9

MIXED-INTEGER PROGRAMMING

Introduction

Many problems in plant operation, design, location, and scheduling involve variables that are not continuous but instead have integer values. Decision variables for which the levels are a dichotomy—to install or not install a new piece of equipment, for example—are termed "0–1" or binary variables. Other integer variables might be real numbers 0, 1, 2, 3, and so on. Sometimes we can treat integer variables as if they were continuous, especially when the range of a variable contains a large number of integers, such as 100 trays in a distillation column, and round the optimal solution to the nearest integer value. Although this procedure leads to a suboptimal solution, the solution is quite acceptable from a practical viewpoint. However, for a small range of a variable such as 1 to 3, when the optimal solution yields a value of 1.3, we have less confidence in rounding. In this section we will illustrate some examples of problem formulation and subsequent solution in which one or more variables are treated as integer variables.

First let us classify the types of problems that are encountered in optimization with discrete variables. The most general case is a *mixed integer programming* (MIP) problem in which the objective function depends on two sets of variables, $\mathbf{x}$ and $\mathbf{y}$; $\mathbf{x}$ is a vector of continuous variables and $\mathbf{y}$ is a vector of integer variables. A problem involving only integer variables is classified as an *integer programming* (IP) problem. Finally, a special case of IP is *binary integer programming* (BIP), in which all of the variables $\mathbf{y}$ are either 0 or 1. Many MIP problems are linear in the objective function and constraints and hence are subject to solution by linear programming. These problems are called *mixed-integer linear programming* (MILP) problems. Problems involving discrete variables in which some of the functions are nonlinear are called *mixed-integer nonlinear programming* (MINLP) problems. We consider both linear and nonlinear MIP problems in this chapter.

9.1 PROBLEM FORMULATION

Here we review some classical formulations of typical integer programming problems that have been discussed in the operations research literature, as well as some problems that have direct applicability to chemical processing:

1. *The knapsack problem.* We have n objects. The weight of the ith object is w_i, and its value is v_i. Select a subset of the objects such that their total weight does not exceed W (the capacity of the knapsack) and their total value is a maximum.

$$\text{Maximize:} \quad f(\mathbf{y}) = \sum_{i=1}^{n} v_i y_i$$

$$\text{Subject to:} \quad \sum_{i=1}^{n} w_i y_i \le \mathrm{W} \quad y_i = 0, 1 \quad i = 1, 2, \ldots, n$$

The binary variable y_i indicates whether an object i is selected ($y_i = 1$) or not selected ($y_i = 0$).

2. *The traveling salesman problem.* The problem is to assign values of 0 or 1 to variables y_{ij}, where y_{ij} is 1 if the salesman travels from city i to city j and 0 otherwise. The constraints in the problem are that the salesman must start at a particular city, visit each of the other cities only once, and return to the original city. A cost (here it is distance) c_{ij} is associated with traveling from city i to city j, and the objective function is to minimize the total cost of the trips to each city visited, that is

$$f(\mathbf{y}) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} y_{ij}$$

subject to the $2n$ constraints

$$\sum_{i=1}^{n} y_{ij} = 1, \qquad \sum_{j=1}^{n} y_{ij} = 1 \qquad \begin{array}{ll} y_{ij} = 0, 1 & i,j = 1, \ldots, n \\ y_{ij} = 0 & i = j \end{array}$$

The two types of equality constraints ensure that each city is only visited once in any direction. We define $y_{ii} = 0$ because no trip is involved. The equality constraints (the summations) ensure that each city is entered and exited exactly once. These are the constraints of an assignment problem (see Section 7.8). In addition, constraints must be added to ensure that the y_{ij} which are set equal to 1 correspond to a single circular tour or cycle, not to two or more disjoint cycles. For more information on how to write such constraints, see Nemhauser and Wolsey (1988).

For a chemical plant analogy, the problem can also be cast in terms of processing n batches on a single piece of equipment in which the equipment is reset between processing the ith and jth batches. The batches can be processed in any order. Here, c_{ij} is the time or cost required to "set up" the equipment to do batch j if it was previously doing batch i, and $y_{ij} = 1$ means batch i is immediately followed by batch j.

3. *Blending problem.* You are given a list of possible ingredients to be blended into a product, from a list containing the weight, value, cost, and analysis of each ingredient. The objective is to select from the list a set of ingredients so as to have a satisfactory total weight and analysis at minimum cost for a blend. Let x_j be the quantity of ingredient j available in continuous amounts and y_k represent ingredients to be used in discrete quantities v_k ($y_k = 1$ if used and $y_k = 0$ if not used). Let c_j and d_k be the respective costs of the ingredients and a_{ij} be the fraction of component i in ingredients j. The problem statement is

$$\text{Minimize:} \quad \sum_j c_j x_j + \sum_k d_k v_k y_k$$

$$\text{Subject to:} \quad W^l \leq \sum_j x_j + \sum_k v_k y_k \leq W^u$$

$$A_i^l \leq \sum_j a_{ij} x_j + \sum_k a_{ik} v_k y_k \leq A_i^u$$

$$0 \leq x_j \leq u_j \quad \text{for all } j$$
$$y_k = (0, 1) \quad \text{for all } k$$

where u_j = upper limit of the jth ingredient,
W^l and W^u = the lower and upper bounds on the weights, respectively
A_i^l and A_i^u = the lower and upper bounds on the analysis for component i, respectively

4. *Location of oil wells (plant location problem).* It is assumed that a specific production–demand versus time relation exists for a reservoir. Several sites for new wells have been designated. The problem is how to select from among the well sites the number of wells to be drilled, their locations, and the production rates from the wells so that the difference between the production–demand curve and flow curve actually obtained is minimized. Refer to Rosenwald and Green (1974) and Murray and Edgar (1978) for a mathematical formulation of the problem. The integer variables are the drilling decisions (0 = not drilled, 1 = drilled) for a set of n possible drilling locations The continuous variables are the different well production rates. This problem is related to the plant location problem and also the *fixed-charge problem* (Hillier and Lieberman, 1986).

Many other problems can be formulated as integer programming problems; refer to the examples in this chapter and Nemhauser and Wolsey (1988) and the supplementary references for additional examples.

Integer and mixed-integer programs are much harder to solve than linear programs. The computation time of even the best available MIP solvers often increases rapidly with the number of integer variables, although this effect is highly problem-dependent. This is partially caused by the exponential increase in the total number of possible solutions with problem size. For example, a traveling salesman problem with n cities has $n!$ tours, and there are 2^n solutions to a problem with n binary variables (some of which may be infeasible).

In this chapter, we discuss solution approaches for MILP and MINLP that are capable of finding an optimal solution and verify that they have done so. Specifically, we consider branch-and-bound (BB) and outer linearization (OL) methods. BB can be applied to both linear and nonlinear problems, but OL is used for nonlinear problems by solving a sequence of MILPs. Chapter 10 further considers branch-and-bound methods, and also describes heuristic methods, which often find very good solutions but are unable to verify optimality.

9.2 BRANCH-AND-BOUND METHODS USING LP RELAXATIONS

Branch and bound (BB) is a class of methods for linear and nonlinear mixed-integer programming. If carried to completion, it is guaranteed to find an optimal solution to linear and convex nonlinear problems. It is the most popular approach and is currently used in virtually all commercial MILP software (see Chapter 7).

Consider the application of BB to a general MILP problem, in which all the integer variables are binary, that is, either 0 or 1. The problem formed by relaxing the "0 or 1" constraint to "anywhere between 0 and 1" is called the LP relaxation of the MILP. BB starts by solving this LP relaxation. If all discrete variables have integer values, this solution solves the MILP. If not, one or more discrete variables has a fractional value. BB chooses one of these variables in its *branching* step and then creates two LP subproblems by fixing this variable first at 0, then at 1. If either of these subproblems has an integer solution, it need not be investigated further. If its objective value is better than the best value found thus far, it replaces this best value. If either subproblem is infeasible, it need not be investigated further. Otherwise, we find another fractional variable and repeat the steps. A clever *bounding* test can also be applied to each subproblem. If the test is satisfied, the subproblem need not be investigated further. This bounding test, together with the rest of the procedure, is explained in the following example.

EXAMPLE 9.1 BRANCH-AND-BOUND ANALYSIS OF AN INTEGER LINEAR PROGRAM

$$\text{Maximize:} \quad f = 86y_1 + 4y_2 + 40y_3$$

$$\text{Subject to:} \quad 774y_1 + 76y_2 + 42y_3 \le 875$$

$$67y_1 + 27y_2 + 53y_3 \le 875$$

$$y_1, y_2, y_3 = 0, 1$$

We can show the various subproblems developed from the stated problem by a tree (Figure E9.1). The objective function and inequality constraints are the same for each subproblem and so are not shown. The upper bound and lower bound for f are represented by *ub* and *lb*, respectively.

Each subproblem corresponds to a node in the tree and represents a *relaxation* of the original IP. One or more of the integer constraints $y_i = 0$ or 1 are replaced by the *relaxed* condition $0 \le y_i \le 1$, which includes the original integers, but also all of the real values in between.

Node 1. The first step is to set up and solve the relaxation of the binary IP via LP. The optimal solution has one fractional (noninteger) variable (y_2) and an objective function value of 129.1. Because the feasible region of the relaxed problem includes the feasible region of the initial IP problem, 129.1 is an *upper bound* on the value of the objective function of the IP. If we knew a feasible binary solution, its objective value would be a *lower bound* on the value of the objective function, but none is assumed here, so the lower bound is set to $-\infty$. There is as yet no *incumbent*, which is the best feasible integer solution found thus far.

At node 1, y_2 is the only fractional variable, and hence any feasible integer solution must satisfy either $y_2 = 0$ or $y_2 = 1$. We create two new relaxations represented by nodes 2 and 3 by imposing these two integer constraints. The process of creating these two relaxed subproblems is called *branching*. The feasible regions of these two LPs are

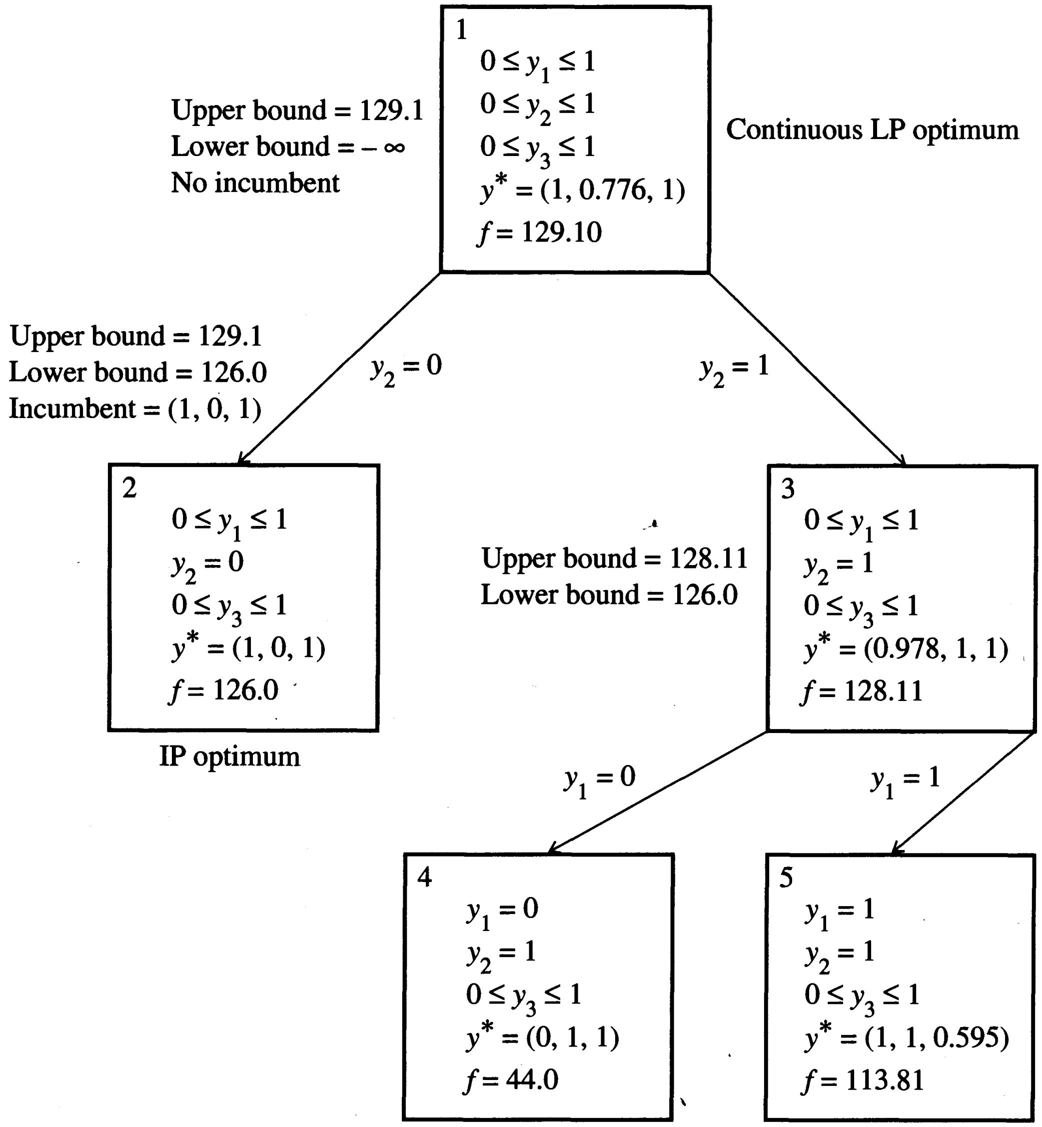

FIGURE E9.1
Decomposition of Example 9.1 via the branch-and-bound method.

partitions of the feasible region of the original IP, and one (or both) contain an optimal integer solution, if one exists (the problem may not have a feasible integer solution).

If the relaxed IP problem at a given node has an optimal binary solution, that solution solves the IP, and there is no need to proceed further. This node is said to be *fathomed*, because we do not need to branch from it. If a relaxed LP problem has several fractional values in the solution, you must select one of them to branch on. It is important to make a good choice. Branching rules have been studied extensively (see Nemhauser and Wolsey, 1988). Finally, if the node 1 problem has no feasible solution, the original IP is infeasible. At this point, the two nodes resulting from branching are unfathomed, and you must decide which to process next. How to make the decision has been well studied (Nemhauser and Wolsey, 1988, Chapter II.4).

Node 2. For this example we choose node 2 and find that the solution to the relaxed problem is a binary solution, so this node is now fathomed. The solution is the first feasible integer solution found, so its objective value of 126.0 becomes the current lower bound. The difference $(ub - lb)$ is called the "*gap*," and its value at this stage is $129.1 - 126.0 = 3.1$. It is common to terminate the BB algorithm when

$$\frac{\text{Gap}}{1.0 + |lb|} \leq tol \tag{9.1}$$

When the gap is smaller than some fraction *tol* of the incumbent's objective value (the factor 1.0 ensures that the test makes sense when $lb = 0$). When $lb = -\infty$, you will always satisfy Equation 9.1. A tol value of 10^{-4} would be a tight tolerance, 0.01 would be neither tight nor loose, and 0.03 or higher would be loose. The termination criterion used in the Microsoft Excel Solver has a default *tol* value of 0.05.

Node 3. The solution of the problem displayed in node 3 is fractional with a value of the objective function equal to 128.11, so the upper bound for this node and all its successors is 128.11. The gap is now 2.11, so $gap/[1 + \text{abs}(lb)] = 0.0166$. If tol in Equation (9.1) is larger than this, the BB algorithm stops. Otherwise, we create two new nodes by branching on y_1.

Node 4. Node 4 has an integer solution, with an objective function value of 44, which is smaller than that of the incumbent obtained previously. The incumbent is unchanged, and this node is fathomed.

Node 5. Node 5 has a fractional solution with an objective function value of 113.81, which is smaller than the lower bound of 126.0. Any successors of this node have objective values less than or equal to 113.81 because their LP relaxations are formed by adding constraints to the current one. Hence we can never find an integer solution with objective value higher than 126.0 by further branching from node 5, so node 5 is fathomed. Because there are no dangling nodes, the problem is solved, with the optimum corresponding to node 2.

EXAMPLE 9.2 BLENDING PRODUCTS INCLUDING DISCRETE BATCH SIZES

In this example we have two production units in a plant designated number 1 and number 2, making products 1 and 2, respectively, from the three feedstocks as shown in Figure E9.2a. Unit 1 has a maximum capacity of 8000 lb/day, and unit 2 of 10,000 lb/day. To make 1.0 lb of product 1 requires 0.4 lb of A and 0.6 lb of B; to make 1.0 lb of product 2 requires 0.3 lb of B and 0.7 lb of C. A maximum of 6000 lb/day of B is available, but there are no limits on the available amounts of A and C. Assume the net revenue after expenses from the manufacture of product 1 is \$0.16/lb, and of product 2 is \$0.20/lb. How much of products 1 and 2 should be produced per day, assuming that each must be made in batches of 2000 lb?

This problem is best formulated by scaling the production variables x_1 and x_2 to be in thousands of pounds per day, and the objective function to have values in thousands of dollars per day. This step ensures that all variables have values between 0 and 10 and often leads to both faster solutions and more readable reports. We formulate this problem as the following mixed-integer linear programming problem:

$$\text{Maximize:} \quad f = 0.16x_1 + 0.2x_2$$

$$\text{Subject to:} \quad x_i = 2y_i \qquad i = 1,2 \qquad (a)$$

$$0.6x_1 + 0.3x_2 \leq 6 \qquad (b)$$

$$0 \leq y_1 \leq 4 \qquad 0 \leq y_2 \leq 5 \qquad y_i \text{ integer} \qquad (c)$$

Constraints (a) ensure that the scaled production amounts are even integers because the y_i are general integers subject to the bounds (c). The bounds on x_i are also implied by (a) and (c), and the x_i need not be declared an integer because they will be an integer if the y_i are.

A BB tree for this problem is in Figure E9.2b. The numbers to the left of each node are the current upper and lower bounds on the objective function, and the values to the right are the (y_1, y_2) values in the optimal solution to the LP relaxation at the node. The solution at node 1 has y_1 fractional, so we branch on y_1, leading to nodes 2 and 3. If node 2 is evaluated first, its solution is an integer, so the node is fathomed, and (2, 5) becomes the incumbent solution. This solution is optimal, but we do not

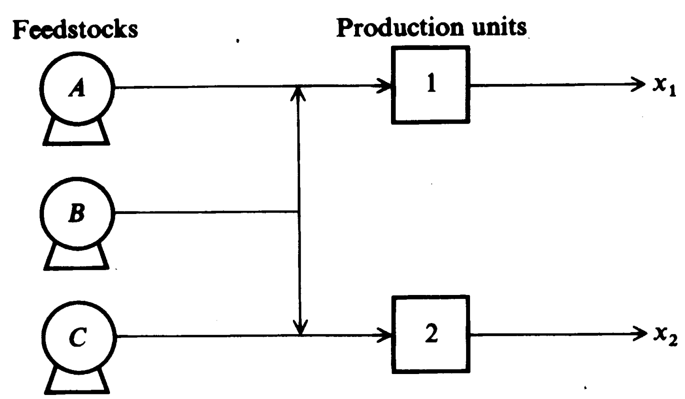

FIGURE E9.2a
Flow chart of a batch plant.

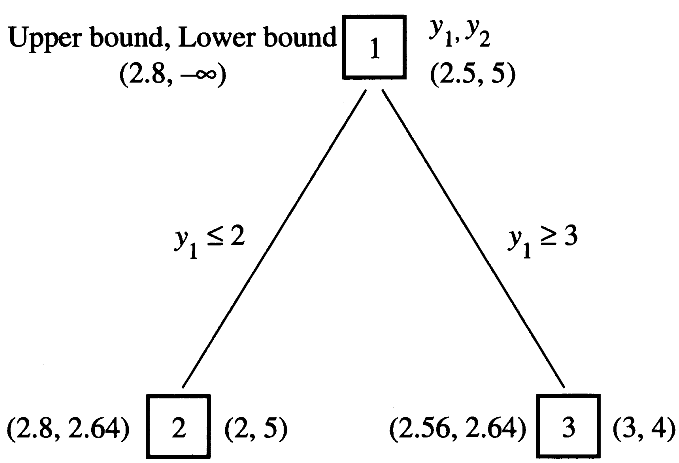

FIGURE E9.2b
Branch-and-bound tree.

	A	B	C	D	E	F	G	H
1	Example 9.2							
2		DECISION VARIABLES						
3								
4			x1	x2	y1	y2		
5		values	4	10	2	5		
6		bounds	8	10	4	5		
7		CONSTRAINTS						
8								
9							value	rhs
10		x1-2y1=0	1		-2		0	0
11		x2-2y2=0		1		-2	0	0
12		.6x1+.3x2<=6	0.6	0.3			5.4	6
13								
14		OBJECTIVE						
15								
16		max .16x1+.2x2	0.16	0.2			2.64	

FIGURE E9.2c
Excel formulation for Example 9.2. Permission by Microsoft.

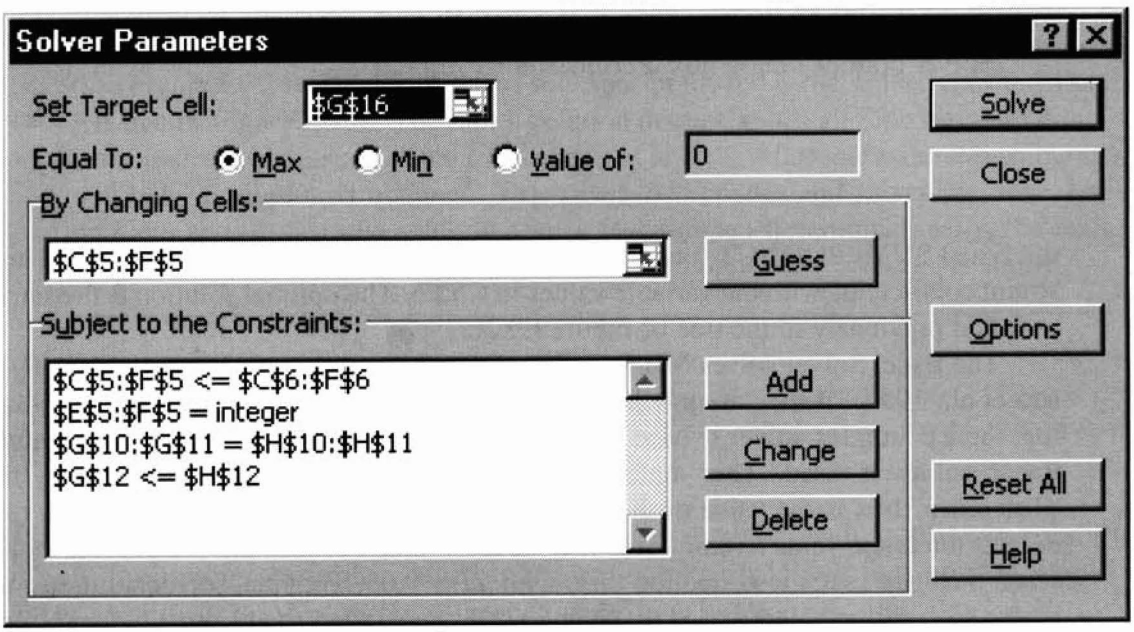

FIGURE E9.2d
Solver dialog for Example 9.2. Permission by Microsoft.

know that yet. Evaluating node 3, its solution is also an integer, so it is fathomed. Its solution has an objective function value of 2.56, smaller than the incumbent, so (2, 5) has been *proven* optimal. It is possible for a BB algorithm to discover an optimal solution at an early stage, but it may take many more steps to prove that it is optimal.

An Excel spreadsheet formulation of this problem is shown in Figures E9.2c and E9.2d. The constraint coefficient matrix is in the range C10:F12 and G10:G12 contains formulas that compute the values of the constraint functions. These formulas use

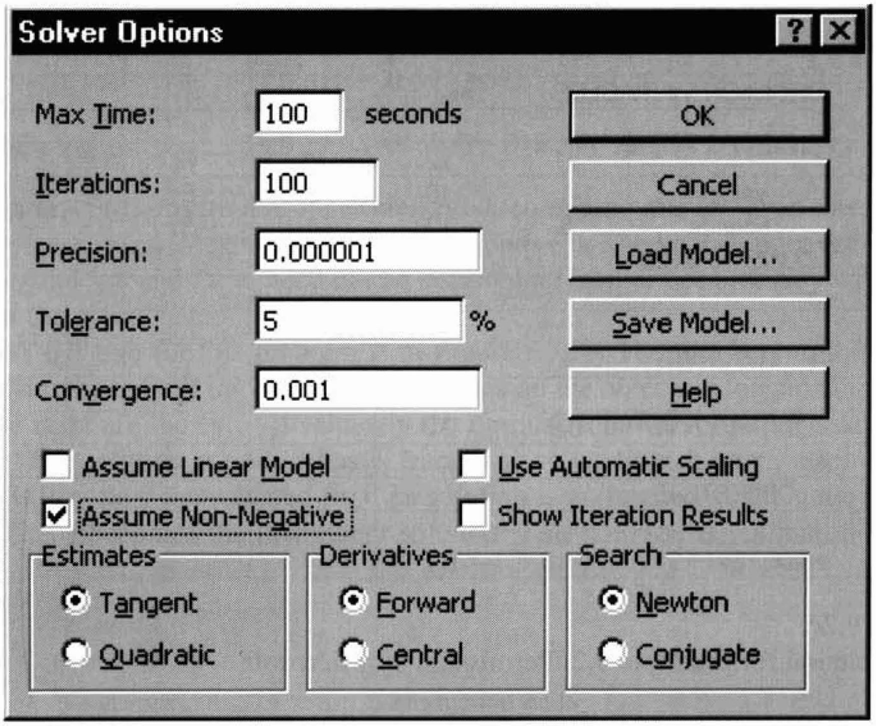

FIGURE E9.2e
Solver options dialog box. Permission by Microsoft.

the Excel SUMPRODUCT function to compute the inner product of the row of constraint coefficients with the variable values in C5:F5. The optimal solution is the same as found previously in the tree of Figure E9.2b.

The Excel Solver solves MILP and MINLP problems using a BB algorithm (Fylstra et al., 1998). If the "assume linear model" box is checked in the OPTIONS dialog, the LP simplex solver is used to solve the LP relaxations; if not, the GRG2 nonlinear solver is used. This dialog is shown in Figure E9.2e. The value in the "Tolerance" box is the value of the tol in Equation (9.1). As shown in the figure, the default tolerance value is 0.05. This is a "loose" value because the BB process stops when the "*gap*" satisfies Equation (9.1) with $tol = 0.05$. The final solver solution can have an objective value that is as much as 5% worse than the optimal value. Users who are unaware of the meaning of the tolerance setting often assume that this final solution is optimal. For problems with few integer variables, you can safely use a tighter tolerance, for example, 0.1%, because such problems are usually solved quickly. For larger problems (e.g., more than 20 binary or integer variables), you can solve first with a loose tolerance. If this effort succeeds quickly, try again with a smaller tolerance.

If you request a sensitivity report after the solver has solved this example, the message "Sensitivity report and limits report are not meaningful for problems with integer constraints" appears (try it and see). A sensitivity report is "not meaningful" for a mixed-integer problem because Lagrange multipliers may not exist for such problems. To see why, recall that, in a problem with no integer variables, the Lagrange multiplier for a constraint is the derivative of the optimal objective value (OV) within the OV. In other words, the OV function may not be differentiable at some points. As an example, consider the constraint

$$0.6x_1 + 0.3x_2 \le 6$$

As shown in Figure E9.2c, this constraint is not active at the optimal solution because its left-hand side value is 5.4. Hence if its right-hand side is changed from 6 to 5.4, the optimal solution is unchanged. Now decrease this right-hand side (*RHS*) just a tiny bit further, to 5.3999). The new optimal objective value (*OV*) is 2.32, sharply worse than the *OV* of 2.64 when the *RHS* is 5.4. This OV change occurs because the small *RHS* decrease does not allow both x_1 and x_2 to retain their current values of 4 and 10, respectively. One or both must decrease, and because both are even integers, each must decrease by a value of 2. A small fractional change is not possible. The best possible change is to have $x_1 = 2$ while x_2 remains at 10. The ratio of *OV* change to *RHS* change is

$$\frac{\Delta OV}{\Delta RHS} = \frac{-0.32}{-0.0001} = 3200$$

Clearly as ΔRHS approaches zero the limit of this ratio does not exist; the ratio approaches infinity because ΔOV remains -0.32. Hence the function $OV\,(RHS)$ is not differentiable at $RHS = 5.4$, so no Lagrange multiplier exists at this point.

We now ask the reader to start Excel, either construct or open this model, and solve it after checking the "Show Iteration Results" box in the Solver Options dialog (see Figure E9.2d). The sequence of solutions produced is the same as is shown in the BB tree of Figure E9.2b. The initial solution displayed has all four variables equal to zero, indicating the start of the LP solution at node 1. After a few iterations, the optimal node 1 solution is obtained. The solver then creates and solves the node 2 subproblem and displays its solution after a few simplex iterations. Finally, the node 3 subproblem is created and solved, after which an optimality message is shown.

9.3 SOLVING MINLP PROBLEMS USING BRANCH-AND-BOUND METHODS

Many problems in plant design and operation involve both nonlinear relations among continuous variables, and binary or integer variables that appear linearly. The continuous variables typically represent flows or process operating conditions, and the binary variables are usually introduced for yes–no decisions. Such problems can be written in the following general form:

$$\text{Minimize:} \quad z = f(\mathbf{x}) + \mathbf{c}^T\mathbf{y} \tag{9.2}$$

$$\text{Subject to:} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0} \tag{9.3}$$

$$\mathbf{g}(\mathbf{x}) + \mathbf{M}\mathbf{y} \le \mathbf{0} \tag{9.4}$$

$$\mathbf{x} \in X, \quad \mathbf{y} \in Y \tag{9.5}$$

where $\mathbf{x}$ is the vector of continuous variables, $\mathbf{y}$ is the vector of integer (usually binary) variables, $\mathbf{M}$ is a matrix, and X and Y are sets. The $\mathbf{y}$'s are typically chosen to control the continuous variables $\mathbf{x}$ by either forcing one (or more) variables to be zero or by allowing them to assume positive values. The choice of $\mathbf{y}$ should be done in such a way that $\mathbf{y}$ appears linearly, because then the problem is much easier to solve. The constraints (9.3) represent mass and energy balances, process input–output transformations, and so forth. The inequalities (9.4) are formulated so that $\mathbf{y}$ influences $\mathbf{x}$ in the desired way—we illustrate how to do this in several examples that follow. The set X is specified by bounds and other inequalities involving $\mathbf{x}$ only, whereas Y is defined by conditions that the components of $\mathbf{y}$ be binary or integer, plus other inequalities or equations involving $\mathbf{y}$ only.

As discussed in Section 9.2, the Excel Solver uses a BB algorithm to solve MILP problems. It uses the same method to solve MINLP problems. The only difference is that for MINLP problems the relaxed subproblems at the nodes of the BB tree are continuous variable NLPs and must be solved by an NLP method. The Excel Solver uses the GRG2 code to solve these NLPs. GRG2 implements a GRG algorithm, as described in Chapter 8.

BB methods are guaranteed to solve either linear or nonlinear problems if allowed to continue until the "gap" reaches zero [see Equation (9.1)], provided that a global solution is found for each relaxed subproblem at each node of the BB tree. A global optimum can always be found for MILPs because both simplex and interior point LP solvers find global solutions to LPs because LPs are convex programming problems. In MINLP, if each relaxed subproblem is smooth and convex, then every local solution is a global optimum, and for these conditions many NLP algorithms guarantee convergence to a global solution.

Sufficient conditions on the functions in the general MINLP in Equations (9.2)–(9.5) to guarantee convexity of each relaxed subproblem are

1. The objective term $f(\mathbf{x})$ is convex.
2. Each component of the vector of equality constraint functions $\mathbf{h}(\mathbf{x})$ is linear.
3. Each component of the vector of inequality constraint functions $\mathbf{g}(\mathbf{x})$ is convex over the set X.
4. The set X is convex.
5. The set Y is determined by linear constraints and the integer restrictions on $\mathbf{y}$.

If these conditions hold, and an arbitrary subset of y variables are fixed at integer values and the integer restrictions on the remaining y's are relaxed, the resulting continuous subproblem (in the x and relaxed y variables) is convex. Although many practical problems meet these conditions, unfortunately many do not, often because some of the equality constraint functions $\mathbf{h}(\mathbf{x})$ are nonlinear. Then you cannot guarantee that the feasible region of each relaxed subproblem is convex, so local solutions may exist that are not global solutions. Consequently, a local NLP solver may terminate at a local solution that is not global in some tree node, and, in a minimization problem, the objective function value (call it "local") is larger than the true optimal value. When the "local" value is tested to see if it exceeds the current upper bound, it may pass this test, and the node will be classified as "fathomed." No further branches are allowed from this node. The "fathomed" classification is false if the true global optimal value at the node is less than the current upper bound. Thus, the BB procedure fails to find any better solutions reached by further branching from this node. A nonoptimal solution to the MINLP may result.

EXAMPLE 9.3 OPTIMAL SELECTION OF PROCESSES

This problem, taken from Floudas (1995), involves the manufacture of a chemical C in process 1 that uses raw material B (see Figure E9.3a). B can either be purchased or manufactured via two processes, 2 or 3, both of which use chemical A as a raw material. Data and specifications for this example problem, involving several nonlinear input–output relations (mass balances), are shown in Table E9.3A. We want to determine which processes to use and their production levels in order to maximize profit. The processes represent design alternatives that have not yet been built. Their fixed costs include amortized design and construction costs over their anticipated lifetime, which are incurred only if the process is used.

To model this problem as a MINLP problem, we first assign the continuous variables to the different streams to represent the flows of the different chemicals. $A2$ and $A3$ are the amounts of A consumed by processes 2 and 3, $B2$ and $B3$ are the amounts of B produced by these processes, BP is the amount of B purchased in an external market, and $C1$ is the amount of C produced by this process. We also define the 0–1 variables, $Y1$, $Y2$, and $Y3$ to represent the existence of each of the processes.

The constraints in this problem are

1. Conversion

$$C1 = 0.9B1$$

$$B2 = \ln(1 + A2)$$

$$B3 = 1.2\ln(1 + A3) \qquad (a)$$

2. Mass balance for B

$$B1 = B2 + B3 + BP \qquad (b)$$

The specifications and limits that apply are as follows:

3. Nonnegativity condition for continuous variables

$$A2, A3, B1, B2, B3, BP, C1 \geq 0 \qquad (c)$$

4. Integer constraints

$$Y1, Y2, Y3 = 0 \text{ or } 1 \qquad (d)$$

5. Maximum demand for C

$$C1 \leq 1 \qquad (e)$$

6. Limits on plant capacity

$$B2 \leq 4Y2$$

$$B3 \leq 5Y3$$

$$C1 \leq 2Y1 \qquad (f)$$

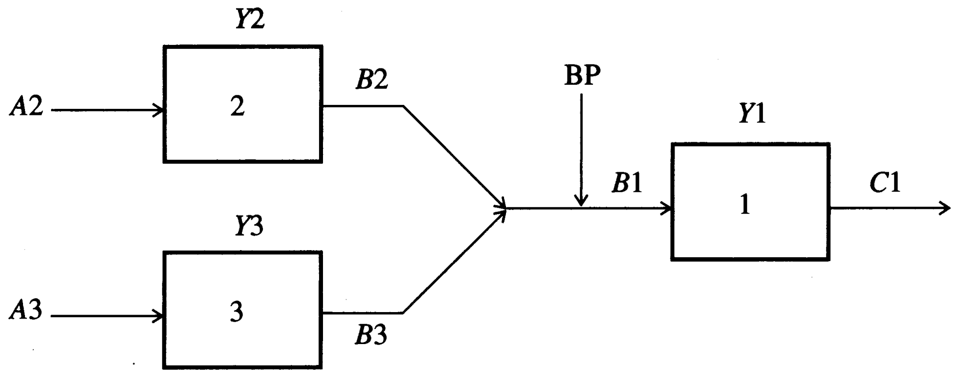

FIGURE E9.3a
Process diagram.

TABLE E9.3A
Problem data

Conversions:	Process 1		$C = 0.9B$
	Process 2		$B = \ln(1 + A)$
	Process 3		$B = 1.2 \ln(1 + A)$
			(A, B, C, in ton/h)
Maximum capacity:	Process 1		2 ton/h of C
	Process 2		4 ton/h of B
	Process 3		5 ton/h of B
Prices:	A:	$ 1,800/ton	
	B:	$ 7,000/ton	
	C:	$13,000/ton	
Demand of C:	1 ton/h maximum		

		Fixed (10^3 \$/h)	*Variable* (10^3 \$/ton of product)
Costs:	Process 1	3.5	2
	Process 2	1	1
	Process 3	1.5	1.2

Note that the constraints in (f) place an upper limit of zero on the amounts produced if a process is not selected and impose the true upper limit if the process is selected. Clearly, with the bounds in step 3, this means that the amounts of $B2$, $B3$, and $C1$ are zero when their binary variables are set to zero. If a binary variable is one, the amounts produced can be anywhere between zero and their upper limits.

Finally, for the objective function, the terms for the profit PR expressed in \$$10^3$/h are given as follows:

1. Income from sales of product C: $13C$
2. Expense for the purchase of chemical B: $7BP$

3. Expense for the purchase of chemical A: $1.8A2 + 1.8A3$
4. Annualized investment or fixed cost for the three processes:

$$3.5Y1 + 2C1 + Y2 + B2 + 1.5Y3 + 1.2B3$$

Note that in the preceding expression the fixed charges are multiplied by the binary variables so that these charges are incurred only if the corresponding process is selected. Combining the preceding terms yields the following objective function:

$$\text{Maximize } PR = 11C1 - 3.5Y1 - Y2 - B2 - 1.5Y3 - 1.2B3$$
$$-7BP - 1.8A2 - 1.8A3 \qquad (g)$$

Relations (a)–(g) define the MINLP problem. It is important to note that the relations between the binary and continuous variables in Equation (f) are *linear*. It is possible to impose the desired relations nonlinearly. For example, one could replace $C1$ by $C1 * Y1$ everywhere $C1$ appears. Then if $Y1 = 0$, $C1$ does not appear, and if $Y1 = 1$, $C1$ does appear. Alternatively, one could replace $C1$ by the conditional expression (if $Y1 = 1$ then $C1$ else 0). Both these alternatives create nonlinear models that are very difficult to solve and should be avoided if possible.

Solution. Figure E9.3b shows the implementation of the MINLP problem in Excel Solver. The input–output relations (a) are in cells F18:F20, and the mass balance (b) is in F22, both written in the form "$f(\mathbf{x}) = 0$." The left- and right-hand sides of the plant capacity limits (f) are in C25:C27 and F25:F27, respectively. The Solver parameter dialog box is in Figure E9.3c. Nonnegativity constraints are imposed by checking the "Assume Nonnegative" box in the options dialog box.

The optimal solution has $Y1 = Y3 = 1$, $Y2 = 0$, so only processes 1 and 3 are used. Because $BP = 0$, there is no purchase of chemical B from an outside source. Total costs are 11.077 (in thousands of dollars per hour), revenues are 13, and the maximum profit is 1.923.

Given the optimal result, we now can ask a number of questions about the process operations, such as

1. Why is process 3 used instead of 2?
2. What happens if the cost of chemical A changes?
3. Why is no B purchased?

These questions can be answered, respectively, by carrying out the following steps:

1. Rerun the base case with $Y2$ fixed at 1 and $Y3$ at 0, thus forcing process 2 to be used rather than 3 while optimizing over the continuous flow variables.
2. Change the cost of A, and reoptimize.
3. Change the cost of purchased B, and reoptimize.

The link between the Excel Solver and the Excel Scenario Manager makes saving and reporting case study information easier. After solving each case, click the "Save Scenario" button on the dialog box that contains the optimality message, which invokes the Excel Scenario Manager. This stores the current decision variable values in a scenario named by the user. After all of the desired scenarios are generated, you can produce the Scenario Summary shown in Table E9.3B by selecting "Scenario Manager" from the Tools menu and choosing "Summary" from the Scenario Manager dialog.

	A	B	C	D	E	F
1	**PROCESS MODEL SOLVED BY THE EXCEL SOLVER (BRANCH AND BOUND)**					
2						
3	**CONTINUOUS VARIABLES**					
4	1. a2=consumption of chemical A in process 2					0.00E+00
5	2. a3=consumption of chemical A in process 3					1.52
6	3. b2=production of chemical B in process 2					0.00E+00
7	4. b3=production of chemical B in process 3					1.111
8	5. bp=amount of B purchased in external market					0.00E+00
9	6. b1=consumption of B by process 1					1.111
10	7. c1=amount of C produced by process 1					1
11	**BINARY VARIABLES**					
12	1. y1=on/off for process 1					1
13	2. y2=on/off for process 2					0
14	3. y3=on/off for process 3					1
15	**CONSTRAINTS**					
16	1. PROCESS OUTPUTS					
17						
18	inout1					0.00E+00
19	inout2					0.00E+00
20	inout3					0.00E+00
21	2. MASS BALANCE					
22	mbal					0.00E+00
23	3. TURN PROCESS OUTPUTS ON OR OFF					
24			Output	Capacity	Binary	Capacity*Binary
25	process 1		1	2	1	2
26	process 2		0.00E+00	4	0	0
27	process 3		1.111	5	1	5
28	4. OUTPUT LIMIT					
29			Value of c1	Limit		
30	limit on c1		1	1		
31	**OBJECTIVE**		Per unit	Total		
32	revenue		13	13		
33	fixed cost					
34	y1		3.5	3.5		
35	y2		1	0		
36	y3		1.5	1.5		
37	**subtotal**			5		
38	operating cost					
39	c1		2	2		
40	b2		1	0		
41	b3		1.2	1.333		
42	**subtotal**			3.333		
43	raw materials costs					
44	a2 cost		1.8	0.000		
45	a3 cost		1.8	2.744		
46	bp cost		7	0.000		
47	**subtotal**			2.744		
48	total cost			11.077		
49	**net profit**			1.923		

FIGURE E9.3b
Excel Solver model. Permission by Microsoft.

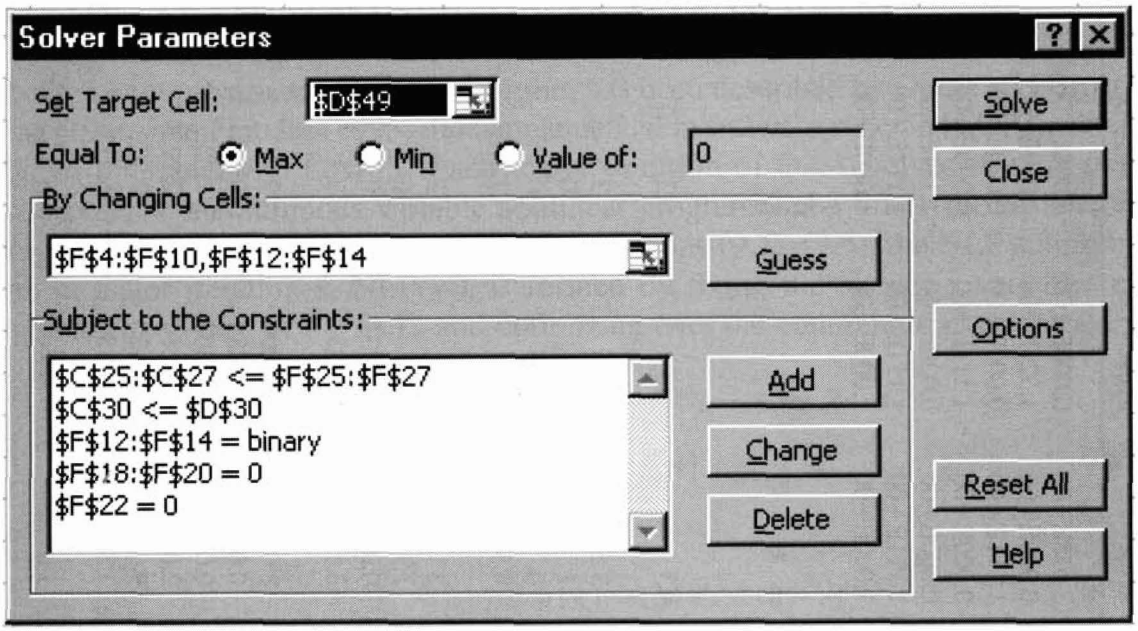

FIGURE E9.3c
Solver parameter dialog box. Permission by Microsoft.

Examination of the "2 instead of 3" column in Table E9.3B shows that process 3 has higher fixed and variable operating costs than 2 (3.33 compared with 3.11) but is more efficient because its output of B is 1.2 times that of process 2. This higher efficiency leads to lower raw material costs for chemical A ($A3$ cost is 2.744, and $A2$ cost is 3.668). This more than offsets the higher operating cost, leading to lower total costs and a larger net profit. This analysis clearly shows that the choice between processes 2 and 3 depends on the cost of A. If the cost of A is reduced enough, process 2 should be preferred. The two "*a*cost" columns in Table E9.3B show that a reduction of A's cost to 1.5 reduces the cost but leaves process 3 as the best choice, but a further reduction to 1.0 switches the optimal choice to process 2.

The last row of Table E9.3B shows why no chemical B is purchased. The cost per unit of B produced is computed by adding the cost of A purchased to the sum of the fixed and variable operating costs (processes 2 and 3) and dividing by the amount of B produced. In the base case this cost is \$2555/ton, so that the market price of B must be lower than this value for an optimal solution to choose purchasing B to producing it. The current price of B is 7, far above this threshold. The "*BP*cost=2" column of Table E9.3B shows that if B's market price is reduced to 2, the maximum profit is attained by shutting down both processes 2 and 3 and purchasing B.

Of course, a BB method can find an optimal solution even when the MINLP does not satisfy the convexity conditions. That occurred in Example 9.3, even though the equality constraints were nonlinear. The GRG2 solver did find global solutions at each node. An optimal solution cannot be guaranteed for nonconvex MINLPs, however, if a local NLP solver is used. As global optimization methods improve, future BB software may include a global NLP solver and thus ensure optimality. Currently, the main drawback to using a global optimizer in a BB algorithm is the long time required to find a global solution to even moderately-sized nonconvex NLPs.

Scenario summary

Changing cells	Base	2 instead of 3	*acost*=1	*acost*=1.5	*BP*cost=2
1. *A*2=consumption of chemical *A* in process 2	0.000E+00	2.038E+00	2.038E+00	0.000E+00	0.000E+00
2. *A*3=consumption of chemical *A* in process 3	1.524E+00	0.000E+00	0.000E+00	1.524E+00	0.000E+00
3. *B*2=production of chemical *B* by process 2	0.000E+00	1.111E+00	1.111E+00	0.000E+00	0.000E+00
4. *B*3=production of chemical *B* by process 3	1.111E+00	0.000E+00	0.000E+00	1.111E+00	0.000E+00
5. *BP*=amount of *B* purchased in external market	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.111E+00
6. *B*1=consumption of *B* by process 1	1.111E+00	1.111E+00	1.111E+00	1.111E+00	1.111E+00
7. *C*1=amount of *C* produced by process 1	1.000E+00	1.000E+00	1.000E+00	1.000E+00	1.000E+00
1. *Y*1 = on-off for process 1	1	1	1	1	1
2. *Y*2 = on-off for process 2	0	1	1	0	0
3. *Y*3 = on-off for process 3	1	0	0	1	0
Result cells					
Revenue	13	13	13	13	13
Fixed cost					
*Y*1	3.5	3.5	3.5	3.5	3.5
*Y*2	0	1	1	0	0
*Y*3	1.5	0	0	1.5	0
Subtotal	5	4.5	4.5	5	3.5
Operating cost					
*C*1	2	2	2	2	2
*B*2	0	1.111	1.111	0	0
*B*3	1.333	0.000	0.000	1.333	0
Subtotal	3.333	3.111	3.111	3.333	2
Raw material costs					
*A*2 cost	0.000	3.668	2.038	0.000	0
*A*3 cost	2.744	0.000	0.000	2.286	0
BP cost	0.000	0.000	0.000	0.000	2.222
Subtotal	2.744	3.668	2.038	2.286	2.222
Total cost	11.077	11.279	9.649	10.620	7.722
Net profit	1.923	1.721	3.351	2.380	5.278
Unit cost of *B* produced	2.55	1.9	1.9	2.55	0

9.4 SOLVING MINLPs USING OUTER APPROXIMATION

The "outer approximation" (OA) algorithm has been described by Duran and Grossman (1986) and Floudas (1995). It is implemented in software called DICOPT, which has an interface with GAMS. Each major iteration of OA involves solving two subproblems: a continuous variable nonlinear program and a linear mixed-integer program. Using the problem statement in Equations (9.2)–(9.5), the NLP subproblem at major iteration k, NLP($\mathbf{y}^k$), is formed by fixing the integer y variables at some set of values, say $\mathbf{y}^k \in Y$, and optimizing over the continuous x variables;

Problem NLP ($\mathbf{y}^k$)

$$\text{Maximize:} \quad \mathbf{c}^T\mathbf{y}^k + f(\mathbf{x}) \tag{9.6}$$

$$\begin{aligned} \text{Subject to:} \quad & \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{g}(\mathbf{x}) + \mathbf{M}\mathbf{y}^k \le \mathbf{0} \\ & \mathbf{x} \in X \end{aligned} \tag{9.7}$$

We redefined the sense of the optimization to be maximization. The optimal objective value of this problem is a lower bound on the MINLP optimal value. The MILP subproblem involves both the $\mathbf{x}$ and $\mathbf{y}$ variables. At iteration k, it is formed by linearizing all nonlinear functions about the optimal solutions of each of the subproblems NLP $(\mathbf{y}^i), i = 1, \dots, k$, and keeping all of these linearizations. If $\mathbf{x}^i$ solves NLP($\mathbf{y}^i$), the MILP subproblem at iteration k is

MILP subproblem

$$\text{Maximize:} \quad \mathbf{c}^T\mathbf{y} + z \tag{9.8}$$

$$\begin{aligned} \text{Subject to:} \quad & z \ge f(\mathbf{x}^i) + \nabla f^T(\mathbf{x}^i)(\mathbf{x} - \mathbf{x}^i), \quad i = 1, \dots, k \\ & \mathbf{h}(\mathbf{x}^i) + \nabla \mathbf{h}^T(\mathbf{x}^i)(\mathbf{x} - \mathbf{x}^i) = \mathbf{0}, \quad i = 1, \dots, k \\ & \mathbf{g}(\mathbf{x}^i) + \nabla \mathbf{g}^T(\mathbf{x}^i)(\mathbf{x} - \mathbf{x}^i) + \mathbf{M}\mathbf{y}^i \le \mathbf{0}, \quad i = 1, \dots, k \\ & \mathbf{x} \in X, \quad \mathbf{y} \in Y \end{aligned} \tag{9.9}$$

The new variable z is introduced to make the objective linear.

$$\begin{aligned} \text{Minimize:} \quad & \mathbf{c}^T\mathbf{y} + z \\ \text{Subject to:} \quad & z \ge f(\mathbf{x}) \end{aligned}$$

is equivalent to minimizing $\mathbf{c}^T\mathbf{y} + f(\mathbf{x})$. Duran and Grossman (1986) and Floudas (1995) show that if the convexity assumptions (1)–(5) of Section 9.3 hold, then the optimal value of this MILP subproblem is an upper bound on the optimal MINLP objective value. Because a new set of linear constraints is added at each iteration, this upper bound decreases (or remains the same) at each iteration. Under the convexity

TABLE 9.1
DICOPT iteration log

Major step	Major iteration	Objective function	CPU time (s)	NLP or MILP iterations	Solver
NLP	1	5.32542	0.17	8	CONOPT2
MILP	1	2.44260	0.16	16	OSL
NLP	2	1.72097	0.11	3	CONOPT2
MILP	2	2.20359	0.16	17	OSL
NLP	3	1.92310	0.11	3	CONOPT2
MILP	3	1.44666	0.17	24	OSL
NLP	4	1.41100	0.11	8	CONOPT2

assumptions, the upper and lower bounds converge to the true optimal MINLP value in a finite number of iterations, so the OA algorithm solves the MINLP problem.

Table 9.1 shows how outer approximation, as implemented in the DICOPT software, performs when applied to the process selection model in Example 9.3. Note that this model does not satisfy the convexity assumptions because its equality constraints are nonlinear. Still DICOPT does find the optimal solution at iteration 3. Note, however, that the optimal MILP objective value at iteration 3 is 1.446, which is *not* an upper bound on the optimal MINLP value of 1.923 because the convexity conditions are violated. Hence the normal termination condition that the difference between upper and lower bounds be less than some tolerance cannot be used, and DICOPT may fail to find an optimal solution. Computational experience on nonconvex problems has shown that retaining the best feasible solution found thus far, and stopping when the objective value of the NLP subproblem fails to improve, often leads to an optimal solution. DICOPT stopped in this example because the NLP solution at iteration 4 is worse (lower) than that at iteration 3.

The NLP solver used by GAMS in this example is CONOPT2, which implements a sparsity—exploiting GRG algorithm (see Section 8.7). The mixed-integer linear programming solver is IBM's Optimization Software Library (OSL). See Chapter 7 for a list of commercially available MILP solvers.

9.5 OTHER DECOMPOSITION APPROACHES FOR MINLP

Generalized Benders decomposition (GBD), derived in Geoffrion (1972), is an algorithm that operates in a similar way to outer approximation and can be applied to MINLP problems. Like OA, when GBD is applied to models of the form (9.2)–(9.5), each major iteration is composed of the solution of two subproblems. At major iteration k, one of these subproblems is NLP($\mathbf{y}^k$), given in Equations (9.6)–(9.7). This is an NLP in the continuous variables $\mathbf{x}$, with $\mathbf{y}$ fixed at $\mathbf{y}^k$. The other GBD subproblem is an integer linear program, as in OA, but it only involves the discrete variables $\mathbf{y}$, whereas the MILP of OA involves both $\mathbf{x}$ and $\mathbf{y}$. The constraints of the GBD MIP subproblem are different from those in the OA MILP subproblem. These constraints are called generalized Benders cuts. They are linear constraints, formed using the Lagrange multipliers of the continuous subproblem, NLP($\mathbf{y}^k$). Only one GBD cut is added at each major iteration. In OA, an entire set of linearized constraints of the form (9.9) is added each time, so the OA MILP subproblems has many more constraints than those in GBD. Each solution of the NLP subproblem in GBD generates a lower bound on the maximum objective value, and the MILP subproblem yields an upper bound. Duran and Grossman (1986) proved that for convex MINLP problems the OA upper bound is never above the GBD lower bound [see also Floudas (1995)]. Hence, for convex problems OA terminates in fewer major iterations than GBD. The OA computing time may not be smaller than that for GBD, however, because the OA subproblems have more constraints and thus usually take longer to solve.

9.6 DISJUNCTIVE PROGRAMMING

A disjunctive program is a special type of MINLP problem whose constraints include the condition that exactly one of several sets of constraints must be satisfied (Raman and Grossmann, 1994). Defining $\vee$ as the logical "exclusive or" operator and Y_i as logical variables (whose values are *true* or *false*), an example of a disjunctive program, taken from Lee and Grossman (2000), is

$$\text{Minimize:}\quad (x_1-3)^2+(x_2-2)^2+c$$

$$\text{Subject to:}\quad \begin{bmatrix} Y_1 \\ x_1^2+x_2^2-1\le 0 \\ c=2 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ (x_1-4)^2+(x_2-1)^2-1\le 0 \\ c=1 \end{bmatrix}$$

$$\vee \begin{bmatrix} Y_3 \\ (x_1-2)^2+(x_2-4)^2-1\le 0 \\ c=3 \end{bmatrix}$$

and

$$0\le x_i\le 8,\qquad i=1,2$$

The logical condition, called a *disjunction*, means that exactly one of the three sets of conditions in brackets must be true: the logical variable must be true, the constraint must be satisfied, and c must have the specified value. Note that c appears in the objective function. There are additional constraints on $\mathbf{x}$; here these are simple bounds, but in general they can be linear or nonlinear inequalities. The single inequality constraint in each bracket may be replaced by several different inequalities. There may also be logical constraints on the Y_i variables, but these constraints are not included in this example.

Disjunctions arise when a set of alternative process units is considered during a process design. The following example is taken from Biegler et al. (1997), p. 519. If one of two reactors is to be selected, we may have the conditions:

> If reactor one is selected, then pressure P in the reactor must lie between 10 and 15, and the reactor fixed cost c is 20.
> If reactor two is selected, then pressure P in the reactor must be between 5 and 10 and the reactor fixed cost c is 30.

See Hooker and Grossman (1999) for more details on occurrence of disjunctions in process synthesis problems.

A generalized disjunctive program (GDP) may be formulated as an MINLP, with binary variables y_i replacing the logical variables Y_i. The most common formulation is called the "big-M" approach because it uses a large positive constant denoted by M to relax or enforce the constraints. This formulation of the preceding example follows:

$$\begin{aligned}
\text{Minimize:}\quad & (x_1 - 3)^2 + (x_2 - 2)^2 + 2y_1 + y_2 + 3y_3 \\
\text{Subject to:}\quad & x_1^2 + x_2^2 - 1 \le M(1 - y_1) \\
& (x_1 - 4)^2 + (x_2 - 1)^2 - 1 \le M(1 - y_2) \\
& (x_1 - 2)^2 + (x_2 - 4)^2 - 1 \le M(1 - y_3) \\
& y_1 + y_2 + y_3 = 1
\end{aligned}$$

and

$$y_i = 0 \quad \text{or} \quad 1, \qquad i = 1,2,3, \qquad 0 \le x_i \le 8, \quad i = 1,2$$

When $y_i = 1$, the ith constraint is enforced and the correct value of c is added to the objective. When $y_i = 0$, the right-hand side of the ith constraint is equal to M, so it is never active if M is large enough. The constraint that sets the sum of the y_i equal to 1 ensures that exactly one constraint is enforced.

The big-M formulation is often difficult to solve, and its difficulty increases as M increases. This is because the NLP relaxation of this problem (the problem in which the condition $y_i = 0$ or 1 is replaced by y_i between 0 and 1) is often weak, that is, its optimal objective value is often much less than the optimal value of the MINLP. An alternative to the big-M formulation is described in Lee and Grossman (2000) using an NLP relaxation, which often has a much tighter bound on the optimal MINLP value. A branch-and-bound algorithm based on this formulation performed much better than a similar method applied to the big-M formulation. An outer approximation approach is also described by Lee and Grossmann (2000).

REFERENCES

Biegler, L. T.; I. E. Grossmann; and A. W. Westerberg. *Systematic Methods of Chemical Process Design*. Prentice-Hall, Englewood Cliffs, NJ (1997).

Duran, M. A.; and I. E. Grossmann. "An Outer Approximation Algorithm for a Class of Mixed—Integer Nonlinear Programs." *Math Prog* **36:** 307–339, (1986).

Floudas, C. *Nonlinear and Mixed—Integer Optimization.* Oxford University Press, New York (1995).

Fylstra, D.; L. Lasdon; A. Waren, and J. Watson. "Design and Use of the Microsoft Excel Solver." *Interfaces* **28** (5): 29–55, (Sept-Oct, 1998).

Geoffrion, A. M. "Generalized Benders Decomposition." *J Optim Theory Appl* **10(4):** 237–260 (1972).

Grossmann, I. E.; and J. Hooker. *Logic Based Approaches for Mixed Integer Programming Models and Their Application in Process Synthesis.* FOCAPD Proceedings, AIChE Symp. Ser. **96** (323): 70–83 (2000).

Hillier, F. S.; and G. J. Lieberman. *Introduction to Operations Research,* 4th ed. Holden-Day, San Francisco, CA (1986), p. 582.

Lee, S.; and I. E. Grossmann. "New Algorithms for Nonlinear Generalized Disjunctive Programming." In press, *Comput Chem Engr.*

Murray, J. E.; and T. F. Edgar. "Optimal Scheduling of Production and Compression in Gas Fields." *J Petrol Technol* 109–118 (January, 1978).

Nemhauser, G. L.; and L. A. Wolsey. *Integer and Combinatorial Optimization.* J. Wiley, New York (1988).

Raman, R.; and I. E. Grossmann. "Modeling and Computational Techniques for Logic Based Integer Programming." *Comput Chem Engr* **18:** 563–578 (1994).

Rosenwald, G. W.; and D. W. Green. "A Method for Determining the Optimal Location of Wells." *Soc Petrol Eng J* pp. 44–54 (February, 1974).

SUPPLEMENTARY REFERENCES

Adjiman, C. S.; I. P. Androulakis; and C. A. Floudas. "Global Optimization of MINLP Problems in Process Synthesis and Design." *Comput Chem Eng* **21:** S445–450 (1997).

Crowder, H. P.; E. L. Johnson; and M. W. Padberg. "Solving Large-Scale Zero-One Linear Programming Problems." *Oper Res* **31:** 803–834 (1983).

Galli, M. R.; and J. Cerda. "A Customized MILP Approach to the Synthesis of Heat Recovery Networks Reaching Specified Topology Targets." *Ind Eng Chem Res* **37:** 2479–2486 (1998).

Grossmann, I. E. "Mixed-Integer Optimization Techniques for Algorithmic Process Synthesis." *Adv Chem Eng* **23:** 172–239, (1996).

Grossmann, I. E.; J. A. Caballero; and H. Yeomans. "Mathematical Programming Approaches to the Synthesis of Chemical Process Systems." *Korean J Chem Eng* **16**(4): 407–426 (1999).

Grossmann, I. E.; and Z. Kravanja. "Mixed-Integer Nonliner Programming: A Survey of Algorithms and Applications." In *Large-Scale Optimization with Applications, Part 2: Optimal Design and Control,* L. T. Biegler et al., eds., pp. 73–100, Springer-Verlag, New York (1997).

Mokashi, S. D.; and A. Kokossis. "The Maximum Order Tree Method: A New Approach for the Optimal Scheduling of Product Distribution Lines." *Comput Chem Eng* **21:** S679–684 (1997).

Morari, M.; and I. Grossmann (eds.). *CACHE Process Design Case Studies, Volume 6: Chemical Engineering Optimization Models with GAMS* (October, 1991). CACHE Corporation, Austin, TX.

Parker, R. G.; and R. Rardin. *Discrete Optimization*. Academic Press, New York (1986).
Schrijver, A. *Theory of Linear and Integer Programming*. Wiley-Interscience, New York (1986).
Skrifvars, H.; S. Leyffer; and T. Westerlund. "Comparison of Certain MINLP Algorithms When Applied to a Model Structure Determination and Parameter Estimation Problem." *Comput Chem Eng* **22:** 1829–1835 (1998).
Turkay, M.; and I. E. Grossmann. "Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Networks." *Comput Chem Eng* **20** (8): 959–978 (1996).
Westerlund, T.; H. Skrifvars; I. Harjunkoski, and R. Pom. "An Extended Cutting Plane Method for a Class of Non-convex MINLP Problems." *Comput Chem Eng* **22:** 357–365 (1998).
Xia, Q.; and S. Macchietto. "Design and Synthesis of Batch Plants—MINLP Solution Based on a Stochastic Method." *Comput Chem Eng* **21:** S697–702 (1997).
Yi, G.; Suh, K.; Lee, B.; and E. S. Lee. "Optimal Operation of Quality Controlled Product Storage." *Comput Chem Eng* **24:** 475–480 (2000).
Zamora, J. M.; and I. E. Grossmann. "A Global MINLP Optimization Algorithm for the Synthesis of Heat Exchanger Networks with No Stream Splits." *Comput Chem Eng* **22:** 367–384 (1998).

PROBLEMS

9.1 A microelectronics manufacturing facility is considering six projects to improve operations as well as profitability. Due to expenditure limitations and engineering staffing constraints, however, not all of these projects can be implemented. The following table gives projected cost, staffing, and profitability data for each project.

Project	Description	First-year expenditure ($)	Second-year expenditure ($)	Engineering hours	Net present value ($)
1	Modify existing production line with new etchers	300,000	0	4000	100,000
2	Build new production line	100,000	300,000	7000	150,000
3	Automate new production line	0	200,000	2000	35,000
4	Install plating line	50,000	100,000	6000	75,000
5	Build waste recovery plant	50,000	300,000	3000	125,000
6	Subcontract waste disposal	100,000	200,000	600	60,000

The resource limitations are

First-year expenditure:	$450,000
Second-year expenditure:	$400,000
Engineering hours:	10,000

A new or modernized production line must be provided (project 1 or 2). Automation is feasible only for the new line. Either project 5 or project 6 can be selected, but not both. Determine which projects maximize the net present value subject to the various constraints.

9.2 An electric utility must determine which generators to start up at the beginning of each day. They have three generators with capacities, operating cost, and start-up costs shown in the following table. A day is divided into two periods, and each generator may be started at the beginning of each period. A generator started in period 1 may be used in period 2 without incurring an additional start-up cost. All generators are turned off at the end of the day.

Demand for power is 2500 megawatts (MW) in period 1 and 3500 MW in period 2. Formulate and solve this problem as a mixed-integer linear program. Define the binary variables carefully.

Generator	Fixed start-up cost ($)	Cost per period per megawatt	Generator capacity in each period (MW)
1	2800	5	1900
2	2000	3	1700
3	1900	8	2900

9.3 An electric utility currently has 700 MW of generating capacity and needs to expand this capacity over the next 5 years based on the following demand forecasts, which determine the minimum capacity required.

Year	Minimum capacity (MW)
1	780
2	860
3	950
4	1060
5	1180

Capacity is increased by installing 10-, 50-, or 100-MW generators. The cost of installation depends on the size and year of installation as shown in the following table.

Generator size (MW)	Year 1	Year 2	Year 3	Year 4	Year 5
10	280	230	188	153	135
50	650	538	445	367	300
100	700	771	640	530	430

Once a generator is installed, it is available for all future years. Formulate and solve the problem of determining the amount of new capacity to install each year so that minimum capacities are met or exceeded and total (undiscounted) installation cost is minimized.

9.4 A manufacturing line makes two products. Production and demand data are shown in the following table.

	Product 1	Product 2
Set-up time (hrs)	6	11
Set-up cost ($)	250	400
Production time/unit (h)	0.5	0.75
Production cost/unit ($)	9	14
Inventory holding cost/unit	3	3
Penalty cost for unsatisfied demand/unit ($)	15	20
Selling price ($/unit)	25	35

Demand data

Product	Week 1	Week 2	Week 3	Week 4
1	75	95	60	90
2	20	30	45	30

Total time available (for production and setup) in each week is 80 h. Starting inventory is zero, and inventory at the end of week 4 must be zero. Only one product can be produced in any week, and the line must be shut down and cleaned at the end of each week. Hence the set-up time and cost are incurred for a product in any week in which that product is made. No production can take place while the line is being set up.

Formulate and solve this problem as an MILP, maximizing total net profit over all products and periods.

9.5 A portfolio manager has $100,000 to invest in a list of 20 stocks. She estimates the return from stock i over the next year as $r(i)$, so that if $x(i)$ dollars are invested in stock i at the start of the year, the end of year value is $[1 + r(i)]*x(i)$. Write an MILP model that determines the amounts to invest in each stock in order to maximize end-of-year portfolio value under the following investment policy: no more than $20,000 can be invested in any stock, and if a stock is purchased at all, at least $5000 worth must be purchased.

9.6

$$\begin{aligned} \text{Maximize:} \quad & f(\mathbf{x}) = 75x_1 + 6x_2 + 3x_3 + 33x_4 \\ \text{Subject to:} \quad & 774x_1 + 76x_2 + 22x_3 + 42x_4 \leq 875 \\ & 67x_1 + 27x_2 + 794x_3 + 53x_4 \leq 875 \\ & x_1, x_2, x_3, x_4 \text{ either 0 or 1} \end{aligned}$$

9.7

$$\begin{aligned} \text{Maximize:} \quad & f(\mathbf{x}) = 2x_1 + x_2 \\ \text{Subject to:} \quad & x_1 + x_2 \leq 5 \\ & x_1 - x_2 \geq 0 \\ & 6x_1 + 2x_2 \leq 21 \\ & x_1, x_2 \geq 0 \text{ and integer} \end{aligned}$$

9.8

$$\text{Minimize:} \quad f(\mathbf{x}) = x_1 + 4x_2 + 2x_3 + 3x_4$$

$$\text{Subject to:} \quad -x_1 + 3x_2 - x_3 + 2x_4 \geq 2$$

$$x_1 + 3x_2 + x_3 + x_4 \geq 3$$

$$x_1, x_2 \geq 0 \text{ and integer}$$

$$x_3, x_4 \geq 0$$

9.9 Determine the minimum sum of transportation costs and fixed costs associated with two plants and two customers based on the following data:

Plant	Annual capacity (in thousands)	Annual fixed charges (in 10^4)
1	2	1
2	1	1

Customer (j)	Demand (j)
1	1
2	1

	Customer (j)	
Plant (i)	1	2
1	$\frac{1}{3}$	$\frac{1}{2}$
2	0	1

Hint: The mathematical statement is

$$\text{Minimize:} \quad f(\mathbf{x}) = \sum_i \sum_j C_{ij}^T x_{ij} + \sum_i C_i^F y_i$$

$$\text{Subject to:} \quad \sum_i x_{ij} = D_j, \quad j = 1, \ldots, n$$

$$\sum_j x_{ij} \leq A_i, \quad i = 1, \ldots, m$$

$$x_{ij} - \min\{D_j, A_j\} \leq 0, \quad i = 1, \ldots, m \quad j = 1, \ldots, n$$

$$y_i = 0, 1 \text{ (integers)}, \quad i = 1, \ldots, m$$

where C_i^T = unit transportation cost from plant i to customer j
C_i^F = fixed cost associated with plant i
x_{ij} = quantity supplied to customer j from plant i
y_i = 1 (plant operates); = 0 (plant is closed)
A_i = capacity of plant i
D_j = demand of customer j

9.10 Four streams are to be allocated to four extractors. The costs of each stream are

	Extractor			
Stream	**1**	**2**	**3**	**4**
1	45	P	5	56
2	27	2	82	74
3	19	55	3	P
4	3	10	4	84

The symbol P means the transfer is prohibited. Minimize the total costs.

9.11

$$\begin{aligned} \text{Minimize:} \quad & f(\mathbf{x}) = 10x_1 + 11x_2 \\ \text{Subject to:} \quad & 9x_1 + 11x_2 \geq 29 \\ & \mathbf{x} \geq 0 \text{ and integer} \end{aligned}$$

9.12

$$\begin{aligned} \text{Maximize:} \quad & f = 5x_1 + 8x_2 + 6x_3 \\ \text{Subject to:} \quad & 9x_1 + 6x_2 + 10x_3 \leq 14 \\ & 20x_1 + 63x_2 + 10x_3 \leq 110 \\ & x_i \geq 0, \quad \text{integer} \end{aligned}$$

9.13

$$\begin{aligned} \text{Maximize:} \quad & f = x_1 + x_2 + x_3 \\ \text{Subject to:} \quad & x_1 + 2x_2 + 2x_3 + 2x_4 + 3x_5 \leq 18 \\ & 2x_1 + x_2 + 2x_3 + 3x_4 + 2x_5 \leq 15 \\ & x_1 - 6x_4 \leq 0 \\ & x_2 - 8x_5 \leq 0 \\ & \text{all } x_j \geq 0, \quad \text{integer} \end{aligned}$$

9.14 A plant location problem has arisen. Two possible sites exist for building a new plant, A and B, and two customer locations are to be supplied, C and D. Demands and production/supply costs are listed as follows.

Use the following notation to formulate the optimization problem, and solve it for the values of I_1 and I_2 as well as the values of S_{ij}. Each plant has a maximum capacity of 500 units per day.

I_i = decision variable (0–1) associated with the decision to build, or not to build, a plant in a given location, and thus incurs the associated fixed daily cost.
C_{ij} = unit cost of supplying customer j from plant i.
C_i = fixed daily cost of plant i
S_{ij} = quantity supplied from the ith plant to the jth customer
R_j = requirement of jth customer
Q_i = capacity of proposed plant

Production and transport costs per unit:

A to *C*	\$1.00
A to *D*	\$3.00
B to *C*	\$4.50
B to *D*	\$1.00

Fixed plant charges per day:

plant *A*	\$700
plant *B*	\$610

Minimum demand (units per day):

Customer *C*	200
Customer *D*	250

9.15 The ABC company runs two refineries supplying three markets, using a pipeline owned by the XYZ company. The basic charge for pipeline use is \$80 per 1000 barrels. If more than 500 barrels are shipped from the refineries to one market, then the charge drops to \$60 per 1000 barrels for the next 1500 barrels. If more than 2000 barrels are shipped from the refineries to market, then the subsequent charge is \$40 per 1000 barrels for any over the 2000.

The objective is to meet demands at M_1, M_2, and M_3, using supplies from R_1 and R_2.
X_{ijk} = number of barrels from source i to destination j at price k.
C_{ijk} = shipping cost of X_{ijk}
I_{jk} = 0–1 variable to indicate whether or not any product is delivered to destination j at price level k.

We can state the general problem briefly as follows:

$$\text{Minimize: } \sum_{ijk} C_{ijk} x_{ijk} \tag{1}$$

$$\text{Subject to: } \sum_{ik} x_{ijk} \geq M_j \text{ for all } j \text{ (must meet demands)} \tag{2}$$

$$\text{and: } \sum_{jk} x_{ijk} \leq R_i \text{ for all } i \text{ (= sources) (cannot exceed supply)} \tag{3}$$

$$\sum_i x_{ij2} - b_{j2} I_{j2} \leq 0 \quad \text{for all } j \tag{4}$$

$$\sum_i x_{ij1} - b_{j1} I_{j2} \geq 0 \quad \text{for all } j \text{ (if any taken at second price must first use all at top price)} \tag{5}$$

$$\sum_i x_{ij3} - b_{j3} I_{j3} \leq 0 \quad \text{for all } j \tag{6}$$

$$\sum_i x_{ij2} - b_{j2} I_{j3} \geq 0 \quad \text{for all } j \text{ (if any taken at third price must first use all at second price)} \tag{7}$$

b_{jk} = upper bound on product delivered to terminal j at kth price level.

$$\begin{aligned} I_{j2} &\leq 1 \quad \text{for all } j \text{ (upper bounds on integer variables)} \\ I_{j3} &\leq 1 \quad \text{for all } j \end{aligned} \tag{8}$$

The detailed matrix for this problem is set out in Table P9.15. Solve for the I_{jk} values and the x_{ijk} values.

TABLE P9.15 Variables

	*x*111	*x*112	*x*113	*x*121	*x*122	*x*123	*x*131	*x*132	*x*133	*x*211	*x*212	*x*213	*x*221	*x*222	*x*223	*x*231	*x*232	*x*233	112	113	122	123	132	133	
Upper bound	0.5	1.5		0.5	1.5		0.5	1.5		0.5	1.5		0.5	1.5		0.5	1.5		1	1	1	1	1	1	
Lower bound																			0	0	0	0	0	0	
Objective	80	60	40	80	60	40	80	60	40	80	60	40	80	60	40	80	60	40							
DEM.M1	1	1	1							1	1	1													≥ 2.0
DEM.M2				1	1	1							1	1	1										≥ 5.0
DEM.M3							1	1	1							1	1	1							≥ 4.0
CAP.R1	1	1	1	1	1	1	1	1	1																≤ 5.0
CAP.R2										1	1	1	1	1	1	1	1	1							≤ 7.5
M1MAXP1	1									1									−.5						≥ 0
M1MINP2		−1									−1								1.5						≥ 0
M1MAXP2		1									1									−1.5					≥ 0
M1MAXP3			−1									−1								0					≥ 0
M2MAXP1				1									1								−.5				≥ 0
M2MINP2					−1									−1							1.5				≥ 0
M2MAXP2					1									1								−1.5			≥ 0
M2MAXP3						−1									−1							3.0			≥ 0
M3MAXP1							1									1							−.5		≥ 0
M3MINP2								−1									−1						1.5		≥ 0
M3MAXP2								1									1							−1.5	≥ 0
M3MAXP3									−1									−1						2.0	≥ 0