JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 21, No. 2, FEBRUARY 1977

An Accelerated Multiplier Method for Nonlinear Programming[1]

J. T. BETTS[2]

Communicated by M. R. Hestenes

Abstract. This paper describes an accelerated multiplier method for solving the general nonlinear programming problem. The algorithm poses a sequence of unconstrained optimization problems. The unconstrained problems are solved using a rank-one recursive algorithm described in an earlier paper. Multiplier estimates are obtained by minimizing the error in the Kuhn–Tucker conditions using a quadratic programming algorithm. The convergence of the sequence of unconstrained problems is accelerated by using a Newton–Raphson extrapolation process. The numerical effectiveness of the algorithm is demonstrated on a relatively large set of test problems.

Key Words. Augmented penalty function, method of multipliers, penalty function methods, nonlinear programming, mathematical programming.

1. Introduction

This paper describes an algorithm for solving the nonlinear programming or mathematical programming problem. The method extends the philosophy of the penalty function method described in Ref. 1 and is closely related to the method proposed by Powell (Ref. 2). An approach similar to Powell's was suggested by Hestenes (Ref. 3) and refined by Miele (Ref. 4). However, none of these methods deal with inequality constraints directly, nor do they exhibit quadratic convergence. The new algorithm retains the favorable numerical properties of the multiplier methods, while extending the approach to inequality constrained problems. The approach poses a sequence of unconstrained problems and uses a rank-one method to solve the problems.

[1] This work was supported by the US Air Force under Contract No. F04701-74-C-0075.

[2] Member of Technical Staff, The Aerospace Corporation, El Segundo, California.

Estimates of the Lagrange multipliers are computed by minimizing the error in the necessary conditions for an optimum, using a quadratic programming algorithm. An estimate of the constrained solution is obtained using a Newton–Raphson extrapolation procedure. Although we do not prove quadratic convergence of the extrapolation steps, numerical experience tends to confirm this assertion. Furthermore, experience indicates that the set of constraints binding at the solution is readily identified without cycling, presumably because the constraint penalty weight is increased from one unconstrained problem to the next.

The problem of interest in this paper is to find the n-vector x that minimizes the scalar function

$$f(x)=f(x_1,\ldots,x_n), \tag{1}$$

called the performance index, subject to the equality constraints

$$c_i(x)=0, \qquad i=1,\ldots,m_1, \tag{2}$$

and the inequality constraints

$$c_i(x)\geq 0, \qquad i=(m_1+1),\ldots,m. \tag{3}$$

The functions $f(x)$ and $c_i(x)$ are assumed to be continuously differentiable to second order in the region

$$x_L \leq x \leq x_U, \tag{4}$$

where x_L and x_U are the specified lower and upper bounds. These bounds determine a region of computability and, unlike the constraints, cannot be violated during the iterative process.

Define the Lagrangian

$$L(x,\lambda)=f(x)+c^T(x)\lambda, \tag{5}$$

where $c(x)$ is the m-vector of all constraints and λ is the m-vector of Lagrange multipliers. At the optimum point (x^*, λ^*),

$$\nabla L(x^*,\lambda^*)=g(x^*)+G(x^*)\lambda^*=0, \tag{6}$$

where ∇L is the gradient vector of the Lagrangian function with respect to x, that is,

$$g(x)=\nabla f(x), \tag{7}$$

and the $n\times m$ Jacobian matrix $G(x)$ is given by

$$G(x)=[\nabla c_1,\ldots,\nabla c_m]=\begin{bmatrix} \partial c_1/\partial x_1 & \cdots & \partial c_m/\partial x_1 \\ & \cdots & \\ \partial c_1/\partial x_n & \cdots & \partial c_m/\partial x_n \end{bmatrix} \tag{8}$$

Furthermore,

$$\lambda_i^* c_i(x^*) = 0, \qquad i = 1, \ldots, m, \tag{9}$$

where

$$\lambda_i^* \leq 0, \qquad i = (m_1 + 1), \ldots, m. \tag{10}$$

In order to distinguish the constraints that are active at a solution, define the set

$$B^* \doteq \{i \mid c_i(x^*) = 0, i = 1, \ldots, m\}, \tag{11}$$

called the *basic* set of constraints. An estimate B of the basic set of constraints shall be referred to as a basis. Clearly, B contains all equality constraints. If the gradients of the constraints in the basis are linearly independent at the solution, then (6), (9), and (10) constitute the Kuhn–Tucker necessary conditions for the existence of an optimum.

Most common nonlinear programming algorithms implicitly assign some priority to either the constraint condition (9) or the Lagrangian condition (6). Projected gradient algorithms are constraint-following, since they attempt to satisfy the constraints at each step and move toward satisfaction of the Lagrangian condition. In contrast, penalty function methods attempt to satisfy the Lagrangian condition for some value of λ and then move toward constraint satisfaction. The algorithm to be described here is of the Lagrangian type and requires both function and gradient information. The algorithm is designed for a class of problems in which the function and gradient evaluations are relatively expensive from a computational standpoint.

The overall iterative process to be developed consists of three major operations which, to some extent, are independent of each other. We shall first discuss an unconstrained minimization procedure; then, a basis determination process; and, finally, an extrapolation method.

2. Unconstrained Optimization Algorithm

In this section, the penalty function algorithm presented in Ref. 1 is generalized to the optimization of the following augmented penalty function:

$$J(x, \lambda, r) = f(x) + c^T(x)\lambda + rP(x) = L(x, \lambda) + rP(x), \tag{12}$$

where r is a scalar referred to as the penalty weight, λ is an m-vector of specified estimates of the Lagrange multipliers, and the constraint penalty

$P(x)$ is defined as

$$P(x)=\sum_{i\in B} c_i^2(x)+\sum_{i\in B'} I[c_i(x)]c_i^2(x). \tag{13}$$

The basis B is an estimate of the basic set of constraints, and the set B' denotes the complement of B. The indicator I is defined as follows:

$$I[c_i(x)]=\begin{cases}1 & \text{if } c_i<0,\\ 0 & \text{if } c_i\geq 0.\end{cases} \tag{14}$$

Expressions for the first and second derivatives are obtained by differentiation of (12). Thus,

$$\nabla J=\nabla L+r\,\nabla P, \tag{15}$$

where

$$\nabla L=\nabla f+G\lambda, \tag{16}$$

$$\nabla P=\sum_{i\in B} 2c_i(x)\,\nabla c_i(x)+\sum_{i\in B'} 2I[c_i(x)]c_i(x)\,\nabla c_i(x). \tag{17}$$

Notice that the first summation is over all constraints in the basis, and the second summation is over all constraints not in the basis. The Hessian matrix is

$$\nabla^2 J=\nabla^2 L+r\,\nabla^2 P. \tag{18}$$

The unconstrained optimization algorithm developed in Ref. 1 can be applied to the minimization of the augmented performance index (12). The implementation is straightforward when it is noted that the objective function used in the earlier algorithm is replaced by the Lagrangian L and the penalty function is modified slightly to account for the basis estimate B. The only additional change made in the earlier algorithm is the determination of the search direction by means of a quadratic programming algorithm when the standard mode of operation would result in violation of the bounds (4). The details of this implementation are found in Ref. 5.

3. Determination of the Basis

The preceding section developed an algorithm for minimizing the augmented penalty function J [Eq. (12)]. The estimates of the multipliers λ and the associated basis B are fixed for any unconstrained optimization problem. In this section, a method for computing estimates of the Lagrange multipliers λ and determining an estimate of the basis is described. This basis determination process assumes that the variables x are fixed.

First, consider how estimates for the Lagrange multipliers can be constructed. At the optimal point x^*, we require that the Kuhn–Tucker conditions (6), (9), and (10) be satisfied. However, at an arbitrary point x, one would expect some error in the Kuhn–Tucker conditions, which we shall define as

$$e(\lambda) = \|A\lambda + b\|^2, \tag{19}$$

where

$$A = \begin{bmatrix} G \\ \hline c_1 & & 0 \\ & \ddots & \\ 0 & & c_m \end{bmatrix}, \tag{20}$$

$$b = \begin{bmatrix} g \\ \hline 0 \end{bmatrix}. \tag{21}$$

Consequently, since the values of λ are unspecified, it seems reasonable to choose them such that $e(\lambda)$ is as small as possible, consistent with the constraints (10). Equivalently, we could minimize the quantity

$$\bar{e}(\lambda) = \lambda^T A^T A\lambda + b^T A\lambda, \tag{22}$$

subject to the constraints

$$\lambda_i \leq 0, \qquad i = (m_1 + 1), \ldots, m. \tag{23}$$

Since $\bar{e}$ is a quadratic function of λ and the constraints (23) are linear, the optimal values of λ can be determined using a quadratic programming algorithm. Fletcher's algorithm (Refs. 6–7) is used for this application. Note that, if the constraints are linearly dependent, then the matrix A does not have rank m.

Let us now address the problem of basis determination. In particular, we are concerned with generating a new basis $\bar{B}$ from the old basis B, the estimates of the multipliers λ, and the values of the constraints c. Clearly, the new basis $\bar{B}$ should contain all the equality constraints. This is stated as the following rule:

Rule 1: if $0 \leq i \leq m_1$, then $i \in \bar{B}$. (24)

One would also expect that any inequality constraints in the final basis would have negative multipliers in accordance with the Kuhn–Tucker condition (23). Furthermore, since an exterior point penalty function is being used,

one would expect any violated constraints to be in the final basis. Consequently, we state the following rules for $m_1 < i \leq m$ and some small positive number δ_1:

Rule 2: if $i \in B$, then $i \in \bar{B}$ iff $\lambda_i < 0$ or $c_i \leq \delta_1$; (25)

Rule 3: if $i \notin B$, then $i \in \bar{B}$ iff $\lambda_i < 0$ and $c_i \leq \delta_1$. (26)

Implicitly, Rule 2 gives the conditions for deleting a constraint from the basis; namely, the ith constraint is deleted from the basis when it is satisfied and has a nonnegative multiplier. Rule 3, in turn, gives conditions for adding a constraint; namely, the ith constraint is added iff its multiplier is negative and the constraint is violated.

Notice that the basis $\bar{B}$ cannot contain more than n constraints; yet, conceivably, there may be more than n constraints which are candidates for addition or deletion. In the basis determination procedure, the most violated (most negative) constraint is considered first. The constraints are then examined in the order of their violation until either the basis is full or all of the constraints have been checked.

In summary, the basis determination process takes place at a fixed point x. At this point, estimates of the Lagrange multipliers are computed by minimizing the error in the Kuhn–Tucker condition (19) or (22), subject to the inequality constraints (23). This process is posed as a quadratic programming problem in the variables λ. Using the estimates λ, the corresponding constraint values c, and the old basis B, we can construct a new basis $\bar{B}$, using Rules 1–3.

4. Extrapolation Procedure

4.1. Introduction. An algorithm has been defined for minimizing the augmented penalty function (12). This is an unconstrained problem which depends on the parameters λ and r; the preceding section detailed a method of estimating λ. It is not immediately obvious, however, that an unconstrained minimum of J corresponds to a point satisfying the Kuhn–Tucker condition

$$\nabla L(x, \lambda) = 0.$$

To demonstrate this fact, recall that an unconstrained minimum is characterized by the condition

$$\nabla J = \nabla L + r \nabla P = 0. \tag{27}$$

Define the multipliers

$$\omega_i = \begin{cases} \lambda_i + 2rc_i & \text{if } i \in B, \\ \lambda_i + 2rI[c_i]c_i & \text{if } i \in B'. \end{cases} \tag{28}$$

Now, it is clear from (16), (17), and (27) that

$$\nabla L(x, \omega) = \nabla f + G\omega = \nabla L(x, \lambda) + r\,\nabla P(x) = 0. \tag{29}$$

Thus, by correctly identifying the multipliers, it is clear that an unconstrained minimum of J corresponds to a point where the Kuhn–Tucker condition (6) is satisfied.

A second important point to note is that the Kuhn–Tucker condition

$$\nabla L(x, \lambda) = 0$$

implicitly defines some function $x(\lambda)$. At least in principle, it should be possible to vary λ, while keeping the Kuhn–Tucker condition satisfied. The primary objective of the extrapolation procedure is to choose λ, such that the constraints are satisfied, that is,

$$P(x(\lambda)) = 0,$$

while keeping (6) satisfied. An approach suggested by Hestenes (Ref. 3) is to use the ω vector defined by (28) as a new estimate for λ, keeping the penalty weight r fixed. In this method, the augmented penalty function is minimized for a sequence of λ's defined by (28); it has been demonstrated that the process is linearly convergent. Another approach, the standard penalty function method, varies the single parameter r, rather than the m values of λ. The constraints are satisfied as r becomes infinite and

$$\lim_{r\to\infty} 2rc_i = \lambda_i^*.$$

While this penalty trajectory is constructed such that

$$\nabla \hat{J} = \nabla f + r\,\nabla P = 0, \tag{30}$$

in general

$$\nabla f(x^*) \neq 0.$$

Therefore, as r becomes infinite, the gradient of the second term becomes larger. Because of the presence of these large gradients in the neighborhood of the solution, the unconstrained minimization of $\hat{J}$ becomes increasingly difficult. In contrast, the augmented penalty function is constructed such that

$$\nabla L(x^*) = 0,$$

as well as

$$\nabla J(x^*) = 0.$$

Consequently, the penalty weight r does not need to be large, and the augmented penalty approach avoids the ill-conditioning of the standard penalty function method.

4.2. Quadratic–Linear Extrapolation. A procedure that defines the multipliers λ such that the sequence of unconstrained minimization problems is quadratically convergent should retain the favorable numerical properties of the Hestenes method, while improving its convergence rate. To this end, let us assume that the basis B is fixed, and denote the m_B constraints in the basis by c_B, the Jacobian matrix by G_B, and the corresponding multipliers by λ_B. Assuming that the multipliers for the constraints in B' are zero, one can construct the function

$$L_B = f(x) + c_B^T(x)\lambda_B.$$

The Taylor series expansion for the gradient of the Lagrangian L_B with respect to the variable $z = (x, \lambda_B)$ is

$$\hat{g}_{L_B} = g_{L_B} + H_{L_B}(\hat{z} - z), \tag{31}$$

where

$$g_{L_B} = g_{L_B}(z) = \begin{bmatrix} \partial L_B/\partial x \\ \partial L_B/\partial \lambda_B \end{bmatrix},$$

$$H_{L_B} = H_{L_B}(x) = \begin{bmatrix} \partial^2 L_B/\partial x \partial x & \partial^2 L_B/\partial x \partial \lambda_B \\ \partial^2 L_B/\partial \lambda_B \partial x & \partial^2 L_B/\partial \lambda_B \partial \lambda_B \end{bmatrix}.$$

Let us choose the point $\hat{z} = (\hat{x}, \hat{\lambda}_B)$ such that

$$\hat{g}_{L_B} = g_{L_B}(\hat{z}) = 0,$$

which is what the Kuhn–Tucker conditions (6) and (9) require. Making use of the definitions (5), (16), and (18), we see that Eq. (31) becomes

$$\begin{bmatrix} \nabla f + G_B \lambda_B \\ c_B \end{bmatrix} + \begin{bmatrix} \nabla^2 L & G_B \\ G_B^T & 0 \end{bmatrix} \begin{bmatrix} \hat{x} - x \\ \hat{\lambda}_B - \lambda_B \end{bmatrix} = 0. \tag{32}$$

Now, (32) is a system of $n + m_B$ linear equations, which can be solved for

$$t = x - \hat{x}. \tag{33}$$

It should be remarked that, because of the partitioned form of the coefficient matrix, t can be determined without inverting the $(n + m_B)$-rank matrix. In fact, Fletcher notes (Ref. 8) that matrix inversion by partitioning leads to the standard projected gradient formulas for the search direction, at the cost of some additional matrix operations. In the computer implementation of the algorithm, we have chosen to solve for t directly. Furthermore, to ensure that the new point does in fact reduce the constraint violation, we choose

$$\bar{x} = x - \beta t, \tag{34}$$

where β is a scalar length determined such that

$$P(\bar{x}) < P(x).$$

A quadratic interpolation procedure has been found adequate to ensure a reduction in the penalty, and nominally $\beta = 1$. It is worth noting that, if B is correct, the performance index is quadratic, and the constraints are linear, then the point $\bar{x}$ is the constrained solution. Numerical experience indicates that this extrapolation procedure exhibits quadratic convergence, although we shall not prove it. Presumably, this is because (34) represents a single Newton–Raphson step with the stepsize chosen to guarantee penalty reduction.

4.3. Extrapolation When the Basis Is Full. The quadratic–linear extrapolation process yields a point which reduces the penalty term and, consequently, better satisfies the constraints. When the basis is full, the penalty term is minimized directly, in lieu of applying the quadratic linear extrapolation procedure. This is most readily implemented by using the unconstrained optimization algorithm defined in Section 3. Specifically, the Lagrangian term $L(x, \lambda)$ in Eq. (12) is ignored for least-square applications (Ref. 9).

5. Choice of the Penalty Weight

In this section, we consider the problem of defining a penalty weight at an arbitrary point x. A possible approach was developed in Ref. 1: choose r such that the norm of the gradient at x is minimized, i.e., such that

$$\hat{e}(r) = \|\nabla J\|^2 = (\nabla L + r\,\nabla P)^T(\nabla L + r\,\nabla P) \tag{35}$$

is a minimum. It is easily demonstrated that the optimal value of r is given by

$$r_e = -\nabla L^T\,\nabla P / \nabla P^T\,\nabla P. \tag{36}$$

A second factor that must be considered when choosing r is that the augmented penalty function must have a minimum. In other words, we require that $\nabla^2 J$ as given by (18) must be positive definite at the solution. This condition cannot necessarily be guaranteed, since the values of λ are estimated; consequently, $\nabla^2 L$ in (18) may not be positive definite. By choosing r large enough, however, the matrix $\nabla^2 J$ can be made positive definite. Since $\nabla^2 L$ is positive definite in some neighborhood of the solution, one can expect that there is some threshold value of r which will ensure the positive definiteness of $\nabla^2 J$. The existence of this threshold value has been alluded to by a number of authors, including Haarhoff and Buys (Ref. 10).

They observe that a value of r somewhat larger than the largest multiplier seems to be effective. We choose to use

$$r_m = 10 \max_i (\lambda_i^2). \tag{37}$$

It has been observed that, for a sufficiently large value of r, the set of inequality constraints in the basis remains fixed. If a large value of r is chosen initially, the unconstrained optimization problem is difficult to solve. Nevertheless, experience shows that forcing the value of r to increase from problem to problem aids in the basis determination process and also prevents cycling between two or more different basis estimates. Thus, if r^k is the value of the penalty weight used for the kth unconstrained minimization problem, let us define

$$r_b = \kappa r^k. \tag{38}$$

Nominally,

$$\kappa = 10 \qquad \text{and } r^1 = 1.$$

On the basis of these considerations, define the value of the penalty weight for the $(k+1)$st problem as

$$r^{k+1} = \max(r_e, r_m, r_b). \tag{39}$$

It is instructive to examine the behavior of this sequence as the point x approaches the solution x^*. The first quantity r_e is large when the basis is incorrect; it approaches zero when the basis is correct and the point is near the solution. The second quantity r_m approaches some constant nonzero value as the multipliers approach their values at the solution. Clearly, the third quantity becomes infinite. Nevertheless, numerical experience indicates that the overall solution is obtained before the penalty weight becomes prohibitively large, presumably because of the expected quadratic convergence of the extrapolation procedure.

Fiacco and McCormick (Ref. 11) demonstrate that, as x approaches the solution x^*, the product rP approaches zero. Although no statement is made regarding the monotonicity of the sequence, an indication of failure of the penalty approach can be made by checking that the product rP at the end of the kth optimization is actually less than the comparable quantity from the preceding problem. This situation can occur if the basis is incorrect or if the constraints are actually inconsistent, that is, they have no solution.

6. Nonlinear Programming Algorithm

The preceding sections have described various subprocesses in an overall nonlinear programming algorithm. In this section, we collect the

procedures to form the complete nonlinear programming algorithm, which we shall refer to as the accelerated multiplier optimization algorithm.

We now state the basic steps of the algorithm.

Step 1. Unconstrained Minimization. For a fixed basis B^k, fixed multipliers λ^k, and fixed penalty weight r^k, minimize the augmented penalty function (12) using the unconstrained optimization algorithm given in Section 2. Call the solution x^k.

Step 2. Check for a decrease in the penalty product rP:
(a) if $r^k P(x^k) < r^{k-1} P(x^{k-1})$, go to Step 3;
(b) if $r^k P(x^k) \geq r^{k-1} P(x^{k-1})$, the constraints may be inconsistent; go to Step 6.

Step 3. Basis Determination. Keeping x^k fixed, compute a new basis $\bar{B}$ and multipliers $\bar{\lambda}$, using the procedure described in Section 3.

Step 4. If the basis $\bar{B}$ is full, go to Step 6.

Step 5. Quadratic–Linear Extrapolation. Beginning at the point x^k, with the constraints in the basis $\bar{B}$ fixed, determine the point $\bar{x}$ using the quadratic–linear extrapolation procedure described in Section 4.2. Go to Step 7.

Step 6. Penalty Function Minimization. Beginning at x^k with the basis $\bar{B}$ fixed, minimize P, using the unconstrained algorithm given in Section 3. Call the solution $\bar{x}$. If $P(\bar{x}) \neq 0$, the constraints may be inconsistent.

Step 7. Basis Determination. Keeping $\bar{x}$ fixed, determine a new basis B^{k+1} and multipliers λ^{k+1}, using the procedure of Section 3. When checking for inconsistent constraints, B^{k+1} must be different from $\bar{B}$; if not, terminate.

Step 8. Convergence Test. Check for convergence, using the procedure given in Section 7. Terminate if convergence tests met.

Step 9. Define Penalty Weight. Compute the new penalty weight r^{k+1}, using method described in Section 5.

Step 10. Update All Information. Set $k = k+1$, $x^k = x^{k+1}$, $r^k = r^{k+1}$, $\lambda^k = \lambda^{k+1}$, $B^k = B^{k+1}$, etc. Return to Step 1.

A number of points regarding the overall algorithm now deserve clarification. First, observe that the sequence of operations always uses the current gradient and Hessian matrix information. For example, the Hessian matrices generated by the unconstrained algorithm in Step 1 are used to initiate either the quadratic–linear extrapolation or the penalty minimization. Secondly, when there is an indication of inconsistent constraints, the basis determination procedure constructs a new basis by deleting all the satisfied constraints from the old basis. If the new basis is the same as the old, one, the algorithm terminates in an error mode. Note that it is possible for the algorithm to terminate because of a local inconsistency in the constraints, i.e., a local minimum of P, where $P \neq 0$, even though a feasible region where $P = 0$ does exist.

Observe also that the algorithm is based on the philosophy of following the surface, satisfying the condition

$$\nabla L(x, \lambda) = 0$$

at the solution. Specifically, each cycle of the algorithm involves the unconstrained minimization which yields a point on the Lagrangian surface. The constraint satisfaction procedures, either quadratic–linear extrapolation or penalty minimization, begin from this Lagrangian surface. Furthermore, the basis determination procedure implements a rather conservative philosophy; namely, constraints are added to the basis only if they are violated and have negative multipliers; they are deleted only when they are satisfied and have positive multipliers. Since the basis determination procedure is used in conjunction with the penalty weight increases, experience indicates that there are relatively few basis changes with this philosophy. Furthermore, since each cycle of the algorithm involves returning to the surface $\nabla L(x, \lambda) = 0$ for a larger value of the penalty weight, cycling between one or more basis estimates should not occur.

7. Convergence Criteria

From a theoretical standpoint, the definition of convergence is quite clearcut; namely, the optimal solution must satisfy the Kuhn–Tucker conditions (6), (9), and (10). From a practical standpoint, however, the exact conditions can never be satisfied, because of limited precision in the computations. Therefore, one indication of convergence is that the error in the Kuhn–Tucker conditions (19) is small:

$$e(x, \lambda) < \delta_1, \tag{40}$$

where δ_1 is some specified small number. The primary weakness in this approach, however, is the need to specify δ_1. Indeed, a number which is small for one problem may be totally inadequate for another. Clearly, the number δ_1 is dependent on the relative scale of the problem quantities.

In essence, the test (40) defines an allowable error in the magnitude of the dependent variables. Because of the potential difficulties with this approach, it is usually desirable to also specify some acceptable error in the variation of the independent variables. Thus, we define the resolution convergence criterion:

$$\sigma_r = \|\hat{z} - z\|(\|z\| + 1)^{-1} < \delta_2. \tag{41}$$

Qualitatively, this test is satisfied when the two estimates of the optimum parameters $z = (x, \lambda_B)$ differ by less than $\log(1/\delta_2)$ significant digits.

Rather than actually compare two estimates of the optimum $\hat{z}$ and z, it is convenient to be able to implement this test at a single point. By using the Taylor series expansion (31) and assuming $\hat{g}_L = 0$ instead of (41), we can write

$$\sigma_r = \|H_{LB}^{-1} g_{L_B}\| (\|z\| + 1)^{-1} < \delta_2. \tag{42}$$

In general, one would expect that, if both the magnitude test (40) and the resolution test (42) are satisfied, then one could conclude that the algorithm has converged. Two other factors tend to obscure the issue. First, since δ_1 and δ_2 are specified by the user, conceivably they could be totally inconsistent. In fact, because of the limited precision of digital computers, if either δ_1 or δ_2 become too small, one can expect all of the numerical processes to break down. It is also conceivable that it might be impossible to satisfy both the convergence tests before the numerical processes break down, especially for poorly scaled problems. Consequently, we define convergence as the satisfaction of both the magnitude test (40) and the resolution test (42), unless one of the quantities δ_1, δ_2, e, or σ_r is so small that numerical difficulties seem evident.

As a final comment, a number of the numerical difficulties discussed can often be reduced or eliminated by proper scaling of the problem. Consequently, the standard mode of operation for the computer implementation of the algorithm scales both the independent variables x and the dependent variables f and c, using the procedure defined in Appendix 1 of Ref. 1.

8. Numerical Experience

The algorithm described in the previous section has been implemented in a digital computer program. A rather extensive set of test problems have been solved, using a CDC-7600 computer; this section presents the results of this numerical experience. Most of the problems have been drawn from the literature, and it is felt that the set is fairly representative of the kind of problems encountered in practice. It is the author's opinion that, despite the theoretical elegance of a general mathematical programming algorithm, one cannot espouse its numerical effectiveness without solving a set of test problems at least as broad as those given.

It is important that the reader be aware of the basic nature of the test problems. The test problems presented have two significant attributes, which are distinct from those encountered in practice. First, the function evaluation process is relatively inexpensive, and errors in the evaluation process are on the order of the machine accuracy. Similarly, accurate gradients can be obtained cheaply for the test problems.

In contrast, the real problems for which this algorithm has been designed possess neither of these attributes. In particular, function evaluations can be quite costly and may contain inaccuracies significantly larger than the machine accuracy. Because the function evaluations are so costly with respect to the computational expense of the optimization algorithm, we prefer to use the number of function evaluations as a measure of algorithm effectiveness.

Secondly, the problems for which this algorithm has been designed may require evaluation of the gradient information numerically. The test problems presented do not involve numerical derivatives, because it is the author's opinion that the process of numerical differentiation is a distinct numerical problem that should be considered independently of the optimization process. No attempt has been made to present an *equivalent* number of function evaluations, because such a quantity is highly dependent upon the numerical differentiation procedure. Such quantities as perturbation sizes and error tolerances can greatly influence the accuracy of numerical derivatives and consequently obscure the overall behavior of the optimization process. Let it suffice to say that the algorithm presented requires gradient information. If this information must be obtained numerically one can expect a twofold degradation (i) because each gradient evaluation will require one or more additional function evaluations and (ii) because inaccurate gradient information will necessitate more optimization iterations.

Table 1. Equality constrained problems.

Problem	Number of function and gradient evaluations	Number of unconstrained problems
11.1	8	1
11.2	13	2
11.3	23	2
11.4	16	2
11.5	19	1
11.6	20	2
11.7	18	2
11.8	18	2
11.9	6	0
11.10	6*	0
11.11	17†	0
11.12	28	1

* Algorithm terminated at an unfeasible point [$\nabla P(x^*) = 0$, although $P(x^*) \neq 0$ and $P(x) = 0$ does exist].

† Inconsistent constraints [$P(x^*) = 0$ does not exist].

The test problems have been organized into two separate categories, (a) equality constrained problems and (b) inequality constrained problems. The complete problems are stated in Appendices B and C with references where applicable. The results are presented in condensed form for each of the categories in Tables 1–2. Specifically, we present the number of function and gradient evaluations required for convergence. For example, the first problem was solved using exactly eight evaluations of the objective function and the constraints and eight evaluations of the gradient vectors. For all problems, convergence was defined by $\delta_1 = \delta_2 = 10^{-5}$; unless noted otherwise, all other parameters were initialized as described in the text. All problems were run using the same algorithmic parameters, and no attempt was made to *tune* the results for each problem. The number of times the augmented penalty function (12) was minimized is presented as an indication of the number of

Table 2. Inequality constrained problems.

Problem	Number of function and gradient evaluations	Number of unconstrained problems
12.1	19	1
12.2	5	1
12.3	21	2
12.4	36	3
12.5	19	1
12.6	54	2
12.7	47	4
12.8	67	3
12.9	34	3
12.10	28	2
12.11	46	2
12.12	42	3
12.13	76	3
12.14	21	2
12.15	18	2
12.16	18	1
12.17	39	3
12.18	51	4
12.19	24	2
12.20	67	3
12.21	14	1
12.22	15	2
12.23	46	4
12.24	52	3
12.25	37	2
12.26	37	3
12.27	4	0

cycles performed by the method. It is worthwhile to note that, although the algorithm has been successfully applied to the 39 test problems described in this paper, as well as the 18 problems in Ref. 5 and a number of practical problems, the algorithm has been unsuccessful in some instances. In particular, the algorithm has failed to solve the second test problem given in Colville (Ref. 12) and problem A given by Box (Ref. 13) when no basis was specified.

9. Summary and Conclusions

This paper describes an accelerated multiplier method for solving the general nonlinear programming problem. The algorithm poses a sequence of unconstrained optimization problems, which are solved using a rank-one recursive procedure described in an earlier paper (Ref. 5). Estimates of the Lagrange multipliers are obtained by minimizing the error in the Kuhn–Tucker conditions using a quadratic programming algorithm. The convergence of the sequence of unconstrained problems is accelerated by using a Newton–Raphson extrapolation process. Although we do not prove quadratic convergence, numerical experience tends to confirm this assertion. Furthermore, although no proof is given that the algorithm cannot cycle, numerical experience with the philosophy of returning to the Lagrangian surface $\nabla L(x, \lambda) = 0$, once per cycle, for increasing penalty weights, seems to indicate that cycling does not occur. The fact that a relatively small number of unconstrained minimizations are required to solve the overall problem, as illustrated by the results presented in Tables 1–2, tends to corroborate the assertion of quadratic convergence without cycling.

10. Appendix A: Nonlinear Programming Test Problems

Appendices B and C present the set of test problems used to assess the effectiveness of the nonlinear programming algorithm described. The problems are organized into two categories: (i) equality constrained problems and (ii) inequality constrained problems.

When no information is given to the contrary, one can assume that all the quantities used by the numerical processes are completely scaled, in the sense described in the Appendix of Ref. 1, in the range $-20 \leq x \leq 20$. Unless noted otherwise, the initial penalty weight is $r^0 = 1$, and the initial basis is assumed to be empty, i.e.,

$$B^0 = \{\Phi\}.$$

The points x^* presented are assumed to have converged in the sense

described in Section 7, where

$$\delta_1 = \delta_2 = 10^{-5}.$$

When the exact solution is known, its value is presented following the computationally obtained value.

11. Appendix B: Equality Constrained Problems

Problem 11.1. (Ref. 14). Minimize

$$f(x) = (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2,$$

subject to

$$c_1(x) = x_1 + 3x_2 = 0,$$
$$c_2(x) = x_3 + x_4 - 2x_5 = 0,$$
$$c_3(x) = x_2 - x_5 = 0,$$
$$-10 \leq x \leq 10.$$

Starting point: $x^0 = (2, 2, 2, 2, 2)$.

Solution point: $x^* = (-7.6744 \times 10^{-1}, 2.5581 \times 10^{-1}, 6.2790 \times 10^{-1}, -1.1627 \times 10^{-1}, 2.5581 \times 10^{-1})$,

$f^* = 4.0930$.

Remarks: No scaling.

Problem 11.2. (Ref. 14). Minimize

$$f(x) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^4,$$

subject to

$$c_1(x) = x_1(1 + x_2^2) + x_3^4 - 4 - 3\sqrt{2} = 0,$$
$$-10 \leq x \leq 10.$$

Starting point: $x^0 = (2, 2, 2)$.

Solution point: $x^* = (1.1048, 1.1966, 1.5352)$,

$f^* = 3.2568 \times 10^{-2}$.

Remarks: No scaling.

Problem 11.3. (Ref. 14). Minimize

$$f(x) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6,$$

subject to

$$c_1(x) = x_4 x_1^2 + \sin(x_4 - x_5) - 2\sqrt{2} = 0,$$
$$c_2(x) = x_2 + x_3^4 x_4^2 - 8 - \sqrt{2} = 0,$$
$$-10 \leq x \leq 10.$$

Starting point: $x^0 = (2, 2, 2, 2, 2)$.

Solution point: $x^* = (1.1661, 1.1821, 1.3802, 1.5060, 6.1092 \times 10^{-1})$,

$f^* = 2.4150 \times 10^{-1}$.

Remarks: No scaling.

Problem 11.4. (Ref. 14). Minimize

$$f(x) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^4 + (x_4 - x_5)^4,$$

subject to

$$c_1(x) = x_1 + x_2^2 + x_3^3 - 2 - 3\sqrt{2} = 0,$$
$$c_2(x) = x_2 - x_3^2 + x_4 + 2 - 2\sqrt{2} = 0,$$
$$c_3(x) = x_1 x_5 - 2 = 0,$$
$$-10 \leq x \leq 10.$$

Starting point: $x^0 = (2, 2, 2, 2, 2)$.

Solution point: $x^* = (1.1911, 1.3626, 1.4728, 1.6350, 1.6790)$,

$f^* = 7.8776 \times 10^{-2}$.

Remarks: No scaling.

Problem 11.5. (Ref. 14). Minimize

$$f(x) = (x_1 - x_2)^2 + (x_2 - x_3)^4,$$

subject to

$$c_1(x) = x_1(1 + x_2^2) + x_3^4 - 3 = 0,$$
$$-10 \leq x \leq 10.$$

Starting point: $x^0 = (2, 2, 2)$.

Solution point: $x^* = (9.9994 \times 10^{-1},\ 9.9994 \times 10^{-1},\ 1.0000) \cong (1, 1, 1)$,

$f^* = 1.6745 \times 10^{-16} \cong 0$.

Remarks: No scaling.

Table 3. Data for Problem 11.6.

i	a_i	t_i
1	0	0
2	50	25
3	50	50
4	75	100
5	75	150
6	75	200
7	100	290
8	100	380

Problem 11.6. Maximize

$$f(x) = \dot{p}_8^2,$$

subject to

$$c_1(x) = q_8 - 100{,}000 = 0,$$

$$c_2(x) = \dot{q}_8 - 1000 = 0,$$

where

$$p_i = \tfrac{1}{2}a_i(t_i - t_{i-1})^2 \cos x_{i-1} + \dot{p}_{i-1}(t_i - t_{i-1}) + p_{i-1},$$

$$q_i = \tfrac{1}{2}(t_i - t_{i-1})^2(a_i \sin x_{i-1} - g) + \dot{q}_{i-1}(t_i - t_{i-1}) + q_{i-1},$$

$$\dot{p}_i = a_i(t_i - t_{i-1}) \cos x_{i-1} + \dot{p}_{i-1},$$

$$\dot{q}_i = (t_i - t_{i-1})(a_i \sin x_{i-1} - g) + \dot{q}_{i-1},$$

$$i = 2, \ldots, 8, \qquad p_1 = q_1 = \dot{p}_1 = \dot{q}_1 = 0, \qquad g = 32.$$

The values of a_i and t_i are given in Table 3 and

$$0 \le x \le 1.58.$$

Starting point: $x^0 = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$.

Solution point: $x^* = (5.4246 \times 10^{-1}, 5.2902 \times 10^{-1}, 5.0844 \times 10^{-1}, 4.8026 \times 10^{-1}, 4.5123 \times 10^{-1}, 4.0918 \times 10^{-1}, 3.5278 \times 10^{-1})$,

$$f^* = 8.3107 \times 10^8.$$

Problem 11.7. (Ref. 15). Maximize

$$f(x) = \exp(-Q/2),$$

where

$$Q=(1-\rho^2)^{-1}[(x_1-\mu_1)^2/\sigma_1^2+2\rho(x_1-\mu_1)(x_2-\mu_2)/\sigma_1\sigma_2+(x_2-\mu_2)^2/\sigma_2^2]$$
$$+(x_3-\mu_3)^2/\sigma_3^2+(x_4-\mu_4)^2/\sigma_4^2+(x_5-\mu_5)^2/\sigma_5^2+(x_6-\mu_6)^2/\sigma_6^2,$$

subject to

$$c_1(x)=(x_1-\mu_1)-0.2\sigma_1+4000(x_2-\mu_2)-2000\sigma_2=0,$$

with $\rho=0.2$ and

$$\begin{aligned}
&\mu_1=10{,}000, && \sigma_1=8000,\\
&\mu_2=1, && \sigma_2=1,\\
&\mu_3=2\times10^6, && \sigma_3=7\times10^6,\\
&\mu_4=10, && \sigma_4=50,\\
&\mu_5=0.001, && \sigma_5=0.05,\\
&\mu_6=1\times10^8, && \sigma_6=5\times10^8,\\
&0\le x_1\le 2\times10^4, && -10\le x_2\le 10,\\
&0\le x_3\le 1\times10^7, && 0\le x_4\le 20,\\
&-1\le x_5\le 1, && 0\le x_6\le 2\times10^8.
\end{aligned}$$

Starting point: $x^0=(6\times10^3, 1.5, 4\times10^6, 2, 3\times10^{-3}, 5\times10^7)$.

Solution point: $x^*=(1.2670\times10^4, 1.2322, 1.9999\times10^6, 10, 1\times10^{-3}, 10^8)$,

$$f^*=9.3676\times10^{-1}.$$

Problem 11.8. (Ref. 15). Maximize

$$f(x)=gI_1\log[(x_1+x_2+x_3+W)/(x_1a_1+x_2+x_3+W)]$$
$$+gI_2\log[(x_2+x_3+W)/(x_2a_2+x_3+W)]$$
$$+gI_3\log[(x_3+W)/(x_3a_3+W)],$$

subject to

$$c_1(x)=x_1+x_2+x_3-1=0,$$
$$0\le x\le 1,$$

where

$$\begin{aligned}
&g=32.174, && a_1=0.09, && a_2=0.07, && a_3=0.13,\\
&W=0.03, && I_1=255, && I_2=280, && I_3=290.
\end{aligned}$$

Starting point: $x^0 = (0.7, 0.2, 0.1)$.

Solution point: $x^* = (0.61781, 0.32820, 5.3985 \times 10^{-2})$,

$f^* = 2.6272 \times 10^4$.

Remarks: Scale the independent variables. Scale the objective function by 10^{-5}.

Problem 11.9. Maximize

$$f(x) = 1,$$

subject to

$$c_1(x) = x_1^2 + x_2^2 - 25 = 0,$$

$$c_2(x) = x_1 x_2 - 9 = 0,$$

$$-100 \leq x \leq 100.$$

Starting point: $x^0 = (2, 1)$.

Solution point: $x^* = (4.6015, 1.9558)$,

$f^* = 1$.

Remarks: No scaling.

Problem 11.10. Maximize

$$f(x) = 1,$$

subject to

$$c_1(x) = x_1^2 + x_2^2 - 25 = 0,$$

$$c_2(x) = x_1 x_2 - 9 = 0,$$

$$-100 \leq x \leq 100.$$

Starting point: $x^0 = (2, 2)$.

Solution point: $x^* = (3.4351, 3.4351)$.

$f^* = 1$.

Remarks: No scaling. Computed solution point is unfeasible. Actual solution point is the same as in Problem 11.9.

Problem 11.11. (Ref. 15). Minimize

$$f(x) = 1,$$

subject to

$$c_1(x) = x_1^2 + x_2^2 - 25 = 0,$$
$$c_2(x) = x_1 x_2 - 25 = 0,$$
$$-100 \leq x \leq 100.$$

Starting point: $x^0 = (5, 8)$.

Remarks: Inconsistent constraints. No scaling.

Problem 11.12. Minimize

$$f(x) = (1 - x_1)^2,$$

subject to

$$c_1(x) = 10(x_2 - x_1^2).$$

Starting point: $x^0 = (-1.2, 1)$.

Solution point: $x^* = (1, 1)$,

$$f^* = 3.2311 \times 10^{-27} \cong 0.$$

Remarks: No scaling.

12. Appendix C: Inequality Constrained Problems

Problem 12.1. (Ref. 14). Minimize

$$f(x) = 2 - (1/120) x_1 x_2 x_3 x_4 x_5,$$

subject to

$$c_i(x) = x_i \geq 0, \qquad i = 1, \ldots, 5,$$
$$c_{i+5}(x) = i - x_i \geq 0, \qquad i = 1, \ldots, 5,$$
$$-10 \leq x \leq 10.$$

Starting point: $x^0 = (2,2,2,2,2)$.

Solution point: $x^* = (1, 2, 3, 4, 5)$,

$$f^* = 1.$$

Remarks: No scaling.

Problem 12.2. Maximize

$$f(x) = 100 - (0.01 x_1^2 + x_2^2),$$

subject to

$$c_1(x) = x_1 - 2 \geq 0,$$
$$c_2(x) = 10x_1 - x_2 - 10 \geq 0,$$
$$-50 \leq x \leq 50.$$

Starting point: $x^0 = (-1, -1)$.

Solution point: $x^* = (2, -1.6957 \times 10^{-17}) \cong (2, 0)$,

$f^* = 99.96$.

Remarks: No scaling.

Problem 12.3. Minimize

$$f(x) = 0.01x_1^2 + x_2^2,$$

subject to

$$c_1(x) = x_1 x_2 - 25 \geq 0,$$
$$c_2(x) = x_1^2 + x_2^2 - 25 \geq 0,$$
$$c_3(x) = x_1 - 2 \geq 0,$$
$$0 \leq x \leq 50.$$

Starting point: $x^0 = (2, 2)$.

Solution point: $x^* = (15.811, 1.5811)$,

$f^* = 5.0$.

Remarks: No scaling.

Problem 12.4. Minimize

$$f(x) = x_1^2 + x_2^2,$$

subject to

$$c_1(x) = x_1^2 + x_2^2 - 1 \geq 0,$$
$$c_2(x) = 9x_1^2 + x_2^2 - 9 \geq 0,$$
$$c_3(x) = x_1 + x_2 - 1 \geq 0,$$
$$c_4(x) = x_1^2 - x_2 \geq 0,$$
$$c_5(x) = x_2^2 - x_1 \geq 0,$$
$$-50 \leq x \leq 50.$$

Starting point: $x^0 = (3, 1)$.

Solution point: $x^* = (1, 1)$,

$f^* = 2$.

Remarks: No scaling.

Problem 12.5. Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_1 x_2 - 1 \geq 0,$$
$$c_2(x) = x_2^2 + x_1 \geq 0,$$
$$c_3(x) = -x_1 + \tfrac{1}{2} \geq 0.$$

Starting point: $x^0 = (-2, 1)$.

Solution point: $x^* = (0.5, 2)$,

$f^* = 3.0650 \times 10^2$.

Problem 12.6. Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_2 + 1.5 \geq 0.$$

Starting point: $x^0 = (-2, 1)$,

$B^0 = \{1\}$.

Solution point: $x^* = (1, 1)$,

$f^* = 7.5439 \times 10^{-19} \cong 0$.

Remarks: No scaling.

Problem 12.7. Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_2^2 + x_1 \geq 0,$$
$$c_2(x) = x_1^2 + x_2 \geq 0,$$

$$c_3(x) = -x_1 + \tfrac{1}{2} \geq 0,$$
$$c_4(x) = x_1 + \tfrac{1}{2} \geq 0,$$
$$c_5(x) = -x_2 + 1 \geq 0.$$

Starting point: $x^0 = (-2, 1)$.

Solution point: $x^* = (0.5, 0.25)$,

$$f^* = 0.25.$$

Remarks: No scaling.

Problem 12.8. Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_2^2 + x_1 \geq 0,$$
$$c_2(x) = x_1^2 + x_2 \geq 0,$$
$$c_3(x) = -x_1 + \tfrac{1}{2} \geq 0,$$
$$c_4(x) = x_1 + \tfrac{1}{2} \geq 0,$$
$$c_5(x) = x_1^2 + x_2^2 - 1 \geq 0.$$

Starting point: $x^0 = (-2, 1)$.

Solution point: $x^* = (0.5, 0.86602)$,

$$f^* = 38.198.$$

Remarks: No scaling.

Problem 12.9. Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_2 - 1.5 \geq 0.$$

Starting point: $x^0 = (-2, 1)$,

$$B^0 = \{1\}.$$

Solution point: $x^* = (1.2243, 1.5)$,

$$f^* = 5.0426 \times 10^{-2}.$$

Remarks: No scaling.

Problem 12.10. Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_2^2 - x_1 \geq 0,$$
$$c_2(x) = x_1^2 - x_2 \geq 0,$$
$$c_3(x) = -x_1 + 0.5 \geq 0,$$
$$c_4(x) = x_1 + 0.5 \geq 0,$$
$$c_5(x) = -x_2 + 1 \geq 0.$$

Starting point: $x^0 = (-2, 1)$,

$r^0 = 1000.$

Solution point: $x^* = (-1.6606 \times 10^{-12}, -3.3207 \times 10^{-7}) \cong (0, 0)$,

$f^* = 1.0.$

Remarks: No scaling.

Problem 12.11. Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_2^2 - x_1 \geq 0,$$
$$c_2(x) = x_1^2 - x_2 \geq 0,$$
$$c_3(x) = -x_1 + \tfrac{1}{2} \geq 0,$$
$$c_4(x) = x_1 + \tfrac{1}{2} \geq 0,$$
$$c_5(x) = x_1^2 + x_2^2 - 1 \geq 0.$$

Starting point: $x^0 = (-2, 1)$,

$r^0 = 10{,}000.$

Solution point: $x^* = (9.9009 \times 10^{-3}, 9.9995 \times 10^{-1})$,

$f^* = 100.99.$

Remarks: No scaling.

Problem 12.12. Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_2 - 1.5 \geq 0.$$

Starting point: $x^0 = (0, 1.5)$,

$B^0 = \{1\}$.

Solution point: $x^* = (1.2243, 1.5)$,

$f^* = 5.0426 \times 10^{-2}$.

Remarks: No scaling.

Problem 12.13. Minimize

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

subject to

$$c_1(x) = x_2 - 1.5 \geq 0.$$

Starting point: $x^0 = (2, 1)$,

$B^0 = \{1\}$.

Solution point: $x^* = (1.2243, 1.5)$,

$f^* = 5.0426 \times 10^{-2}$.

Remarks: No scaling.

Problem 12.14. Minimize

$$f(x) = x_1^2 + x_2^2 + x_3^2,$$

subject to

$$c_1(x) = x_1 - 1 \geq 0,$$

$$c_2(x) = x_1^2 + x_2^2 - 1 \geq 0,$$

$$-10 \leq x \leq 10.$$

Starting point: $x^0 = (1, 1, 1)$.

Solution point: $x^* = (1, -6.3007 \times 10^{-6}, -1.4704 \times 10^{-22}) \cong (1, 0, 0)$,

$f^* = 1.0$.

Remarks: No scaling.

Problem 12.15. Minimize

$$f(x) = 9x_1^2 + x_2^2 + 9x_3^2,$$

subject to

$$c_1(x) = x_2 - 1 \geq 0,$$
$$c_2(x) = x_1 x_2 - 1 \geq 0,$$
$$c_3(x) = 1 - x_3 \geq 0,$$
$$-10 \leq x \leq 10.$$

Starting point: $x^0 = (1, 1, 1)$.

Solution point: $x^* = (0.57735, 1.7320, -2.5096 \times 10^{-21})$,

$f^* = 6.0$.

Remarks: No scaling.

Problem 12.16. (Ref. 16). Minimize

$$f(x) = (x_1 - 2)^2 + x_2^2,$$

subject to

$$c_1(x) = x_1 \geq 0,$$
$$c_2(x) = x_2 \geq 0,$$
$$c_3(x) = (1 - x_1)^3 - x_2 \geq 0.$$

Starting point: $x^0 = (-2, -2)$.

Solution point: $x^* = (0.99681, 2.7463 \times 10^{-8}) \cong (1, 0)$,

$f^* = 1.0063 \cong 1$.

Remarks: No scaling. Linear dependent constraints at the solution.

Problem 12.17. (Ref. 12). Minimize

$$f(x) = \sum_{j=1}^{5} \epsilon_j x_j + \sum_{j=1}^{5} \sum_{i=1}^{5} \gamma_{ij} x_i x_j + \sum_{j=1}^{5} \delta_j x_j^3,$$

subject to

$$c_i(x) = \sum_{j=1}^{5} \alpha_{ij} x_j - \beta_i \geq 0, \qquad i = 1, \ldots, 10,$$

$$c_{j+10}(x) = x_j \geq 0, \qquad j = 1, \ldots, 5,$$

where α, β, γ, δ, ϵ are given in Tables 4–7 and

$$-100 \leq x \leq 100.$$

Starting point: $x^0 = (0, 0, 0, 0, 1)$.

Table 4. Data for Problem 12.17.

j	ϵ_j	δ_j
1	−15	4
2	−27	8
3	−36	10
4	−18	6
5	−12	2

Table 5. Data for Problem 12.17.

i	β_i	i	β_i
1	−40	6	−1
2	−2	7	−40
3	−0.25	8	−60
4	−4	9	5
5	−4	10	1

Table 6. Data for Problem 12.17, γ_{ij}.

	j				
i	1	2	3	4	5
1	30	−20	−10	32	−10
2	−20	39	−6	−31	32
3	−10	−6	10	−6	−10
4	32	−31	−6	39	−20
5	−10	32	−10	−20	30

Table 7. Data for Problem 12.17, α_{ij}.

	j				
i	1	2	3	4	5
1	−16	2	0	1	0
2	0	−2	0	0.4	2
3	−3.5	0	2	0	0
4	0	−2	0	−4	−1
5	0	−9	−2	1	−2.8
6	2	0	−4	0	0
7	−1	−1	−1	−1	−1
8	−1	−2	−3	−2	−1
9	1	2	3	4	5
10	1	1	1	1	1

Solution point: $x^* = (0.3, 0.33346, 0.4, 0.42831, 0.22396)$,

$$f^* = -32.348.$$

Remarks: Independent variable scaling.

Problem 12.18. (Ref. 12). Minimize

$$f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141,$$

subject to

$$c_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \geq 0,$$

$$c_2(x) = 92 - c_1(x) \geq 0,$$

$$c_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 90 \geq 0,$$

$$c_4(x) = 20 - c_3(x) \geq 0,$$

$$c_5(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 20 \geq 0,$$

$$c_6(x) = 5 - c_5(x) \geq 0,$$

$$c_7(x) = x_1 - 78 \geq 0,$$

$$c_8(x) = 102 - x_1 \geq 0,$$

$$c_9(x) = x_2 - 33 \geq 0,$$

$$c_{10}(x) = 45 - x_2 \geq 0,$$

$$c_{11}(x) = x_3 - 27 \geq 0,$$

$$c_{12}(x) = 45 - x_3 \geq 0,$$

$$c_{13}(x) = x_4 - 27 \geq 0,$$

$$c_{14}(x) = 45 - x_4 \geq 0,$$

$$c_{15}(x) = x_5 - 27 \geq 0,$$

$$c_{16}(x) = 45 - x_5 \geq 0,$$

$$-1000 \leq x \leq 1000.$$

Starting point: $x^0 = (78.62, 33.44, 31.07, 44.18, 35.32)$.

Solution point: $x^* = (78, 33, 29.995, 45, 36.775)$,

$$f^* = -3.0665 \times 10^4.$$

Problem 12.19. (Ref. 14). Minimize

$$f(x) = 2 - x_1 x_2 x_3,$$

subject to

$$c_1(x) = x_1 + 2x_2 + 2x_3 - x_4 = 0,$$

$$c_{2i}(x) = x_i \geq 0, \qquad i = 1, 2, 3, 4,$$

$$c_{2i+1}(x) = 1 - x_i \geq 0, \qquad i = 1, 2, 3,$$

$$c_9(x) = 2 - x_4 \geq 0,$$

$$-10 \leq x \leq 10.$$

Starting point: $x^0 = (2, 2, 2, 2)$.

Solution point: $x^* = 0.66666, 0.33333, 0.33333, 2)$,

$$f^* = 1.9259.$$

Remarks: No scaling.

Problem 12.20. (Ref. 12). Minimize

$$f(x) = \sum_{i=1}^{16} \sum_{j=1}^{16} A_{ij}(x_i^2 + x_i + 1)(x_j^2 + x_j + 1),$$

subject to

$$c_1(x) = 0.22x_1 + 0.20x_2 + 0.19x_3 + 0.25x_4 + 0.15x_5 + 0.11x_6 + 0.12x_7 + 0.13x_8 + x_9 - 2.5 = 0,$$

$$c_2(x) = -1.46x_1 - 1.3x_3 + 1.82x_4 - 1.15x_5 + 0.8x_7 + x_{10} - 1.10 = 0,$$

$$c_3(x) = 1.29x_1 - 0.89x_2 - 1.16x_5 - 0.96x_6 - 0.49x_8 + x_{11} + 3.10 = 0,$$

$$c_4(x) = -1.10x_1 - 1.06x_2 + 0.95x_3 - 0.54x_4 - 1.78x_6 - 0.41x_7 + x_{12} + 3.50 = 0,$$

$$c_5(x) = -1.43x_4 + 1.51x_5 + 0.59x_6 - 0.33x_7 - 0.43x_8 + x_{13} - 1.30 = 0,$$

$$c_6(x) = -1.72x_2 - 0.33x_3 + 1.62x_5 + 1.24x_6 + 0.21x_7 - 0.26x_8 + x_{14} - 2.10 = 0,$$

$$c_7(x) = 1.12x_1 + 0.31x_4 + 1.12x_7 - 0.36x_9 + x_{15} - 2.30 = 0,$$

$$c_8(x) = 0.45x_2 + 0.26x_3 - 1.10x_4 + 0.58x_5 - 1.03x_7 + 0.1x_8 + x_{16} + 1.50 = 0,$$

$$c_{i+8}(x) = x_i \geq 0, \qquad i = 1, \ldots, 16,$$

$$c_{i+24}(x) = 5 - x_i \geq 0, \qquad i = 1, \ldots, 16,$$

Table 8. Data for Problem 12.20, A_{ij}.

	j															
i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1			1			1	1								1
2		1	1				1			1						
3			1				1		1	1				1		
4				1			1				1				1	
5					1	1				1		1				1
6						1		1							1	
7							1				1		1			
8								1		1					1	
9									1			1				1
10										1				1		
11											1		1			
12												1		1		
13													1	1		
14														1		
15															1	
16																1

where A_{ij} is given in Table 8 and

$$-10 \leq x \leq 20.$$

Starting point: $x_i^0 = 0, i = 1, \ldots, 16.$

Solution point: $x^* = (0.039847, 0.79198, 0.20287, 0.84435, 1.2699, 0.93473, 1.6819, 0.15530, 1.5678, -5.6843 \times 10^{-14}, 0, 0, 0.66020, 0, 0.67425, -5.6843 \times 10^{-14}),$

$$f^* = 2.4489 \times 10^2.$$

Problem 12.21. (Ref. 13). Maximize

$$f(x) = [9 - (x_1 - 3)^2](x_2^3/27\sqrt{3}),$$

subject to

$$c_1(x) = x_1 \geq 0,$$

$$c_2(x) = x_2 \geq 0,$$

$$c_3(x) = x_1/\sqrt{3} - x_2 \geq 0,$$

$$c_4(x) = x_1 + \sqrt{3}x_2 \geq 0,$$

$$c_5(x) = 6 - c_4(x) \geq 0.$$

Starting point: $x^0 = (1, 0.5)$.

Solution point: $x^* = (3, 1.7320) \cong (3, \sqrt{3})$,

$f^* = 1$.

Remarks: No scaling.

Problem 12.22. (Ref. 13). Maximize

$$f(x) = x_1 x_2 x_3,$$

subject to

$$c_i(x) = x_i \geq 0, \qquad i = 1, 2, 3,$$

$$c_{i+3}(x) = 42 - x_i \geq 0, \qquad i = 1, 2, 3,$$

$$c_7(x) = x_1 + 2x_2 + 2x_3 \geq 0,$$

$$c_8(x) = 72 - c_7(x) \geq 0,$$

$$-100 \leq x \leq 100.$$

Starting point: $x^0 = (10, 10, 10)$,

$r^0 = 100$.

Solution point: $x^* = (24, 12, 12)$,

$f^* = 3.456 \times 10^3$.

Problem 12.23. (Ref. 17). Minimize

$$f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4,$$

subject to

$$c_1(x) = -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 \geq 0,$$

$$c_2(x) = -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 \geq 0,$$

$$c_3(x) = -2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 \geq 0.$$

Starting point: $x^0 = (0, 0, 0, 0)$.

Solution point: $x^* = (0, 1, 2, -1)$,

$f^* = -44$.

Problem 12.24. (Ref. 17). Minimize

$$f(x) = \sum_{i=1}^{44} \{b_i - x_1 - (0.49 - x_1) \exp[-x_2(a_i - 8)]\}^2,$$

Table 9. Data for Problem 12.24.

i	a_i	b_i	i	a_i	b_i
1	8	0.49	23	22	0.41
2	8	0.49	24	22	0.40
3	10	0.48	25	24	0.42
4	10	0.47	26	24	0.40
5	10	0.48	27	24	0.40
6	10	0.47	28	26	0.41
7	12	0.46	29	26	0.40
8	12	0.46	30	26	0.41
9	12	0.45	31	28	0.41
10	12	0.43	32	28	0.40
11	14	0.45	33	30	0.40
12	14	0.43	34	30	0.40
13	14	0.43	35	30	0.38
14	16	0.44	36	32	0.41
15	16	0.43	37	32	0.40
16	16	0.43	38	34	0.40
17	18	0.46	39	36	0.41
18	18	0.45	40	36	0.38
19	20	0.42	41	38	0.40
20	20	0.42	42	38	0.40
21	20	0.43	43	40	0.39
22	22	0.41	44	42	0.39

subject to

$$c_1(x) = -x_1x_2 + 0.49x_2 - 0.09 \geq 0,$$

$$c_2(x) = x_1 - 0.4 \geq 0,$$

where a_i and b_i are given in Table 9.

Starting point: $x^0 = (0.42, 5)$.

Solution point: $x^* = (0.41995, 1.2848)$,

$$f^* = 2.8459 \times 10^{-2}.$$

Problem 12.25. (Ref. 17). Minimize

$$f(x) = (x_1 - 10)^3 + (x_2 - 20)^3,$$

subject to

$$c_1(x) = x_1 - 13 \geq 0,$$

$$c_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2 - 100 \geq 0,$$

$$c_3(x) = -(x_1-6)^2-(x_2-5)^2+82.81 \geq 0,$$

$$c_4(x) = x_2 \geq 0,$$

$$-100 \leq x \leq 100.$$

Starting point: $x^0 = (20.1, 5.84)$.

Solution point: $x^* = (14.095, 0.84296)$,

$$f^* = -6.9618 \times 10^3.$$

Problem 12.26. (Ref. 13). Maximize

$$f(x) = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + a_5x_5,$$

subject to

$$c_1(x) = x_1 \geq 0,$$

$$c_2(x) = x_2 \geq 0,$$

$$c_3(x) = x_3 \geq 0,$$

$$c_4(x) = x_4 \geq 0,$$

$$c_5(x) = x_5 \geq 0,$$

$$c_6(x) = 2.4x_1 - x_2 \geq 0,$$

$$c_7(x) = -1.2x_1 + x_2 \geq 0,$$

$$c_8(x) = 60x_1 - x_3 \geq 0,$$

$$c_9(x) = -20x_1 + x_3 \geq 0,$$

$$c_{10}(x) = 9.3x_1 - x_4 \geq 0,$$

$$c_{11}(x) = -9.0x_1 + x_4 \geq 0,$$

$$c_{12}(x) = 7.0x_1 - x_5 \geq 0,$$

$$c_{13}(x) = -6.5x_1 + x_5 \geq 0,$$

$$c_{14}(x) = a_6x_1 + a_7x_2 + a_8x_3 + a_9x_4 + a_{10}x_5 \geq 0,$$

$$c_{15}(x) = -a_6x_1 - a_7x_2 - a_8x_3 - a_9x_4 - a_{10}x_5 + 294{,}000 \geq 0,$$

$$c_{16}(x) = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 \geq 0,$$

$$c_{17}(x) = -a_{11}x_1 - a_{12}x_2 - a_{13}x_3 - a_{14}x_4 - a_{15}x_5 + 294{,}000 \geq 0,$$

$$c_{18}(x) = a_{16}x_1 + a_{17}x_2 + a_{18}x_3 + a_{19}x_4 + a_{20}x_5 \geq 0,$$

$$c_{19}(x) = -a_{16}x_1 - a_{17}x_2 - a_{18}x_3 - a_{19}x_4 - a_{20}x_5 + 277{,}200 \geq 0,$$

Table 10. Data for Problems 12.26 and 12.27.

i	a_i	i	a_i
0	−24345		
1	−8720288.849	11	−155011.1084
2	150512.5253	12	4360.53352
3	−156.6950325	13	12.9492344
4	476470.3222	14	10236.884
5	729482.8271	15	13176.786
6	−145421.402	16	−326669.5104
7	2931.1506	17	7390.68412
8	−40.427932	18	−27.8986976
9	5106.192	19	16643.076
10	15711.36	20	30988.146

where the a_i are given in Table 10 and

$$-1000 \leq x \leq 1000.$$

Starting point: $x^0 = (2.52, 5.04, 94.5, 23.31, 17.136)$,

$$r^0 = 10.$$

Solution point: $x^* = (4.5374, 10.889, 272.24, 42.198, 31.762)$,

$$f^* = 5.2803 \times 10^6.$$

Problem 12.27. (Ref. 13). Maximize

$$f(x) = a_0 + a_1x_1 + a_2x_1x_2 + a_3x_1x_3 + a_4x_1x_4 + a_5x_1x_5,$$

subject to

$$c_1(x) = x_1 \geq 0,$$

$$c_2(x) = x_2 - 1.2 \geq 0,$$

$$c_3(x) = x_3 - 20 \geq 0,$$

$$c_4(x) = x_4 - 9 \geq 0,$$

$$c_5(x) = x_5 - 6.5 \geq 0,$$

$$c_6(x) = 2.4 - x_2 \geq 0,$$

$$c_7(x) = 60 - x_3 \geq 0,$$

$$c_8(x) = 9.3 - x_4 \geq 0,$$

$$c_9(x) = 7.0 - x_5 \geq 0,$$

$$c_{10}(x) = a_6x_1 + a_7x_1x_2 + a_8x_1x_3 + a_9x_1x_4 + a_{10}x_1x_5 \geq 0,$$

$$c_{11}(x) = 294{,}000 - c_{10}(x) \geq 0,$$

$$c_{12}(x) = a_{11}x_1 + a_{12}x_1x_2 + a_{13}x_1x_3 + a_{14}x_1x_4 + a_{15}x_1x_5 \geq 0,$$

$$c_{13}(x) = 294{,}000 - c_{12}(x) \geq 0,$$

$$c_{14}(x) = a_{16}x_1 + a_{17}x_1x_2 + a_{18}x_1x_3 + a_{19}x_1x_4 + a_{20}x_1x_5 \geq 0,$$

$$c_{15}(x) = 277{,}200 - c_{14}(x) \geq 0,$$

where the a_i are given in Table 10 and

$$-1000 \leq x \leq 1000.$$

Starting point: $x^0 = (2.52, 2, 37.5, 9.25, 6.8)$,

$$B^0 = \{6, 7, 8, 9, 15\}.$$

Solution point: $x^* = (4.5375, 2.3999, 60, 9.2999, 6.9999)$,

$$f^* = 5.2802 \times 10^6.$$

References

1. BETTS, J. T., *An Improved Penalty Function Method for Solving Constrained Parameter Optimization Problems*, Journal of Optimization Theory and Applications, Vol. 16, Nos. 1/2, 1975.
2. POWELL, M. J. D., *A Method for Nonlinear Constraints in Minimization Problems*, Optimization, Edited by R. Fletcher, Academic Press, New York, New York, 1969.
3. HESTENES, M. R., *Multiplier and Gradient Methods*, Journal of Optimization Theory and Applications, Vol. 4, No. 5, 1969.
4. MIELE, A., CRAGG, E. E., IYER, R. R., and LEVY, A. V., *Use of the Augmented Penalty Function in Mathematical Programming Problems, Part 1*, Journal of Optimization Theory and Applications, Vol. 8, No. 2, 1971.
5. BETTS, J. T., *An Accelerated Multiplier Method for Nonlinear Programming*, The Aerospace Corporation, El Segundo, California, Report No. TR-0075(5901-03), 1974.
6. FLETCHER, R., *A General Quadratic Programming Algorithm*, Journal of the Institute of Mathematics Applications, Vol. 7, pp. 76–91, 1971.
7. FLETCHER, R., *A FORTRAN Subroutine for Quadratic Programming*, UKAEA Research Group, Harwell, England, Report No. AERE-R6370, 1970.
8. FLETCHER, R., *Minimizing General Functions Subject to Linear Constraints*, Numerical Methods for Nonlinear Optimization, Edited by F. A. Lootsma, Academic Press, New York, New York, 1972.

9. BETTS, J. T., *Solving the Nonlinear Least Square Problem: Application of a General Method*, The Aerospace Corporation, El Segundo, California, Report No. TR-0074(4901-03)-3, 1974.
10. HAARHOFF, P. C., and BUYS, J. D., *A New Method for the Optimization of a Nonlinear Function Subject to Nonlinear Constraints*, Computer Journal, Vol. 13, No. 2, 1970.
11. FIACCO, A. V., and MCCORMICK, G. P., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, New York, 1968.
12. COLVILLE, A. R., A Comparative Study on Nonlinear Programming Codes, IBM New York Scientific Center, Yorktown Heights, New York, Report No. 320-2949, 1968.
13. BOX, M. J., *A Comparison of Several Current Optimization Methods and the Use of Transformations in Constrained Problems*, Computer Journal, Vol. 9, pp. 67–77, 1966.
14. MIELE, A., CRAGG, E. E., and LEVY, A. V., *Use of the Augmented Penalty Function in Mathematical Programming Problems, Part 2*, Journal of Optimization Theory and Applications, Vol. 8, No. 2, 1971.
15. PICKETT, H. E., *A Contribution to the Thaumaturgy of Nonlinear Programming*, The Aerospace Corporation, San Bernardino, California, Report No. ATR-71(S9990)-1, 1970.
16. KUHN, H. W., and TUCKER, A. W., *Nonlinear Programming*, Proceedings of Second Berkeley Symposium, Edited by J. Neyman, University of California Press, Berkeley, California, 1951.
17. GOULD, F. J., *Nonlinear Tolerance Programming*, Numerical Methods for Nonlinear Optimization, Edited by F. A. Lootsma, Academic Press, New York, New York, 1972.