Chapter 11

Network Optimization

11.1 Introduction

Network optimization is a special type of linear programming model. Network models have three main advantages over linear programming:

1. They can be solved very quickly. Problems whose linear program would have 1000 rows and 30,000 columns can be solved in a matter of seconds. This allows network models to be used in many applications (such as real-time decision making) for which linear programming would be inappropriate.
2. They have naturally integer solutions. By recognizing that a problem can be formulated as a network program, it is possible to solve special types of integer programs without resorting to the ineffective and time consuming integer programming algorithms.
3. They are intuitive. Network models provide a language for talking about problems that is much more intuitive than the "variables, objective, and constraints" language of linear and integer programming.

Of course these advantages come with a drawback: network models cannot formulate the wide range of models that linear and integer programs can. However, they occur often enough that they form an important tool for real decision making.

11.2 Terminology

A *network* or *graph* consists of points, and lines connecting pairs of points. The points are called *nodes* or vertices. The lines are called *arcs*. The arcs may have a direction on them, in which case they are called *directed arcs*. If an arc has no direction, it is often called an *edge*. If all the arcs in a network are directed, the network is a *directed network*. If all the arcs are undirected, the network is an *undirected network*.

Two nodes may be connected by a series of arcs. A *path* is a sequence of distinct arcs (no nodes repeated) connecting the nodes. A *directed path* from node i to node j is a sequence of arcs, each of whose direction (if any) is towards j. An *undirected path* may have directed arcs pointed in either direction.

A path that begins and ends at the same node is a *cycle* and may be either directed or undirected.

A network is *connected* if there exists an undirected path between any pair of nodes. A connected network without any cycle is called a *tree*, mainly because it looks like one.

11.3 Examples

There are many examples of using network flows in practice. Here are a few:

11.3.1 Shortest Paths

Consider a phone network. At any given time, a message may take a certain amount of time to traverse each line (due to congestion effects, switching delays, and so on). This time can vary greatly minute by minute and telecommunication companies spend a lot of time and money tracking these delays and communicating these delays throughout the system. Assuming a centralized switcher knows these delays, there remains the problem of routing a call so as to minimize the delays. So, in figure 11.1, what is the least delay path from LA to Boston? How can we find that path quickly?

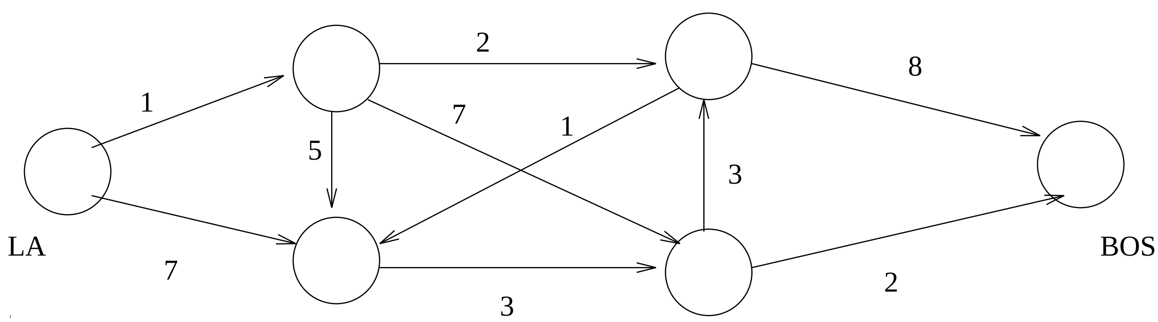

Figure 11.1: Phone Network

This is an example of a particular type of network model, called the *shortest path problem*. In such a problem, you have a network with costs on the edges and two special nodes: a start node and a finish node. The goal is to find a path from the start node to the finish node whose total weight is minimized.

Here is another problem that might not appear to be a shortest path problem, but really is:

At a small but growing airport, the local airline company is purchasing a new tractor for a tractor-trailer train to bring luggage to and from the airplanes. A new mechanized luggage system will be installed in 3 years, so the tractor will not be needed after that. However, because it will receive heavy use, and maintenance costs are high, it may still be more economical to replace the tractor after 1 or 2 years. The following table gives the total net discounted cost associated with purchasing a tractor in year i and trading it in in year j (where year 0 is now):

		j	
	1	2	3
0	8	18	31
i 1	—	10	21
2	—	—	12

The problem is to determine at what times the tractor should be replaced (if ever) to minimize the total costs for tractors. How can this be formulated as a shortest path problem?

11.3.2 Maximum Flow

Another type of model again has a number on each arc, but now the number corresponds to a capacity. This limits the flow on the arc to be no more than that value. For instance, in a distribution system, the capacity might be the limit on the amount of material (in tons, say) that can go over a particular distribution channel. We would then be concerned with the capcity of the network: how much can be sent from a source node to the destination node? Using the same network as above, treating the numbers as capacities, how much can be sent from LA to Boston?

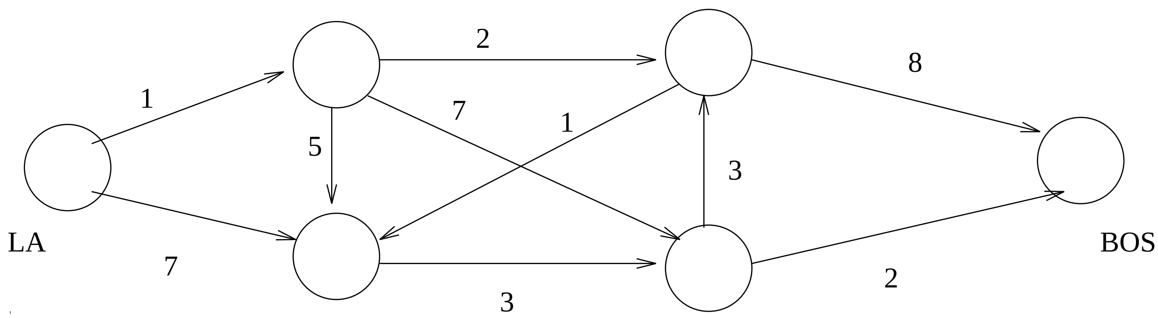

Figure 11.2: Distribution Network

Associated with the maximum flow is a bottleneck: a set of arcs whose total capacity equals the maximum flow, and whose removal leaves no path from source to destination in the network. It is actually a nontrivial result to show that the maximum flow equals the size of the minimum bottleneck. It is also an interesting task to find the bottleneck in the above example.

Maximum flow models occur in applications where cost is not an issue, and the goal is to maximize the number of items handled (in a broad sense). Here is a similar problem that can be formulated as a maximum flow problem (this was part of a conversation with a Chicago consulting firm on how to improve their internal operations):

Over the next three months, there are four projects to be completed. Project A requires 8 person-months of work, and must be done by the end of the second month. Project B requires 3 person-months and must be done by end of the first month. Project C can start at the beginning of the second month, and requires 7 person-months of work, no more than 2 of which are in the second month. Project D requires 6 person-months, no more than 3 of which occur in any month. There are 9 consultants available in the first and second month, and 6 in the third. Is it possible to meet the project deadlines? Surprisingly, this is a maximum flow problem. One nice (and surprising) aspect of the result is that this shows that no consultant need split a month's work between jobs.

Exercise 78 The Smith's, Jones', Johnson's, and Brown's are attending a picnic. There are four members of each family. Four cars are available to transport them: two cars that can hold four people, and two that can hold three. No more than 2 members of any family can be in any car. The goal is to maximize the number of people who can attend the picnic. Formulate this as a maximum flow problem. Give the nodes and arcs of the network and state which is the starting node and which is the destination. For each arc, give its capacity.

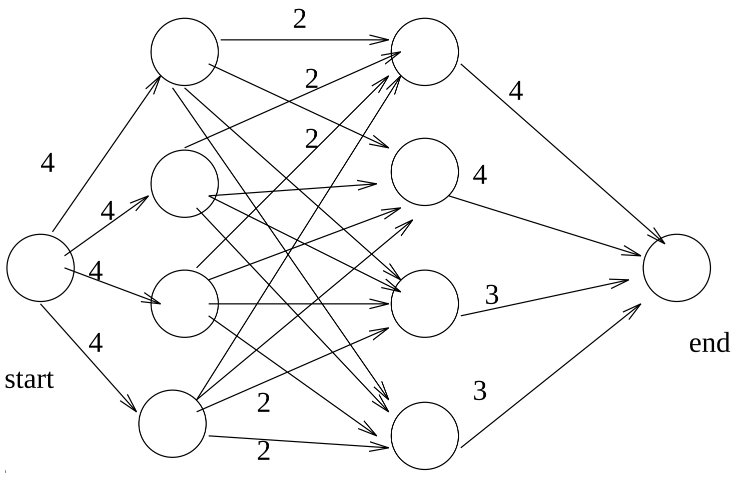

Figure 11.3: Solution of the exercise.

11.3.3 Transportation Problem

Consider the following snow removal problem: there are a number of districts in a city. After a snowfall, the snow in each area must be moved out of the district into a convenient location. In Montreal (from where this example is taken), these locations are large grates (leading to the sewer system), a couple large pits, and a couple entry points to the river. Each of these destinations has a capacity. The goal is to minimize the distance traveled to handle all of the snow.

This problem is an example of a transportation problem. In such a problem, there are a set of nodes called sources, and a set of nodes called destinations. All arcs go from a source to a destination. There is a per–unit cost on each arc. Each source has a supply of material, and each destination has a demand. We assume that the total supply equals the total demand (possibly adding a fake source or destination as needed). For the snow removal problem, the network might look like that in figure 11.4.

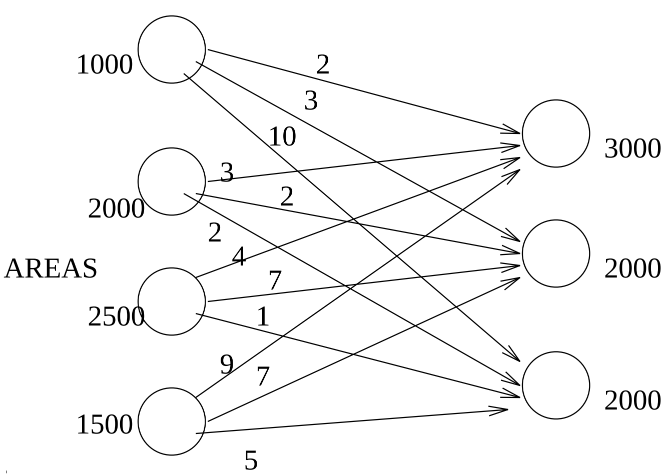

Figure 11.4: Snow Transportation Network

Transportation problems are often used in, surprise, transportation planning. For instance, in an application where goods are at a warehouse, one problem might be to assign customers to a warehouse so as to meet their demands. In such a case, the warehouses are the sources, the customers are the destinations, and the costs represent the per–unit transportation costs.

Here is another example of this:

One of the main products of P&T Company is canned peas. The peas are prepared at three canneries (near Bellingham, Washington; Eugene, Oregon; and Albert Lea, Minnesota) and are then shipped by truck to four distributing warehouses in Sacremento, California; Salt Lake City, Utah; Rapid City, South Dakota; and Albuquerque, New Mexico. Because shipping costs are a major expense, management has begun a study to reduce them. For the upcoming season, an estimate has been made of what the output will be from each cannery, and how much each warehouse will require to satisfy its customers. The shipping costs from each cannery to each warehouse has also been determined. This is summarized in the next table.

Shipping cost per truckload	Warehouse 1	2	3	4	Output
Cannery 1	464	513	654	867	75
Cannery 2	352	416	690	791	125
Cannery 3	995	682	388	685	100
Requirement	80	65	70	85	

You should find it an easy exercise to model this as a linear program. If we let x_{ij} be the number of truckloads shipped from cannery i to warehouse j, the problem is to

$$
\begin{array}{ll}
\text{minimize} & 464x_{11} + 513x_{12} + 654x_{13} + 867x_{14} + 352x_{21} + \ldots + 685x_{34} \\
\text{subject to} & \\
& x_{11} + x_{12} + x_{13} + x_{14} = 75 \\
& x_{21} + x_{22} + x_{23} + x_{24} = 125 \\
& x_{31} + x_{32} + x_{33} + x_{34} = 100 \\
& x_{11} + x_{21} + x_{31} = 80 \\
& x_{12} + x_{22} + x_{32} = 65 \\
& x_{13} + x_{23} + x_{33} = 70 \\
& x_{14} + x_{24} + x_{34} = 85 \\
& x_{ij} \geq 0 \text{ for all } i \text{ and } j.
\end{array}
$$

This is an example of the **transportation model**. As has been pointed out, this problem has a lot of nice structure. All the coefficients are 1 and every variable appears in exactly two constraints. It is this structure that lets the simplex algorithm be specialized into an extremely efficient algorithm.

What defines a transportation model? In general, the transportation model is concerned with distributing (literally or figuratively) a commodity from a group of supply centers, called **sources** to a group of receiving centers, called **destinations** to minimize total cost.

In general, source i has a supply of s_i units, and destination j has a demand for d_j units. The cost of distributing items from a source to a destination is proportional to the number of units. This data can be conveniently represented in a table like that for the sample problem.

We will generally assume that the total supply equals the total demand. If this is not true for a particular problem, *dummy* sources or destinations can be added to make it true. The text refers to such a problem as a *balanced transportation problem*. These dummy centers may have zero distribution costs, or costs may be assigned to represent unmet supply or demand.

For example, suppose that cannery 3 makes only 75 truckloads. The total supply is now 25 units too little. A dummy supply node can be added with supply 25 to balance the problem, and the cost from the dummy to each warehouse can be added to represent the cost of not meeting the warehouse's demand.

The transportation problem has a couple of nice properties:

Feasibility. As long as the supply equals demand, there exists a feasible solution to the problem.

Integrality. If the supplies and demands are integer, every basic solution (including optimal ones) have integer values. Therefore, it is not necessary to resort to integer programming to find integer solutions. Linear programming suffices. Note that this does not mean that each destination will be supplied by exactly one source.

In the pea shipping example, a basic solution might be to ship 20 truckloads from cannery 1 to warehouse 2 and the remaining 55 to warehouse 4, 80 from cannery 2 to warehouse 1 and 45 to warehouse 2 and, finally, 70 truckloads from cannery 3 to warehouse 3 and 30 to warehouse 4. Even though the linear programming formulation of the pea shipping example has seven constraints other than nonnegativity, a basic solution has only six basic variables! This is because the constraints are linearly dependent: the sum of the first three is identical to the sum of the last four. As a consequence, the feasible region defined by the constraints would remain the same if we only kept six of them. In general, a basic solution to the transportation model will have a number of basic variables equal to the number of sources plus the number of destinations minus one.

Exercise 79 Formulate the following production problem as a transportation model. The demands for a given item are 150, 250, 200 units for the next three months. The demand may be satisfied by

- excess production in an earlier month held in stock for later consumption,
- production in the current month,
- excess production in a later month backordered for preceding months.

The variable production cost per unit in any month is \$6.00. A unit produced for later consumption will incur a storage cost at the rate of \$1 per unit per month. On the other hand, backordered items incur a penalty cost of \$3.00 per unit per month. The production capacity in each of the next three months is 200 units. The objective is to devise a minimum cost production plan.

Exercise 80 Avertz RentaCar needs to redeploy its automobiles to correct imbalances in the system. Currently Avertz has too many cars in New York (with 10 cars excess) and Chicago (12 cars excess). Pittsburgh would like up to 6 cars, Los Angeles up to 14 cars, and Miami up to 7 cars (note that more cars are demanded than are available). The cost of transporting a car from one city to another is given by:

	Pittsburgh	Los Angeles	Miami
New York	50	250	100
Chicago	25	200	125

(a) Formulate this problem as a transportation problem. Clearly give the nodes and arcs. For each node, give the supply or demand at the node. For each arc, give the cost. Assume that unmet demand at a city has no cost but that no city can end up with excess cars.

(b) It turns out that unmet demand costs \$50/car in Pittsburgh, \$75/car in LA, and \$100/car in Miami. Update your transportation formulation in (a) to account for this change.

11.3.4 Assignment Problem

A special case of the transportation problem is the *assignment problem* which occurs when each supply is 1 and each demand is 1. In this case, the integrality implies that every supplier will be assigned one destination and every destination will have one supplier. The costs give the charge for assigning a supplier and destination to each other.

Example 11.3.1 *A company has three new machines of different types. There are four different plants that could receive the machines. Each plant can only receive one machine, and each machine can only be given to one plant. The expected profit that can be made by each plant if assigned a machine is as follows:*

		Plant		
	1	*2*	*3*	*4*
Machine 1	*13*	*16*	*12*	*11*
Machine 2	*15*	*0*	*13*	*20*
Machine 3	*5*	*7*	*10*	*6*

This is a transportation problem with all supplies and demands equal to 1, so it is an assignment problem.

Note that a balanced problem must have the same number of supplies and demands, so we must add a dummy machine (corresponding to receiving no machine) and assign a zero cost for assigning the dummy machine to a plant.

Exercise 81 Albert, Bob, Carl, and David are stranded on a desert island with Elaine, Francine, Gert, and Holly. The "compatibility measures" in the next table indicate the happiness each couple would experience if they spent all their time together. If a couple spends only a partial amount of time together, then the happiness is proportional to the fraction of time spent. So if Albert and Elaine spend half their time together, they earn happiness of 7/2.

	E	F	G	H
A	7	5	8	2
B	7	8	9	4
C	3	5	7	9
D	5	5	6	7

(a) Let x_{ij} be the fraction of time that man i spends with woman j. Formulate a linear program that maximizes the total happiness of the island (assume that no man spends time with any other man, no woman spends time with any other woman, and no one spends time alone).

(b) Explain why exactly four x_{ij} will be 1 and the rest will be 0 in an optimal solution to this linear program. This result is called the *Marriage Theorem* for obvious reasons.

(c) Do you think that this theorem will hold if we allow men to spend time with men and women to spend time with women?

11.4 Unifying Model: Minimum Cost Network Flows

All of the above models are special types of network flow problems: they each have a specialized algorithm that can find solutions hundreds of times faster than plain linear programming.

They can all also be seen as examples of a much broader model, the minimum cost network flow model. This model represents the broadest class of problem that can be solved much faster than linear programming while still retaining such nice properties as integrality of solution and appeal of concept.

Like the maximum flow problem, it considers flows in networks with capacities. Like the shortest path problem, it considers a cost for flow through an arc. Like the transportation problem, it allows multiple sources and destinations. In fact, all of these problems can be seen as *special cases* of the minimum cost flow problem.

Consider a directed network with n nodes. The decision variables are x_{ij}, the flow through arc (i, j). The given information includes:

- c_{ij}: cost per unit of flow from i to j (may be negative),
- u_{ij}: capacity (or upper bound) on flow from i to j,
- b_i: net flow generated at i.

This last value has a sign convention:

- $b_i > 0$ if i is a supply node,
- $b_i < 0$ if i is a demand node,
- $b_i = 0$ if i is a transshipment node.

The objective is to minimize the total cost of sending the supply through the network to satisfy the demand.

Note that for this model, it is not necessary that every arc exists. We will use the convention that summations are only taken over arcs that exist. The linear programming formulation for this problem is:

$$\begin{array}{ll} \text{Minimize} & \sum_i \sum_j c_{ij} x_{ij} \\ \text{Subject to} & \sum_j x_{ij} - \sum_j x_{ji} = b_i \text{ for all nodes } i, \\ & 0 \le x_{ij} \le u_{ij} \text{ for all arcs } (i, j). \end{array}$$

Again, we will assume that the network is balanced, so $\sum_i b_i = 0$, since dummies can be added as needed. We also still have a nice integrality property. If all the b_i and u_{ij} are integral, then the resulting solution to the linear program is also integral.

Minimum cost network flows are solved by a variation of the simplex algorithm and can be solved more than 100 times faster than equivalently sized linear programs. From a modeling point of view, it is most important to know the sort of things that can and cannot be modeled in a single network flow problem:

Can do

1. Lower bounds on arcs. If a variable x_{ij} has a lower bound of l_{ij}, upper bound of u_{ij}, and cost of c_{ij}, change the problem as follows:
   - Replace the upper bound with $u_{ij} - l_{ij}$,
   - Replace the supply at i with $b_i - l_{ij}$,
   - Replace the supply at j with $b_i + l_{ij}$,

   Now you have a minimum cost flow problem. Add $c_{ij}l_{ij}$ to the objective after solving and l_{ij} to the flow on arc (i, j) to obtain a solution of the original problem.
2. Upper bounds on flow through a node. Replace the node i with nodes i' and i''. Create an arc from i' to i'' with the appropriate capacity, and cost 0. Replace every arc (j, i) with one from j to i' and every arc (i, j) with one from i'' to j. Lower bounds can also be handled this way.
3. Convex, piecewise linear costs on arc flows (for minimization). This is handled by introducing multiple arcs between the nodes, one for each portion of the piecewise linear function. The convexity will assure that costs are handled correctly in an optimal solution.

Can't do

1. Fixed cost to use a node.
2. Fixed cost to use an arc.
3. "Proportionality of flow." That is, if one unit enters node i, then you insist that .5 units go to node j and .5 to node k.
4. Gains and losses of flow along arcs, as in power distribution.

Note that although these cannot be done in a single network, it may be possible to use the solutions to multiple networks to give you an answer. For instance, if there is only one arc with a fixed cost, you can solve both with and without the arc to see if it is advantageous to pay the fixed cost.

Exercise 82 Here is an example of a problem that doesn't look like a network flow problem, but it really is:

A company must meet the following demands for cash at the beginning of each of the next six months:

Month	1	2	3	4	5	6
Needs	\$200	\$100	\$50	\$80	\$160	\$140

At the beginning of month 1, the company has \$150 in cash and \$200 worth of bond 1, \$100 worth of bond 2 and \$400 worth of bond 3. Of course, the company will have to sell some bonds to meet demands, but a penalty will be charged for any bonds sold before the end of month 6. The penalties for selling \$1 worth of each bond are shown in the table below.

		Month of sale					
		1	2	3	4	5	6
Bond	1	\$0.07	\$0.06	\$0.06	\$0.04	\$0.03	\$0.03
	2	\$0.17	\$0.17	\$0.17	\$0.11	\$0	\$0
	3	\$0.33	\$0.33	\$0.33	\$0.33	\$0.33	\$0

(a) Assuming that all bills must be paid on time, formulate a balanced transportation problem that can be used to minimize the cost of meeting the cash demands for the next six months.

(b) Assume that payment of bills can be made after they are due, but a penalty of \$0.02 per month is assessed for each dollar of cash demands that is postponed for one month. Assuming all bills must be paid by the end of month 6, develop a transshipment model that can be used to minimize the cost of paying the next six months' bills.

[*Hint:* Transshipment points are needed, in the form C_t = cash available at beginning of month t after bonds for month t have been sold, but before month t demand is met. Shipments into C_t occur from bond sales and C_{t-1}. Shipments out of C_t occur to C_{t+1} and demands for months $1, 2, \ldots, t$.]

Exercise 83 Oil R' Us has oil fields in San Diego and Los Angeles. The San Diego field can produce up to 500,000 barrels per day (bpd); Los Angeles can produce up to 400,000 bpd. Oil is sent to a refinery, either in Dallas or Houston. It costs \$700 to refine 100,000 bpd in Dallas and \$900 to refine 100,000 bpd in Houston. The refined oil is then shipped to either Chicago or New York. Chicago requires exactly 400,000 bpd and New York requires exactly 300,000 bpd. The shipping costs (per 100,000 bpd) are given as follows:

	Dallas	Houston	New York	Chicago
LA	\$300	\$110	—	—
San Diego	\$420	\$100	—	—
Dallas	—	—	\$450	\$550
Houston	—	—	\$470	\$530

(a) Formulate this problem as a minimum cost flow problem. Clearly give the network. Give the cost and capacity on each arc and the net supply requirement on each node.

(b) Suppose Dallas could process no more than 300,000 bpd. How would you modify your formulation?

Exercise 84 A company produces a single product at two plants, A and B. A customer demands 12 units of the product this month, and 20 units of the product next month. Plant A can produce 7 items per month. It costs \$12 to produce each item at A this month and \$18 to produce at A next month. Plant B can produce 14 items per month. It costs \$14 to produce each item at B this month and \$20 to produce each item at B next month. Production from this month can be held in inventory to satisfy next month's demand at a cost of \$3/unit.

(a) Formulate the problem of determining the least cost way of meeting demand as a balanced transportation problem. Give the costs and node supplies, and describe the interpretation of each node and arc.

(b) Suppose no more than six items can be held in inventory between this month and next month. Formulate the resulting problem as a minimum cost flow problem. Draw the network, give the interpretation for each node and arc, and give the supply/demand at each node and the cost and capacity for each arc.

11.5 Generalized Networks

The final model I would like to familiarize you with is called the generalized network model. In this model, there may be gains or loses as flow goes along an arc. Each arc has a *multiplier* to represent these gains and loses. For instance, if 3 units of flow enter an arc with a multiplier of .5, then 1.5 unit of flow exit the arc. Such a model can still be represented as a linear program, and there are specialized techniques that can solve such models much faster than linear programming (but a bit slower than regular network flow problems). The optimal solution need not be integer however.

The standard example of a generalized network is in power generation: as electricty moves over wires, there is some unavoidable loss along the way. This loss is represented as a mutliplier. Here is another example on how generalized networks might be used:

Consider the world's currency market. Given two currencies, say the Yen and the USDollar, there is an exchange rate between them (currently about 110 Yen to the Dollar). It is axiomatic of a arbitrage-free market that there is no method of converting, say, a Dollar to Yen then to Deutsch Marks, to Francs, then Pounds, and to Dollars so that you end up with more than a dollar. How would you recognize when there is an arbitrage possibility?

For this model, we use a node to represent each currency. An arc between currency i and j gives the exchange rate from i to j. We begin with a single dollar and wish to send flow through the network and maximize the amount we generate back at the USDollar node. This can be drawn as in figure 11.5.

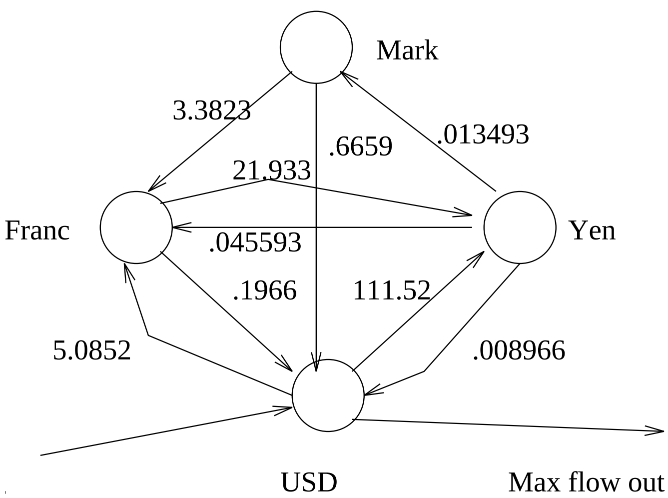

Figure 11.5: Financial Exchange Network

These are actual trades made on November 10, 1996. It is not obvious, but the Dollar-Yen-Mark-Dollar conversion actually makes \$0.002. How would you formulate a linear program to recognize this?

11.6 Conclusions

One class can only provide a bare introduction to this area. The points we would like you to take away are:

1) Networks are an important subclass of linear programs that are intuitive, easy to solve, and have nice integrality properties.

2) Problems that might not look like networks might be networks.

3) Networks provide a useful way to think about problems even if there are additional constraints or variables that preclude use of networks for modeling the whole problem.

In practice, you would generally save using the fastest network codes until the final implementation phase. Until then, linear programming codes will tend to be sufficiently fast to prove the concepts.