Chapter 10

Network Implementation

Draft date: 08-01-06

10.1 Shipping

Consider the following transshipment problem.

Problem 10.1.1 *Eumerica makes bottled air at three plants in Vienna, Athens, and Moscow, and ships crates of their products to distributors in Venice, Frankfurt, and Paris. Each day the Athens plant produces* 23 *thousand crates, while Vienna can produce up to* 15 *thousand, and Moscow can produce up to* 20 *thousand. In addition, Venice must receive* 17 *thousand and Paris must receive* 25 *thousand crates, while Frankfurt can receive up to* 27 *thousand. The company pays Arope Trucking to transport their products at the following per-crate Eurodollar costs.*

100	*Vienna to Frankfurt*	200	*Venice to Paris*
120	*Frankfurt to Athens*	210	*Paris to Venice*
130	*Athens to Frankfurt*	260	*Frankfurt to Venice*
140	*Frankfurt to Paris*	280	*Venice to Moscow*
150	*Paris to Athens*	290	*Moscow to Frankfurt*
170	*Athens to Vienna*		

Eumerica would like to tell Arope which shipments to make between cities so as to minimize cost.

It is helpful in this case to interpret the given information visually. In Figure 10.1, we've drawn the cities about where they are in the United States — each city in the diagram is called a **node**. Every direct trucking route from one city to another is drawn as an **arc** in the diagram. Each node is labeled by its **demand** in the problem (we will think of a **supply** as a negative demand), and each arc is labeled by the per item **cost** of shipping along its route. The entire structure of nodes, arcs, demands and costs is what we call a **network**. (Without the demand and cost labels, it is just a **directed graph** (or **digraph**), and further, without direction on the arcs, the structure is called a graph; its undirected arcs are called edges and its nodes are typically called vertices.)

node

arc

demand/supply

cost

network

directed graph

digraph

Our immediate aim here is to visualize the networks that correspond to the tableaux during each step of the Simplex algorithm. Then we will attempt to interpret the tableau pivot decision-making process in the network environment in the hopes of being able to ignore tableaux entirely. As in Chapter 5, where we implemented the Simplex algorithm in a matrix setting, the result here will not be a new algorithm, but instead merely the network implementation of Simplex. Note that each

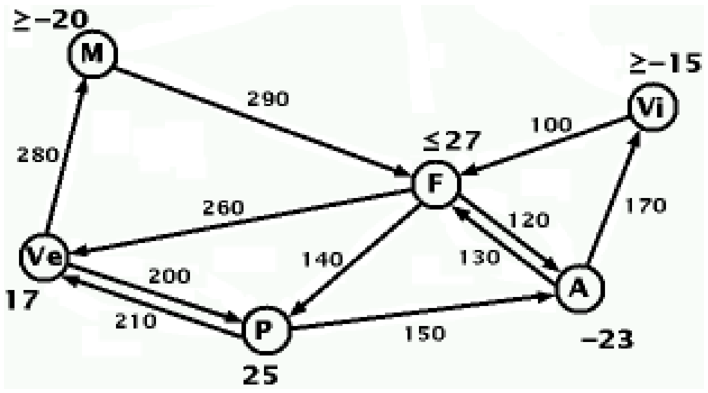

Figure 10.1: Network for Problem 10.1.1

node A corresponds to a constraint, while each arc AB (from the **tail** A to the **head** B) corresponds to the variable y_{AB} whose value equals the number of (thousands of) items shipped on the arc. We arrive at the following general LOP.

tail/head

Problem 10.1.2

$$\begin{array}{lrl}
\text{Min.} & w = & 130y_{AF} + 170y_{AVi} + 120y_{FA} + 140y_{FP} + 260y_{FVe} + 290y_{MF} \\
& & +150y_{PA} + 210y_{PVe} + 280y_{VeM} + 200y_{VeP} + 100y_{ViF} \\
& & \\
\text{s.t.} & & -y_{AF} - y_{AVi} + y_{FA} + y_{PA} = -23 \\
& & y_{AF} - y_{FA} - y_{FP} - y_{FVe} + y_{MF} + y_{ViF} \leq 27 \\
& & -y_{MF} + y_{VeM} \geq -20 \\
& & y_{FP} - y_{PA} - y_{PVe} + y_{VeP} = 25 \\
& & y_{FVe} + y_{PVe} - y_{VeM} - y_{VeP} = 17 \\
& & y_{AVi} - y_{ViF} \geq -15 \\
& & \\
\& & & y_{\text{all}} \geq 0
\end{array}$$

We purposely ordered the constraints (rows) and variables (columns) alphabetically so as to make least subscript decisions easily identifiable. We also avoid standard form here by adding or subtracting appropriate slack variables. The resulting initial tableau is as follows. (Note that the infinity (slack) arcs are also listed alphabetically, ignoring the infinity symbol, and that these are listed after all original (problem) arcs — keep this in mind for later.)

Tableau 10.1.3 .

$$\left[\begin{array}{ccccccccccc|cccc|c}
-1 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -23 \\
1 & 0 & -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 27 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & -20 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 25 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 17 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -15 \\
\hline
130 & 170 & 120 & 140 & 260 & 290 & 150 & 210 & 280 & 200 & 100 & 0 & 0 & 0 & 1 & 0
\end{array}\right]$$

There are several things worth noting about Tableau 10.1.3. First, every nonslack column has exactly one -1 and 1. Of course, this makes sense because every column represents an arc *from* one vertex *to* another. Second, this is the kind of sparse example alluded to in Section 5.2 — the density of nonzero entries is only $2/n$, where n is the number of nodes. Third, we can think of the slack columns as arcs as well by imagining an invisible **infinity node**, whose corresponding tableau row equals the negative of the sum of all other node rows. The resulting tableau is below.

infinity node

Tableau 10.1.4 .

$$\left[\begin{array}{ccccccccccc|cccc|c}
-1 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -23 \\
1 & 0 & -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 27 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & -20 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 25 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 17 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -15 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & -11 \\
\hline
130 & 170 & 120 & 140 & 260 & 290 & 150 & 210 & 280 & 200 & 100 & 0 & 0 & 0 & 1 & 0
\end{array}\right]$$

The corresponding minimization problem, having only equalities, nonnegative variables, and demand sum 0, is in **standard network form**.

standard network form

Problem 10.1.5

$$\begin{array}{lrl}
\text{Min.} & w = & 130y_{AF}+170y_{AVi}+120y_{FA}+140y_{FP}+260y_{FVe}+290y_{MF} \\
& & +150y_{PA}+210y_{PVe}+280y_{VeM}+200y_{VeP}+100y_{ViF} \\
\\
\text{s.t.} & & -y_{AF}-y_{AVi}+y_{FA}+y_{PA}=-23 \\
& & y_{AF}-y_{FA}-y_{FP}-y_{FVe}+y_{MF}+y_{ViF}+y_{\infty F}=27 \\
& & -y_{MF}+y_{VeM}-y_{M\infty}=-20 \\
& & y_{FP}-y_{PA}-y_{PVe}+y_{VeP}=25 \\
& & y_{FVe}+y_{PVe}-y_{VeM}-y_{VeP}=17 \\
& & y_{AVi}-y_{ViF}-y_{Vi\infty}=-15 \\
& & -y_{\infty F}+y_{M\infty}+y_{Vi\infty}=-11 \\
\\
\& & & y_{\text{all}}\geq 0
\end{array}$$

We say that Tableau 10.1.3 is the initial **reduced tableau** for Problem 10.1.5, while Tableau 10.1.4 is its initial **augmented tableau**. There may be several reduced tableaux for a given network problem — the point is that we must remove a (any) redundant row from a tableau since it needs to have full row rank (see Chapter 5). The initial **augmented network** is Figure 10.2. Here, our concern is to include the redundancy, as we shall soon see.

reduced/augmented tableau/network

Starting from the reduced Tableau 10.1.3, we need to fill the basis using Phase 0 (see Chapter 6). However, for networks, we will modify the Phase 0 rules by ignoring any current partial basis (some of whose coefficients are negative, by the way) and instead pivotting in least subscript variables greedily from scratch, without any other regard.

Workout 10.1.6 *Starting with Tableau 10.1.3, use modified Phase 0 to find the first basis.*

(a) *At each stage, starting with no arcs, draw each basic arc pivotted in.*

(b) *In the third step, arc FA cannot be chosen for the basis. Explain why not.*

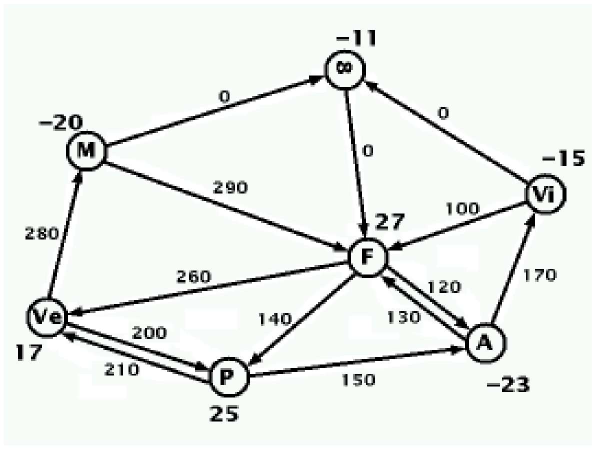

Figure 10.2: Augmented network for Problem 10.1.1

[i] *algebraically.*

[ii] *visually.*

(c) *In the sixth step, arc PA cannot be chosen for the basis. Explain why not*

[i] *algebraically.*

[ii] *visually.*

One thing we will learn that the network implementation has in common with the matrix implementation is that they share their dependence on initial information rather than current information. In both cases one can specify any basis possible and work from there.

Workout 10.1.7 *Let $\beta = \{AVi, FP, MF, PA, VeM, M\infty\}$ in Problem 10.1.2.*

(a) *Draw the network corresponding to β.*

(b) *Use Figure 10.1 to label the arcs of your result in part a by the shipments needed to satisfy all demands. [HINT: Start with finding the shipment across AVi, VeM or $M\infty$ (why?).]*

(c) *Verify your results from part b by finding the tableau associated with basis β.*

Workout 10.1.8 *Let $\beta = \{AVi, FP, MF, PA, VeM, M\infty\}$ in Problem 10.1.2.*

(a) *Draw the network corresponding to β.*

(b) *Use Figure 10.1 to label the nodes of your result in part a so that the node labels differ by the costs of the basic arcs. [HINT: Start anywhere with any value you like, and work outwards.]*

(c) *Find the tableau associated with basis β and compare it to your results from part b as follows. For each arc not in the basis, subtract your tail label from your head label. Then subtract the result from the arc cost. What do you notice?*