A DESCENT NUMERICAL METHOD FOR OPTIMIZATION PROBLEMS
WITH NONDIFFERENTIABLE COST FUNCTIONALS∗

DIMITRI P. BERTSEKAS† AND SANDY K. MITTER‡

Abstract. In this paper we consider the numerical solution of convex optimization problems with
nondifferentiable cost functionals. We propose a new algorithm, the ‘‘mirror-descent’’ method, a large step,
double iterative algorithm which converges rapidly under very general assumptions. We discuss the
application of the algorithm to some problems of nonlinear programming and optimal control and we
show that the ‘‘subgradient method’’ contains as a special case a minimum algorithm due to Polak-Ribiere
[5].

1. General remarks. One of the most common approaches toward the
numerical solution of optimization problems with or without constraints is the
use of descent algorithms such as the steepest descent, conjugate gradient, quasi-
Newton methods, methods of feasible directions, etc. These descent methods have
enjoyed a great deal of popularity due to their reliability, simplicity, and good
convergence properties. In their usual form all these algorithms require the exis-
tence of the gradient of the function to be minimized both for explicit use in
the calculations and as a guarantee of their convergence to a local minimum. In many
optimization problems, however, often arising in an economics framework, the
natural cost functional of the problem turns out to be nondifferentiable. Such
problems have received considerable attention recently and are the subject of this
paper.

Early work on optimization problems with nondifferentiable cost functionals
can be traced to the early sixties with the research of Dubovitskii and Milyutin [1],
[2] which apparently served as a starting point for subsequent work of Soviet
scientists [3]–[6]. At about the same time the theory of subdifferentiability of convex
functions was developed by Moreau [7], [8], Rockafellar [9], [10], and Bredstedt
and Rockafellar [11]. The notion of the subdifferentia of a convex function (set of
all supporting hyperplanes to the graph of the function) provided an efficient
generalization of the notion of the ordinary gradient and formed the basis for the
development of generalized necessary and sufficient conditions for optimality (see
e.g. [10]). Necessary conditions which generalize the Pontryagin maximum
principle of optimal control in very elegant form have been given by Neustadt [12],
Heins and Mitter [13], and Rockafellar [14]. The latter reference contains also
some generalizations of known results in the calculus of variations.

∗Received by the editors March 27, 1972, and in revised form November 12, 1972.
†Department of Engineering-Economic Systems, Stanford University, Stanford, California 94305.
The work of this author was supported by the National Science Foundation under Grant NSF-GR-
20277.
‡Decision and Control Sciences Group, Electronics Systems Laboratory, Massachusetts Institute
of Technology, Cambridge, Massachusetts 02139. The work of this author was supported by the Air
637
Further necessary conditions for optimal control problems with nondifferen-
tiable cost functionals were given by Luenberger [15]. Some additional
results along the same lines can be found in the thesis by Ghanem [16]. Luenberger's
results were somewhat generalized for the case of discrete-time systems using
subdifferential theory by the authors [17]. Questions related to stochastic op-
timization problems with nondifferentiable cost functionals have been examined
in [35], [36]. Such problems occur often in stochastic programming. A method for
approximating a nondifferentiable convex function by a smooth function was also
given in reference [35]. Necessary conditions for optimality for nonlinear,
nonconvex programming problems without differentiability were obtained by
Bazaraa, Goode and Shetty [18], [19] and for minimax problems by Danzak,
Dem'yanov and Pshenichnyi [20], [21], [23]. Among existing nonlinear pro-
gramming algorithms, the convex cutting plane algorithm [25], [37] can be used
for the solution of convex nondifferentiable optimization problems.

In the area of descent numerical methods a minimization algorithm has been
reported by Ermol'ev [22], [23] and credited to Stoer [24]. This algorithm is
applicable to unconstrained convex programming problems with nondifferen-
tiable cost. It reportedly has slow convergence properties [33] although computa-
tional examples using the algorithm are not available in the English literature. A
similar algorithm has been proposed by Polyak [33]. Descent algorithms for the
solution of minimax problems have been given by Dem'yanov [21], Pshenichnyi
[5], Biruk and Pshenichnyi [26], and Levitin [34]. It should be noted that many
optimization problems with nondifferentiable cost functionals can be converted
into minimax problems. The generalization of the simplest descent method
for the numerical solution of optimization problems with nondifferentiable cost
functions was given by Luenberger [15]; however, a proof of convergence of this
algorithm is not presently available. The problem appears to be that the algo-
rithmic map in this algorithm is not closed (using Zangwill's terminology [25]).
The e-subgradient method, first presented in [17], circumvents this closure problem
as will be seen in what follows. Other papers related to optimization problems
with nondifferentiable cost functionals include those of Polyak [38], Minch [39],
Auslender [40], [41], and Bats [42].

In this paper we present a new descent algorithm for constrained or uncon-
strained minimization problems where the cost function is convex but not neces-
sarily differentiable. This algorithm, the e-subgradient method, is a large step,
double iterative algorithm that converges rapidly under very general assumptions.
The algorithm was first presented in [17] and is based on the notion of the e-
subgradient of a convex function. In § 2 we describe the algorithm and we prove
its convergence. In § 3 we consider some practical aspects of the algorithm and we
demonstrate by means of examples its application. Finally, in § 4 we delineate
some classes of problems for which the e-subgradient method compares favorably
with existing algorithms. In addition we show that the e-subgradient method
contains as a special case a minimax algorithm due to Pshenichnyi [5].

2. The e-subgradient method. In this section we describe a descent algorithm
for the minimization of a convex function subject to convex constraints. Rather
than considering explicitly the constraints, however, we shall allow the function
be minimized to take the value $+\infty$. Thus the problem of finding the minimum of a function $g\cdot h$ over a set $X$ is equivalent to finding the minimum of the extended real-valued function $f(x) = g(x) + h(x)$, where $h(\cdot|X)$ is the indicator function of $X$, i.e., $h(x|X) = 0$ for $x \in X$, $h(x|X) = +\infty$ for $x \not\in X$. Stating the problem formally:

Find $\inf_{x \in X} f(x)$ where $f: R^N \rightarrow (-\infty, +\infty] = \text{convex function which is lower semi-continuous}$ with $\inf_{x \in X} f(x) > -\infty$ and $f(x) < +\infty$ for at least one $x \in R^N$.

With the above assumptions, the function $f$ is a closed proper convex function as defined in [10]. A detailed discussion of closed proper convex functions can be found in the same reference. A basic concept for the algorithm that we shall present is the notion of $\varepsilon$-subgradient. This notion was introduced in [9, 11] in connection with investigations related to the existence and characterization of subgradients of convex functions.

Let $x$ be a point such that $f(x) = \infty$ and $\varepsilon > 0$ any positive scalar. A vector $x^* \in R^N$ is said to be an $\varepsilon$-subgradient of $f$ at $x$ if

$$f(z) \geq f(x) - \varepsilon + \langle z - x, x^* \rangle$$

for all $z \in R^N$.

where $\langle \cdot, \cdot \rangle$ denotes the usual inner product in $R^N$. The set $\partial_\varepsilon f(x)$ of all $\varepsilon$-subgradients at $x$ will be called the $\varepsilon$-subdifferential of $f$ at $x$. This set is nonempty, closed and convex. It is evident that for $0 < \varepsilon_1 < \varepsilon_2$ we have

$$\partial_\varepsilon f(x) \subseteq \partial_{\varepsilon_2} f(x) \subseteq \partial_{\varepsilon_1} f(x).$$

A useful characterization of the set $\partial_\varepsilon f(x)$ is given by the equation [10, p. 220]

$$\hat{\partial}_\varepsilon f(x) = \{x^*: f(x^*) + \langle x^*, x - x^* \rangle \leq \varepsilon\},$$

where

$$f(x^*) = \sup_{x \in R^N} \langle x, x^* \rangle - f(x)$$

is the conjugate convex function of $f$ [10]. The support function of $\hat{\partial}_\varepsilon f(x)$ is given by the following useful equation [10, p. 220]:

$$\partial_\varepsilon f(x) = \{y \in R^N: y = \inf_{x \in R^N} \frac{f(x + \lambda x^*) - f(x)}{\lambda} + \varepsilon\}.$$ 

The set $\partial_\varepsilon f(x)$ has some interesting properties from the algorithmic point of view as shown by the following two propositions.

**Proposition 1.** Let $x$ be a vector such that $f(x) < \infty$. Then

$$0 \leq f(x) - \inf_{x \in R^N} f(x) \leq \varepsilon \rightarrow 0 \hat{\partial}_\varepsilon f(x).$$

**Proof.** By the definition (1),

$$0 \leq \hat{\partial}_\varepsilon f(x) = f(z) - f(x) - \varepsilon$$

for all $z \in R^N$.

which is equivalent to the desired relations. Q.E.D.

**Proposition 2.** Let $x$ be a point such that $f(x) < \infty$ and $0 \notin \hat{\partial}_\varepsilon f(x)$. Let $y$ be any vector such that

$$\sup_{x \in R^N} \langle y, x^* \rangle < 0.$$
Then we have

\[ f(x) = \inf_{x \in X} f(x + \lambda y) > x. \]

**Proof.** Assume the contrary, i.e., \( \inf_{x \in X} f(x + \lambda y) - f(x) + \epsilon \geq 0 \). Then we have

\[
\inf_{x \in X} f(x + \lambda y) - f(x) + \epsilon \geq 0 
\]

for all \( \lambda > 0 \).

This implies by using (4)

\[
\sup_{x \in X} \langle x^*, p \rangle = \inf_{x \in X} \frac{f(x + \lambda y) - f(x) + \epsilon}{\lambda} \geq 0.
\]

Since \( \partial f(x) \) is closed this implies that \( 0 \in \partial f(x) \) which contradicts the hypothesis. Q.E.D.

In the case where \( 0 \notin \partial f(x) \), a possible method for finding a vector \( j(x) \in R^n \) such that \( \sup_{x \in X} \langle x^*, j(x) \rangle < 0 \) is the following. Let \( \| \cdot \| \) be the usual Euclidean norm in \( R^n \) and let \( x^* \) be the unique vector of minimum norm in \( \partial f(x) \). Then the vector

\[
j(x) = -\langle x^*/\|x^*\| \rangle
\]

satisfies \( \sup_{x \in X} \langle x(x), x^* \rangle = -\|x^*\| < 0 \).

Propositions 1 and 2 form the basis for the algorithm that we shall present.

The former provides a termination criterion for the algorithm. The latter states that whenever the value \( f(x) \) exceeds the optimal value by more than \( \epsilon \), then by a descent along a vector \( y \) satisfying (5) we can decrease the value of the cost by at least \( \epsilon \). Consider the following algorithm.

**Subgradient Method.**

Step 1. Select a vector \( x_0 \) such that \( f(x_0) < \infty \), a scalar \( t_0 > 0 \) and a scalar \( a, 0 < a < 1 \).

Step 2. Given \( x_k \) and \( t_k > 0 \), set \( s_{k+1} = a^k s_k \), where \( k \) is the smallest nonnegative integer such that \( 0 \notin \partial f(x_k) \).

Step 3. Find a vector \( x_{k+1} \) such that

\[
\sup_{x \in X} \langle x_{k+1}, x^* \rangle < 0.
\]

Step 4. Set \( x_{k+1} = x_k + \lambda s_{k+1} \), where \( s_{k+1} > 0 \) is such that

\[
f(x_{k+1}) - f(x_k) \geq \lambda s_{k+1}.
\]

Return to Step 2.

It should be mentioned that if \( x_k \) is not a minimizing point of \( f \) there always exists a nonnegative integer \( k \) such that \( 0 \notin \partial f(x_k) \), since by Proposition 1 we
have
\[ 0 \leq \lambda_{e_{n+1}, f(x_n)} = \inf_{x_i} f(x_i) - \inf_{x_i} f(x) > 0. \]

Also by Proposition 2 there exists a scalar \( \lambda \) such that
\[ f(x) - f(x_i + \lambda x) > 0, \]
thus showing that Step 4 can always be carried out. In fact, one can show that the set of all scalars \( \lambda \) satisfying (9) is an open bounded interval or an open half-line. One way of finding a scalar \( \lambda \) satisfying (9) is by means of the one-dimensional minimization
\[ f(x_0 + \lambda x) = \min_{\lambda \geq 0} f(x_0 + \lambda x), \]
assuming the minimum is attained. This in turn can be guaranteed whenever the set of minimizing points of \( f \) is nonempty and compact, since in this case all the level sets of \( f \) are compact [10, Cor. 8.7.1]. We note also that Steps 2 and 3 can be carried out by means of an auxiliary minimization problem as will be discussed in detail in the next section.

We now prove the convergence of the \( \phi \)-subgradient method.

Proposition 3. Consider the vectors \( x_n \) generated by the \( \phi \)-subgradient method. Then either \( f(x_n) = \min_{x \in M} f(x) \) for some \( m \geq 0 \) or the generated infinite sequence \( \{x_n\} \) satisfies:
(a) \( \lim_{n \to \infty} f(x_n) = \inf_{x \in M} f(x) \).
(b) In addition, the set \( M = \{x \in \mathcal{E}, f(x) = \inf_{x \in M} f(x)\} \) is nonempty and bounded, then:
(c) Every convergent subsequence of \( \{x_n\} \) has its limit in \( M \), and at least one such subsequence exists.
(d) For every \( \varepsilon > 0 \) there exists an \( m \geq 0 \) such that \( x_n \in M + \mathcal{B} \) for all \( n \geq m \), where \( B = \{x : \|x\| \leq \varepsilon\} \) is the unit ball in \( \mathbb{R}^n \).
(e) If the minimum of \( f \) is attained at a single point \( \bar{x} \) then \( x_0 \to \bar{x} \).
Proof. By Proposition 2 we have
\[ f(x_0) - f(x_0) > e_{n+1} \]
and hence,
\[ f(x_0) = \inf_{x \in M} f(x) \]
for all \( n \geq 0 \).

Since \( e_0 > 0 \) the above inequality implies \( |\lambda|_k \to 0 \). This implies that \( e_{n+1} < e_k \) for an infinite number of integers \( k \). In view of Step 2 of the algorithm we have for those integers: \( 0 < f(x_0) - \inf_{x \in M} f(x) \leq e \). Since \( f(x_0) \) is a decreasing sequence, it follows that \( \lim_{n \to \infty} f(x_n) = \inf_{x \in M} f(x) \) and (a) is proved. To prove (b) noting that \( x_n \in F_0, \) where \( F_0 = \{x : f(x) \leq f(x_0)\} \) and since \( M \) is nonempty and bounded, \( F_0 \) is compact (see [10, Cor. 8.7.1]). Therefore the sequence \( \{x_n\} \) has at least one convergent subsequence. The fact that the limits of all convergent subsequences belong to \( M \) follows from (a) and Cor. 27.2.1 in [10]. Part (c) follows from (a) and Thm. 27.2.1 in [10]. Q.E.D.

The above proposition establishes that the \( \phi \)-subgradient method has attractive convergence properties. In fact, it converges to the optimal value even if an
optimal solution does not exist. A further attractive feature of the method is that it guarantees substantial progress at every iteration (Step 4) and that the progress of the computation is monitored constantly via the parameter $r$ (Step 2). The price for this substantial progress is the computations necessary to find the direction of descent in Steps 2 and 3. In the next section we shall describe some practical aspects of the algorithm and demonstrate by means of examples its application.

3. Practical aspects of the $c$-subgradient method. A cursory examination of the $c$-subgradient method reveals that in fact the most difficult step in a single iteration is finding the direction of descent $v_\kappa$. However, contrary to most descent algorithms, the chosen direction of descent in the $c$-subgradient method can lead to guaranteed substantial reductions of the value of the cost functional in a single iteration. To demonstrate this fact consider the following lemma.

**Lemma.** Asssume that the scalars $c_0$ and $a$ in the $c$-subgradient method are such that

$$f(x_0) - \inf_{x} f(x) \leq c_0, \quad 1/2 \leq a < 1.$$  \hspace{1cm} (10)

Then for all $n \geq 1$,

$$f(x_n) - \inf_{x} f(x) < (1 - a)c_0 \leq (1 - a)c_{n-1}. \hspace{1cm} (11)$$

**Proof.** We have $f(x_0) - \inf_{x} f(x) \leq c_0$ implying that $0 \in \partial_c f(x_0)$. Hence in Step 2 we have $x_i = x_0$. This in turn implies that $0 \in \partial_{c_i} f(x_i)$, or equivalently,

$$f(x_i) - \inf_{x} f(x) \leq c_i / a.$$ \hspace{1cm} (Proving the $(1)$ for $n = 1$. Since $1/2 \leq a < 1$, the last inequality implies that $f(x_n) - \inf_{x} f(x) < c_i$ and the same argument as above can be used to prove $(11)$ for $n = 2$ and every $n$. Q.E.D.)

It is evident now from (11) that a substantial reduction of the value of the cost functional is possible by choosing the value of the parameter $a$ high enough. On the other hand, a value of the parameter $a$ close to unity leads to an increased number of iterations in order to find the scalar $c_{n-1}$ from $c_n$. It is performed in Step 2 of the algorithm. Thus, in practice, one must settle on a compromise value for the parameter $a$ depending on how difficult it is to carry out a single check $0 \in \partial_{c_i} f(x_i)$ in Step 2.

Another possibility is to modify the algorithm so that the value of the parameter $a$ is adjusted during the iterations in Step 2 on the basis of information already obtained. A number of convergent schemes are possible. We do not discuss these schemes since they are not theoretically interesting but rather relate to the intelligent programming of the method.

We now turn to the important question of how the calculation of the direction of descent is to be carried out once the value of the parameter $a$ is selected. As
mentioned in the previous section it is possible to carry out Steps 2 and 3 of the algorithm by solving the following minimization problem:

\[
\min_{x \in \mathbb{R}^n} \|x^*\|.
\]

Newly we have \(0 \in \partial_{\mu_0} f(x_k)\) if and only if the problem (12) has a zero optimal value and therefore Step 2 of the algorithm can be carried out by solving problem (12) successively for \(k = 0, 1, \ldots\). There exists an integer \(k\) for which problem (12) has a nonzero optimal value. Let \(x^*\) be the optimal solution of problem (12) for the first such integer \(k\). Then a suitable direction of descent \(y_k\) satisfying (8) in Step 3 of the algorithm is given by

\[
y_k = -x^*||x^*||.
\]

One efficient method for solving the minimization problem (12) is to solve successively the unconstrained problem

\[
\min_{x \in \mathbb{R}^n} \|x^*\|^2 + P(x^*),
\]

where \(P(\cdot)\) is a (moderate) penalty function

\[
r(x^*) \geq 0 \quad \text{for all } x^*.
\]

\[
P(x^*) = 0 \quad \text{if and only if } x^* \in \partial_{\mu_0} f(x_k).
\]

It is clear that problem (14) has a zero optimal value if and only if problem (12) has a zero optimal value. Furthermore, when \(k\) is such that problem (12) has a nonzero value, problem (14) yields an approximate solution \(x^*\) to problem (12). In this case one can either increase the penalty and obtain a more accurate solution or obtain an approximate direction of descent \(y_k\) from

\[
y_k = -x^*||x^*||.
\]

The approximate direction \(y_k\) is considered acceptable if it yields a point \(x_{k+1}\) satisfying \(f(x_k) - f(x_{k+1}) > c_{k+1}\) in Step 4. If \(y_k\) is not acceptable we increase the penalty in problem (14) and resolve the problem in order to obtain a more accurate direction of descent.

The preceding discussion clearly demonstrates that the application of the \(\varepsilon\)-subgradient method to a specific problem requires the solution of minimization problems of the form

\[
\min_{x \in \mathbb{R}^n} \|x\|.
\]

At first sight it would therefore appear that the \(\varepsilon\)-subgradient method can be applied only to the limited class of functions for which the \(\varepsilon\)-subdifferential \(\partial_{\varepsilon} f(x)\) has a convenient characterization. We shall demonstrate in what follows in this section that this is not the case and, in fact, the method can be applied to most functions likely to be encountered in practice. This is due to the fact that problem (16) can be cast into the usual nonlinear programming framework even if a convenient closed form characterization of the set \(\partial_{\varepsilon} f(x)\) is not available.
By making use of the characterization (2) of the \(\partial f(x)\) in terms of the conjugate convex function \(f^*\), problem (16) can be written as

\[
\text{minimize } ||x^*||
\]

subject to

\[
f^*(x^*) + f(x) = \langle x, x^* \rangle \leq \varepsilon.
\]

Now there is a class of simple functions \(f\) for which the conjugate

\[
f^*(x^*) = \sup_{x^*} \{ \langle x, x^* \rangle - f(x) \}
\]

has a convenient closed form. Such functions include:

(a) Positively homogeneous closed convex functions, i.e., support functions of given sets [10, §13]. Thus if

\[
f(x) = \sigma (\langle x, x^* \rangle) = \sup_{x^*} \langle x, x^* \rangle,
\]

then

\[
f^*(x^*) = \delta (\langle x, x^* \rangle) = \begin{cases} 
0 & \text{if } x^* \in \mathcal{X}, \\
\infty & \text{if } x^* \notin \mathcal{X},
\end{cases}
\]

where \(\mathcal{X}\) is the closure of the convex hull of \(X\). This class includes all norms and seminorms in \(\mathbb{R}^d\) as well as linear functions. In addition, the conjugates of powers greater than one of norms and seminorms in \(\mathbb{R}^d\) (including quadratic forms) are given in (10, §15).

(b) Exponentials and logarithms of coordinates of \(x\) (see [10, §12]).

(c) Indicator functions of affine sets (linear manifolds), convex cones and unit balls with respect to a norm or a seminorm [10, §13].

(d) Indicator functions of sets with known support functions, [10, §13]. If \(X\) is a closed convex set and

\[
f(x) = \delta (x, X),
\]

then

\[
f^*(x^*) = \sigma (\langle x, x^* \rangle) = \sup_{x^*} \langle x, x^* \rangle.
\]

We note that constraint sets which are characterized by their support function are encountered, for example, in some optimal control problems as will be discussed in some detail in §4.

Now from this class of simple functions one can build more complicated functions by means of various operations such as summation, affine transformation, maximization, etc. The conjugates of such functions are characterized by the following well-known relations:

\[
(f_1 + f_2 + \cdots + f_m)^*(x^*) = \min_{x^*} \left\{ \sum_{i=1}^m f_i(x^*) \right\}
\]

[10, Thm. 16.4],

where \(f_i, i = 1, \ldots, m\), are closed proper convex functions with a common point in the relative interior of their effective domain, and the function \(f_1 + \cdots + f_m\) is
defined by
\[ f_1 + f_2 + \cdots + f_m(x) = f_1(x) + f_2(x) + \cdots + f_m(x), \]
(19)
\[ f^*(A^*y^*) = \min_{y \in \mathcal{F}} f^*(y^*), \]
where \( f: \mathbb{R}^n \to \mathbb{R} \) is a closed proper convex function, \( A \) is a linear transformation from \( \mathbb{R}^n \) to \( \mathbb{R}^m \), \( A^* \) denotes its adjoint, the function \( f^* \) is the composition of \( f \) and \( A \), and, in addition, the range of \( A \) contains a point in the relative interior of the effective domain of \( f \).

(20) \( \max \{ f_1, \ldots, f_m \} (x^*) = \min \left\{ \sum_{i=1}^m \lambda_i f_i(x^*) \right\} \quad ([10, \text{Thm. } 16.5]), \]
where \( f_i, \ i = 1, \ldots, m, \) are convex real-valued functions and the function \( \max \{ f_1, \ldots, f_m \} \) is defined by
\[ \max \{ f_1(x), \ldots, f_m(x) \} = \max \{ f_1(x), \ldots, f_m(x) \}, \]
(21)
\[ g^*(x^*) = f(x^*) + \langle c, x^* \rangle, \]
where \( g(x) = f(x - c), f: \mathbb{R}^n \to (-\infty, +\infty) \) is a closed proper convex function and \( c \in \mathbb{R}^n \) is a given vector.

The equations (18)–(21) can be used in order to put the minimization problem (17) in the standard nonlinear programming framework for a wide variety of functions. As an illustration, consider the case where the function \( f \) to be minimized by means of the \( \varepsilon \)-subgradient method has the form
\[ f(x) = f_1(x) + f_2(x) + \cdots + f_m(x). \]
By using (18) the optimization problem (17) can be written as
\[
\begin{align*}
\text{minimize} & \quad \langle x, x^* \rangle \\
\text{subject to} & \quad \sum_{i=1}^m f_i(x^*) \end{align*}
\]
It can be easily seen that the above problem is equivalent to
\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^m f_i(x^*) + f(x) - \langle x^*, x \rangle \leq \varepsilon.
\end{align*}
\]
This latter problem is in the standard nonlinear programming framework whenever the functions \( f_i \) belong to the class of simple functions mentioned earlier. As another example consider the case where the function \( f \) has the form
\[ f(x) = \max \{ f_1(A_1x), \ldots, f_m(A_mx) \}. \]
where \( A_1, \ldots, A_m \) are linear transformations and \( f_1, \ldots, f_m \) are real-valued convex functions. By using (19), (20), the optimization problem (17) for this function can be written as

\[
\begin{align*}
\text{minimize} \quad & \|x^*\| \\
\text{subject to} \quad & \min_{\lambda_i \geq 0} \left\{ \sum_{i=1}^m \lambda_i \min_{t \in [0,1]} f_i(t) \right\} + f(x) - \langle x^*, x \rangle \leq \varepsilon, \\
& \sum_{i=1}^m \lambda_i = 1
\end{align*}
\]

or equivalently,

\[
\begin{align*}
\text{minimize} \quad & \sum_{i=1}^m \lambda_i \langle A_i y_i \rangle \\
\text{subject to} \quad & \sum_{i=1}^m \lambda_i f_i(y_i) + f(x) - \sum_{i=1}^m \lambda_i \langle A_i y_i \rangle \leq \varepsilon, \\
& \lambda_i \geq 0, \quad \sum_{i=1}^m \lambda_i = 1.
\end{align*}
\]

Similarly, one can write the optimization problem (17) in standard form whenever the function to be minimized involves simultaneously sums, compositions with linear transformations and maxima of the basic simple functions referred to earlier. Thus the \( \rho \)-subgradient method can be applied for the minimization of a wide class of functions. This class of functions can be further enlarged by making use of the following technique to eliminate some of the constraints of the minimization problem.

Consider the convex programming problem

(22a)

\[
\begin{align*}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & x \in X, \quad f_j(x) \leq 0, \quad j = 1, \ldots, m,
\end{align*}
\]

where \( f_1, f_2, \ldots, f_m \) are real-valued convex functions and \( X \) is a closed convex set. Let \( x \) be an optimal solution of this problem and assume that there exists a point \( x \in X \) such that \( f_j(x) < 0, \quad j = 1, \ldots, m \). Then there exist nonnegative Lagrange multipliers, \( \lambda_1, \ldots, \lambda_m \), corresponding to \( \varepsilon \) (23), (37) such that \( x \) minimizes

\[
f(x) + \sum_{i=1}^m \lambda_i f_i(x)
\]

subject to \( x \in X \). Furthermore, it is known [15] that if \( k \) is a scalar such that

(23)

\[
k > \max \{ \lambda_1, \ldots, \lambda_m \}
\]
then \( x \) is an optimal solution to the problem

\[
\begin{equation}
\min \{ f(x) + k \sum_{i=1}^n \max \{ 0, f_i(x) \} \}
\end{equation}
\]

subject to \( x \in X \).

Conversely, every optimal solution of problem (24) is an optimal solution of problem (22) so that the two problems are equivalent and either one of the two can be solved in place of the other. Concerning the selection of the scalar \( k \), it can be easily proved that if \( \mu \) is a strict lower bound for the optimal value of problem (22), then

\[
k = \max \left\{ \frac{f_i(x) - \mu}{-f_i(x)} : i = 1, 2, \ldots, m \right\}
\]

satisfies (23), where \( \mathbf{x} \) is a vector such that \( \mathbf{x} \in X \) and \( f_i(x) < 0, i = 1, 2, \ldots, m \).

We shall close this section by showing explicitly the form of the auxiliary minimization problem (17) for a specific problem.

Example. Consider the problem

\[
\begin{equation}
\min \left\{ \max_{\|Q\|_2} \langle x, y \rangle + \max \{ 0, \frac{1}{2} \| Q x + \langle c, x \rangle \|_2^2 \} \right\}
\end{equation}
\]

subject to \( x \in X = \{ x \in \mathbb{R}^m : x_i \geq 0, i = 1, \ldots, m \} \).

In the above problem, \( x, y \) are vectors in \( \mathbb{R}^n \); \( \cdot \) denotes the Euclidean norm in \( \mathbb{R}^n \); \( Q \) is a positive definite matrix and \( c \) is a given vector. By defining

\[
\begin{align*}
f_1(x) &= \max_{\|Q\|_2} \langle x, y \rangle, \\
f_2(x) &= (1/2) \| Q x + \langle c, x \rangle \|_2^2, \\
f_3(x) &= \delta_{[\mathbb{R}^m]}(x) = \\
&= \begin{cases} 0 & \text{if } x \in X, \\
\infty & \text{if } x \notin X, \end{cases}
\end{align*}
\]

the problem is written

\[
\begin{equation}
\min f(x) = f_1(x) + \max \{ 0, f_2(x) \} + f_3(x).
\end{equation}
\]

The auxiliary optimization problem to be solved in Steps 2 and 3 of the c-subgradient method is

\[
\begin{equation}
\min \| x^* \|
\end{equation}
\]

subject to \( f'(x^*) + f(x) - \langle x^*, x \rangle \leq 0 \). By using (18) and (20) and the fact that the conjugate of the zero function is the function

\[
(0)^*(x^*) = \begin{cases} 0 & \text{if } x^* = 0, \\
\infty & \text{if } x^* \neq 0,
\end{cases}
\]

the above problem is equivalent to

\[
\begin{equation}
\min \| x^* \| + \min \{ 0, \| x^* \| \}
\end{equation}
\]
subject to
\[
 f(x^0) + \lambda f(x) + f(x) + f(x) - (\lambda x^* + \lambda x^* + x_0, x) \leq \epsilon_r,
\]
\[
 0 \leq \lambda \leq 1.
\]
We have
\[
 f(x) = \begin{cases} 
 0 & \text{if } x^* \geq 0, \quad |x^*| \leq 1, \\
 \infty & \text{otherwise}, 
\end{cases}
\]
\[
 f(x) = \begin{cases} 
 \lambda x^* - c_1 Q^{-1} c_1 - c_1, & \text{if } x^* \leq 0, \\
 \infty & \text{otherwise}, 
\end{cases}
\]
\[
 f(x) = \begin{cases} 
 0 & \text{if } x^* \geq 0, \\
 \infty & \text{otherwise}, 
\end{cases}
\]
where the inequalities \( x^* \geq 0, x^* \leq 0 \) are interpreted to be componentwise. Hence problem (20) takes the form
\[
\text{minimize } |x^* + \lambda x^* + x_0|^2
\]
subject to
\[
 (\lambda/2)(x^*_0 - c_1 Q^{-1} c_1 - c_1) + f(x) - (\lambda x^* + \lambda x^* + x_0, x) \leq \epsilon_r,
\]
\[
 0 \leq x^*_0. \quad |x^*_0| \leq 1. \quad x^*_0 \leq 0. \quad 0 \leq \lambda \leq 1.
\]
a nonlinear program with linear and quadratic constraints. If \((x^*, x_0, \lambda, \theta)\) is an optimal solution of the above problem then \(x^* = x^* + \lambda x^* + x_0\) is an optimal solution of the auxiliary optimization problem of Steps 2 and 5 of the \( \epsilon \)-subgradient method.

4. Applications. In this section we attempt to delineate some classes of problems for which the \( \epsilon \)-subgradient method compares favorably with existing methods. It is well known that many optimization problems with nondifferentiable cost functionals can be converted into nonlinear programming problems where all functions involved are differentiable. For example consider the problem
\[
\begin{align*}
 &\text{minimize } \max (f_1(x), \ldots, f_m(x)), \\
 \text{subject to } &f_j(x) \leq y, \quad j = 1, \ldots, m,
\end{align*}
\]
where the functions \( f_i \) are convex and differentiable. This problem is equivalent to the problem
\[
\begin{align*}
 &\text{minimize } y \\
 \text{subject to } &f_j(x) \leq y, \quad j = 1, \ldots, m,
\end{align*}
\]
where \( y \) is a scalar auxiliary variable. This latter problem can be solved by any of the existing algorithms for differentiable functions such as, for instance, the \( \epsilon \)-perturbation feasible direction method [25]. Alto problem (26) can be solved by using Dem'yanov's minimax algorithm [21] which is closely related to the feasible direction method mentioned above. It appears that either one of the two algorithms is preferable to the \( \epsilon \)-subgradient method for the solution of problem (26). This is due to the considerable computation necessary in order to find the direction of descent in the \( \epsilon \)-subgradient method. More generally, one can say that if the optimization problem can be converted to a nonlinear program where all functions
involved are differentiable, standard methods should, in most cases, be preferable to the ε-subgradient method.

The ε-subgradient method, however, should be considered advantageous when applied to problems which cannot be converted to nonlinear programming problems involving differentiable functions since it has the advantage of fast convergence. One class of such problems is characterized by the presence of terms of the form \( \max_{x \in Y} \langle x, y \rangle \) either in the cost function or the constraints. The first known algorithm involving functions of the form \( \max_{x \in Y} \langle x, y \rangle \) is the one of Polyak's (5) who considered the problem

\[
\begin{align*}
\text{minimize} & \quad \max_{x \in Y} \langle x, y \rangle \\
\text{subject to} & \quad x \in A,
\end{align*}
\]

where \( Y \) is a convex compact set and \( A \) is a given hyperplane. When the ε-subgradient method is applied to problem (26), the direction of descent is determined by solving the auxiliary optimization problem

\[
\begin{align*}
\text{minimize} & \quad \| x^*_t + \varepsilon s \|_2 \\
\text{subject to} & \quad x^*_t \in Y, \quad \max_{x \in Y} \langle x, y \rangle - \langle x^*_t, y \rangle \leq \varepsilon, \\
& \quad x^*_t \in A^t,
\end{align*}
\]

where \( A^t \) is the one-dimensional subspace orthogonal to the hyperplane \( A \). This is exactly the same optimization problem by means of which the direction of descent is determined in Polyak's method and thus the ε-subgradient method and Polyak's method are identical when applied to problem (28). The ε-subgradient method, however, can be applied to much more general problems involving terms of the form \( \max_{x \in Y} \langle x, y \rangle \). One such example was given in the previous section. For such problems the ε-subgradient method compares favorably with, for example, Dem'yanov's minmax algorithm which involves comparable computations for finding the direction of descent but does not converge as fast as the ε-subgradient method.

The ε-subgradient method can also be used effectively for problems where some of the constraint sets are not given explicitly but instead can be specified from their support function. For such problems methods of feasible directions, for example, are not applicable. As an example, consider the following optimal control problem where some of the constraint sets are characterized as reachable sets of a differential system.

Consider the linear system

\[
\begin{align*}
\dot{x}(t) & = A(t)x(t) + B(t)u(t)
\end{align*}
\]

over the time interval \( [t_0, T] \) which is controllable from \( t_0 \) to \( T \) and where \( \dot{x}(t) \) is a Lebesgue integrable \( n \times n \) matrix, and \( B(t) \) is a continuous \( n \times m \) matrix function on \( [t_0, T] \). The \( u \) vector-valued function \( u(t) \) is assumed to be measurable in \( [t_0, T] \) and such that

\[
\begin{align*}
u(t) \in U \quad \text{almost everywhere in} \quad [t_0, T],
\end{align*}
\]
where $U$ is a nonempty compact subset of $\mathbb{R}^n$. Assume further that the initial condition is constrained to lie in $X_0$, a convex compact subset of $\mathbb{R}^n$:

$$X_0 \subseteq X.$$ (31)

Consider the problem of minimizing

$$J[x(t_0), u] = F[x(T)]$$ (32)

where $F$ is a closed proper convex function in $\mathbb{R}^n$ subject to the constraints (29)–(31).

Then under our assumptions, for every pair $(x(t_0), u)$ satisfying (30) and (31), there exists a unique absolutely continuous solution of (29). The set $X(T)$ of reachable states $x(T)$ at time $T$ corresponding to the constraints (30), (31) is convex and compact by a theorem of Neustadt [30], and its support function is given by (31), (32)

$$\phi(x) := \sup_{x(t_0)} \phi(x) = \sup_{x(t_0) \in X_0} \phi(x) := \sup_{x(t_0) \in X_0} \frac{1}{\alpha} \sup_{\tau \in [0, T]} \left[ \frac{1}{2} \left( \frac{1}{\alpha} \|x_0 \| + \|F\| \|x_0 \| + \|U\| \right) \right],$$

where $\Phi(\alpha, \tau)$ is the unique absolutely continuous transition matrix corresponding to the matrix $A(t)$.

The problem can now be recast as one of minimizing the extended real-valued convex function

$$f(x,T) = F[x(T)] + \phi(x(T))$$

and the $\epsilon$-subgradient method can be used for its solution. The direction of descent is determined by solving the optimization problem

$$\text{minimize} \|x_T - x_T^\ast\|$$

subject to

$$F(x) + \phi(x) = F[x(T)] + \phi(x(T)) - \langle x_T - x_T^\ast, x(T) \rangle \leq \epsilon.$$ (33)

For the problem that we consider there is some difficulty associated with the one-dimensional line search in Step 4 of the $\epsilon$-subgradient method since it is not easy to check feasibility of any given terminal state. This difficulty can be circumvented by finding a point along the direction of descent such that the value of the function $F$ has decreased by $\epsilon$ or a little less. It can be easily seen that such a point is feasible and that the algorithm will still be convergent.

5. Conclusions. The $\epsilon$-subgradient method is a descent algorithm which can solve efficiently some convex minimization problems with nondifferentiable cost functionals which cannot be solved by standard nonlinear programming methods. It converges fast under very general assumptions but requires the solution of an auxiliary optimization problem in order to determine the direction of descent at each iteration. Presently, we do not have any computational experience with the method. It is hoped that such computational experience will be gained in the near future.
REFERENCES


[41] ———, Recherche de points de null d'une fonction, Cahiers Centre Études Recherche Opér., 12 (1979), no. 2.