A Method of Feasible Directions Using Function Approximations, with Applications to Min Max Problems*

R. Klessig AND E. Polak

Department of Electrical Engineering and Computer Sciences,
and the Electronics Research Laboratory,
University of California, Berkeley, California 94720

Submitted by L. A. Zadeh

This paper presents a demonstrably convergent method of feasible directions for solving the problem \( \min \{ \phi(x) \mid g^i(x) \leq 0 \ i = 1, 2, \ldots, m \} \), which approximates, adaptively, both \( \phi(x) \) and \( \nabla \phi(x) \). These approximations are necessitated by the fact that in certain problems, such as when \( \phi(x) = \max \{ f(x, y) \mid y \in \Omega_y \} \), a precise evaluation of \( \phi(x) \) and \( \nabla \phi(x) \) is extremely costly. The adaptive procedure progressively refines the precision of the approximations as an optimum is approached and as a result should be much more efficient than fixed precision algorithms.

It is outlined how this new algorithm can be used for solving problems of the form \( \min_{y \in \Omega_y} \max_{x \in \Omega_x} f(x, y) \) under the assumption that \( \Omega_x = \{ x \mid g^j(x) \leq 0, j = 1, \ldots, s \} \subseteq \mathbb{R}^n \), \( \Omega_y = \{ y \mid \zeta^i(y) < 0, i = 1, \ldots, t \} \subseteq \mathbb{R}^m \), with \( f, g^j, \zeta^i \) continuously differentiable, \( f(x, \cdot) \) concave, \( \zeta^i \) convex for \( i = 1, \ldots, t \), and \( \Omega_x, \Omega_y \) compact.

1. Introduction

One of the major classes of algorithms for solving nonlinear programming problems of the form \( \min_{x \in \Omega} \{ \phi(x) \mid g(x) \leq 0 \} \) (with \( \phi: \mathbb{R}^n \to \mathbb{R}^1 \), \( g: \mathbb{R}^n \to \mathbb{R}^m \) continuously differentiable) is the class of methods of feasible directions [1-7]. All these algorithms have the following feature in common: To compute \( x_{i+1} \) from \( x_i \), one must compute both \( \phi(x_i) \) and \( \nabla \phi(x_i) \). Although usually this results in no difficulty, there are some cases where the need to compute \( \phi(x_i) \) (and \( \nabla \phi(x_i) \)) leads to severe complications. For example, suppose that

\[ \phi(x) = \max_{y \in \Omega_y} f(x, y). \]

Then, to compute \( \phi(x) \) we must bring in a subprocedure (probably also a method of feasible directions) which constructs a sequence \( \{ y_j \} \) such that...
f(x, y) \rightarrow \phi(x) as j \rightarrow \infty. Therefore, if viewed constructively, a method of feasible directions cannot be applied to such a problem, since we would have to compute an infinite sequence \( \{x_i\} \), each element of which is only obtainable as the limit point of an infinite sequence \( \{y_{ij}\}_{j=0}^{\infty} \). Even if one adopts a non-theoretical point of view, it is clear that the computation of adequate approximations to \( \phi(x) \) and to \( \nabla \phi(x) \) is bound to be extremely time consuming when

\[
\phi(x) = \max_{y \in \Delta} f(y, x).
\]

We shall show in this paper how one particular method of feasible directions (due to Polak [4]) can be modified so as to eliminate both the theoretical and practical difficulties indicated above. A similar treatment also appears to be possible for some of the other methods of feasible directions. To obtain our new algorithm, we need to extend a method for implementing theoretical algorithms discussed in [8–10]. This is done in the next section. After that we construct a method of feasible directions using function approximations, and, finally, we indicate how it applies to min–max problems.

2. A Model for Implementation

Let \( \mathcal{X} \) be a normed linear space and let \( T \) be a closed subset of \( \mathcal{X} \). Suppose that \( T \) contains a set \( \Delta \) of desirable points and that we wish to find an \( x \in \Delta \). Quite commonly, a theoretical algorithm for finding an \( x \in \Delta \) will make use of a search function \( A: T \rightarrow 2^T \) and of a stop rule (surrogate cost) function \( c: T \rightarrow \mathbb{R}^1 \) and will have the form below.

**Algorithm Model 2.1.**

1. **Step 0.** Select an \( x_0 \in T \) and set \( i = 0 \).
2. **Step 1.** Compute a \( y \in A(x_i) \).
3. **Step 2.** If \( c(y) \geq c(x_i) \), stop; else, set \( x_{i+1} = y \) and go to Step 3.
4. **Step 3.** Set \( i = i + 1 \) and go to Step 1.

When the function \( A(\cdot) \) and \( c(\cdot) \) appearing in (2.1) cannot be evaluated in a reasonable manner, one needs to approximate \( A(x) \) and \( c(x) \) somehow. In our algorithms, we shall use sequences \( \{A_j(\cdot)\}_{j=0}^{\infty} \) and \( \{C_j(\cdot)\}_{j=0}^{\infty} \) of approximating functions, where \( A_j: T \rightarrow 2^T \) and \( C_j: T \rightarrow \mathbb{R}^1 \) for \( j = 0, 1, 2, \ldots \). We shall assume that the functions \( c(\cdot) \), \( C_j(\cdot) \) and \( A_j(\cdot) \), and the sets \( T \) and \( \Delta \) have the following properties.

**Assumptions 2.2.**

(i) \( c(\cdot) \) is continuous on \( T \);
(ii) \( T \) is compact;
(iii) Given any \( x \in T \) satisfying \( x \notin \Delta \), there exists an \( \epsilon(x) > 0 \), a \( \delta(x) > 0 \) and an integer \( N(x) \geq 0 \) such that

\[
c_j(y) - c_j(x') \leq -\delta(x), \quad \forall y \in A_j(x'), \quad \forall x' \in B(x, \epsilon(x)),
\]
\[
\forall c_j(x') \in C_j(x'), \quad \forall c_j(y) \in C_j(y), \quad \forall j \geq N(x),
\] (2.3)

where

\[
B(x, \epsilon) \triangleq \{ x' \in T \mid \| x' - x \| \leq \epsilon \};
\] (2.4)

(iv) Given any integer \( j \geq 0 \), there exists a \( w_j > -\infty \) such that

\[
c_j(x) \geq w_j, \quad \forall c_j(x) \in C_j(x), \quad \forall x \in T.
\] (2.5)

(v) Given any \( \gamma > 0 \), there exist an integer \( M(\gamma) \geq 0 \) such that

\[
| c_j(x) - c(x) | \leq \gamma, \quad \forall c_j(x) \in C_j(x), \quad \forall j \geq M(\gamma), \quad \forall x \in T.
\] (2.6)

In terms of these new functions, algorithm 2.1 expands as follows.

**Algorithm Model 2.7.**

- **Step 0.** Select an \( x_0 \in T \); select parameters \( \epsilon_0 > 0 \), \( \alpha \in (0, 1) \), and an integer \( j_0 \geq 0 \). Set \( i = 0 \), \( j = j_0 \), \( q(0) = j_0 \), and \( \epsilon = \epsilon_0 \).

- **Step 1.** Compute a \( c_j(x_i) \in C_j(x_i) \).

- **Step 2.** Compute a \( y \in A_j(x_i) \) and a \( c_j(y) \in C_j(y) \).

- **Step 3.** If \( c_j(y) - c_j(x_i) > -\epsilon \), set \( j = j + 1 \), \( \epsilon = \alpha \epsilon \) and go to Step 1; else set \( x_{i+1} = y \), \( \epsilon_{i+1} = \epsilon \), \( q(i + 1) = j \) and go to Step 4.

- **Step 4.** Set \( i = i + 1 \) and go to Step 2.

**Comment 2.8.** The \( \epsilon \)-test in Step 3 above serves the purpose of ensuring that the integer \( j \) used at \( x_i \) was sufficiently large for the approximations \( A_j(x_i) \), \( C_j(x_i) \), \( C_j(y) \), to \( A(x_i) \), \( c(x_i) \), \( c(y) \), to be adequate. It is borrowed from a similar implementation of (2.1) given in (A.1.1) of [10].

**Comment 2.9.** The sequences \( \{q(i)\} \) and \( \{\epsilon_i\} \) are defined in (2.7) only because we shall need them later. Note that for \( i = 0, 1, 2, 3, \ldots \)

\[
\epsilon_i = \alpha^{q(i)} \epsilon_0, \quad (2.10)
\]
\[
x_{i+1} \in A_{q(i+1)}(x_i). \quad (2.11)
\]

The following lemmas will enable us to state the convergence properties of algorithm (2.7).

**Lemma 2.12.** Suppose that the algorithm 2.7 jams up at a point \( x_i \), cycling indefinitely between Steps 3 and 1. Then \( x_i \in \Delta \).
Proof. Suppose that the algorithm 2.7 jams up at $x_i$ and that $x_i \not\in \Delta$. Then by (2.2, iii) there exist an $\epsilon(x_i) > 0$, a $\delta(x_i) > 0$ and an integer $N(x_i) > 0$, such that

$$c_j(y) - c_j(x_i) \leq -\delta(x_i), \quad \forall y \in A_j(x_i), \quad \forall c_j(x_i) \in C_j(x_i),$$

$$\forall c_j(y) \in C_j(y), \quad \forall j \geq N(x_i).$$

(2.13)

Since the algorithm is cycling indefinitely between Steps 3 and 1, it must be constructing sequences $\{y_r\}_{r=0}^\infty$, $\{c_{q(i)+r}(x_i)\}_{r=0}^\infty$ and $\{c_{q(i)+r}(y_r)\}_{r=0}^\infty$, such that

$$y_r \in A_{q(i)+r}(x_i), \quad c_{q(i)+r}(x_i) \in C_{q(i)+r}(x_i),$$

$$c_{q(i)+r}(y_r) \in C_{q(i)+r}(y_r), \quad r = 0, 1, 2,...$$

(2.14)

and

$$c_{q(i)+r}(y_r) - c_{q(i)+r}(x_i) > -a^{q(i)+r}\epsilon_0 = -a^r \epsilon_i, \quad r = 0, 1, 2,...$$

(2.15)

However, there exists an integer $p > 0$ such that

$$a^{q(i)+p}\epsilon_0 \leq \delta(x_i); \quad q(i) + p \geq N(x_i).$$

(2.16)

Consequently, for $r \geq p$, (2.15) contradicts (2.13) and (2.16) and hence we conclude that we must have $x_i \in \Delta$. 

LEMMA 2.17. Consider the sequences $\{\epsilon_i\}$ and $\{q(i)\}$ generated by algorithm 2.7 while constructing a sequence $\{x_i\} \subset T$. If $\{x_i\}$ is infinite, then $q(i) \to \infty$ and $\epsilon_i \to 0$ as $i \to \infty$.

Proof. Suppose that $\{x_i\}$ is infinite. Then $\{\epsilon_i\}$ is an infinite, monotonically decreasing sequence bounded from below by zero. Consequently, $\epsilon_i \to \epsilon^* \geq 0$ for $i \to \infty$. Suppose that $\epsilon^* > 0$. We shall show that this leads to a contradiction.

Since $\epsilon_i \to \epsilon^*$ and $\epsilon^* > 0$, it follows from (2.10) that there exists an integer $N'$ such that for $i \geq N'$, $\epsilon_i = \epsilon_{i+1} = \cdots = \epsilon^*$ and $q(i) = q(i+1) = \cdots = q^*$. It now follows from the test in Step 2 of (2.7) that for $i \geq N'$,

$$c_{q(i+1)}(x_i) \in C_{q^*}(x_i) \quad \text{and} \quad c_{q(i+1)}(x_{i+1}) \in C_{q^*}(x_{i+1}).$$

We may therefore write

$$c_{q(i+1)}(x_{i+1}) - c_{q(i+1)}(x_i) = c_{q^*}(x_{i+1}) - c_{q^*}(x_i) \leq -\epsilon_{i+1}$$

$$= -\epsilon^*, \quad \forall i \geq N',$$

(2.18)

where

$$c_{q^*}(x_i) = c_{q(i+1)}(x_i) \quad \text{and} \quad c_{q^*}(x_{i+1}) = c_{q(i+1)}(x_{i+1}).$$
Therefore, we must have \( c_q(x_i) \to -\infty \) as \( i \to \infty \). But, by (2.5) \( c_q(x_i) \geq \omega_q > -\infty \), and hence we have a contradiction. Therefore \( \epsilon^* = 0 \).

Finally, since \( \epsilon_i \to 0 \) as \( i \to \infty \), it follows from (2.10) that \( q(i) \to \infty \) as \( i \to \infty \). \( \square \)

**Proposition 2.19.** Suppose the algorithm (2.7) constructs an infinite sequence \( \{x_i\}_{i=0}^\infty \). Let \( \Lambda \) denote the set of accumulation points of \( \{x_i\}_{i=0}^\infty \). Then, given any \( \gamma > 0 \), there exists an integer \( P(\gamma) \) such that

\[
\min\{|x_i - x^*| \mid x^* \in \Lambda\} \leq \gamma, \quad \forall i \geq P(\gamma). \tag{2.20}
\]

**Theorem 2.21.** Algorithm 2.7 will either jam up at a point \( x_i \), cycling indefinitely between Steps 3 and 1, in which case \( x_i \in \Delta \), or else, it will construct an infinite sequence \( \{x_i\} \) which has at least one accumulation point in \( \Delta \).

**Proof.** The first part of the theorem was established in Lemma 2.12. Hence, suppose that \( \{x_i\} \) is infinite. To obtain a contradiction, suppose that \( \Lambda \cap \Delta = \emptyset \), where \( \Lambda \) is the set of accumulation points of \( \{x_i\} \). Since \( T \) is compact, \( \Lambda \) is a nonempty compact set, and hence (because we have assumed that \( \Lambda \cap \Delta = \emptyset \)) it follows from (2.2, iii) that there exist an \( \epsilon_A > 0 \), a \( \delta_A > 0 \) and an integer \( N_A \geq 0 \) such that

\[
c_j(y) - c_j(x') \leq -\delta_A, \quad \forall y \in A_j(x'), \quad \forall x' \in B(x^*, \epsilon_A),

\forall c_j(y) \in C_j(y), \quad \forall c_j(x') \in C_j(x'), \quad \forall x^* \in A, \quad \forall j \geq N_A. \tag{2.22}
\]

Let \( P(\epsilon_A) \) be defined as in (2.19) (for \( \gamma = \epsilon_A \)). Then, since \( q(i) \to \infty \) as \( i \to \infty \) by Lemma (2.17) there exists an integer \( N_1 \geq P(\epsilon_A) \) such that \( q(i) \geq N_1 \) for all \( i \geq N_1 \), and hence

\[
c_{q(i+1)}(x_{i+1}) - c_{q(i+1)}(x_i) \leq -\delta_A, \quad \forall c_{q(i+1)}(x_i) \in C_{q(i+1)}(x_i),

\forall c_{q(i+1)}(x_{i+1}) \in C_{q(i+1)}(x_{i+1}), \quad \forall i \geq N_1. \tag{2.23}
\]

Now, from (2.2, v) [see (2.6)], we conclude that there exists an integer \( N_2 \geq N_1 \) such that

\[
|c_j(x_i) - c(x_i)| \leq \delta_A/4, \quad \forall c_j(x_i) \in C_j(x_i), \quad \forall i \geq N_2, \quad \forall j \geq q(i). \tag{2.24}
\]

Hence, since \( q(i + 1) \geq q(i) \), for \( i = 0, 1, 2,... \),

\[
c_{q(i)}(x_i) \geq c(x_i) - \delta_A/4 \geq c_{q(i+1)}(x_i) - \delta_A/2,

\forall c_{q(i)}(x_i) \in C_{q(i)}(x_i), \quad \forall c_{q(i+1)}(x_i) \in C_{q(i+1)}(x_i), \quad \forall i \geq N_2. \tag{2.25}
\]
Combining (2.25) with (2.23), we now get
\[
\begin{align*}
\forall c_{q(i)}(x_i) \in C_q(x_i), \quad \forall c_{q(i+1)}(x_{i+1}) \in C_q(x_{i+1}), \\
\forall c_{q(i+1)}(x_{i+1}) \in C_q(x_{i+1}), \quad \forall i \geq N_2,
\end{align*}
\]
and therefore we must have \( c_{q(i)}(x_i) \to -\infty \) as \( i \to \infty \), for any \( c_{q(i)}(x_i) \in C_q(x_i) \), \( i = 0, 1, \ldots \).

Now let \( K \subseteq \{0, 1, 2, \ldots \} \) be such that \( x_i \to x^* \in A \) as \( i \to \infty \), \( i \in K \). Then, by (2.2, v), and Lemma 2.17, there exists an integer \( N_3 \geq 0 \) such that
\[
| c_{q(i)}(x_i) - c(x_i) | \leq | c(x^*) | / 4,
\forall c_{q(i)}(x_i) \in C_q(x_i), \quad \forall i \geq N_3, \quad i \in K,
\]
and also, since \( c(\cdot) \) is continuous,
\[
| c(x_i) - c(x^*) | \leq | c(x^*) | / 4, \quad \forall i \geq N_3, \quad i \in K,
\]
where \( c(x^*) \geq -\infty \) because \( c(\cdot) \) is continuous on \( T \). Combining (2.27) and (2.28), we obtain
\[
\begin{align*}
\forall c_{q(i)}(x_i) \in C_q(x_i), \quad \forall i \geq N_3, \quad i \in K,
\end{align*}
\]
which contradicts our previous conclusion that \( c_{q(i)}(x_i) \to -\infty \) as \( i \to \infty \), for any \( c_{q(i)}(x_i) \in C_q(x_i) \), based on the hypothesis that \( A \cap \Delta = \emptyset \). Hence \( A \cap \Delta \neq \emptyset \) and we are done. \( \square \)

Theorem 2.21 states that when the sequence \( \{x_i\} \) is infinite, it must have at least one accumulation point in \( \Delta \), the set of desirable points. Clearly, if \( x_i \to x^* \) as \( i \to \infty \), \( x^* \in \Delta \). The reader may well wonder as to the value of algorithm 2.7 when the sequences it constructs have more than one accumulation point. Although, at present, we cannot make a general statement, we can assert that it is sometimes possible to add to an algorithm of the form of (2.7) a simple subprocedure which sifts out a subsequence, \( \forall \) of whose accumulation points are in \( \Delta \). In such a case, we obtain an algorithm of value. In particular, we shall see that the above assertion applies to the algorithm which we shall develop in the next section.

With these preliminaries out of the way, we shall now construct a new method of feasible directions, using function approximations.
3. A Method of Feasible Directions with Approximations

Consider the problem

$$\min \{\phi(x) \mid g(x) \leq 0\}, \quad (3.1)$$

where $\phi: \mathbb{R}^n \to \mathbb{R}$ and $g: \mathbb{R}^n \to \mathbb{R}^m$ are continuously differentiable functions.

Let $\Omega_x \subseteq \mathbb{R}^n$ be defined by

$$\Omega_x = \{x \mid g(x) \leq 0\}. \quad (3.2)$$

Now, for any $x \in \Omega_x$ and for any $\epsilon \geq 0$, let the index set $I_x(x, \epsilon) \subseteq \{1, 2, \ldots, m\}$ be defined by

$$I_x(x, \epsilon) = \{q \in \{1, 2, \ldots, m\} \mid g_q(x) \geq -\epsilon\}, \quad (3.3)$$

let $S \subseteq \mathbb{R}^n$ be defined by

$$S = \{h \in \mathbb{R}^n \mid \|h\|_{\infty} \leq 1\}, \quad (3.4)$$

and let $\theta: \Omega_x \times \mathbb{R}^+ \to \mathbb{R}$ be defined by

$$\theta(x, \epsilon) = \min_{h \in S} \max_{q \in I_x(x, \epsilon)} \langle \nabla \phi(x), h \rangle; \langle \nabla g_q(x), h \rangle, q \in I_x(x, \epsilon) \rangle.$$

Note that $\theta(x, \epsilon) \leq 0$ for all $x \in \Omega_x$, for all $\epsilon \geq 0$. We can now state a well-known necessary condition of optimality for (3.1).

**Proposition 3.5.** Suppose that $\bar{x} \in \Omega_x$ solves (3.1), i.e.,

$$\phi(\bar{x}) = \min \{\phi(x) \mid x \in \Omega_x\}.$$

Then for every $\epsilon \geq 0$, $\theta(\bar{x}, \epsilon) = 0$. □

Now suppose that to compute $\phi(x)$ and $\nabla \phi(x)$ we must use a subprocedure which constructs two sequences $\{\phi_j(x)\}_{j=0}^{\infty}$, $\{\nabla \phi_j(x)\}_{j=0}^{\infty}$, such that $\phi_j(x) \to \phi(x)$ and $\nabla \phi_j(x) \to \nabla \phi(x)$ as $j \to \infty$. In constructing an algorithm which truncates these sequences we shall need the following hypotheses to hold [cf. (2.2)].

**Assumptions 3.6.**

(i) The set $\Omega_x$ in (3.2) is compact.

(ii) For $j = 0, 1, 2, \ldots$, $\Phi_j: \mathbb{R}^n \to \mathbb{R}$, $\nabla \Phi_j: \mathbb{R}^n \to \mathbb{R}^n$ are functions such that given any $\gamma > 0$ there exists an integer $M(\gamma) \geq 0$ such that

$$|\phi_j(x) - \phi(x)| \leq \gamma, \quad \forall x \in \Omega_x, \quad \forall \phi_j(x) \in \Phi_j(x), \quad \forall j \geq M(\gamma), \quad (3.7)$$

$$\|
abla \phi_j(x) - \nabla \phi(x)\| \leq \gamma, \quad \forall x \in \Omega_x, \quad \forall \nabla \phi_j(x) \in \nabla \Phi_j(x), \quad \forall j \geq M(\gamma). \quad (3.8)$$
(iii) Given any integer \( j \geq 0 \), there exists a \( w_j > -\infty \) such that
\[
\phi_j(x) \geq w_j, \quad \forall \phi_j(x) \in \Phi_j(x), \quad \forall x \in \Omega_x. \tag{3.9}
\]

**Definition 3.10.** We define \( \dot{\theta}: \Omega_x \times \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^1 \) and \( H: \Omega_x \times \mathbb{R}^n \times \mathbb{R}^+ \rightarrow 2^S \) as
\[
\dot{\theta}(x, u, \epsilon) = \min_{h \in S} \max\{\langle u, h \rangle; \langle \nabla g^q(x), h \rangle, q \in I_x(x, \epsilon)\}; \tag{3.11}
\]
and
\[
H(x, u, \epsilon) = \{h \in S | \dot{\theta}(x, u, \epsilon) = \max\{\langle u, h \rangle; \langle \nabla g^q(x), h \rangle, q \in I_x(x, \epsilon)\}\}. \tag{3.12}
\]

**Note.** \( \dot{\theta}(x, u, \epsilon) \) and a vector \( h \in H(x, u, \epsilon) \) can be computed by solving a linear programming problem (see Section 4.3 in [10]).

We shall now modify algorithm 4.3.26 in [10] so as to make it correspond to algorithm model 2.7, and, in addition, we shall add a sifting subprocedure to extract a subsequence \( \{x_i\}_{i \in K} \) all of whose accumulation points \( x^* \) will be shown to satisfy \( \dot{\theta}(x^*, 0) = 0 \). For the sake of convenience, we break up the following algorithm into two subprocedures.

**Extension of the Polak Method of Feasible Directions 3.13.**

**Subprocedure I. Method of feasible directions with approximations.**

Begin:

**Step 0.** Select parameters \( \epsilon_0^1 > 0, \epsilon_0^2 > 0, \epsilon_0^3 > 0, \lambda_{\min} \in (0, 1), \alpha \in (0, 1), \alpha_2 \in (0, 1), \alpha_3 \in (0, 1) \) and an integer \( j_0 \geq 0 \); compute an \( x_0 \in \Omega_x; \)
set \( i = 0, j = j_0, k = 0, r_2^1 = r_3^1 = 0, \epsilon^2 = \epsilon^3 = \epsilon^3 - \epsilon^3 \).

**Step 1.** Set \( \epsilon^1 = \epsilon^0 \).

**Step 2.** Compute a \( \phi_j(x_i) \in \Phi_j(x_i) \) and a \( \nabla_j \phi(x_i) \in \nabla_j \Phi(x_i) \).

**Step 3.** Compute \( \dot{\theta}(x_i, \nabla_j \phi(x_i), \epsilon^1) \) and a vector
\[
h(x_i, \nabla_j \phi(x_i), \epsilon^1) \in H(x_i, \nabla_j \phi(x_i), \epsilon^1).
\]

**Step 4.** If \( \dot{\theta}(x_i, \nabla_j \phi(x_i), \epsilon^1) = 0 \), compute \( \dot{\theta}(x_i, \nabla_j \phi(x_i), 0) \) and go to Step 5; else, go to Step 6.

**Step 5.** If \( \dot{\theta}(x_i, \nabla_j \phi(x_i), 0) = 0 \), set \( x' = x_i \), set \( \phi_j(x') = \phi_j(x_i) \) and go to Step 14; else set \( \epsilon^1 = \alpha x \epsilon^1 \) and go to Step 3.

**Step 6.** If \( \dot{\theta}(x_i, \nabla_j \phi(x_i), \epsilon^1) \leq -\epsilon^1 \), go to Step 7; else, set \( \epsilon^1 = \alpha x \epsilon^1 \) and go to Step 3.

1 Note that a \( \nabla_j \phi(x_i) \in \nabla_j \Phi(x_i) \) may already be available because of its computation in Step 17 and hence need not be recomputed.
Step 7. Set $\lambda = 1$.

Step 8. Compute $G = g(x_i + \lambda h(x_i, \nabla_j \phi(x_i), \epsilon^1))$.

Step 9. If $G \leq 0$, go to Step 10; else, set $\lambda = \lambda/2$ and go to Step 8.

Step 10. Compute

$$D = \phi_j(x_i + \lambda h(x_i, \nabla_j \phi(x_i), \epsilon^1)) - \phi_j(x_i) - \frac{\lambda}{2} \langle \nabla_j \phi(x_i), h(x_i, \nabla_j \phi(x_i), \epsilon^1) \rangle.$$

Step 11. Compute

$$a = \phi_j(x_i + \lambda h(x_i, \nabla_j \phi(x_i), \epsilon^1)) - \phi_j(x_i).$$

Step 12. If $D > 0$ go to Step 13; else set $x' = x_i + \lambda h(x_i, \nabla_j \phi(x_i), \epsilon^1)$, set $\phi_j(x') = \phi_j(x_i + \lambda h(x_i, \nabla_j \phi(x_i), \epsilon^1))$ and go to Step 14.

Step 13. If $\lambda \geq \lambda_{\text{min}}/2$, set $\lambda = \lambda/2$ and go to Step 8; else set $x' = x_i$, set $\phi_j(x') = \phi_j(x_i)$ and go to Step 14.

Step 14. If $\phi_j(x') - \phi_j(x_i) \leq -\epsilon^2$, go to Step 15; else, set $j = j + 1$, set $\epsilon^2 = \alpha_2 \epsilon^2$ and go to Step 1.

Step 15. Set $x_{i+1} = x'$, set $q(i + 1) = j$, $\epsilon_{i+1}^2 = \epsilon^2$.

Comment. Do not compute $q(i + 1)$ and $\epsilon_{i+1}^2$. These quantities are introduced only for the convenience of the proofs to follow.

End:

Step 16. Set $i = i + 1$.

Subprocedure II. Sieve

Begin:

Step 17. Compute $\nabla_j \phi(x') \in \nabla_j \phi(x')$.

Step 18. Compute $\tilde{\theta}(x', \nabla_j \phi(x'), \epsilon^3)$.

Step 19. If $\tilde{\theta}(x', \nabla_j \phi(x'), \epsilon^3) \geq -\epsilon^3$, go to Step 20; else, go to Step 1.

Step 20. Set $x_k = x'$, set $\epsilon_k^3 = \epsilon^3$, set $p(k) = q(i)$.

Comment. Do not compute $\epsilon_k^3$ and $p(k)$. These quantities are introduced only for the convenience of the proofs to follow.

End:

Step 21. Set $\epsilon^3 = \alpha_2 \epsilon^3$, set $k = k + 1$, and go to Step 1.

We shall now show that Subprocedure I (Steps 0-16) of algorithm 3.13 corresponds to the model 2.7, with the functions $A_j(\cdot)$ being defined by the Steps 1-13 of (3.13), and with $\Phi_j(\cdot)$, $\epsilon^2$ and $\alpha_2$ in (3.13) taking the place of $C_j(\cdot)$, $\epsilon$ and $\alpha$ in (2.7). The additional parameters in Step 0 of (3.13) are used either to define the $A_j(\cdot)$ or in the sifting Subprocedure II, defined by Steps 17-21 of (3.13).
First, we must show that the maps $A(x)$ are well defined by Steps 1–13 of (3.13), i.e., that Subprocedure I of (3.13) cannot jam up before reaching Step 14. We shall do this in the following lemmas.

**Proposition 3.14.** For any $x \in \Omega_x$, there exists a $\rho(x) > 0$ such that

$$I_\epsilon(x, \epsilon) = I_\epsilon(x, 0), \quad \forall \epsilon \in [0, \rho(x)]$$

(3.15)

$$\theta(x, u, \epsilon) = \theta(x, u, 0), \quad \forall \epsilon \in [0, \rho(x)], \quad \forall u \in \mathbb{R}^n.$$  

(3.16)

**Lemma 3.17.** Subprocedure I of algorithm 3.13 cannot cycle indefinitely in the loop defined by Steps 3–6.

**Proof.** Suppose that $\theta(x_i, u, 0) = 0$ for some $u \in \mathbb{R}^n$. Then, since $I_\epsilon(x_i, 0) \subseteq I_\epsilon(x_i, \epsilon)$ for all $\epsilon > 0$, we must have

$$0 = \theta(x_i, u, 0) \leq \theta(x_i, u, \epsilon) \leq 0,$$

(3.18)

and hence $\theta(x_i, u, \epsilon) = 0$. So that when $\theta(x_i, \nabla \phi(x_i), 0) = 0$, algorithm 3.13 proceeds from Step 3 to Step 4 to Step 5 and hence to Step 14. Now suppose that $\theta(x_i, \nabla \phi(x_i), 0) < 0$. It then follows from Proposition 3.14 that when $\epsilon$ has become reduced to the point where

$$\epsilon \leq \min\{\rho(x_i), -\theta(x_i, \nabla \phi(x_i), 0)\},$$

which is a finite process, we shall have $\theta(x_i, \nabla \phi(x_i), \epsilon) \leq -\epsilon$ and algorithm will proceed from Step 6 to Step 7. Consequently, algorithm (3.13) cannot jam up in the loop defined by Steps 3–6. □

**Proposition 3.19.** Suppose that $x_i \in \Omega_x$, $u \in \mathbb{R}^n$, $\epsilon > 0$, and $j \geq 0$ are such that $\theta(x_i, u, \epsilon) \leq -\epsilon$. Then there exists a $\lambda(x_i, \epsilon) > 0$ such that

$$g(x_i + \lambda h) \leq 0, \quad \forall \lambda \in [0, \lambda(x_i, \epsilon)], \quad \forall h \in \mathcal{H}(x_i, u, \epsilon).$$

(3.20)

**Proposition 3.21.** Subprocedure I of algorithm 3.13 cannot cycle indefinitely in the loop defined by Steps 8 and 9. □

We have thus established that Steps 1–13 of algorithm 3.13 define a map $A_A: \Omega_x \rightarrow 2^{\mathbb{R}^n} (x' \in A_A(x))$ with $x'$ defined in Step 5, in Step 12, or in Step 13, as may be appropriate. If we let $\mathbb{R}^n, \Omega_x$, Steps 1–13, $\Phi(x), \phi(x)$, and $\{x \in \Omega_x | \theta(x, 0) = 0\}$ correspond to $\mathcal{A}, T, A_A(x), C_A(x), \epsilon$, and $\Delta$, respectively, we see that Steps 0–16 of algorithm 3.13 correspond to algorithm model 2.7. Thus, to conclude that Theorem 2.21 applies to Subprocedure I of algorithm 3.13, we must show that the assumptions 2.2 (i)–(v) are satisfied. It follows directly from (3.1) and (3.6) that the assumptions (2.2, i), (2.2, ii),...
METHOD OF FEASIBLE DIRECTIONS

593

(2.2, iv) and (2.2, v) are satisfied. It remains to show that assumption (2.2, iii) is satisfied. This will require several lemmas.

**Definition.** For any \( x \in \Omega_x \) and any \( \epsilon > 0 \), we define

\[
B_\epsilon(x) \triangleq \{ x' \in \Omega_x \mid \| x' - x \| \leq \epsilon \}.
\]  

(3.22)

**Proposition 3.23.** Given any \( x \in \Omega_x \) and any \( \gamma > 0 \), there exists a \( \rho(x, \gamma) > 0 \) such that

\[
\theta(x', \epsilon) \leq \theta(x, 0) + \gamma, \quad \forall x' \in B_\epsilon(x, \rho(x, \gamma)), \quad \forall \epsilon \in [0, \rho(x, \gamma)].
\]  

(3.24)

**Corollary 3.25.** Given any \( x \in \Omega_x \) and any \( \gamma > 0 \), there exist a \( \rho(x, \gamma) > 0 \) and an integer \( M'(\gamma) \) such that

\[
\theta(x', u, \epsilon) \leq \theta(x, 0) + \gamma, \quad \forall x' \in B_\epsilon(x, \rho(x, \gamma)), \quad \forall u \in \nabla f(x'), \quad \forall j \geq M'(\gamma), \quad \forall \epsilon \in [0, \rho(x, \gamma)].
\]  

(3.26)

**Proof.** Since \( S \) is compact, it follows from (3.8) that there exists an integer \( M'(\gamma) \) such that

\[
|\langle u, h \rangle - \langle \nabla f(x'), h \rangle| \leq \gamma/2,
\]  

(3.27)

\[
\forall x' \in \Omega_x, \quad \forall h \in S, \quad \forall u \in \nabla f(x'), \quad \forall j \geq M'(\gamma).
\]

Hence,

\[
\tilde{\theta}(x', u, \epsilon) \leq \theta(x', \epsilon) + \gamma/2,
\]  

(3.28)

\[
\forall x' \in \Omega_x, \quad \forall u \in \nabla f(x'), \quad \forall \epsilon > 0, \quad \forall j \geq M'(\gamma).
\]

Finally, utilizing (3.28) and (3.24), where we replace \( \gamma \) by \( \gamma/2 \), we obtain (3.26). \( \square \)

**Lemma 3.29.** Suppose that \( x \in \Omega_x \) satisfies \( \theta(x, 0) < 0 \). Then there exists an \( \epsilon(x) > 0 \) and an integer \( N(x) \geq 0 \) such that for all \( x_i \in B_\epsilon(x, \epsilon(x)) \) and for all integers \( j \geq N(x) \), algorithm 3.13 satisfies \( \theta(x_i, \nabla f(x_i), \epsilon^1) \leq - \epsilon^1 \) in Step 6, and reaches Step 7, with \( \epsilon^1 \) satisfying

\[
\epsilon^1 > \epsilon(x).
\]  

(3.30)

**Proof.** Suppose that \( x \in \Omega_x \) is such that \( \theta(x, 0) < 0 \). Then, by Corollary 3.25, there exist a \( \rho(x) > 0 \) and an integer \( N(x) \geq 0 \) such that

\[
\tilde{\theta}(x_i, u, \epsilon^1) \leq \frac{1}{2} \theta(x, 0) < 0, \quad \forall x_i \in B_\epsilon(x, \rho(x)), \quad \forall u \in \nabla f(x_i), \quad \forall \epsilon^1 \in [0, \rho(x)], \quad \forall j \geq N(x).
\]  

(3.31)
Let $\bar{e}(x) = \min\{\rho(x), -\frac{1}{2} \theta(x, 0)\}$. Then, by (3.31),
\[
\bar{e}(x_i, u, e^1) \leq -e^1, \quad \forall x_i \in B_2(x, e(x)), \forall u \in \nabla_j \Phi(x_i), \forall e^1 \in [0, \bar{e}(x)], \forall j \geq N(x).
\] (3.32)

Since Step 6 of algorithm 3.13 requires that (3.32) be satisfied with $e^1 = \alpha_i e_0^1$, for some integer $p \geq 0$, we see that if we set $e(x) = \alpha_i \bar{e}(x)$, then (3.32) can always be satisfied with $e^1 = \alpha_i e_0^1 > e(x)$, for some integer $p$, and hence we are done. □

**Corollary 3.33.** Suppose that $x \in \Omega_x$ satisfies $\theta(x, 0) < 0$, and suppose that $e(x) > 0$ and the integer $N(x) \geq 0$ are such that the conclusion of Lemma 3.29 holds. Then there exists an integer $l(x) \geq 0$ such that
\[
g(x_i + (\frac{1}{2})^p h) \leq 0, \quad \forall x_i \in B_2(x, e(x)), \forall h \in \hat{H}(x_i, u, e^1(x_i, u)), \forall u \in \nabla_j \Phi(x_i), \forall j \geq N(x), \quad p = l(x), l(x) + 1, l(x) + 2, \ldots
\] (3.34)

where $e^1(x_i, u)$ is the value of $e^1$ at which algorithm 3.13 passes from Step 6 to Step 7, for the computed $u \in \nabla_j \Phi(x_i)$.

**Proof.** By Lemma 3.29, for $j \geq N(x)$ and $x_i \in B_2(x, e(x))$,
\[
e^1(x_i, u) > e(x) > 0, \quad \forall u \in \nabla_j \Phi(x_i).
\]

Let $x_i \in B(x, e(x))$ and $u \in \nabla_j \Phi(x_i)$ be arbitrary. Then since the algorithm 3.13 ensures that $\theta(x_i, u, e^1(x_i, u)) \leq -e^1(x_i, u)$, and $e^1(x_i, u) > e(x)$, we must have either $\langle \nabla g^q(x_i), h \rangle \leq -e(x)$ for all $h \in \hat{H}(x_i, u, e^1(x_i, u))$, or else $g^q(x_i) \leq -e(x)$, $q \in \{1, 2, \ldots, m\}$. Since $B_2(x, e(x))$ and $S$ are both compact and the functions $g^q(\cdot)$ are continuously differentiable, the existence of an integer $l(x) \geq 0$ for which (3.34) holds now follows directly (cf. (3.19)). □

**Theorem 3.35.** Suppose that $x \in \Omega_x$ satisfies $\theta(x, 0) < 0$. Then there exist an $e(x) > 0$, an integer $N'(x) \geq 0$ and an integer $l'(x) \geq 0$ such that
\[
\phi_j(x_i + (\frac{1}{2})^{l'(x)} h) - \phi_j(x_i) - (\frac{1}{2})^{l'(x)+1} \langle \nabla_j \Phi(x_i), h \rangle \leq 0, \tag{3.36}
\]
\[
g(x_i + (\frac{1}{2})^{l'(x)} h) \leq 0, \tag{3.37}
\]
\[
\forall x_i \in B_2(x, e(x)), \forall \phi_j(x_i + (\frac{1}{2})^{l'(x)} h) \in \Phi_j(x_i + (\frac{1}{2})^{l'(x)} h), \forall \phi_j(x_i) \in \Phi_j(x_i), \forall h \in \hat{H}(x_i, u, e^1(x_i, u)), \forall u \in \nabla_j \Phi(x_i), \forall j \geq N'(x),
\]

where $e^1(x_i, u)$ is the value of $e^1$ at which the test $\theta(x_i, \nabla_j \Phi(x_i), e^1) \leq -e^1$ is satisfied in Step 6 of algorithm 3.13.
Proof. Suppose $\theta(x, 0) < 0$. Then by Lemma (3.29), there exist an $\epsilon(x) > 0$ and an $N(x) \geq 0$ such that (3.30) holds. Now by the mean value theorem, for any $h \in S$, $u \in \mathbb{R}^n$, and $\lambda \geq 0$,

$$\phi(x_i + \lambda h) - \phi(x_i) - \frac{1}{2} \lambda \langle u, h \rangle = \lambda \langle \nabla\phi(x_i + \lambda h), h \rangle - \frac{1}{2} \langle \nabla u, h \rangle \rangle,$$

(3.38)

where $\bar{\lambda} \in [0, \lambda]$.

Since $B_2(x, \epsilon(x))$ and $S$ are compact, it follows from (3.6, ii) that there exist an integer $N''(x) \geq N(x)$ and a $\lambda'(x) > 0$ such that

$$\langle \nabla\phi(x_i + \lambda h), h \rangle \leq \langle u, h \rangle + \frac{1}{2} \epsilon(x),$$

$$\forall x_i \in B_2(x, \epsilon(x)), \ \forall u \in \nabla \Phi(x_i), \ \forall h \in H(x_i, u, \epsilon(x), u), \ \forall \lambda \in [0, \lambda'(x)], \ \forall j \geq N''(x).$$

(3.39)

Since for $u \in \nabla \Phi(x_i)$,

$$\langle u, h \rangle \leq -\epsilon(x),$$

for all $h \in H(x_i, u, \epsilon(x), u)$,

(3.38) and (3.39) imply that

$$\phi(x_i + \lambda h) - \phi(x_i) - \frac{1}{2} \lambda \langle u, h \rangle \leq \lambda \langle u, h \rangle + \frac{1}{2} \epsilon(x) - \frac{1}{2} \langle u, h \rangle \rangle \leq -\frac{1}{2} \epsilon(x)$$

$$\forall x_i \in B_2(x, \epsilon(x)), \ \forall h \in H(x_i, u, \epsilon(x), u),$$

$$\forall u \in \nabla \Phi(x_i), \ \forall \lambda \in [0, \lambda'(x)], \ \forall j \geq N''(x).$$

(3.40)

Now, because of the manner in which $\epsilon(x) > 0$ and $N''(x) \geq 0$ were chosen, it follows from Corollary 3.33 that there exists an integer $l(x) \geq 0$ such that

$$g(x_i + \frac{1}{2} \rho h) \leq 0, \ \ \forall x_i \in B_2(x_i, \epsilon(x)),$$

$$\forall h \in H(x_i, u, \epsilon(x), u),$$

$$\forall u \in \nabla \Phi(x_i), \ \forall j \geq N''(x), \ \forall \rho \geq l(x),$$

(3.41)

where $p$ is assumed to be an integer.

Let $l'(x)$ be the smallest integer satisfying $(\frac{1}{2})^{l'(x)} \leq \lambda'(x)$ and $l'(x) \geq l(x)$. Then, by (3.6, ii), there exists an integer $N'(x) \geq N''(x)$, such that

$$|\phi_j(x) - \phi(x)| \leq \left(\frac{1}{2}\right)^{l'(x)} \epsilon(x)$$
for all $x \in \Omega_x$, for all $\phi_j(x) \in \Phi_j(x)$, for all $j \geq N'(x)$, and hence, from (3.40),
for $\lambda = (\frac{1}{2})^{l'(x)}$, we obtain

$$
\phi_j(x_i + (\frac{1}{2})^{l'(x)} h) - \phi_j(x_i) - (\frac{1}{2})^{l'(x) + 1} \langle u, h \rangle
\leq \left( \frac{1}{2} \right)^{l'(x)} \left[ \frac{\epsilon(x)}{4} - \frac{3}{8} \epsilon(x) \right] = - \left( \frac{1}{2} \right)^{l'(x)} \frac{\epsilon(x)}{8} < 0.
$$

(3.42)


\begin{align*}
\forall u \in \nabla_j \Phi(x_i), & \quad \forall \phi_j(x_i) \in \Phi_j(x_i), \\
\forall \phi_j(x_i + (\frac{1}{2})^{l'(x)} h) \in \Phi_j(x_i + (\frac{1}{2})^{l'(x)} h), & \\
\forall x_i \in B(x, \epsilon(x)), & \quad \forall h \in \tilde{H}(x_i, u, \epsilon_i(x_i, u)), \quad \forall j \geq N'(x).
\end{align*}

Hence (3.36) holds. Since $l'(x) \geq l(x)$, it follows from (3.41) that (3.37) also holds, and so we are done. □

**Corollary 3.43.** Suppose that $x \in \Omega_x$ satisfies $\theta(x, 0) < 0$. Then there exists an $\epsilon(x) > 0$, a $\delta(x) > 0$ and an integer $N'(x) > 0$ such that

$$
M_{x_{i+1}} - M_{x_i} < - \delta(x), \quad \forall x_i \in \Omega_x, \quad \forall j \geq N'(x),
$$

and for all $x_{i+1} = x_i + \lambda h, \quad h \in \tilde{H}(x_i, u, \epsilon_i(x_i, u)) = \epsilon_i(x_i, u)$, which algorithm 3.13 can construct from the given $x_i$, where $\epsilon_i(x_i, u)$ is the value of $\epsilon_i$ for which the test $\theta(x_i, u, \epsilon_i(x_i, u)) < - \epsilon_i(x_i, u)$ is satisfied in Step 6.

**Proof.** Let $\epsilon(x) > 0$, $N'(x) \geq N(x) \geq 0$ and $l'(x)$ be such that (3.39), (3.36) and (3.37) hold. Then, clearly, for all $x_i \in B(x, \epsilon(x))$, for all $j \geq N'(x)$, algorithm 3.13 will construct $x_{i+1} = x_i + \lambda h$, with $h \in \tilde{H}(x_i, u, \epsilon_i(x_i, u))$ and

$$
\lambda \triangleq (\frac{1}{2})^{l'(x)}, \quad u \in \nabla_j \Phi(x_i).
$$

Consequently, we must have

$$
\phi_j(x_{i+1}) - \phi_j(x_i) \leq (\frac{1}{2})^{l'(x)} \langle u, h \rangle \leq - (\frac{1}{2})^{l'(x)} \epsilon_i(x_i, u)
\leq - (\frac{1}{2})^{l'(x)} \epsilon(x) \triangleq - \delta(x),
$$

(3.45)

\begin{align*}
\forall x_i \in B(x, \epsilon(x)), & \quad \forall u \in \nabla_j \Phi(x_i), \quad \forall \phi_j(x_i) \in \Phi_j(x_i), \\
\forall \phi_j(x_{i+1}) \in \Phi_j(x_{i+1}), & \quad \forall j \geq N'(x),
\end{align*}

and hence we are done. □

**Theorem 3.46.** Subprocedure I of algorithm 3.13 satisfies the assumptions (2.2,i−v).

**Proof.** That the assumptions (2.2, i), (2.2, ii), (2.2, iv) and (2.2, v) are satisfied follows directly from (3.6) and the correspondence previously
specified. That assumption (2.2, iii) is satisfied follows from Corollary 3.43 and the specified correspondences.

In view of Theorems 3.46 and 2.21 and the correspondences, the following is obvious.

**Corollary 3.47.** Subprocedure I of algorithm 3.13 will either jam up at a point \( x_i \), cycling indefinitely in the loop defined by Steps 1–14, in which case \( x_i \) satisfies the optimality condition \( \theta(x_i, 0) = 0 \), or else it will construct an infinite sequence \( \{x_i\} \) which has at least one accumulation point \( x^* \) satisfying \( \theta(x^*, 0) = 0 \).

We shall now establish the convergence properties of the sequence \( \{z_k\} \) sieved out by Subprocedure II of algorithm 3.13 from an infinite sequence \( \{x_i\} \) constructed by Subprocedure I of (3.13). For this purpose we shall need the following propositions, the proofs of which we omit, either because they are obvious or because they can easily be established by following the reasoning used for analogous results in the first part of this section.

**Definition.** For any \( x \in \Omega_z \) and for any \( \epsilon > 0 \), we define

(i) the index set \( I(x, \epsilon) \) by

\[
I(x, \epsilon) = \{q \in \{1, 2, \ldots, m\} | g^q(x) > - \epsilon\},
\]

(ii) the function \( \overline{\theta} : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^1 \) by

\[
\overline{\theta}(x, \epsilon) = \min_{h \in S} \max \{\langle \nabla \phi(x), h \rangle; \langle \nabla g^q(x), h \rangle, q \in I(x, \epsilon)\},
\]

(iii) the function \( \overline{\theta} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^1 \) by

\[
\overline{\theta}(x, u, \epsilon) = \min_{h \in S} \max \{\langle u, h \rangle; \langle \nabla g^q(x), h \rangle, q \in I(x, \epsilon)\},
\]

(iv) the function \( \overline{H} : \mathbb{R}^n \times \mathbb{R}^+ \to 2^S \) by

\[
\overline{H}(x, u, \epsilon) = \{h \in S | \overline{\theta}(x, u, \epsilon) = \max \{\langle u, h \rangle; \langle \nabla g^q(x), h \rangle, q \in I(x, \epsilon)\}\}.
\]

**Proposition 3.52.** For every \( x \in \Omega_z \) and every \( \epsilon \geq 0 \),

\[
I(x, \epsilon) \subset I_2(x, \epsilon),
\]

\[
\overline{\theta}(x, \epsilon) \leq \theta(x, \epsilon),
\]

\[
\overline{\theta}(x, u, \epsilon) \leq \overline{\theta}(x, u, \epsilon),
\]

\[
\forall u \in \nabla_j \Phi(x), \quad j = 0, 1, 2, \ldots.
\]
Proposition 3.56. Given any \( x \in \Omega_x \) and \( \epsilon \geq 0 \), there exists a \( \rho(x, \epsilon) > 0 \) such that
\[
\bar{I}(x', \epsilon) \supset \bar{I}(x, \epsilon), \quad \forall x' \in B_\rho(x, \rho(x, \epsilon)).
\] (3.57)

Proposition 3.58. Given any \( x \in \Omega_x \), any \( \epsilon > 0 \) and any \( \gamma > 0 \), there exists a \( \rho(x, \epsilon) > 0 \) such that
\[
\theta(x', \epsilon) \geq \theta(x, \epsilon) - \gamma, \quad \forall x' \in B_\rho(x, \rho(x, \epsilon)).
\] (3.59)

Corollary 3.60. Given any \( x \in \Omega_x \), any \( \epsilon > 0 \) and any \( \gamma > 0 \), there exists a \( \sigma(x, \epsilon) > 0 \) and an integer \( j(x, \epsilon) \geq 0 \) such that
\[
\sigma(x', u, \epsilon) \geq \sigma(x, \epsilon) - \gamma,
\]
\[\forall x' \in B_\sigma(x, \sigma(x, \epsilon)), \quad \forall u \in \nabla \mathcal{F}(x'), \quad \forall j \geq j(x, \epsilon).\] (3.61)

Lemma 3.62. Suppose that the sequence \( \{x_i\} \) generated by Subprocedure I of algorithm 3.13 is infinite. Then the sequence \( \{x_k\} \) sieved out by Subprocedure II of algorithm 3.13 is also infinite.

Proof. We see that according to Steps 19 and 20 of (3.13), Subprocedure II sets \( x_k = x_i \) and \( k = k + 1 \), whenever \( \bar{\theta}(x_i, u, \varepsilon^3) \geq -\varepsilon^3 \), with \( u \in \nabla \mathcal{F}(x_i) \), where \( \varepsilon^3 = \sigma_k \varepsilon_0^2 \). Consequently, to establish the lemma, it suffices to show that for any \( \varepsilon^3 > 0 \) there exists a subsequence \( \{x_i\}_{i \in K(\varepsilon^3)} \subseteq \{x_i\} \) such that
\[
\bar{\theta}(x_i, u_i, \varepsilon^3) \geq -\varepsilon^3, \quad \forall u_i \in \nabla \mathcal{F}(x_i), \quad \forall i \in K(\varepsilon^3).\] (3.63)

We recall that according to Lemma 2.17, we must have \( q(i) \to \infty \) as \( i \to \infty \), since \( \{x_i\} \) is infinite. Next, according to Corollary 3.47 there exists a subsequence \( \{x_i\}_{i \in K_1} \) such that \( x_i \to x^* \) as \( i \to \infty \), \( i \in K_1 \), and \( \theta(x^*, 0) = 0 \). Since \( \bar{I}(x^*, \epsilon) \supset I(x^*, 0) \) for all \( \epsilon > 0 \), we conclude that
\[
0 \geq \bar{\theta}(x^*, \epsilon) \geq \theta(x^*, 0) = 0, \quad \forall \epsilon > 0,
\] (3.64)
i.e., \( \bar{\theta}(x^*, \epsilon) = 0 \) for all \( \epsilon > 0 \). Let \( \varepsilon^3 > 0 \) be arbitrary. Since \( x_i \to x^* \), as \( i \to \infty \) for \( i \in K_1 \), it follows from Corollary 3.60 (and because of the fact that \( I(x, \epsilon) \supset \bar{I}(x, \epsilon), \forall x \in \Omega_x, \forall \epsilon > 0 \)) that there exists an integer \( j(x^*, \varepsilon^3) \) such that
\[
\bar{\theta}(x_i, u_i, \varepsilon^3) \geq \bar{\theta}(x_i, u_i, \varepsilon^3) \geq \bar{\theta}(x^*, \varepsilon^3) - \varepsilon^3 = -\varepsilon^3
\]
\[\forall u_i \in \nabla \mathcal{F}(x_i), \quad \forall i \geq j(x^*, \varepsilon^3) \quad \text{and} \quad i \in K_1.\] (3.65)

Let \( K(\varepsilon^3) = \{i \in K_1 \mid i \geq j(x^*, \varepsilon^3)\} \). Then we see that (3.63) holds for this index set \( K(\varepsilon^3) \), and we are done. \( \Box \)
THEOREM 3.66. Suppose that Subprocedure I of algorithm 3.13 generates an infinite sequence \( \{x_i\} \). Then every accumulation point of the sequence \( \{x_k\} \) constructed by Subprocedure II of algorithm 3.13 belongs to the set
\[
\{ x \in \Omega_n \mid \theta(x, 0) = 0 \}.
\]

Proof. Suppose that \( z_k \to z^* \) as \( k \to \infty \) for \( k \in K \). Since by Lemma 2.17 \( q(i) \to \infty \) as \( i \to \infty \), and \( \{x_k\} \) is infinite, \( p(k) \to \infty \) as \( k \to \infty \) and \( \epsilon_k^3 \to 0 \) as \( k \to \infty \), where \( p(k) \), \( \epsilon_k^3 \) are as defined in Step 20 of (3.13). Hence, from Corollary 3.25 we conclude that
\[
\lim_{k \to \infty} \theta(x_k, u_k, \epsilon_k^3) \leq \theta(z^*, 0) \leq 0, \quad \forall u_k \in \nabla p(k) \Phi(z_k).
\] (3.67)

However, by construction, there exists a sequence \( \{u_k\}_{k=0}^\infty \), \( (u_k = \nabla \psi(x_i)) \) for some \( j \) and \( i \) such that \( u_k \in \nabla p(k) \Phi(z_k) \) and
\[
0 \geq \theta(x_k, u_k, \epsilon_k^3) \geq -\epsilon_k^3, \quad k = 0, 1, 2, \ldots
\] (3.68)

and hence
\[
\lim_{k \to \infty} \theta(x_k, u_k, \epsilon_k^3) = 0.
\]

Substituting into (3.67) we find that \( \theta(z^*, 0) = 0 \), and we are done. \( \square \)

We can summarize our preceding results as follows.

THEOREM 3.69. Algorithm 3.13 will either jam up at a point \( x_i \), cycling indefinitely in the loop defined by Steps 1–14, in which case \( x_i \) satisfies the optimality condition \( \theta(x_i, 0) = 0 \), or else, it will construct an infinite sequence \( \{x_k\} \) every accumulation point of which belongs to the set \( \{ z^* \in \Omega_n \mid \theta(z^*, 0) = 0 \} \).

4. Solution of Min–Max Problems

Problem 4.1. Let \( \Omega_x \subset \mathbb{R}^n \) and \( \Omega_y \subset \mathbb{R}^m \) be two compact sets defined by
\[
\Omega_x = \{ x \in \mathbb{R}^n \mid g(x) \leq 0 \}, \quad (4.2)
\]
\[
\Omega_y = \{ y \in \mathbb{R}^m \mid \zeta(y) \leq 0 \}, \quad (4.3)
\]
where \( g: \mathbb{R}^n \to \mathbb{R}^t \) and \( \zeta: \mathbb{R}^m \to \mathbb{R}^t \) are continuously differentiable. We also assume that \( \Omega_y \) is convex with interior. Let \( f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l \) be continuously differentiable such that \( f(x, \cdot) \) is strictly concave for all \( x \in V \) where \( V \) is an open set containing \( \Omega_x \). Our problem is to find \( \xi \in \Omega_x \), \( \eta \in \Omega_y \) such that
\[
f(\xi, \eta) = \min_{x \in \Omega_x} \max_{y \in \Omega_y} f(x, y). \quad (4.4)
\]
To see how algorithm 3.13 can be used to solve (4.1), we need to introduce two functions. We define $\xi: V \rightarrow \Omega_y$ by

$$f(x, \xi(x)) = \max_{y \in \Omega_y} f(x, y),$$

and we define $\phi: V \rightarrow \mathbb{R}^1$ by

$$\phi(x) = \max_{y \in \Omega_y} f(x, y).$$

Now it is easy to show that $\phi$ is continuously differentiable and its derivative is given by

$$\nabla \phi(x) = \nabla_x f(x, \xi(x)).$$

Thus problem 4.1 is equivalent to

$$\min\{\phi(x) \mid g(x) \leq 0\},$$

which is precisely the problem that (3.13) is designed to solve [see (3.1)]. Furthermore, we see that in order to evaluate $\phi(x)$ and $\nabla \phi(x)$ we must evaluate $\xi(x)$, which requires the solution of an optimization problem. Because of this, the usual methods of feasible directions, which require many evaluations of $\phi(x)$ and $\nabla \phi(x)$, are likely to be very costly when applied to (4.8). Thus, the idea of approximating $\xi(x)$ (and hence $\phi(x)$ and $\nabla \phi(x)$) is appealing in this case.

Of course, it still remains to be shown that $\xi(x)$ can be approximated in such a way that assumptions 3.6 hold. In fact, such an approximation can be made. The basic idea is to let $\xi_j(x)$ be the result of $j$ iterations of the Polak method of Feasible Directions (many other algorithms also could be used) applied to $\max\{f(x, y) \mid \xi(y) \leq 0\}$ and then set

$$\phi_j(x) = f(x, \xi_j(x)) \quad \text{and} \quad \nabla \phi_j(x) = \nabla_x f(x, \xi_j(x)).$$

(Any point in $\Omega_y$ can be used as the initial point for the $j$ iterations constructing $\xi_j(x)$.) Because of space considerations, we omit the details here. The interested reader can find the rather lengthy development of the fact that this scheme satisfies the assumptions 3.6 in either [13] or [14].

Finally we remark that the conditions of (4.1) can be slightly relaxed by only requiring that $f(x, \cdot)$ be concave for all $x \in V$. With this relaxation, the function $\phi$ as defined in (4.6) is only directionally differentiable. This difficulty can be avoided by replacing $f(x, y)$ by

$$f^*(x, y, \omega) = f(x, y) - \frac{\omega}{2} \|y\|^2$$

Note that $\xi$ is well defined because of the convexity of $\Omega_y$ and the strict concavity of $f(x, \cdot)$.
with $\omega > 0$ in (4.1). Then (3.13) can be applied until
\[
\min_{x \in \Omega_x} \max_{y \in \Omega_y} f^p(x, y, \omega)
\]
is "almost solved," according to the test
\[
\tilde{b}(x_i, \nabla f^p(x_i, \omega_k), \epsilon^3) \geq -\epsilon^3,^3
\]
at which point $\omega$ is halved and the process is repeated. (Thus we are using a procedure similar to a penalty function method.) It can be shown that this procedure yields a sequence $\{x_i\} \subset \Omega_x$ such that if $x^*$ is an accumulation point of $\{x_i\}$, then $x^*$ satisfies a necessary condition of optimality. The details of this can also be found in [13] or [14].

**CONCLUSION**

We have shown in this paper that, when well-known methods of feasible directions cannot practically be applied to certain problems because of the great cost of precise function and derivative calculations, it is possible to insert into such methods stable and efficient approximation procedures which do not disrupt the convergence properties of the original algorithm. The approximation procedures described in this paper are quite general and it may be hoped that they will find their way into many algorithms when frequent precise function and derivative calculations are not practically feasible.

**REFERENCES**


^3 Here $\tilde{b}(x, \omega) = \max_{y \in \Omega_y} f^p(x, y, \omega)$ and $\epsilon^3$ is as in Step 19 of (3.13).


