## A production problem

A firm produces $n$ different goods using $m$ different raw materials. Let $b_i$, $i = 1, \ldots, m$, be the available amount of the $i$th raw material. The $j$th good, $j = 1, \ldots, n$, requires $a_{ij}$ units of the $i$th material and results in a revenue of $c_j$ per unit produced. The firm faces the problem of deciding how much of each good to produce in order to maximize its total revenue.

In this example, the choice of the decision variables is simple. Let $x_j$, $j = 1, \ldots, n$, be the amount of the $j$th good. Then, the problem facing the firm can be formulated as follows:

$$
\begin{array}{llll}
\text{maximize} & c_1 x_1 + \cdots + c_n x_n & & \\
\text{subject to} & a_{i1} x_1 + \cdots + a_{in} x_n \leq b_i, & & i = 1, \ldots, m, \\
& x_j \geq 0, & & j = 1, \ldots, n.
\end{array}
$$

## Production planning by a computer manufacturer

The example that we consider here is a problem that Digital Equipment Corporation (DEC) had faced in the fourth quarter of 1988. It illustrates the complexities and uncertainties of real world applications, as well as the usefulness of mathematical modeling for making important strategic decisions.

In the second quarter of 1988, DEC introduced a new family of (single CPU) computer systems and workstations: GP-1, GP-2, and GP-3, which are general purpose computer systems with different memory, disk storage, and expansion capabilities, as well as WS-1 and WS-2, which are workstations. In Table 1.1, we list the models, the list prices, the average disk usage per system, and the memory usage. For example, GP-1 uses four 256K memory boards, and 3 out of every 10 units are produced with a disk drive.

| System | Price | # disk drives | # 256K boards |
|---|---|---|---|
| **GP-1** | $60,000 | 0.3 | 4 |
| **GP-2** | $40,000 | 1.7 | 2 |
| **GP-3** | $30,000 | 0 | 2 |
| **WS-1** | $30,000 | 1.4 | 2 |
| **WS-2** | $15,000 | 0 | 1 |

**Table 1.1:** Features of the five different DEC systems.

Shipments of this new family of products started in the third quarter and ramped slowly during the fourth quarter. The following difficulties were anticipated for the next quarter:

(a) The in-house supplier of CPUs could provide at most 7,000 units, due to debugging problems.

(b) The supply of disk drives was uncertain and was estimated by the manufacturer to be in the range of 3,000 to 7,000 units.

(c) The supply of 256K memory boards was also limited in the range of 8,000 to 16,000 units.

On the demand side, the marketing department established that the maximum demand for the first quarter of 1989 would be 1,800 for GP-1 systems, 300 for GP-3 systems, 3,800 systems for the whole GP family, and 3,200 systems for the WS family. Included in these projections were 500 orders for GP-2, 500 orders for WS-1, and 400 orders for WS-2 that had already been received and had to be fulfilled in the next quarter.

In the previous quarters, in order to address the disk drive shortage, DEC had produced GP-1, GP-3, and WS-2 with no disk drive (although 3 out of 10 customers for GP-1 systems wanted a disk drive), and GP-2, WS-1 with one disk drive. We refer to this way of configuring the systems as the constrained mode of production.

In addition, DEC could address the shortage of 256K memory boards by using two alternative boards, instead of four 256K memory boards, in the GP-1 system. DEC could provide 4,000 alternative boards for the next quarter.

It was clear to the manufacturing staff that the problem had become complex, as revenue, profitability, and customer satisfaction were at risk. The following decisions needed to be made:

(a) The production plan for the first quarter of 1989.

(b) Concerning disk drive usage, should DEC continue to manufacture products in the constrained mode, or should it plan to satisfy customer preferences?

(c) Concerning memory boards, should DEC use alternative memory boards for its GP-1 systems?

(d) A final decision that had to be made was related to tradeoffs between shortages of disk drives and of 256K memory boards. The manufacturing staff would like to concentrate their efforts on either decreasing the shortage of disks or decreasing the shortage of 256K memory boards. Hence, they would like to know which alternative would have a larger effect on revenue.

In order to model the problem that DEC faced, we introduce variables $x_1$, $x_2$, $x_3$, $x_4$, $x_5$, that represent the number (in thousands) of GP-1, GP-2, GP-3, WS-1, and WS-2 systems, respectively, to be produced in the next quarter. Strictly speaking, since $1000x_i$ stands for number of units, it must be an integer. This can be accomplished by truncating each $x_i$ after the third decimal point; given the size of the demand and the size of the variables $x_i$, this has a negligible effect and the integrality constraint on $1000x_i$ can be ignored.

DEC had to make two distinct decisions: whether to use the constrained mode of production regarding disk drive usage, and whether to use alternative memory boards for the GP-1 system. As a result, there are four different combinations of possible choices.

We first develop a model for the case where alternative memory boards are not used and the constrained mode of production of disk drives is selected. The problem can be formulated as follows:

$$\text{maximize} \quad 60x_1 + 40x_2 + 30x_3 + 30x_4 + 15x_5 \qquad \text{(total revenue)}$$

subject to the following constraints:

$$\begin{array}{rcll}
x_1 + x_2 + x_3 + x_4 + x_5 & \leq & 7 & \text{(CPU availability)} \\
4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 & \leq & 8 & \text{(256K availability)} \\
x_2 + x_4 & \leq & 3 & \text{(disk drive availability)} \\
x_1 & \leq & 1.8 & \text{(max demand for GP-1)} \\
x_3 & \leq & 0.3 & \text{(max demand for GP-3)} \\
x_1 + x_2 + x_3 & \leq & 3.8 & \text{(max demand for GP)} \\
x_4 + x_5 & \leq & 3.2 & \text{(max demand for WS)} \\
x_2 & \geq & 0.5 & \text{(min demand for GP-2)} \\
x_4 & \geq & 0.5 & \text{(min demand for WS-1)} \\
x_5 & \geq & 0.4 & \text{(min demand for WS-2)} \\
x_1, x_2, x_3, x_4, x_5 & \geq & 0. &
\end{array}$$

Notice that the objective function is in millions of dollars. In some respects, this is a pessimistic formulation, because the 256K memory and disk drive availability were set to 8 and 3, respectively, which is the lowest value in the range that was estimated. It is actually of interest to determine the solution to this problem as the 256K memory availability ranges from 8 to 16, and the disk drive availability ranges from 3 to 7, because this provides valuable information on the sensitivity of the optimal solution on availability. In another respect, the formulation is optimistic because, for example, it assumes that the revenue from GP-1 systems is $60x_1$ for any $x_1 \leq 1.8$, even though a demand for 1,800 GP-1 systems is not guaranteed.

In order to accommodate the other three choices that DEC had, some of the problem constraints have to be modified, as follows. If we use the unconstrained mode of production for disk drives, the constraint $x_2+x_4 \leq 3$ is replaced by

$$0.3x_1 + 1.7x_2 + 1.4x_4 \leq 3.$$

Furthermore, if we wish to use alternative memory boards in GP-1 systems, we replace the constraint $4x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 \leq 8$ by the two constraints

$$\begin{aligned} 2x_1 &\leq 4, \\ 2x_2 + 2x_3 + 2x_4 + x_5 &\leq 8. \end{aligned}$$

The four combinations of choices lead to four different linear programming problems, each of which needs to be solved for a variety of parameter values because, as discussed earlier, the right-hand side of some of the constraints is only known to lie within a certain range. Methods for solving linear programming problems, when certain parameters are allowed to vary, will be studied in Chapter 5, where this case study is revisited.

## Multiperiod planning of electric power capacity

A state wants to plan its electricity capacity for the next $T$ years. The state has a forecast of $d_t$ megawatts, presumed accurate, of the demand for electricity during year $t = 1, \ldots, T$. The existing capacity, which is in oil-fired plants, that will not be retired and will be available during year $t$, is $e_t$. There are two alternatives for expanding electric capacity: coal-fired or nuclear power plants. There is a capital cost of $c_t$ per megawatt of coal-fired capacity that becomes operational at the beginning of year $t$. The corresponding capital cost for nuclear power plants is $n_t$. For various political and safety reasons, it has been decided that no more than 20% of the total capacity should ever be nuclear. Coal plants last for 20 years, while nuclear plants last for 15 years. A least cost capacity expansion plan is desired.

The first step in formulating this problem as a linear programming problem is to define the decision variables. Let $x_t$ and $y_t$ be the amount of coal (respectively, nuclear) capacity brought on line at the beginning of year $t$. Let $w_t$ and $z_t$ be the total coal (respectively, nuclear) capacity available in year $t$. The cost of a capacity expansion plan is therefore,

$$\sum_{t=1}^{T}(c_t x_t + n_t y_t).$$

Since coal-fired plants last for 20 years, we have

$$w_t = \sum_{s=\max\{1,t-19\}}^{t} x_s, \qquad t = 1, \ldots, T.$$

Similarly, for nuclear power plants,

$$z_t = \sum_{s=\max\{1,t-14\}}^{t} y_s, \qquad t = 1, \ldots, T.$$

Since the available capacity must meet the forecasted demand, we require

$$w_t + z_t + e_t \geq d_t, \qquad t = 1, \ldots, T.$$

Finally, since no more than 20% of the total capacity should ever be nuclear, we have

$$\frac{z_t}{w_t + z_t + e_t} \leq 0.2,$$

which can be written as

$$0.8z_t - 0.2w_t \leq 0.2e_t.$$

Summarizing, the capacity expansion problem is as follows:

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{t=1}^{T}(c_t x_t + n_t y_t) & \\
\text{subject to} & w_t - \displaystyle\sum_{s=\max\{1,t-19\}}^{t} x_s = 0, & t = 1, \ldots, T, \\
& z_t - \displaystyle\sum_{s=\max\{1,t-14\}}^{t} y_s = 0, & t = 1, \ldots, T, \\
& w_t + z_t \geq d_t - e_t, & t = 1, \ldots, T, \\
& 0.8z_t - 0.2w_t \leq 0.2e_t, & t = 1, \ldots, T, \\
& x_t, y_t, w_t, z_t \geq 0, & t = 1, \ldots, T.
\end{array}$$

We note that this formulation is not entirely realistic, because it disregards certain economies of scale that may favor larger plants. However, it can provide a ballpark estimate of the true cost.

## A scheduling problem

In the previous examples, the choice of the decision variables was fairly straightforward. We now discuss an example where this choice is less obvious.

A hospital wants to make a weekly night shift (12pm-8am) schedule for its nurses. The demand for nurses for the night shift on day $j$ is an integer $d_j$, $j = 1, \ldots, 7$. Every nurse works 5 days in a row on the night shift. The problem is to find the minimal number of nurses the hospital needs to hire.

One could try using a decision variable $y_j$ equal to the number of nurses that work on day $j$. With this definition, however, we would not be able to capture the constraint that every nurse works 5 days in a row. For this reason, we choose the decision variables differently, and define $x_j$ as the number of nurses starting their week on day $j$. (For example, a nurse whose week starts on day 5 will work days $5, 6, 7, 1, 2$.) We then have the following problem formulation:

$$\begin{array}{ll}
\text{minimize} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\
\text{subject to} & x_1 \phantom{{}+ x_2 + x_3} + x_4 + x_5 + x_6 + x_7 \geq d_1 \\
& x_1 + x_2 \phantom{{}+ x_3 + x_4} + x_5 + x_6 + x_7 \geq d_2 \\
& x_1 + x_2 + x_3 \phantom{{}+ x_4 + x_5} + x_6 + x_7 \geq d_3 \\
& x_1 + x_2 + x_3 + x_4 \phantom{{}+ x_5 + x_6} + x_7 \geq d_4 \\
& x_1 + x_2 + x_3 + x_4 + x_5 \phantom{{}+ x_6 + x_7} \geq d_5 \\
& \phantom{x_1 +{}} x_2 + x_3 + x_4 + x_5 + x_6 \phantom{{}+ x_7} \geq d_6 \\
& \phantom{x_1 + x_2 +{}} x_3 + x_4 + x_5 + x_6 + x_7 \geq d_7 \\
& x_j \geq 0, \qquad x_j \text{ integer.}
\end{array}$$

This would be a linear programming problem, except for the constraint that each $x_j$ must be an integer, and we actually have a linear *integer programming* problem. One way of dealing with this issue is to ignore ("relax") the integrality constraints and obtain the so-called *linear programming relaxation* of the original problem. Because the linear programming problem has fewer constraints, and therefore more options, the optimal cost will be less than or equal to the optimal cost of the original problem. If the optimal solution to the linear programming relaxation happens to be integer, then it is also an optimal solution to the original problem. If it is not integer, we can round each $x_j$ upwards, thus obtaining a feasible, but not necessarily optimal, solution to the original problem. It turns out that for this particular problem, an optimal solution can be found without too much effort. However, this is the exception rather than the rule: finding optimal solutions to general integer programming problems is typically difficult; some methods will be discussed in Chapter 11.

## Choosing paths in a communication network

Consider a communication network consisting of $n$ nodes. Nodes are connected by communication links. A link allowing one-way transmission from node $i$ to node $j$ is described by an ordered pair $(i, j)$. Let $\mathcal{A}$ be the set of all links. We assume that each link $(i, j) \in \mathcal{A}$ can carry up to $u_{ij}$ bits per second. There is a positive charge $c_{ij}$ per bit transmitted along that link. Each node $k$ generates data, at the rate of $b^{k\ell}$ bits per second, that have to be transmitted to node $\ell$, either through a direct link $(k, \ell)$ or by tracing a sequence of links. The problem is to choose paths along which all data reach their intended destinations, while minimizing the total cost. We allow the data with the same origin and destination to be split and be transmitted along different paths.

In order to formulate this problem as a linear programming problem, we introduce variables $x_{ij}^{k\ell}$ indicating the amount of data with origin $k$ and destination $\ell$ that traverse link $(i,j)$. Let

$$b_i^{k\ell} = \begin{cases} b^{k\ell}, & \text{if } i = k, \\ -b^{k\ell}, & \text{if } i = \ell, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, $b_i^{k\ell}$ is the net inflow at node $i$, from outside the network, of data with origin $k$ and destination $\ell$. We then have the following formulation:

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in\mathcal{A}} \sum_{k=1}^{n} \sum_{\ell=1}^{n} c_{ij} x_{ij}^{k\ell} \\ \text{subject to} \quad & \sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{k\ell} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{k\ell} = b_i^{k\ell}, && i,k,\ell = 1,\ldots,n, \\ & \sum_{k=1}^{n} \sum_{\ell=1}^{n} x_{ij}^{k\ell} \le u_{ij}, && (i,j) \in \mathcal{A}, \\ & x_{ij}^{k\ell} \ge 0, && (i,j) \in \mathcal{A}, \quad k,\ell = 1,\ldots,n. \end{aligned}$$

The first constraint is a flow conservation constraint at node $i$ for data with origin $k$ and destination $\ell$. The expression

$$\sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{k\ell}$$

represents the amount of data with origin and destination $k$ and $\ell$, respectively, that leave node $i$ along some link. The expression

$$\sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{k\ell}$$

represents the amount of data with the same origin and destination that enter node $i$ through some link. Finally, $b_i^{k\ell}$ is the net amount of such data that enter node $i$ from outside the network. The second constraint expresses the requirement that the total traffic through a link $(i,j)$ cannot exceed the link's capacity.

This problem is known as the *multicommodity flow* problem, with the traffic corresponding to each origin-destination pair viewed as a different commodity. A mathematically similar problem arises when we consider a transportation company that wishes to transport several commodities from their origins to their destinations through a network. There is a version of this problem, known as the minimum cost *network flow* problem, in which we do not distinguish between different commodities. Instead, we are given the amount $b_i$ of external supply or demand at each node $i$, and the objective is to transport material from the supply nodes to the demand nodes, at minimum cost. The network flow problem, which is the subject of Chapter 7, contains as special cases some important problems such as the shortest path problem, the maximum flow problem, and the assignment problem.