Homework 11 Guidelines

#1 Network Primal Simplex

(1) Start from a spanning tree T

(2) Compute (primal) flow x_{ij} ($x_{ij} = 0$ if $(i,j) \notin T$)

Assume $x_{ij} \geq 0$ i.e. a feasible solution

(3) Compute dual (y_i) and dual slack variables (z_{ij})

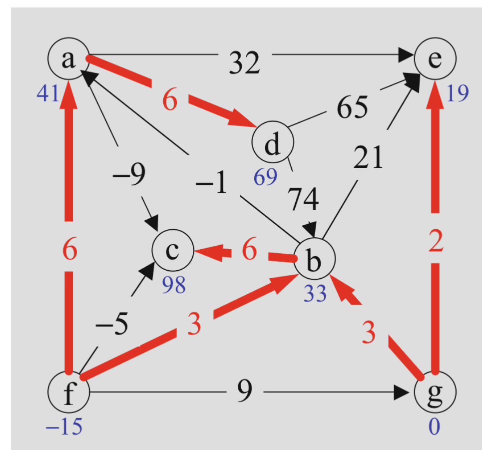
$\forall (i,j) \in T$, $y_j - y_i + \underset{0}{\cancel{z_{ij}}} = c_{ij}$ ($z_{ij} = 0$)

Complementary slackness $x_{ij} z_{ij} = 0$

$\forall (i,j) \notin T$, $z_{ij} = c_{ij} - (y_j - y_i)$

(4) Choose (i,j) $z_{ij} < 0$ (if none, opt.)

(5) Update tree



#2 Network Dual Simplex

(1) Start from a spanning tree T

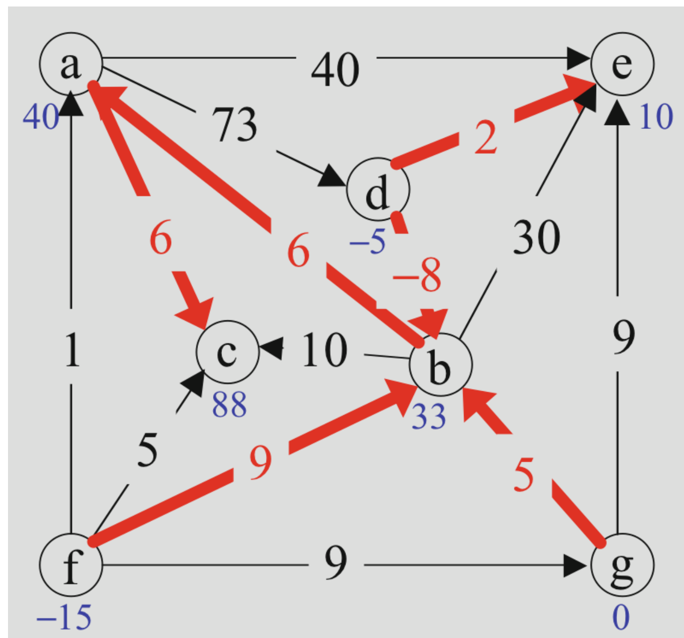
(2) Compute (primal) flow x_{ij} ($x_{ij} = 0$ if $(i,j) \notin T$)

dual variables, y_j, z_{ij}

Assume $z_{ij} \geq 0$ i.e. a feasible dual solution

(3) Choose $x_{ij} < 0$ (if none, opt.)

(4) Update tree



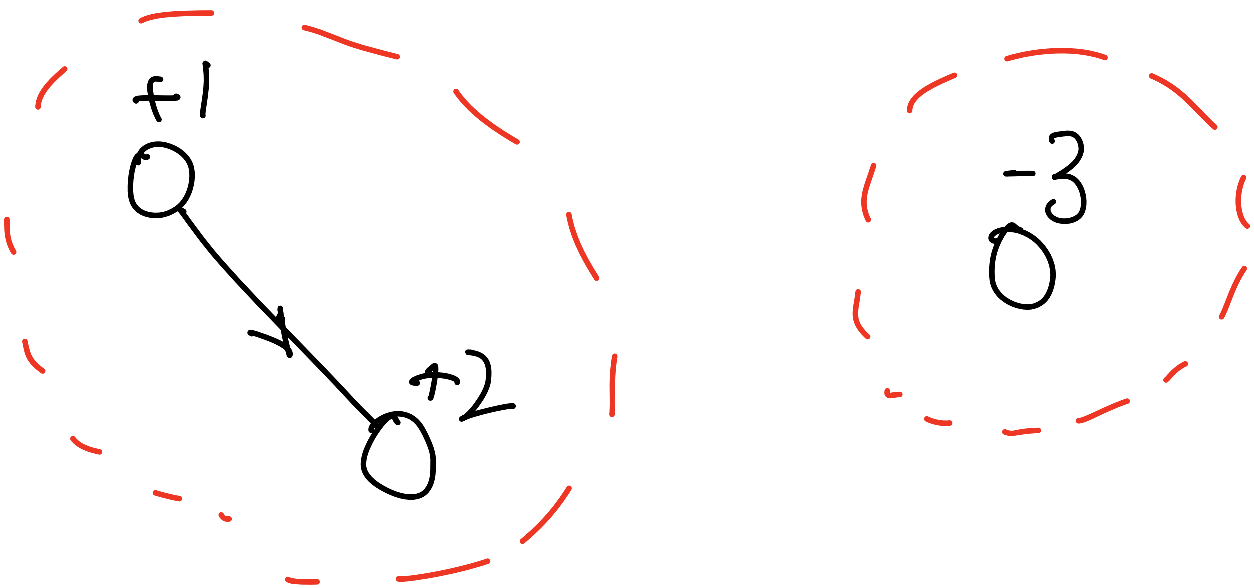
#3 (Ex 14.12) If $\tilde{B}^{-1}$ exists, then arcs represented by B forms a spanning tree.

$\Updownarrow$

i.e. $\tilde{B}X = Y$ is uniquely solvable for any Y

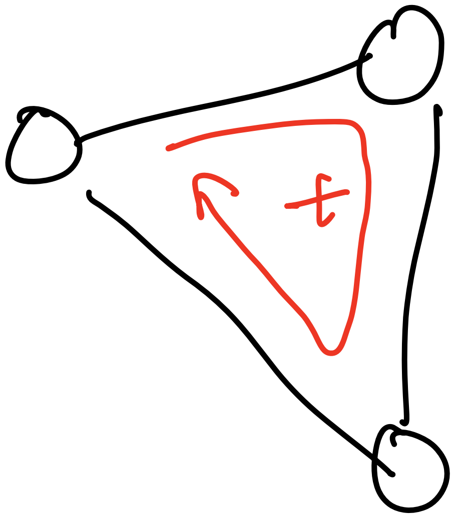
Proof (by contradiction)

(1) what if the nodes are disconnected?



No Solution

(2) What if the arcs form a loop?

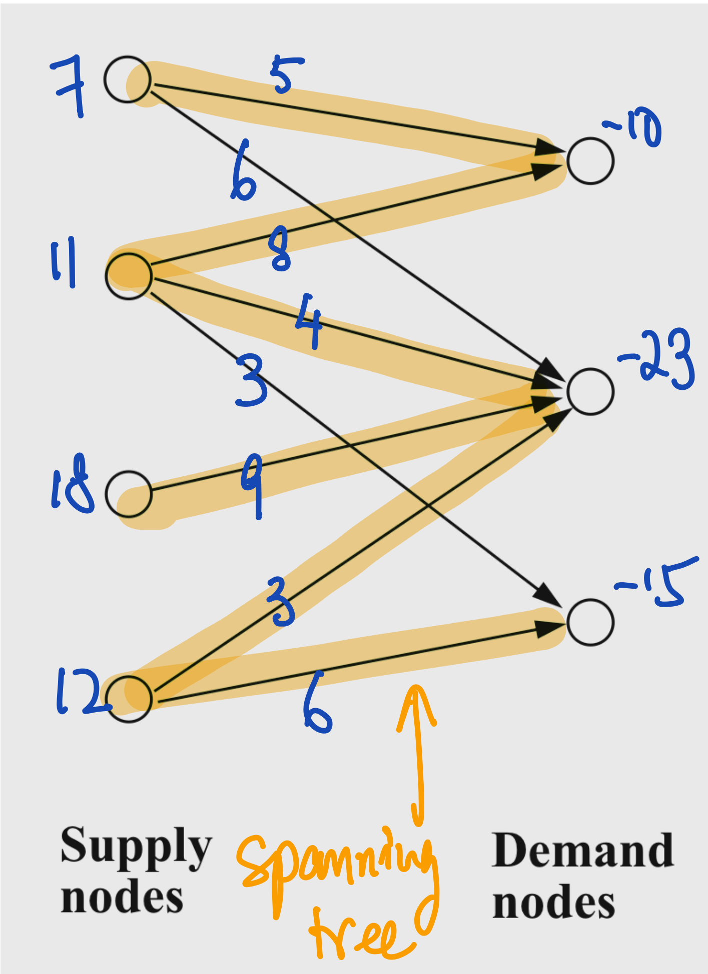


Infinitely many solutions

Hence the graph must be connected, with no loops

⇒ it must be a spanning tree

#4 15.1



s_i \ d_j	−10	−23	−15
7	5	6	*
11	8	4	3
18	*	9	*
12	*	3	6

c_{ij}

y_i \ y_j	5	1	4
0	7	5	*
−3	3	8	−4
−8	*	18	*
−2	*	−3	15

x_{ij}

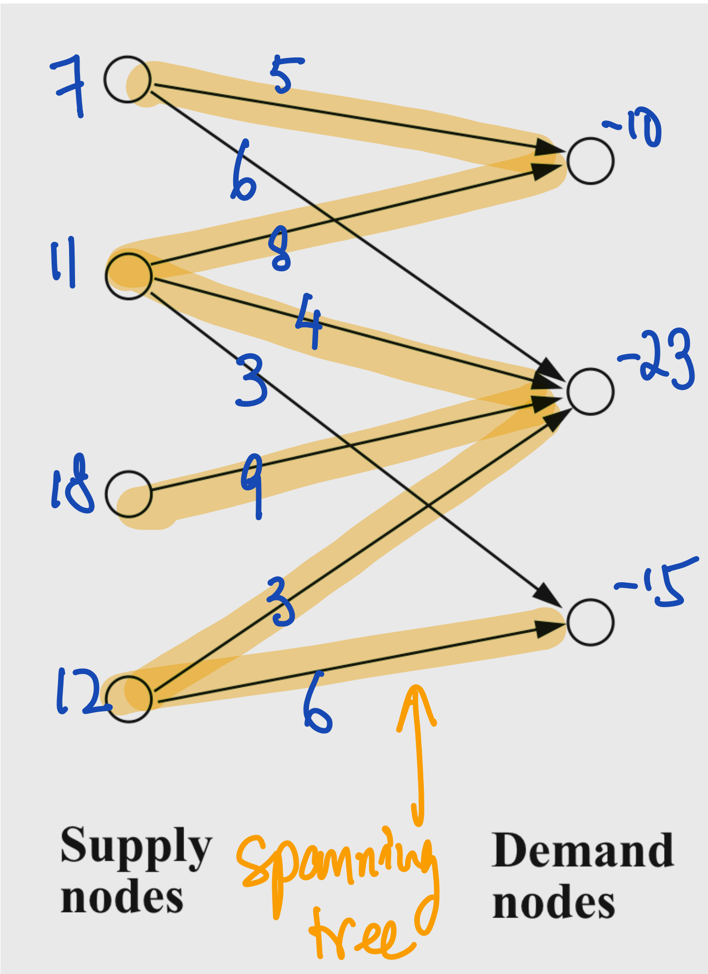
$y_j - y_i = c_{ij}$

$5 - 0 = 5$

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

$1 - 0 + 5 = 6$

#4 15.1



s_i \ d_j	−10	−23	−15
7	5	6	*
11	8	4	3
18	*	9	*
12	*	3	6

c_{ij}

$y_j - y_i = c_{ij}$

$5 - 0 = 5$

y_i \ y_j	5	1	4
0	7	5	*
−3	3	8	−4
−8	*	18	*
−2	*	−3	15

z_{ij}

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

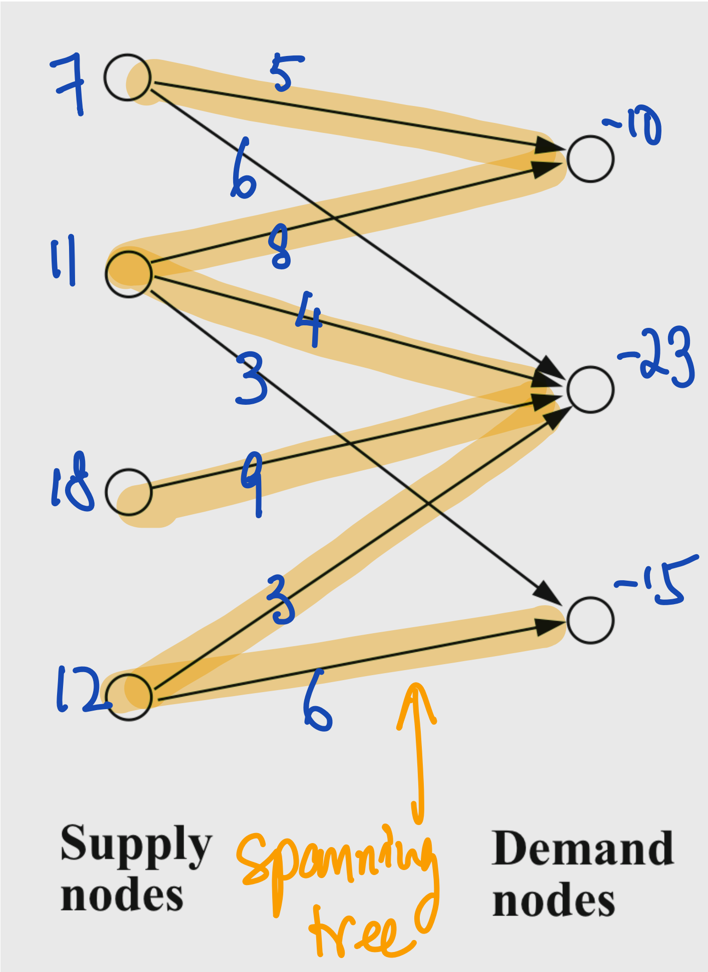
$1 - 0 + 5 = 6$

x_{ij}

dual infeasible

primal feasible

#4 15.1



y_i \ y_j	5	1	4
0	7	5	*
−3	3	8	−4
−8	*	18	*
−2	*	−3	15

$(i=2,\ j=3)$

(1) $y_j - y_i + z_{ij} = c_{ij}$

$4 - (-3) + (-4) = 3$

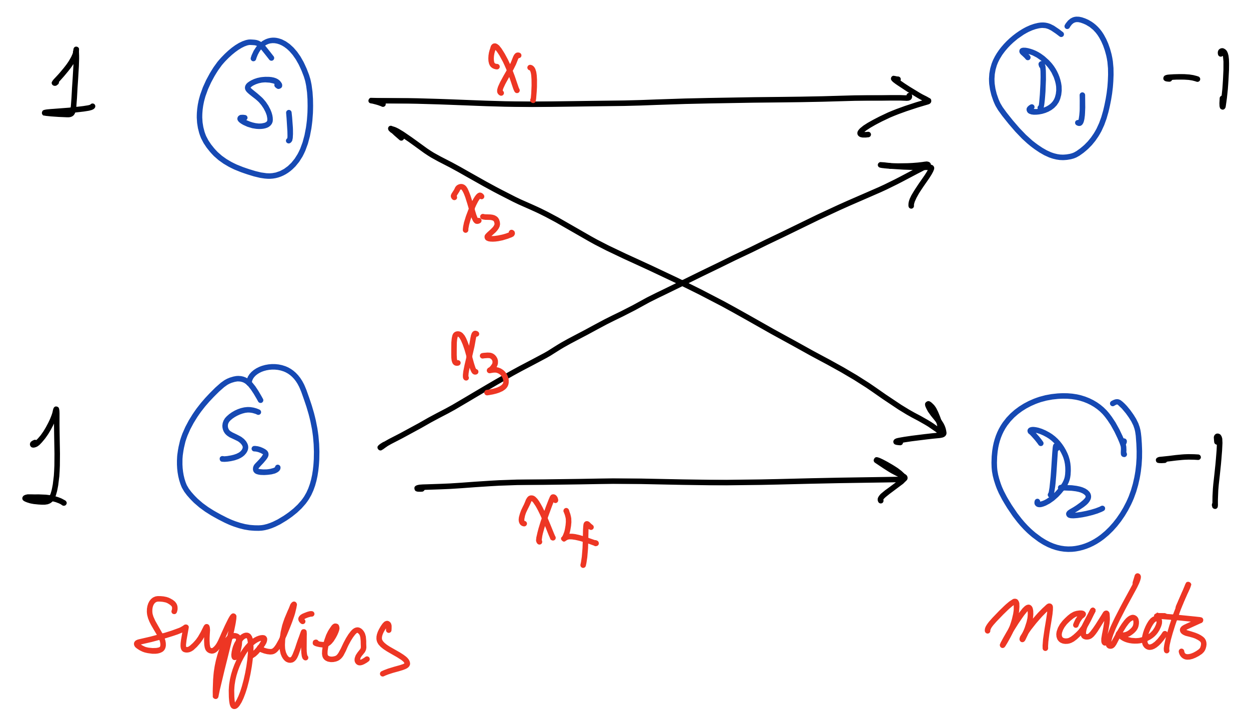
(2) Change c_{23} to 8. Then dual feasible $\rightarrow z_{ij} = 1 > 0$

(Phase I)

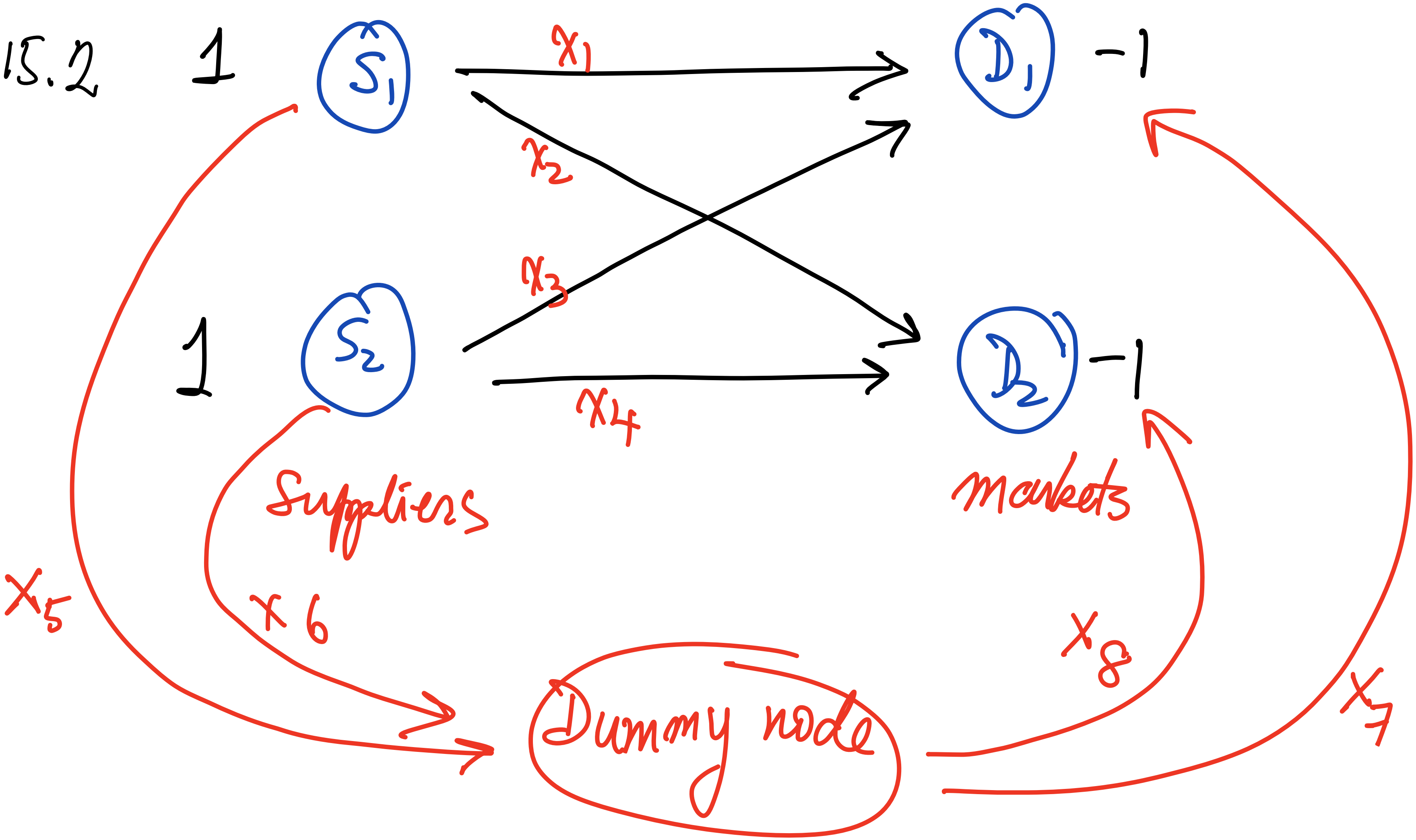
(Then perform phase II)

#4 15.2

$$
\begin{array}{lrl}
\text{maximize} & 7x_1 - 3x_2 + 9x_3 + 2x_4 & \\
\text{subject to} & x_1 + x_2 \le 1 & \leftarrow S_1 \\
& x_3 + x_4 \le 1 & \leftarrow S_2 \\
& x_1 + x_3 \ge 1 & \leftarrow D_1 \\
& x_2 + x_4 \ge 1 & \leftarrow D_2 \\
& x_1,\ x_2,\ x_3,\ x_4 \ge 0. &
\end{array}
$$



#4 15.2



$$x_1 + x_2 + x_5 = 1 \quad \leftarrow S_1$$

$$x_3 + x_4 + x_6 = 1 \quad \leftarrow S_2$$

$$x_1 + x_3 + x_7 = 1 \quad \leftarrow D_1$$

$$x_2 + x_4 + x_8 = 1 \quad \leftarrow D_2$$

$$x_5 + x_6 = x_7 + x_8 \quad \leftarrow \text{Dummy node}$$

#4 15.2

$$\begin{array}{lrcrcrcrcl} \text{maximize} & 7x_1 & - & 3x_2 & + & 9x_3 & + & 2x_4 & & \\ \text{subject to} & x_1 & + & x_2 & & & & & \leq & 1 \\ & & & & & x_3 & + & x_4 & \leq & 1 \\ & x_1 & & & + & x_3 & & & \geq & 1 \\ & & & x_2 & & & + & x_4 & \geq & 1 \\ & & & & & x_1, & x_2, & x_3, \; x_4 & \geq & 0. \end{array}$$

$$\begin{array}{rrrrcl} x_1 & + \; x_2 & & & \leq & 1 \\ & & x_3 & + \; x_4 & \leq & 1 \\ -x_1 & & -x_3 & & \leq & -1 \\ & -x_2 & & -x_4 & \leq & -1 \\ \hline & & & 0 & \leq & 0 \end{array}$$

+ +

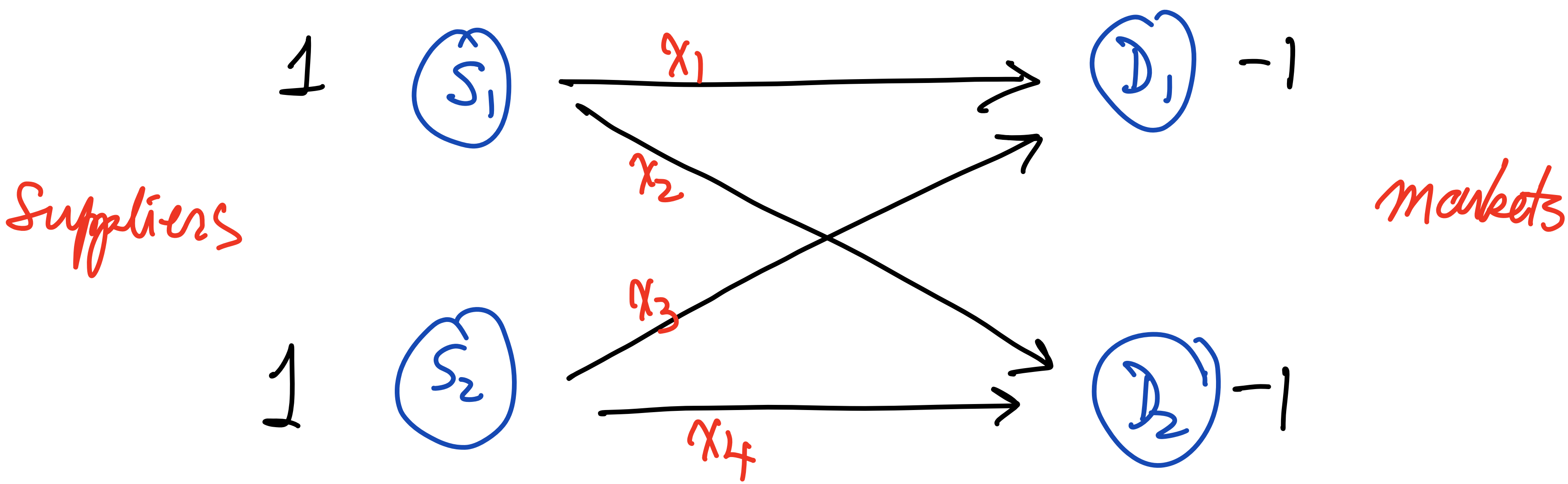
#4 15.2

$$\begin{aligned} \text{maximize} \quad & 7x_1 - 3x_2 + 9x_3 + 2x_4 \\ \text{subject to} \quad & x_1 + x_2 \leq 1 \\ & x_3 + x_4 \leq 1 \\ & x_1 + x_3 \geq 1 \\ & x_2 + x_4 \geq 1 \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

$$\begin{aligned} x_1 + x_2 &\leq 1 \\ x_3 + x_4 &\leq 1 \\ -x_1 - x_3 &\leq -1 \\ -x_2 - x_4 &\leq -1 \\ \hline 0 &\leq 0 \end{aligned}$$

#4 15.2

$$
\begin{array}{lrcl}
\text{maximize} & 7x_1 - 3x_2 + 9x_3 + 2x_4 & & \\
\text{subject to} & x_1 + x_2 & \leq & 1 \\
& x_3 + x_4 & \leq & 1 \\
& x_1 + x_3 & \geq & 1 \\
& x_2 + x_4 & \geq & 1 \\
& x_1, x_2, x_3, x_4 & \geq & 0.
\end{array}
$$



By Integrality Theorem x_1, x_2, x_3, x_4 are integers.

$x_1 = 1,\ x_4 = 1 \Rightarrow \zeta = 9$; $\quad x_2 = 1,\ x_3 = 1,\ \zeta = 7$

#15.4

Job	Duration (weeks)	Must be preceded by
0. Sign contract with buyer	0	–
1. Framing	2	0
2. Roofing	1	1
3. Siding	3	1
4. Windows	2.5	3
5. Plumbing	1.5	3
6. Electrical	2	2,4
7. Inside finishing	4	5,6
8. Outside painting	3	2,4
9. Complete the sale to buyer	0	7,8

(a),(b)

$\min t_9$

s.t.

$t_2 - t_1 \geqslant 2$

$t_3 - t_1 \geqslant 2$

$t_4 - t_3 \geqslant 3$

$t_5 - t_3 \geqslant 3$

$t_6 - t_2 \geqslant 1$

$t_6 - t_4 \geqslant 2.5$

$t_7 - t_5 \geqslant 1.5$

$t_7 - t_6 \geqslant 2$

$t_8 - t_2 \geqslant 1$

$t_8 - t_4 \geqslant 2.5$

$t_9 - t_7 \geqslant 4$

$t_9 - t_8 \geqslant 3$

15.6 (a)

(P)

$$\min \; t_9$$

x_{12}	$t_2 - t_1 \geq 2$
x_{13}	$t_3 - t_1 \geq 2$
x_{34}	$t_4 - t_3 \geq 3$
x_{35}	$t_5 - t_3 \geq 3$
x_{26}	$t_6 - t_2 \geq 1$
x_{46}	$t_6 - t_4 \geq 2.5$
x_{57}	$t_7 - t_5 \geq 1.5$
→ x_{67}	$t_7 - t_6 \geq 2$
x_{28}	$t_8 - t_2 \geq 1$
x_{48}	$t_8 - t_4 \geq 2.5$
x_{79}	$t_9 - t_7 \geq 4$
x_{89}	$t_9 - t_8 \geq 3$

$t_9 \geq$

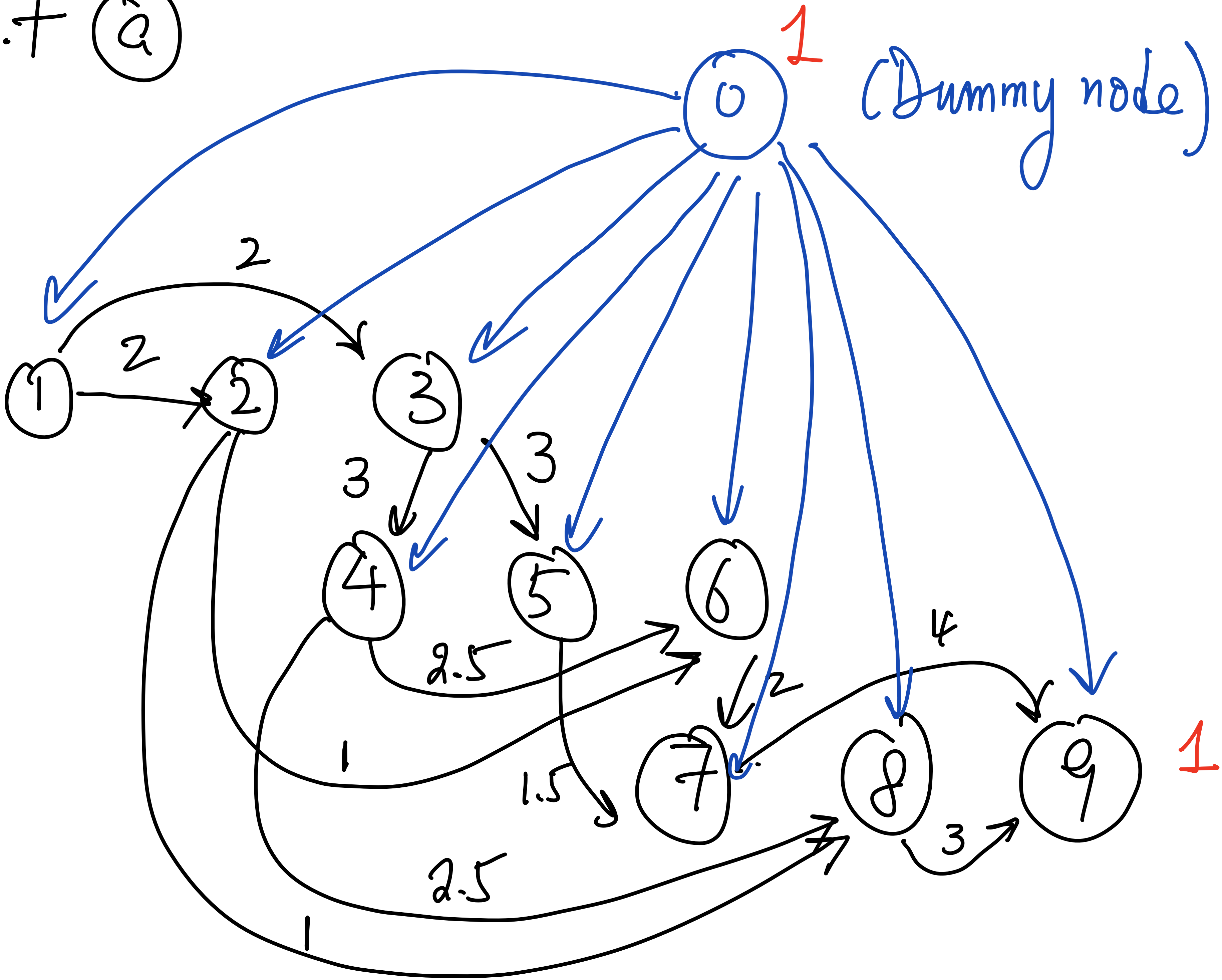
(D)

$$\max \; 2x_{12} + 2x_{13} + \cdots + 4x_{79} + 3x_{89}$$

s.t.

$$-x_{12} - x_{13} \leq 0$$
$$x_{12} - x_{26} - x_{28} \leq 0$$
$$x_{13} - x_{34} - x_{35} \leq 0$$
$$x_{34} - x_{46} - x_{48} \leq 0$$
$$x_{35} - x_{57} \leq 0$$
$$x_{26} + x_{46} - x_{67} \leq 0$$
$$x_{57} + x_{67} - x_{79} \leq 0$$
$$x_{28} + x_{48} - x_{89} \leq 0$$
$$x_{79} + x_{89} \leq 1$$

15.7 (a)



(b) $x_{ij} z_{ij} = 0$

i.e. $x_{ij}(t_j - t_i - d_i) = 0$

Hence if $t_j > t_i + d_i$, then $x_{ij} = 0$

or if $x_{ij} = 1$, then $t_j = t_i + d_i$

15.8

$t_1 = 0$

$t_2 = t_1 + 2 = 2$

$t_3 = t_1 + 2 = 2$

$t_4 = t_3 + 3 = 5$

$t_5 = t_3 + 3 = 5$

$t_6 = \max\{t_2 + 1, t_4 + 2.5\} = \max\{3, 7.5\} = 7.5$

$t_7 = \max\{t_5 + 1.5, t_6 + 2\} = \max\{6.5, 9.5\} = 9.5$

$t_8 = \max\{t_2 + 1, t_4 + 2.5\} = \ldots\ldots = 7.5$

$t_9 = \max\{t_7 + 4, t_8 + 3\} = \max\{13.5, 10.5\} = \underline{13.5}$

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}	*Relations*	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)

[V] #15.4, 15-6, 15.7, 15-8 Bradley-Hax-Magnanti Applied Mathematical Programming

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

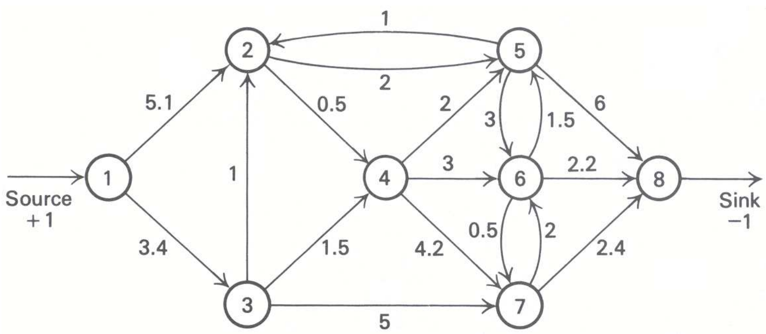


Figure 8.4 Network for a shortest-path problem.

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.

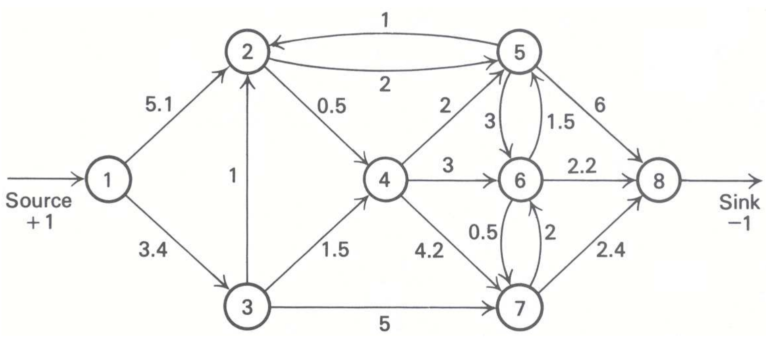


Figure 8.5 Task-oriented network.

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

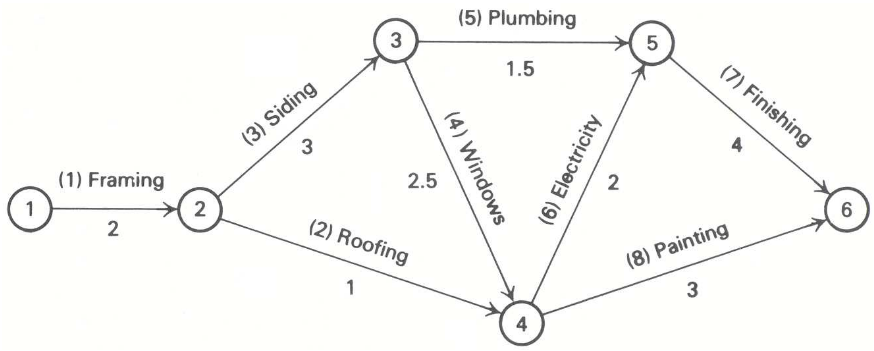


Figure 8.6 Event-oriented network.

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:

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

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

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

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

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