MATHEMATICS OF COMPUTATION
VOLUME 45, NUMBER 171
JULY 1985, PAGES 173–180

An Algorithm for Nonsmooth Convex Minimization With Errors

By Krzysztof C. Kiwiel

Abstract. A readily implementable algorithm is given for minimizing any convex, not necessarily differentiable, function f of several variables. At each iteration the method requires only one approximate evaluation of f and its ε-subgradient, and finds a search direction by solving a small quadratic programming problem. The algorithm generates a minimizing sequence of points, which converges to a solution whenever f has any minimizers.

1. Introduction. This paper presents an algorithm for minimizing a convex, but not necessarily differentiable, function $f\colon R^N \to R$. We suppose that, given $x \in R^N$ and $\varepsilon > 0$, we have a finite process which will find a number $f_\varepsilon(x)$, satisfying

$$f(x) - \varepsilon \leqslant f_\varepsilon(x) \leqslant f(x) + \varepsilon, \tag{1}$$

and one ε-subgradient $g_\varepsilon(x)$ of f at x; i.e., an arbitrary element of the ε-subdifferential

$$\partial_\varepsilon f(x) = \left\{ g \in R^N : f(y) \geqslant f(x) + \langle g, y - x \rangle - \varepsilon \ \forall y \in R^N \right\}$$

of f at x.

The above assumption is realistic in many applications (see, e.g., Shor [10]). For instance, if f is a max-type function of the form

$$f(x) = \sup\{ \phi_u(x) : u \in U \} \quad \text{for all } x \in R^N, \tag{2}$$

where each $\phi_u\colon R^N \to R$ is convex and U is an infinite set, then it may be impossible to calculate $f(x)$. However, for any positive ε one can usually find in finite time an ε-solution to the maximization problem in (2); i.e., an element $u_\varepsilon \in U$ satisfying $\phi_{u_\varepsilon}(x) \geqslant f(x) - \varepsilon$. Then one may set $f_\varepsilon(x) = \phi_{u_\varepsilon}(x)$ and take $g_\varepsilon(x)$ as any subgradient of ϕ_{u_ε} at x. On the other hand, in some applications, calculating u_ε for a prescribed $\varepsilon > 0$ may require much less work than computing u_0. This is, for instance, the case when the maximization in (2) involves solving a linear or discrete programming problem by the methods of Gabasov and Kirilova [3].

The algorithm presented in this paper is a descent method. At each iteration a trial point is found by solving a small quadratic programming subproblem derived from a piecewise linear (polyhedral) approximation to f around the current iterate. The trial point is accepted as the next iterate only if this decreases the objective value;

Received November 14, 1983; revised May 8, 1984.
1980 *Mathematics Subject Classification.* Primary 65K05; Secondary 90C25.
Key words and phrases. Mathematical programming, nonsmooth optimization, convex nondifferentiable optimization, descent methods, aggregate subgradients.

0025-5718/85 $1.00 + $.25 per page

otherwise, the new subgradient information collected at the trial point enriches the next polyhedral approximation so as to deal with the nondifferentiability of f. The accuracy tolerance ε is automatically reduced as the method proceeds. The reduction is, on the one hand, slow enough to save work by allowing inexact evaluations of f far from a solution, and, on the other hand, sufficiently fast to ensure that the method generates a minimizing sequence of points. Moreover, this sequence converges to a solution whenever f attains its infimum.

Our algorithm is an extension of the aggregate subgradient method due to Kiwiel [5], who modified a descent method of Lemarechal [7] ([5] and [7] require exact evaluations). Vasilyev [2, Section 3.7] proposed a comparable method by extending the Wolfe conjugate subgradient method [11], which is less efficient than that of Lemarechal [7]. We hope that our algorithm is superior to that of [2], since it is simpler, does not use resets which slow down convergence, and has much stronger convergence properties.

Some other methods, e.g., [1] and [9], reduce ε in a preassigned way, depending only on the number of iterations performed. This, of course, may be inefficient. The ε-subgradient method of Polyak [9] is nonmonotone in objective values, while the descent method of Auslender [1] requires continuous differentiability of f.

It will be seen that there is great freedom of choice in how the algorithm can be run. Since any description of specific computational techniques would take up too much space, numerical results [4] will appear elsewhere.

We may add that it would be difficult to extend the algorithm to the nonconvex case, as was done by Kiwiel [6] for exact evaluations. Convexity is essential for avoiding line searches, which would be impossible in the nonconvex case because the noisy function f_ε is not semismooth (see Mifflin [8]).

The method is derived and stated in Section 2. In Section 3 we establish its global convergence. Computational modifications are discussed in Section 4. Finally, we have a conclusion section.

We shall use the following notation. R^N denotes N-dimensional Euclidean space with the usual inner product $\langle \cdot, \cdot \rangle$ and associated norm $|\cdot|$. All vectors are row vectors. We will denote by "conv" the convex hull.

2. The Method. The algorithm to be described will generate a sequence of points $x^1, x^2, \ldots$ of R^N and search directions $d^1, d^2, \ldots$ in R^N and stepsizes $t_L^1, t_L^2, \ldots$ in $\{0, 1\}$, related by $x^{k+1} = x^k + t_L^k d^k$ for $k = 1, 2, \ldots$, where x^1 is a given starting point. The sequence $\{x^k\}$ is intended to converge to the required solution. The method will also calculate trial points $y^{k+1} = x^k + d^k$ and accuracy tolerances $\varepsilon^k > 0$, for $k = 1, 2, \ldots$, and ε^k-subgradients $g^k = g_{\varepsilon^k}(y^k)$, for all $k \geq 1$, where $y^1 = x^1$ and $\varepsilon^1 > 0$ is given.

With each trial point y^j we associate the following linearization of f:

$$f_j(x) = f_{\varepsilon^j}(y^j) + \langle g^j, x - y^j \rangle - 2\varepsilon^j \quad \text{for all } x \in R^N, \tag{3}$$

which satisfies

$$f(x) \geq f_j(x) \quad \text{for all } x, \tag{4}$$

since $g^j \in \partial_{\varepsilon'} f(y^j)$ and $f(y^j) \geqslant f_{\varepsilon'}(y^j) - \varepsilon^j$. Thus, at the kth iteration of the method, the polyhedral function

$$\bar{f}^k(x) = \max\{ f_j(x) : j = 1,\ldots,k \} \tag{5}$$

would incorporate all the past subgradient information about f. Hence, one could try to find a descent direction for f at x^k by solving the subproblem

$$\text{minimize } \bar{f}^k(x^k + d) \text{ over all } d \in R^N. \tag{6}$$

However, such a cutting plane approach (see [2]) has the following drawbacks. First, subproblem (6) may have no finite solutions. Second, and more importantly, using subproblem (6) would lead to difficulties with storage and computation after a large number of iterations, since $\bar{f}^k$ has k linear pieces at the kth iteration. These two drawbacks can be eliminated by using the ideas of Lemarechal [7] and Kiwiel [5], respectively, as follows.

At the kth iteration we shall have a nonempty set $J^k \subset \{1,\ldots,k\}$ and the linearizations f_j, $j \in J^k$, given by the $(N+1)$-vectors (g^j, f_j^k) in the form

$$f_j(x) = f_j^k + \langle g^j, x - x^k \rangle \quad \text{for all } x,$$

where $f_j^k = f_j(x^k)$ for $j \in J^k$. To save storage and work per iterations, J^k will be a small subset of $\{1,\ldots,k\}$, e.g., $J^k = \{k\}$. Of course, replacing $\{1,\ldots,k\}$ by J^k in (5) could yield a poor model of f. Therefore, the dropped past subgradient information will be compensated for by using the aggregate linearization

$$\tilde{f}^{k-1}(x) = f_p^k + \langle p^{k-1}, x - x^k \rangle \quad \text{for all } x,$$

generated at the $(k-1)$st iteration by the aggregate subgradient (p^{k-1}, f_p^k), satisfying

$$(p^{k-1}, f_p^k) \in \operatorname{conv}\{ (g^j, f_j^k) : 1 \leqslant j \leqslant k \}, \tag{7}$$

so that we have

$$\tilde{f}^{k-1}(x) \in \operatorname{conv}\{ f_j(x) : 1 \leqslant j \leqslant k \} \quad \text{for all } x,$$

$$f(x) \geqslant \tilde{f}^{k-1}(x) \quad \text{for all } x.$$

Observe that the aggregate subgradient (p^{k-1}, f_p^k) generates a linearization of f in a manner similar to any ordinary "augmented" subgradient (g^j, f_j^k). It will be updated recursively so as to accumulate information about nondifferentiabilities of f. The available linearizations will define the kth lower polyhedral approximation of f:

$$\hat{f}^k(x) = \max\{ \tilde{f}^{k-1}(x), f_j(x) : j \in J^k \} \quad \text{for all } x.$$

Since we want d^k to be a descent direction for f at x^k, we shall find it to

$$\text{minimize } \hat{f}^k(x^k + d) + \tfrac{1}{2}|d|^2 \text{ over all } d \in R^N, \tag{8}$$

where the penalty term $|d|^2/2$ will tend to keep $y^{k+1} = x^k + d^k$ in the region where $\hat{f}^k(\cdot)$ is close to $f(\cdot)$, and will ensure that d^k exists. Moreover,

$$v^k = \hat{f}^k(x^k + d^k) - [f_{\varepsilon^k}(x^k) + \varepsilon^k]$$

yields an estimate of $f(x^k + d^k) - f(x^k)$, which will be used for testing whether y^{k+1} is better than x^k.

Having motivated the search-direction-finding subproblems, we may now state the method, commenting on its rules as follows.

ALGORITHM 2.1.

Step 0 (*Initialization*). Select a starting point $x^1 \in R^N$, an initial accuracy tolerance $\varepsilon^1 \geqslant 0$, and a stopping parameter $\varepsilon_s \geqslant 0$. Choose two line search parameters m_L and m_R satisfying $0 < m_L < m_R < 1$. Set $y^1 = x^1$, $J^1 = \{1\}$, $p^0 = g^1 = g_{\varepsilon^1}(y^1)$, $f_p^1 = f_1^1 = f_{\varepsilon^1}(y^1)$. Set the counters $k = 1$, $l = 0$, and $k(0) = 1$.

Step 1 (*Direction finding*). Find the solution $(d^k, v^k) \in R^N \times R$ to the following kth quadratic programming subproblem:

$$\begin{aligned} &\text{minimize } \tfrac{1}{2}|d|^2 + v \text{ over all } (d, v) \in R^N \times R \text{ satisfying} \\ &-\alpha_j^k + \langle g^j, d\rangle \leqslant v \text{ for } j \in J^k,\ -\alpha_p^k + \langle p^{k-1}, d\rangle \leqslant v, \end{aligned} \tag{9}$$

where

$$\alpha_j^k = f_\varepsilon(x^k) + \varepsilon - f_j^k \quad \text{and} \quad \alpha_p^k = f_\varepsilon(x^k) + \varepsilon - f_p^k \tag{10}$$

for $\varepsilon = \varepsilon^k$. Calculate Lagrange multipliers λ_j^k, $j \in J^k$, and λ_p^k of (9) and set

$$(p^k, \tilde{f}_p^k) = \sum_{j \in J^k} \lambda_j^k (g^j, f_j^k) + \lambda_p^k (p^{k-1}, f_p^k), \tag{11}$$

$$\tilde{\alpha}_p^k = f_\varepsilon(x^k) + \varepsilon - \tilde{f}_p^k. \tag{12}$$

Step 2 (*Accuracy updating*). If $\varepsilon > -(m_R - m_L)v^k/5$, set $\varepsilon^k = \varepsilon/2$ and go to Step 1; otherwise, proceed.

Step 3 (*Stopping criterion*). If $v^k \geqslant -\varepsilon_s$, terminate; otherwise, proceed.

Step 4 (*Trial point testing*). Set $y^{k+1} = x^k + d^k$. If

$$f_{\varepsilon^k}(y^{k+1}) \leqslant f_{\varepsilon^k}(x^k) + m_L v^k - 2\varepsilon^k, \tag{13}$$

then set $t_L^k = 1$ (serious step), set $k(l+1) = k+1$, and increase l by 1; otherwise (i.e., if (13) is violated), set $t_L^k = 0$. Set $x^{k+1} = x^k + t_L^k d^k$.

Step 5 (*Linearization updating*). Set $\varepsilon^{k+1} = \varepsilon^k$. Choose a (possibly empty) set $\hat{J}^k \subset J^k$ and set $J^{k+1} = \hat{J}^k \cup \{k+1\}$. Compute $g^{k+1} = g_{\varepsilon^{k+1}}(y^{k+1})$ and

$$\begin{aligned} f_{k+1}^{k+1} &= f_{\varepsilon^{k+1}}(y^{k+1}) + \langle g^{k+1}, x^{k+1} - y^{k+1}\rangle - 2\varepsilon^{k+1}, \\ f_j^{k+1} &= f_j^k + \langle g^j, x^{k+1} - x^k\rangle \quad \text{for } j \in \hat{J}^k, \end{aligned} \tag{14}$$

$$f_p^{k+1} = \tilde{f}_p^k + \langle p^k, x^{k+1} - x^k\rangle. \tag{15}$$

Increase k by 1 and go to Step 1.

A few comments on the algorithm are in order.

As in [5], the kth subproblem dual is to find values of the multipliers λ_j^k, $j \in J^k$, and λ_p^k to

$$\begin{aligned} &\underset{\lambda_j, \lambda_p}{\text{minimize}}\ \frac{1}{2}\left|\sum_{j \in J^k} \lambda_j g^j + \lambda_p p^{k-1}\right|^2 + \sum_{j \in J^k} \lambda_j \alpha_j^k + \lambda_p \alpha_p^k, \\ &\text{subject to } \lambda_j \geqslant 0 \text{ for } j \in J^k, \lambda_p \geqslant 0,\ \sum_{j \in J^k} \lambda_j + \lambda_p = 1. \end{aligned} \tag{16}$$

Any solution of (16) is a Lagrange multiplier vector for (9) and yields the unique solution (d^k, v^k) of (9) as follows:

$$d^k = -p^k, \tag{17}$$

$$v^k = -\left\{ |p^k|^2 + \sum_{j \in J^k} \lambda_j^k \alpha_j^k + \lambda_p^k \alpha_p^k \right\} = -\left\{ |p^k|^2 + \tilde{\alpha}_p^k \right\}, \tag{18}$$

where p^k and $\tilde{\alpha}_p^k$ are given by (11)–(12). Moreover, any Lagrange multipliers of (9) also solve (16). In particular, we always have

$$\lambda_j^k \geqslant 0 \quad \text{for } j \in J^k, \qquad \lambda_p^k \geqslant 0, \qquad \sum_{j \in J^k} \lambda_j^k + \lambda_p^k = 1. \tag{19}$$

Thus, one may equivalently solve the dual subproblem (16) in Step 1.

Our rules for aggregating the past subgradient information may be explained as follows. First, by combining (7), (11), (15) and (19) with the fact that a convex combination of convex combinations is again a convex combination, one can show (see [5] for details) that

$$\left(p^k, \tilde{f}_p^k \right) \in \operatorname{conv}\left\{ \left(g^j, f_j^k \right) : 1 \leqslant j \leqslant k \right\} \quad \text{for each } k,$$

so that

$$\tilde{f}^k(x) = \tilde{f}_p^k + \left\langle p^k, x - x^k \right\rangle \in \operatorname{conv}\left\{ f_j(x) : 1 \leqslant j \leqslant k \right\} \quad \text{for all } x,$$

and, hence, by (4),

$$\begin{aligned} f(x) &\geqslant \tilde{f}_p^k + \left\langle p^k, x - x^k \right\rangle \\ &= f_\varepsilon(x^k) + \varepsilon + \left\langle p^k, x - x^k \right\rangle - \left[f_\varepsilon(x^k) + \varepsilon - \tilde{f}_p^k \right] \\ &\geqslant f(x^k) + \left\langle p^k, x - x^k \right\rangle - \tilde{\alpha}_p^k \quad \text{for all } x. \end{aligned} \tag{20}$$

This shows that p^k is an $\tilde{\alpha}_p^k$-subgradient of f at x^k. (We may add that g^j is an α_j^k-subgradient of f at x^k.) Second, it is easy to deduce from (17)-(18) that (d^k, v^k) solves the reduced subproblem

$$\begin{aligned} &\text{minimize } \tfrac{1}{2}|d|^2 + v \text{ over all } (d, v) \in R^{N+1} \\ &\text{satisfying } -\tilde{\alpha}_p^k + \left\langle p^k, d \right\rangle \leqslant v, \end{aligned}$$

which corresponds to replacing $\hat{f}^k$ by $\tilde{f}^k$ in (8). Thus, in a sense, $(p^k, \tilde{f}_p^k)$ embodies all the past subgradient information which determined d^k.

To justify the stopping criterion, we note that (20), (18) and the Cauchy-Schwarz inequality imply

$$f(x) \geqslant f(x^k) - |v^k|^{1/2}|x - x^k| + v^k \quad \text{for all } x, \tag{21}$$

so that the algorithm's rules yield

$$f(x^k) \leqslant f(x) + \varepsilon_s^{1/2}|x - x^k| + \varepsilon_s \quad \text{for all } x$$

and $f_{\varepsilon^k}(x^k) \leqslant f(x^k) + \varepsilon^k$, with $\varepsilon^k \leqslant \varepsilon_s/5$, upon termination at Step 3; in particular, x^k is optimal if $\varepsilon_s = 0$.

Our rules for increasing the accuracy of objective evaluations stem from the criteria of Step 4, which is always entered with $v^k < 0$ and

$$5\varepsilon^k \leqslant -(m_R - m_L)v^k. \tag{22}$$

Criterion (13) ensures that the sequence $\{f(x^k)\}$ is nonincreasing, since

$$(23) \qquad f(x^{k+1}) \leqslant f_{\varepsilon^k}(y^{k+1}) + \varepsilon^k \leqslant f_{\varepsilon^k}(x^k) - \varepsilon^k + m_L v^k \leqslant f(x^k) + m_L v^k$$

if $t_L^k > 0$. Thus each serious step leads to a significant decrease in the objective value. On the other hand, if (13) does not hold, then $x^{k+1} = x^k$ and

$$-\alpha_{k+1}^{k+1} + \langle g^{k+1}, d^k \rangle = f_{\varepsilon^k}(y^{k+1}) - f_{\varepsilon^k}(x^k) - 3\varepsilon^k \geqslant m_L v^k - 5\varepsilon^k$$

if $\varepsilon^{k+1} = \varepsilon^k$, and, hence, by (22),

$$(24) \qquad -\alpha_{k+1}^{k+1} + \langle g^{k+1}, d^k \rangle \geqslant m_R v^k.$$

Since $m_R v^k > v^k$ and $k+1 \in J^{k+1}$, we deduce from (24) that (d^k, v^k) cannot solve the $(k+1)$st subproblem (9). Thus each null step results in a significant modification of the next search-direction-finding subproblem.

If one chooses $\hat{J}^k = \varnothing$, i.e., $J^k = \{k\}$ for all k, then only two past subgradients are used for each search direction finding. Of course, using more than two subgradients will increase the speed of convergence, but at the cost of increased storage and work per iteration. This flexibility may be exploited in particular applications. In fact, at any iteration one may calculate more than one ε-subgradient, especially for functions of the form (2), and use such additional subgradient information for the next search direction finding. Our global convergence analysis covers such modifications.

We end this section by commenting on the relationship of our method with other existing algorithms. For $\varepsilon^1 = 0$, i.e., $\varepsilon^k = 0$ for all k, Algorithm 2.1 reduces to the aggregate subgradient method of Kiwiel [5], while if, in addition, $J^k = \{1, \ldots, k\}$ for all k, then we obtain the descent method of Lemarechal [7] (see [5]). The method of Vasilyev [2] uses subproblems of the form (9) or, equivalently, (16), but neglects the linearization errors α_j^k and α_p^k by setting them all to zero. This unnecessary distortion of polyhedral approximations to f necessitates periodic restarts of the method from the current point, after which no significant progress occurs until the method accumulates new subgradient information.

3. Convergence. In this section we establish global convergence of the method.

THEOREM 3.1. *Suppose that Algorithm* 2.1 *generates a sequence* $\{x^k\}$ *with the stopping parameter* ε_s *set to zero. Then*:

(i) *If the method terminates at the kth iteration, then* x^k *minimizes* f.

(ii) *If the method cycles infinitely many times between Steps* 1 *and* 2 *at the kth iteration, then* x^k *minimizes* f.

(iii) *If* $\{x^k\}$ *is infinite, then* $f(x^k) \downarrow \inf\{f(x): x \in R^N\}$. *Moreover,* $\{x^k\}$ *converges to a minimum point of* f *if* f *attains its infimum.*

Proof. (i) The first assertion was proved in Section 2.

(ii) If the method cycles infinitely between Steps 1 and 2, then ε and v^k converge to zero, because ε is halved at Step 2, $m_R - m_L > 0$, and $-v^k = |p^k|^2 + \tilde{\alpha}_p^k \geqslant 0$, since $\tilde{\alpha}_p^k \geqslant 0$ from (20). Hence, (21) yields that x^k minimizes f.

(iii) Suppose that $\{x^k\}$ is infinite. We shall now show that if $\bar{x}$ is an accumulation point of $\{x^k\}$, then $\bar{x}$ minimizes f. First, suppose that $\{x^{k(l)}\}_{l \in L} \to \bar{x}$ for some infinite set $L \subset \{1, 2, \ldots\}$. Then we may let $K = \{k(l+1) - 1: l \in L\}$ (so that

$x^k \overset{K}{\to} \bar{x}$ and $t_L^k = 1$ for all $k \in K$) to deduce from (23) and the continuity of f [2, p.43] that $f(x^k) \downarrow f(\bar{x})$ and $v^k \overset{K}{\to} 0$. Passing to the limit in (21) with $k \in K$, we obtain $f(x) \geqslant f(\bar{x})$ for all $x \in R^N$, as desired. Second, suppose that there are only finitely many serious steps; i.e., $x^k = \bar{x}$ for all large k. If we have $\varepsilon^k \downarrow 0$, then part (ii) above shows that $\bar{x}$ minimizes f. Suppose, for contradiction purposes, that we have constant $\varepsilon^k = \varepsilon > 0$ for all large k. Let w^k denote the optimal value of subproblem (16), for any k, so that

$$w^k = \tfrac{1}{2}|p^k|^2 + \tilde{\alpha}_p^k, \tag{25a}$$

$$w^k \leqslant \min\left\{\tfrac{1}{2}|\mu g^k + (1-\mu)p^{k-1}|^2 + \mu\alpha_k^k + (1-\mu)\alpha_p^k : 0 \leqslant \mu \leqslant 1\right\} \tag{25b}$$

from (10)–(12), (19) and the fact that $k \in J^k$. Since $g^{k+1} \in \partial_\varepsilon f(y^{k+1})$ with $y^{k+1} = \bar{x} + d^k$ for large k, the local boundedness of $\partial_\varepsilon f(\cdot)$ [2, p. 78] implies that $\{g^k\}$ will be bounded if $\{d^k\}$ is. Moreover, we have $\alpha_p^{k+1} = \tilde{\alpha}_p^k$, since $x^{k+1} = x^k$ and $\varepsilon^k = \varepsilon$, for large k. Therefore, one may proceed as in the proof of Theorem 3.5 in [5] to deduce from (25), (17), (18) and (24) the existence of a constant $C < +\infty$ such that

$$0 \leqslant w^{k+1} \leqslant w^k - (1 - m_R)^2 (w^k)^2 / 8C^2$$

with fixed $m_R \in (0,1)$, for all large k. This implies $w^k \downarrow 0$, and, hence, $v^k \to 0$ from (18), (25a) and the nonnegativity of $\tilde{\alpha}^k$. But then $\varepsilon > -(m_R - m_L)v^k/5$ for large k, which contradicts the rules of Step 2. Thus $\bar{x}$ must minimize f, if it exists. In view of this result, one may use relations (20) and (23) to complete the proof as in Section 3 of [5] by deducing that $\{x^k\}$ converges if $f(x^k) \geqslant f(\tilde{x})$ for some fixed $\tilde{x} \in R^N$ and all k.

We omit the proof of the following result, since it may be derived from the proof of Lemma 3.9 in [5] by using relation (22).

PROPOSITION 3.2. *If the method generates an infinite sequence* $\{x^k\}$, *then either* $f(x^k) \downarrow -\infty$ *or* $v^k \to 0$ *and* $\varepsilon^k \to 0$ *as* $k \to \infty$.

COROLLARY 3.3. *If* f *is bounded from below and the stopping parameter* ε_s *is positive, then the method will terminate in a finite number of iterations, producing an approximately optimal point.*

4. Modifications. In this section we briefly discuss possible computational modifications of the method.

In Algorithm 2.1 each decrease of ε requires a new calculation of $f_\varepsilon(x^k)$ and d^k. To save this work, one may additionally decrease ε on entering Step 4, e.g., by replacing ε with $\kappa\varepsilon$ if ε is close to $-(m_R - m_L)v^k/5$, where $\kappa \in (0,1)$ is a parameter. It can be verified that such modifications do not impair the preceding convergence results.

Suppose that relation (1) is replaced by

$$f(x) - \varepsilon \leqslant f_\varepsilon(x) \leqslant f(x),$$

which may be justified when f is of the form (2). Then one may delete the factor 2 in (3), (13) and (14) and the "$+\varepsilon$" from (10) and (12), thus increasing the accuracy of the polyhedral model of f. In this case Step 2 may use the test $\varepsilon > -(m_R - m_L)v^k/3$ instead of the previous one. The preceding convergence results remain valid.

5. Conclusions. We have presented a readily implementable method for minimizing nonsmooth convex functions. It may be viewed as an extension of [5] to the case when objective and subgradient evaluations are subject to controllable errors. Relaxed accuracy requirements on evaluations allow one to save work at the initial stages of calculations, while their automatic tightening at later iterations ensures global convergence with no additional assumptions on the objective function that are typically required by other methods [1], [2], [9].

The method can be extended to constrained problems [4]. Preliminary computational experience [4] indicates that the algorithm is promising. However, much more work remains to be done before all its possible implementations are fully explored. We hope to pursue this subject in the near future.

Acknowledgment. The author would like to thank the referees for several helpful suggestions.

Systems Research Institute
Polish Academy of Sciences
Newelska 6
01-447 Warsaw, Poland

1. A. AUSLENDER, *Programmation Convexe avec Erreurs: Methodes d'ε-sous Gradients*, IX International Symposium on Mathematical Programming (Abstracts), Budapest, 1976.

2. V. F. DEMYANOV & L. V. VASILYEV, *Nondifferentiable Optimization*, Nauka, Moscow, 1981. (Russian)

3. R. GABASOV & F. M. KIRILOVA, *Methods of Linear Programming, Part 3, Special Problems*, Izdatel'stvo BGU, Minsk, 1980. (Russian)

4. K. C. KIWIEL, *Efficient Algorithms for Nonsmooth Optimization and Their Applications*, Ph.D. Thesis, Technical University of Warsaw, Warsaw, Poland, 1983. (Polish)

5. K. C. KIWIEL, "An aggregate subgradient method for nonsmooth convex minimization," *Math. Programming*, v. 27, 1983, pp. 320–341.

6. K. C. KIWIEL, "A linearization algorithm for nonsmooth minimization," *Math. Oper. Res.* (To appear.)

7. C. LEMARECHAL, *Nonsmooth Optimization and Descent Methods*, Report No. RR-78-4, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1978.

8. R. MIFFLIN, "A modification and an extension of Lemarechal's algorithm for nonsmooth minimization," *Nondifferential and Variational Techniques in Optimization* (D. C. Sorensen and R. J.-B. Wets, eds.), Mathematical Programming Study 17, North-Holland, Amsterdam, 1982, pp. 77–90.

9. B. T. POLYAK, "A general method for solving extremal problems," *Dokl. Akad. Nauk SSSR*, v. 174, 1967, pp. 33–36; English transl. in *Soviet Math. Dokl.*, v. 8, 1967, pp. 593–597.

10. N. Z. SHOR, *Methods for Minimizing Nondifferentiable Functions and Their Applications*, Naukova Dumka, Kiev, 1979. (Russian); English transl., Springer-Verlag, Berlin, 1985.

11. P. WOLFE, "A method of conjugate subgradients for minimizing nondifferentiable functions," *Nondifferentiable Optimization* (M. Balinski and P. Wolfe, eds.), Mathematical Programming Study 3, North-Holland, Amsterdam, 1975, pp. 145–173.