JOURNAL OF APPROXIMATION THEORY 41, 217–243 (1984)

Subgradient Projection Algorithm, II

V. P. SREEDHARAN

Department of Mathematics, Michigan State University, East Lansing, Michigan 48824, U.S.A.

Communicated by E. W. Cheney

Received November 8, 1982; revised July 22, 1983

Using only easily computable portions of certain ε-subdifferentials an implementable convergent algorithm for finding the minimizer of a non-differentiable convex program is given. At each iteration cycle certain projections are computed and corrected suitably. The negatives of these directions are feasible directions of strict descent for the objective function. The convergence of the algorithm is proved.

1. INTRODUCTION

This paper presents an implementable algorithm for the minimization of a certain type of non-differentiable convex function subject to a finite collection of differentiable convex constraints. The algorithm below is obtained by modifying and extending the subgradient projection algorithm we gave in [11]. All the introductory remarks in [11] apply to this paper as well. In a certain sense, the work in this paper is the subgradient counterpart to Rosen's [9] "Part II: Non-linear constraints" paper. The algorithm proposed here avoids the possibility of "jamming," a situation where the generated sequence clusters or even converges to non-optimal points. For the original gradient projection [9] this possibility is not excluded. The algorithms of Wolfe [12] and Lemarechal [2] generalize classical methods of unconstrained optimization in the differentiable case to the corresponding non-differentiable case by replacing the gradient with an appropriately chosen subgradient. This paper accomplishes the analogous task for the constrained case with the attendant complications. Our algorithm also generalizes the work of Rosen [9] and Polak [4] and is an extension of the algorithm in [11]. In implementing the algorithm we will have to compute only certain portions of the ε-subdifferentials. This is easily accomplished here, in contrast to some algorithms in the literature, where the complete ε-subdifferential is called for. The complete ε-subdifferential uses non-local information and in general it is a prohibitive task to compute it. The proof that the algorithm converges is somewhat involved and is given in Sections 5

0021-9045/84 $3.00

and 6. The computational details and experience with the algorithm is reported in paper [3]. The computational details in [10] apply directly to our earlier paper [11], but many details in [10] definitely have implications to this paper as well.

2. PROBLEM

We consider the following problem. Let $\Omega \subset \mathbb{R}^d$ be a nonempty open convex subset and f, g_i, $v_j: \Omega \to \mathbb{R}$, $i = 1,..., m$; $j = 1,..., r$ all be convex differentiable functions on Ω. Let

$$X = \{x \in \Omega \mid g_i(x) \leqslant 0, i = 1,..., m\}$$

be bounded and assume that Slater's constraint qualification (SQ) is satisfied:

$$\text{There exists some } a \in X \text{ such that } g_i(a) < 0, \qquad i = 1,..., m. \tag{SQ}$$

Let f be strictly convex also, i.e.,

$$2f((x+y)/2) < f(x) + f(y), \qquad x, y \in X, \quad x \neq y.$$

Let

$$v(x) = \max\{v_j(x) \mid 1 \leqslant j \leqslant r\}.$$

Our problem is to minimize $f(x) + v(x)$ subject to the constraint $x \in X$. We denote this problem by (P). More explicitly,

$$\begin{aligned} &g_i(x) \leqslant 0, \qquad i = 1,..., m, \\ &f(x) + v(x) \qquad (\min). \end{aligned} \tag{P}$$

Note that f, g_i, v_j are all continuously differentiable because they are convex and differentiable on open Ω.

3. NOTATION

Let $x \in X$ and $\varepsilon \geqslant 0$. We define the sets of indices $I_\varepsilon(x)$ and $J_\varepsilon(x)$ by

$$I_\varepsilon(x) = \{1 \leqslant i \leqslant m \mid g_i(x) \geqslant -\varepsilon\}, \tag{3.1}$$

$$J_\varepsilon(x) = \{1 \leqslant j \leqslant r \mid v_j(x) \geqslant v(x) - \varepsilon\}. \tag{3.2}$$

Naturally,

$$I_0(x) = \{1 \leqslant i \leqslant m \mid g_i(x) = 0\}, \tag{3.3}$$

and

$$J_0(x) = \{1 \leqslant j \leqslant r \mid v_j(x) = v(x)\}. \tag{3.4}$$

Using these index sets we define the following convex subsets

$$C_\varepsilon(x) = \text{cone}\{\nabla g_i(x) \mid i \in I_\varepsilon(x)\}, \tag{3.5}$$

and

$$K_\varepsilon(x) = \text{conv}\{\nabla v_j(x) \mid j \in J_\varepsilon(x)\}. \tag{3.6}$$

Here and elsewhere we denote by cone S the convex cone generated by S with apex at 0, and by conv S the convex hull of the set S. Note that when S is empty, cone $S = \{0\}$, whereas conv S is empty.

For any non-empty closed convex subset $S \subset \mathbb{R}^d$ there is a unique point $a \in S$ nearest to the origin, which we denote by $N[S]$. The point $a = N[S]$ is characterized by the inequality

$$ax \geqslant |a|^2 \qquad \text{for all} \quad x \in S. \tag{3.7}$$

Here and henceforth the standard Euclidean inner product of two vectors in $\mathbb{R}^d$ is denoted simply by juxtaposing the vectors. The corresponding Euclidean length is denoted by $|\cdot|$.

4. Algorithm

In this section we present a subgradient projection algorithm for solving problem (P). We start by doing a simple unconstrained minimization of the C^1 function f. Then we carry out the iterative scheme of the main algorithm, the subgradient projection algorithm.

4.1. Algorithm.

Step 1. Do an unconstrained minimization of the function f, say, using a method of conjugate gradient descents. If no minimizer exists in Ω GO TO Step 3. If minimizer c exists, check whether $c \in X$. If $c \notin X$, GO TO Step 3. If $c \in X$, proceed.

Step 2. Compute $\nabla v_j(c)$, $j = 1, \ldots, r$. If $\nabla v_j(c) = 0$ for every j, STOP; c is the unique minimizer of problem (P). If $\nabla v_j(c) \neq 0$ for some j, proceed.

Step 3. Start with arbitrary $x_0 \in X$ and $k = 0$. Let $\varepsilon_0 > 0$ be such that $\varepsilon_0 < -\max_{1 \leqslant i \leqslant m} g_i(a)$. Set $\varepsilon = \varepsilon_0$. (Recall that a is known, a priori, in the problem.)

Step 4. Compute $y_0 = N[\nabla f(x_k) + K_0(x_k) + C_0(x_k)]$. If $y_0 = 0$, STOP; x_k is the solution of problem (P). If $y_0 \neq 0$, proceed.

Step 5. Compute $y_\varepsilon = N[\nabla f(x_k) + K_\varepsilon(x_k) + C_\varepsilon(x_k)]$.

Step 6. If $|y_\varepsilon|^2 > \varepsilon$, set $\varepsilon_k = \varepsilon$, $s_k = y_\varepsilon$ and GO TO Step 8.

Step 7. Replace ε by $\varepsilon/2$ and GO TO Step 5.

Step 8. Let $I = I_{\varepsilon_k}(x_k)$. If I is empty, let $u_k = 0$ and $M_k = 0$. If I is non-empty, let $\gamma_{ij} = \nabla g_i(x_k) \nabla g_j(x_k)$, $i, j \in I$. Solve the linear program

$$\sum_i \gamma_{ij} \mu_i \geqslant |\nabla g_j(x_k)|, \qquad j \in I,$$

$$\mu_i \geqslant 0, \qquad i \in I,$$

$$\sum_{i \in I} \mu_i \qquad (\min).$$

It is shown later that this linear program has a minimal solution $(\bar{\mu}_i)$. Let

$$u_k = \sum_{i \in I} \bar{\mu}_i \nabla g_i(x_k).$$

Set

$$M_k = \{|\nabla f(x_k)| + \max_{1 \leqslant j \leqslant r} |\nabla g_j(x_k)|\} \, |u_k|.$$

Step 9. Let $\lambda_k = |s_k|^2/(2M_k + 1)$ and $t_k = s_k + \lambda_k u_k$.

Step 10. Find $\bar{\alpha}_k$, where

$$\bar{\alpha}_k = \max\{\alpha \mid x_k - \alpha t_k \in X,\ g_i(x_k - \alpha t_k) \leqslant g_i(x_k),\ \forall i \in I\}.$$

It will be shown that $\bar{\alpha}_k > 0$.

Step 11. Find $\alpha_k \in [0, \bar{\alpha}_k]$ such that there exists

$$z_k \in \nabla f(x_k - \alpha_k t_k) + K_0(x_k - \alpha_k t_k).$$

with

$$z_k t_k = 0.$$

If no such z_k exists, set $\alpha_k = \bar{\alpha}_k$.

Step 12. Define $x_{k+1} = x_k - \alpha_k t_k$. Increment k by 1 and GO TO Step 4.

4.2. Note that any method of finding the critical points of smooth convex f, the zeros of the equation $\nabla f(x) = 0$, may be substituted for Step 1 above. This would be particularly useful when f has a nice analytic expression. The unconstrained minimization is done at the start to exclude a very special easy case of problem (P). (See Lemmas 5.10 and 6.6.)

4.3. In practice to improve the convergence of the algorithm one may wish to reset $\varepsilon = \varepsilon_0$ in Step 4 during the initial iteration cycles of the algorithm. This should avoid the possibility of taking small steps when one is not "near" the optimal solution. After these iterations we revert back to the algorithm as given above with a minor change. We set $\varepsilon = \varepsilon_0$ in Step 3, but instead of using an arbitrary $x_0 \in X$ to start the algorithm, we take x_0 to be the last available x_k. These changes do not affect our convergence proof, though, strictly speaking, in Sections 5 and 6 we will have to say that (ε_k) is eventually non-increasing, in place of (ε_k) is non-increasing.

4.4. Steps 4 and 5 can be implemented as special quadratic programs, as was done in Rubin [10]. Step 11 requires a properly constructed line search, some comparisons and univariate minimizations. See Rubin [10] and Owens [3]. In practice, the statement, "if $y_0 = 0$, STOP" in Step 4 will be replaced by "STOP, if $|y_0| \leqslant \eta$," $\eta > 0$, a stopping rule parameter. Also obvious practical modifications for stopping in Step 2 will have to be included in a computer program. For computational details and experience we refer to the paper by Owens [3].

4.5. By increasing the dimension of the constraint space by 1 and by increasing the number of constraints by r one can rewrite (P) as a differentiable convex program to which Polak [4] is applicable, e.g.,

$$\begin{cases} g_i(x) \leqslant 0, & i = 1,..., m, \\ v_j(x) \leqslant y, & j = 1,..., r, \\ f(x) + y & \text{(min)}. \end{cases}$$

Let us mention some of the basic differences between our algorithm and Polak's. Our procedure faces lower dimensional subproblems. Incidentally, we believe this to be a reason for the comparatively rapid convergence we found with our algorithm on problems tested (see [3]). Our method also addresses non-differentiability directly. See also remarks in Rockafellar [7, pp. 2–3] in connection with this point of view. Polak projects the gradient of the differentiable objective function on the supporting tangent vector spaces. Following Rosen [8, 9] he constructs the appropriate projection matrices for this. In contrast we use the point in a portion of ε-subdifferential nearest to 0, which we obtain by suitable quadratic programs. We do not stipulate a certain assumption of linear independency Sect. 4.5 of [4, paragraph 92]. Our method, like Polak's, is a method of feasible directions. However, we build feasibility in an entirely different way. Also note that ε-binding maximands are not used in Step 11.

4.6. Practical implementation on the computer shows that Algorithm 4.1 is viable and applies well to a broad class of problems, linear [10, 11] and non-linear (see also Sect. 7). In fact, the convergence is quite good, as found by Rubin [10] and Owens [3]. Owens [3] retested some of the classic examples of Wolfe [12], Powell (reported in [12] and [10]), Wolfe [13], Dem'yanov and Malozemov [1], and others. Even in examples constructed to exhibit jamming the present algorithm converged quickly. On smooth problems the convergence was not any slower than some algorithms for the differentiable case with no anti-jamming precautions. Numerical results on Polak's [4] algorithm appear to be unavailable. For details of these we refer to Owens [3].

5. Feasibility of the Algorithm

5.0. We now turn to the task of proving that the steps in the algorithm are well formulated, i.e., are implementable and that, in fact, the algorithm converges to the solution of (P). Through a sequence of lemmas we prove feasibility of the algorithm in this section. Using these lemmas a proof that the algorithm converges is given in the next section. The proof is more involved than the corresponding proof in [11].

We need some more terminology and notation. When $F: \mathbb{R}^d \to]-\infty, \infty]$ is a convex function its *ε-subdifferential* $\partial_\varepsilon F(x)$, where $\varepsilon \geqslant 0$, is defined by saying

$$u \in \partial_\varepsilon F(x) \qquad \text{iff} \quad F(y) \geqslant F(x) + u(y-x) - \varepsilon, \ \forall y \in \mathbb{R}^d. \tag{5.0.1}$$

$\partial_0 F(x)$ is the *subdifferential* of F at x which we denote by $\partial F(x)$. Any $u \in \partial F(x)$ is referred to as a *subgradient* of F at x. More explicitly, u satisfies the subgradient inequality

$$F(y) \geqslant F(x) + u(y-x), \qquad \forall y \in \mathbb{R}^d. \tag{5.0.2}$$

Note, however, that $\partial F(x)$ can be empty. See Rockafellar [6] for all these and related notions. Let χ denote the indicator function of the set X, namely $\chi(x) = 0$, if $x \in X$, and $\chi(x) = \infty$, if $x \notin X$. Then $F = f + v + \chi$ is convex on the whole space and minimizing $F(x)$, $x \in \mathbb{R}^d$ is equivalent to the constrained minimization problem (P). We keep the earlier notation and formulate the lemmas. We begin by collecting some properties of the index sets introduced in Section 3.

5.1. Lemma. *To each $x \in X$ and $\varepsilon \geqslant 0$ there is a neighborhood V of x such that*

$$I_\varepsilon(y) \subset I_\varepsilon(x), \qquad \forall y \in V \cap X, \tag{5.1.1}$$

and

$$J_\varepsilon(y) \subset J_\varepsilon(x), \qquad \forall y \in V \cap X. \tag{5.1.2}$$

Proof. We verify (5.1.1). For every $i \notin I_\varepsilon(x)$, $g_i(x) < -\varepsilon$. Hence there is a neighborhood V of x such that $g_i(y) < -\varepsilon$, for every $y \in V \cap X$. This means that $i \notin I_\varepsilon(y)$, proving (5.1.1). Analogously one proves (5.1.2) by considering the functions $w_j = v_j - v$.

5.2. LEMMA. *Given $x \in X$, there exists $\rho > 0$ such that*

$$I_\varepsilon(x) = I_0(x) \qquad \text{for} \quad 0 \leqslant \varepsilon \leqslant \rho, \tag{5.2.1}$$

$$J_\varepsilon(x) = J_0(x) \qquad \text{for} \quad 0 \leqslant \varepsilon \leqslant \rho. \tag{5.2.2}$$

Proof. Given $x \in X$, note that

$$I_0(x) \subset I_\varepsilon(x) \subset I_\rho(x) \qquad \text{if} \quad 0 \leqslant \varepsilon \leqslant \rho. \tag{5.2.3}$$

In case $I_0(x) = \{1,..., m\}$, the lemma is clear. Hence assume that $\{1,..., m\} \backslash I_0(x)$ is non-empty. In this case, there exists $\rho > 0$ such that

$$\max_{i \notin I_0(x)} g_i(x) < -\rho.$$

This implies that whenever $i \notin I_0(x)$, then $i \notin I_\rho(x)$. In view of (5.2.3) we conclude $I_0(x) = I_\rho(x)$ and (5.2.1) follows. By considering $v_j - v$ analogously we see the validity of (5.2.2).

5.3. LEMMA. *Let $x_k \in X$ and $(x_{k'})$ a subsequence such that $x_{k'} \to x \in X$ and $\varepsilon_{k'} \downarrow 0$. Then*

$$I_{\varepsilon_{k'}}(x_{k'}) \subset I_0(x) \qquad \textit{for all sufficiently large } k'.$$

Proof. We may assume that $\{1,..., m\} \backslash I_0(x)$ is nonempty. There exists $\varepsilon > 0$ such that

$$\max_{i \notin I_0(x)} g_i(x) < -\varepsilon. \tag{5.3.1}$$

Let $i \notin I_0(x)$ so that $g_i(x) < 0$. Then $g_i(x_{k'}) < 0$ for k' sufficiently large, for all $i \notin I_0(x)$. Also $\varepsilon_{k'} < \varepsilon$ for all k' sufficiently large. If possible, let $i \in I_{\varepsilon_{k'}}(x_{k'}) \backslash I_0(x)$; we shall derive a contradiction. Since $i \in I_{\varepsilon_{k'}}(x_{k'})$,

$$g_i(x_{k'}) \geqslant -\varepsilon_{k'} > -\varepsilon.$$

Hence $g_i(x) \geqslant -\varepsilon$. Since $i \notin I_0(x)$, this contradicts (5.3.1), completing the proof.

5.4. LEMMA. *Let $x_k \in X$ and $\varepsilon_k \downarrow \varepsilon > 0$. Suppose that $(x_{k'})$ is a subsequence of (x_k) converging to x. Then for the subsequence $(x_{k'})$ we have*

$$I_0(x) \subset I_{\varepsilon_k}(x_{k'}) \qquad \text{for all } k' \text{ sufficiently large.} \tag{5.4.1}$$

Similarly, $J_0(x) \subset J_{\varepsilon_k}(x_{k'})$, for all k' sufficiently large.

Proof. Let $i \in I_0(x)$. Since $g_i(x) = 0$, $0 \geqslant g_i(x_{k'}) \to 0$; and hence $g_i(x_{k'}) \geqslant -\varepsilon$ for all k' sufficiently large. For these sufficiently large k', we see that $g_i(x_{k'}) \geqslant -\varepsilon_{k'}$, which proves the lemma.

5.5. LEMMA. *For each $x \in X$ and $\varepsilon > 0$, there exists a $\delta > 0$ such that*

$$J_0(x) \subset J_\varepsilon(y) \qquad \text{whenever} \quad |x - y| < \delta, \quad y \in X.$$

Proof. This is essentially the same as Lemma 5.6 in [11].

5.6. LEMMA. *For all $\varepsilon \geqslant 0$ and $x \in X$, $K_\varepsilon(x) \subset \partial_\varepsilon v(x)$.*

Proof. Trivial changes in the proof of Lemma 5.1 in [11] yields this lemma.

5.7. LEMMA. *$\partial v(x) = K_0(x)$ for every $x \in X$.*

Proof. This is a known result. See, for example, [5]. Also, minor alteration in the proof of Lemma 5.4 in [11] yields a direct simple proof of this lemma.

5.8. From the last lemma and [6] we immediately see that

$$v'(x, y) = \max\{uy \mid u \in K_0(x)\}, \tag{5.8.1}$$

where $v'(x; y) = \lim_{\alpha \downarrow 0}\{v(x + \alpha y) - v(x)\}/\alpha$ is the directional derivative of v at x in the direction y.

5.9. LEMMA. *$\partial F(x) = \nabla f(x) + K_0(x) + C_0(x)$ for all $x \in X$.*

Proof. The functions f, v and the indicator function χ of the set X are proper and convex. It is well known that for $x \in X$, $\partial\chi(x) = C_0(x)$. Since X has non-empty interior by Rockafellar [6] the lemma follows.

The next two lemmas show that the stopping criteria in Steps 2 and 4 of the algorithm are well chosen.

5.10. LEMMA. *If $c \in X$ is such that $\nabla f(c) = \nabla v_j(c) = 0$, $j = 1, \ldots, r$, then c is the minimizer of F.*

Proof. In this case $K_0(c) = \{0\}$, because each $\nabla v_j(c) = 0$. Also since $\nabla f(c) = 0$, we see that $0 \in \partial F(c)$, as $0 \in C_0(c)$. This implies that c is a minimizer of F; uniqueness of c is ensured by the strict convexity of F.

5.11. LEMMA. *If $y_0 = 0$ in Step 4 of the algorithm, then x_k is the minimizer of F.*

Proof. $y_0 = 0$ implies that $0 \in \partial F(x_k)$, a necessary and sufficient condition for x_k to minimize F. The strict convexity of f ensures that the minimizer of F is unique.

5.12. LEMMA. Step 7 of the algorithm is not executed infinitely often in any one iteration.

Proof. If Step 7 is executed infinitely often in a certain iteration then the index k remains unchanged from that iteration onwards. By Lemma 5.2 there exists arbitrarily small $\varepsilon > 0$ such that $I_\varepsilon(x_k) = I_0(x_k)$ and $J_\varepsilon(x_k) = J_0(x_k)$. Due to expressions (3.1), (3.2), (3.3), and (3.4)

$$\nabla f(x_k) + K_\varepsilon(x_k) + C_\varepsilon(x_k) = \nabla f(x_k) + K_0(x_k) + C_0(x_k),$$

for such ε. Hence we find that $y_0 = y_\varepsilon$ for arbitrarily small $\varepsilon > 0$. Since Step 7 is executed indefinitely and $\varepsilon \downarrow 0$, we must have $y_\varepsilon \to 0$. Hence $y_0 = 0$; in which case we could not have reached Step 7 at all; a contradiction.

5.13. Step 8 of the algorithm in general requires the solution of a linear program. We have to show that this linear program has a minimal solution. We do this now. Recall that one says that a convex cone C is *pointed* iff C contains no lines or equivalently $C \cap (-C) = \{0\}$.

5.14. LEMMA. *Let $a_1, ..., a_n$ be nonzero vectors such that* cone$\{a_1, ..., a_n\}$ *is pointed. Let $\Gamma = [\gamma_{ij}]$ be the $n \times n$ Gramian matrix, where $\gamma_{ij} = a_i a_j$. Then the linear programming problem*

$$\begin{aligned} &\sum_i \gamma_{ij}\mu_i \geqslant |a_j|, && j = 1, ..., n, \\ &\mu_i \geqslant 0, && i = 1, ..., n, \\ &\mu_1 + \cdots + \mu_n && \text{(min)} \end{aligned} \tag{5.14.1}$$

has a minimizer.

Proof. We first show that (5.14.1) is feasible. For this purpose consider the auxiliary linear programming problem

$$\begin{aligned} (\Gamma\mu)_j &\geqslant |a_j|, \\ \mu_i &\geqslant 0, \\ 0\mu \quad & \text{(min)}. \end{aligned} \tag{5.14.2}$$

Here $\mu = (\mu_1, ..., \mu_n)$. Note that $\Gamma = A'A$, where A is the matrix whose columns are $a_1, ..., a_n$ and A' denotes the transpose of A. Hence, the linear program dual to (5.14.2) is

$$\begin{aligned} (\Gamma\lambda)_j \leqslant 0, \qquad & \lambda_i \geqslant 0, \\ \lambda_1 |a_1| + \cdots + \lambda_n |a_n| \qquad & (\max). \end{aligned} \tag{5.14.3}$$

Clearly 0 is feasible for (5.14.3). We now show that 0 is a maximal solution of (5.14.3) with value 0 and hence its dual (5.14.2) also has an optimal solution with value zero. This then would show that (5.14.2) is feasible. If λ is any vector feasible for (5.14.3), then $\lambda\Gamma\lambda \leqslant 0$, i.e., $\lambda A'A\lambda \leqslant 0$. This yields $|A\lambda|^2 \leqslant 0$ and hence $A\lambda = 0$. Since the cone$\{a_1, ..., a_n\}$ is pointed with $\lambda \geqslant 0$, this implies $\lambda = 0$; proving that 0 is the only feasible solution of (5.14.3) and therefore 0 is the optimal solution of (5.14.3). The feasibility of (5.14.1) is now clear.

The linear program dual to (5.14.1) is the problem

$$\begin{aligned} (\Gamma\lambda)_j \leqslant 1, \qquad & j = 1, ..., n, \\ \lambda_i \geqslant 0, \qquad & i = 1, ..., n, \\ \lambda_1 |a_1| + \cdots + \lambda_n |a_n| \qquad & (\max). \end{aligned} \tag{5.14.4}$$

The vector $\lambda = 0$ is clearly feasible for (5.14.4). This, in view of the just proven feasibility of (5.14.1) and the duality theorem of linear programming, implies that (5.14.1) has an optimal solution, completing the proof of the lemma.

We next find an upper bound for the value of problem (5.14.1) and then use this to obtain an upper bound for the length of the vector $\bar{\mu}_1 a_1 + \cdots + \bar{\mu}_n a_n$, where $(\bar{\mu}_1, ..., \bar{\mu}_n)$ is a minimal solution of (5.14.1).

5.15. LEMMA. *Let $a_1, ..., a_n$ be as in Lemma* 5.14. *Let $e_j = a_j/|a_j|$, $E =$ conv$\{e_j \mid 1 \leqslant j \leqslant n\}$, and $w = N|E|$. Let*

$$u = \bar{\mu}_1 a_1 + \cdots + \bar{\mu}_n a_n, \tag{5.15.1}$$

where $(\bar{\mu}_1, ..., \bar{\mu}_n)$ is a minimizer of (5.14.1). *Then*

$$1 \leqslant |u| \leqslant (\max_i |a_i|)/(|w|^2 \min_i |a_i|). \tag{5.15.2}$$

Proof. Since the cone$\{a_1, ..., a_n\}$ is pointed, it is easily seen that $0 \notin E$ and hence $w \neq 0$. By (3.7), $wz \geqslant |w|^2$, $\forall z \in E$. This yields the inequality

$$wa_j \geqslant |w|^2 |a_j|, \qquad \forall j. \tag{5.15.3}$$

Now there exist $\lambda_i \geqslant 0$, $\sum \lambda_i = 1$ such that $w = \sum \lambda_i e_i$. For these λ_i, by (5.15.3) we have

$$\sum_i \lambda_i(a_i a_j)/|a_i| \geqslant |w|^2 |a_j|. \tag{5.15.4}$$

If we define μ_i by $\mu_i = \lambda_i/(|a_i|\,|w|^2)$, then (5.15.4) shows that $(\mu_1,..., \mu_n)$ is feasible for (5.14.1) and hence

$$\begin{aligned}\sum_i \bar{\mu}_i \leqslant \sum_i \mu_i &= |w|^{-2} \sum_i \lambda_i/|a_i| \\ &\leqslant |w|^{-2}/(\min_i |a_i|).\end{aligned} \tag{5.15.5}$$

Hence

$$\begin{aligned}|u| = \left|\sum \bar{\mu}_i a_i\right| &\leqslant \sum \bar{\mu}_i |a_i| \\ &\leqslant (\max_i |a_i|) \sum \bar{\mu}_i \\ &\leqslant (\max_i |a_i|)/(|w|^2 \min_i |a_i|).\end{aligned}$$

Also from the relation $\sum_i \gamma_{ij} \bar{\mu}_i \geqslant |a_j|$ in (5.14.1) we get $(\sum_i \bar{\mu}_i a_i)\, a_j \geqslant |a_j|$, i.e., $|a_j| \leqslant u a_j$. By Schwarz's inequality we get $|u| \geqslant 1$, since $a_j \neq 0$, $\forall j$. This completes the verification of (5.15.2).

5.16. LEMMA. *Let $\varepsilon_0 > 0$ be as in Step 3 of the algorithm and x any point in X. Then the* cone $C_\varepsilon(x)$ *is pointed for $0 \leqslant \varepsilon \leqslant \varepsilon_0$. Moreover, $\nabla g_i(x) \neq 0$, $\forall i \in I_\varepsilon(x)$.*

Proof. Since $0 \leqslant \varepsilon \leqslant \varepsilon_0 < -\max_{1 \leqslant i \leqslant m} g_i(a)$ we see that $g_i(a) < -\varepsilon$, $\forall i$, so that $I_\varepsilon(a)$ is empty and hence $C_\varepsilon(a) = \{0\}$. The statement $\nabla g_i(a) \neq 0$, $\forall i \in I_\varepsilon(a)$ is indeed true. The lemma needs no proof if $I_\varepsilon(x)$ is empty. So consider the case when $I_\varepsilon(x)$ is nonempty. In this case $x \neq a$. Put $u = a - x$. Now

$$g_i(a) \geqslant g_i(x) + \nabla g_i(x)(a-x) \geqslant -\varepsilon + \nabla g_i(x) u, \qquad \forall i \in I_\varepsilon(x).$$

Hence

$$\nabla g_i(x) u \leqslant g_i(a) + \varepsilon < 0, \qquad \forall i \in I_\varepsilon(x). \tag{5.16.1}$$

This shows that $\nabla g_i(x) \neq 0$ for every $i \in I_\varepsilon(x)$. Also, if $z = \sum \lambda_i \nabla g_i(x)$, where $\lambda_i \geqslant 0$, $i \in I_\varepsilon(x)$, then $zu < 0$, if $\lambda_i > 0$ for some $i \in I_\varepsilon(x)$. Hence $zu < 0$ for every nonzero z in $C_\varepsilon(x)$, proving that $C_\varepsilon(x)$ is pointed; for z, $-z \in C_\varepsilon(x)$ implies that $z = 0$.

5.17. LEMMA. *Let $\varepsilon_k > 0$ be as in the algorithm, $I_{\varepsilon_k}(x_k)$ nonempty, and u_k be as defined in Step 8 of the algorithm. Then*

$$\text{(i)} \quad u_k s_k \geqslant 0, \tag{5.17.1}$$

$$\text{(ii)} \quad \nabla g_i(x_k)\, u_k \geqslant |\nabla g_i(x_k)|, \qquad i \in I_{\varepsilon_k}(x_k), \tag{5.17.2}$$

where s_k is defined in Step 6 of the algorithm.

Proof. Since $s_k = N|\nabla f(x_k) + K_{\varepsilon_k}(x_k) + C_{\varepsilon_k}(x_k)|$, for $i \in I_{\varepsilon_k}(x_k)$, both s_k and $s_k + \nabla g_i(x_k)$ belong to $\nabla f(x_k) + K_{\varepsilon_k}(x_k) + C_{\varepsilon_k}(x_k)$. By (3.7) then $s_k(s_k + \nabla g_i(x_k) - s_k) \geqslant 0$. Thus

$$\nabla g_i(x_k)\, s_k \geqslant 0, \qquad \forall i \in I_{\varepsilon_k}(x_k). \tag{5.17.3}$$

By Lemma 5.16 $C_{\varepsilon_k}(x_k)$ is pointed and hence by Lemma 5.14 the linear program in Step 8 of the algorithm has an optimal solution $(\bar{\mu}_i)$ such that

$$u_k = \sum \{\bar{\mu}_i \nabla g_i(x_k) \mid i \in I_{\varepsilon_k}(x_k)\}.$$

This with (5.17.3) implies (5.17.1). If $\Gamma = |\gamma_{ij}|$, where $\gamma_{ij} = \nabla g_i(x_k)\, \nabla g_j(x_k)$, then by the linear program in Step 8

$$|\nabla g_i(x_k)| \leqslant (\bar{\mu}\Gamma)_i = \nabla g_i(x_k)\, u_k.$$

This is inequality (5.17.2), completing the proof.

5.18. LEMMA. *Let $s_k \neq 0$ and $t_k = s_k + \lambda u_k$, u_k as in Lemma 5.17. Then $-t_k$ is a feasible direction at x_k for every $\lambda > 0$.*

Proof. If $I_0(x_k)$ is empty then every direction is feasible at x_k. So assume that $I_0(x_k)$ is non-empty. By convexity of X, if the lemma were false, then there is $\delta > 0$ such that $x_k - \alpha t_k \notin X$, $0 < \alpha \leqslant \delta$. There exists $i \in I_0(x_k)$ such that $g_i(x_k - \alpha t_k) > 0$, $0 < \alpha \leqslant \delta$. This yields

$$g_i(x_k - \alpha t_k) - g_i(x_k) > 0, \qquad 0 < \alpha \leqslant \delta.$$

Dividing by $\alpha > 0$ and allowing $\alpha \downarrow 0$ we get $\nabla g_i(x_k)\, t_k \leqslant 0$. But by the previous lemma

$$\nabla g_i(x_k)\, t_k = \nabla g_i(x_k)\, s_k + \lambda \nabla g_i(x_k)\, u_k > 0, \qquad \forall \lambda > 0,$$

a contradiction.

5.19. LEMMA. *Let $s_k \neq 0$ and λ_k, t_k be as in Step 9 of the algorithm. Then $-t_k$ is a feasible direction of strict descent at x_k.*

Proof. Note that if $I_{\varepsilon_k}(x_k)$ is empty then $u_k = 0$ so that t_k reduces to s_k in this case. The directional derivative of a convex function is a sublinear function of the direction, hence

$$\begin{aligned} F'(x_k; -t_k) &= F'(x_k; -s_k - \lambda_k u_k) \\ &\leqslant F'(x_k; -s_k) + \lambda_k F'(x_k; -u_k). \end{aligned} \tag{5.19.1}$$

By (5.8.1)

$$v'(x_k; -w) = \max\{-wy \mid y \in K_0(x_k)\}. \tag{5.19.2}$$

Hence

$$\begin{aligned} F'(x_k; -s_k) &= f'(x_k; -s_k) + v'(x_k; -s_k) \\ &= -\nabla f(x_k)\, s_k + \max\{-s_k y \mid y \in K_0(x_k)\} \\ &= -\min\{(\nabla f(x_k) + y)\, s_k \mid y \in K_0(x_k)\}. \end{aligned} \tag{5.19.3}$$

Since $K_0(x_k) \subset K_{\varepsilon_k}(x_k)$, $\nabla f(x_k) + y \in \nabla f(x_k) + K_{\varepsilon_k}(x_k) + C_{\varepsilon_k}(x_k)$, and so by (3.7),

$$s_k(\nabla f(x_k) + y) \geqslant |s_k|^2. \tag{5.19.4}$$

By (5.19.3) and (5.19.4) we see that

$$F'(x_k; -s_k) \leqslant -|s_k|^2. \tag{5.19.5}$$

Again by (5.19.2)

$$\begin{aligned} v'(x_k; -u_k) &= \max\{-u_k y \mid y \in K_0(x_k)\} \\ &= \max\{-\nabla v_j(x_k)\, u_k \mid j \in J_0(x_k)\}. \end{aligned} \tag{5.19.6}$$

By (5.19.1), (5.19.5), and (5.19.6) we arrive at

$$F'(x_k; -t_k) \leqslant -|s_k|^2 + \lambda_k\{-\nabla f(x_k)\, u_k + \max_{j \in J_0(x_k)} (-\nabla v_j(x_k)\, u_k)\}. \tag{5.19.7}$$

Now

$$\begin{aligned} &\{-\nabla f(x_k)\, u_k + \max_{j \in J_0(x_k)} (-\nabla v_j(x_k)\, u_k)\} \\ &\leqslant \{|\nabla f(x_k)| + \max_{j \in J_0(x_k)} |\nabla v_j(x_k)|\}\, |u_k| \\ &\leqslant \{|\nabla f(x_k)| + \max_{1 \leqslant j \leqslant r} |\nabla v_j(x_k)|\}\, |u_k| \\ &= M_k \qquad \text{by Step 8 of the algorithm.} \end{aligned} \tag{5.19.8}$$

By Step 9 of the algorithm $\lambda_k = |s_k|^2/(2M_k + 1)$; hence

$$F'(x_k; -t_k) \leqslant -|s_k|^2 + M_k |s_k|^2/(2M_k + 1)$$
$$\leqslant -|s_k|^2/2 < 0. \tag{5.19.9}$$

This inequality proves the lemma because $-t_k$ is feasible by Lemma 5.18.

5.20. Lemma. *The number $\bar{\alpha}_k$ defined in Step 7 of the algorithm is positive.*

Proof. If $I_{\varepsilon_k}(x_k)$ is empty then in view of Lemma 5.18 the lemma is clear. So consider the case when $I_{\varepsilon_k}(x_k)$ is not empty. Using (5.17.3) and (5.17.2) we see that for $i \in I_{\varepsilon_k}(x_k)$

$$\nabla g_i(x_i)\, t_k = \nabla g_i(x_k)\, s_k + \lambda_k \nabla g_i(x_k)\, u_k$$
$$\geqslant \lambda_k |\nabla g_i(x_k)|. \tag{5.20.1}$$

Now $\lambda_k > 0$, since $s_k \neq 0$. Also by Lemma 5.16, $|\nabla g_i(x_k)| > 0$. This shows that there exists $\delta > 0$ such that $g_i(x_k - \alpha t_k) \leqslant g_i(x_k)$, $\forall i \in I_{\varepsilon_k}(x_k)$, $0 \leqslant \alpha \leqslant \delta$. This fact with Lemma 5.18 proves that $\bar{\alpha}_k$ is positive.

The next two lemmas explain the choice of α_k and z_k in Step 11 of the algorithm.

5.21. Lemma. *Let $s_k \neq 0$ and define φ on $[0, \bar{\alpha}_k]$ by $\varphi(\alpha) = F(x_k - \alpha t_k)$. If $\bar{\alpha}_k$ is not a minimizer of φ on $[0, \bar{\alpha}_k]$, then z_k satisfying Step 11 of the algorithm exists.*

Proof. This is essentially Lemma 5.12 of [11] and the proof in [11] carries over verbatim with s_k occuring in the proof of Lemma 5.12 in [11] replaced by t_k here.

The number α_k determined in Step 11 of the algorithm has the following property:

5.22. Lemma. *Let $s_k \neq 0$ and φ be as in the previous lemma. Then α_k is the unique minimizer of φ on $[0, \bar{\alpha}_k]$. Moreover, α_k is positive.*

Proof. This corresponds to Lemma 5.13 of [11] and the proof therein carries over with s_k replaced by t_k.

5.23. Corollary. *Let $s_k \neq 0$ and $x_{k+1} = x_k - \alpha_k t_k$ as in Step 12 of the algorithm. Then $F(x_{k+1}) < F(x_k)$.*

Proof. This follows from Lemma 5.22 and the observation that $F'(x_k; -t_k) < 0$.

6. CONVERGENCE OF THE ALGORITHM

Lemmas of the previous section prove that the algorithm is feasible and that F decreases at each iteration. We now turn to lemmas leading to a convergence proof.

6.1. LEMMA. *Let $\bar{x} \in X$ be the minimizer of F and $\bar{x}$ be a cluster point of the sequence (x_k). Then (x_k) converges to $\bar{x}$.*

Proof. Same as proof of Lemma 5.15 of [11].

6.2. LEMMA. *Let 0 be a cluster point of the sequence (s_k). Then the sequence (x_k) converges to $\bar{x}$, the minimizer of F.*

Proof. We pass to corresponding subsequences $(s_{k'})$ and $(x_{k'})$ such that $s_{k'} \to 0$ and $x_{k'} \to \hat{x} \in X$. We shall show that $\hat{x}$ minimizes F, so that by the previous lemma $x_k \to \bar{x}$. Since the restriction of F to X is continuous from within X, to prove that $\hat{x}$ is a minimizer of F, it suffices to show that $F(y) \geqslant F(\hat{x})$ for all $y \in \operatorname{int} X$. Note that $\operatorname{int} X$ is non-empty. We now verify that for every $y \in \operatorname{int} X$,

$$g_i(y) < 0, \qquad i = 1, ..., m. \tag{6.2.1}$$

Recall that a satisfies $g_i(a) < 0$, for every i, and hence to prove (6.2.1) we may assume that $y \neq a$. Since $y \in \operatorname{int} X$, there exists $\alpha > 0$ such that $z = y + \alpha(y - a) \in X$. Hence $(1 + \alpha) y = z + \alpha a$ and by convexity of g_i

$$(1 + \alpha) g_i(y) \leqslant g_i(z) + \alpha g_i(a) < 0,$$

which proves (6.2.1).

By (6.2.1) we have

$$\begin{aligned} 0 > g_i(y) &\geqslant g_i(\hat{x}) + \nabla g_i(\hat{x})(y - \hat{x}) \\ &= \nabla g_i(\hat{x})(y - \hat{x}), \qquad \forall i \in I_0(\hat{x}). \end{aligned} \tag{6.2.2}$$

Since $s_{k'} \to 0$ the sequence $(\varepsilon_{k'})$ decreases to zero. Also $x_{k'} \to \hat{x}$. Hence by Lemma 5.3

$$I_{\varepsilon_{k'}}(x_{k'}) \subset I_0(\hat{x}), \tag{6.2.3}$$

for sufficiently large k'. By the continuity of the function $x \mapsto \nabla g_i(x)(y - x)$ at $\hat{x}$, the fact that $x_{k'} \to \hat{x}$, the relations (6.2.2) and (6.2.3), we find that for sufficiently large k'

$$\nabla g_i(x_{k'})(y - x_{k'}) < 0, \qquad \forall i \in I_{\varepsilon_{k'}}(x_{k'}). \tag{6.2.4}$$

At this stage we complete the proof of this lemma by repeating the reasoning from Eq. (5.16.3) onwards of Lemma 5.16 of [11].

6.3. LEMMA. *If the sequence* (ε_k) *defined in the algorithm converges to zero, then the sequence* (x_k) *converges to* $\bar{x}$, *the minimizer of* F.

Proof. By Lemma 5.12, Step 7 of the algorithm is executed finitely often per iteration. Hence a subsequence $(\varepsilon_{k'})$ of (ε_k) may be chosen such that

$$\varepsilon_{k'+1} = \varepsilon_{k'}/2 \qquad \text{and} \qquad \|y_{\varepsilon_{k'}}\|^2 \leqslant \varepsilon_{k'}.$$

where y_ε was defined in Step 5 of the algorithm. Since $\varepsilon_{k'} \to 0$, $y_{\varepsilon_{k'}} \to 0$. We replace all the occurrences of $s_{k'}$ in the proof of the previous lemma by $y_{\varepsilon_{k'}}$ and repeat the reasoning therein to see the validity of the present lemma.

6.4. LEMMA. *The sequence* (s_k) *is bounded.*

Proof. Note that

$$\nabla f(x_k) + K_0(x_k) \subset \nabla f(x_k) + K_{\varepsilon_k}(x_k) + C_{\varepsilon_k}(x_k),$$

so that

$$\|s_k\| \leqslant \|\nabla f(x_k)\| + \|N[K_0(x_k)]\|.$$

Since $K_0(x_k) = \operatorname{conv}\{\nabla v_j(x_k) \mid j \in J_0(x_k)\}$,

$$\begin{aligned} \|N[K_0(x_k)]\| &\leqslant \max\{\|\nabla v_j(x_k)\| \mid j \in J_0(x_k)\} \\ &\leqslant \max_{1 \leqslant j \leqslant r} \|\nabla v_j(x_k)\|. \end{aligned}$$

Hence

$$\|s_k\| \leqslant \max_{x \in X} \|\nabla f(x)\| + \max_{x \in X} \max_{1 \leqslant j \leqslant r} \|\nabla v_j(x)\|.$$

The right-hand side of the above inequality is finite since the functions f and v_j are all of class C^1 on the compact set X. This proves the lemma.

6.5. LEMMA. *Let* $\varepsilon_0 > 0$ *be as in Step 3 of the algorithm and* $0 \leqslant \varepsilon \leqslant \varepsilon_0$. *Let* $x \in X$ *and* $I_\varepsilon(x)$ *be non-empty. Then there exists* $\delta > 0$ *such that*

$$d(y) \geqslant \tfrac{1}{2} d(x) > 0, \qquad \forall \|y - x\| < \delta, \quad y \in X, \tag{6.5.1}$$

where

$$d(y) = \inf \left\{ \left\| \sum_i \lambda_i e_i(y) \right\| \;\middle|\; \lambda_i \geqslant 0, \sum_i \lambda_i = 1, i \in I_\varepsilon(y) \right\}.$$

and

$$e_i(y) = \nabla g_i(y)/|\nabla g_i(y)|, \qquad i \in I_\varepsilon(y).$$

Proof. By Lemma 5.16, $\nabla g_i(x) \neq 0$, $\forall i \in I_\varepsilon(x)$ and the cone $C_\varepsilon(x)$ is pointed. This ensures that $0 \notin \operatorname{conv}\{e_i(x) \mid i \in I_\varepsilon(x)\}$ and hence $d(x) > 0$. Choose $\delta > 0$ such that by Lemma 5.1, $I_\varepsilon(x) \supset I_\varepsilon(y)$ if $|x - y| < \delta$, $y \in X$. We can also require that $\delta > 0$ be such that $\nabla g_i(y) \neq 0$ and

$$|e_i(y) - e_i(x)| < \tfrac{1}{2} d(x), \tag{6.5.2}$$

$\forall i \in I_\varepsilon(x)$, $y \in X$, $|y - x| < \delta$.

Now

$$d(y) = \inf \left\{ \left| \sum \lambda_i e_i(y) \right| \,\middle|\, \lambda_i \geqslant 0, \sum \lambda_i = 1, i \in I_\varepsilon(y) \right\},$$

$$\geqslant \inf \left\{ \left| \sum \lambda_i e_i(y) \right| \,\middle|\, \lambda_i \geqslant 0, \sum \lambda_i = 1, i \in I_\varepsilon(x) \right\}.$$

Also

$$\left| \sum_{i \in I_\varepsilon(x)} \lambda_i e_i(y) \right| \geqslant \left| \sum_{i \in I_\varepsilon(x)} \lambda_i e_i(x) \right| - \left| \sum_{i \in I_\varepsilon(x)} \lambda_i (e_i(y) - e_i(x)) \right|.$$

This in conjunction with (6.5.2) yields the inequality

$$d(y) \geqslant \inf \left\{ \left| \sum \lambda_i e_i(x) \right| \,\middle|\, \lambda_i \geqslant 0, \sum \lambda_i = 1, i \in I_\varepsilon(x) \right\} - \frac{1}{2} d(x)$$

$$= \frac{1}{2} d(x).$$

We have already verified that if 0 is a cluster point of either the sequence $(|s_k|)$ or the sequence (ε_k) then (x_k) converges to the minimizer of problem (P). So let us consider the situation when $(|s_k|)$ and (ε_k) are both bounded away from zero. Since (ε_k) is a nonincreasing positive sequence (ε_k) is bounded away from zero iff there is $\varepsilon > 0$ such that $\varepsilon_k \downarrow \varepsilon$. From the steps of the algorithm this can happen iff $\varepsilon_k = \varepsilon$, eventually. Hence in the following lemmas we shall assume that the (ε_k) defined in the algorithm is such that $\varepsilon_k = \varepsilon > 0$, eventually, and that (s_k) is bounded away from 0.

6.6. Lemma. *Let (s_k) be bounded away from zero. Then the sequences (t_k) and (α_k) are both bounded. Moreover, (t_k) is bounded away from zero.*

Proof. Let

$$\delta = \min_{x \in X} \{ |\nabla f(x)| + \max_{1 \leqslant j \leqslant r} |\nabla v_j(x)| \}.$$

Since we have got past Step 2 of the algorithm the continuous function in braces $\{\cdots\}$ is positive on compact X and hence $\delta > 0$. By Step 8 of the algorithm

$$M_k = \{|\nabla f(x_k)| + \max_{1 \leqslant j \leqslant r} |\nabla v_j(x_k)|\} |u_k|$$
$$\geqslant \delta |u_k|,$$

and hence

$$\lambda_k |u_k| = |u_k| |s_k|^2/(2M_k + 1)$$
$$\leqslant M_k |s_k|^2/\{\delta(2M_k + 1)\} < |s_k|^2/(2\delta). \tag{6.6.1}$$

Since $t_k = s_k + \lambda_k u_k$ with (s_k) bounded, by (6.6.1) we find that (t_k) is bounded.

We now show that (t_k) is bounded away from zero. If not, since (t_k) is a bounded sequence, there exists a subsequence $(t_{k'})$ such that $t_{k'} \to 0$. Since the sequence $(s_{k'})$ is also bounded, by passing to another subsequence again denoted by (k'), we can require that $s_{k'} \to s \neq 0$. This implies that $s_{k'} + \lambda_{k'} u_{k'} \to 0$. Hence, $\lambda_{k'} u_{k'} \to -s$ and

$$(\lambda_{k'} u_{k'})\, s_{k'} \to -|s|^2. \tag{6.6.2}$$

By (5.17.1), $u_{k'} s_{k'} \geqslant 0$ for every k' which shows that $(\lambda_{k'} u_{k'})\, s_{k'} = \lambda_{k'}(u_{k'} s_{k'}) \geqslant 0$. By (6.6.2) $s = 0$, a contradiction. Hence (t_k) is bounded away from zero.

Now $\alpha_k |t_k|$ is bounded above by the diameter of X. Since we just showed that $|t_k|$ is bounded away from zero, we conclude that (α_k) is a bounded sequence.

6.7. LEMMA. *Let (ε_k) be as defined in the algorithm. Suppose that $\varepsilon_k = \varepsilon > 0$, eventually. Let (k') be a subsequence such that $I_\varepsilon(x_{k'})$ is nonempty for every k' with $x_{k'} \to x \in X$. Then there exists $M > 0$ and $\theta \geqslant 1$ such that the following hold*:

$$1 \leqslant |u_{k'}| \leqslant \theta, \qquad \forall k', \tag{6.7.1}$$

$$|s_{k'}|^2/(2M + 1) \leqslant \lambda_{k'} \leqslant |s_{k'}|^2, \qquad \forall k'. \tag{6.7.2}$$

Proof. By Lemmas 5.1 and 6.5 there exists $\delta > 0$ such that

$$I_\varepsilon(x) \supset I_\varepsilon(y), \tag{6.7.3}$$

$$d(y) \geqslant d(x)/2, \qquad \forall y \in X, \qquad |y - x| < \delta, \tag{6.7.4}$$

where d was defined in Lemma 6.5. There exists p such that $|x_{k'} - x| < \delta$ if $k' \geqslant p$. Hence, $I_\varepsilon(x_{k'}) \subset I_\varepsilon(x)$, $\forall k' \geqslant p$. This shows that $I_\varepsilon(x)$ is non-empty and hence by Lemma 5.16, $\nabla g_i(x) \neq 0$ for every $i \in I_\varepsilon(x)$. By reducing $\delta > 0$, if necessary, we may assume that

$$\tfrac{1}{2}|\nabla g_i(x)| \leqslant |\nabla g_i(y)| \leqslant \tfrac{3}{2}|\nabla g_i(x)|, \tag{6.7.5}$$

$\forall i \in I_\varepsilon(x)$, $y \in X$, $|y-x| < \delta$. By Lemma 6.5, $d(x) > 0$. Using Lemma 5.15, (6.7.3), (6.7.4), and (6.7.5) we get

$$\begin{aligned} |u_{k'}| &\leqslant \big(\max_{i \in I_\varepsilon(x_{k'})} |\nabla g_i(x_{k'})|\big)/\big(d^2(x_{k'}) \min_{i \in I_\varepsilon(x_{k'})} |\nabla g_i(x_{k'})|\big) \\ &\leqslant 4\big(\max_{i \in I_\varepsilon(x)} |\nabla g_i(x_{k'})|\big)/\big(d^2(x) \min_{i \in I_\varepsilon(x)} |\nabla g_i(x_{k'})|\big) \\ &\leqslant 12\big(\max_{i \in I_\varepsilon(x)} |\nabla g_i(x)|\big)/\big(d^2(x) \min_{i \in I_\varepsilon(x)} |\nabla g_i(x)|\big) \\ &= \beta_x, \qquad \forall k' \geqslant p. \end{aligned} \tag{6.7.6}$$

Since β_x is a positive real number, there exists $\theta > 0$ such that $|u_{k'}| \leqslant \theta$, for every k'. By (5.15.2), $|u_{k'}| \geqslant 1$ as well, so that (6.7.1) is verified.

By Step 8 of the algorithm,

$$\begin{aligned} M_{k'} &= \big(|\nabla f(x_{k'})| + \max_{1 \leqslant j \leqslant r} |\nabla v_j(x_{k'})|\big)\,|u_{k'}| \\ &\leqslant \max_{z \in X} \big(|\nabla f(z)| + \max_{1 \leqslant j \leqslant r} |\nabla v_j(z)|\big)\theta \\ &= M \qquad \text{(say)}. \end{aligned} \tag{6.7.7}$$

By Step 9 of the algorithm,

$$\lambda_{k'} = |s_{k'}|^2/(2M_{k'} + 1) \geqslant |s_{k'}|^2/(2M + 1).$$

The inequality (6.7.2) is now evident, completing the proof of the lemma.

6.8. Lemma. *Let (ε_k) be as defined in the algorithm. Suppose $\varepsilon_k = \varepsilon > 0$ eventually and (s_k) is bounded away from zero. Then the sequence (α_k) converges to zero.*

Proof. If (α_k) does not converge to zero by Lemma 6.6 there is a subsequence $(\alpha_{k'})$ such that $\alpha_{k'} \to \alpha > 0$. We distinguish two cases.

Case 1. We assume that $I_{\varepsilon_{k'}}(x_{k'})$ is empty for an infinity of indices k'. Passing to a subsequence, again denoted by k', we can require that $I_{\varepsilon_{k'}}(x_{k'})$ is empty for every k'. Due to the boundedness of (s_k) and compactness of X we can require $s_{k'} \to s \neq 0$, $x_{k'} \to x \in X$. In the present case, $u_{k'}$ defined in Step 8 of the algorithm is zero and hence $t_{k'} = s_{k'}$ for all k'. Since $(F(x_k))$ is a decreasing sequence all its subsequences converge to $F(x)$. Hence $F(x_{k'+1}) \to$

$F(x)$ also. Since $x_{k'+1} = x_{k'} - \alpha_{k'} s_{k'}$, in the present case, $x_{k'+1} \to x - \alpha s$. We therefore find that

$$F(x - \alpha s) = F(x). \tag{6.8.1}$$

Since

$$F(x_{k'} - \alpha_{k'} s_{k'}) \leqslant F(x_{k'} - \lambda s_{k'}), \qquad \forall \lambda \in [0, \bar{\alpha}_{k'}]$$

and F is convex we find that

$$\begin{aligned} F(x_{k'} - \alpha_{k'} s_{k'}) &\leqslant F(x_{k'} - \alpha_{k'} s_{k'}/2) \\ &\leqslant \{F(x_{k'} - \alpha_{k'} s_{k'}) + F(x_{k'})\}/2 \\ &\leqslant F(x_{k'}). \end{aligned} \tag{6.8.2}$$

In the limit we get

$$F(x - \alpha s) \leqslant F(x - \alpha s/2) \leqslant F(x). \tag{6.8.3}$$

In view of (6.8.1), (6.8.3), and the strict convexity of F we find $\alpha = 0$, a contradiction.

Case 2. We now consider the case when $\alpha_{k'} \to \alpha > 0$ and $I_{\varepsilon_{k'}}(x_{k'}) = I_{\varepsilon}(x_{k'})$ are non-empty for all sufficiently large k'. Once more, we may assume that $s_{k'} \to s \neq 0$ and $x_{k'} \to x \in X$. The hypotheses of Lemma 6.7 are now applicable, so that (6.7.1) and (6.7.2) hold. We may therefore pass to another subsequence, again denoted by k', such that $u_{k'} \to u$, $\lambda_{k'} \to \lambda$. By (6.7.1) and (6.7.2) we also see that

$$|u| \geqslant 1 \qquad \text{and} \qquad \lambda \geqslant |s|^2/(2M + 1). \tag{6.8.4}$$

By Step 9 of the algorithm $t_{k'} = s_{k'} + \lambda_{k'} u_{k'}$ and hence

$$t_{k'} \to s + \lambda u = t \qquad \text{(say)}. \tag{6.8.5}$$

Now $t \neq 0$, by Lemma 6.6. Since $x_{k'+1} = x_{k'} - \alpha_{k'} t_{k'} \to x - \alpha t$, and since $(F(x_k))$ is decreasing to $F(x)$, we see that

$$F(x - \alpha t) = F(t). \tag{6.8.6}$$

Also as in Case 1 above,

$$F(x - \alpha t) \leqslant F(x - \alpha t/2) \leqslant F(x), \tag{6.8.7}$$

which contradicts the strict convexity of F, since $\alpha > 0$. The proof of the lemma is now complete.

6.9. LEMMA. *Let* (ε_k) *be as defined in the algorithm with* $\varepsilon_k = \varepsilon > 0$ *eventually and* (s_k) *bounded away from zero. Let the subsequence* $x_{k'} \to x \in X$. *Then there is a subsequence of* (k'), *again denoted by* (k'), *such that* $I_0(x_{k'}) = I_0(x)$ *for all* k'.

Proof. Since the index sets $I_0(x_{k'})$ are subsets of $\{1, \ldots, m\}$, we can pass to yet another subsequence, again denoted by k', such that $I_0(x_{k'}) = I$, for every k'. We will show that $I = I_0(x)$. If $i \in I$, then $g_i(x_{k'}) = 0$, so that in the limit $g_i(x) = 0$. This shows that $I \subset I_0(x)$. There is nothing to prove if $I_0(x)$ is empty. To prove the reverse inclusion, let $i \in I_0(x) \backslash I$. We shall derive a contradiction. Since $x_{k+1} = x_k - \alpha_k t_k$, with (t_k) shown bounded by Lemma 6.6, and $\alpha_k \to 0$ by Lemma 6.8 we see that $|x_{k+1} - x_k| \to 0$ as $k \to \infty$. Let $M = \max_{z \in X} |\nabla g_i(z)|$. Then there exists k_0 such that

$$|x_{k+1} - x_k| < \varepsilon/(2M), \qquad \forall k \geqslant k_0. \tag{6.9.1}$$

Since $i \notin I$, $g_i(x_{k'}) < 0$. Also since $g_i(x_{k'}) \to 0$, in the sequence of integers (k'), we can find $p \geqslant k_0$ such that

$$0 > -\delta = g_i(x_p) > -\varepsilon/2. \tag{6.9.2}$$

Let q be the first index such that $q > p$ and

$$g_i(x_q) \geqslant -\delta/2. \tag{6.9.3}$$

Now

$$\begin{aligned} g_i(x_{q-1}) &\geqslant g_i(x_q) + \nabla g_i(x_q)(x_{q-1} - x_q) \\ &> -(\delta/2) - M\varepsilon/(2M) > -\varepsilon. \end{aligned}$$

This shows that $i \in I_\varepsilon(x_{q-1})$. By Step 10 of the algorithm

$$g_i(x_q) \leqslant g_i(x_{q-1}). \tag{6.9.4}$$

By (6.9.3) and (6.9.4),

$$g_i(x_{q-1}) \geqslant -\delta/2. \tag{6.9.5}$$

Note that $q - 1 \geqslant p$. If $q - 1 = p$, then (6.9.5) contradicts (6.9.2). If $q - 1 > p$, then (6.9.5) contradicts the choice of q as the smallest index greater than p for which (6.9.3) holds. Hence $I_0(x) = I$, and the proof of the lemma is now complete.

We are finally in a position to prove the convergence of the algorithm.

6.10. THEOREM. *Algorithm* 4.1 *generates either a terminating sequence whose last term is the minimizer of problem* (P) *or an infinite sequence converging to the minimizer of problem* (P).

Proof. In view of Lemmas 5.10 and 5.11, we need only consider the case in which Algorithm 4.1 generates an infinite sequence (x_k). In this case, $s_k \neq 0$ for every k. We assume that (x_k) fails to converge to the solution of (P) and derive a contradiction.

Due to the remarks after Lemma 6.5 we may assume that (s_k) is bounded away from zero and that the non-increasing positive sequence (ε_k) is such that $\varepsilon_k = \varepsilon > 0$, eventually for all k. We distinguish two cases.

Case 1. We assume that there are an infinity of indices k for which z_k in Step 11 of the algorithm are defined and arrive at a contradiction. Denote this subsequence of indices by (k'). Let us consider the situation when $I_\varepsilon(x_{k'})$ are nonempty for all k', eventually. Passing to a further subsequence, if necessary, but denoting the new subsequence again by (k'), because X is compact, $(s_{k'})$, $(u_{k'})$, $(M_{k'})$, $(\lambda_{k'})$ all bounded (Lemmas 6.4, 6.6, and 6.7) we may assume that

$$x_{k'} \to x \in X, \qquad s_{k'} \to s \neq 0, \qquad M_{k'} \to M \geqslant 0, \qquad u_{k'} \to u \neq 0, \tag{6.10.1}$$
$$\lambda_{k'} \to \lambda > 0, \qquad t_{k'} = s_{k'} + \lambda_{k'} u_{k'} \to s + \lambda u = t \neq 0.$$

By Lemma 6.8, $\alpha_k \to 0$ and hence $x_{k'+1} = x_{k'} - \alpha_{k'} t_{k'} \to x$. Passing to a still further subsequence, again denoted (k'), we may assume that there exists sets I, J, and J' such that

$$I_\varepsilon(x_{k'}) = I, \qquad J_\varepsilon(x_{k'}) = J, \qquad J_0(x_{k'+1}) = J', \tag{6.10.2}$$

for all k'. Since $(x_{k'})$ and $(x_{k'+1})$ both converge to x, by Lemmas 5.5 and 5.1 we find that $J_0(x) \subset J_\varepsilon(x_{k'})$ and $J_0(x_{k'+1}) \subset J_0(x)$, respectively, for large enough k'. Hence by (6.10.2) we find that $J' \subset J$. Let us set

$$K(x_{k'}) = \operatorname{conv}\{\nabla v_j(x_{k'}) \mid j \in J\}, \tag{6.10.3}$$

and

$$C(x_{k'}) = \operatorname{cone}\{\nabla g_i(x_{k'}) \mid i \in I\}. \tag{6.10.4}$$

For each k' we have

$$z_{k'} \in \nabla f(x_{k'+1}) + K_0(x_{k'+1}) \tag{6.10.5}$$

and

$$z_{k'} t_{k'} = 0, \tag{6.10.6}$$

where

$$K_0(x_{k'+1}) = \operatorname{conv}\{\nabla v_j(x_{k'+1}) \mid j \in J'\} \tag{6.10.7}$$

and

$$s_{k'} = N[\nabla f(x_{k'}) + K(x_{k'}) + C(x_{k'})]. \tag{6.10.8}$$

Now the carrier (i.e., point-to-set map) $y \mapsto \partial v(y) = K_0(y)$ is upper semicontinuous. (See Rockafellar [6].) Clearly, for each $y \in X$, $\nabla f(y) + K_0(y)$ is a closed set. Hence $y \mapsto \nabla f(y) + K_0(y)$ is a closed carrier. From (6.10.5) we see that $(z_{k'})$ is a bounded sequence, and hence passing to a further subsequence assume that $z_{k'} \to z$. Since $x_{k'+1} \to x$ due to (6.10.5) and the closedness of the carrier $y \mapsto \nabla f(y) + K_0(y)$ we conclude that

$$z \in \nabla f(x) + K_0(x). \tag{6.10.9}$$

Due to (6.10.8)

$$s_{k'} \left(\nabla f(x_{k'}) + \sum_{j \in J} \lambda_j \nabla v_j(x_{k'}) + \sum_{i \in I} \mu_i \nabla g_i(x_{k'}) \right) \geqslant |s_{k'}|^2, \tag{6.10.10}$$

where λ_j, μ_i are all $\geqslant 0$, with $\sum_{j \in J} \lambda_j = 1$. For fixed (λ_j) and (μ_i), we allow $k' \to \infty$ in (6.10.10) to get

$$s \left(\nabla f(x) + \sum_{j \in J} \lambda_j \nabla v_j(x) + \sum_{i \in I} \mu_i \nabla g_i(x) \right) \geqslant |s|^2.$$

This shows that

$$s = N[\nabla f(x) + K^* + C^*], \tag{6.10.11}$$

where

$$K^* = \operatorname{conv}\{\nabla v_j(x) \mid j \in J\} \qquad \text{and} \qquad C^* = \operatorname{cone}\{\nabla g_i(x) \mid i \in I\}.$$

This with (3.7) gives us the inequality

$$(\nabla f(x) + y)s \geqslant |s|^2, \qquad \forall y \in K^*.$$

By Lemma 5.4, $J_0(x) \subset J$ and hence $K_0(x) \subset K^*$. Moreover, by (5.19.3)

$$\begin{aligned} F'(x; -s) &= -\min\{(\nabla f(x) + y)s \mid y \in K_0(x)\} \\ &\leqslant -|s|^2 \qquad \text{by above.} \end{aligned} \tag{6.10.12}$$

As in (5.19.7) from (6.10.12) we now get

$$F'(x; -t) \leqslant -|s|^2 + \lambda\{-\nabla f(x)u + \max_{j \in J_0(x)} (-\nabla v_j(x)u)\}. \tag{6.10.13}$$

Note that (6.10.1) with (5.19.8) shows that

$$(|\nabla f(x)| + \max_{1 \leqslant j \leqslant r} |v_j(x)|)\,|u| \leqslant M.$$

Since $\lambda = |s|^2/(2M+1)$, by (6.10.13)

$$F'(x; -t) \leqslant -|s|^2 + |s|^2 M/(2M+1) \leqslant -|s|^2/2 < 0. \tag{6.10.14}$$

We now show that $-t$ is a feasible direction at x. The proof of Lemma 5.18 shows that to show $-t$ is feasible, it is sufficient to show that

$$\nabla g_i(x)s \geqslant 0 \quad \text{and} \quad \nabla g_i(x)u > 0. \quad \forall i \in I_0(x). \tag{6.10.15}$$

Since $C_0(x) \subset C^*$, the argument used in deriving (5.17.3) applied now with (6.10.11) yields $\nabla g_i(x)s \geqslant 0$, $\forall i \in I_0(x)$. By (5.17.2) we have

$$\nabla g_i(x_{k'})\, u_{k'} \geqslant |\nabla g_i(x_{k'})|, \qquad \forall i \in I_{\varepsilon_k}(x_{k'}).$$

Allowing $k' \to \infty$ and using Lemma 5.4, we get

$$\begin{aligned} \nabla g_i(x)u &\geqslant |\nabla g_i(x)|, && \forall i \in I_0(x) \\ &> 0 && \text{by Lemma 5.16.} \end{aligned}$$

Thus (6.10.15) has been verified. This in view of (6.10.14) shows that $-t$ is a feasible direction of strict descent at x.

At this stage let us consider the situation when $I_\varepsilon(x_{k'})$ are empty for an infinity of k'. Renaming this subsequence again as (k'), by Step 8 of the algorithm we see that $u_{k'} = 0$, $\lambda_{k'} = |s_{k'}|^2$, and $s_{k'} = t_{k'}$, $\forall k'$. By Lemma 5.4 $I_0(x)$ is empty, so $-s$ is a feasible direction of strict descent at x in this case.

As in Lemma 5.19 we form the function φ, where $\varphi(\alpha) = f(x - \alpha t) + v(x - \alpha t)$. Passing to the limit in (6.10.6) we get $zt = 0$. In view of (6.10.9) and Lemma 5.22 we will have to conclude that 0 is a minimizer of φ, contradicting our observation that $-t$ is a feasible direction of strict descent at x.

Case 2. We now take up the case when z_k is undefined for all but a finite number of indices k. This being the case, we might as well assume that z_k is undefined for every k. Then by Step 11, $\alpha_k = \bar{\alpha}_k$ for all k.

We observe that this entails that $I_\varepsilon(x_k)$ are non-empty for all $k \geqslant 1$. If $I_\varepsilon(x_0) = \varnothing$, then since $\alpha_0 = \bar{\alpha}_0$, x_1 belongs to the boundary of X, so that $I_0(x_1) \neq \varnothing$, a fortiori, $I_\varepsilon(x_1) \neq \varnothing$. If $I_\varepsilon(x_k) \neq \varnothing$, then since $\alpha_k = \bar{\alpha}_k$, either $I_\varepsilon(x_k) \cap I_\varepsilon(x_{k+1}) \neq \varnothing$ or $I_\varepsilon(x_{k+1}) \backslash I_\varepsilon(x_k) \neq \varnothing$, i.e., some constraint which is ε-binding at the kth iteration remains ε-binding for $(k+1)$ or else a new constraint has become binding and hence ε-binding also at the $(k+1)$ iteration. So inductively $I_\varepsilon(x_k) \neq \varnothing$, $\forall k \geqslant 1$.

We just verified that if z_k are undefined for all k, then $I_\varepsilon(x_k)$ are non-empty for all $k \geqslant 1$. Once more, we can pass to a subsequence (k') such that $x_{k'} \to x \in X$, $s_{k'} \to 0$. By Lemma 6.7, we pass to yet another subsequence, again denoted by (k'), such that $\lambda_{k'} \to \lambda > 0$, $u_{k'} \to u$, $|u| \geqslant 1$. Then

$$t_{k'} = s_{k'} + \lambda_{k'} u_{k'} \to s + \lambda u = t \qquad \text{(say)}. \tag{6.10.16}$$

By Lemma 6.6 $t \neq 0$ and by Lemma 6.8 $\alpha_k \to 0$, so that by the above $x_{k'+1} \to x$ also. Using Lemma 6.9 we can also require the subsequence (k') to satisfy

$$I_0(x_{k'}) = I_0(x) = I_0(x_{k'+1}), \qquad \forall k'.$$

By passing to a further subsequence, as usual denoted by (k'), we may assume that $I_\varepsilon(x_{k'}) = I$, for all k'. Note that I is non-empty. In this case, since $\alpha_{k'} = \bar{\alpha}_{k'}$ and $I_0(x_{k'}) = I_0(x_{k'+1})$ for every k' by Step 10 of the algorithm, we find that for each k' there exists $i \in I$ such that

$$g_i(x_{k'} - \alpha_{k'} t_{k'}) = g_i(x_{k'}). \tag{6.10.17}$$

Since i must be one of the indices 1 through m, by passing to yet another subsequence, again denoted (k'), we can find a fixed i such that (6.10.17) holds for every k'. Now by Lemma 5.17 and (5.17.3)

$$\begin{aligned}\nabla g_i(x_{k'})\, t_{k'} &= \nabla g_i(x_{k'})(s_{k'} + \lambda_{k'} u_{k'}) \\ &= \nabla g_i(x_{k'})\, s_{k'} + \lambda_{k'} \nabla g_i(x_{k'})\, u_{k'} \\ &\geqslant \lambda_{k'} |\nabla g_i(x_{k'})| > 0,\end{aligned} \tag{6.10.18}$$

since $\lambda_{k'} > 0$ and $\nabla g_i(x_{k'}) \neq 0$, because $i \in I_\varepsilon(x_{k'})$.

We let $\varphi(\alpha) = g_i(x_{k'} - \alpha t_{k'})$, $0 \leqslant \alpha \leqslant \alpha_{k'}$ and observe that $\varphi'(\alpha)$ is a continuous non-decreasing function of α in the interval $[0, \alpha_{k'}]$. Also by (6.10.18), $\varphi'(0+) = -\nabla g_i(x_{k'})\, t_{k'} < 0$. By (6.10.17), $\varphi(\alpha_{k'}) = \varphi(0)$ and $\alpha_{k'} > 0$. We therefore conclude that $\varphi'(\alpha_{k'}-) \geqslant 0$. This means that

$$-t_{k'} \nabla g_i(x_{k'} - \alpha_{k'} t_{k'}) \geqslant 0,$$

i.e.,

$$\nabla g_i(x_{k'+1})\, t_{k'} \leqslant 0. \tag{6.10.19}$$

Allowing $k' \to \infty$ in (6.10.18) and (6.10.19) we get

$$0 \geqslant \nabla g_i(x) t \geqslant \lambda\, |\nabla g_i(x)|.$$

Since $\lambda > 0$, we see that $\nabla g_i(x) = 0$. But since $i \in I_\varepsilon(x_{k'})$, $g_i(x_{k'}) \geqslant -\varepsilon$ and hence $g_i(x) \geqslant -\varepsilon$. We therefore get

$$g_i(a) < -\varepsilon_0 \leqslant -\varepsilon \leqslant g_i(x).$$

This shows that x is not a minimizer of g_i and since g_i is convex, we are contradicting the fact that $\nabla g_i(x) = 0$. The proof that the algorithm generates a sequence converging to the optimal solution is now complete.

7. Mixed Constraints

The algorithm in Section 4 can be combined with that in [11] to handle the presence of affine and non-affine convex constraints. In problem (P) of Section 2, let $g_1, ..., g_p$ all be non-affine, convex, and differentiable on Ω and $g_{p+1}, ..., g_m$ all affine. We now replace condition (SQ) of Section 2 by the generalized Slater's constraint qualification (GSQ), i.e.,

$$\text{There exists } a \in X \text{ such that} \\ g_i(a) < 0, i = 1, ..., p \quad \text{and} \quad g_i(a) \leqslant 0, i = p+1, ..., m. \quad \text{(GSQ)}$$

This affects only the choices of feasible direction and maximum feasible step. The algorithm becomes:

7.1. Algorithm. All steps are the same as in Algorithm 4.1 except that in Steps 8 and 10 define I by

$$I = I_{\varepsilon_k}(x_k) \cap [1, p].$$

Also the proof of convergence in the previous section carries over to this more general case with minor changes.

Acknowledgments

Thanks are due to Professor Robert W. Owens for many valuable discussions and for his preprint [3]. Thanks are also due to the referee for his helpful remarks.

References

1. V. F. Dem'yanov and V. N. Malozemov, "Introduction to Minimax," Wiley, New York, 1974.
2. C. Lemarechal, An extension of Davidon methods to nondifferentiable problems, *Math. Programming Stud.* **3** (1975), 95–109.
3. R. W. Owens, Implementation of subgradient projection algorithm II, *Internat. J. Comput. Math.* **16** (1984).
4. E. Polak, "Computational Methods in Optimization," Academic Press, New York, 1971.
5. B. N. Pshenichnyi, "Necessary Conditions for an Extremum," Dekker, New York, 1971.
6. R. T. Rockafellar, "Convex Analysis," Princeton Univ. Press, Princeton, N.J., 1970.
7. R. T. Rockafellar, "The Theory of Subgradients and its Applications to Problems of Optimization," Hedermann, Berlin, 1981.

8. J. B. ROSEN, The gradient projection method for nonlinear programming. Part I. Linear constraints, *J. SIAM* **8** (1960), 181–217.
9. J. B. ROSEN, The gradient projection method for nonlinear programming. Part II. Nonlinear constraints, *J. SIAM* **9** (1961), 514–532.
10. P. RUBIN, Implementation of a subgradient projection algorithm, *Internat. J. Comput. Math.* **12** (1983), 321–328.
11. V. P. SREEDHARAN, A subgradient projection algorithm, *J. Approx. Theory* **35** (2) (1982), 111–126.
12. P. WOLFE, A method of conjugate subgradients for minimizing nondifferentiable functions, *Math. Programming Stud.* **3** (1975), 145–173.
13. P. WOLFE, On the convergence of gradient methods under constraints, *IBM J. Res. Develop.* **16** (1972), 407–411.