JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 8, No. 2, 1971

Use of the Augmented Penalty Function in Mathematical Programming Problems,[1] Part 1

A. MIELE,[2] E. E. CRAGG,[3] R. R. IYER,[4] AND A. V. LEVY[5]

Abstract. In this paper, the problem of minimizing a function $f(x)$ subject to a constraint $\varphi(x) = 0$ is considered. Here, f is a scalar, x an n-vector, and φ a q-vector, with $q < n$. The use of the augmented penalty function is explored in connection with the *ordinary gradient algorithm*. The augmented penalty function $W(x, \lambda, k)$ is defined to be the linear combination of the augmented function $F(x, \lambda)$ and the constraint error $P(x)$, where the q-vector λ is the Lagrange multiplier and the scalar k is the penalty constant.

The ordinary gradient algorithm is constructed in such a way that the following properties are satisfied in toto or in part: (a) descent property on the augmented penalty function, (b) descent property on the augmented function, (c) descent property on the constraint error, (d) constraint satisfaction on the average, or (e) individual constraint satisfaction. Properties (d) and (e) are employed to first order only.

With the above considerations in mind, two classes of algorithms are developed. For algorithms of Class I, the multiplier is determined so that the error in the optimum condition is minimized for given x; for algorithms of Class II, the multiplier is determined so that the constraint is satisfied to first order.

Algorithms of Class I have properties (a), (b), (c) and include

[1] Paper received October 2, 1970. This research was supported by the National Science Foundation, Grant No. GP-27271. The authors are indebted to Dr. J. N. Damoulakis for analytical and numerical assistance. Discussions with Professor H. Y. Huang are acknowledged.

[2] Professor of Astronautics and Director of Aero-Astronautics Group, Department of Mechanical and Aerospace Engineering and Materials Science, Rice University, Houston, Texas.

[3] Assistant Professor of Computer Science, University of Southwestern Louisiana, Lafayette, Louisiana.

[4] Postdoctoral Fellow, Department of Mechanical and Aerospace Engineering and Materials Science, Rice University, Houston, Texas.

[5] Graduate Student in Aero-Astronautics, Department of Mechanical and Aerospace Engineering and Materials Science, Rice University, Houston, Texas.

Algorithms (I-α) and (I-β). In the former algorithm, the penalty constant is held unchanged for all iterations; in the latter, the penalty constant is updated at each iteration so as to ensure satisfaction of property (d).

Algorithms of Class II have properties (a), (c), (e) and include Algorithms (II-α) and (II-β). In the former algorithm, the penalty constant is held unchanged for all iterations; in the latter, the penalty constant is updated at each iteration so as to ensure satisfaction of property (b).

Four numerical examples are presented. They show that algorithms of Class II exhibit faster convergence than algorithms of Class I and that algorithms of type (β) exhibit faster convergence than algorithms of type (α). Therefore, Algorithm (II-β) is the best among those analyzed. This is due to the fact that, in Algorithm (II-β), individual constraint satisfaction is enforced and that descent properties hold for the augmented penalty function, the augmented function, and the constraint error.

1. Introduction

Over the past several years, considerable work has been done on the solution of the constrained minimization problem, that is, the problem of minimizing a function $f(x)$ subject to a constraint $\varphi(x) = 0$. Here, f is a scalar x an n-vector, and φ a q-vector, with $q < n$.

Recently, Hestenes suggested the use of the augmented penalty function (Ref. 1)

$$W(x, \lambda, k) = f(x) + \lambda^T\varphi(x) + k\varphi^T(x)\,\varphi(x) \tag{1}$$

in order to solve the above problem. Here, the q-vector λ is the Lagrange multiplier and the scalar $k > 0$ is the penalty constant. Hestenes' algorithm, termed method of multipliers, is made of several cycles. In each cycle, the multiplier and the penalty constant are held unchanged, while the vector x is viewed as unconstrained. After convergence is obtained for given values of λ and k, the present cycle is terminated. Then, the next cycle is started, with the multiplier and the penalty constant being updated according to the following simple rules:

$$\lambda_* = \lambda + 2k\varphi(x), \qquad k_* = k. \tag{2}$$

Hence, the penalty constant is held unchanged throughout the algorithm.

To start the algorithm, some assumption concerning the multiplier is necessary. The simplest assumption is

$$\lambda = 0 \tag{3}$$

and is equivalent to stating that the augmented penalty function (1) and the penalty function

$$U(x, k) = f(x) + k\varphi^T(x)\,\varphi(x) \tag{4}$$

are identical for the first cycle of the algorithm.

It is the feeling of these authors that Hestenes' method of multipliers exhibits better convergence characteristics than the classical penalty function method (see, for example, Refs. 2–5). However, it is also felt that Hestenes' method can be improved if the multiplier and the penalty constant are chosen at each iteration so as to obtain certain desirable properties, for instance, (a) descent property on the augmented penalty function, (b) descent property on the augmented function, (c) descent property on the constraint error, and either (d) constraint satisfaction on the average or (e) individual constraint satisfaction.

In this paper, the use of the augmented penalty function is explored in connection with the ordinary gradient algorithm so as to ensure satisfaction of properties (a) through (e), in toto or in part. In a subsequent paper (Ref. 6), the use of the augmented penalty function is explored in connection with the conjugate gradient algorithm.

2. Statement of the Problem

We consider the problem of minimizing the function

$$f = f(x) \tag{5}$$

subject to the constraint

$$\varphi(x) = 0. \tag{6}$$

In the above equations, f is a scalar, x an n-vector, and φ a q-vector,[6] where $q < n$. It is assumed that the first and second partial derivatives of the functions f and φ with respect to x exist and are continuous; it is also assumed that the constrained minimum exists.

2.1. Exact First-Order Conditions. From theory of maxima and minima, it is known that the previous problem can be recast as that of minimizing the *augmented function*

$$F(x, \lambda) = f(x) + \lambda^T\varphi(x) \tag{7}$$

[6] All vectors are column vectors.

subject to the constraint (6). Here, λ is a q-vector Lagrange multiplier, and the superscript T denotes the transpose of a matrix. If

$$F_x(x, \lambda) = f_x(x) + \varphi_x(x)\lambda \tag{8}$$

denotes the gradient of the augmented function,[7] the optimum solution x, λ must satisfy the simultaneous equations

$$\varphi(x) = 0, \qquad F_x(x, \lambda) = 0. \tag{9}$$

2.2. Approximate Solutions. In general, the system (9) is nonlinear; consequently, approximate methods must be employed. These are of two kinds: first-order methods (such as those discussed in subsequent sections of this report) and second-order methods. Here, we introduce the scalar quantities

$$P(x) = \varphi^T(x)\,\varphi(x), \qquad Q(x, \lambda) = F_x^T(x, \lambda)\,F_x(x, \lambda), \tag{10}$$

which measure the errors in the constraint and the optimum condition, respectively. We observe that $P = 0$ and $Q = 0$ for the optimum solution, while $P > 0$ and/or $Q > 0$ for any approximation to the solution. When approximate methods are used, they must ultimately lead to values of x, λ such that

$$P(x) \leqslant \epsilon_1, \qquad Q(x, \lambda) \leqslant \epsilon_2. \tag{11}$$

Alternatively, (11) can be replaced by

$$R(x, \lambda) \leqslant \epsilon_3, \tag{12}$$

where

$$R(x, \lambda) = P(x) + Q(x, \lambda) \tag{13}$$

denotes the cumulative error in the constraint and the optimum condition. Here, ϵ_1, ϵ_2, ϵ_3 are small, preselected numbers. Note that satisfaction of Ineq. (12) implies satisfaction of Ineqs. (11), if one chooses $\epsilon_1 = \epsilon_2 = \epsilon_3$.

3. Ordinary Gradient Algorithm

In this section, we construct an ordinary gradient algorithm based on the consideration of the *augmented penalty function*

$$W(x, \lambda, k) = F(x, \lambda) + kP(x), \tag{14}$$

[7] In Eq. (8), the gradients f_x and F_x denote n-vectors, and the matrix φ_x is $n \times q$.

where

$$F(x, \lambda) = f(x) + \lambda^T\varphi(x), \qquad P(x) = \varphi^T(x)\,\varphi(x). \tag{15}$$

In this algorithm, the Lagrange multiplier and the penalty constant are determined so as to ensure satisfaction of the following properties, in toto or in part: (a) descent property on the augmented penalty function, (b) descent property on the augmented function, (c) descent property on the constraint error, and either (d) constraint satisfaction on the average or (e) individual constraint satisfaction. Properties (d) and (e) are employed to first order only.

3.1. Basic Algorithm. Let x denote the nominal point, $\tilde{x}$ the varied point, and Δx the displacement leading from the nominal point to the varied point. Let λ denote the Lagrange multiplier, k the penalty constant, and α the gradient stepsize. With this understanding, we consider the ordinary gradient algorithm represented by

$$\begin{aligned} F_x(x, \lambda) &= f_x(x) + \varphi_x(x)\lambda, \\ P_x(x) &= 2\varphi_x(x)\,\varphi(x), \\ W_x(x, \lambda, k) &= F_x(x, \lambda) + kP_x(x), \\ \Delta x &= -\alpha W_x(x, \lambda, k), \\ \tilde{x} &= x + \Delta x. \end{aligned} \tag{16}$$

For given nominal point x, Lagrange multiplier λ, and penalty constant k, Eqs. (16) constitute a complete iteration leading to the varied point $\tilde{x}$ providing one specifies the gradient stepsize α.

3.2. Basic Descent Property. When the displacement (16-4) is employed, the first variation of the augmented penalty function is given by[8]

$$\delta W(x, \lambda, k) = W_x^T(x, \lambda, k)\,\Delta x = -\alpha W_x^T(x, \lambda, k)\,W_x(x, \lambda, k) \tag{17}$$

and is negative for $\alpha > 0$. Therefore, if α is sufficiently small, the augmented penalty function decreases during any iteration.

3.3. Gradient Stepsize. The descent property established in the previous section is instrumental in the determination of the optimum gradient

[8] In the computation of the first variation of $W(x, \lambda, k)$, the Lagrange multiplier λ and the penalty constant k are held unchanged.

stepsize for given nominal point x, Lagrange multiplier λ, and penalty constant k. If Eqs. (16-4) and (16-5) are combined, the position vector at the end of the step becomes

$$\tilde{x} = x - \alpha W_x(x, \lambda, k). \tag{18}$$

This is a one-parameter family of varied points $\tilde{x}$, for which the augmented penalty function (14) takes the form

$$W(\tilde{x}, \lambda, k) = W[x - \alpha W_x(x, \lambda, k), \lambda, k] = W(\alpha). \tag{19}$$

Along the straight line defined by Eq. (18), the function (19) admits the derivative

$$W_\alpha(\alpha) = -W_x^T(\tilde{x}, \lambda, k)\, W_x(x, \lambda, k) \tag{20}$$

which, at $\alpha = 0$, yields

$$W_\alpha(0) = -W_x^T(x, \lambda, k)\, W_x(x, \lambda, k), \tag{21}$$

a result consistent with (17).

Precise Search. Having established that $W_\alpha(0) < 0$, we now assume that a minimum of $W(\alpha)$ exists. Then, we employ some one-dimensional search scheme (for instance, quadratic interpolation, cubic interpolation, or quasi-linearization) to determine the value of α for which

$$W_\alpha(\alpha) = 0. \tag{22}$$

Ideally, this procedure should be used iteratively until the modulus of the slope satisfies any of the following inequalities:

$$| W_\alpha(\alpha)| \leqslant \epsilon_4 \qquad \text{or} \qquad | W_\alpha(\alpha)| \leqslant \epsilon_5 \,| W_\alpha(0)|, \tag{23}$$

where ϵ_4 and ϵ_5 are small, preselected numbers. Of course, the value of α satisfying Ineq. (23) must be such that

$$W(\alpha) < W(0). \tag{24}$$

Approximate Search. In practice, the rigorous determination of α might require excessive computing time. Therefore, one might renounce solving Eq. (22) with a particular degree of precision and determine the gradient stepsize in a noniterative fashion, for instance, by employing the first optimum value of α supplied by the search procedure. Of course, this value of α is acceptable only if Ineq. (24) is satisfied. Otherwise, α must be replaced by a smaller value (for example, with a bisection process) until Ineq. (24) is met. This is guaranteed by the descent property (17).

3.4. Remark. When the displacement (16-4) is employed, the first variations of the function $F(x, \lambda)$ and $P(x)$ are given by[9]

$$\begin{aligned} \delta F(x, \lambda) &= F_x^T(x, \lambda)\,\Delta x = -\alpha F_x^T(x, \lambda)\, W_x(x, \lambda, k), \\ \delta P(x) &= P_x^T(x)\,\Delta x = -\alpha P_x^T(x)\, W_x(x, \lambda, k), \end{aligned} \tag{25}$$

and, in the light of (16-3), can be rewritten as

$$\begin{aligned} \delta F(x, \lambda) &= -\alpha F_x^T(x, \lambda)[F_x(x, \lambda) + kP_x(x)], \\ \delta P(x) &= -\alpha P_x^T(x)[F_x(x, \lambda) + kP_x(x)]. \end{aligned} \tag{26}$$

These relations are employed in the following sections in order to construct particular algorithms meeting properties (a) through (e) of the introduction, in toto or in part.

4. Algorithms of Class I

In this section, we present two algorithms having one common feature: the Lagrange multiplier λ is determined as in Ref. 7, that is, by minimizing the error in the optimum condition (10-2) with respect to λ for given x. Owing to the fact that

$$Q(x, \lambda) = [f_x(x) + \varphi_x(x)\lambda]^T[f_x(x) + \varphi_x(x)\lambda], \tag{27}$$

the Lagrange multiplier is determined by the relation

$$Q_\lambda(x, \lambda) = 0, \tag{28}$$

which implies that

$$\varphi_x^T(x)\, \varphi_x(x)\lambda + \varphi_x^T(x) f_x(x) = 0. \tag{29}$$

This linear vector equation is equivalent to q linear scalar relations, in which the only unknown is the Lagrange multiplier.

[9] In the computation of the first variation of $F(x, \lambda)$, the multiplier λ is held unchanged.

Descent Properties. Premultiplication of (29) by $2\varphi^T(x)$ yields the following relation:

$$P_x^T(x) F_x(x, \lambda) = 0, \tag{30}$$

meaning that the gradients of the augmented function and the constraint error are orthogonal to one another. Therefore, Eqs. (26) become

$$\delta F(x, \lambda) = -\alpha F_x^T(x, \lambda) F_x(x, \lambda), \qquad \delta P(x) = -\alpha k P_x^T(x) P_x(x), \tag{31}$$

which show that the first variations of the augmented function and the constraint error are negative for $\alpha > 0$. Therefore, if α is sufficiently small, the augmented function and the constraint error decrease during any iteration.

Algorithm (I-α). This algorithm is represented by Eqs. (16), with λ determined by Eq. (29) and k held at a preselected value; it exhibits descent properties on the functions $W(x, \lambda, k)$, $F(x, \lambda)$, and $P(x)$, regardless of the choice of the penalty constant. Because of the descent property on $P(x)$, the displacement Δx leads toward the constraint. It can be anticipated that, if the penalty constant is either very small or very large, Algorithm (I-α) may require a large number of iterations for convergence to the desired solution; therefore, k must be in a proper range.

Algorithm (I-β). To circumvent the difficulties of Algorithm (I-α), the following procedure can be adopted: instead of choosing the penalty constant arbitrarily, we select it in such a way that, on the average, the constraints are satisfied to first order. Thereby, we determine k from the relation

$$\delta P(x) + 2\mu P(x) = 0, \tag{32}$$

where

$$\mu = C\alpha \tag{33}$$

is the restoration stepsize and C a constant to be specified. If the function $f(x)$, the constraint $\varphi(x)$, and the vector x are scaled in such a way that the optimum gradient stepsize α is $O(1)$, then the choice $C = 1$ is appropriate (see Refs. 8–9). From (31-2), (32), (33), we deduce that

$$k = 2CP(x)/P_x^T(x) P_x(x). \tag{34}$$

In conclusion, Algorithm (I-β) is represented by Eqs. (16), with λ determined by Eq. (29) and k determined by Eq. (34). In this algorithm, k must be updated at each iteration, while this is not the case with Algorithm (I-α).

5. Algorithms of Class II

In this section, we present two algorithms having one common feature: the Lagrange multiplier λ is determined as in Refs. 8–9, that is, by satisfying the constraint equation (6) to first order. Thereby, we determine λ from the relation

$$\delta\varphi(x) + \mu\varphi(x) = 0, \tag{35}$$

where

$$\mu = C\alpha \tag{36}$$

is the restoration stepsize and C a constant to be specified. If the function $f(x)$, the constraint $\varphi(x)$, and the vector x are scaled in such a way that the optimum gradient stepsize α is $O(1)$, then the choice $C = 1$ is appropriate (see Refs. 8–9). When the displacement (16-4) is employed, the first variation of the constraint error is given by

$$\delta\varphi(x) = \varphi_x^T(x)\,\Delta x = -\alpha\varphi_x^T(x)\,W_x(x, \lambda, k), \tag{37}$$

that is,

$$\delta\varphi(x) = -\alpha\varphi_x^T(x)[F_x(x, \lambda) + kP_x(x)]. \tag{38}$$

From (16-1), (35), (36), (38), we see that the Lagrange multiplier satisfies the relation

$$\varphi_x^T(x)\,\varphi_x(x)\lambda + \varphi_x^T(x)[f_x(x) + kP_x(x)] - C\varphi(x) = 0. \tag{39}$$

This linear vector equation is equivalent to q linear scalar relations, in which the only unknown is the Lagrange multiplier.

Descent Properties. Premultiplication of (39) by $2\varphi^T(x)$ yields the following relation:

$$P_x^T(x)\,F_x(x, \lambda) + kP_x^T(x)\,P_x(x) - 2CP(x) = 0, \tag{40}$$

so that Eqs. (26) become

$$\begin{gathered} \delta F(x, \lambda) = -\alpha F_x^T(x, \lambda)\,F_x(x, \lambda) - k\alpha[2CP(x) - kP_x^T(x)\,P_x(x)], \\ \delta P(x) = -2C\alpha P(x). \end{gathered} \tag{41}$$

Equation (41-2) shows that the first variation of the constraint error is negative for $\alpha > 0$. Therefore, if α is sufficiently small, the constraint error decreases during any iteration.

Algorithm (II-α). This algorithm is represented by Eqs. (16), with λ determined by Eq. (39), and k held at a preselected value; it exhibits descent properties of the functions $W(x, \lambda, k)$ and $P(x)$, regardless of the choice of the penalty constant.

Algorithm (II-β). This algorithm is represented by Eqs. (16), with λ determined by Eq. (39) and the penalty constant given by

$$k = 2CP(x)/P_x^T(x)\, P_x(x). \tag{42}$$

Consequently, Eq. (40) reduces to

$$P_x^T(x)\, F_x(x, \lambda) = 0, \tag{43}$$

meaning that the gradients of the augmented function and the constraint error are orthogonal to one another. In turn, Eq. (41-1) becomes

$$\delta F(x, \lambda) = -\alpha F_x^T(x, \lambda)\, F_x(x, \lambda), \tag{44}$$

which shows that the first variation of the augmented function is negative for $\alpha > 0$. Therefore, if α is sufficiently small, the augmented function decreases during any iteration. In conclusion, Algorithm (II-β) has descent properties on the functions $W(x, \lambda, k)$, $F(x, \lambda)$, and $P(x)$.

5.1. Remark. The problem of minimizing the function (5) subject to the constraint (6) can be reformulated as that of minimizing the penalty function (4) subject to the constraint (6). With this understanding, the combined gradient-restoration algorithm of Refs. 8–9 yields Algorithm (II-α), providing the function $f(x)$ is replaced by the penalty function $U(x, k)$, and providing the augmented function $F(x, \lambda)$ is replaced by the augmented penalty function $W(x, \lambda, k)$.

6. Summary of Algorithms

The previous algorithms are represented by the relations

$$\begin{aligned}
F_x(x, \lambda) &= f_x(x) + \varphi_x(x)\lambda, \\
P_x(x) &= 2\varphi_x(x)\, \varphi(x), \\
W_x(x, \lambda, k) &= F_x(x, \lambda) + kP_x(x), \\
\Delta x &= -\alpha W_x(x, \lambda, k), \\
\tilde{x} &= x + \Delta x,
\end{aligned} \tag{45}$$

in which the Lagrange multiplier and the penalty constant are determined as follows:

Algorithm (I-α)

$$\varphi_x^T(x)\,\varphi_x(x)\lambda + \varphi_x^T(x) f_x(x) = 0, \qquad k = \text{const}, \tag{46}$$

Algorithm (I-β)

$$\varphi_x^T(x)\,\varphi_x(x)\lambda + \varphi_x^T(x) f_x(x) = 0, \qquad k = 2CP(x)/P_x^T(x)\,P_x(x), \tag{47}$$

Algorithm (II-α)

$$\varphi_x^T(x)\,\varphi_x(x)\lambda + \varphi_x^T(x)[f_x(x) + kP_x(x)] - C\varphi(x) = 0, \qquad k = \text{const}, \tag{48}$$

Algorithm (II-β)

$$\varphi_x^T(x)\,\varphi_x(x)\lambda + \varphi_x^T(x)[f_x(x) + kP_x(x)] - C\varphi(x) = 0, \qquad k = 2CP(x)/P_x^T(x)\,P_x(x). \tag{49}$$

In all of the algorithms, the optimum gradient stepsize α is determined by a one-dimensional search on the augmented penalty function

$$W(x, \lambda, k) = F(x, \lambda) + kP(x), \tag{50}$$

where

$$F(x, \lambda) = f(x) + \lambda^T\varphi(x), \qquad P(x) = \varphi^T(x)\,\varphi(x), \tag{51}$$

with the Lagrange multiplier and the penalty constant held unchanged during the search. Concerning the constant C, the choice $C = 1$ is appropriate if the function $f(x)$, the constraint $\varphi(x)$, and the vector x are scaled in such a way that the optimum gradient stepsize α is $O(1)$.

6.1. Remark. If $q = 1$, that is, if there is only one scalar constraint, Eqs. (47) and (49) reduce to

$$\varphi_x^T(x)\,\varphi_x(x)\lambda + \varphi_x^T(x) f_x(x) = 0, \qquad k = \tfrac{1}{2}C/[\varphi_x^T(x)\,\varphi_x(x)], \tag{52}$$

where φ_x is $n \times 1$. Therefore, in this special case, Algorithms (I-β) and (II-β) are identical.

7. Experimental Conditions and Numerical Examples

In order to illustrate the theory, four numerical examples were developed using a Burroughs B-5500 computer and double-precision arithmetic. The algorithms were programmed in Extended ALGOL. The constant C was specified to be $C = 1$, thereby weighting evenly the gradient component and the restoration component of the displacement Δx. In each iteration, one-step quasilinearization was employed on the search function $W(\alpha)$ followed by a bisection process until the inequality

$$W(\alpha) < W(0) \tag{53}$$

was satisfied. *Convergence* was defined as follows:

$$R(x, \lambda) \leqslant 10^{-12}, \tag{54}$$

and the number of iterations at convergence N_* was recorded. Conversely, *nonconvergence* was defined by means of the inequalities

$$\text{(a) } N \geqslant 1000, \qquad \text{or} \quad \text{(b) } N_s \geqslant 20. \tag{55}$$

Here, N is the iteration number, and N_s is the number of bisections of the gradient stepsize α required to satisfy Ineq. (53); these bisections are started from the first optimum value of α supplied by quasilinearization on the search function $W(\alpha)$.

Example 7.1. Consider the problem of minimizing the function

$$f = (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2 \tag{56}$$

subject to the constraints

$$x_1 + 3x_2 = 0, \qquad x_3 + x_4 - 2x_5 = 0, \qquad x_2 - x_5 = 0. \tag{57}$$

This function admits the relative minimum $f = 4.0930$ at the point defined by

$$x_1 = -0.7674, \quad x_2 = 0.2558, \quad x_3 = 0.6279, \quad x_4 = -0.1162, \quad x_5 = 0.2558, \tag{58}$$

and

$$\lambda_1 = 2.0465, \qquad \lambda_2 = 2.2325, \qquad \lambda_3 = -5.9534. \tag{59}$$

The nominal point chosen to start the algorithm is the point of coordinates

$$x_1 = x_2 = x_3 = x_4 = x_5 = 2, \tag{60}$$

not consistent with (57).

Example 7.2. Consider the problem of minimizing the function

$$f = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^4 \tag{61}$$

subject to the constraint

$$x_1(1 + x_2^2) + x_3^4 - 4 - 3\sqrt{2} = 0. \tag{62}$$

This function admits the relative minimum $f = 0.3256 \times 10^{-1}$ at the point defined by

$$x_1 = 1.1048, \qquad x_2 = 1.1966, \qquad x_3 = 1.5352, \tag{63}$$

and

$$\lambda_1 = -0.1072 \times 10^{-1}. \tag{64}$$

The nominal point chosen to start the algorithm is the point of coordinates

$$x_1 = x_2 = x_3 = 2, \tag{65}$$

not consistent with (62).

Example 7.3. Consider the problem of minimizing the function

$$f = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6 \tag{66}$$

subject to the constraints

$$x_4 x_1^2 + \sin(x_4 - x_5) - 2\sqrt{2} = 0, \qquad x_2 + x_3^4 x_4^2 - 8 - \sqrt{2} = 0. \tag{67}$$

This function admits the relative minimum $f = 0.2415$ at the point defined by

$$x_1 = 1.1661, \quad x_2 = 1.1821, \quad x_3 = 1.3802, \quad x_4 = 1.5060, \quad x_5 = 0.6109, \tag{68}$$

and

$$\lambda_1 = -0.8553 \times 10^{-1}, \qquad \lambda_2 = -0.3187 \times 10^{-1}. \tag{69}$$

The nominal point chosen to start the algorithm is the point of coordinates

$$x_1 = x_2 = x_3 = x_4 = x_5 = 2, \tag{70}$$

not consistent with (67).

Example 7.4. Consider the problem of minimizing the function

$$f = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^4 + (x_4 - x_5)^4 \tag{71}$$

subject to the constraints

$$x_1 + x_2^2 + x_3^3 - 2 - 3\sqrt{2} = 0, \quad x_2 - x_3^2 + x_4 + 2 - 2\sqrt{2} = 0, \quad x_1 x_5 - 2 = 0. \tag{72}$$

This function admits the relative minimum $f = 0.7877 \times 10^{-1}$ at the point defined by

$$x_1 = 1.1911, \quad x_2 = 1.3626, \quad x_3 = 1.4728, \quad x_4 = 1.6350, \quad x_5 = 1.6790, \tag{73}$$

and

$$\lambda_1 = -0.3882 \times 10^{-1}, \quad \lambda_2 = -0.1672 \times 10^{-1}, \quad \lambda_3 = -0.2879 \times 10^{-3}. \tag{74}$$

The nominal point chosen to start the algorithm is the point of coordinates

$$x_1 = x_2 = x_3 = x_4 = x_5 = 2, \tag{75}$$

not consistent with (72).

8. Numerical Results and Conclusions

For the previous examples and experimental conditions, algorithms of Class I and Class II were tested in versions (α) and (β): in the former, the penalty constant is held unchanged for all iterations; in the latter, the penalty constant is updated at each iteration. The numerical results are given in Tables 1–4.

Tables 1 and 3 refer to Algorithms (I-α) and (II-α) and show the number of iterations at convergence N_* for several values of the penalty constant k. Since N_* exhibits a relative minimum with respect to k, the proper choice of the penalty constant is essential for rapid convergence.

Tables 2 and 4 refer to Algorithms (I-β) and (II-β). Clearly, the number of iterations at convergence for Algorithm (I-β) is close to the minimum with respect to k of the number of iterations at convergence for Algorithm (I-α); analogously, the number of iterations at convergence for Algorithm (II-β) is close to the minimum with respect to k of the number of iterations at convergence for Algorithm (II-α).

From the tables, it appears that algorithms of Class II exhibit faster convergence than algorithms of Class I, and algorithms of type (β) exhibit faster convergence than algorithms of type (α). Therefore, Algorithm (II-β) is the best among those analyzed: this is due to the fact that individual constraint satisfaction is enforced and that descent properties hold for the augmented penalty function, the augmented function, and the constraint error.

Table 1. Number of iterations at convergence N_* for Algorithm (I-α).

	Example			
k	7.1	7.2	7.3	7.4
10^{-4}	(a)	263	(a)	(a)
10^{-3}	(a)	37	721	870
10^{-2}	(a)	23	199	97
10^{-1}	408	49	321	35
10^{0}	176	779	(a)	138
10^{1}	126	(a)	(a)	(a)
10^{2}	(a)	(a)	(a)	(a)

Table 2. Number of iterations at convergence N_* for Algorithm (I-β).

	Example			
k	7.1	7.2	7.3	7.4
Eq. (47-2)	111	22	304	57

Table 3. Number of iterations at convergence N_* for Algorithm (II-α).

	Example			
k	7.1	7.2	7.3	7.4
10^{-4}	24	36	27	18
10^{-3}	24	31	36	17
10^{-2}	24	51	99	16
10^{-1}	25	548	758	39
10^{0}	103	(a)	(a)	231
10^{1}	(a)	(a)	(a)	(a)
10^{2}	(a)	(a)	(a)	(a)

Table 4. Number of iterations at convergence N_* for Algorithm (II-β).

	Example			
k	7.1	7.2	7.3	7.4
Eq. (49-2)	23	22	41	18

References

1. HESTENES, M. R., *Multiplier and Gradient Methods*, Journal of Optimization Theory and Applications, Vol. 4, No. 5, 1969.
2. KELLEY, H. J., *Methods of Gradients*, Optimization Techniques, Edited by G. Leitmann, Academic Press, New York.
3. BRYSON, A. E., JR., and HO, Y. C., *Applied Optimal Control*, Blaisdell Publishing Company, Waltham, Massachusetts, 1969.
4. FIACCO, A. V., and MCCORMICK, G. P., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, 1968.
5. DAVIES, D., and SWANN, W. H., *Review of Constrained Optimization*, Optimization, Edited by R. Fletcher, Academic Press, New York, 1969.
6. 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.
7. MIELE, A., and LEVY, A. V., *Modified Quasilinearization and Optimal Initial Choice of the Multipliers, Part 1, Mathematical Programming Problems*, Journal of Optimization Theory and Applications, Vol. 6, No. 5, 1970.
8. MIELE, A., HEIDEMAN, J. C., and LEVY, A. V., *Mathematical Programming for Constrained Minimal Problems, Part 3, Combined Gradient-Restoration Algorithm*, Rice University, Aero-Astronautics Report No. 68, 1970.
9. HEIDEMAN, J. C., and LEVY, A. V., *Mathematical Programming for Constrained Minimal Problems, Part 4, Combined Gradient-Restoration Algorithm: Further Numerical Examples*, Rice University, Aero-Astronautics Report No. 69, 1970.