Math 323 Operations Research **Notes on Chapter 8** C.K. Li

Network models

Basic definitions

A graph or a network consists of a vertex set $V = \{v_1, \ldots, v_n\}$ and a set of arcs A containing a selection of order pairs (v_i, v_j) of vertices. The vertex v_i is the initial node, and the vertex if the terminal node of the arc.

A chain is a sequence of arcs such that each arc (starting from the second one) has one vertex in common with the previous arc.

A path is a chain such that (each arc starting from the second one) has the its initial node equal to the terminal node of the previous arc.

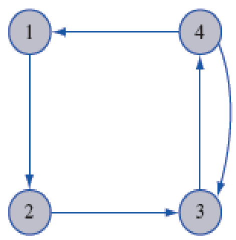


FIGURE 1
Example of a Network

8.2 Shortest path problems

Shortest path problem. Given a network so that every arc is associated with a distance. Find a path from vertex i to vertex t with shortest distance, i.e., the sum of distance of the arcs of the path is minimum.

Dijkstra's algorithm

1. List the vertices $[1, \ldots, n]$ with 1 as the initial vertex, n is the final vertex, and set assign a permanent label 0 to the initial vertex 1.

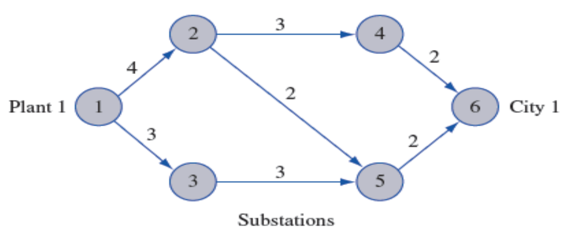
2. Assign temporary label to each other vertex j as the cost of the arc $(1, j)$, which is ∞ if $(1, j)$ is not an arc. Select a vertex with minimum label to be permanent.

3. Suppose k is the permanent vertex labeled most recently. Update the label of each non-permanent vertex j using the minimum of the original label of j and the sum of label k and the cost from k to j, which is ∞ if (j, k) is not an arc.

4. Select the undated temporary label with minimum cost as a new permanent labeled vertex. If it is the terminal vertex, we are done. Back track the permanent label to recover the shortest path. Else, go back to Step 3.

Example

FIGURE 2
Network for Powerco



[0* 4 3 ∞ ∞ ∞]

[0* 4 3* ∞ ∞ ∞]

Remark We can formulate the shortest path problem as a transshipment problem.

- Set each vertex except the final one as a supply point with supply value 1.
- Set each vertex except the initial one as a demand point with demand value 1.
- Assign cost c_{ij} from supply point i to demand point j.
- Find the minimum cost for the transportation problem.

TABLE 3
Transshipment Representation of Shortest-Path Problem and Optimal Solution (1)

Node	2	3	4	5	6	Supply
1	[4] 1	[3]	[M]	[M]	[M]	1
2	[0]	[M]	[3]	[2] 1	[M]	1
3	[M]	[0] 1	[M]	[3]	[M]	1
4	[M]	[M]	[0] 1	[M]	[2]	1
5	[M]	[M]	[M]	[0]	[2] 1	1
Demand	1	1	1	1	1	

To illustrate the preceding ideas, we formulate the balanced transportation problem associated with finding the shortest path from node 1 to node 6 in Figure 2. We want to send one unit from node 1 to node 6. Node 1 is a supply point, node 6 is a demand point, and nodes 2, 3, 4, and 5 will be transshipment points. Using $s = 1$, we obtain the balanced transportation problem shown in Table 3. This transportation problem has two optimal solutions:

1 $z = 4 + 2 + 2 = 8$, $x_{12} = x_{25} = x_{56} = x_{33} = x_{44} = 1$ (all other variables equal 0). This solution corresponds to the path 1–2–5–6.

2 $z = 3 + 3 + 2 = 8$, $x_{13} = x_{35} = x_{56} = x_{22} = x_{44} = 1$ (all other variables equal 0). This solution corresponds to the path 1–3–5–6.

Related problem - equipment replacement. Suppose an equipment is purchased. There are repair cost, trade in cost, etc.

Find the minimum net purchase cost = purchase cost + maintenance cost − trade in price.

Example Buying a new car costs \$12000 For instance, annual maintenance cost (M) for a car of n year old, and the trade in (T) price for a car of n year old.

n	0	1	2	3	4	5
M	2000	4000	5000	9000	12000	
T		7000	6000	2000	1000	0

Set up a network with vertices $V = \{1, \ldots, 6\}$. The arc from (i, j) with $i < j$ has a cost

c_{ij} = buying price of a car + maintenance cost from year i to $j - 1$ - trade in cost in year.

We can build up the network and find the "shortest path".

$c_{12} = 2 + 12 - 7 = 7$

$c_{13} = 2 + 4 + 12 - 6 = 12$

$c_{14} = 2 + 4 + 5 + 12 - 2 = 21$

$c_{26} = 2 + 4 + 5 + 9 + 12 - 1 = 31$

$c_{34} = 2 + 12 - 7 = 7$

$c_{35} = 2 + 4 + 12 - 6 = 12$

$c_{36} = 2 + 4 + 5 + 12 - 2 = 21$

$c_{16} = 2 + 4 + 5 + 9 + 12 + 12 - 0 = 44$

$c_{23} = 2 + 12 - 7 = 7$

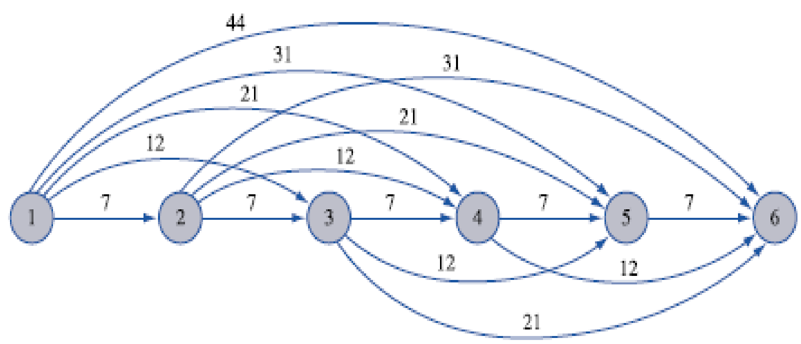
$c_{24} = 2 + 4 + 12 - 6 = 12$

$c_{45} = 2 + 12 - 7 = 7$

$c_{46} = 2 + 4 + 12 - 6 = 12$

$c_{56} = 2 + 12 - 7 = 7$

FIGURE 3
Network for Minimizing Car Costs



8.3 Maximum flow - minimum cut problems

Problem In a network so that each arc (i, j) has a flow capacity c_{ij} constraint. Find the maximum flow value from a source vertex so to a sink vertex si.

EXAMPLE 3 **Maximum Flow**

Sunco Oil wants to ship the maximum possible amount of oil (per hour) via pipeline from node *so* to node *si* in Figure 6. On its way from node *so* to node *si*, oil must pass through some or all of stations 1, 2, and 3. The various arcs represent pipelines of different diameters. The maximum number of barrels of oil (millions of barrels per hour) that can be pumped through each arc is shown in Table 8. Each number is called an **arc capacity.** Formulate an LP that can be used to determine the maximum number of barrels of oil per hour that can be sent from *so* to *si*.

TABLE 8
Arc Capacities for Sunco Oil

Arc	Capacity
(*so*, 1)	2
(*so*, 2)	3
(1, 2)	3
(1, 3)	4
(3, *si*)	1
(2, *si*)	2

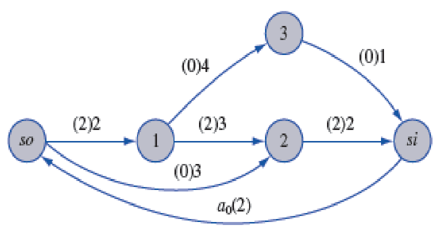


FIGURE 6
Network for Sunco Oil

$$\max z = x_0$$

$$
\begin{aligned}
\text{s.t.} \quad x_{so,1} &\le 2 && \text{(Arc capacity constraints)}\\
x_{so,2} &\le 3 \\
x_{12} &\le 3 \\
x_{2,si} &\le 2 \\
x_{13} &\le 4 \\
x_{3,si} &\le 1 \\
x_0 &= x_{so,1} + x_{so,2} && \text{(Node } so \text{ flow constraint)}\\
x_{so,1} &= x_{12} + x_{13} && \text{(Node 1 flow constraint)}\\
x_{so,2} + x_{12} &= x_{2,si} && \text{(Node 2 flow constraint)}\\
x_{13} &= x_{3,si} && \text{(Node 3 flow constraint)}\\
x_{3,si} + x_{2,si} &= x_0 && \text{(Node } si \text{ flow constraint)}\\
x_{ij} &\ge 0
\end{aligned}
$$

One optimal solution to this LP is $z = 3$, $x_{so,1} = 2$, $x_{13} = 1$, $x_{12} = 1$, $x_{so,2} = 1$, $x_{3,si} = 1$, $x_{2,si} = 2$, $x_0 = 3$. Thus, the maximum possible flow of oil from node *so* to *si* is 3 million barrels per hour, with 1 million barrels each sent via the following paths: *so*–1–2–*si*, *so*–1–3–*si*, and *so*–2–*si*.

Remark We can always set up the maximum flow problem as a transportation problem:

$$\max Z = x_0$$

$$\text{subject to:} \quad \sum_i f_{ij} - \sum_k f_{jk} = 0, \quad j = 1, \ldots, n.$$

$$\sum f_{so,i} = x_0, \qquad \sum f_{j,si} = x_0, \qquad 0 \le f_{ij} \le c_{ij}.$$

Ford-Fulkerson Algorithm

Consider a capacitated network with source vertex so and sink vertex si. Partition $V = S \cup \overline{S}$ with $so \in S, si \in \overline{S}$. Define the cut associated with $(S, \overline{S})$ as $K(S, \overline{S}) = \sum_{(i,j)\in(S\times\overline{S}} c_{ij}$. Then maximum flow in the network equals the minimum cut.

1. Set initial flow to be 0.

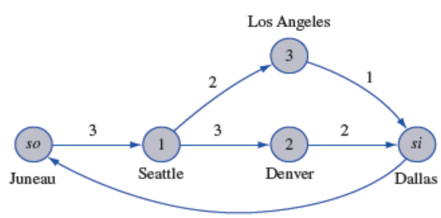
2. Find a chain from so to si consisting of non-saturated forward arcs $c_{ij} - f_{ij} > 0$, and backward arcs with non-zero flows f_{rs}; increase the flow by the minimum of the values $c_{ij} - f_{ij}$ and f_{rs}.

   This can be done by labeling the vertices starting from so; after adding a new round of newly labeled vertices, move on to the next round by labeling those vertices connected with those labeled in the last round by forward non-saturated forward arcs or and backward arcs with positive flow until we reach si, or find it impossible.

3. If no such chain exists, then we have an optimal flow. (Letting S be the vertices reachable from so with a positive chain. Then $K(S, \overline{S})$ is a minimum cut.)

Example

FIGURE 7
Network for Fly-by-Night Airlines



Solution The appropriate network is given in Figure 7. Here the capacity of arc (i, j) is the maximum number of daily flights between city i and city j. The optimal solution to this maximum flow problem is $z = x_0 = 3$, $x_{J,S} = 3$, $x_{S,L} = 1$, $x_{S,De} = 2$, $x_{L,D} = 1$, $x_{De,D} = 2$. Thus, Fly-by-Night can send three flights daily connecting Juneau and Dallas. One flight connects via Juneau–Seattle–L.A.–Dallas, and two flights connect via Juneau–Seattle–Denver–Dallas.

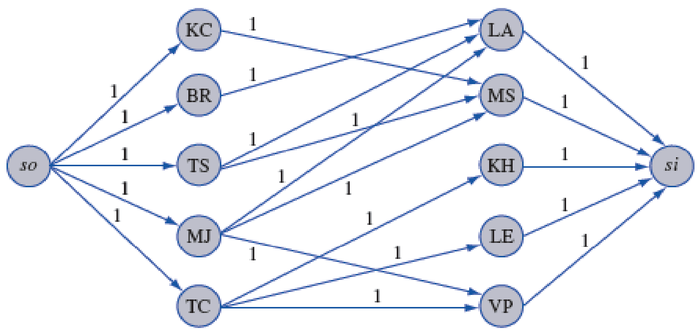
A special case - Matching problem. Try to match n boys $b_1, \ldots, b_n$ with n girls $g_1, \ldots, g_n$ with $c_{i,j} = 1$ if b_i know g_j. Set a source vertex so and a sink vertex si such that capacity constraint from so to b_i equals 1, and the capacity constraint from g_i to si equals 1, for $i = 1, \ldots, n$.

Example

Compatibilities for Matching

	Loni Anderson	Meryl Streep	Katharine Hepburn	Linda Evans	Victoria Principal
Kevin Costner	—	C	—	—	—
Burt Reynolds	C	—	—	—	—
Tom Selleck	C	C	—	—	—
Michael Jackson	C	C	—	—	C
Tom Cruise	—	—	C	C	C

FIGURE 8 Network for Matchmaker



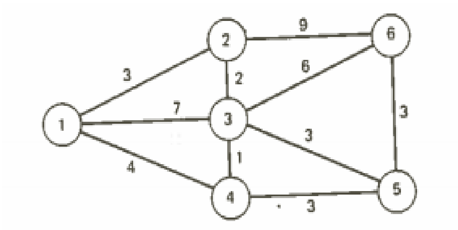
Hall's Theorem There is a complete matching if and only if every group of k boys know at least k girls for $k = 1, \ldots, n$.

Theorem Given a capacitated network with source vertex so and sink vertex si. Then there is a flow with value x_0 if and only if

$$\sum_{i,j \in S \times \overline{S}} c_{ij} - \sum_{i,j \in \overline{S} \times S} c_{ij} \geq x_0.$$

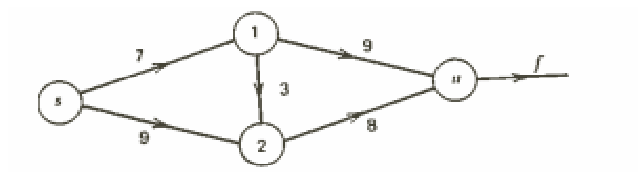
Additional Examples and Algorithms

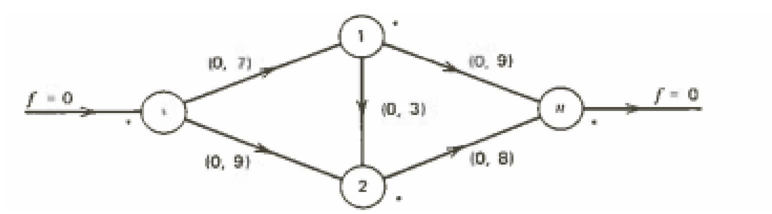
Example 1. Shortest Path

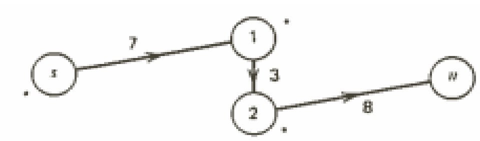


0. $[0^*, 3, 7, 4, \infty, \infty]$
1. $[0^*, 3^*, 7, 4, \infty, \infty]$
2. $[0^*, 3^*, 5, 4^*, \infty, 12]$
3. $[0^*, 3^*, 5*, 4^*, 7, 12]$
4. $[0^*, 3^*, 5^*, 4^*, 7^*, 11]$
5. $[0^*, 3^*, 5^*, 4^*, 7^*, 10^*]$

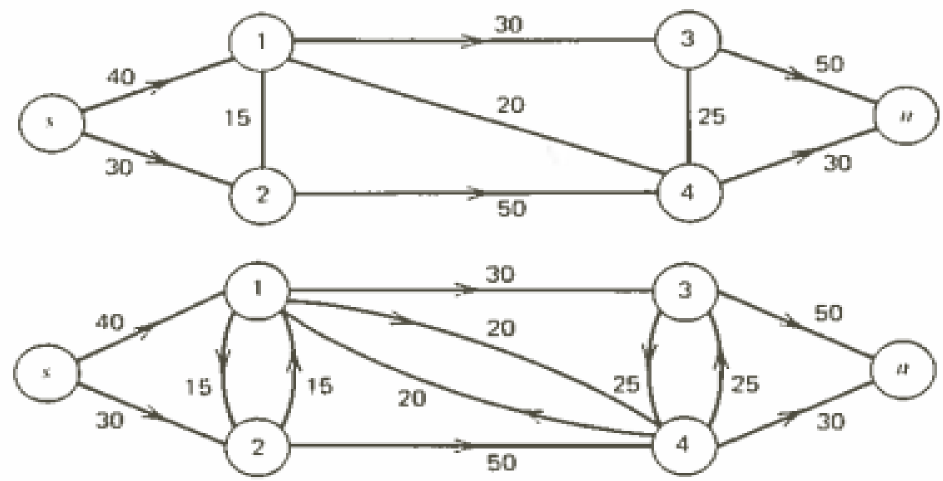
Example 2. A maximal flow problem



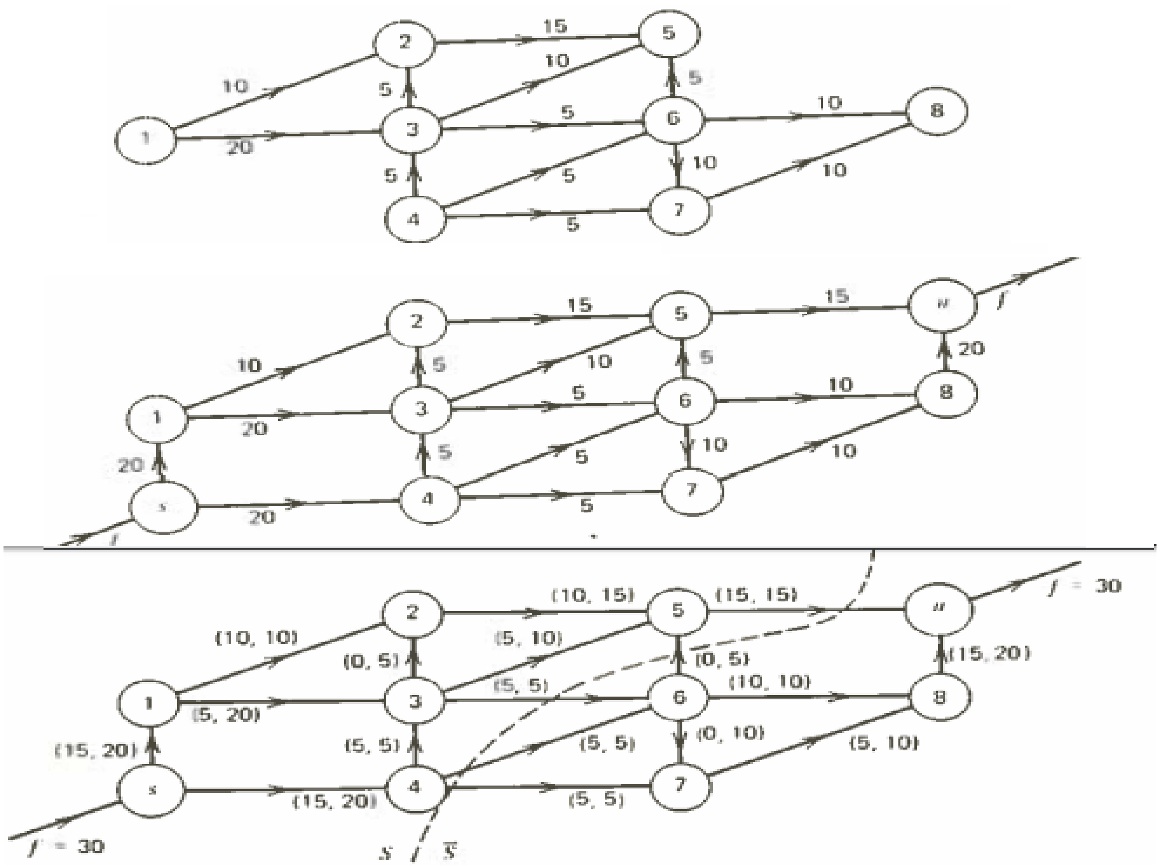




Example 3. Another maximal flow example with an undirected arc



Example 4. A transshipment example with multiple sources and sinks



8.4 Critical Path Method (CPM), Project Evaluation and Review Techniques (PERT)

One can use network models to deal with scheduling problem of large complex projects with many activities - construction, building, manufacturing, launching (scientific, commercial, industrial) projects, etc.

Basic set up

1. Node 1 represents the start of the project. An arc should lead from node 1 to represent each activity that has no predecessors.
2. A node (called the finish node) representing the completion of the project should be included in the network.
3. Number the nodes in the network so that the node representing the completion of an activity always has a larger number than the node representing the beginning of an activity (there may be more than one numbering scheme that satisfies rule 3).
4. 4 An activity should not be represented by more than one arc in the network.
5. Two nodes can be connected by at most one arc.

FIGURE 26
Activity A Must Be Completed Before Activity B Can Begin

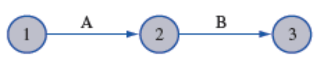


FIGURE 27
Activities A and B Must Be Completed Before Activity C Can Begin

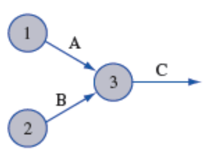


FIGURE 28
Activity A Must Be Completed Before Activities B and C Can Begin

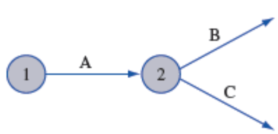


FIGURE 29
Violation of Rule 5

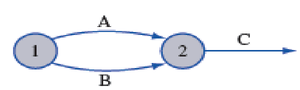
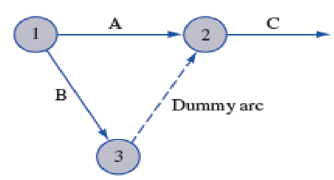


FIGURE 30
Use of Dummy Activity



Remark To avoid violating rules 4 and 5, it is sometimes necessary to utilize a dummy activity that takes zero time.

Example Suppose activities A and B are both predecessors of activity C and can begin at the same time. In the absence of rule 5, we could represent this by Figure 29. However, because nodes 1 and 2 are connected by more than one arc, Figure 29 violates rule 5. By using a dummy activity (indicated by a dotted arc), as in Figure 30, we may represent the fact that A and B are both predecessors of C. Figure 30 ensures that activity C cannot begin until both A and B are completed, but it does not violate rule 5.

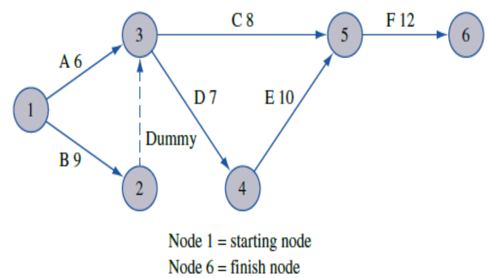
Example Widgetco is about to introduce a new product (product 3).

- One unit of product 3 is produced by assembling 1 unit of product 1 and 1 unit of product 2.
- Before production begins on either product 1 or 2, raw materials must be purchased and workers must be trained.
- Before products 1 and 2 can be assembled into product 3, the finished product 2 must be inspected. A list of activities and their predecessors and of the duration of each activity is given in Table 12. Draw a project diagram for this project.

TABLE 12

Duration of Activities and Predecessor Relationships for Widgetco

Activity	Predecessors	Duration (Days)
A = train workers	—	6
B = purchase raw materials	—	9
C = produce product 1	A, B	8
D = produce product 2	A, B	7
E = test product 2	D	10
F = assemble products 1 and 2	C, E	12



Definition The early event time for node i, represented by $ET(i)$, is the earliest time at which the event corresponding to node i can occur.

The late event time for node i, represented by $LT(i)$, is the latest time at which the event corresponding to node i can occur without delaying the completion of the project.

Computation of early event time

Set $ET(1) = 0$. In gneral, if $ET(j)$ is known for $j < i$, we can find $LT(i)$ as follows.

Step 1 Find each prior event to node i that is connected by an arc to node i. These events are the immediate predecessors of node i.

Step 2 To the ET for each immediate predecessor of the node i add the duration of the activity connecting the immediate predecessor to node i.

Step 3 $ET(i)$ equals the maximum of the sums computed in step 2.

Computation of late event time

Set $LT(n)$ to be the finish time. In general, if $LT(j)$ is known for $j > i$, we can find $LT(i)$ as follows:

Step 1 Find each node that occurs after node i and is connected to node i by an arc.

These events are the immediate successors of node i.

Step 2 From the LT for each immediate successor to node i, subtract the duration of the activity joining the successor the node i.

Step 3 $LT(i)$ is the smallest of the differences determined in step 2.

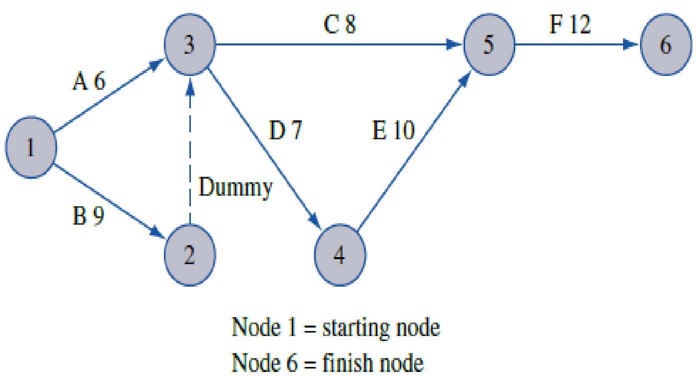


TABLE 13
ET and *LT* for Widgetco

Node	$ET(i)$	$LT(i)$
1	0	0
2	9	9
3	9	9
4	16	16
5	26	26
6	38	38

Total flow

For an arbitrary arc representing activity (i, j), the total float, represented by $TF(i, j)$, of the activity represented by (i, j) is the amount by which the starting time of activity (i, j) could be delayed beyond its earliest possible starting time without delaying the completion of the project (assuming no other activities are delayed). So, $TF(i, j) + ET(i) + t_{ij} \leq LT(j)$, and hence

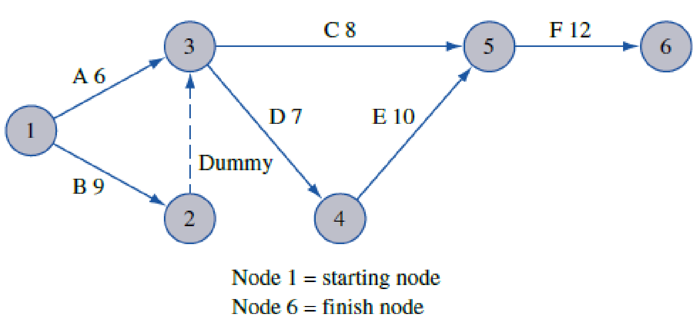
$$TF(i, j) = LT(j) - ET(i) - t_{ij}.$$

Activity B: $TF(1, 2) = LT(2) - ET(1) - 9 = 0$
Activity A: $TF(1, 3) = LT(3) - ET(1) - 6 = 3$
Activity D: $TF(3, 4) = LT(4) - ET(3) - 7 = 0$
Activity C: $TF(3, 5) = LT(5) - ET(3) - 8 = 9$
Activity E: $TF(4, 5) = LT(5) - ET(4) - 10 = 0$
Activity F: $TF(5, 6) = LT(6) - ET(5) - 12 = 0$
Dummy activity: $TF(2, 3) = LT(3) - ET(2) - 0 = 0$

Find a Critical Path Definitions Any activity with a total float of zero is a critical activity.

A path from node 1 to the finish node that consists entirely of critical activities is called a critical path.

- In our example, activities B, D, E, F, and the dummy activity are critical activities and the path 1-2--3-4-5-6 is the critical path.
- We can use the minimal path algorithm to find the critical path with $TF(i, j)$ as the distance.
- It is possible for a network to have more than one critical path.
- A critical path in any project network is the longest path from the start node to the finish node.
- Any delay in the duration of a critical activity will delay the completion of the project, so it is advisable to monitor closely the completion of critical activities.



Free float

The free float of the activity corresponding to arc (i, j), denoted by $FF(i, j)$, is the amount by which the starting time of the activity corresponding to arc (i, j) (or the duration of the activity) can be delayed without delaying the start of any later activity beyond its earliest possible starting time. So, $ET(i) + t_{ij} + FF(i, j) \leq ET(j)$, and hence

$$FF(i, j) = ET(j) - ET(i) - t_{ij}.$$

In our example, we have the following.

Activity B: $FF(1, 2) = 9 - 0 - 9 = 0$

Activity A: $FF(1, 3) = 9 - 0 - 6 = 3$

Activity D: $FF(3, 4) = 16 - 9 - 7 = 0$

Activity C: $FF(3, 5) = 26 - 9 - 8 = 9$

Activity E: $FF(4, 5) = 26 - 16 - 10 = 0$

Activity F: $FF(5, 6) = 38 - 26 - 12 = 0$

Using Linear Programming to Find a Critical Path

$$\begin{aligned}
&\min z = x_6 - x_1 \\
&\text{s.t.} \quad x_3 \geq x_1 + 6 \qquad &&\text{(Arc (1, 3) constraint)} \\
&\qquad\;\; x_2 \geq x_1 + 9 &&\text{(Arc (1, 2) constraint)} \\
&\qquad\;\; x_5 \geq x_3 + 8 &&\text{(Arc (3, 5) constraint)} \\
&\qquad\;\; x_4 \geq x_3 + 7 &&\text{(Arc (3, 4) constraint)} \\
&\qquad\;\; x_5 \geq x_4 + 10 &&\text{(Arc (4, 5) constraint)} \\
&\qquad\;\; x_6 \geq x_5 + 12 &&\text{(Arc (5, 6) constraint)} \\
&\qquad\;\; x_3 \geq x_2 &&\text{(Arc (2, 3) constraint)} \\
&\text{All variables urs}
\end{aligned}$$

An optimal solution to this LP is $z = 38$, $x_1 = 0$, $x_2 = 9$, $x_3 = 9$, $x_4 = 16$, $x_5 = 26$, and $x_6 = 38$. This indicates that the project can be completed in 38 days.

Crashing the project

Suppose that by allocating additional resources to an activity, Widgetco can reduce the duration of any activity by as many as 5 days. The cost per day of reducing the duration of an activity is shown in Table 14. To find the minimum cost of completing the project by the 25-day deadline, define variables A, B, C, D, E, and F as follows:

A = number of days by which duration of activity A is reduced

⋮ ⋮

F = number of days by which duration of activity F is reduced

x_j = time that the event corresponding to node j occurs

TABLE 14

A	B	C	D	E	F
\$10	\$20	\$3	\$30	\$40	\$50

Then Widgetco should solve the following LP:

$$\min z = 10A + 20B + 3C + 30D + 40E + 50F$$

$$\begin{array}{llll}
\text{s.t.} & A \le 5 & x_2 \ge x_1 + 9 - B & \text{(Arc (1, 2) constraint)} \\
& B \le 5 & x_3 \ge x_1 + 6 - A & \text{(Arc (1, 3) constraint)} \\
& C \le 5 & x_5 \ge x_3 + 8 - C & \text{(Arc (3, 5) constraint)} \\
& D \le 5 & x_4 \ge x_3 + 7 - D & \text{(Arc (3, 4) constraint)} \\
& E \le 5 & x_5 \ge x_4 + 10 - E & \text{(Arc (4, 5) constraint)} \\
& F \le 5 & x_6 \ge x_5 + 12 - F & \text{(Arc (5, 6) constraint)} \\
& & x_3 \ge x_2 + 0 & \text{(Arc (2, 3) constraint)} \\
& & x_6 - x_1 \le 25 & \\
& & A, B, C, D, E, F \ge 0, x_j \text{urs} &
\end{array}$$

The first six constraints stipulate that the duration of each activity can be reduced by at most 5 days. As before, the next seven constraints ensure that event j cannot occur until after node i occurs and activity (i, j) is completed. For example, activity B (arc (1, 2)) now has a duration of $9 - B$. Thus, we need the constraint $x_2 \ge x_1 + (9 - B)$. The constraint $x_6 - x_1 \le 25$ ensures that the project is completed within the 25-day deadline. The objective function is the total cost incurred in reducing the duration of the activities. An optimal solution to this LP is $z = \$390$, $x_1 = 0$, $x_2 = 4$, $x_3 = 4$, $x_4 = 6$, $x_5 = 13$, $x_6 = 25$, $A = 2$, $B = 5$, $C = 0$, $D = 5$, $E = 3$, $F = 0$. After reducing the durations of projects B, A, D, and E by the given amounts, we obtain the project network pictured in Figure 35. The reader should verify that A, B, D, E, and F are critical activities and that 1–2–3–4–5–6 and 1–3–4–5–6 are both critical paths (each having length 25). Thus, the project deadline of 25 days can be met for a cost of \$390.

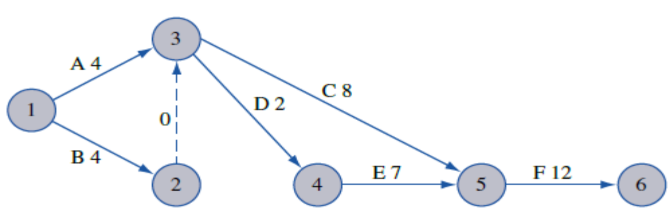


FIGURE 35
Duration of Activities after Crashing

Matlab code

```
c = [0 0 0 0 0 0 10 20 3 30 40 50];
A = [1 -1 0 0 0 0 0 -1 0 0 0 0; 1 0 -1 0 0 0 -1 0 0 0 0 0;
     0 0 1 0 -1 0 0 0 -1 0 0 0; 0 0 1 -1 0 0 0 0 0 -1 0 0;
     0 0 0 1 -1 0 0 0 0 0 -1 0; 0 0 0 0 1 -1 0 0 0 0 0 -1;
     0 1 -1 0 0 0 0 0 0 0 0 0; -1 0 0 0 0 1 0 0 0 0 0 0];
b = [-9 -6 -8 -7 -10 -12 0 25];
AA = [1 0 0 0 0 0 0 0 0 0 0 0];
bb = [0];
LB = [-20 -20 -20 -20 -20 -20 0 0 0 0 0 0];
UB = [100 100 100 100 100 100 5 5 5 5 5 5];
[x,fval] = linprog(c,A,b,AA,bb,LB,UB)
Optimal solution found.
   x^T = [0  4  4  6  13  25  2  5  0  5  3  0]
   fval = 390
```

PERT: Program Evaluation and Review Technique

- CPM assumes that the duration of each activity is known with certainty.
- For many projects, this is clearly not applicable.
- PERT is an attempt to correct this shortcoming of CPM by modeling the duration of each activity as a random variable.
- For each activity, PERT requires that the estimate the following three quantities:

  $a =$ estimate of the activitys duration under the most favorable conditions

  $b =$ estimate of the activitys duration under the least favorable conditions

  $m =$ most likely value for the activitys duration

- Let $\mathbf{T}_{ij}$ (random variables are printed in boldface) be the duration of activity (i, j).
- PERT requires the assumption that $\mathbf{T}_{ij}$ follows a beta distribution.
- The specific definition of a beta distribution need not concern us, but it is important to realize that it can approximate a wide range of random variables, including many positively skewed, negatively skewed, and symmetric random variables.
- If $\mathbf{T}_{ij}$ follows a beta distribution, then it can be shown that the mean and variance of $\mathbf{T}_{ij}$ may be approximated by

$$E(\mathbf{T}_{ij}) = \frac{a + 4m + b}{6} \qquad \text{and} \qquad var(\mathbf{T}_{ij}) = \frac{(b-a)^2}{36}.$$

- PERT requires the assumption that the durations of all activities are independent.
- Then for any path in the project network, the mean and variance of the time required to complete the activities on any path P are given by

$$\sum_{(i,j)\in P} E(\mathbf{T}_{ij}) \qquad \text{and} \qquad \sum_{(i,j)\in P} var(\mathbf{T}_{ij}).$$

- Let CP be the random variable denoting the total duration of the activities on a critical path found by CPM.
- PERT assumes that the critical path $\hat{P}$ found by CPM contains enough activities to allow us to invoke the Central Limit Theorem and conclude that

$$CP = \sum_{(i,j)\in \hat{P}} \mathbf{T}_{ij}$$

  is normally distributed.

- Then one can answer questions concerning the probability that the project will be completed by a given date.

Example Consider

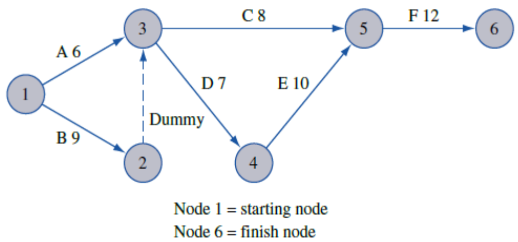


TABLE 15
a, *b*, and *m* for Activities in Widgeto

Activity	*a*	*b*	*m*
(1, 2)	5	13	9
(1, 3)	2	10	6
(3, 5)	3	13	8
(3, 4)	1	13	7
(4, 5)	8	12	10
(5, 6)	9	15	12

Then we have

$$E(\mathbf{T}_{12}) = \frac{\{5 + 13 + 36\}}{6} = 9 \qquad \mathrm{var}\mathbf{T}_{12} = \frac{(13-5)^2}{36} = 1.78$$
$$E(\mathbf{T}_{13}) = \frac{\{2 + 10 + 24\}}{6} = 6 \qquad \mathrm{var}\mathbf{T}_{13} = \frac{(10-2)^2}{36} = 1.78$$
$$E(\mathbf{T}_{35}) = \frac{\{3 + 13 + 32\}}{6} = 8 \qquad \mathrm{var}\mathbf{T}_{35} = \frac{(13-3)^2}{36} = 2.78$$
$$E(\mathbf{T}_{34}) = \frac{\{1 + 13 + 28\}}{6} = 7 \qquad \mathrm{var}\mathbf{T}_{34} = \frac{(13-1)^2}{36} = 4$$
$$E(\mathbf{T}_{45}) = \frac{\{8 + 12 + 40\}}{6} = 10 \qquad \mathrm{var}\mathbf{T}_{45} = \frac{(12-8)^2}{36} = 0.44$$
$$E(\mathbf{T}_{56}) = \frac{\{9 + 15 + 48\}}{6} = 12 \qquad \mathrm{var}\mathbf{T}_{56} = \frac{(15-9)^2}{36} = 1$$

Of course, the fact that arc (2, 3) is a dummy arc yields

$$E(\mathbf{T}_{23}) = \mathrm{var}\ \mathbf{T}_{23} = 0$$

Recall that the critical path for Example 6 was 1–2–3–4–5–6. From Equations (6) and (7),

$$E(\mathbf{CP}) = 9 + 0 + 7 + 10 + 12 = 38$$
$$\mathrm{var}\mathbf{CP} = 1.78 + 0 + 4 + 0.44 + 1 = 7.22$$

Then the standard deviation for **CP** is $(7.22)^{1/2} = 2.69$.

Question What is the probability that the project will be completed within 35 days?

Answer Assume that 1-2-3-4-5-6 is always the critical path. Then project will be completed within 35 days is just Prob($CP \leq 35$), and assume that CP is normally distributed. Then by the transformation $Z = (CP - 38)/2.69$ so that Z is a standardized normalized distributed random variable with mean 0 and standard deviation 1, we have

$$\mathrm{Prob}(CP \leq 35) = \mathrm{Prob}((CP - 38)/2.69 \leq (35 - 38)/2.69) = \mathrm{Prob}(Z \leq -1.12) = 0.13$$

by the standard normal distribution table.

Difficulties with PERT

1. The assumption that the activity durations are independent is difficult to justify.
2. Activity durations may not follow a beta distribution.
3. The assumption that the critical path found by CPM will always be the critical path for the project may not be justified.

- The last difficulty is the most serious.
- For instance, in our example, we assumed that 1-2-3-4-5-6 would always be the critical path.
- If, however, activity A were significantly delayed and activity B were completed ahead of schedule, then the critical path might be 1-3-4-5-6.
- One needs probability, simulation techniques to deal with the problem.

Minimum Spanning Tree Problems

The following method (MST algorithm) may be used to find a minimum spanning tree for a network:

Step 1 Begin at any node i, and join node i to the node in the network (node j) that is closest to node i. The two nodes i and j now form a connected set of nodes $C = \{i, j\}$ and arc (i, j) will be in the minimum spanning tree. The remaining nodes in the network (C') are the unconnected set of nodes.

Step 2 Choose a member of $C'(n)$ that is closest to some node in C. Let m represent the node in C that is closest to n. Then the arc (m, n) will be in the minimum spanning tree. Update C and C'. Because n is now connected to $\{i, j\}$, C now equals $\{i, j, n\}$, and we must eliminate node n from C'.

Step 3 Repeat this process until a minimum spanning tree is found. Ties for closest node and arc may be broken arbitrarily.

Remark

We assume that the network is connected and has undirected arcs (edges) with costs.

Example

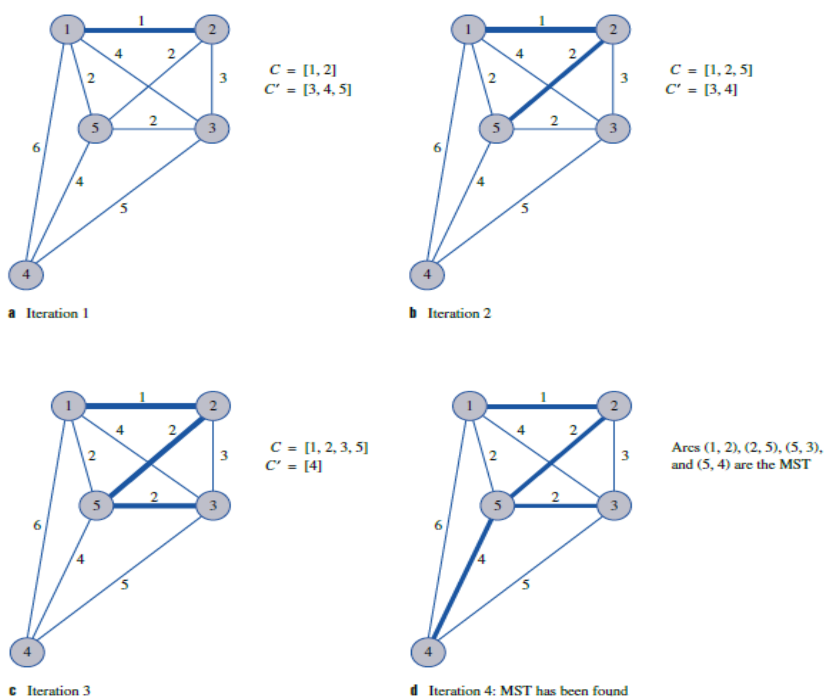


FIGURE 49
MST Algorithm for Computer Example

Minimum-Cost Network Flow Problems

The transportation, assignment, transshipment, shortest-path, maximum-flow, and critical path problems are all special cases of the minimum-cost network flow problem (MCNFP).

x_{ij} = number of units of flow sent from node i to node j through arc (i, j)

b_i = net supply (outflow − inflow) at node i

c_{ij} = cost of transporting one unit of flow from node i to node j via arc (i, j)

L_{ij} = lower bound on flow through arc (i, j) (if there is no lower bound, let $L_{ij} = 0$)

U_{ij} = upper bound on flow through arc (i, j) (if there is no upper bound, let $U_{ij} = \infty$)

Then an MCNFP may be written as

$$\min \sum_{\text{all arcs}} c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_j x_{ij} - \sum_k x_{ki} = b_i \qquad \text{(for each node } i \text{ in the network)}$$

$$L_{ij} \le x_{ij} \le U_{ij} \qquad \text{(for each arc in the network)}$$

The first set of constraints are the **flow balance equations,** and the second set of constraints express limitations on arc capacities.

Transportation problems as MCNF problems

TABLE 28

1	2	4 (Node 1)
3	4	5 (Node 2)
6 (Node 3)	3 (Node 4)	

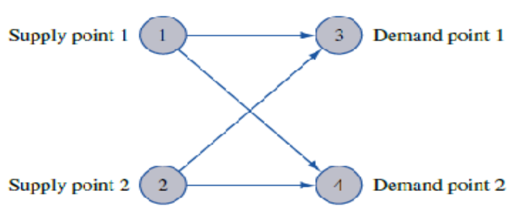


FIGURE 45 Representation of Transportation Problem as an MCNFP

TABLE 29 MCNFP Representation of Transportation Problem

$\min z = x_{13} + 2x_{14} + 3x_{23} + 4x_{24}$						
x_{13}	x_{14}	x_{23}	x_{24}		rhs	Constraint
1	1	0	0	=	4	Node 1
0	0	1	1	=	5	Node 2
−1	0	−1	0	=	−6	Node 3
1	−1	0	−1	=	−3	Node 4
All variables non-negative						

Maximum flow problems as MCNF problems

TABLE 30
MCNFP Representation of Maximum-Flow Problem

				max $z = x_0$					
$x_{so,1}$	$x_{so,2}$	x_{13}	x_{12}	$x_{3,si}$	$x_{2,si}$	x_0		rhs	Constraint
1	1	0	0	0	0	−1	=	0	Node *so*
−1	0	1	1	0	0	0	=	0	Node 1
0	−1	0	−1	0	1	0	=	0	Node 2
0	0	−1	0	1	0	0	=	0	Node 3
0	0	0	0	−1	−1	1	=	0	Node *si*
1	0	0	0	0	0	0	≤	2	Arc (*so*, 1)
0	1	0	0	0	0	0	≤	3	Arc (*so*, 2)
0	0	1	0	0	0	0	≤	4	Arc (1, 3)
0	0	0	1	0	0	0	≤	3	Arc (1, 2)
0	0	0	0	1	0	0	≤	1	Arc (3, *si*)
0	0	0	0	0	1	0	≤	2	Arc (2, *si*)
				All variables nonnegative					

Remarks

- One can formulate other network problems as MCNF problems; see Example 7 in Chapter 8.
- One can apply a change of variables $x_{ij} - L_{ij}$ and assume that $L_{ij} = 0$. Note that b_i will be changed to

$$\tilde{b}_i = b_i - \sum_j L_{ij} + \sum_k L_{ki} \geq \sum_j (x_{ij} - L_{ij}) - \sum_k (x_{ki} - L_{ki}).$$

- One can use linprog(c,A,b,AA,bb,LB,UB) command in Matlab to solve the network problem by setting up

```
the cost vector c = [c_{ij}],  A = [ ], b = [ ],
AA x = bb, the network constraints,  LB, UB.
```

- Note that we can delete one of network constraints $\tilde{A}x = \tilde{b}$ to get $AAx = bb$ because each column of $\tilde{A}$ has a "1" and a "−1".

  So, the sum of rows of $\tilde{A} = [0, \ldots, 0]$, i.e., the rows are linearly dependent.

Network Simplex Method

Step 1 Determine a starting bfs. The $n - 1$ basic variables will correspond to a spanning tree. Indicate nonbasic variables at their upper bound by dashed arcs.

Step 2 Compute $y_1, y_2, \ldots y_n$ (often called the *simplex multipliers*) by solving $y_1 = 0$, $y_i - y_j = c_{ij}$ for all basic variables x_{ij}. For all nonbasic variables, determine the row 0 coefficient $\bar{c}_{ij}$ from $\bar{c}_{ij} = y_i - y_j - c_{ij}$. The current bfs is optimal if $\bar{c}_{ij} \leq 0$ for all $x_{ij} = L_{ij}$ and $\bar{c}_{ij} \geq 0$ for all $x_{ij} = U_{ij}$. If the bfs is not optimal, then choose the nonbasic variable that most violates the optimality conditions as the entering basic variable.

Step 3 Identify the cycle (there will be exactly one!) created by adding the arc corresponding to the entering variable to the current spanning tree of the current bfs. Use conservation of flow to determine the new values of the variables in the cycle. The variable that first hits its upper or lower bound as the value of the entering basic variable is changed exits the basis.

Step 4 Find the new bfs by changing the flows of the arcs in the cycle found in step 3. Go to step 2.

EXAMPLE 9 Network Simplex Solution to MCNFP

Use the network simplex to solve the MCNFP in Figure 56.

Solution A bfs requires that we find a spanning tree (three arcs that connect nodes 1, 2, 3, and 4 and do not form a cycle). Any arcs not in the spanning tree may be set equal to their upper or lower bound. By trial and error, we find the bfs in Figure 57 involving the spanning tree (1, 2), (1, 3), and (2, 4).

To find y_1, y_2, y_3, and y_4 we solve

$$y_1 = 0, \qquad y_1 - y_2 = 4, \qquad y_2 - y_4 = 3, \qquad y_1 - y_3 = 3$$

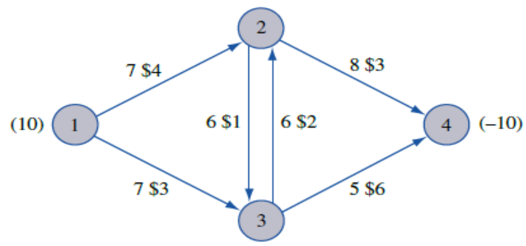


FIGURE 56 Example of Network Simplex

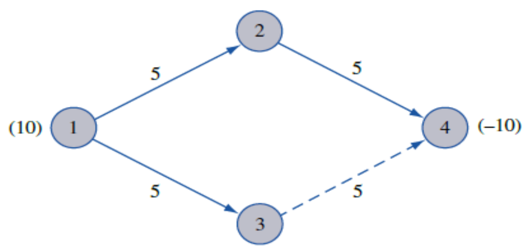


FIGURE 57 bfs for Example 9

This yields $y_1 = 0$, $y_2 = -4$, $y_3 = -3$, and $y_4 = -7$. The row 0 coefficients for each nonbasic variable are

$$\bar{c}_{34} = -3 - (-7) - 6 = -2 \quad \text{(Violates optimality condition)}$$
$$\bar{c}_{23} = -4 - (-3) - 1 = -2 \quad \text{(Satisfies optimality condition)}$$
$$\bar{c}_{32} = -3 - (-4) - 2 = -1 \quad \text{(Satisfies optimality condition)}$$

Thus, x_{34} enters the basis. We set $x_{34} = 5 - \theta$ and obtain the cycle in Figure 58. From arc (1, 2), we find $5 + \theta \leq 7$ or $\theta \leq 2$. From arc (1, 3), we find $5 - \theta \geq 0$ or $\theta \leq 5$. From arc (2, 4), we find $5+ \theta \leq 8$ or $\theta \leq 3$. From arc (3, 4), we find $5 - \theta \geq 0$ or $\theta \leq 5$. Thus, we can set $\theta = 2$. Now x_{12} exits the basis at its upper bound, and x_{34} enters, yielding the bfs in Figure 59.

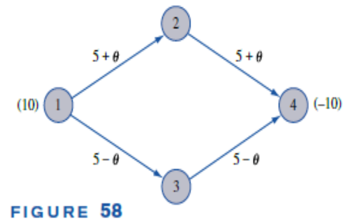


FIGURE 58
Cycle Created When x_{34} Enters the Basis

The new bfs is associated with the spanning tree (1, 3), (2, 4), and (3, 4). Solving for the new values of the simplex multipliers, we obtain

$$y_1 = 0, \quad y_1 - y_3 = 3, \quad y_3 - y_4 = 6, \quad y_2 - y_4 = 3$$

This yields $y_1 = 0, y_2 = -6, y_3 = -3, y_4 = -9$. The coefficient of each nonbasic variable in row 0 is given by

$\bar{c}_{12} = 0 - (-6) - 4 = 2$ (Satisfies optimality condition)

$\bar{c}_{23} = -6 - (-3) - 1 = -4$ (Satisfies optimality condition)

$\bar{c}_{32} = -3 - (-6) - 2 = 1$ (Violates optimality condition)

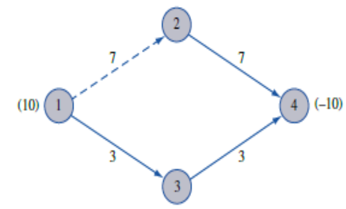


FIGURE 59
bfs After x_{12} Exits and x_{34} Enters

Now x_{32} enters the basis, yielding the cycle in Figure 60. From arc (2, 4), we find $7 + \theta \leq 8$ or $\theta \leq 1$); from arc (3, 4), we find $3 - \theta \geq 0$ or $\theta \leq 3$. From arc (3, 2), we find $\theta \leq 6$. So we now set $\theta = 1$ and have x_{24} exit from the basis at its upper bound. The new bfs is given in Figure 61.

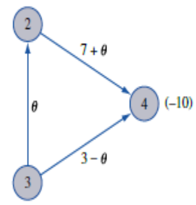


FIGURE 60
Cycle Created When x_{32} Enters Basis

The current set of basic values corresponds to the spanning tree (1, 3), (3, 2), and (3, 4). The new values of the simplex multipliers are found by solving

$$y_1 = 0, \quad y_1 - y_3 = 3, \quad y_3 - y_2 = 2, \quad y_3 - y_4 = 6$$

which yields $y_1 = 0, y_2 = -5, y_3 = -3, y_4 = -9$. The coefficient of each nonbasic variable in row 0 is now

$\bar{c}_{23} = -5 - (-3) - 1 = -3$ (Satisfies optimality condition)

$\bar{c}_{12} = 0 - (-5) - 4 = 1$ (Satisfies optimality condition)

$\bar{c}_{24} = -5 - (-9) - 3 = 1$ (Satisfies optimality condition)

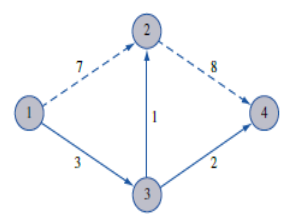


FIGURE 61
New bfs When x_{32} Enters and x_{24} Exits

Thus, the current bfs is optimal. The optimal solution to the MCNFP is

Basic variables: $x_{13} = 3, \quad x_{32} = 1, \quad x_{34} = 2$

Nonbasic variables at their upper bound: $x_{12} = 7, \quad x_{24} = 8$

Nonbasic variable at lower bound: $x_{23} = 0$

The optimal z-value is obtained from

$$z = 7(4) + 3(3) + 1(2) + 8(3) + 2(6) = \$75$$