SUMMARY OF OPTIMIZATION THEORY AND METHODS

A

This appendix will attempt to present in a very concise way basic concepts of optimization, optimality conditions, and an outline of the major methods that are used in Chapter 9 and in Part IV. A bibliography is given at the end of the appendix for readers who may wish to do further reading on this subject.

A.1 BASIC CONCEPTS

We will consider the following constrained optimization problem (Bazaraa and Shetty, 1979; Minoux, 1986):

$$\begin{aligned} &\min f(x) \\ &s.t. \;\; h(x) = 0 \\ &g(x) \le 0 \\ &x \in R^n \end{aligned} \tag{P}$$

where $f(x)$ is the objective function, $h(x) = 0$ is the set of m equations in n variables x, and $g(x) \le 0$ is the set of r inequality constraints. In general, the number of variables n will be greater than the number of equations m, and the difference $(n - m)$ is commonly denoted as the number of degrees of freedom of the optimization problem.

Any optimization problem can be represented in the above form. For example, if we maximize a function, this is equivalent to minimizing the negative of that function. Also, if we have inequalities that are greater or equal to zero, we can reformulate them as in-

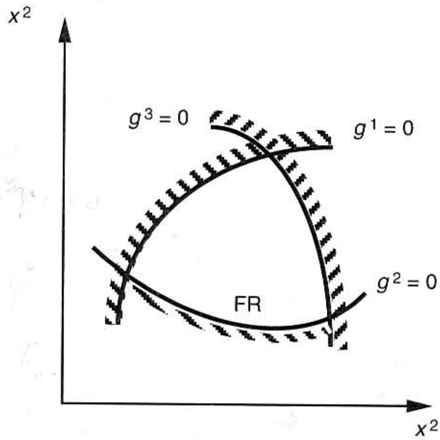

FIGURE A.1 Feasible region for three inequalities.

equalities that are less or equal than zero multiplying the two terms of the inequality by minus one, and reversing the sign of the inequality.

DEFINITION 1

The feasible region *FR* of problem (*P*) is given by

$$FR = \{ x \mid h(x) = 0, g(x) \leq 0, x \in R^n \}$$

Figure A.1 presents an example of a feasible region in two dimensions that involves three inequalities. Note that the boundary of the region is given by those points for which $g_i(x) = 0$, $i = 1,2,3$. Also, the infeasible side of a constraint is represented by dashed lines. In Figure A.2, if we add the equation $h(x) = 0$, the feasible region reduces to the straight line in boldface.

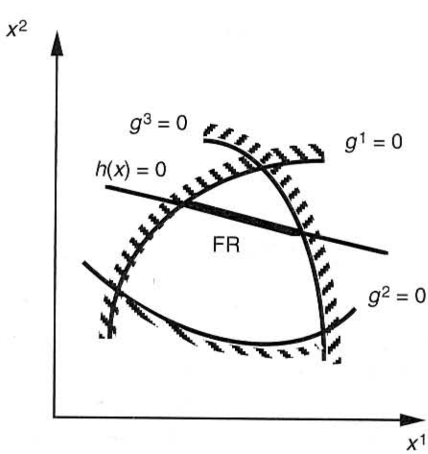

FIGURE A.2 Feasible region for three inequalities and one equation.

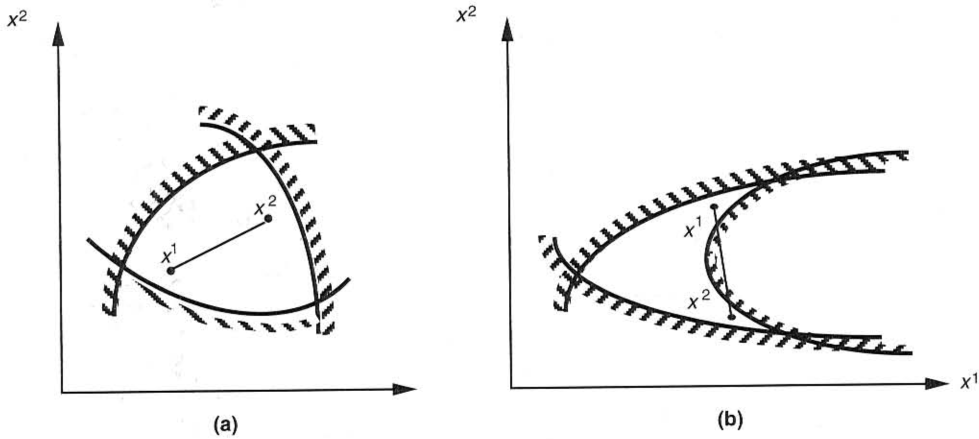

FIGURE A.3 (a) Convex feasible region; (b) nonconvex feasible region.

DEFINITION 2

FR is convex *iff* for any $x^1, x^2 \in FR$,

$$x = \alpha x^1 + (1 - \alpha)\, x^2 \in FR, \ \forall\, \alpha \in [0,1].$$

Figure A.3a presents an example of a convex feasible region; the region in Figure A.3b is nonconvex, since some of the points of the line that results from joining x^1 and x^2 lie outside the region *FR*.

The following is a useful sufficiency condition for the convexity of a feasible region.

PROPERTY 1

If $h(x) = 0$ consists of linear functions, and $g(x)$ of convex functions, then *FR* is a convex feasible region.

DEFINITION 3

$f(x)$ is a convex function *iff* for any $x^1, x^2 \in R$,

$$f(\alpha\, x^1 + [1 - \alpha]\, x^2) \le \alpha\, f(x^1) + [1 - \alpha]\, f(x^2) \quad \forall\, \alpha \in [0,1].$$

Figure A.4a presents an example of a convex function whose value is underestimated in the interval $[x^1, x^2]$ by the linear combination of the function values at the extremes of the interval. Figure A.4b presents an example of a function that is not convex. It should also be noted that if the above expression holds as a strict inequality for the points in the interval (x_1, x_2), then $f(x)$ is said to be strictly convex.

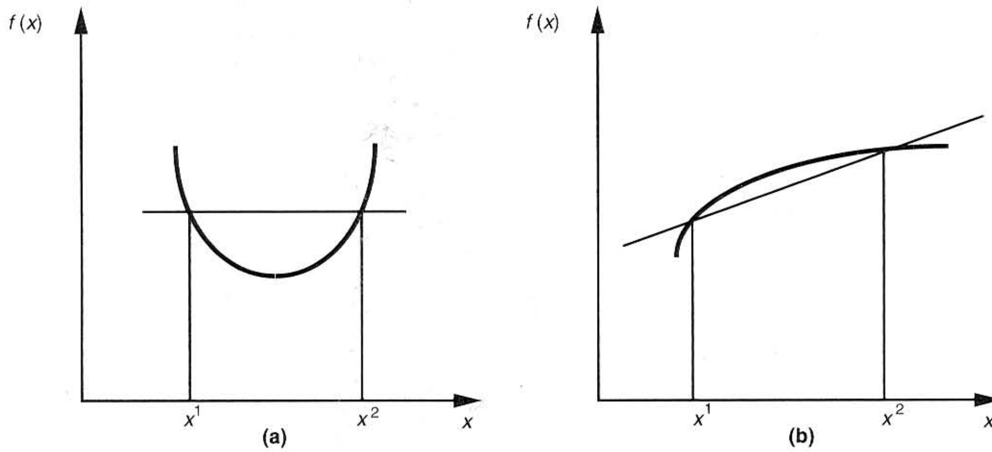

FIGURE A.4 (a) Convex function; (b) nonconvex function.

DEFINITION 4

$f(x)$ has a local minimum at $\hat{x} \in FR$, *iff* $\exists \delta > 0, f(x) \geq f(\hat{x})$ for $| x - \hat{x} | < \delta, x \in FR$.

If strict inequality holds the local minimum is a strong local minimum (see Figure A.5a); otherwise it is a weak local minimum (see Figure A.5b).

DEFINITION 5

$f(x)$ has a global minimum at $\hat{x} \in FR$, *iff* $f(x) \geq f(\hat{x}) \quad \forall x \in FR$.

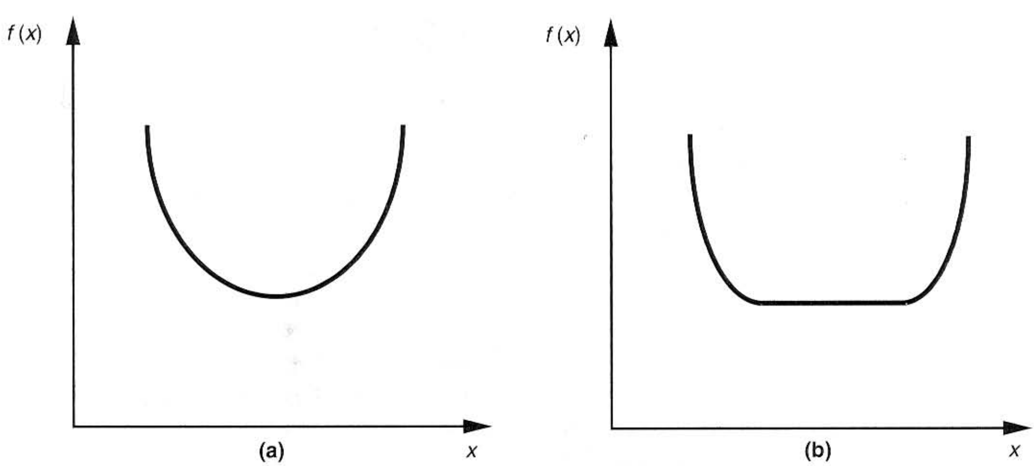

FIGURE A.5 (a) Function with strong local minimum; (b) function with weak local minimum.

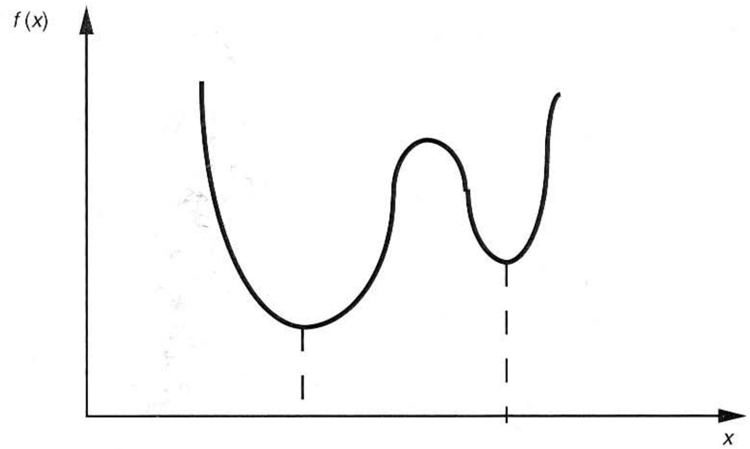

FIGURE A.6 Function with two local minima.

Clearly, every global minimum is a local minimum, but the converse is not true. Figure A.6 presents an example of a function with two strong local minima, one of them being the global minimum.

A.2 OPTIMALITY CONDITIONS

A.2.1 Unconstrained Minimization

Consider first the unconstrained optimization problem,

$$\min f(x)$$
$$x \in R^n$$

where $f(x)$ is assumed to be a continuous differentiable function.

First order conditions, which are necessary for a local minimum at $\hat{x}$, are given by a stationary point; that is, an $\hat{x}$ satisfying $\nabla f(\hat{x}) = 0$. This implies the solution of the following system of n equations in n unknowns,

$$\frac{\partial f}{\partial x_1} = 0$$
$$\frac{\partial f}{\partial x_2} = 0$$
$$\vdots$$
$$\frac{\partial f}{\partial x_n} = 0$$

Second order conditions for a strong local minimum, which are sufficient conditions, require the Hessian matrix H of second partial derivatives to be positive definite. For two dimensions the matrix H is given by,

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}$$

Note that this matrix is symmetric.

The matrix H is said to be positive definite *iff* $\Delta x^T H \Delta x > 0$, $\forall \ \Delta x \neq 0$. The two following properties are useful for establishing in practice the positive definiteness of the Hessian matrix:

1. H is positive definite *iff* the eigenvalues $\rho_i > 0$, $i = 1,2,...n$.
2. If H is positive definite, then $f(x)$ is strictly convex .

That is, from property (1) we can establish the positive definiteness if the eigenvalues calculated from matrix H are all strictly positive. Property (2) simply states that functions whose Hessian matrix is positive definite are strictly convex functions. Therefore, analyzing the Hessian matrix of a function is one way to determine if a given function is convex.

The following is a useful sufficient condition for the uniqueness of a local minimum in an unconstrained optimization problem.

THEOREM 1

If $f(x)$ is strictly convex and differentiable, then if there exists a stationary point at $\hat{x}$, it will correspond to a unique local minimum.

A.2.2 Minimization with Equalities

Consider next the constrained optimization problem with only equalities:

$$\min f(x)$$

$$s.t.\ h(x) = 0$$

$$x \in R^n$$

In this case, the necesssary conditions for a constrained local minimum are given by the stationary point of the Lagrangian function

$$L = f(x) + \sum_{j=1}^{m} \lambda_j h_j(x)$$

where λ_j are the Lagrange multipliers. The stationary conditions are given by,

a. $$\frac{\partial L}{\partial x} = \nabla f(x) + \sum_{j=1}^{m} \lambda_j \nabla h_j(x) = 0$$

b. $$\frac{\partial L}{\partial \lambda_j} = h_j(x) = 0 \qquad j = 1, 2 ... m$$

Note that (a) and (b) define a system of $n + m$ equations in $n + m$ unknowns (x, λ). Also, note that equation (a) implies that the gradients of the objective function and equalities must be linearly dependent, while equation (b) implies feasibility of the equalities. It must also be pointed out that for the above equations to be valid a "constraint qualification" (e.g., see Bazaraa and Shetty, 1979) must hold. In convex problems this qualification is always satisfied.

Second order sufficient conditions for a strong local minimum are satisfied when the Hessian of the Lagrangian is positive definite. That is, given an allowable direction p that lies in the null space, $\nabla h^T p = 0$, we have $p^T \nabla^2 L(x^*, \lambda^*) p > 0$, where $\nabla^2 L(x^*, \lambda^*) = \nabla^2 f(x^*) + \lambda^*_i \nabla^2 h_i(x^*)$.

A.2.3 Minimization with Equalities and Inequalities

Consider the constrained optimization problem with equalities and inequalities,

$$\begin{aligned} &\min f(x) \\ s.t.\ \ &h(x) = 0 \\ &g(x) \leq 0 \\ &x \in R^n \end{aligned} \qquad \text{(P)}$$

In this case the necessary conditions for a local minimum at $\hat{x}$ are given by the Karush-Kuhn-Tucker conditions:

a. Linear dependence of gradients

$$\nabla f(x) + \sum_{j=1}^{m} \lambda_j \nabla h_j(x) + \sum_{j=1}^{r} \mu_j \nabla g_j(x) = 0$$

b. Constraint feasibility

$$h_j(x) = 0 \quad j = 1,2...m \qquad g_j(x) \leq 0 \quad j = 1,2...r$$

c. Complementarity conditions

$$\mu_j g_j(x) = 0, \qquad \mu_j \geq 0 \quad j = 1,2...r$$

where μ_j are the Kuhn-Tucker multipliers corresponding to the inequalities, and which are restricted to be non-negative. Note that the complementarity conditions in (c) imply a zero

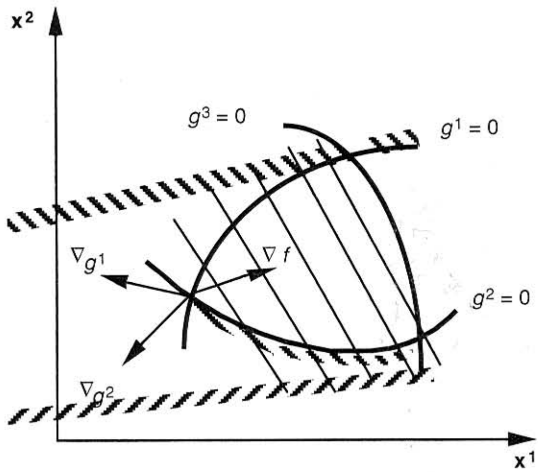

FIGURE A.7 Geometrical representation of a point satisfying the Karush-Kuhn-Tucker conditions.

value for the multipliers of the inactive inequalities (i.e., $g_j(x) < 0$), and in general a nonzero value for the active inequalities (i.e., $g_j(x) = 0$). Figure A.7 presents a geometrical representation of a point satisfying the Karush-Kuhn-Tucker conditions. Note that ∇f is given by a linear combination of the gradients of the active constraints ∇g_1, ∇g_2.

It can also be shown that the multipliers μ_j are given by

$$\mu_j = -\left(\frac{\partial f}{\partial g_j}\right)_{\delta g_i = 0, i \neq j}$$

In other words, they represent the decrease of the objective for an increase in the constraint function; or alternatively, the increase of the objective for a decrease in the constraint function. From the latter, it follows that active inequalities must exhibit a nonnegative value of the multipliers.

The following is a useful sufficient condition on the uniqueness of a local optimum in constrained optimization problems.

THEOREM 2

If $f(x)$ is convex and the feasible region FR is convex, then if there exists a local minimum at $\hat{x}$,

- **i.** It is a global minimum.
- **ii.** The Karush-Kuhn-Tucker conditions are necessary and sufficient.

The difficulty with the equations in (a),(b),(c) for the optimality conditions of problem (P) is that they cannot be solved directly as is the case when only equalities are present. In general the solution to these equations is accomplished by an iterative active set strategy, which in a simplified form consists of the following steps:

Step 1: Assume no active inequalities. Set the index set of active inequalities $J_A = \varnothing$, and the multipliers $\mu_j = 0, j = 1,2,...r$.

Step 2: Solve the equations in (a) and (b) for x, the multipliers λ_j of the equalities, and the multipliers μ_j of the active inequalities (in 1st iteration there are none):

$$\nabla f(x) + \sum_{j=1}^{m} \lambda_j \nabla h_j(x) + \sum_{j \in J_A} \mu_j \nabla g_j(x) = 0$$

$$h_j(x) = 0 \quad j = 1,2...m \qquad g_j(x) = 0 \quad j \in J_A$$

Step 3. If $g(x) \le 0$ and $\mu_j \ge 0, j = 1,2,...r$, STOP, solution found. Otherwise go to step 4.

Step 4: a. If one or more multipliers μ_j are negative, remove from J_A that active inequality with the largest negative multiplier.

b. Add to J_A the violated inequalities $g_j(x) > 0$.
Return to step 2.

The above is only a very general procedure and is suitable for hand calculations of small problems.

A.3 OPTIMIZATION METHODS

In this section we will present a brief overview of the different types of optimization methods covered in Parts II and IV. The emphasis will be on practical aspects, and only in the case of mixed-integer nonlinear programming we will present some more detail on the actual methods.

A.3.1 Linear Programming

When only linear functions are involved in problem (P), and the continuous variables x are restricted to non-negative values, this gives rise to the LP problem:

$$\begin{aligned} &\min Z = c^T x \\ s.t. \quad &A\,x \leqq a \\ &x \ge 0 \end{aligned} \tag{LP}$$

where the sign $\leqq$ denotes equalities and/or inequalities. Since linear functions are convex, from Property 1 and Theorem 2, the LP has a unique minimum. This may, however, be a weak minimum, for which alternate variable values may give rise to the same minimum objective function value.

The standard solution method is the simplex algorithm [Hillier and Lieberman, 1986] which exploits the fact that in an LP the optimum lies at a vertex of the feasible region (see Figure A.8). At this optimum, the Karush-Kuhn-Tucker conditions are satisfied.

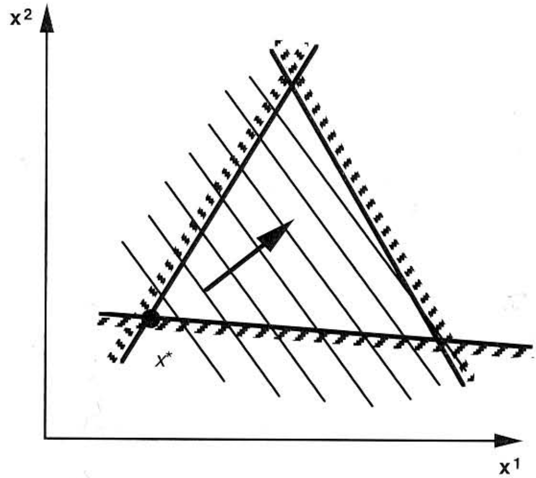

FIGURE A.8. Optimum lies at vertex x^* for LP problem.

Many refinements have been developed over the last three decades for the simplex method, and most of the current commercial computer codes (e.g., OSL, CPLEX, LINDO) are based on this method. Very large scale problems (thousands of variables and constraints) that are sparse (i.e., few variables in each constraint) can be solved quite efficiently. As a general guideline, the computational effort in the simplex algorithm is dependent mostly on the number of constraints (rows in LP terminology), not so much on the number of variables (columns). In problems with many rows and relatively few variables, it is advisable to solve the LP through its dual problem .

For variables x that can be positive and negative in an LP, these are replaced by $x = x^P - x^N$, where x^P and x^N are non-negative. If x^N is zero we get a positive value, and if x^P is zero we get a negative value. This manipulation should only be used when the variable x appears with a positive coefficient in the minimization of an objective function.

Recently, interior point methods for LP (Marsten et al., 1990) have been developed that are polynomially bounded in time. Although these methods are theoretically superior to the simplex algorithm, it is only for extremely large scale problems that substantial computational savings have been observed (e.g., problems with 100,000 constraints and variables).

As a final point, it is important to note that special classes of LP problems can be solved more efficiently than with standard LP codes. The best known case are network flow problems (see Minoux, 1986) where the matrix of coefficients involves only 0, 1, −1, elements. In this case the simplex method can be implemented with symbolic computations leading to order of magnitude reductions in computational time.

A.3.2 Mixed-Integer Linear Programming

This is an extension of the LP problem where a subset of the variables are restricted to integer values (most commonly to 0-1). The general form of the MILP problem is given by,

$$\min Z = a^T y + c^T x$$
$$s.t.\ \ By + Ax \leqq b \qquad \text{(MILP)}$$
$$y \in \{0,1\}^t \qquad x \geq 0$$

where y corresponds to a vector of t binary variables.

The MILP problem is very useful for modeling a number of discrete decisions with the binary variables y (see Chapter 15). Typical examples are the following:

a. Multiple choice constraints
Select only one item:

$$\sum_{j=1}^{t} y_i = 1$$

Select at most one item:

$$\sum_{j=1}^{t} y_j \leq 1$$

Select at least one item:

$$\sum_{j=1}^{t} y_j \geq 1$$

b. Implication constraints.
If item k is selected, item j must be selected, but not vice versa: $y_k - y_j \leq 0$
If a binary variable y is zero, an associated continuous variable x must also be zero:

$$x - Uy \leq 0,\ \ x \geq 0$$

where U is an upper limit to x.

c. Either-or constraints (disjunctive constraints)
Either constraint $g_l(x) \leq 0$ or constraint $g_2(x) \leq 0$ must hold:

$$g_1(x) - U\,y \leq 0, \qquad g_2(x) - \mathrm{U}\,(1 - y) \leq 0$$

where U is a large value.

A simple-minded approach to obtain the global optimum of the above MILP would be to solve the LPs that result from considering all the 0-1 combinations of the binary variables. However, the number of combinations is 2^t, which is too large for even modest number of variables (e.g., for 20 binaries there are 10^6 combinations).

A second approach is to relax the 0-1 constraints as continous variables that must lie between 0 and 1; that is, $0 \leq y_t \leq 1$. The problem is then solved as an LP. The difficulty here is that except for special cases (e.g., assignment problems), one or more binary variables will exhibit noninteger values at the optimum LP solution. The relaxed LP, however, is useful in providing a lower bound to the optimal mixed-integer solution.

In general, one cannot simply round the noninteger values of the binary variables in the relaxed LP solution to the nearest integer point. Firstly, because the rounding may be infeasible (see Figure A.9a), or secondly because it may be nonoptimal (see Figure A.9b). The standard method for solving MILP problems is the branch and bound method (Nemhauser and Wolsey, 1988), which was briefly outlined in Chapter 15 in the context of the synthesis of a separation sequence. For the MILP we start by solving first the relaxed LP problem. If integer values are obtained for the binary variables, we stop, as we have solved the problem. If, on the other hand, no integer values are obtained, the basic idea is then to examine through the use of bounds a subset of nodes in a binary tree to locate the global mixed-integer solution. In the tree the binary variables are successively restricted one by one to 0-1 values at each node where the corresponding LP is solved. This can be done quite efficiently by updating the successive LPs through few dual simplex iterations.

Nodes with noninteger solutions provide a lower bound, and nodes with feasible mixed-integer solutions provide an upper bound. The former nodes are fathomed whenever the lower bound is greater or equal than the current best upper bound. For the tree enumeration one has to consider branching rules to decide which binary variable is fixed next in the tree. These rules range from simply picking the first non-zero value to the use of penalties to estimate which binary produces the smallest degradation in the LP. Also, in a similar way as in the implicit enumeration described in Chapter 15, the tree can be enumerated through a depth-first method, a breadth-first method, or combination of the two.

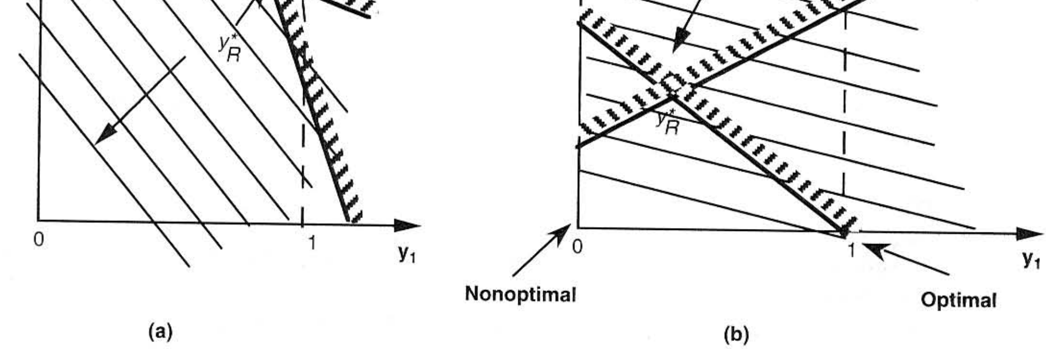

FIGURE A.9 (a) Infeasible rounding of relaxed integer solution; (b) nonoptimal rounding of relaxed integer solution.

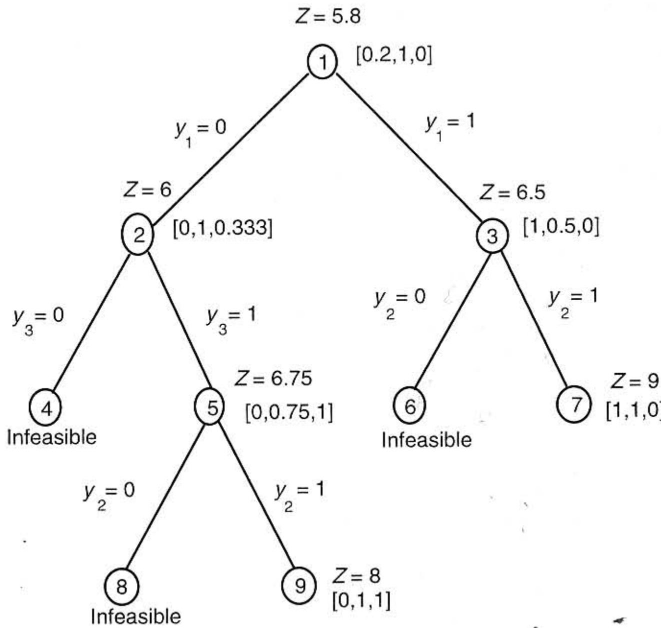

FIGURE A.10 Branch and bound tree for example problem (MIPEX).

The more advanced MILP packages allow the specialized user to specify the search option to be used. Figure A.10 presents an example of a tree search with branch and bound in the MILP problem:

$$\begin{aligned} \min \quad & Z = x + y_1 + 3y_2 + 2y_3 \\ st. \quad & -x + 3y_1 + 2y_2 + y_3 \leq 0 \\ & -5y_1 - 8y_2 - 3y_3 \leq -9 \\ & x \geq 0\ , y_1, y_2, y_3 = \{0,1\} \end{aligned} \qquad \text{(MIPEX)}$$

The branch and bound tree using a breadth-first enumeration is shown in Figure A.10. The numbers in the circles represents the order in which 9 nodes out of the 15 nodes in the tree are examined to find the optimum. Note that the relaxed solution (node 1) has a lower bound of $Z = 5.8$, and that the optimum is found in node 9 where $Z = 8$, $y_1 = 0$, $y_2 = y_3 = 1$, and $x = 3$.

Although the general performance of the branch and bound method can greatly vary from one problem to another, as a general guideline the computational expense tends to be proportional first to the number of 0-1 variables, secondly to the number of constraints, and thirdly to the number of continuous variables. Another criterion, which is often more relevant, is the gap between the objective function value of the relaxed LP and the optimal MILP solution. The smaller this gap the easier it is usually to solve the MILP problem since the LP relaxation is "tighter." The importance of developing a proper MILP formulation that adheres as much as possible to the above guidelines cannot be underemphasized.

As for computer packages, most LP codes include extensions for solving MILP problems (e.g., OSL, CPLEX, LINDO, ZOOM).

A.3.3 Nonlinear Programming

In this case, the problem corresponds to:

$$\begin{aligned} & \min f(x) \\ s.t.\quad & h(x) = 0 \\ & g(x) \le 0 \\ & x \in R^n \end{aligned} \qquad \text{(NLP)}$$

where in general $f(x)$, $h(x)$, $g(x)$, are nonlinear functions.

The more efficient NLP methods solve this problem by determining directly a point that satifies the Karush-Kuhn-Tucker conditions. As pointed out in Theorem 2, global minumum solutions can be guaranteed for the case when the objective and constraints are nonlinear convex functions, and the equalities are linear. Since the Karush-Kuhn-Tucker conditions involve gradients of the objective and constraints, these must be supplied by the user either in analytical form or through the use of numerical perturbations. However, the latter option is expensive for problems with large number of variables.

Currently the two major methods for NLP are the successive quadratic programming (SQP) algorithm (Han, 1976; Powell, 1978) and the reduced gradient method (Murtagh and Saunders, 1978, 1982). In the case of the (SQP) algorithm (see Chapter 9 for more details) the basic idea is to solve at each iteration a quadratic programming subproblem of the form:

$$\begin{aligned} \min\quad & \nabla f(x^k)^T d + 1/2\, d^T B^k d \\ s.t.\quad & h(x^k) + \nabla h(x^k)^T d = 0 \\ & g(x^k) + \nabla g(x^k)^T d \le 0 \end{aligned} \qquad \text{(QP)}$$

where x^k is the current point, B^k is the estimation of the Hessian matrix of the Lagrangian, and d is the predicted search direction. The matrix B^k is usually estimated with the BFGS update formula, and the QP is solved with standard methods for quadratic programming (e.g., QPSOL routine). Since the point x^k will in general be infeasible, the next point x^{k+1} is set to $x^{k+1} = x^k + \alpha\, d$, where the step size α is determined so as to reduce a penalty function that tries to balance the improvement in the objective and the violation of the constraints.

An important point about the SQP algorithm is the fact that the QP with the exact Hessian matrix of the Lagrangian in B can be shown to be equivalent to applying Newton's method to the Karush-Kuhn-Tucker conditions. Thus, fast convergence can be achieved with this algorithm.

In the reduced gradient method, on the other hand, the basic idea is to solve a sequence of subproblems with linearized constraints, where the subproblems are solved by variable elimination. In the particular implementation of MINOS by Murtagh and Saunders, the NLP is reformulated through the introduction of slack variables to convert the inequalities into equalities; that is, the NLP reduces to

$$\min f(x) \qquad \text{(NLP1)}$$
$$s.t.\ r(x) = 0$$

Linear approximations of the constraints are then considered with an augmented Lagrangian for the objective function:

$$\min \phi(x) = f(x) + (\lambda^k)^T [r(x) - r(x^k)] \qquad \text{(NLP2)}$$
$$s.t.\ J(x^k)\, x = b$$

where λ^k is the vector of Lagrange multipliers, and $J(x^k)$ is the jacobian of $r(x)$ evaluated at the point x^k. Subproblem NLP2, which is a linearly constrained optimization problem, can be represented by

$$\min \phi(x)$$
$$s.t.\ A\,x = b$$

where A is a mxn matrix with m < n. The above problem can be solved with the reduced gradient method as follows. Firstly, the vector x is partitioned into the vector v of m dependent variables, and the vector u of $(n - m)$ independent variables. Likewise, the matrix A is partitioned into a (mxm) square matrix B, and a $mx(n - m)$ matrix C. The reduced gradient can then be computed from the equation

$$g_R = Z^T \nabla \phi(x^k)$$

where x^k is a feasible point satisfying the linear constraints, and Z is a transformation matrix given by

$$Z = [-B^{-1}C \mid I]$$

With the reduced gradient the Newton step, Δu in the reduced space can be computed from

$$H_R\, \Delta u = -g_R$$

where H_R is the reduced Hessian matrix, which is estimated through a Quasi-Newton update formula (e.g., BFGS formula). The change in the dependent variables, Δv, is then obtained by solving the linear equations

$$B\, \Delta v = -C\, \Delta u$$

In summary, in the reduced gradient method the subproblem (NLP2) is solved as an inner optimization problem, while in the outer optimization the new point is set as $x^{k+1} = x^k + \alpha\, \Delta x$ where α is the step size that is used to reduce the augmented Lagrangian in (NLP2), and $\Delta x = [\Delta v | \Delta u]$

The importance of the reduced gradient method is that by efficient implementation for the solution of the above equations (see Murtagh and Saunders, 1982) and realizing that some of the tools for large-scale LP can be used, sparsity can be readily exploited. In this way large nonlinear optimization problems can be solved very effectively. In comparing the SQP algorithm and the reduced gradient method, the following general guidelines apply:

1. SQP requires fewer iterations than the reduced gradient method. However, there may be difficulties in applying it to large-scale problems since in general the matrix B^k, which is of dimension $n \times n$, will become dense due to the Quasi-Newton updates. The SQP method is best suited for "black-box" models (e.g., process simulators) that involve relatively few variables (e.g., up to 50) and where the gradients must be obtained by numerical perturbation. It should be noted, however, that the SQP algorithm can be effectively applied to large-scale problems that involve few decision variables by using decomposition techniques.
2. The reduced gradient method, as per the implementation in MINOS is best suited for problems involving a significant number of linear constraints, and where analytical derivatives can be supplied for the nonlinear functions. With this structure, MINOS can solve problems with several hundred variables and constraints. Compared to SQP, MINOS will require a larger number of function evaluations, but the computational time per iteration will be smaller. Furthermore, in the limiting case when all the functions are linear the method reduces to the simplex algorithm for linear programming.

A.3.4 Mixed-Integer Nonlinear Programming

MINLP problems are usually the hardest to solve unless a special structure can be exploited. The following particular formulation, which is linear in the 0-1 variables and linear/nonlinear in the continuous variables, will be considered:

$$\begin{aligned} \min Z = \ & c^T y + f(x) \\ s.t. \quad & h(x) = 0 \\ & g(x) \leq 0 \\ & A x = a \\ & B y + C x \leq d \\ & E y \leq e \\ & x \in X = \{x \mid x \in R^n, x^L \leq x \leq x^U\} \\ & y \in \{0,1\}^t \end{aligned} \qquad \text{(MINLP)}$$

As explained in Chapter 15, this special MINLP structure arises in process synthesis problems.

This mixed-integer nonlinear program can in principle also be solved with the branch and bound method presented in section A.3.2. The major difference here is that the examination of each node requires the solution of a nonlinear program rather than the solution of an LP. Provided the solution of each NLP subproblem is unique, similar properties as in the case of the MILP would hold with which the rigorous global solution of the MINLP can be guaranteed.

An important drawback of the branch and bound method for MINLP is that the solution of the NLP subproblems can be expensive since they cannot be readily updated as in the case of the MILP. Therefore, in order to reduce the computational expense involved in solving many NLP subproblems, we can resort to two other methods: Generalized Benders decomposition (Geoffrion, 1972) and Outer-Approximation (Duran and Grossmann, 1986). Below we first briefly describe the latter method with the equality relaxation variant by Kocis and Grossmann (1987).

The basic idea in the OA/ER algorithm is to solve an alternating sequence of NLP and MILP master problems. The NLP subproblems arise for a fixed choice of the binary variables, and involve the optimization of the continuous variables x with which an upper bound to the original MINLP is obtained (assuming minimization problem). The MILP master problem, on the other hand, provides a global linear approximation to the MINLP in which the objective function is underestimated and the nonlinear feasible region is overestimated. Furthermore, the linear approximations to the nonlinear equations are relaxed as inequalities. This MILP master problem accumulates the different linear approximations of previous iterations so as to produce an increasingly better approximation of the original MINLP problem. At each iteration the master problem predicts new values of the binary variables y and a lower bound to the objective function Z. The search is terminated when no lower bound can be found below the current best upper bound which then leads to an infeasible MILP.

The specific steps of this algorithm, assuming feasible solutions for the NLP subproblems, are as follows:

Step 1: Select an initial value of the binary variables y^1. Set the iteration counter $K = 1$. Initialize the lower bound $Z_L^0 = -\infty$, and the upper bound $Z_U = +\infty$.

Step 2: Solve the NLP subproblem for the fixed value y^k, to obtain the solution x^k and the multipliers λ^k for the equations $h(x) = 0$.

$$
\begin{aligned}
Z(y^k) = \min\ & c^T y^k + f(x) \\
s.t.\quad & h(x) = 0 \\
& g(x) \le 0 \\
& A\,x = a \\
& C\,x \le d - B\,y^k \\
& x \in X
\end{aligned}
$$

Step 3: Update the bounds and prepare the information for the master problem:

a. Update the current upper bound; if $Z(y^K) < Z_U$, set $Z_U = Z(y^K)$, $y^* = y^K$, $x^* = x^K$.

b. Derive the integer cut, IC^K, to make infeasible the choice of the binary y^K from subsequent iterations:

$$IC^K = \{\sum_{i e B^K} y_i - \sum_{i e N^K} y \leq |B^K| - 1\}$$

where $B^k = \{i \mid y_i^K = 1\}$, $N^K = \{i \mid y_i^K = 0\}$

c. Define the diagonal direction matrix T^K for relaxing the equations into inequalities based on the sign of the multipliers λ^k. The diagonal elements are given by:

$$t_{jj}^K = \begin{Bmatrix} -1 \text{ if } \lambda_j^K < 0 \\ +1 \text{ if } \lambda_j^K > 0 \\ 0 \text{ if } \lambda_j^K = 0 \end{Bmatrix} \qquad j = 1, 2 \ldots m$$

d. Obtain the following linear outer-approximations for the nonlinear terms $f(x)$, $h(x)$, $g(x)$ by performing first order linearizations at the point x^K:

$$(w^K)^T x - w_c^K = f(x^K) + \nabla f(x^K)^T (x - x^K)$$

$$R^K x - r^K = h(x^K) + \nabla h(x^K)^T (x - x^K)$$

$$S^K x - s^K = g(x^K) + \nabla g(x^K)^T (x - x^K)$$

Step 4: a. Solve the following MILP master problem:

$$\begin{aligned} Z_L^K = \min\ & c^T y + \mu \\ s.t.\ & (w^k)\, x - \mu \leq w_c^k \\ & T^k R^k x \leq T^k r^k \qquad k = 1, 2, \ldots K \\ & S^k x \leq s^k \\ & y \in IC^k \\ & By + Cx \leq d \\ & A x = a \\ & E y \leq e \\ & Z_L^{K-1} \leq c^T y + \mu \leq Z_U \\ & y \in \{0,1\}^t \quad x \in X \quad \mu \in R^1 \end{aligned} \qquad \text{(MOA)}$$

b. If the MILP master problem has no feasible solution, stop. The optimal solution is x^*, y^*, Z_U.

c. If the MILP master problem has a feasible solution, the new binary value y^{K+1} is obtained. Set $K = K + 1$, return to step 2.

It should be noted that in step 2, there is the possibility that the NLP subproblem may not have a feasible solution for the selected value of the binary variable y^K. When this is the case, the value of x^K and λ^k can be obtained by solving the following NLP in which the infeasibility is minimized:

$$\begin{aligned} &\min\ u \\ s.t.\quad & h(x) = 0 \\ & g(x) \le u \\ & A\,x = a \\ & C\,x - d - B\,y \le u \\ & x \in X \qquad u \in R^1 \end{aligned}$$

Furthermore, the objective function value is set to $Z\,(y^k) = +\infty$

It should be noted that sufficient conditions to obtain the global optimum solution require convexity in the nonlinear terms $f(x)$, $g(x)$, and quasi-convexity in the relaxed non-linear equations $T^k\ h(x)$. When these conditions are not met, there is the possibility that the master problem may cut off the global optimum solution as discussed below.

Also, as an interesting point it should be noted that for the limiting case when $f(x)$, $g(x)$, and $h(x)$ are linear, the MILP master problem provides an exact representation of the MINLP, and therefore the OA/ER algorithm would converge in no more than two iterations. For nonlinear problems, computational experience indicates that the master problems provide an increasingly good approximation with which convergence can be typically achieved in only 3 to 5 iterations.

In the Generalized-Benders decomposition the above steps are virtually identical except that the MILP master problem in step 4(a) (assuming feasible NLP subproblems) is given at any iteration K by:

$$\begin{aligned} & Z_{GB}^{K} = \min\ \alpha \\ & s.t.\ \alpha \ge f(x^k) + c^T y + (\mu k)^T\,[Cx^k + By - d] \quad k = 1,2,\ldots K \qquad \text{(MGB)} \\ & \alpha \in R^1,\ y \in \{0,1\}^m \end{aligned}$$

where α is the largest Lagrangian approximation obtained from the solution of the K NLP subproblems; x^k and μ^k correspond to the optimal solution and multiplier of the kth NLP subproblem; Z_{GB}^{K} corresponds to the predicted lower bound at iteration K.

Note that in both master problems the predicted lower bounds, Z_{GB}^{K}, and Z_{OA}^{K} increase monotonically as iterations K proceed since the linear approximations are refined by accumulating the Lagrangian (in MGB) or linearizations (in MOA) of previous iterations. It should be noted also that in both cases rigorous lower bounds, and therefore convergence to the global optimum, can only be ensured when certain convexity conditions hold (see Geoffrion, 1972; Duran and Grossmann, 1986).

In comparing the two methods, it should be noted that the lower bounds predicted by the outer approximation method are always greater than or equal to the lower bounds predicted by Generalized-Benders decomposition. This follows from the fact that the Lagrangian cut in GBD represents a surrogate constraint from the linearization in the OA algorithm (Quesada and Grossmann, 1992). Hence, the Outer-Approximation method will require the solution of fewer NLP subproblems and MILP master problems. On the other hand, the MILP master in Outer-Approximation is more expensive to solve so that Generalized Benders may require less time if the NLP subproblems are inexpensive to solve. As discussed in Sahinidis and Grossmann (1991), fast convergence with GBD can only be achieved if the NLP relaxation is tight.

As a simple example of an MINLP consider the problem:

$$
\begin{aligned}
\min Z = {} & y_1 + 1.5y_2 + 0.5y_3 + x_1^2 + x_2^2 \\
s.t. \quad & (x_1 - 2)^2 - x_2 \le 0 \\
& x_1 - 2y_1 \ge 0 \\
& x_1 - x_2 - 4(1 - y_2) \le 0 \\
& x_1 - (1 - y_1) \ge 0 \\
& x_2 - y_2 \ge 0 \\
& x_1 + x_2 \ge 3y_3 \\
& y_1 + y_2 + y_3 \ge 1 \\
& 0 \le x_1 \le 4,\ 0 \le x_2 \le 4 \\
& y_1, y_2, y_3 = 0, 1
\end{aligned}
\tag{6}
$$

Note that the nonlinearities involved in problem (6) are convex. Figure A.11 shows the convergence of the OA and the GBD methods to the optimal solution using as a start-

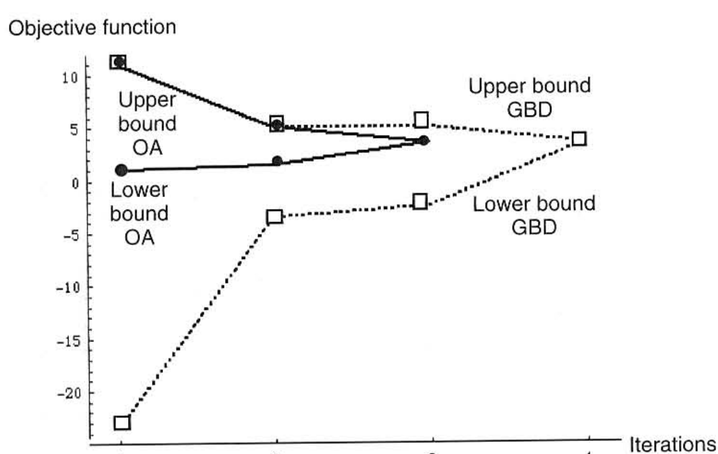

FIGURE A.11 Progress of iterations of OA and GBD for MINLP in (6).

ing point $y_1 = y_2 = y_3 = 1$. The optimal solution is $Z = 3.5$, with $y_1 = 0$, $y_2 = 1$, $y_3 = 0$, $x_1 = 1$, $x_2 = 1$. Note that the OA algorithm requires three major iterations, while GBD requires four, and that the lower bounds of OA are much stronger.

In the application of Generalized-Benders decomposition and Outer-Approximation, two major difficulties that can arise are the computational expense involved in the master problem if the number of 0-1 variables is large, and nonconvergence to the global optimum due to the nonconvexities involved in the nonlinear functions.

As for the question of nonconvexities, one approach is to modify the definition of the MILP master problem so as to avoid cutting off feasible mixed-integer solutions. Viswanathan and Grossmann (1990) proposed an augmented-penalty version of the MILP master problem for outer-approximation, which has the following form:

$$Z_L^K = \min\ c^T y + \mu + \sum_{k=1}^{K} (\rho^k)^T (p^k + q^k + r^k) \qquad \text{(MOA)}$$

$$\begin{aligned} s.t. \quad & \left.\begin{aligned} &(w^k)\, x - \mu \le w_c^k + p^k \\ &T^k R^k x \le T^k r^k + q^k \\ &S^k x \le s^k + r^k \\ &y \in IC^k \end{aligned}\right\} \quad k = 1,2,\ldots K \\ & By + Cx \le d \\ & A\,x = a \\ & E\,y \le e \\ & y \in \{0,1\}^t \quad x \in X \quad \mu \in R^1; \quad p^k, q^k, r^k \ge 0 \end{aligned}$$

in which the slacks p^k, q^k, r^k have been added to the function linearizations, and in the objective function with weights ρ^k that are sufficiently large but finite. Since in this case one cannot guarantee a rigorous lower bound, the search is terminated when there is no further improvement in the solution of the NLP subproblem. This version of the method together with the original version have been implemented in the computer code DICOPT++, which has shown to be successful in a number of applications. It should also be noted that if the MINLP is convex, the above master problem reduces to the original OA algorithm since the slacks will take a value of zero. For an updated review of MINLP methods see Grossmann and Kravanja (1995).

A.4 COMPUTER CODES AND REFERENCES

The following computer software can be used for solving different classes of problems:

1. For LP and MILP:
   - LINDO by Linus Schrage. Interactive program that is easy to use.