the procedure of determining the shadow prices and then the reduced costs, to check the optimality conditions of the simplex method, can be repeated.

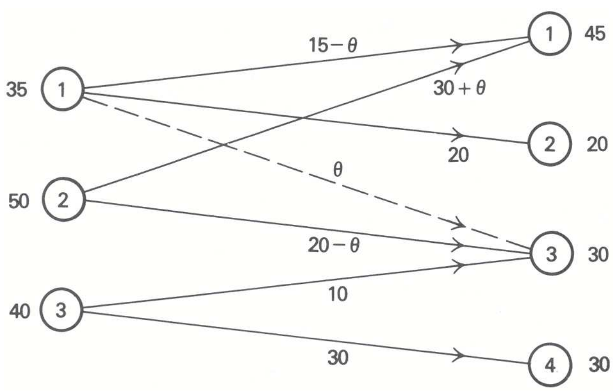


Figure 8.12 Introducing a new variable.

It should be pointed out that the current solution usually is written out not in network form but in tableau form, for case of computation. The current solution given in Fig. 8.12 would then be written as in Tableau 6.

Note that the row and column totals are correct for any value of θ satisfying $0 \leq \theta \leq 15$. The tableau form for the current basic solution is convenient to use for computations, but the justification of its use is most easily seen by considering the corresponding spanning tree. In what follows we will use the tableau form to illustrate the computations. After increasing θ to 15, the resulting basic solution is given in Tableau

7 (ignoring θ) and the new multipliers and reduced costs in Tableau 8.

Tableau 6

$15-\theta$	20	θ		35
$30+\theta$		$20-\theta$		50
		10	30	40
45	20	30	30	

Tableau 7 Current Basic Solution

	$20-\theta$	$15+\theta$		35
45		5		50
	θ	$10-\theta$	30	40
45	20	30	30	

Tableau 8 Reduced Costs

[8] 2	[6] —	[10] —	[9] 10	0
[9] —	[12] 3	[13] —	[7] 5	3
[14] 2	[9] -3	[16] —	[5] —	6
6	6	10	-1	u_i / v_j

Since the only negative reduced cost corresponds to $\bar{c}_{32} = -3$, x_{32} is next introduced into the basis. Adding the arc 3–2 to the spanning tree corresponds to increasing the allocation to cell 3–2 in the tableau and creates a unique loop O_3–D_2–O_1–D_3–O_3. The flow θ on this arc may be increased until $\theta = 10$, corresponding to x_{33} dropping from the basis. The resulting basic solution is Tableau 9 and the new multipliers and reduced costs are given in Tableau 10.

Tableau 9 Current Basic Solution

	10	25		35
45		5		50
	10		30	40
45	20	30	30	

Tableau 10 Reduced Costs

[8] 2	[6] —	[10] —	[9] 7	0
[9] —	[12] 3	[13] —	[7] 2	3
[14] 5	[9] —	[16] 3	[5] —	3
6	6	10	2	u_i / v_j

Since all of the reduced costs are now nonnegative, we have the optimal solution.

8.6 ADDITIONAL TRANSPORTATION CONSIDERATIONS

In the previous section, we described how the simplex method has been specialized in order to solve transportation problems efficiently. There are three steps to the approach:

1. finding an initial basic feasible solution;
2. checking the optimality conditions; and
3. constructing an improved basic feasible solution, if necessary.

The problem we solved was quite structured in the sense that total supply equaled total demand, shipping between all origins and destinations was permitted, degeneracy did not occur, and so forth. In this section, we address some of the important variations of the approach given in the previous section that will handle these situations.

Supply Not Equal to Demand

We have analyzed the situation where the total availability is equal to the total requirement. In practice, this is usually not the case, since often either demand exceeds supply or supply exceeds demand. Let us see how to transform a problem in which this assumption does not hold to the previously analyzed problem with equality of availability and requirement. Two situations will exhaust all the possibilities:

First, assume that the total availability exceeds the total requirement; that is,

$$\sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j = d > 0.$$

In this case, we create an artificial destination $j = n + 1$, with corresponding ‘‘requirement’’ $b_{n+1} = d$, and make the corresponding cost coefficients to this destination equal to zero, that is, $c_{i,n+1} = 0$ for $i = 1, 2, \ldots, m$. The variable $x_{i,n+1}$ in the optimal solution will show how the excess availability is distributed among the sources.

Second, assume that the total requirement exceeds the total availability, that is,

$$\sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i = d > 0.$$

In this case, we create an artificial origin $i = m+1$, with corresponding ‘‘availability’’ $a_{m+1} = d$, and assign zero cost coefficients to this destination, that is, $c_{m+1,j} = 0$ for $j = 1, 2, \ldots, n$. The optimal value for the variable $x_{m+1,j}$ will show how the unsatisfied requirements are allocated among the destinations.

In each of the two situations, we have constructed an equivalent transportation problem such that the total ‘‘availability’’ is equal to the total ‘‘requirement.’’

Prohibited Routes

If it is impossible to ship *any* goods from source i to destination j, we assign a very high cost to the corresponding variable x_{ij}, that is, $c_{ij} = M$, where M is a very large number, and use the procedure previously discussed. If these prohibited routes cannot be eliminated from the optimal solution then the problem is infeasible.

Alternatively, we can use the two-phase simplex method. We start with any initial basic feasible solution, which may use prohibited routes. The first phase will ignore the given objective function and minimize the sum of the flow along the prohibited routes. If the flow on the prohibited routes cannot be driven to zero, then no feasible solution exists without permitting flow on at least one of the prohibited routes. If, on the other hand, flow on the prohibited routes *can* be made zero, then an initial basic feasible solution *without* positive flow on prohibited routes has been constructed. It is necessary then simply to prohibit flow on these routes in the subsequent iterations of the algorithm.

Degeneracy

Degeneracy can occur on two different occasions during the computational process described in the previous section. First, during the computation of the initial solution of the transportation problem, we can simultaneously eliminate a row and a column at an intermediate step. This situation gives rise to a basic solution with less than $(m + n - 1)$ strictly positive variables. To rectify this, one simply assigns a zero value to a cell in either the row or column to be simultaneously eliminated, and treats that variable as a basic variable in the remaining computational process.

As an example, let us apply the northwest-corner rule to the case in Tableau 11.

Tableau 11

	D1	D2	D3	D4	*Supply*
O1	20	5	0		25 5 0
O2			30		30
O3			10		10
O4			10	40	50 40
Demand	20 0	5 0	50 20 10	40	

In this instance, when making $x_{12} = 5$, we simultaneously satisfy the first row availability and the second column requirement. We thus make $x_{13} = 0$, and treat it as a basicvariable in the rest of the computation.

A second situation where degeneracy arises is while improving a current basic solution. A tie might be found when computing the new value to be given to the entering basic variable, $x_{st} = \theta$. In this case, more than one of the old basic variables will take the value zero simultaneously, creating a new basic solution with less than $(m + n - 1)$ strictly positive values. Once again, the problem is overcome by choosing the variable to leave the basis arbitrarily from among the basic variables reaching zero at $x_{st} = \theta$, and treating the remaining variables reaching zero at $x_{st} = \theta$ as basic variables at zero level.

Vogel Approximation

Finally, we should point out that there have been a tremendous number of procedures suggested for finding an initial basic feasible solution to the transportation problem. In the previous section, we mentioned four methods: northwest corner, minimum matrix, minimum column, and minimum row.

The first of these ignores the costs altogether, while the remaining methods allocate costs in such a way that the last fewassignments of flows often results in very high costs being incurred. The high costs are due to the lack of choice as to how the final flows are to be assigned to routes, once the initial flows have been established. The initial flows are not chosen with sufficient foresight as to how they might impair later flow choices.

The Vogel approximation method was developed to overcome this difficulty and has proved to be so effective that it is sometimes used to obtain an approximation to the optimal solution of the problem. The method, instead of sequentially using the least-cost remaining arc, bases its selection on the difference between the two lowest-cost arcs leaving an origin or entering a destination. This difference indicates where departure from the lowest-cost allocations will bring the highest increase in cost. Therefore, one assigns the maximum possible amount to that arc that has the lowest cost in that row or column having the greatest cost difference. If this assignment exhausts the demand at that destination, the corresponding column is eliminated from further consideration; similarly, if the assignment exhausts the supply at that origin, the corresponding row is eliminated. In either case, the origin and destination cost differences are recomputed, and the procedure continues in the same way.

The Vogel approximation method is applied to our illustrative example in Tableau 12, and the resulting basic feasible solution is given in Fig. 8.13. It is interesting to note that the approximation finds the optimal solution in this particular case, as can be seen by comparing the initial basis from the Vogel approximation in Fig. 8.13 with the optimal basis in Tableau 7. This, of course, does not mean that the Vogel approximation is the best procedure for determining the initial basic feasible solution. Any such comparison among computational procedures must compare total time required, from preprocessing to final optimal

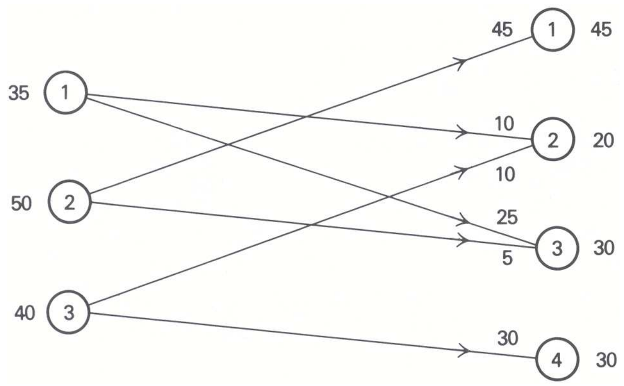


Figure 8.13 Initial basis for Vogel approximation method.

solution.

Tableau 12 Applying the Vogel Approximation Method

First iteration:

	Dallas	Atlanta	San Francisco	Phila-delphia	Supply	Row difference
Cleveland	[8]	[6]	[10]	[9]	35	2
Chicago	[9]	[12]	[13]	[7]	50	2
Boston	[14]	[9]	[16]	[5] 30	~~40~~ 10	4 ←
Demand	45	20	30	~~30~~		
Column difference	1	3	3	2		

Second iteration:

	Dallas	Atlanta	San Francisco	Supply	Row difference
Cleveland	[8]	[6]	[10]	35	2
Chicago	[9]	[12]	[13]	50	3
Boston	[14]	[9] 10	[16]	~~10~~	5 ←
Demand	45	~~20~~ 10	30		
Column difference	1	3	3		

Third iteration:

	Dallas	Atlanta	San Francisco	Supply	Row difference
Cleveland	[8]	[6] 10	[10]	~~35~~ 25	2
Chicago	[9]	[12]	[13]	50	3
Demand	45	~~10~~	30		
Column difference	1	6 ↑	3		

Fourth iteration:

	Dallas	San Francisco	Supply	Row difference
Cleveland	[8]	[10] 25	25	2
Chicago	[9] 45	[13] 5	~~50~~ 45	4 ←
Demand	~~45~~	30		
Column difference	1	3		

8.7 THE SIMPLEX METHOD FOR NETWORKS

The application of the simplex method to the transportation problem presented in the previous section takes advantage of the network structure of the problem and illustrates a number of properties that extend to the general minimum-cost flow problem. All of the models formulated in Section 8.2 are examples of the general minimum-cost flow problem, although
a number, including the transportation problem, exhibit further special structure of their own. Historically, many different algorithms have been developed for each of these models; but, rather than consider each model separately, we will develop the essential step underlying the efficiency of all of the simplex-based algorithms for networks.

We already have seen that, for the transportation problem, a basis corresponds to a spanning tree, and that introducing a new variable into the basis adds an arc to the spanning tree that forms a unique loop. The variable to be dropped from the basis is then determined by finding which variable in the loop drops to zero first when flow is increased on the new arc. It is this property, that bases correspond to spanning trees, that extends to the general minimum-cost flow problem and makes solution by the simplex method very efficient.

In what follows, network interpretations of the steps of the simplex method will be emphasized; therefore, it is convenient todefine some of the network concepts that will be used. Though these concepts are quite intuitive, the reader should be cautioned that the terminology of network flow models has not been standardized, and that definitions vary from one author to another.

Formally, a *network* is defined to be any finite collection of points, called nodes, together with a collection of directed arcs that connect particular pairs of these nodes. By convention, we do not allow an arc to connect a node to itself, but we do allow more than one arc to connect the same two nodes. We will be concerned only with *connected* networks in the sense that *every* node can be reached from every other node by following a sequence of arcs, where the direction of the arcs is ignored. In linear programming, if a network is *disconnected*, then the problem it describes can be treated as separate problems, one for each connected subnetwork.

A *loop* is a sequence of arcs, where the direction of the arcs is ignored, connecting a particular node to itself. In Fig. 8.1, the node sequences 3–4–5–3 and 1–2–3–1 are both examples of loops.

A *spanning tree* is a connected subset of a network including all nodes and containing no loops. Figure 8.7 shows two examples of spanning trees for the minimum-cost flow problem of Fig. 8.1.

It is the concept of a spanning tree, which proved most useful in solving the transportation problem in the previous section, that will be the foundation of the algorithm for the general minimum-cost flow problem.

Finally, an *end* is a node of a network with exactly one arc incident to it. In the first example of Fig. 8.7, nodes 1,2, and 4 are ends, and in the second examples, nodes 1, 2, and 5 are ends. It is easy to see that every tree must have at least two ends. If you start with any node i in a tree and follow any arc away from it, you eventually come to an end, since the tree contains no loops. If node i is an end, then you have two ends. If node i is not an end, there is another arc from node i that will lead to a second end, since again there are no loops in the tree.

In the transportation problem we saw that there are $(m + n - 1)$ basic variables, since any one of the equations is redundant under the assumption that the sum of the supplies equals the sum of the demands. This implies that the number of arcs in any spanning tree corresponding to a basis in the transportation problem is

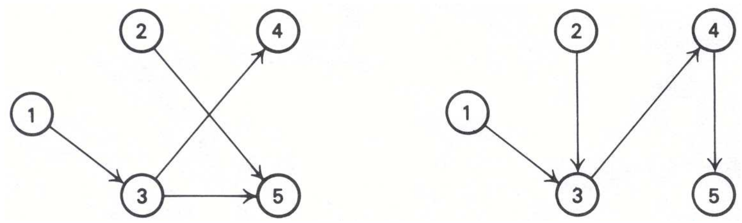


Figure 8.14 Examples of spanning trees.

Tableau 8.13 Tree Variables

x_{13}	x_{25}	x_{34}	x_{35}	*Righthand side*
1				20
	1			0
−1		1	1	0
		−1		−5
	−1		−1	−15

always *one less than* the number of nodes in the network. Note that the number of arcs is one less than the number of nodes in each of the trees shown in Fig. 8.14, since they each contain 5 nodes and 4 arcs. In fact, this characterization of spanning trees holds for the general minimum-cost flow problem.

Spanning-Tree Characterization. A subnetwork of anetwork with n nodes is a spanning tree if and only if it is connected and contains $(n - 1)$ arcs.

We can briefly sketch an inductive proof to show the spanning-tree characterization. The result is clearly true for the two node networks containing one arc. First, we show that is a subnetwork of an n-node network is a spanning tree, it contains $(n - 1)$ arcs. Remove any end and incident arc from the n-node network. The reduced network with $(n - 1)$ nodes is still a tree, and by our inductive assumption it must have $(n - 2)$ arcs. Therefore, the original network with n nodes must have had $(n - 1)$ arcs. Next, we show that if an n-node connected subnetwork has $(n - 1)$ arcs and no loops, it is a spanning tree. Again, remove any end and its incident arc from the n-node network. The reduced network is connected, has $(n - 1)$ nodes, $(n - 2)$ arcs, and no loops; and by our inductive assumption, it must be a spanning tree. Therefore, the original network with n nodes and $(n - 1)$ arcs must be a spanning tree.

The importance of the spanning-tree characterization stems from the relationship between a spanning tree and a basis in the simplex method. We have already seen that a basis for the transportation problem corresponds to a spanning tree, and it is this property that carries over to the general network-flow model.

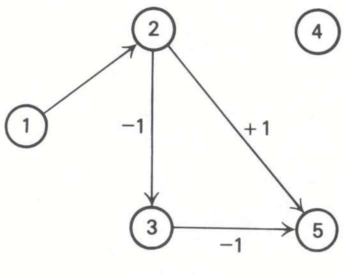
Spanning-Tree Property of Network Bases. In a general minimum-cost flow model, a spanning tree for the network corresponds to a basis for the simplex method.

This is an important property since, together with the spanning-tree characterization, it implies that the number of basic variables is always one less than the number of nodes in a general network-flow problem. Now let us intuitively argue that the spanning-tree property holds, first by showing that the variables corresponding to a spanning tree constitute a basis, and second by showing that a set of basic variables constitutes a spanning tree.

First, assume that we have a network with n nodes, which is a spanning tree. In order to show that the variables corresponding to the arcs in the tree constitute a basis, it is sufficient to show that the $(n - 1)$ tree variables are uniquely determined. In the simplex method, this corresponds to setting the nonbasic variables to specific values and uniquely determining the basic variables. First, set the flows on all arcs not in the tree to either their upper or lower bounds, and update the righthand-side values by their flows. Then choose any node corresponding to an end in the subnetwork, say node k. (There must be at least two ends in the spanning tree since it contains no loops.) Node k corresponds to a row in the linear-programming tableau for the tree with exactly one nonzero coefficient in it. To illustrate this the tree variables of the first example in Fig. 8.14 are given in Tableau 13.

Since there is only one nonzero coefficient in row k, the corresponding arc incident to node k must have flow across it equal to the righthand-side value for that row. In the example above, $x_{13} = 20$. Now, drop node k from further consideration and bring the determined variable over to the righthand side, so that the righthand side of the third constraint becomes $+20$. Now we have an $(n - 1)$-node subnetwork with $(n - 2)$ arcs and no loops. Hence, we have a tree for the reduced network, and the process may be repeated. At each iteration exactly one flow variable is determined. On the last iteration, there are two equations corresponding to two nodes, and one arc joining them. Since we have assumed that the total net flow into or out of the network is zero, the last tree variable will satisfy the last two equations. Hence we have shown that a spanning tree in a network corresponds to a basis in the simplex method. Further, since we have already shown that a tree for a connected network with n nodes contains $(n-1)$ arcs, we have shown that the number of basic variables for a connected network-flow problem is $(n-1)$.

Now assume that we have a network with n nodes and that we know the $(n-1)$ basic variables. To show that these variables correspond to a tree, we need only show that the subnetwork corresponding to the basic variables does not contain any loops. We establish this property by assuming that there *is* a loop and showing that this leads to a contradiction. In Fig. 8.15, we have a four-arc network containing a loop and its associated tableau.



x_{12}	x_{23}	x_{25}	x_{35}	*Righthand side*
1				20
−1	1	1		0
	−1		1	0
				−5
		−1	−1	−15

Figure 8.15 Network containing a loop.

If there exists a loop, then choose weights for the column corresponding to arcs in the loop such that the weight is $+1$ for a forward arc in the loop and -1 for a backward arc in the loop. If we then add the columns corresponding to the loop weighted in this manner, we produce a column containing all zeros. In Fig. 8.15, the loop is 2–5–3–2, and adding the columns for the variables x_{25}, x_{53}, and x_{32} with weights 1, -1, and -1, respectively, produces a zero column. This implies that the columns corresponding to the loop are not independent. Since a basis consists of a set of $(n-1)$ independent columns, any set of variables containing a loop cannot be a basis. Therefore, $(n-1)$ variables corresponding to a basis in the simplex method must be a spanning tree for the network.

If a basis in the simplex method corresponds to a tree, what, then, is the interpretation of introducing a new variable into the basis? Introducing a new variable into the basis adds an arc to the tree, and since every node of the tree is connected to every other node by a sequence of arcs, the addition will form a loop in the subnetwork corresponding to the tree. In Fig. 8.16, arc 4–5 is being introduced into the tree, forming the loop 4–5–3–4. It is easy to argue that adding an arc to a tree creates a *unique* loop in the augmented network. *At least one loop* must be created, since adding the arc connects two nodes that were already connected. Further, *no more than one loop* is created, since, if adding the one arc created more than one loop, the entering arc would have to be common to two distinct loops, which, upon removal of their common arcs, would yield a single loop that was part of the original network.

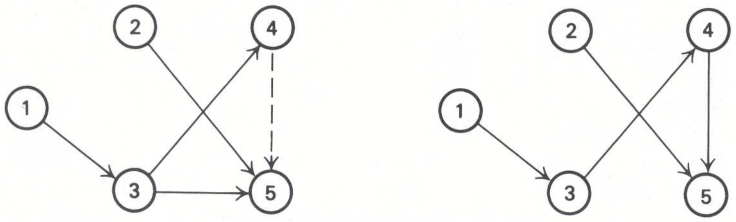


Figure 8.16 Introducing a new arc in a tree.

To complete a basis change in the simplex method, one of the variables currently in the basis must be dropped. It is clear that the variable to be dropped must correspond to one of the arcs in the loop, since dropping any arc not in the loop would leave the loop in the network and hence would not restore the spanning-tree property. We must then be able to determine the arcs that form the loop.

This is accomplished easily by starting with the subnetwork including the loop and eliminating all ends in this network. If the reduced network has no ends, *it* must be the loop. Otherwise, repeat the process of eliminating all the ends in the reduced network, and continue. Since there is a unique loop created by adding an arc to a spanning tree, dropping any arc in that loop will create a new spanning tree, since each node will be connected to every other node by a sequence of arcs, and the resulting network will contain no loops. Figure 8.16 illustrates this process by dropping arc 3–5. Clearly, any arc in the loop could have been dropped, to produce alternative trees.

In the transportation problem, the unique loop was determined easily, although we did not explicitly show how this could be guaranteed. Once the loop is determined, we increased the flow on the incoming arc and adjusted the flows on the other arcs in the loop, until the flow in one of the arcs in the loop was reduced to zero. The variable corresponding to the arc whose flow was reduced to zero was then dropped out of the basis. This is essentially the same procedure that will be employed by the general minimum-cost flow problem, except that the special rules of the simplex method with upper and lower bounds on the variables will be employed. In the next section an example is carried out that applies the simplex method to the general minimum-cost flow problem.

Finally, we should comment on the integrality property of the general minimum-cost flow problem. We saw that, for the transportation problem, since the basis corresponds to a spanning tree, as long as the supplies and demands are integers, the flows on the arcs for a basic solution are integers. This is also true for the general minimum-cost flow problem, so long as the net flows at any node are integers and the upper and lower bounds on the variables are integers.

Integrality Property. In the general minimum-cost flow problem, assuming that the upper and lower bounds on the variables are integers and the righthand-side values for the flow-balance equations are integers, the values of the basic variables are also integers when the nonbasic variables are set to their upper or lower bounds.

In the simplex method with upper and lower bounds on the variables, the nonbasic variables are at either their upper orlower bound. If these bounds are integers, then the net flows at all nodes, when the flows on the nonbasic arcs are included, are also integers. Hence, the flows on the arcs corresponding to the basic variables also will be integers, since these flows are determined by first considering all ends in the corresponding spanning tree and assigning a flow to the incident arc equal to the net flow at the node. These assigned flows must clearly be integers. The ends and the arcs incident to them are then eliminated, and the process is repeated, yielding an integer assignment of flows to the arcs in the reduced tree at each stage.

Tableau 8.14 Basis variables

x_{13}	x_{25}	x_{34}	x_{35}	*Righthand side*	*Row no.*
1				20	1
	1			0	2
−1		1	1	0	3
		−1		−5	4
	−1		−1	−15	5

The integrality property of the general minimum-cost flow problem was established easily by using the fact that a basis corresponds to a spanning tree. Essentially, all ends could be immediately evaluated, then eliminated, and the procedure repeated. We were able to solve a system of equations by recognizing that at least one variable in the system could be evaluated by inspection at each stage, since at each stage at least

Tableau 15 A basis is triangular.

x_{25}	x_{34}	x_{35}	x_{13}	*Righthand side*	*Row no.*
1				0	2
	−1			−5	4
−1		−1		−15	5
	1	1	−1	0	3

one equation would have only one basic variable in it. A system of equations with this property is called *triangular.*

In Tableau E8.14 we have rewritten the system of equations corresponding to the tree variables given in Tableau E8.13. Then we have arbitrarily dropped the first equation, since a connected network with n nodes has $(n - 1)$ basic variables. We have rearranged the remaining variables and constraints to exhibit the triangular form in Tableau 15.

The variables on the diagonal of the triangular system then may be evaluated sequentially, starting with the first equation. Clearly $x_{25} = 0$. Then, moving the evaluated variable to the righthand side, we have a new triangular system with one less equation. Then the next diagonal variable may be evaluated in the same way and the procedure repeated. It is easy to see that, for our example, the values of the variables are $x_{25} = 0$, $x_{34} = 5$, $x_{35} = 15$, $x_{13} = 20$. Note that the value of x_{13} satisfies the first equation that was dropped. It should be pointed out that many other systems of equations besides network-flow problems can be put in the form of a triangular system and therefore can be easily solved.

8.8 SOLVING THE MINIMUM-COST FLOW PROBLEM

In this section we apply the simplex method to the general minimum-cost flow problem, using the network concepts developed in the previous section. Consider the minimum-cost flow problem given in Section 8.1 and repeated here in Fig. 8.17 for reference.

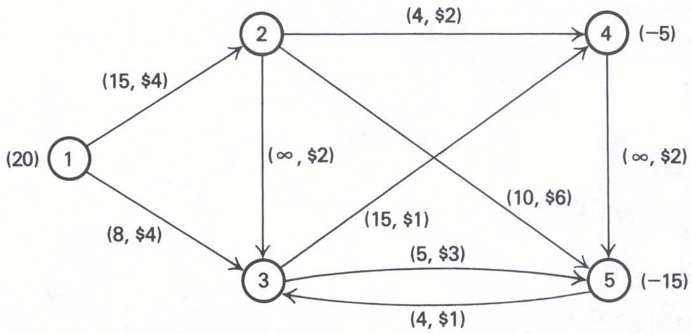


Figure 8.17 Minimum-cost flow problem.

This problem is more complicated than the transportation problem since it contains intermediate nodes (points of transshipment) and capacities limiting the flow on some of the arcs.

In order to apply the simplex method to this example, we must first determine a basic feasible solution. Whereas, in the case of the transportation problem, an initial basic feasible solution is easy to determine (by the northwest-corner method, the minimum matrix method, or the Vogel approximation method), in the general case an initital basic feasible solution may be difficult to find. The difficulty arises from the fact that the upper and lower bounds on the variables are treated implicitly and, hence, nonbasic variables may be at either bound. We will come back to this question later. For the moment, assume that we have been given the initial basic feasible solution shown in Fig. 8.18.

The dash–dot arcs 1–3 and 3–5 indicate nonbasic variables at their upper bounds of 8 and 5, respectively.

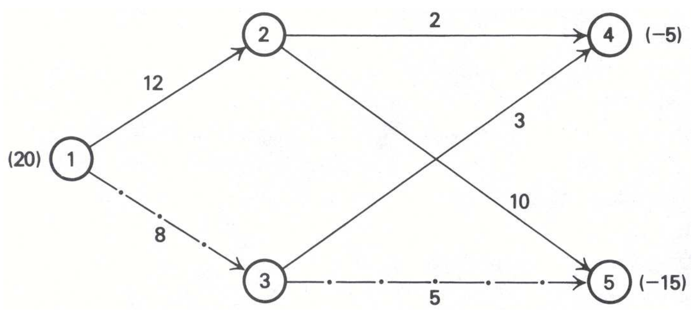


Figure 8.18 Initial basic feasible solution.

The arcs not shown are nonbasic at their lower bounds of zero. The solid arcs form a spanning tree for the network and constitute a basis for the problem.

To detemine whether this initial basic feasible solution is optimal, we must compute the reduced costs of all nonbasic arcs. To do this, we first determine multipliers $y_i (i = 1, 2, \ldots, n)$ and, if these multipliers satisfy:

$$\begin{aligned} \bar{c}_{ij} &= c_{ij} - y_i + y_j \geq 0 && \text{if } x_{ij} = \ell_{ij}, \\ \bar{c}_{ij} &= c_{ij} - y_i + y_j = 0 && \text{if } \ell_{ij} < x_{ij} < u_{ij}, \\ \bar{c}_{ij} &= c_{ij} - y_i + y_j \leq 0 && \text{if } x_{ij} = u_{ij}, \end{aligned}$$

then we have an optimal solution. Since the network-flow problem contains a redundant constraint, any one multiplier may be chosen arbitrarily, as was indicated in previous sections. Suppose $y_2 = 0$ is set arbitrarily; the remaining multipliers are determined from the equations:

$$c_{ij} - y_i + y_j = 0$$

for basic variables. The resulting multipliers for the initial basis are given as node labels in Fig. 8.19. These were determined from the cost data given in Fig. 8.17.

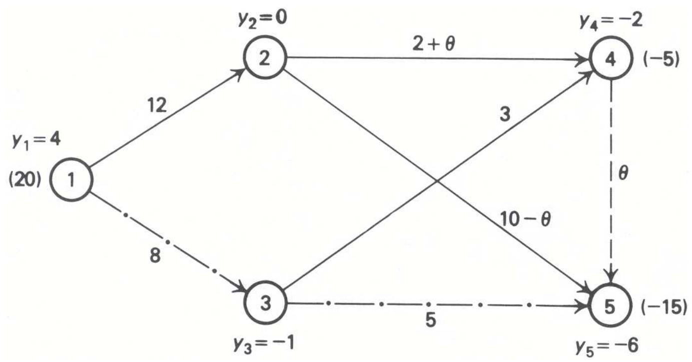


Figure 8.19 Iteration 1.

Given these multipliers, we can compute the reduced costs of the nonbasis variables by $\bar{c}_{ij} = c_{ij} - y_i + y_j$. The reduced costs are determined by using the given cost data in Fig. 8.17 as:

$$\begin{aligned} \bar{c}_{13} &= 4 - 4 + (-1) &&= -1, \\ \bar{c}_{23} &= 2 - 0 + (-1) &&= 1, \\ \bar{c}_{35} &= 3 - (-1) + (-6) &&= -2, \\ \bar{c}_{45} &= 2 - (-2) + (-6) &&= -2, \leftarrow \\ \bar{c}_{53} &= 1 - (-6) + (-1) &&= 6. \end{aligned}$$

In the simplex method with bounded variables, the nonbasic variables are at either their upper or lower bounds. An improved solution can be found by either:

1. increasing a variable that has a negative reduced cost and is currently at its lower bound; or
2. decreasing a variable that has a positive reduced cost and is currently at its upper bound.

In this case, the only promising candidate is x_{45}, since the other two negative reduced costs correspond to nonbasic variables at their upper bounds. In Fig. 8.19 we have added the arc 4–5 to the network, forming the unique loop 4–5–2–4 with the basic variables. If the flow on arc 4–5 is increased by θ, the remaining arcs in the loop must be appropriately adjusted. The limit on how far we can increase θ is given by arc 2–4, which has an upper bound of 4. Hence, $\theta = 2$ and x_{24} becomes nonbasic at its upper bound. The corresponding basic feasible solution is given in Fig. 8.20, ignoring the dashed arc 2–3.

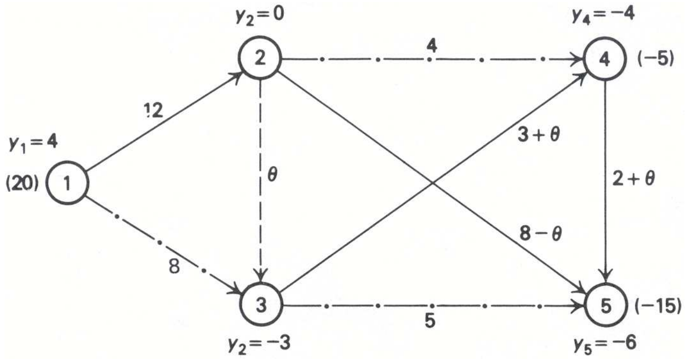


Figure 8.20 Iteration 2.

The new multipliers are computed as before and are indicated as node labels in Fig. 8.20. Note that not all of the multipliers have to be recalculated. Those multipliers corresponding to nodes that are connected to node 2 by the same sequence of arcs as before will not change labels. The reduced costs for the new basis are then:

$$\begin{aligned}
\bar{c}_{13} &= 4 - 4 + (-3) &&= -3,\\
\bar{c}_{23} &= 2 - 0 + (-3) &&= -1, \leftarrow\\
\bar{c}_{24} &= 2 - 0 + (-4) &&= -2,\\
\bar{c}_{35} &= 3 - (-3) + (-6) &&= 0,\\
\bar{c}_{53} &= 1 - (-6) + (-3) &&= 4.
\end{aligned}$$

Again there is only one promising candidate, x_{23}, since the other two negative reduced costs correspond to nonbasic variables at their upper bounds. In Fig. 8.20 we have added the arc 2–3 to the network, forming the unique loop 2–3–4–5–2 with the basic variables. If we increase the flow on arc 2–3 by θ and adjust the flows on the remaining arcs in the loop to maintain feasibility, the increase in θ is limited by arc 2–5. When $\theta = 8$, the flow on arc 2–5 is reduced to zero and x_{25} becomes nonbasic at its lower bound. Figure 8.21 shows the corresponding basic feasible solution.

The new multipliers are computed as before and are indicated as node labels in Fig. 8.8. The reduced costs for the new basis are then:

$$\begin{aligned}
\bar{c}_{13} &= 4 - 4 + (-2) &&= -2,\\
\bar{c}_{24} &= 2 - 0 + (-3) &&= -1,\\
\bar{c}_{25} &= 6 - 0 + (-5) &&= 1,\\
\bar{c}_{35} &= 3 - (-2) + (-5) &&= 0,\\
\bar{c}_{53} &= 1 - (-5) + (-2) &&= 4.
\end{aligned}$$

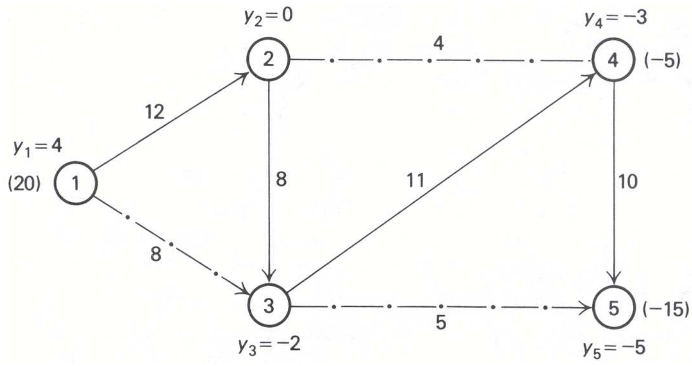


Figure 8.21 Optimal solution.

This is an optimal solution, since all negative reduced costs correspond to nonbasic variables at their upper bounds and all positive reduced costs correspond to nonbasic variables at their lower bounds.

The reduced cost $\bar{c}_{35} = 0$ indicates that alternative optimal solutions may exist. In fact, it is easily verified that the solution given in Fig. 8.22 is an alternative optimal solution.

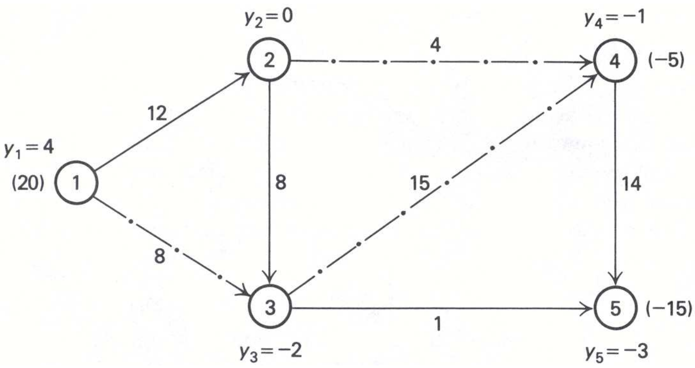


Figure 8.22 Alternative optimal solution.

Hence, given an initial basic feasible solution, it is straightforward to use the concepts developed in the previous section to apply the simplex method to the general minimum-cost flow problem. There are two essential points to recognize: (1) a basis for the simplex method corresponds to a spanning tree for the network, and (2) introducing a new variable into the basis forms a unique loop in the spanning tree, and the variable that drops from the basis is the limiting variable in this loop.

Finally, we must briefly discuss how to find an initial basic feasible solution if one is not readily available. There are a number of good heuristics for doing this, depending on the particular problem, but almost all of these procedures involve adding some artificial arcs at some point. It is always possible to add uncapacitated arcs from the points of supply to the points of demand in such a fashion that a basis is formed. In our illustrative example, we could have simply added the artificial arcs 1–4 and 1–5 and had the initial basis given in Fig. 8.23.

Then either a phase I procedure is performed, minimizing the sum of the flows on the artificial arcs, or a very high cost is attached to the artificial arcs to drive them out of the basis. Any heuristic procedure for finding an initial basis usually attempts to keep the number of artificial arcs that have to be added as small as possible.

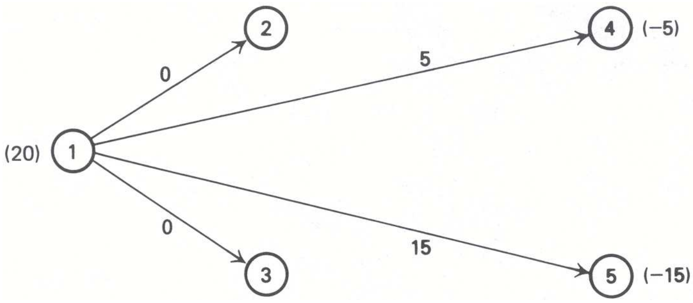


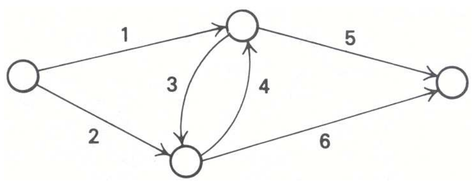
Figure 8.23 Artificial initial basis.

EXERCISES

1. A gas company owns a pipeline network, sections of which are used to pump natural gas from its main field to its distribution center. The network is shown below, where the direction of the arrows indicates the only direction in which the gas can be pumped. The pipeline links of the system are numbered one through six, and the intermediate nodes are large pumping stations. At the present time, the company nets 1200 mcf (million cubic feet) of gas per month from its main field and must transport that entire amount to the distribution center. The following are the maximum usage rates and costs associated with each link:

	1	*2*	*3*	*4*	*5*	*6*
Maximum usage: mcf/month	500	900	700	400	600	1000
Tariff: $/mcf	20	25	10	15	20	40

The gas company wants to find those usage rates that minimize total cost of transportation.



a) What are the decision variables?
b) Formulate the problem as a linear program.
c) For the optimal solution, do you expect the dual variable associated with the maximum usage of link 1 to be positive, zero, or negative and why?
d) Suppose there were maximum usage rates on the pumping stations; how would your formulation change?

2. On a particular day during the tourist season a rent-a-car company must supply cars to four destinations according to the following schedule:

Destination	*Cars required*
A	2
B	3
C	5
D	7

The company has three branches from which the cars may be supplied. On the day in question, the inventory status of each of the branches was as follows:

Branch	*Cars available*
1	6
2	1
3	10

The distances between branches and destinations are given by the following table:

Branch	Destination A	B	C	D
1	7	11	3	2
2	1	6	0	1
3	9	15	8	5

Plan the day's activity such that supply requirements are met at a minimum cost (assumed proportional to car-miles travelled).

3. The National Association of Securities Dealers Automated Quotation Systems (NASDAQ) is a network system in which quotation information in over-the-counter operations is collected. Users of the system can receive, in a matter of seconds, buy and sell prices and the exact bid and ask price of each market maker that deals in a particular security. There are 1700 terminals in 1000 locations in almost 400 cities. The central processing center is in Trumbull, Conn., with concentration facilities in New York, Atlanta, Chicago, and San Francisco. On this particular day, the market is quiet, so there are only a few terminals being used. The information they have has to be sent to one of the main processing facilities. The following table gives terminals (supply centers), processing facilities (demand centers), and the time that it takes to transfer a message.

Terminals	Trumbull	N.Y.	Atlanta	Chicago	San Fran.	*Supply*
Cleveland	6	6	9	4	10	45
Boston	3	2	7	5	12	90
Houston	8	7	5	6	4	95
Los Angeles	11	12	9	5	2	75
Washington,D.C.	4	3	4	5	11	105
Demand	120	80	50	75	85	

a) Solve, using the minimum matrix method to find an initial feasible solution.
b) Are there alternative optimal solutions?

4. A large retail sporting-goods chain desires to purchase 300, 200, 150, 500, and 400 tennis racquets of five different types. Inquiries are received from four manufacturers who will supply not more than the following quantities (all five types of racquets combined).

M1	600
M2	500
M3	300
M4	400

The store estimates that its profit per racquet will vary with the manufacturer as shown below:

Manufacturer	*Racquets* R1	R2	R3	R4	R5
M1	5.50	7.00	8.50	4.50	3.00
M2	6.00	6.50	9.00	3.50	2.00
M3	5.00	7.00	9.50	4.00	2.50
M4	6.50	5.50	8.00	5.00	3.50

How should the orders be placed?

5. A construction project involves 13 tasks; the tasks, their estimated duration, and their immediate predecessors are shown in the table below:

Task	*Immediate predecessors*	*Duration*
Task 1	—	1
Task 2	1	2
Task 3	—	3
Task 4	—	4
Task 5	1	2
Task 6	2,3	1
Task 7	4	2
Task 8	5	6
Task 9	5	10
Task 10	6,7	5
Task 11	8,10	3
Task 12	8,10	3
Task 13	12	2

Our objective is to find the schedule of tasks that minimizes the total elapsed time of the project.

a) Draw the event- and task-oriented networks for this problem and formulate the corresponding linear program.
b) Solve to find the critical path.

6. The Egserk Catering Company manages a moderate-sized luncheon cafeteria featuring prompt service, delectable cuisine, and luxurious surroundings. The desired atmosphere requires fresh linen napkins, which must be available at the start of each day. Normal laundry takes one full day at 1.5 cents per napkin; rapid laundry can be performed overnight but costs 2.5 cents a napkin. Under usual usage rates, the current napkin supply of 350 is adequate to permit complete dependence upon the normal laundry; however, the additional usage resulting from a three-day seminar to begin tomorrow poses a problem. It is known that the napkin requirements for the next three days will be 300, 325, and 275, in that order. It is now midafternoon and there are 175 fresh napkins, and 175 soiled napkins ready to be sent to the laundry. It is against the health code to have dirty napkins linger overnight. The cafeteria will be closed the day after the seminar and, as a result, all soiled napkins on the third day can be sent to normal laundry and be ready for the next business day.

   The caterer wants to plan for the napkin laundering so as to minimize total cost, subject to meeting all his fresh napkin requirements and complying with the health code.

a) What are the decision variables?
b) Formulate the problem as a linear program.
c) Interpret the resulting model as a network-flow problem. Draw the corresponding network diagram.
d) For the optimal solution, do you expect the dual variable associated with tomorrow's requirement of 300 to be positive, zero, or negative, and why?
e) Suppose you could hold over dirty napkins at no charge; how would your formulation change?

7. An automobile association is organizing a series of car races that will last for four days. The organizers know that $r_j \geq 0$ special tires in good condition will be required on each of the four successive days, $j = 1, 2, 3, 4$. They can meet these needs either by buying new tires at P dollars apiece or by reshaping used tires (reshaping is a technique by which the grooves on the tire are deepened, using a special profile-shaped tool). Two kinds of service are available for reshaping: normal service, which takes one full day at N dollars a tire, and quick service, which takes overnight at Q dollars a tire. How should the association, which starts out with no special tires, meet the daily requirements at minimal cost?

a) Formulate a mathematical model for the above problem. Does it exhibit the characteristics of a network problem? Why? (*Hint.* Take into account the fact that, at the end of day j, some used tires may not be sent to reshaping.)
b) If the answer to (a) is *no*, how can the formulation be manipulated to become a network problem? Draw the associated network. (*Hint.* Add a redundant constraint introducing a fictitious node.)
c) Assume that a tire may be reshaped only once. How does the above model change? Will it still be network problem?

8. Conway Tractor Company has three plants located in Chicago, Austin (Texas), and Salem (Oregon). Three customers located respectively in Tucson (Arizona), Sacramento (California), and Charlestown (West Virginia) have placed additional orders with Conway Tractor Company for 10, 8, and 10 tractors, respectively. It is customary for Conway Tractor Company to quote to customers a price on a *delivered* basis, and hence the company absorbs the delivery costs of the tractors. The manufacturing cost does not differ significantly from one plant to another, and the following tableau shows the delivery costs incurred by the firm.

	Destination		
Plant	Tucson	Sacramento	Charlestown
Chicago	150	200	70
Austin	70	120	80
Salem	80	50	170

The firm is now facing the problem of assigning the extra orders to its plants to minimize delivery costs and to meet all orders (The Company, over the years, has established a policy of first-class service, and this includes quick and reliable delivery of all goods ordered). In making the assignment, the company has to take into account the limited additional manufacturing capacity at its plants in Austin and Salem, of 8 and 10 tractors, respectively. There are no limits on the additional production capacity at Chicago (as far as these extra orders are concerned).

a) Formulate as a transportation problem.
b) Solve completely.

9. A manufacturer of electronic calculators produces its goods in Dallas, Chicago, and Boston, and maintains regional warehousing distribution centers in Philadelphia, Atlanta, Cleveland, and Washington, D.C. The company's staff has determined that shipping costs are directly proportional to the distances from factory to storage center, as listed here.

	Warehouses			
Mileage from:	Philadelphia	Atlanta	Cleveland	Washington
Boston	300	1000	500	400
Chicago	500	900	300	600
Dallas	1300	1000	1100	1200

The cost per calculator-mile is \$0.0002 and supplies and demands are:

Supply		*Demand*	
Boston	1500	Philadelphia	2000
Chicago	2500	Atlanta	1600
Dallas	4000	Cleveland	1200
		Washington	3200

a) Use the Vogel approximation method to arrive at an initial feasible solution.
b) Show that the feasible solution determined in (a) is optimal.

c) Why does the Vogel approximation method perform so well, compared to other methods of finding an initial feasible solution?

10. Colonel Cutlass, having just taken command of the brigade, has decided to assign men to his staff based on previous experience. His list of major staff positions to be filled is adjutant (personnel officer), intelligence officer, operations officer, supply officer, and training officer. He has five men he feels could occupy these five positions. Below are their years of experience in the several fields.

	Adjutant	*Intelligence*	*Operations*	*Supply*	*Training*
Major Muddle	3	5	6	2	2
Major Whiteside	2	3	5	3	2
Captain Kid	3	—	4	2	2
Captain Klutch	3	—	3	2	2
Lt. Whiz	—	3	—	1	—

Who, based on experience, should be placed in which positions to give the greatest total years of experience for all jobs? (*Hint*. A basis, even if degenerate, is a spanning tree.)

11. Consider the following linear program:

$$\text{Minimize } z = 3x_{12} + 2x_{13} + 5x_{14} + 2x_{41} + x_{23} + 2x_{24} + 6x_{42} + 4x_{34} + 4x_{43},$$

subject to:

$$\begin{aligned}
x_{12} + x_{13} + x_{14} - x_{41} \quad & \leq 8, \\
x_{12} - x_{23} - x_{24} + x_{42} \quad & \geq 4, \\
x_{34} - x_{13} - x_{23} - x_{43} \quad & \leq 4, \\
x_{14} + x_{34} + x_{24} - x_{42} - x_{42} - x_{43} \quad & \geq 5, \\
\text{all } x_{ij} \quad & \geq 0.
\end{aligned}$$

a) Show that this is a network problem, stating it in general minimum-cost flow form. Draw the associated network and give an interpretation to the flow in this network.
b) Find an initial feasible solution. (*Hint*. Exploit the triangular property of the basis.)
c) Show that your initial solution is a spanning tree.
d) Solve completely.

12. A lot of three identical items is to be sequenced through three machines. Each item must be processed first on machine 1, then on machine 2, and finally on machine 3. It takes 20 minutes to process one item on machine 1, 12 minutes on machine 2, and 25 minutes on machine 3. The objective is to minimize the total work span to complete all the items.

a) Write a linear program to achieve our objective. (*Hint*. Let x_{ij} be the starting time of processing item i on machine j. Two items may not occupy the same machine at the same time; also, an item may be processed on machine $(j+1)$ only after it has been completed on machine j.)
b) Cast the model above as a network problem. Draw the associated network and give an interpretation in terms of flow in networks. (*Hint*. Formulate and interpret the dual problem of the linear program obtained in (a).)
c) Find an initial feasible solution; solve completely.

13. A manufacturer of small electronic calculators is working on setting up his production plans for the next six months. One product is particularly puzzling to him. The orders on hand for the coming season are:

Month	*Orders*
January	100
February	150
March	200
April	100
May	200
June	150

The product will be discontinued after satisfying the June demand. Therefore, there is no need to keep any inventory after June. The production cost, using regular manpower, is $10 per unit. Producing the calculator on overtime costs an additional $2 per unit. The inventory-carrying cost is $0.50 per unit per month. If the regular shift production is limited to 100 units per month and overtime production is limited to an additional 75 units per month, what is the optimal production schedule? (*Hint.* Treat regular and overtime capacities as sources of supply for each month.)

14. Ships are available at three ports of origin and need to be sent to four ports of destination. The number of ships available at each origin, the number required at each destination, and the sailing times are given in the tableau below. Our objective is to minimize the total number of sailing days.

Origin \ *Destination*	1	2	3	4	*Number of ships available*
1	5	4	3	2	5
2	10	8	4	7	5
3	9	9	8	4	5
Number of ships required	1	4	4	6	15

a) Find an initial basic feasible solution.
b) Show that your initial basis is a spanning tree.
c) Find an initial basic feasible solution using the other two methods presented in the text. Solve completely, starting from the three initial solutions found in parts (a) and (c). Compare how close these solutions were to the optimal one.
d) Which of the dual variables may be chosen arbitrarily, and why?
e) Give an economic interpretation of the optimal simplex multipliers associated with the origins and destinations.

15. A distributing company has two major customers and three supply sources. The corresponding unit from each supply center to each customer is given in the following table, together with the total customer requirements and supply availabilities.

Supply center	*Customer* 1	*Customer* 2	*Available supplies*
1	−1	3	300
2	1	6	400
3	1	5	900
Customer requirements	800	500	

Note that Customer 1 has strong preferences for Supplier 1 AND will be willing not only to absorb all the transportation costs but also to pay a premium price of $1 per unit of product coming from Supplier 1.

a) The top management of the distributing company feels it is obvious that Supply Center 1 should send all its available products to Customer 1. Is this necessarily so? (*Hint.* Obtain the least-cost solution to the problem. Explore whether alternative optimal solutions exist where not all the 300 units available in Supply Center 1 are assigned to Customer 1.)
b) Assume Customer 2 is located in an area where all shipments will be subject to taxes defined as a percentage of the unit cost of a product. Will this tax affect the optimal solution of part (a)?
c) Ignore part (b). What will be the optimal solution to the original problem if Supply Center 1 increases its product availability from 300 units to 400 units?

16. After solving a transportation problem with positive shipping costs c_{ij} along all arcs, we increase the supply at one source and the requirement at one destination in a manner that will maintain equality of total supply and total demand.

a) Would you expect the shipping cost in the modified problem with a larger total shipment of goods to be higher than the optimal shipping plan from the original problem?
b) Solve the following transportation problem:

	Unit shipping costs to destinations			
Source	*D1*	*D2*	*D3*	*Supplies*
S1	4	2	4	15
S2	12	8	4	15
Requirements	10	10	10	

c) Increase the supply at source S1 by 1 unit and the demand at demand center *D*3 by 1 unit, and re-solve the problem. Has the cost of the optimal shipping plan decreased? Explain this behavior.

17. Consider a very elementary transportation problem with only two origins and two destinations. The supplies, demands, and shipping costs per unit are given in the following tableau.

	D1	*D2*	*Units supplied*
S1	5	2	20
S2	8	4	80
Units demanded	50	50	

Since the total number of units supplied equals the total number of units demanded, the problem may be formulated with equality constraints. An optimal solution to the problem is:

$$x_{11} = 20, \qquad x_{12} = 0, \qquad x_{21} = 30, \qquad x_{22} = 50;$$

and a corresponding set of shadow prices on the nodes is:

$$y_{s1} = 4, \qquad y_{s2} = 0, \qquad y_{d1} = 1, \qquad y_{d2} = 4.$$

a) Why is the least expensive route not used?
b) Are the optimal values of the decision variables unique?
c) Are the optimal values of the shadow prices unique?
d) Determine the ranges on the righthand-side values, changed one at a time, for which the basis remains unchanged.
e) What happens when the ranges determined in (d) are exceeded by some small amount?

18. Consider a transportation problem identical to the one given in Exercise 17. One way the model may be formulated is as a linear program with inequality constraints. The formulation and solution are given below.

	x_{11}	x_{12}	x_{21}	x_{22}	*Relation*	RHS	*Shadow prices*
Supply 1	1	1	0	0	$\leq$	20	−3
Supply 2	0	0	1	1	$\leq$	80	0
Demand 1	1	0	1	0	=	50	8
Demand 2	0	1	0	1	=	50	4
Costs	5	2	8	4	=	z (min)	
Solution	20	0	30	50			

For this formulation of the model:

a) Are the optimal values of the shadow prices unique?
b) Determine the ranges on the righthand-side values, changed one at a time, for which the basis remains unchanged.
c) Reconcile the results of (b) with those obtained in Exercise 17.

19. Suppose that there are three suppliers S_1, S_2, and S_3 in a distribution system that can supply 5, 5, and 6 units, respectively, of the company's good. The distribution system contains five demand centers, that require 2, 2, 4, 4, and 3 units each of the good. The transportation costs, in dollars per unit, are as follows:

	D_1	D_2	D_3	D_4	D_5
S_1	2	1	2	3	3
S_2	2	2	2	1	−1
S_3	3	3	2	1	2

a) Compute an optimum shipping schedule. Is the optimal solution unique?
b) Find the range over which the cost of transportation from S_1 to D_3 can vary while maintaining the optimal basis found in part (a).
c) To investigate the sensitivity of the solution to this problem, we might consider what happens if the amount supplied from any *one* supplier and the amount demanded by any *one* demand center were both increased. Is it possible for the total shipping costs to decrease by increasing the supply and the demand for any particular choice of supply and demand centers? Establish a limit on these increases as specific pairs of supply and demand centers are selected.
d) A landslide has occurred on the route from S_2 to D_5. If you bribe the state highway crew with \$10, they will clear the slide. If not, the route will remain closed. Should you pay the bribe?

20. An oil company has three oil fields and five refineries. The production and transportation costs from each oil field each refinery, in dollars per barrel, are given in the table below:

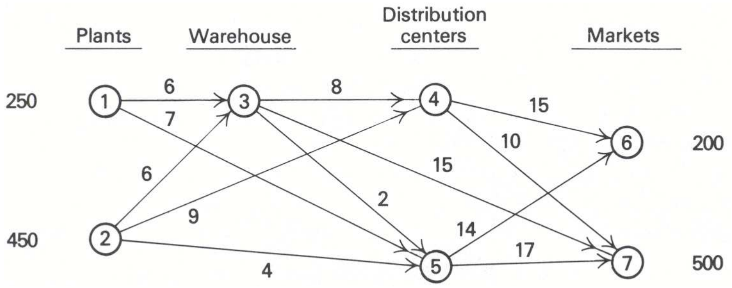
	Refineries					
Oil field	R1	R2	R3	R4	R5	*Availability*
OF1	5	3	3	3	7	4
OF2	5	4	4	2	1	6
OF3	5	4	2	6	2	7
Requirements	2	2	3	4	4	

The corresponding production capacity of each field and requirements of each refinery, in *millions of barrels* per week, are also given.

a) What is the optimum weekly production program? Parts (b), (c), and (d) are independent, each giving modifications to part (a).
b) Suppose that field OF1 has worked *under* capacity so far, and that its production increases by one unit (i.e., 1 million barrels). What is the new optimal production plan? Has the optimal basis changed? How does the objective function change? What is the range within which the production of field OF1 may vary?
c) Because of pipeline restrictions, it is impossible to send more than 1 million barrels from OF2 to R5. How would you have formulated the problem if it has been stated in this form from the very beginning? (*Hint.* Decompose R5 into two destinations: one with a requirement of one unit (i.e., 1 million barrels), the other with a requirement of three. Prohibit the route from OF2 to the second destination of R5.) Change the optimum solution in (a) to find the new optimum program.

d) Suppose that fields OF1, OF2, and OF3 have additional overtime capacities of 1, 1, and 1.5 units, respectively (that is, 1, 1, and 1.5 million barrels, respectively). This causes an increase in the corresponding production costs of 0.5, 1.5, and 2 dollars per barrel, respectively. Also assume that the refinery requirements are increased by one million barrels at each refinery and that there is no convenient route to ship oil from field OF2 to refinery R3. What is the optimum program?

21. Consider the following transshipment problem where goods are shipped from two plants to either a warehouse or two distribution centers and then on to the two end markets.



The production rates in units per month are 250 and 450 for plants 1 and 2, respectively. The demands of the two markets occur at rates 200 and 500 units per month. The costs of shipping one unit over a particular route is shown adjacent to the appropriate arc in the network.

a) Redraw the above transshipment network as a network consisting of only origins and destinations, by replacing all intermediate nodes by two nodes, an origin and destination, connected by an arc, from the destination back to the origin, whose cost is zero.

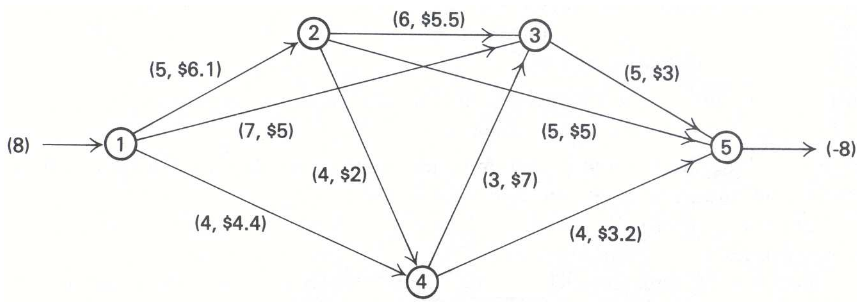
b) The network in (a) is a transportation problem except that a backwards arc, with flow x_{ii}, from newly created destination i to its corresponding newly created source, is required. Convert this to a transportation network by substituting $x'_{ii} = B - x_{ii}$. How do you choose a value for the constant B?

c) Certain arcs are inadmissible in the original transshipment formulation; how can these be handled in the reformulated transportation problem?

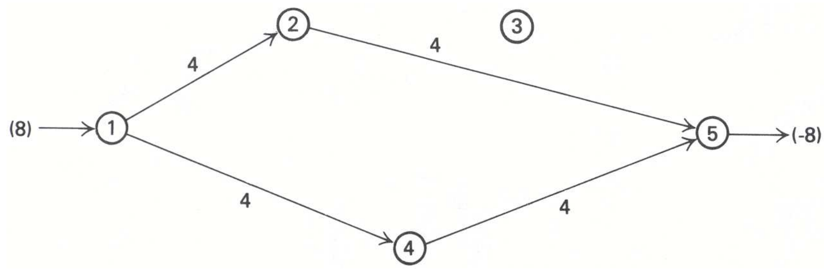
d) Interpret the linear-programming tableau of the original transshipment network and that of the reformulated transportation network.

e) Can any transshipment problem be transformed into an equivalent transportation problem?

22. Consider the following minimum-cost flow model:



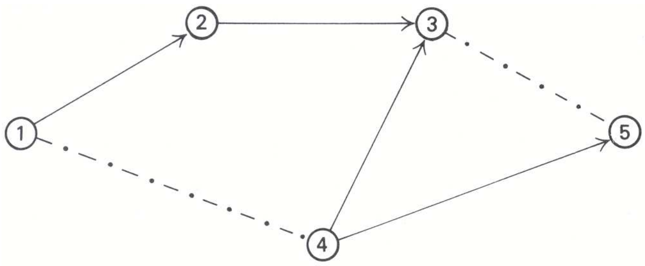
We wish to send eight units from node 1 to node 5 at minimum cost. The numbers next to the arcs indicate upper bounds for the flow on an arc and the cost per unit of flow. The following solution has been proposed, where the numbers next to the arcs are flows.



The total cost of the proposed solution is \$66.8.

a) Is the proposed solution a feasible solution? Is it a basic feasible solution? Why?
b) How can the proposed solution be modified to constitute a basic feasible solution?
c) Determine multipliers on the nodes associated with the basic feasible solution given in (b). Are these multipliers unique?
d) Show that the basic feasible solution determine in (b) is not optimal.
e) What is the next basis suggested by the reduced costs? What are the *values* of the new basic variables? Nonbasic variables?

23. For the minimum-cost flow model given in Exercise 22, suppose that the spanning tree indicated by the solid lines in the following network, along with the dash–dot arcs at their upper bounds, has been proposed as a solution:



a) What are the flows on the arcs corresponding to this solution?
b) Determine a set of shadow prices for the nodes.
c) Show that this solution is optimal.
d) Is the optimal solution unique?

24. For the minimum-cost flow model given in Exercise 22, with the optimal solution *determined in Exercise 23*, answer the following questions.

a) For each nonbasic variable, determine the range on its objective-function coefficient so that the current basis remains optimal.
b) For each basic variable, determine the range on its objective-function coefficient so that the current basis remains optimal.
c) Determine the range on each righthand-side value so that the basis remains unchanged.
d) In question (c), the righthand-side ranges are all tight, in the sense that any change in one righthand-side value *by itself* will apparently change the basis. What is happening?

25. The following model represents a simple situation of buying and selling of a seasonal product for profit:

$$\text{Maximize } z = -\sum_{t=1}^{T} p_t x_t + \sum_{t=1}^{T} s_t y_t,$$

subject to:

$$I + \sum_{j=1}^{t}(x_j - y_i) \leq C \qquad (t = 1, 2, \ldots, T),$$

$$y_1 \leq I,$$

$$y_2 \leq I + (x_1 - y_1),$$

$$y_3 \leq I + (x_1 - y_1) + (x_2 - y_2),$$

$$y_t \leq I + \sum_{j=1}^{t-1}(x_j - y_j) \qquad (t = 4, 5, \ldots, T),$$

$$x_t \geq 0, \; y_t \geq 0 \qquad \text{for } t = 1, 2, \ldots, T.$$

In this formulation, the decision variables x_t and y_t denote, respectively, the amount of the product purchased and sold, in time period t. The given data is:

p_t = per unit purchase price in period t,
s_t = per unit selling price in period t,
I = amount of the product on hand initially,
C = capacity for storing the product in any period.

The constraints state (i) that the amount of the product on hand at the end of any period cannot exceed the storage capacity C, and (ii) that the amount of the product sold at the beginning of period t cannot exceed its availability from previous periods.

a) Let

$$w_t = C - I - \sum_{j=1}^{t}(x_j - y_j) \qquad \text{and} \qquad z_t = I + \sum_{j=1}^{t-1}(x_j - y_j) - y_t,$$

denote slack variables for the constraints (with $z_1 = I - y_1$). Show that the given model is equivalent to the formulation on page 363.

b) State the dual to the problem formulation in part (a), letting u_i denote the dual variable for the constraint $+x_i$, and v_j denote the dual variable for the constraint containing $+y$.

c) Into what class of problems does this model fall? Determine the nature of the solution for $T = 3$.

26. A set of words (for example, ace, bc, dab, dfg, fe) is to be transmitted as messages. We want to investigate the possibility of representing each word by one of the letters *in the word* such that the words will be represented uniquely. If such a representation is possible, we can transmit a single letter instead of a complete word for a message we want to send.

a) Using as few constraints and variables as possible, formulate the possibility of transmitting letters to represent words as the solution to a mathematical program. Is there anything special about the structure of this program that facilitates discovery of a solution?

b) Suppose that you have a computer code that computes the solution to the following transportation problem:

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

subject to:

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

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad (j = 1, 2, \ldots, m),$$

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

and requires that:

(i) $a_i,\ b_j,\ c_{ij} \geq 0$ and integer $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$,

(ii) $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$.

$$\text{Maximize } z = -\sum_{t=1}^{T} p_t x_t + \sum_{t=1}^{T} s_t y_t,$$

subject to:

$$\begin{array}{llllllllllllllll}
x_1 & & & & & -y_1 & & & & +w_1 & & & & & & = C - I, \\
& x_2 & & & & & -y_2 & & & -w_1 & +w_2 & & & & & = 0, \\
& & x_3 & & & & & -y_3 & & & -w_2 & +w_3 & & & & = 0, \\
& & & \ddots & & & & & \ddots & & & \ddots & & & & \vdots \\
& & & & x_T & & & & -y_T & & & -w_{T-1} & +w_T & & & = 0, \\
& & & & & y_1 & & & & & & & & +z_1 & & = I, \\
-x_1 & & & & & & +y_2 & & & & & & & -z_1 & +z_2 & = 0, \\
& -x_2 & & & & & & +y_3 & & & & & & & -z_2 + z_3 & = 0, \\
& & \ddots & & & & & & \ddots & & & & & & \ddots & \vdots \\
& & & -x_{T-1} & & & & & +y_T & & & & & & -z_{T-1} + z_T & = 0,
\end{array}$$

$$x_t \geq 0,\ y_t \geq 0,\ w_t \geq 0,\ z_t \geq 0 \text{ for } t = 1, \ldots, T.$$

How would you use this code to solve the problem posed in (a)? Answer the question in general, and use the specific problem defined above as an illustration.

27. The Defense Communications Agency is responsible for operating and maintaining a world-wide communications system. It thinks of costs as being proportional to the ''message units'' transmitted in one direction over a particular link in the system. Hence, under normal operating conditions it faces the following minimum-cost flow problem:

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

subject to:

$$\sum_j x_{ij} - \sum_k x_{ki} = b_i \qquad (i = 1, 2, \ldots, n),$$
$$0 \leq x_{ij} \leq u_{ij},$$

where
c_{ij} = cost per message unit over link $(i - j)$.
b_i = message units generated (or received) at station i,
u_{ij} = upper bound on number of message units that can be transmitted over link $(i - j)$.

Suppose that the agency has been given a budget of B dollars to spend on increasing the capacity of any link in the system. The price for increasing capacity on link $(i - j)$ is p_{ij}.

a) Formulate a linear program (not necessarily a network) with only $(n + 1)$ constraints, that will minimize operating costs subject to this budget constraint. (*Hint.* You must allow for investing in additional capacity for each link.)
b) How can the ''near'' network model formulated in (a) be analyzed by network-programming techniques? (*Hint.* How could parametric programming be used?)

28. The general minimum-cost flow problem is given as follows:

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

subject to:

$$\sum_j x_{ij} - \sum_k x_{ki} = b_i \qquad (i = 1, 2, \ldots, n),$$
$$\ell_{ij} \leq x_{ij} \leq u_{ij}.$$

a) Assuming that the lower bounds on the variables are all finite, this is, $\ell_{ij} > -\infty$, show that any problem of this form can be converted to a transportation problem with lower bounds on the variables of zero and nonnegative, or infinite, upper bounds. (*Hint*. Refer to Exercise 22.)
b) Comment on the efficiency of solving minimum-cost flow problems by a bounded-variables transportation method, versus the simplex method for general networks.

29. One difficulty with solving the general minimum-cost flow problem with upper and lower bounds on the variables lies in determining an initial basic feasible solution. Show that an initial basic feasible solution to this problem can be determined by solving an appropriate maximum-flow problem. (*Hints*. (1) Make a variable substitution to eliminate the nonzero lower bounds on the variables. (2) Form a ''super source,'' connected to all the source nodes, and a ''super sink,'' connected to all the sink nodes, and maximize the flow from super sink to super source.)

ACKNOWLEDGMENTS

Exercises 1 and 6 are due to Sherwood C. Frey, Jr., of the Harvard Business School.
Exercise 6, in turn, is based on ''The Caterer Problem,'' in *Flows in Networks* by L. R. Ford and D. R. Fulkerson, Princeton University Press, 1962. In their 1955 article ''Generalizations of the Warehousing Model,'' from the *Operational Research Quarterly*, A. Charnes and W. W. Cooper introduced the transformation used in Exercise 25.