C H A P T E R

16 AGGREGATE AND WORKFORCE PLANNING

And I remember misinformation followed us like a plague,
Nobody knew from time to time if the plans were changed.
Paul Simon

16.1 Introduction

A variety of manufacturing management decisions require information about what a plant will produce over the next year or two. Exampes include the following:

1. *Staffing.* Recruiting and training new workers is a time-consuming process. Management needs a long-term production plan to decide how many and what type of workers to add and when to bring them on-line in order to meet production needs. Conversely, eliminating workers is costly and painful, but sometimes necessary. Anticipating reductions via a long-term plan makes it possible to use natural attrition, or other gentler methods, in place of layoffs to achieve at least part of the reductions.
2. *Procurement.* Contracts with suppliers are frequently set up well in advance of placing actual orders. For example, a firm might need an opportunity to "certify" the subcontractor for quality and other performance measures. Additionally, some procurement lead times are long (e.g., for high-technology components they may be six months or more). Therefore, decisions regarding contracts and long-lead-time orders must be made on the basis of a long-term production plan.
3. *Subcontracting.* Management must arrange contracts with subcontractors to manufacture entire components or to perform specific operations well in advance of actually sending out orders. Determining what types of subcontracting to use requires long-term projections of production requirements and a plan for in-house capacity modifications.
4. *Marketing.* Marketing personnel should make decisions on which products to promote on the basis of both a demand forecast *and* knowledge of which products have tight capacity and which do not. A long-term production plan incorporating planned capacity changes is needed for this.

The module in which we address the important question of what will be produced and when it will be produced over the long range is the **aggregate planning (AP)** module. As Figure 13.2 illustrated, the AP module occupies a central position in the production planning and control (PPC) hierarchy. The reason, or course, is that so many important decisions, such as those listed, depend on a long-term production plan.

Precisely because so many different decisions hinge on the long-range production plan, many different formulations of the AP module are possible. Which formulation is appropriate depends on what decision is being addressed. A model for determining the time of staffing additions may be very different from a model for deciding which products should be manufactured by outside subcontractors. Yet a different model might make sense if we want to address both issues simultaneously.

The staffing problem is of sufficient importance to warrant its own module in the hierarchy of Figure 13.2, the **workforce planning (WP)** module. Although high-level workforce planning (projections of total staffing increases or decreases, institution of training policies) can be done using only a rough estimate of future production based on the demand forecast, low-level staffing decisions (timing of hires or layoffs, scheduling usage of temporary hires, scheduling training) are often based on the more detailed production information contained in the aggregate plan. In the context of the PPC hierarchy in Figure 13.2, we can think of the AP module as either refining the output of the WP module or working in concert with the WP module. In any case, they are closely related. We highlight this relationship by treating aggregate planning and workforce planning together in this chapter.

As we mentioned in Chapter 13, linear programming is a particularly useful tool for formulating and solving many of the problems commonly faced in the aggregate planning and workforce planning modules. In this chapter, we will formulate several typical AP/WP problems as linear programs (LPs). We will also demonstrate the use of linear programming (LP) as a solution tool in various examples. Our goal is not so much to provide specific solutions to particular AP programs, but rather to illustrate general problem-solving approaches. The reader should be able to combine and extend our solutions to cover situations not directly addressed here.

Finally, while this chapter will not make an LP expert out of the reader, we do hope that he or she will become aware of how and where LP can be used in solving AP problems. If managers can recognize that particular problems are well suited to LP, they can easily obtain the technical support (consultants, internal experts) for carrying out the analysis and implementation. Unfortunately, far too few practicing managers make this connection; as a result, many are hammering away at problems that are well suited to linear programming with manual spreadsheets and other ad hoc approaches.

16.2 Basic Aggregate Planning

We start with a discussion of simple aggregate planning situations and work our way up to more complex cases. Throughout the chapter, we assume that we have a **demand forecast** available to us. This forecast is generated by the forecasting module and gives estimates of periodic demand over the **planning horizon.** Typically, periods are given in months, although further into the future they can represent longer intervals. For instance, periods 1 to 12 might represent the next 12 months, while periods 13 to 16 might represent the four quarters following these 12 months. A typical planning horizon for an AP module is one to three years.

16.2.1 A Simple Model

Our first scenario represents the simplest possible AP module. We consider this case not because it leads to a practical model, but because it illustrates the basic issues, provides a basis for considering more realistic situations, and showcases how linear programming can support the aggregate planning process. Although our discussion does not presume any background in linear programming, the reader interested in how and why LP works is advised to consult Appendix 16A, which provides an elementary overview of this important technique.

For modeling purposes, we consider the situation where there is only a single product, and the entire plant can be treated as a single resource. In every period, we have a demand forecast and a capacity constraint. For simplicity, we assume that demands represent customer orders that are due at the end of the period, and we neglect randomness and yield loss.

It is obvious under these simplifying assumptions that if demand is less than capacity in every period, the optimal solution is to simply produce amounts equal to demand in every period. This solution will meet all demand just-in-time and therefore will not build up any inventory between periods. However, if demand exceeds capacity in some periods, then we must work ahead (i.e., produce more than we need in some previous period). If demand cannot be met even by working ahead, we want our model to tell us this. To model this situation in the form of a linear program, we introduce the following notation:

t = an index of time periods, where $t = 1, \ldots, \bar{t}$, so $\bar{t}$ is planning horizon for problem

d_t = demand in period t, in physical units, standard containers, or some other appropriate quantity (assumed due at end of period)

c_t = capacity in period t, in same units used for d_t

r = profit per unit of product sold (not including inventory-carrying cost)

h = cost to hold one unit of inventory for one period

X_t = quantity produced during period t (assumed available to satisfy demand at end of period t)

S_t = quantity sold during period t (we assume that units produced in t are available for sale in t and thereafter)

I_t = inventory at end of period t (after demand has been met); we assume I_0 is given as data

In this notation, X_t, S_t, and I_t are **decision variables.** That is, the computer program solving the LP is free to choose their values so as to minimize the objective, provided the constraints are satisfied. The other variables—d_t, c_t, r, h—are **constants,** which must be estimated for the actual system and supplied as data. Throughout this chapter, we use the convention of representing variables with capital letters and constants with lowercase letters.

We can represent the problem of maximizing net profit minus inventory carrying cost subject to capacity and demand constraints as

$$\text{Maximize} \quad \sum_{t=1}^{\bar{t}} rS_t - hI_t \tag{16.1}$$

Subject to:

$$S_t \le d_t \qquad t = 1, \ldots, \bar{t} \tag{16.2}$$

$$X_t \le c_t \qquad t = 1, \ldots, \bar{t} \tag{16.3}$$

$$I_t = I_{t-1} + X_t - S_t \qquad t = 1, \ldots, \bar{t} \tag{16.4}$$

$$X_t, S_t, I_t \ge 0 \qquad t = 1, \ldots, \bar{t} \tag{16.5}$$

The objective function computes net profit by multiplying unit profit r by sales S_t in each period t, and subtracting the inventory carrying cost h times remaining inventory I_t at the end of period t, and summing over all periods in the planning horizon. Constraints (16.2) limit sales to demand. If possible, the computer will make all these constraints tight, since increasing the S_t values increases the objective function. The only reason that these constraints will not be tight in the optimal solution is that capacity constraints (16.3) will not permit it.[1] Constraints (16.4), which are of a form common to almost all multiperiod aggregate planning models, are known as **balance constraints.** Physically, all they represent is conservation of material; the inventory at the end of period $t\,(I_t)$ is equal to the inventory at the end of period $t-1\,(I_{t-1})$ plus what was produced during period $t\,(X_t)$ minus the amount sold in period t (S_t). These constraints are what force the computer to choose values for X_t, S_t, and I_t that are consistent with our verbal definitions of them. Constraints (16.5) are simple nonnegativity constraints, which rule out negative production or inventory levels. Many, but not all, computer packages for solving LPs automatically force decision variables to be nonnegative unless the user specifies otherwise.

16.2.2 An LP Example

To make the above formulation concrete and to illustrate the mechanics of solving it via linear programming, we now consider a simple example. The Excel spreadsheet shown in Figure 16.1 contains the unit profit r of \$10, the one-period unit holding cost h of \$1, the initial inventory I_0 of 0, and capacity and demand data c_t and d_t for the next six months. We will make use of the rest of the spreadsheet in Figure 16.1 momentarily. For now, we can express LP (16.1)–(16.5) for this specific case as

$$\text{Maximize}\quad 10(S_1+S_2+S_3+S_4+S_5+S_6)-1(I_1+I_2+I_3+I_4+I_5+I_6) \tag{16.6}$$

Subject to:

Demand constraints

$$S_1 \leq 80 \tag{16.7}$$
$$S_2 \leq 100 \tag{16.8}$$
$$S_3 \leq 120 \tag{16.9}$$
$$S_4 \leq 140 \tag{16.10}$$
$$S_5 \leq 90 \tag{16.11}$$
$$S_6 \leq 140 \tag{16.12}$$

Capacity constraints

$$X_1 \leq 100 \tag{16.13}$$
$$X_2 \leq 100 \tag{16.14}$$
$$X_3 \leq 100 \tag{16.15}$$
$$X_4 \leq 120 \tag{16.16}$$
$$X_5 \leq 120 \tag{16.17}$$
$$X_6 \leq 120 \tag{16.18}$$

[1] If we want to consider demand as inviolable, we could remove constraints (16.2) and replace S_t with d_t in the objective and constraints (16.4). The problem with this, however, is that if demand is capacity-infeasible, the computer will just come back with a message saying "infeasible," which doesn't tell us why. The formulation here will be feasible regardless of demand; it simply won't make sales equal to demand if there is not enough capacity, and thus we will know what demand we are incapable of meeting from the solution.

FIGURE 16.1

Input spreadsheet for linear programming example

	A	B	C	D	E	F	G	H
1	**Constants:**							
2	**r**	10						
3	**h**	1						
4	**I_O**	0						
5	**t**	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
6	**c_t**	100	100	100	120	120	120	660
7	**d_t**	80	100	120	140	90	140	670
8								
9	**Variables:**							
10	**t**	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
11	**X_t**	0	0	0	0	0	0	0
12	**S_t**	0	0	0	0	0	0	0
13	**I_t**	0	0	0	0	0	0	0
14								
15	**Objective:**							
16	**Net Profit:**	$0		r*(S_1+S_2+S_3+S_4+S_5+S_6) - h*(I_1+I_2+I_3+I_4+I_5+I_6)				
17								
18	**Constraints:**							
19	**S_1**	0	<=	80	**d_1**			
20	**S_2**	0	<=	100	**d_2**			
21	**S_3**	0	<=	120	**d_3**			
22	**S_4**	0	<=	140	**d_4**			
23	**S_5**	0	<=	90	**d_5**			
24	**S_6**	0	<=	140	**d_6**			
25	**X_1**	0	<=	100	**c_1**			
26	**X_2**	0	<=	100	**c_2**			
27	**X_3**	0	<=	100	**c_3**			
28	**X_4**	0	<=	120	**c_4**			
29	**X_5**	0	<=	120	**c_5**			
30	**X_6**	0	<=	120	**c_6**			
31	**I_1-I_0-X_1+S_1**	0	=	0				
32	**I_2-I_1-X_2+S_2**	0	=	0				
33	**I_3-I_2-X_3+S_3**	0	=	0				
34	**I_4-I_3-X_4+S_4**	0	=	0				
35	**I_5-I_4-X_5+S_5**	0	=	0				
36	**I_6-I_5-X_6+S_6**	0	=	0				
37					**Note: X_t, S_t and I_t must be >= 0**			

Inventory balance constraints

$$I_1 - X_1 + S_1 = 0 \quad (16.19)$$

$$I_2 - I_1 - X_2 + S_2 = 0 \quad (16.20)$$

$$I_3 - I_2 - X_3 + S_3 = 0 \quad (16.21)$$

$$I_4 - I_3 - X_4 + S_4 = 0 \quad (16.22)$$

$$I_5 - I_4 - X_5 + S_5 = 0 \quad (16.23)$$

$$I_6 - I_5 - X_6 + S_6 = 0 \quad (16.24)$$

Nonnegativity constraints

$$X_1, X_2, X_3, X_4, X_5, X_6 \geq 0 \quad (16.25)$$

$$S_1, S_2, S_3, S_4, S_5, S_6 \geq 0 \quad (16.26)$$

$$I_1, I_2, I_3, I_4, I_5, I_6 \geq 0 \quad (16.27)$$

Some linear programming packages allow entry of a problem formulation in a format almost identical to (16.6) to (16.27) via a text editor. While this is certainly convenient for very small problems, it can become prohibitively tedious for large ones. Because of this, there is considerable work going on in the OM research community to develop **modeling languages** that provide user-friendly interfaces for describing large-scale optimization problems (see Fourer, Gay, and Kernighan 1993 for an excellent example of a modeling language). Conveniently for us, LP is becoming so prevalent that our spreadsheet package, Microsoft Excel, has an LP solver built right into it. We can represent and solve formulation (16.6) to (16.27) right in the spreadsheet shown in Figure 16.1. The following technical note provides details on how to do this.

Technical Note—Using the Excel LP Solver

Although the reader should consult the Excel documentation for details about the release in use, we will provide a brief overview of the LP solver in Excel 5.0. The first step is to establish cells for the decision variables (B11:G13 in Figure 16.1). We have initially entered zeros for these, but we can set them to be anything we like; thus, we could start by setting $X_t = d_t$, which would be closer to an optimal solution than zeros. The spreadsheet is a good place to play what-if games with the data. However, eventually we will turn over the problem of finding optimal values for the decision variables to the LP solver. Notice that for convenience we have also entered a column that totals X_t, S_t, and I_t. For example, cell H11 contains a formula to sum cells B11:G11. This allows us to write the objective function more compactly.

Once we have specified decision variables, we construct an objective function in cell B16. We do this by writing a formula that multiplies r (cell B2) by total sales (cell H12) and then subtracts the product of h (cell B3) and total inventory (cell H13). Since all the decision variables are zero at present, this formula also returns a zero; that is, the net profit on no production with no initial inventory is zero.

Next we need to specify the constraints (16.7) to (16.27). To do this, we need to develop formulas that compute the left-hand side of each constraint. For constraints (16.7) to (16.18) we really do not need to do this, since the left-hand sides are only X_t and S_t and we already have cells for these in the variables portion of the spreadsheet. However, for clarity, we will copy them to cells B19:B30. We will not do the same for the nonnegativity constraints (16.25) to (16.27), since it is a simple matter to choose all the decision variables and force them to be greater than or equal to zero in the Excel Solver menu. Constraints (16.19) to (16.24) require us to do work, since the left-hand sides are formulas of multiple variables. For instance, cell B31 contains a formula to compute $I_1 - I_0 - X_1 + S_1$ (that is, B13 − B4 − B11 + B12). We have given these cells names to remind us of what they represent, although any names could be used, since they are not necessary for the computation. We have also copied the values of the right-hand sides of the constraints into cells D19:D36 and labeled them in column E for clarity. This is not strictly necessary, but does make it easier to specify constraints in the Excel Solver, since whole blocks of constraints can be specified (for example, B19:B30 $\leq$ D19:D30). The equality and inequality symbols in column C are also unnecessary, but make the formulation easier to read.

To use the Excel LP Solver, we choose **Formula/Solver** from the menu. In the dialog box that comes up (see Figure 16.2), we specify the cells containing the objective, choose to maximize or minimize, and specify the cells containing decision variables (this can be done by pointing with the mouse). Then we add constraints by choosing **Add** from the constraints section of the form. Another dialog box (see Figure 16.3) comes up in which we fill in the cell containing the left-hand side of the constraint, choose the relationship ($\geq$, $\leq$, or $=$), and fill in the right-hand side.

Note that the actual constraint is not shown explicitly in the spreadsheet; it is entered only in the **Solver** menu. However, the right-hand side of the constraint can be another cell in the spreadsheet or a constant. By specifying a range of cells for the right-hand side and a constant for the left-hand side, we can add a whole set of constraints in a single command. For instance, the range B11:G13 represents all the decision variables, so if we use this range as the left-hand side, a $\geq$ symbol, and a zero for the right-hand side, we will represent all the nonnegativity constraints (16.25) to (16.27). By choosing the **Add** button after each constraint we enter, we can add all the model constraints. When we are done, we choose the **OK** button, which returns us to the original form. We have the option to edit or delete constraints at any time.

Finally, before running the model, we must tell Excel that we want it to use the LP solution algorithm.[2] We do this by choosing the **Options** button to bring up another dialog box (see Figure 16.4) and choosing the **Assume Linear Model** option. This form also allows us to limit the time the model will run and to specify certain tolerances. If the model does not

[2]Excel can also solve nonlinear optimization problems and will apply the nonlinear algorithm as a default. Since LP is *much* more efficient, we definitely want to choose it as long as our model meets the requirements. All the formulations in this chapter are linear and therefore can use LP.

Figure 16.2

Specification of objectives and constraints in Excel

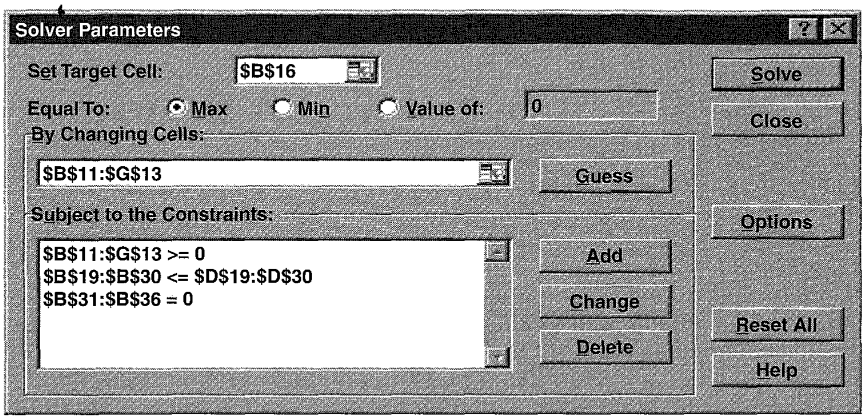


Figure 16.3

Add constraint dialog box in Excel

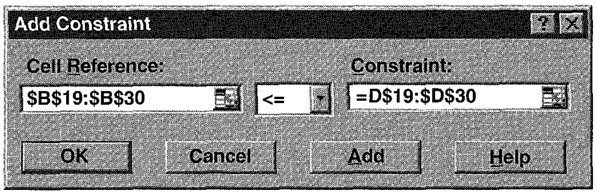
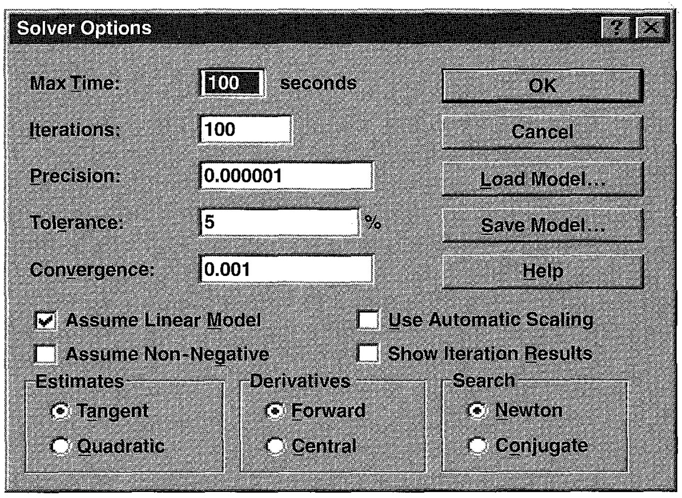


Figure 16.4

Setting Excel to use linear programming



converge to an answer, the most likely reason is an error in one of the constraints. However, sometimes increasing the search time or reducing tolerances will fix the problem when the solver cannot find a solution. The reader should consult the Excel manual for more detailed documentation on this and other features, as well as information on upgrades that may have occurred since this writing. Choosing the **OK** button returns us to the original form.

Once we have done all this, we are ready to run the model by choosing the **Solve** button. The program will pause to set up the problem in the proper format and then will go through a sequence of trial solutions (although not for long in such a small problem as this).

Basically, LP works by first finding a feasible solution—one that satisfies all the constraints—and then generating a succession of new solutions, each better than the last. When no further improvement is possible, it stops and the solution is optimal: It maximizes or minimizes the objective function. Appendix 16A provides background on how this process works.

The algorithm will stop with one of three answers:

1. *Could not find a feasible solution.* This probably means that the problem is infeasible; that is, there is no solution that satisfies all the constraints. This could be due to a typing error (e.g., a plus sign was incorrectly typed as a minus sign) or a real infeasibility (e.g., it is not possible to meet demand with capacity). Notice that by clever formulation, one can avoid having the algorithm terminate with this depressing message when real infeasibilities exist. For instance, in formulation (16.6) to (16.27), we did not force sales to be equal to demand. Since cumulative demand exceeds cumulative capacity, it is obvious that this would not have been feasible. By setting separate sales and production variables, we let the computer tell us where demand cannot be met. Many variations on this trick are possible.

2. *Does not converge.* This means either that the algorithm could not find an optimal solution within the allotted time (so increasing the time or decreasing the tolerances under the **Options** menu might help) or that the algorithm is able to continue finding better and better solutions indefinitely. This second possibility can occur when the problem is **unbounded:** The objective can be driven to infinity by letting some variables grow positive or negative without bound. Usually this is the result of a failure to properly constrain a decision variable. For instance, in the above model, if we forgot to specify that all decision variables must be nonnegative, then the model will be able to make the objective arbitrarily large by choosing negative values of $I_t, t = 1, \ldots, 6$. Of course, we do not generate revenue via negative inventory levels, so it is important that nonnegativity constraints be included to rule out this nonsensical behavior.[3]

3. *Found a solution.* This is the outcome we want. When it occurs, the program will write the optimal values of the decision variables, objective value, and constraints into the spreadsheet. Figure 16.5 shows the spreadsheet as modified by the LP algorithm. The program also offers three reports—Answer, Sensitivity, and Limits—which write information about the solution into other spreadsheets. For instance, highlighting the Answer report generates a spreadsheet with the information shown in Figures 16.6 and 16.7. Figure 16.8 contains some of the information contained in the report generated by choosing Sensitivity.

Now that we have generated a solution, let us interpret it. Both Figure 16.5—the final spreadsheet—and Figure 16.6 show the optimal decision variables. From these we see that it is not optimal to produce at full capacity in every period. Specifically, the solution calls for producing only 110 units in month 5 when capacity is 120. This might seem odd given that demand exceeds capacity. However, if we look more carefully, we see that cumulative demand for periods 1 to 4 is 440 units, while cumulative capacity

[3]We will show how to modify the formulation to allow for backordering, which is like allowing negative inventory positions, without this inappropriately affecting the objective function, later in this chapter.

FIGURE 16.5

Output spreadsheet for LP example

	A	B	C	D	E	F	G	H
1	**Constants:**							
2	r	10						
3	h	1						
4	I_0	0						
5	t	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
6	c_t	100	100	100	120	120	120	660
7	d_t	80	100	120	140	90	140	670
8								
9	**Variables:**							
10	t	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
11	X_t	100	100	100	120	110	120	650
12	S_t	80	100	120	120	90	140	650
13	I_t	20	20	0	0	20	0	60
14								
15	**Objective:**							
16	**Net Profit:**	$6,440		r*(S_1+S_2+S_3+S_4+S_5+S_6) - h*(I_1+I_2+I_3+I_4+I_5+I_6)				
17								
18	**Constraints:**							
19	S_1	80	<=	80	**d_1**			
20	S_2	100	<=	100	**d_2**			
21	S_3	120	<=	120	**d_3**			
22	S_4	120	<=	140	**d_4**			
23	S_5	90	<=	90	**d_5**			
24	S_6	140	<=	140	**d_6**			
25	X_1	100	<=	100	**c_1**			
26	X_2	100	<=	100	**c_2**			
27	X_3	100	<=	100	**c_3**			
28	X_4	120	<=	120	**c_4**			
29	X_5	110	<=	120	**c_5**			
30	X_6	120	<=	120	**c_6**			
31	I_1-I_0-X_1+S_1	0	=	0				
32	I_2-I_1-X_2+S_2	0	=	0				
33	I_3-I_2-X_3+S_3	0	=	0				
34	I_4-I_3-X_4+S_4	0	=	0				
35	I_5-I_4-X_5+S_5	0	=	0				
36	I_6-I_5-X_6+S_6	0	=	0				
37					**Note: X_t, S_t and I_t must be >= 0**			

for those periods is only 420 units. Thus, even when we run flat out for the first four months, we will fall short of meeting demand by 20 units. Demand in the final two months is only 230 units, while capacity is 240 units. Since our model does not permit backordering, it does not make sense to produce more than 230 units in months 5 and 6. Any extra units cannot be used to make up a previous shortfall.

Figure 16.7 gives more details on the constraints by showing which ones are **binding** or **tight** (i.e., equal to the right-hand side) and which ones are **nonbinding** or **slack,** and by how much. Most interesting are the constraints on sales, given in (16.7) to (16.12), and capacity, in (16.13) to (16.18). As we have already noted, the capacity constraint on X_5 is nonbinding. Since we only produce 110 units in month 5 and have capacity for 120, this constraint is slack by 10 units. This means that if we changed this constraint by a little (e.g., reduced capacity in month 5 from 120 to 119 units), it would not change the optimal solution at all.

In this same vein, all sales constraints are tight except that for S_4. Since sales are limited to 140, but optimal sales are 120, this constraint has slackness of 20 units. Again, if we were to change this sales constraint by a little (e.g., limit sales to 141 units), the optimal solution would remain the same.

In contrast with these slack constraints, consider a binding constraint. For instance, consider the capacity constraint on X_1, which is the seventh one shown in Figure 16.7. Since the model chooses production equal to capacity in month 1, this constraint is tight. If we were to change this constraint by increasing or decreasing capacity, the solution would change. If we **relax** the constraint by increasing capacity, say, to 101 units, then we will be able to satisfy an additional unit of demand and therefore the net profit will

FIGURE 16.6

Optimal values report for LP example

Microsoft Excel 5.0 Answer Report
Worksheet: [BASICAP.XLS]Figure 16.5
Report Created: 5/15/95 12:22

Target Cell (Max)

Cell	Name	Original Value	Final Value
$B$16	Net Profit	0	6440

Adjustable Cells

Cell	Name	Original Value	Final Value
$B$12	S_1	0	80
$C$12	S_2	0	100
$D$12	S_3	0	120
$E$12	S_4	0	120
$F$12	S_5	0	90
$G$12	S_6	0	140
$B$11	X_1	0	100
$C$11	X_2	0	100
$D$11	X_3	0	100
$E$11	X_4	0	120
$F$11	X_5	0	110
$G$11	X_6	0	120
$B$13	I_1	0	20
$C$13	I_2	0	20
$D$13	I_3	0	0
$E$13	I_4	0	0
$F$13	I_5	0	20
$G$13	I_6	0	0

FIGURE 16.7

Optimal constraint status for LP example

Microsoft Excel 5.0 Answer Report
Worksheet: [BASICAP.XLS]Figure 16.5
Report Created: 5/15/95 12:22

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$B$19	S_1	80	$B$19<=$D$19	Binding	0
$B$20	S_2	100	$B$20<=$D$20	Binding	0
$B$21	S_3	120	$B$21<=$D$21	Binding	0
$B$22	S_4	120	$B$22<=$D$22	Not Binding	20
$B$23	S_5	90	$B$23<=$D$23	Binding	0
$B$24	S_6	140	$B$24<=$D$24	Binding	0
$B$25	X_1	100	$B$25<=$D$25	Binding	0
$B$26	X_2	100	$B$26<=$D$26	Binding	0
$B$27	X_3	100	$B$27<=$D$27	Binding	0
$B$28	X_4	120	$B$28<=$D$28	Binding	0
$B$29	X_5	110	$B$29<=$D$29	Not Binding	10
$B$30	X_6	120	$B$30<=$D$30	Binding	0
$B$31	I_1-I_0-X_1+S_1	0	$B$31=0	Binding	0
$B$32	I_2-I_1-X_2+S_2	0	$B$32=0	Binding	0
$B$33	I_3-I_2-X_3+S_3	0	$B$33=0	Binding	0
$B$34	I_4-I_3-X_4+S_4	0	$B$34=0	Binding	0
$B$35	I_5-I_4-X_5+S_5	0	$B$35=0	Binding	0
$B$36	I_6-I_5-X_6+S_6	0	$B$36=0	Binding	0
$B$12	S_1	80	$B$12>=0	Not Binding	80
$C$12	S_2	100	$C$12>=0	Not Binding	100
$D$12	S_3	120	$D$12>=0	Not Binding	120
$E$12	S_4	120	$E$12>=0	Not Binding	120
$F$12	S_5	90	$F$12>=0	Not Binding	90
$G$12	S_6	140	$G$12>=0	Not Binding	140
$B$11	X_1	100	$B$11>=0	Not Binding	100
$C$11	X_2	100	$C$11>=0	Not Binding	100
$D$11	X_3	100	$D$11>=0	Not Binding	100
$E$11	X_4	120	$E$11>=0	Not Binding	120
$F$11	X_5	110	$F$11>=0	Not Binding	110
$G$11	X_6	120	$G$11>=0	Not Binding	120
$B$13	I_1	20	$B$13>=0	Not Binding	20
$C$13	I_2	20	$C$13>=0	Not Binding	20
$D$13	I_3	0	$D$13>=0	Binding	0
$E$13	I_4	0	$E$13>=0	Binding	0
$F$13	I_5	20	$F$13>=0	Not Binding	20
$G$13	I_6	0	$G$13>=0	Binding	0

increase. Since we will produce the extra item in month 1, hold it for three months to month 4 at a cost of $1 per month, and then sell it for $10, the overall increase in the objective from this change will be $\$10 - 3 = \$7$. Conversely, if we **tighten** the constraint by decreasing capacity, say to 99 units, then we will only be able to carry 19 units from month 1 to month 3 and will therefore lose one unit of demand in month 3. The loss in net profit from this unit will be $8 ($10 − $2 for two months' holding).

The sensitivity data generated by the LP algorithm shown in Figure 16.8 gives us more direct information on the sensitivity of the final solution to changes in the constraints. This report has a line for every constraint in the model and reports three important pieces of information:[4]

1. The **shadow price** represents the amount the optimal objective will be increased by a unit increase in the right-hand side of the constraint.
2. The **allowable increase** represents the amount by which the right-hand side can be increased before the shadow price no longer applies.
3. The **allowable decrease** represents the amount by which the right-hand side can be decreased before the shadow price no longer applies.

Appendix 16A gives a geometric explanation of how these numbers are computed.

[4]The report also contains sensitivity information about the coefficients in the objective function. See Appendix 16A for a discussion of this.

FIGURE 16.8

Sensitivity analysis for LP example

Microsoft Excel 5.0 Sensitivity Report
Worksheet: [BASICAP.XLS]Figure 16.5
Report Created: 5/15/95 12:22

Changing Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$B$12	S_1	80	0	10	1E+30	3
$C$12	S_2	100	0	10	1E+30	2
$D$12	S_3	120	0	10	1E+30	1
$E$12	S_4	120	0	10	1	7
$F$12	S_5	90	0	10	1E+30	10
$G$12	S_6	140	0	10	1E+30	9
$B$11	X_1	100	0	0	1E+30	7
$C$11	X_2	100	0	0	1E+30	8
$D$11	X_3	100	0	0	1E+30	9
$E$11	X_4	120	0	0	1E+30	10
$F$11	X_5	110	0	0	1	9
$G$11	X_6	120	0	0	1E+30	1
$B$13	I_1	20	0	-1	3	7
$C$13	I_2	20	0	-1	2	7
$D$13	I_3	0	0	-1	1	7
$E$13	I_4	0	-11	-1	11	1E+30
$F$13	I_5	20	0	-1	1	9
$G$13	I_6	0	-2	-1	2	1E+30

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$B$19	S_1	80	3	80	0	20
$B$20	S_2	100	2	100	0	20
$B$21	S_3	120	1	120	0	20
$B$22	S_4	120	0	140	1E+30	20
$B$23	S_5	90	10	90	10	90
$B$24	S_6	140	9	140	10	20
$B$25	X_1	100	7	100	20	0
$B$26	X_2	100	8	100	20	0
$B$27	X_3	100	9	100	20	0
$B$28	X_4	120	10	120	20	120
$B$29	X_5	110	0	120	1E+30	10
$B$30	X_6	120	1	120	20	10
$B$31	I_1-I_0-X_1+S_1	0	7	0	20	0
$B$32	I_2-I_1-X_2+S_2	0	8	0	20	0
$B$33	I_3-I_2-X_3+S_3	0	9	0	20	0
$B$34	I_4-I_3-X_4+S_4	0	10	0	20	120
$B$35	I_5-I_4-X_5+S_5	0	0	0	110	10
$B$36	I_6-I_5-X_6+S_6	0	1	0	20	10

To see how these data are interpreted, consider the information in Figure 16.8 on the seventh line of the constraint section for the capacity constraint $X_1 \leq 100$. The shadow price is $7, which means that if the constraint is changed to $X_1 \leq 101$, net profit will increase by $7, precisely as we computed above. The allowable increase is 20 units, which means that each unit capacity increase in period 1 up to a total of 20 units increases net profit by $7. Therefore, an increase in capacity from 100 to 120 will increase net profit by $20 \times 7 = \$140$. Above 20 units, we will have satisfied all the lost demand in month 4, and therefore further increases will not improve profit. Thus, this constraint will become nonbinding once the right-hand side exceeds 120. Notice that the allowable decrease is zero for this constraint. What this means is that the shadow price of $7 is not valid for decreases in the right-hand side. As we computed above, the decrease in net profit from a unit decrease in the capacity in month 1 is $8. In general, we can only determine the impact of changes outside the allowable increase or decrease range by actually changing the constraints and rerunning the LP solver.

The above examples are illustrative of the following general behavior of linear programming models:

1. Changing the right-hand sides of nonbinding constraints by a small amount does not affect the optimal solution. The shadow price of a nonbinding constraint is always zero.

2. Increasing the right-hand side of a binding constraint will increase the objective by an amount equal to the shadow price times the size of the increase, provided that the increase is smaller than the allowable increase.
3. Decreasing the right-hand side of a binding constraint will decrease the objective by an amount equal to the shadow price times the size of the decrease, provided that the decrease is smaller than the allowable decrease.
4. Changes in the right-hand sides beyond the allowable increase or decrease range have an indeterminate effect and must be evaluated by resolving the modified model.
5. All these sensitivity results apply to changes in *one right-hand side variable at a time*. If multiple changes are made, the effects are not necessarily additive. Generally, multiple-variable sensitivity analysis must be done by resolving the model under the multiple changes.

16.3 Product Mix Planning

Now that we have set up the basic framework for formulating and solving aggregate planning problems, we can examine some commonly encountered situations. The first realistic aggregate planning issue we will consider is that of product mix planning. To do this, we need to extend the model of the previous section to consider multiple products explicitly. As mentioned previously, allowing multiple products raises the possibility of a "floating bottleneck." That is, if the different products require different amounts of processing time on the various workstations, then the workstation that is most heavily loaded during a period may well depend on the mix of products run during that period. If flexibility in the mix is possible, we can use the AP module to adjust the mix in accordance with available capacity. And if the mix is essentially fixed, we can use the AP module to identify bottlenecks.

16.3.1 Basic Model

We start with a direct extension of the previous single-product model in which demands are assumed fixed and the objective is to minimize the inventory carrying cost of meeting these demands. To do this, we introduce the following notation:

i = an index of product, $i = 1, \ldots, m$, so m represents total number of products
j = an index of workstation, $j = 1, \ldots, n$, so n represents total number of workstations
t = an index of period, $t = 1, \ldots, \bar{t}$, so $\bar{t}$ represents planning horizon
$\bar{d}_{it}$ = maximum demand for product i in period t
$\underline{d}_{it}$ = minimum sales[5] allows of product i in period t
a_{ij} = time required on workstation j to produce one unit of product i
c_{jt} = capacity of workstation j in period t in units consistent with those used to define a_{ij}
r_i = net profit from one unit of product i
h_i = cost[6] to hold one unit of product i for one period t

[5]This might represent firm commitments that we do not want the computer program to violate.
[6]It is common to set h_i equal to the raw materials cost of product i times a one-period interest rate to represent the opportunity cost of the money tied up in inventory; but it may make sense to use higher values to penalize inventory that causes long, uncompetitive cycle times.

X_{it} = amount of product i produced in period t
S_{it} = amount of product i sold in period t
I_{it} = inventory of product i at end of period t (I_{i0} is given as data)

Again, X_{it}, S_{it}, and I_{it} are decision variables, while the other symbols are constants representing input data. We can give a linear program formulation of the problem to maximize net profit minus inventory carrying cost subject to upper and lower bounds on sales and capacity constraints as

$$\text{Maximize} \quad \sum_{t=1}^{\bar{t}} \sum_{i=1}^{m} r_i S_{it} - h_i I_{it} \tag{16.28}$$

Subject to:

$$\underline{d}_{it} \leq S_{it} \leq \bar{d}_{it} \qquad \text{for all } i, t \tag{16.29}$$

$$\sum_{i=1}^{m} a_{ij} X_{it} \leq c_{jt} \qquad \text{for all } j, t \tag{16.30}$$

$$I_{it} = I_{it-1} + X_{it} - S_{it} \qquad \text{for all } i, t \tag{16.31}$$

$$X_{it}, S_{it}, I_{it} \geq 0 \qquad \text{for all } i, t \tag{16.32}$$

In comparison to the previous single-product model, we have adjusted constraints (16.29) to include lower, as well as upper, bounds on sales. For instance, the firm may have long-term contracts that obligate it to produce certain minimum amounts of certain products. Conversely, the market for some products may be limited. To maximize profit, the computer has incentive to set production so that all these constraints will be tight at their upper limits. However, this may not be possible due to capacity constraints (16.30). Notice that unlike in the previous formulation, we now have capacity constraints for each workstation in each period. By noting which of these constraints are tight, we can identify those resources that limit production. Constraints (16.31) are the multiproduct version of the balance equations, and constraints (16.32) are the usual nonnegativity constraints.

We can use LP (16.28)–(16.32) to obtain several pieces of information, including

1. **Demand feasibility.** We can determine whether a set of demands is capacity-feasible. If the constraint $S_{it} \leq \bar{d}_{it}$ is tight, then the upper bound on demand $\bar{d}_{it}$ is feasible. If not, then it is capacity-infeasible. If demands given by the lower bounds on demand $\underline{d}_{it}$ are capacity-infeasible, then the computer program will return a "could not find a feasible solution" message and the user must make changes (e.g., reduce demands or increase capacity) in order to get a solution.
2. **Bottleneck locations.** Constraints (16.30) restrict production on each workstation in each period. By noting which of these constraints are binding, we can determine which workstations limit capacity in which periods. A workstation that is consistently binding in many periods is a clear bottleneck and requires close management attention.
3. **Product mix.** If we are unable, for capacity reasons, to attain all the upper bounds on demand, then the computer will reduce sales below their maximum for some products. It will try to maximize revenue by producing those products with high net profit, but because of the capacity constraints, this is not a simple matter, as we will see in the following example.

16.3.2 A Simple Example

Let us consider a simple product mix example that shows why one needs a formal optimization method instead of a simpler ad hoc approach for these problems. We simplify matters by assuming a planning horizon of only one period. While this is certainly not a realistic assumption in general, in situations where we know in advance that we will never carry inventory from one period to the next, solving separate one-period problems for each period *will* yield the optimal solution. For example, if demands and cost coefficients are constant from period to period, then there is no incentive to build up inventory and therefore this will be the case.

Consider a situation in which a firm produces two products, which we will call products 1 and 2. Table 16.1 gives descriptive data for these two products. In addition to the direct raw material costs associated with each product, we assume a \$5,000 per week fixed cost for labor and capital. Furthermore, there are 2,400 minutes (five days per week, eight hours per day) of time available on workstations A to D. We assume that all these data are identical from week to week. Therefore, there is no reason to build inventory in one week to sell in a subsequent week. (If we can meet maximum demand this week with this week's production, then the same thing is possible next week.) Thus, we can restrict our attention to a single week, and the only issue is the appropriate amount of each product to produce.

A Cost Approach. Let us begin by looking at this problem from a simple cost standpoint. Net profit per unit of product 1 sold is \$45 (\$90 − 45), while net profit per unit of product 2 sold is \$60 (\$100 − 40). This would seem to indicate that we should emphasize production of product 2. Ideally, we would like to produce 50 units of product 2 to meet maximum demand, but we must check the capacity of the four workstations to make sure this is possible. Since workstation B requires the most time to make a unit of product 2 (30 minutes) among the four workstations, this is the potential constraint. Producing 50 units of product 2 on workstation B will require

$$30 \text{ minutes per unit} \times 50 \text{ units} = 1{,}500 \text{ minutes}$$

This is less than the available 2,400 minutes on workstation B, so producing 50 units of product 2 is feasible.

Now we need to determine how many units of product 1 we can produce with the leftover capacity. The unused time on workstations A to D after subtracting the time to

TABLE 16.1 Input Data for Single-Period AP Example

Product	1	2
Selling price	\$90	\$100
Raw material cost	\$45	\$40
Maximum weekly sales	100	50
Minutes per unit on workstation A	15	10
Minutes per unit on workstation B	15	30
Minutes per unit on workstation C	15	5
Minutes per unit on workstation D	15	5

make 50 units of product 2 we compute as

$$2{,}400 - 10(50) = 1{,}900 \text{ minutes on workstation A}$$
$$2{,}400 - 30(50) = 900 \text{ minutes on workstation B}$$
$$2{,}400 - 5(50) = 2{,}150 \text{ minutes on workstation C}$$
$$2{,}400 - 5(50) = 2{,}150 \text{ minutes on workstation D}$$

Since one unit of product 1 requires 15 minutes of time on each of the four workstations, we can compute the maximum possible production of product 1 at each workstation by dividing the unused time by 15. Since workstation B has the least remaining time, it is the potential bottleneck. The maximum production of product 1 on workstation B (after subtracting the time to produce 50 units of product 2) is

$$\frac{900}{15} = 60$$

Thus, even though we can sell 100 units of product 1, we only have capacity for 60.

The weekly profit from making 60 units of product 1 and 50 units of product 2 is

$$\$45 \times 60 + \$60 \times 50 - \$5{,}000 = \$700$$

Is this the best we can do?

A Bottleneck Approach. The preceding analysis is entirely premised on costs and considers capacity only as an afterthought. A better method might be to look at cost *and* capacity, by computing a ratio representing *profit per minute of bottleneck time used* for each product. This requires that we first identify the bottleneck, which we do by computing the minutes required on each workstation to satisfy maximum demand and seeing which machine is most overloaded.[7] This yields

$$15(100) + 10(50) = 2{,}000 \text{ minutes on workstation A}$$
$$15(100) + 30(50) = 3{,}000 \text{ minutes on workstation B}$$
$$15(100) + 5(50) = 1{,}750 \text{ minutes on workstation C}$$
$$15(100) + 5(50) = 1{,}750 \text{ minutes on workstation D}$$

Only workstation B requires more than the available 2,400 minutes, so we designate it the bottleneck. Hence, we would like to make the most profitable use of our time on workstation B. To determine which of the two products does this, we compute the ratio of net profit to minutes on workstation B as

$$\frac{\$45}{15} = \$3 \text{ per minute spent processing product 1}$$
$$\frac{\$60}{30} = \$2 \text{ per minute spent processing product 2}$$

This calculation indicates the reverse of our previous cost analysis. Each minute spent processing product 1 on workstation B nets us \$3, as opposed to only \$2 per minute spent on product 2. Therefore, we should emphasize production of product 1, not product 2. If we produce 100 units of product 1 (the maximum amount allowed by the demand constraint), then since all workstations require 15 min per unit of one, the unused time on each workstation is

$$2{,}400 - 15(100) = 900 \text{ minutes}$$

[7]The alert reader should be suspicious at this point, since we know that the identity of the "bottleneck" can depend on the product mix in a multiproduct case.

Then since workstation B is the slowest operation for producing product 2, this is what limits the amount we can produce. Each unit of product 2 requires 30 minutes on B; thus, we can produce

$$\frac{900}{30} = 30$$

units of product 2. The net profit from producing 100 units of product 1 and 30 units of product 2 is

$$\$45 \times 100 + \$60 \times 30 - \$5{,}000 = \$1{,}300$$

This is clearly better than the $700 we got from using our original analysis and, it turns out, is the best we can do. But will this method always work?

A Linear Programming Approach. To answer the question of whether the previous "bottleneck ratio" method will always determine the optimal product mix, we consider a slightly modified version of the previous example, with data shown in Table 16.2. The only changes in these data relative to the previous example are that the processing time of product 2 on workstation B has been increased from 30 to 35 minutes and the processing times for products 1 and 2 on workstation D have been increased from 15 and 5 to 25 and 14, respectively.

To execute our ratio-based approach on this modified problem, we first check for the bottleneck by computing the minutes required on each workstation to meet maximum demand levels:

$$\begin{aligned} 15(100) + 10(50) &= 2{,}000 \text{ minutes on workstation A} \\ 15(100) + 35(50) &= 3{,}250 \text{ minutes on workstation B} \\ 15(100) + 5(50) &= 1{,}750 \text{ minutes on workstation C} \\ 25(100) + 14(50) &= 3{,}200 \text{ minutes on workstation D} \end{aligned}$$

Workstation B is still the most heavily loaded resource, but now workstation D also exceeds the available 2,400 minutes.

If we designate workstation B as the bottleneck, then the ratio of net profit to minute of time on the bottleneck is

$$\frac{\$45}{15} = \$3.00 \text{ per minute spent processing product 1}$$

$$\frac{\$60}{35} = \$1.71 \text{ per minute spent processing product 2}$$

TABLE 16.2 Input Data for Modified Single-Period AP Example

Product	1	2
Selling price	$90	$100
Raw material cost	$45	$40
Maximum weekly sales	100	50
Minutes per unit on workstation A	15	10
Minutes per unit on workstation B	15	35
Minutes per unit on workstation C	15	5
Minutes per unit on workstation D	25	14

which, as before, indicates that we should produce as much product 1 as possible. However, now it is workstation D that is slowest for product 1. The maximum amount that can be produced on D in 2,400 minutes is

$$\frac{2{,}400}{25} = 96$$

Since 96 units of product 1 use up all available time on workstation D, we cannot produce any product 2. The net profit from this mix, therefore, is

$$\$45 \times 96 - \$5{,}000 = -\$680$$

This doesn't look very good—we are losing money. Moreover, while we used workstation B as our bottleneck for the purpose of computing our ratios, it was workstation D that determined how much product we could produce. Therefore, perhaps we should have designated workstation D as our bottleneck. If we do this, the ratio of net profit to minute of time on the bottleneck is

$$\frac{\$45}{25} = \$1.80 \text{ per minute spent processing product 1}$$

$$\frac{\$60}{14} = \$4.29 \text{ per minute spent processing product 2}$$

This indicates that it is more profitable to emphasize production of product 2. Since workstation B is slowest for product 2, we check its capacity to see how much product 2 we can produce, and we find

$$\frac{2{,}400}{35} = 68.57$$

Since this is greater than maximum demand, we should produce the maximum amount of product 2, which is 50 units. Now we compute the unused time on each machine as

$$2{,}400 - 10(50) = 1{,}900 \text{ minutes on workstation A}$$
$$2{,}400 - 35(50) = 650 \text{ minutes on workstation B}$$
$$2{,}400 - 5(50) = 2{,}150 \text{ minutes on workstation C}$$
$$2{,}400 - 14(50) = 1{,}700 \text{ minutes on workstation D}$$

Dividing the unused time by the minutes required to produce one unit of product 1 on each workstation gives us the maximum production of product 1 on each to be

$$\frac{1{,}900}{15} = 126.67 \text{ units on workstation A}$$

$$\frac{650}{15} = 43.33 \text{ units on workstation B}$$

$$\frac{2{,}150}{15} = 143.33 \text{ units on workstation C}$$

$$\frac{1{,}700}{25} = 68 \text{ units on workstation D}$$

Thus, workstation B limits production of product 1 to 43 units, so total net profit for this solution is

$$\$45 \times 43 + \$60 \times 50 - \$5{,}000 = -\$65$$

This is better, but we are still losing money. Is this the best we can do?

Finally, let's bring out our big gun (not really that big, since it is included in popular spreadsheet programs) and solve the problem with a linear programming package.

Letting X_1 (X_2) represent the quantity of product 1 (2) produced, we formulate a linear programming model to maximize profit subject to the demand and capacity constraints as

$$\text{Maximize} \quad 45X_1 + 60X_2 - 5{,}000 \tag{16.33}$$

Subject to:

$$X_1 \leq 100 \tag{16.34}$$
$$X_2 \leq 50 \tag{16.35}$$
$$15X_1 + 10X_2 \leq 2{,}400 \tag{16.36}$$
$$15X_1 + 35X_2 \leq 2{,}400 \tag{16.37}$$
$$15X_1 + 5X_2 \leq 2{,}400 \tag{16.38}$$
$$25X_1 + 14X_2 \leq 2{,}400 \tag{16.39}$$

Problem (16.33)–16.39) is trivial for any LP package. Ours (Excel) reports the solution to this problem to be

$$\text{Optimal objective} = \$557.94$$
$$X_1^* = 75.79$$
$$X_2^* = 36.09$$

Even if we round this solution down (which will certainly still be capacity-feasible, since we are reducing production amounts) to integer values

$$X_1^* = 75$$
$$X_2^* = 36$$

we get an objective of

$$\$45 \times 75 + \$60 \times 36 - \$5{,}000 = \$535$$

So making as much product 1 as possible and making as much product 2 as possible both result in negative profit. But making a *mix* of the two products generates positive profit!

The moral of this exercise is that even simple product mix problems can be subtle. No trick that chooses a dominant product or identifies the bottleneck before knowing the product mix can find the optimal solution in general. While such tricks can work for specific problems, they can result in extremely bad solutions in others. The only method guaranteed to solve these problems optimally is an exact algorithm such as those used in linear programming packages. Given the speed, power, and user-friendliness of modern LP packages, one should have a very good reason to forsake LP for an approximate method.

16.3.3 Extensions to the Basic Model

A host of variations on the basic problem given in formulation (16.28)–(16.32) are possible. We discuss a few of these next; the reader is asked to think of others in the problems at chapter's end.

Other Resource Constraints. Formulation (16.28)–(16.32) contains capacity constraints for the workstations, but not for other resources, such as people, raw materials, and transport devices. In some systems, these may be important determinants of overall capacity and therefore should be included in the AP module.

Generically, if we let

b_{ij} = units of resource j required per unit of product i

k_{jt} = number of units of resource j available in period t

X_{it} = amount of product i produced in period t

we can express the capacity constraint on resource j in period t as

$$\sum_{i=1}^{m} b_{ij} X_{it} \leq k_{jt} \tag{16.40}$$

Notice that b_{ij} and k_{jt} are the nonworkstation analogs to a_{ij} and c_{jt} in formulation (16.28)–(16.32).

As a specific example, suppose an inspector must check products 1, 2, and 3, which require 1, 2, and 1.5 hours, respectively, per unit to inspect. If the inspector is available a total of 160 hours per month, then the constraint on this person's time in month t can be represented as

$$X_{1t} + 2X_{2t} + 1.5X_{3t} \leq 160$$

If this constraint is binding in the optimal solution, it means that inspector time is a bottleneck and perhaps something should be reorganized to remove this bottleneck. (The plant could provide help for the inspector, simplify the inspection procedure to speed it up, or use quality-at-the-source inspections by the workstation operators to eliminate the need for the extra inspection step.)

As a second example, suppose a firm makes four different models of circuit board, all of which require one unit of a particular component. The component contains leading-edge technology and is in short supply. If k_t represents the total number of these components that can be made available in period t, then the constraint represented by component availability in each period t can be expressed as

$$X_{1t} + X_{2t} + X_{3t} + X_{4t} \leq k_t$$

Many other resource constraints can be represented in analogous fashion.

Utilization Matching. As our discussion so far shows, it is straightforward to model capacity constraints in LP formulations of AP problems. However, we must be careful about how we use these constraints in actual practice, for two reasons.

1. *Low-level complexity.* An AP module will necessarily gloss over details that can cause inefficiency in the short term. For instance, in the product mix example of the previous section, we assumed that it was possible to run the four machines 2,400 minutes per week. However, from our factory physics discussions of Part II, we know that it is virtually impossible to avoid some idle time on machines. Any source of randomness (machine failures, setups, errors in the scheduling process, etc.) can diminish utilization. While we cannot incorporate these directly in the AP model, we can account for their aggregate effect on utilization.
2. *Production control decisions.* As we noted in Chapter 13, it may be economically attractive to set the production quota below full average capacity, in order to achieve predictable customer service without excessive overtime costs. If the quota-setting module indicates that we should run at less than full utilization, we should include this fact in the aggregate planning module in order to maintain consistency.

These considerations may make it attractive to plan for production levels below full capacity. Although the decision of how close to capacity to run can be tricky, the mechanics of reducing capacity in the AP model are simple. If the c_{jt} parameters represent practical estimates of realistic full capacity of workstation j in period t, adjusted for setups, worker breaks, machine failures, and other reasonable detractors, then we can simply deflate capacity by multiplying these by a constant factor. For instance, if either historical experience or the quote-setting module indicates that it is reasonable to run at a fraction q of full capacity, then we can replace constraints (16.30) in LP (16.28)–(16.32) by

$$\sum_{i=t}^{m} a_{ij} X_{it} \leq q c_{jt} \qquad \text{for all } j, t$$

The result will be that a binding capacity constraint will occur whenever a workstation is loaded to $100q$ percent of capacity in a period.

Backorders. In LP (16.28)–(16.32), we forced inventory to remain positive at all times. Implicitly, we were assuming that demands had to be met from inventory or lost; no backlogging of unmet demand was allowed. However, in many realistic situations, demand is not lost when not met on time. Customers expect to receive their orders even if they are late. Moreover, it is important to remember that aggregate planning is a long-term planning function. Just because the model says a particular order will be late, that does not mean that this must be so in practice. If the model predicts that an order due nine months from now will be backlogged, there may be ample time to renegotiate the due date. For that matter, the demand may really be only a forecast, to which a firm customer due date has not yet been attached. With this in mind, it makes sense to think of the aggregate planning module as a tool for reconciling projected demands with available capacity. By using it to identify problems that are far in the future, we can address them while there is still time to do something about them.

We can easily modify LP (16.28)–(16.32) to permit backordering as follows:

$$\text{Maximize} \quad \sum_{t=1}^{\bar{t}} r_i S_{it} - h_i I_{it}^{+} - \pi_{it}^{-} \tag{16.41}$$

Subject to:

$$\underline{d}_{it} \leq S_{it} \leq \bar{d}_{it} \qquad \text{for all } i, t \tag{16.42}$$

$$\sum_{i=1}^{m} a_{ij} X_{it} \leq c_{jt} \qquad \text{for all } j, t \tag{16.43}$$

$$I_{it} = I_{it-1} + X_{it} - S_{it} \qquad \text{for all } i, t \tag{16.44}$$

$$I_{it} = I_{it}^{+} - I_{it}^{-} \qquad \text{for all } i, t \tag{16.45}$$

$$X_{it}, S_{it}, I_{it}^{+}, I_{it}^{-} \geq 0 \qquad \text{for all } i, t \tag{16.46}$$

The main change was to redefine the inventory variable I_{it} as the difference $I_{it}^{+} - I_{it}^{-}$, where I_{it}^{+} represents the inventory of product i carried from period t to $t+1$ and I_{it}^{-} represents the number of backorders carried from period t to $t+1$. Both I_{it}^{+} and I_{it}^{-} must be nonnegative. However, I_{it} can be either positive or negative, and so we refer to it as the **inventory position** of product i in period t. A positive inventory position indicates on-hand inventory, while a negative inventory position indicates outstanding backorders. The coefficient π_i is the backorder analog to the holding cost h_i and represents the penalty to carry one unit of product i on backorder for one period of time. Because both I_{it}^{-}

and I_{it}^{+} appear in the objective with negative coefficients, the LP solver will never make both of them positive for the same period. This simply means that we won't both carry inventory and incur a backorder penalty in the same period.

In terms of modeling, the most troublesome parameters in this formulation are the backorder penalty coefficients π_i. What is the cost of being late by one period on one unit of product i? For that matter, why should the lateness penalty be linear in the number of periods late or the number of units that are late? Clearly, asking someone in the organization for these numbers is out of the question. Therefore, one should view this type of model as a tool for generating various long-term production plans. By increasing or decreasing the π_i coefficients relative to the h_i coefficients, the analyst can increase or decrease the relative penalty associated with backlogging. High π_i values tend to force the model to build up inventory to meet surges in demand, while low π_i values tend to allow the model to be late on satisfying some demands that occur during peak periods. By generating both types of plans, the user can get an idea of what options are feasible and select among them.

To accomplish this, we need not get overly fine with the selection of cost coefficients. We could set them with the simple equations

$$h_i = \alpha p_i \tag{16.47}$$

$$\pi_i = \beta \tag{16.48}$$

where α represents the one-period interest rate, suitably inflated to penalize uncompetitive cycle times caused by excess inventory, and p_i represents the raw materials cost of one unit of product i, so that αp_i represents the interest lost on the money tied up by holding one unit of product i in inventory. Analogously, β represents a (somewhat artificial) cost per period of delay on any product. The assumption here is that the true cost of being late (expediting costs, lost customer goodwill, lost future orders, etc.) is independent of the cost or price of the product. If Equations (16.47) and (16.48) are valid, then the user can fix α and generate many different production plans by varying the single parameter β.

Overtime. The previous representations of capacity assume each workstation is available a fixed amount of time in each period. Of course, in many systems there is the possibility of increasing the time via the use of overtime. Although we will treat overtime in greater detail in our upcoming discussion of workforce planning, it makes sense to note quickly that it is a simple matter to represent the option of overtime in a product mix model, even when labor is not being considered explicitly.

To do this, let

$l'_j =$ cost of one hour of overtime at workstation j; a cost parameter

$O_{jt} =$ overtime at workstation j in period t in hours; a decision variable

We can modify LP (16.41)–(16.46) to allow overtime at each workstation as follows:

$$\text{Maximize} \quad \sum_{t=1}^{\bar{t}} \{r_i S_{it} - h_i I_{it}^{+} - \pi_i I_{it}^{-} - \sum_{j=1}^{n} l'_j O_{jt}\} \tag{16.49}$$

Subject to:

$$\underline{d}_{it} \le S_{it} \le \bar{d}_{it} \qquad \text{for all } i, t \tag{16.50}$$

$$\sum_{i=1}^{m} a_{ij} X_{it} \le c_{jt} + O_{jt} \qquad \text{for all } j, t \tag{16.51}$$

$$I_{it} = I_{it-1} + X_{it} - S_{it} \qquad \text{for all } i, t \tag{16.52}$$

$$I_{it} = I_{it}^{+} - I_{it}^{-} \qquad \text{for all } i, t \qquad (16.53)$$

$$X_{it}, S_{it}, I_{it}^{+}, I_{it}^{-} O_{jt} \geq 0 \qquad \text{for all } i, j, t \qquad (16.54)$$

The two changes we have made to LP (16.41)–(16.46) were to

1. Subtract the cost of overtime at stations $1, \ldots, n$, which is $\sum_{t=1}^{\bar{t}} \sum_{j=1}^{n} l'_j O_{jt}$, from the objective function.
2. Add the hours of overtime scheduled at station j in period t, denoted by O_{jt}, to the capacity of this resource c_{jt} in constraints (16.51).

It is natural to include both backlogging and overtime in the same model, since these are both ways of addressing capacity problems. In LP (16.49)–(16.54), the computer has the option of being late in meeting demand (backlogging) or increasing capacity via overtime. The specific combination it chooses depends on the relative cost of back-ordering (π_i) and overtime (l'_j). By varying these cost coefficients, the user can generate a range of production plans.

Yield Loss. In systems where product is scrapped at various points in the line due to quality problems, we must release extra material into the system to compensate for these losses. The result is that workstations upstream from points of yield loss are more heavily utilized than if there were no yield loss (because they must produce the extra material that will ultimately be scrapped). Therefore, to assess accurately the feasibility of a particular demand profile relative to capacity, we must consider yield loss in the aggregate planning module in systems where scrap is an issue.

We illustrate the basic effect of yield loss in Figure 16.9. In this simple line, α, β, and γ represent the fraction of product that is lost to scrap at workstations A, B, and C, respectively. If we require d units of product to come out of station C, then, on average, we will have to release $d/(1-\gamma)$ units into station C. To get $d/(1-\gamma)$ units out of station B, we will have to release $d/[(1-\beta)(1-\gamma)]$ units into B on average. Finally, to get the needed $d/[(1-\beta)(1-\gamma)]$ out of B, we will have to release $d/[(1-\alpha)(1-\beta)(1-\gamma)]$ units into A.

We can generalize the specific example of Figure 16.9 by defining

y_{ij} = cumulative yield from station j onward (including station j) for product i

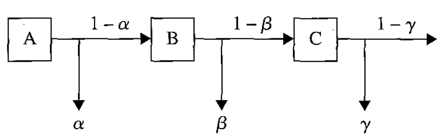
If we want to get d units of product i out of the end of the line on average, then we must release

$$\frac{d}{y_{ij}} \qquad (16.55)$$

units of i into station j. These values can easily be computed in the manner used for the example in Figure 16.9 and updated in a spreadsheet or database as a function of the estimated yield loss at each station.

Using Equation (16.55) to adjust the production amounts X_{it} in the manner illustrated in Figure 16.9, we can modify the LP formulation (16.28)–(16.32) to consider

FIGURE 16.9
Yield loss in a three-station line



yield loss as follows:

$$\text{Maximize} \quad \sum_{t=1}^{\bar{t}} r_i S_{it} - h_i I_{it} \tag{16.56}$$

Subject to:

$$\underline{d}_{it} \leq S_{it} \leq \bar{d}_{it} \quad \text{for all } i, t \tag{16.57}$$

$$\sum_{i=1}^{m} \frac{a_{ij} X_{it}}{y_{ij}} \leq c_{jt} \quad \text{for all } j, t \tag{16.58}$$

$$I_{it} = I_{it-1} + X_{it} - S_{it} \quad \text{for all } i, t \tag{16.59}$$

$$X_{it}, S_{it}, I_{it} \geq 0 \quad \text{for all } i, t \tag{16.60}$$

As one would expect, the net effect of this change is to reduce the effective capacity of workstations, particularly those at the beginning of the line. By altering the y_{ij} values (or better yet, the individual yields that make up the y_{ij} values), the planner can get a feel for the sensitivity of the system to improvements in yields. Again as one would intuitively expect, the impact of reducing the scrap rate toward the end of the line is frequently much larger than that of reducing scrap toward the beginning of the line. Obviously, scrapping product late in the process is very costly and should be avoided wherever possible. If better process control and quality assurance in the front of the line can reduce scrap later, this is probably a sound policy. An aggregate planning module like that given in LP (16.56)–(16.60) is one way to get a sense of the economic and logistic impact of such a policy.

16.4 Workforce Planning

In systems where the workload is subject to variation, due to either a changing workforce size or overtime load, it may make sense to consider the aggregate planning (AP) and workforce planning (WP) modules in tandem. Questions of how and when to resize the labor pool or whether to use overtime instead of workforce additions can be posed in the context of a linear programming formulation to support both modules.

16.4.1 An LP Model

To illustrate how an LP model can help address the workforce-resizing and overtime allocation questions, we will consider a simple single-product model. In systems where product routings and processing times are either almost identical, so that products can be aggregated into a single product, or entirely separate, so that routings can be analyzed separately, the single-product model can be reasonable. In a system where bottleneck identification is complicated by different processing times and interconnected routings, a planner would most likely need an explicit multiproduct model. This involves a straightforward integration of a product mix model, like those we discussed earlier, with a workforce-planning model like that presented next.

We introduce the following notation, paralleling that which we have used up to now, with a few additions to address the workforce issues.

$j =$ an index of workstation, $j = 1, \ldots, n$, so n represents total number of workstations

$t =$ an index of period, $t = 1, \ldots, \bar{t}$, so $\bar{t}$ represents planning horizon

$\bar{d}_t$ = maximum demand in period t
$\underline{d}_t$ = minimum sales allowed in period t
a_j = time required on workstation j to produce one unit of product
b = number of worker-hours required to produce one unit of product
c_{jt} = capacity of workstation j in period t
r = net profit per unit of product sold
h = cost to hold one unit of product for one period
l = cost of regular time in dollars per worker-hour
l' = cost of overtime in dollars per worker-hour
e = cost to increase workforce by one worker-hour per period
e' = cost to decrease workforce by one worker-hour per period
X_t = amount produced in period t
S_t = amount sold in period t
I_t = inventory at end of t (I_0 is given as data)
W_t = workforce in period t in worker-hours of regular time (W_0 is given as data)
H_t = increase (hires) in workforce from period $t-1$ to t in worker-hours
F_t = decrease (fires) in workforce from period $t-1$ to t in worker-hours
O_t = overtime in period t in hours

We now have several new parameters and decision variables for representing the workforce considerations. First, we need b, the labor content of one unit of product, in order to relate workforce requirements to production needs. Once the model has used this parameter to determine the number of labor hours required in a given month, it has two options for meeting this requirement. Either it can schedule overtime, using the variable O_t and incurring cost at rate l'_t, or it can resize the workforce, using variables H_t and F_t and incurring a cost of e (e') for every worker added (laid off).

To model this planning problem as an LP, we will need to make the assumption that the cost of worker additions or deletions is linear in the number of workers added or deleted; that is, it costs twice as much to add (delete) two workers as it does to add (delete) one. Here we are assuming that e is an estimate of the hiring, training, outfitting, and lost productivity costs associated with bringing on a new worker. Similarly, e' represents the severance pay, unemployment costs, and so on associated with letting a worker go.

Of course, in reality, these workforce-related costs may not be linear. The training cost per worker may be less for a group than for an individual, since a single instructor can train many workers for roughly the same cost as a single one. On the other hand, the plant disruption and productivity falloff from introducing many new workers may be much more severe than those from introducing a single worker. Although one can use more sophisticated models to consider such sources of nonlinearity, we will stick with an LP model, keeping in mind that we are capturing general effects rather than elaborate details. Given that the AP and WP modules are used for long-term general planning purposes and rely on speculative forecasted data (e.g., of future demand), this is probably a reasonable choice for most applications.

We can write the LP formulation of the problem to maximize net profit, including labor, overtime, holding, and hiring/firing costs, subject to constraints on sales and capacity, as

$$\text{Maximize} \quad \sum_{t=1}^{\bar{t}} \{rS_t - hI_t - lW_t - l'O_t - eH_t - e'F_t\} \tag{16.61}$$

Subject to:

$$\underline{d}_t \leq S_t \leq \bar{d}_t \quad \text{for all } t \tag{16.62}$$

$$a_j X_t \leq c_{jt} \quad \text{for all } j, t \tag{16.63}$$

$$I_t = I_{t-1} + X_t - S_t \quad \text{for all } t \tag{16.64}$$

$$W_t = W_{t-1} + H_t - F_t \quad \text{for all } t \tag{16.65}$$

$$bX_t \leq W_t + O_t \quad \text{for all } t \tag{16.66}$$

$$X_t, S_t, I_t, O_t, W_t, H_t, F_t \geq 0 \quad \text{for all } t \tag{16.67}$$

The objective function in formulation (16.61) computes profit as the difference between net revenue and inventory carrying costs, wages (regular and overtime), and workforce increase/decrease costs. Constraints (16.62) are the usual bounds on sales. Constraints (16.63) are capacity constraints for each workstation. Constraints (16.64) are the usual inventory balance equations. Constraints (16.65) and (16.66) are new to this formulation. Constraints (16.65) define the variables W_t, $t = 1, \ldots, \bar{t}$, to represent the size of the workforce in period t in units of worker-hours. Constraints (16.66) constrain the worker-hours required to produce X_t, given by bX_t, to be less than or equal to the sum of regular time plus overtime, namely, $W_t + O_t$. Finally, constraints (16.67) ensure that production, sales, inventory, overtime, workforce size, and labor increases/decreases are all nonnegative. The fact that $I_t \geq 0$ implies no backlogging, but we could easily modify this model to account for backlogging in a manner like that used in LP (16.41)–(16.46).

16.4.2 A Combined AP/WP Example

To make LP (16.61)–(16.67) concrete and to give a flavor for the manner in which modeling, analysis, and decision making interact, we consider the example presented in the spreadsheet of Figure 16.10. This represents an AP problem for a single product with unit net revenue of $1,000 over a 12-month planning horizon. We assume that each worker works 168 hours per month and that there are 15 workers in the system at the beginning of the planning horizon. Hence, the total number of labor hours available at the start of the problem is

$$W_0 = 15 \times 168 = 2{,}520$$

There is no inventory in the system at the start, so $I_0 = 0$.

The cost parameters are estimated as follows. Monthly holding cost is $10 per unit. Regular time labor (with benefits) costs $35 per hour. Overtime is paid at time-and-a-half, which is equal to $52.50 per hour. It costs roughly $2,500 to hire and train a new worker. Since this worker will account for 168 hours per month, the cost in terms of dollars per worker-hour is

$$\frac{\$2{,}500}{168} = \$14.88 \approx \$15 \text{ per hour}$$

Since this number is only a rough approximation, we will round to an even $15. Similarly, we estimate the cost to lay off a worker to be about $1,500, so the cost per hour of

FIGURE 16.10

Initial spreadsheet for workforce planning example

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Parameters:**												
2	r	1000											
3	h	10											
4	l	35											
5	l′	52.5											
6	e	15											
7	e′	9											
8	b	12											
9	I_0	0											
10	W_0	2520											
11	**t**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
12	d_t	200	220	230	300	400	450	320	180	170	170	160	180
13													
14	**Decision Variables:**												
15	**t**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
16	Xt	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	Wt	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	Ht	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	Ft	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	It	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	Ot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22													
23	**Objective:**												
24	**Profit:**	$2,980,600.00											
25													
26	**Constraints:**												
27	I1-I0-X1	0.00	=	-200	**-d_1**								
28	I2-I1-X2	0.00	=	-220	**-d_2**								
29	I3-I2-X3	0.00	=	-230	**-d_3**								
30	I4-I3-X4	0.00	=	-300	**-d_4**								
31	I5-I4-X5	0.00	=	-400	**-d_5**								
32	I6-I5-X6	0.00	=	-450	**-d_6**								
33	I7-I6-X7	0.00	=	-320	**-d_7**								
34	I8-I7-X8	0.00	=	-180	**-d_8**								
35	I9-I8-X9	0.00	=	-170	**-d_9**								
36	I10-I9-X10	0.00	=	-170	**-d_10**								
37	I11-I10-X11	0.00	=	-160	**-d_11**								
38	I12-I11-X12	0.00	=	-180	**-d_12**								
39	W1-W0-H1+F1	-2520.00	=	0									
40	W2-W1-H2+F2	0.00	=	0									
41	W3-W2-H3+F3	0.00	=	0									
42	W4-W3-H4+F4	0.00	=	0									
43	W5-W4-H5+F5	0.00	=	0									
44	W6-W5-H6+F6	0.00	=	0									
45	W7-W6-H7+F7	0.00	=	0									
46	W8-W7-H8+F8	0.00	=	0									
47	W9-W8-H9+F9	0.00	=	0									
48	W10-W9-H10+F10	0.00	=	0									
49	W11-W10-H11+F11	0.00	=	0									
50	W12-W11-H12+F12	0.00	=	0									
51	bX1-W1-O1	0.00	<=	0									
52	bX2-W2-O2	0.00	<=	0									
53	bX3-W3-O3	0.00	<=	0									
54	bX4-W4-O4	0.00	<=	0									
55	bX5-W5-O5	0.00	<=	0									
56	bX6-W6-O6	0.00	<=	0									
57	bX7-W7-O7	0.00	<=	0									
58	bX8-W8-O8	0.00	<=	0									
59	bX9-W9-O9	0.00	<=	0									
60	bX10-W10-O10	0.00	<=	0									
61	bX11-W11-O11	0.00	<=	0									
62	bX12-W12-O12	0.00	<=	0									
63		*Note: All decision variables must be >= 0*											

reduction in the monthly workforce is

$$\frac{\$1{,}500}{168} = \$8.93 \approx \$9 \text{ per hour}$$

Again, we will use the rounded value of \$9, since data are rough.

Notice that the projected demands (d_t) in the spreadsheet have a seasonal pattern to them, building to a peak in months 5 and 6, and tapering off thereafter. We will assume that backordering is not an option and that demands must be met, so the main issue will be how to do this.

Let us begin by expressing LP (16.61)–(16.67) in concrete terms for this problem. Because we are assuming that demands are met, we set $S_t = d_t$, which eliminates the need for separate sales variables S_t and sales constraints (16.62). Furthermore, to keep things simple, we will assume that the only capacity constraints are those posed by labor (i.e., it requires 12 hours of labor to produce each unit of product). No other machine or resource constraints need be considered. Thus we can omit constraints (16.63). Under these assumptions, the resulting LP formulation is

$$\begin{aligned}\text{Maximize}\quad & 1{,}000(d_1 + \cdots + d_{12}) - 10(I_1 + \cdots + I_{12})\\ & -35(W_1 + \cdots + W_{12}) - 52.5(O_1 + \cdots + O_{12})\\ & -15(H_1 + \cdots + H_{12}) - 9(F_1 + \cdots + F_{12}) \qquad (16.68)\end{aligned}$$

Subject to:

$$I_1 - I_0 - X_1 = -d_1 \qquad (16.69)$$
$$I_2 - I_1 - X_2 = -d_2 \qquad (16.70)$$
$$I_3 - I_2 - X_3 = -d_3 \qquad (16.71)$$
$$I_4 - I_3 - X_4 = -d_4 \qquad (16.72)$$
$$I_5 - I_4 - X_5 = -d_5 \qquad (16.73)$$
$$I_6 - I_5 - X_6 = -d_6 \qquad (16.74)$$
$$I_7 - I_6 - X_7 = -d_7 \qquad (16.75)$$
$$I_8 - I_7 - X_8 = -d_8 \qquad (16.76)$$
$$I_9 - I_8 - X_9 = -d_9 \qquad (16.77)$$
$$I_{10} - I_9 - X_{10} = -d_{10} \qquad (16.78)$$
$$I_{11} - I_{10} - X_{11} = -d_{11} \qquad (16.79)$$
$$I_{12} - I_{11} - X_{12} = -d_{12} \qquad (16.80)$$
$$W_1 - H_1 + F_1 = 2{,}520 \qquad (16.81)$$
$$W_2 - W_1 - H_2 + F_2 = 0 \qquad (16.82)$$
$$W_3 - W_2 - H_3 + F_3 = 0 \qquad (16.83)$$
$$W_4 - W_3 - H_4 + F_4 = 0 \qquad (16.84)$$
$$W_5 - W_4 - H_5 + F_5 = 0 \qquad (16.85)$$
$$W_6 - W_5 - H_6 + F_6 = 0 \qquad (16.86)$$
$$W_7 - W_6 - H_7 + F_7 = 0 \qquad (16.87)$$
$$W_8 - W_7 - H_8 + F_8 = 0 \qquad (16.88)$$
$$W_9 - W_8 - H_9 + F_9 = 0 \qquad (16.89)$$
$$W_{10} - W_9 - H_{10} + F_{10} = 0 \qquad (16.90)$$
$$W_{11} - W_{10} - H_{11} + F_{11} = 0 \qquad (16.91)$$
$$W_{12} - W_{11} - H_{12} + F_{12} = 0 \qquad (16.92)$$
$$12X_1 - W_1 - O_1 \leq 0 \qquad (16.93)$$
$$12X_2 - W_2 - O_2 \leq 0 \qquad (16.94)$$
$$12X_3 - W_3 - O_3 \leq 0 \qquad (16.95)$$

$$12X_4 - W_4 - O_4 \leq 0 \quad (16.96)$$
$$12X_5 - W_5 - O_5 \leq 0 \quad (16.97)$$
$$12X_6 - W_6 - O_6 \leq 0 \quad (16.98)$$
$$12X_7 - W_7 - O_7 \leq 0 \quad (16.99)$$
$$12X_8 - W_8 - O_8 \leq 0 \quad (16.100)$$
$$12X_9 - W_9 - O_9 \leq 0 \quad (16.101)$$
$$12X_{10} - W_{10} - O_{10} \leq 0 \quad (16.102)$$
$$12X_{11} - W_{11} - O_{11} \leq 0 \quad (16.103)$$
$$12X_{12} - W_{12} - O_{12} \leq 0 \quad (16.104)$$
$$X_t, I_t, O_t, W_t, H_t, F_t \geq 0 \qquad t = 1, \ldots, 12 \quad (16.105)$$

Objective (16.68) is identical to objective (16.61), except that the S_t variables have been replaced with d_t constants.[8] Constraints (16.69)–(16.80) are the usual balance constraints. For instance, constraint (16.69) simply states that

$$I_1 = I_0 + X_1 - d_1$$

That is, inventory at the end of month 1 equals inventory at the end of month 0 (i.e., the beginning of the problem) plus production during month 1, minus sales (demand) in month 1. We have arranged these constraints so that all decision variables are on the left-hand side of the equality and constants (d_t) are on the right-hand side. This is often a convenient modeling convention, as we will see in our analysis.

Constraints (16.81) to (16.92) are the labor balance equations given in constraints (16.65) of our general formulation. For instance, constraint (16.81) represents the relation

$$W_1 = W_0 + H_1 - F_1$$

so that the workforce at the end of month 1 (in units of worker-hours) is equal to the workforce at the end of month 0, plus any additions in month 1, minus any subtractions in month 1.

Constraints (16.93) to (16.104) ensure that the labor content of the production plan does not exceed available labor, which can include overtime. For instance, constraint (16.93) can be written as

$$12X_1 \leq W_1 + O_1$$

In the spreadsheet shown in Figure 16.10, we have entered the decision variables X_t, W_t, H_t, F_t, I_t, and O_t into cells B16:M21. Using these variables and the various coefficients from the top of the spreadsheet, we express objective (16.68) as a formula in cell B24. Notice that this formula reports a value equal to the unit profit times total demand, or

$$1{,}000(200 + 220 + 230 + 300 + 400 + 450 + 320$$
$$+ 180 + 170 + 170 + 160 + 180) = \$2{,}980{,}000$$

because all other terms in the objective are zero when the decision variables are set at zero.

We enter formulas for the left-hand sides of constraints (16.69) to (16.80) in cells B27:B38, the left-hand sides of constraints (16.81) to (16.92) in cells B39:B50, and the

[8]Since the d_t values are fixed, the first term in the objective function is not a function of our decision variables and could be left out without affecting the solution. We have kept it in so that our model reports a sensible profit function.

left-hand sides of constraints (16.93) to (16.104) in cells B51:B62. Notice that many of these constraints are not satisfied when all decision variables are equal to zero. This is hardly surprising, since we cannot expect to earn revenues from sales of product we have not made.

A convenient aspect of using a spreadsheet for solving LP models is that it provides us with a mechanism for playing with the model to gain insight into its behavior. For instance, in the spreadsheet of Figure 16.11 we try a **chase solution** where we set production equal to demand ($X_t = d_t$) and leave $W_t = W_0$ in every period. Although this satisfies the inventory balance constraints in cells B27:B38, and the workforce balance constraints in cells B39:B50, it violates the labor content constraints in cells B52:B57. The reason, of course, is that the current workforce is not sufficient to meet demand without using overtime. We could try adding overtime by adjusting the O_t variables in cells B21:M21. However, searching around for an optimal solution can be difficult, particularly in large models. Therefore, we will let the LP solver in the software do the work for us.

Using the procedure we described earlier, we specify constraints (16.69) to (16.105) in our model and turn it loose. The result is the spreadsheet in Figure 16.12. Based on the costs we chose, it turns out to be optimal not to use any overtime. (Overtime costs $\$52.5 - 35 = 15.50$ per hour each month, while hiring a new worker costs only \$15 per hour as a one-time cost.) Instead, the model adds 1,114.29 hours to the workforce, which represents

$$\frac{1,114.29}{168} = 6.6$$

new workers. After the peak season of months 4 to 7, the solution calls for a reduction of $1,474.29 + 120 = 1,594.29$ hours, which implies laying off

$$\frac{1,594.29}{168} = 9.5$$

workers. Additionally, the solution involves building in excess of demand in months 1 to 4 and using this inventory to meet peak demand in months 5 to 7. The net profit resulting from this solution is \$1,687,337.14.

From a management standpoint, the planned layoffs in months 8 and 9 might be a problem. Although we have specified penalties for these layoffs, these penalties are highly speculative and may not accurately consider the long-term effects of hiring and firing on worker morale, productivity, and the firm's ability to recruit good people. Thus, it probably makes sense to carry our analysis further.

One approach we might consider would be to allow the model to hire but not fire workers. We can easily do this by eliminating the F_t variables or, since this requires fairly extensive changes in the spreadsheet, specifying additional constraints of the form

$$F_t = 0 \qquad t = 1, \ldots, 12$$

Rerunning the model with these additional constraints produces the spreadsheet in Figure 16.13. As we expect, this solution does not include any layoffs. Somewhat surprising, however, is the fact that it does not involve any new hires either (that is, $H_t = 0$ for every period). Instead of increasing the workforce size, the model has chosen to use overtime in months 3 to 7. Evidently, if we cannot fire workers, it is uneconomical to hire additional people.

However, when one looks more closely at the solution in Figure 16.13, a problem becomes evident. Overtime is too high. For instance, month 6 has more hours of overtime than hours of regular time! This means that our workforce of 15 people has

FIGURE 16.11

Infeasible "chase" solution

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Parameters:**												
2	r	1000											
3	h	10											
4	l	35											
5	l′	52.5											
6	e	15											
7	e′	9											
8	b	12											
9	I_0	0											
10	W_0	2520											
11	t	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
12	d_t	200	220	230	300	400	450	320	180	170	170	160	180
13													
14	**Decision Variables:**												
15	t	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
16	Xt	200.00	220.00	230.00	300.00	400.00	450.00	320.00	180.00	170.00	170.00	160.00	180.00
17	Wt	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00
18	Ht	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	Ft	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	It	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	Ot	10.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22													
23	**Objective:**												
24	**Profit:**	$1,921,600.00											
25													
26	**Constraints:**												
27	I1-I0-X1	-200.00	=	-200	**-d_1**								
28	I2-I1-X2	-220.00	=	-220	**-d_2**								
29	I3-I2-X3	-230.00	=	-230	**-d_3**								
30	I4-I3-X4	-300.00	=	-300	**-d_4**								
31	I5-I4-X5	-400.00	=	-400	**-d_5**								
32	I6-I5-X6	-450.00	=	-450	**-d_6**								
33	I7-I6-X7	-320.00	=	-320	**-d_7**								
34	I8-I7-X8	-180.00	=	-180	**-d_8**								
35	I9-I8-X9	-170.00	=	-170	**-d_9**								
36	I10-I9-X10	-170.00	=	-170	**-d_10**								
37	I11-I10-X11	-160.00	=	-160	**-d_11**								
38	I12-I11-X12	-180.00	=	-180	**-d_12**								
39	W1-W0-H1+F1	0.00	=	0									
40	W2-W1-H2+F2	0.00	=	0									
41	W3-W2-H3+F3	0.00	=	0									
42	W4-W3-H4+F4	0.00	=	0									
43	W5-W4-H5+F5	0.00	=	0									
44	W6-W5-H6+F6	0.00	=	0									
45	W7-W6-H7+F7	0.00	=	0									
46	W8-W7-H8+F8	0.00	=	0									
47	W9-W8-H9+F9	0.00	=	0									
48	W10-W9-H10+F10	0.00	=	0									
49	W11-W10-H11+F11	0.00	=	0									
50	W12-W11-H12+F12	0.00	=	0									
51	bX1-W1-O1	-120.00	<=	0									
52	bX2-W2-O2	120.00	<=	0									
53	bX3-W3-O3	240.00	<=	0									
54	bX4-W4-O4	1080.00	<=	0									
55	bX5-W5-O5	2280.00	<=	0									
56	bX6-W6-O6	2880.00	<=	0									
57	bX7-W7-O7	1320.00	<=	0									
58	bX8-W8-O8	-360.00	<=	0									
59	bX9-W9-O9	-480.00	<=	0									
60	bX10-W10-O10	-480.00	<=	0									
61	bX11-W11-O11	-600.00	<=	0									
62	bX12-W12-O12	-360.00	<=	0									
63		*Note: All decision variables must be >= 0*											

2,880/15 = 192 hours of overtime in the month, or about 48 hours per week per worker. This is obviously excessive.

One way to eliminate this overtime problem is to add some more constraints. For instance, we might specify that overtime is not to exceed 20 percent of regular time. This would correspond to the entire workforce working an average of one full day of

FIGURE 16.12
LP optimal solution

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Parameters:**												
2	r	1000											
3	h	10											
4	l	35											
5	l′	52.5											
6	e	15											
7	e′	9											
8	b	12											
9	I_0	0											
10	W_0	2520											
11	t	1	2	3	4	5	6	7	8	9	10	11	12
12	d_t	200	220	230	300	400	450	320	180	170	170	160	180
13													
14	**Decision Variables:**												
15	t	1	2	3	4	5	6	7	8	9	10	11	12
16	Xt	302.86	302.86	302.86	302.86	302.86	302.86	302.86	180.00	170.00	170.00	170.00	170.00
17	Wt	3634.29	3634.29	3634.29	3634.29	3634.29	3634.29	3634.29	2160.00	2040.00	2040.00	2040.00	2040.00
18	Ht	1114.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	Ft	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1474.29	120.00	0.00	0.00	0.00
20	It	102.86	185.71	258.57	261.43	164.29	17.14	0.00	0.00	0.00	0.00	10.00	0.00
21	Ot	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22													
23	**Objective:**												
24	**Profit:**	$1,687,337.14											
25													
26	**Constraints:**												
27	I1-I0-X1	-200.00	=	-200	-d_1								
28	I2-I1-X2	-220.00	=	-220	-d_2								
29	I3-I2-X3	-230.00	=	-230	-d_3								
30	I4-I3-X4	-300.00	=	-300	-d_4								
31	I5-I4-X5	-400.00	=	-400	-d_5								
32	I6-I5-X6	-450.00	=	-450	-d_6								
33	I7-I6-X7	-320.00	=	-320	-d_7								
34	I8-I7-X8	-180.00	=	-180	-d_8								
35	I9-I8-X9	-170.00	=	-170	-d_9								
36	I10-I9-X10	-170.00	=	-170	-d_10								
37	I11-I10-X11	-160.00	=	-160	-d_11								
38	I12-I11-X12	-180.00	=	-180	-d_12								
39	W1-W0-H1+F1	0.00	=	0									
40	W2-W1-H2+F2	0.00	=	0									
41	W3-W2-H3+F3	0.00	=	0									
42	W4-W3-H4+F4	0.00	=	0									
43	W5-W4-H5+F5	0.00	=	0									
44	W6-W5-H6+F6	0.00	=	0									
45	W7-W6-H7+F7	0.00	=	0									
46	W8-W7-H8+F8	0.00	=	0									
47	W9-W8-H9+F9	0.00	=	0									
48	W10-W9-H10+F10	0.00	=	0									
49	W11-W10-H11+F11	0.00	=	0									
50	W12-W11-H12+F12	0.00	=	0									
51	bX1-W1-O1	0.00	<=	0									
52	bX2-W2-O2	0.00	<=	0									
53	bX3-W3-O3	0.00	<=	0									
54	bX4-W4-O4	0.00	<=	0									
55	bX5-W5-O5	0.00	<=	0									
56	bX6-W6-O6	0.00	<=	0									
57	bX7-W7-O7	0.00	<=	0									
58	bX8-W8-O8	0.00	<=	0									
59	bX9-W9-O9	0.00	<=	0									
60	bX10-W10-O10	0.00	<=	0									
61	bX11-W11-O11	0.00	<=	0									
62	bX12-W12-O12	0.00	<=	0									
63		*Note: All decision variables must be >= 0*											

overtime per week in addition to the normal five-day workweek. We could do this by adding constraints of the form

$$O_t \leq 0.2W_t \qquad t = 1, \ldots, 12 \tag{16.106}$$

Doing this to the spreadsheet of Figure 16.13 and resolving results in the spreadsheet

FIGURE 16.13

Optimal solution when $F_t = 0$

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Parameters:**												
2	r	1000											
3	h	10											
4	l	35											
5	l′	52.5											
6	e	15											
7	e′	9											
8	b	12											
9	l_0	0											
10	W_0	2520											
11	t	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
12	d_t	200	220	230	300	400	450	320	180	170	170	160	180
13													
14	**Decision Variables:**												
15	t	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
16	Xt	210.00	210.00	230.00	300.00	400.00	450.00	320.00	180.00	170.00	170.00	160.00	180.00
17	Wt	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00	2520.00
18	Ht	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	Ft	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	It	10.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	Ot	0.00	0.00	240.00	1080.00	2280.00	2880.00	1320.00	0.00	0.00	0.00	0.00	0.00
22													
23	**Objective:**												
24	**Profit:**	$1,512,000.00											
25													

	A	B	C	D	E
26	**Constraints:**				
27	I1-I0-X1	-200.00	=	-200	**-d_1**
28	I2-I1-X2	-220.00	=	-220	**-d_2**
29	I3-I2-X3	-230.00	=	-230	**-d_3**
30	I4-I3-X4	-300.00	=	-300	**-d_4**
31	I5-I4-X5	-400.00	=	-400	**-d_5**
32	I6-I5-X6	-450.00	=	-450	**-d_6**
33	I7-I6-X7	-320.00	=	-320	**-d_7**
34	I8-I7-X8	-180.00	=	-180	**-d_8**
35	I9-I8-X9	-170.00	=	-170	**-d_9**
36	I10-I9-X10	-170.00	=	-170	**-d_10**
37	I11-I10-X11	-160.00	=	-160	**-d_11**
38	I12-I11-X12	-180.00	=	-180	**-d_12**
39	W1-W0-H1+F1	0.00	=	0	
40	W2-W1-H2+F2	0.00	=	0	
41	W3-W2-H3+F3	0.00	=	0	
42	W4-W3-H4+F4	0.00	=	0	
43	W5-W4-H5+F5	0.00	=	0	
44	W6-W5-H6+F6	0.00	=	0	
45	W7-W6-H7+F7	0.00	=	0	
46	W8-W7-H8+F8	0.00	=	0	
47	W9-W8-H9+F9	0.00	=	0	
48	W10-W9-H10+F10	0.00	=	0	
49	W11-W10-H11+F11	0.00	=	0	
50	W12-W11-H12+F12	0.00	=	0	
51	bX1-W1-O1	0.00	<=	0	
52	bX2-W2-O2	0.00	<=	0	
53	bX3-W3-O3	0.00	<=	0	
54	bX4-W4-O4	0.00	<=	0	
55	bX5-W5-O5	0.00	<=	0	
56	bX6-W6-O6	0.00	<=	0	
57	bX7-W7-O7	0.00	<=	0	
58	bX8-W8-O8	-360.00	<=	0	
59	bX9-W9-O9	-480.00	<=	0	
60	bX10-W10-O10	-480.00	<=	0	
61	bX11-W11-O11	-600.00	<=	0	
62	bX12-W12-O12	-360.00	<=	0	
63		*Note: All decision variables must be >= 0*			

shown in Figure 16.14. The overtime limits have forced the model to resort to hiring. Since layoffs are still not allowed, the model hires only 508.57 hours worth of workers, or

$$\frac{508.57}{168} = 3$$

FIGURE 16.14

Optimal solution when $F_t = 0$ and $O_t \leq 0.2W_t$

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Parameters:**												
2	r	1000											
3	h	10											
4	I	35											
5	I′	52.5											
6	e	15											
7	e′	9											
8	b	12											
9	I_0	0											
10	W_0	2520											
11	t	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
12	d_t	200	220	230	300	400	450	320	180	170	170	160	180
13													
14	**Decision Variables:**												
15	t	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
16	Xt	302.86	302.86	302.86	302.86	302.86	302.86	302.86	180.00	170.00	170.00	160.00	180.00
17	Wt	3028.57	3028.57	3028.57	3028.57	3028.57	3028.57	3028.57	3028.57	3028.57	3028.57	3028.57	3028.57
18	Ht	508.57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	Ft	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	It	102.86	185.71	258.57	261.43	164.29	17.14	0.00	0.00	0.00	0.00	0.00	0.00
21	Ot	605.71	605.71	605.71	605.71	605.71	605.71	605.71	0.00	0.00	0.00	0.00	0.00
22													
23	**Objective:**												
24	**Profit:**	$1,467,871.43											
25													
26	**Constraints:**												
27	I1-I0-X1	-200.00	=	-200	**-d_1**								
28	I2-I1-X2	-220.00	=	-220	**-d_2**								
29	I3-I2-X3	-230.00	=	-230	**-d_3**								
30	I4-I3-X4	-300.00	=	-300	**-d_4**								
31	I5-I4-X5	-400.00	=	-400	**-d_5**								
32	I6-I5-X6	-450.00	=	-450	**-d_6**								
33	I7-I6-X7	-320.00	=	-320	**-d_7**								
34	I8-I7-X8	-180.00	=	-180	**-d_8**								
35	I9-I8-X9	-170.00	=	-170	**-d_9**								
36	I10-I9-X10	-170.00	=	-170	**-d_10**								
37	I11-I10-X11	-160.00	=	-160	**-d_11**								
38	I12-I11-X12	-180.00	=	-180	**-d_12**								
39	W1-W0-H1+F1	0.00	=	0									
40	W2-W1-H2+F2	0.00	=	0									
41	W3-W2-H3+F3	0.00	=	0									
42	W4-W3-H4+F4	0.00	=	0									
43	W5-W4-H5+F5	0.00	=	0									
44	W6-W5-H6+F6	0.00	=	0									
45	W7-W6-H7+F7	0.00	=	0									
46	W8-W7-H8+F8	0.00	=	0									
47	W9-W8-H9+F9	0.00	=	0									
48	W10-W9-H10+F10	0.00	=	0									
49	W11-W10-H11+F11	0.00	=	0									
50	W12-W11-H12+F12	0.00	=	0									
51	bX1-W1-O1	0.00	<=	0									
52	bX2-W2-O2	0.00	<=	0									
53	bX3-W3-O3	0.00	<=	0									
54	bX4-W4-O4	0.00	<=	0									
55	bX5-W5-O5	0.00	<=	0									
56	bX6-W6-O6	0.00	<=	0									
57	bX7-W7-O7	0.00	<=	0									
58	bX8-W8-O8	-868.57	<=	0									
59	bX9-W9-O9	-988.57	<=	0									
60	bX10-W10-O10	-988.57	<=	0									
61	bX11-W11-O11	-1108.57	<=	0									
62	bX12-W12-O12	-868.57	<=	0									
63		*Note: All decision variables must be >= 0*											

new workers, as opposed to the 6.6 workers hired in the original solution in Figure 16.12. To attain the necessary production, the solution uses overtime in months 1 to 7. Notice that the amount of overtime used in these months is exactly 20 percent of regular time work hours, that is,

$$3{,}028.57 \times 0.2 = 605.71$$

What this means is that new constraints (16.106) are binding for periods 1 to 7, which we would be told explicitly if we printed out the sensitivity analysis reports generated by the LP solver. This implies that if it is possible to work more overtime in any of these months, we can improve the solution.

Notice that the net profit in the model of the spreadsheet shown in Figure 16.14 is $1,467,871.43, which is a 13 percent decrease over the original optimal solution of $1,687,337.14 in Figure 16.12. At first glance, it may appear that the policies of no layoffs and limits on overtime are expensive. On the other hand, it may really be telling us that our original estimates of the costs of hiring and firing were too low. If we were to increase these costs to represent, for example, long-term disruptions caused by labor changes, the optimal solution might be very much like the one arrived at in Figure 16.14.

16.4.3 Modeling Insights

In addition to providing a detailed example of a workforce formulation in LP (16.61)–(16.67), we hope that our discussion has helped the reader appreciate the following aspects of using an optimization model as the basis for an AP or WP module.

1. *Multiple modeling approaches.* There are often many ways to model a given problem, none of which is "correct" in any absolute sense. The key is to use cost coefficients and constraints to represent the main issues in a sensible way. In this example, we could have generated solutions without layoffs by either increasing the layoff penalty or placing constraints on the layoffs. Both approaches would achieve the same qualitative conclusions.

2. *Iterative model development.* Modeling and analysis almost never proceed in an ideal fashion in which the model is formulated, solved, and interpreted in a single pass. Often the solution from one version of the model suggests an alternate model. For instance, we had no way of knowing that eliminating layoffs would cause excessive overtime in the solution. We didn't know we would need constraints on the level of overtime until we saw the spreadsheet output of Figure 16.13.

16.5 Conclusions

In this chapter, we have given an overview of the issues involved in aggregate and workforce planning. A key observation behind our approach is that, because the aggregate planning and workforce planning modules use long time horizons, precise data and intricate modeling detail are impractical or impossible. We must recognize that the production or workforce plans that these modules generate will be adjusted as time evolves. The lower levels in the PPC hierarchy must handle the nuts-and-bolts challenge of converting the plans to action. The keys to a good AP module are to keep the focus on long-term planning (i.e., avoiding putting too many short-term control details in the model) and to provide links for consistency with other levels in the hierarchy. Some of the issues related to consistency were discussed in Chapter 13. Here, we close with some general observations about the aggregate and workforce planning functions:

1. *No single AP or WP module is right for every situation.* As the examples in this chapter show, aggregate and workforce planning can incorporate many different decision problems. A good AP or WP module is one that is tailored to address the specific issues faced by the firm.

2. *Simplicity promotes understanding.* Although it is desirable to address different issues in the AP/WP module, it is even more important to keep the model understandable. In general, these modules are used to generate candidate production and workforce plans, which will be examined, combined, and altered manually before being published as "The Plan." To generate a spectrum of plans (and explain them to others), the user must be able to trace changes in the model to changes in the plan. Because of this, it makes sense to start with as simple a formulation as possible. Additional detail (e.g., constraints) can be added later.

3. *Linear programming is a useful AP/WP tool.* The long planning horizon used for aggregate and workforce planning justifies ignoring many production details; therefore, capacity checks, sales restrictions, and inventory balances can be expressed as linear constraints. As long as we are willing to approximate actual costs with linear functions, an LP solver is a very efficient method for solving many problems related to the AP and WP modules. Because we are working with speculative long-range data, it generally does not make sense to use anything more sophisticated than LP (e.g., nonlinear or integer programming) in most aggregate and workforce planning situations.

4. *Robustness matters more than precision.* No matter how accurate the data and how sophisticated the model, the plan generated by the AP or WP module will never be followed exactly. The actual production sequence will be affected by unforeseen events that could not possibly have been factored into the module. This means that the mark of a good long-range production plan is that it enables us to do a reasonably good job even in the face of such contingencies. To find such a plan, the user of the AP module must be able to examine the consequences of various scenarios. This is another reason to keep the model reasonably simple.

Appendix 16A
Linear Programming

Linear programming is a powerful mathematical tool for solving constrained optimization problems. The name derives from the fact that LP was first applied to find optimal schedules or "programs" of resource allocation. Hence, although LP generally does involve using a computer program, it does not entail programming on the part of the user in the sense of writing code.

In this appendix, we provide enough background to give the user of an LP package a basic idea of what the software is doing. Readers interested in more details should consult one of the many good texts on the subject (e.g., Eppen and Gould 1988 for an application-oriented overview, Murty 1983 for more technical coverage).

Formulation

The first step in using linear programming is to formulate a practical problem in mathematical terms. There are three basic choices we must make to do this:

1. **Decision variables** are quantities under our control. Typical examples for aggregate planning and workforce planning applications of LP are production quantities, number of workers to hire, and levels of inventory to hold.
2. **Objective function** is what we want to maximize or minimize. In most AP/WP applications, this is typically either to maximize profit or minimize cost. Beyond simply stating the objective, however, we must specify it in terms of the decision variables we have defined.