International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

Randomized Search Directions in Descent Methods for Minimizing Certain Quasi-Differentiable Functions

Kiwiel, K.

IIASA Collaborative Paper
February 1984

RANDOMIZED SEARCH DIRECTIONS IN DESCENT METHODS FOR MINIMIZING CERTAIN QUASI-DIFFERENTIABLE FUNCTIONS

Krzysztof C. Kiwiel*

December 1984
CP-84-56

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

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

PREFACE

Several descent methods have recently been proposed for minimizing smooth compositions of max-type functions. The methods generate many search directions at each iteration. It is shown here that a random choice of only two search directions at each iteration suffices to retain convergence to inf-stationary points with probability 1. Use of this technique may significantly decrease the effort involved in quadratic programming and line searches, thus allowing efficient implementations of the methods.

This paper is a contribution to research on non-smooth optimization currently underway in the System and Decision Sciences Program.

A.B. Kurzhanskii

Chairman

System and Decision
Sciences Program

1. Introduction

We are concerned with methods for minimizing a nondifferentiable and nonconvex function $f : R^N \to R$ of the form

$$f(x) = g(x, \max_{j \in J_1} h_{j1}(x), \ldots, \max_{j \in J_M} h_{jM}(x)), \tag{1.1}$$

where the functions $g : R^N \times R^M \to R$ and $h_{ji} : R^N \to R$ are continuously differentiable, and $I := \{1, \ldots, M\}$ and J_i, $i \in I$, are nonempty finite sets of indices. Such functions abound in applications (e.g. minimax problems, l_1 and l_∞ approximation problems, exact penalty methods) and have been studied in several papers; see, for instance, Auslender (1981), Ben-Tal and Zowe (1982), Bertsekas (1977), Fletcher (1981), Papavassilopoulos (1981).

Most of the past works assumed that the function $g(x, y_1, \ldots, y_M)$ is nondecreasing with respect to each y_i, $i \in I$. In this case the derivative

$$f'(x;d) = \lim_{t \downarrow 0} [f(x + td) - f(x)]/t$$

of f at x in a direction $d \in R^N$ is a convex function of d, and this facilitates the development of both necessary optimality conditions (Ben-Tal and Zowe (1982)) and descent methods (Auslender (1981), Kiwiel (1984a), Fletcher (1981)). The approach of Bertsekas (1977) and Papavassilopoulos (1981), which is based on augmented Lagrangians, requires some other assumptions which may be difficult to verify a priori.

When $g(x, \cdot)$ fails to preserve order, $f'(x;d)$ can be expressed as a difference of two convex functions of d (Demyanov and Rubinov (1983)), and hence $f(x+d)-f(x)$ cannot be approximated by just one simple convex function of d. Therefore the descent methods of Demyanov et al. (1983) and Kiwiel (1984b) construct at each iteration several convex models of $f(x+\cdot)-f(x)$ for finding several search directions. Then line searches along all the directions produce the next approximation to a solution.

Of course, calculating many search directions through quadratic programming may require much work. Also performing several one-dimensional minimizations (Demyanov et al. (1983)) requires many function evaluations, even though this effort can be

decreased if Armijo-type contractions are used (Kiwiel, 1984b).

This paper shows that a random choice of only two search direction finding subproblems among the candidate subproblems at each iteration suffices for retaining with probability 1 (w.p. 1) convergence of descent methods to inf-stationary points of f, i.e. points $\bar{x}$ satisfying the necessary condition of minimality

$$f'(\bar{x};d) \geq 0 \quad \text{for all} \quad d \in R^N.$$

Clearly, employing only two search directions at each iteration may decrease significantly the work involved in quadratic programming and line searches of the methods in Demyanov et al. (1983) and Kiwiel (1984b), thus enabling their efficient implementations.

It is worth observing that the ideas of this paper may be readily incorporated in the methods of Demyanov et al.(1983) and Kiwiel (1984 c) for solving constrained minimization problems with functions of the form (1.1), or with pointwise maxima of such functions. We hope, therefore, that the technique of randomization introduced here will prove useful in implementing many other algorithms for quasidifferentiable optimization. We intend to pursue this subject, including numerical experiments, in the near future.

The paper is organized as follows. In Section 2 we modify the algorithm of Kiwiel (1984 b). Its convergence w.p.1 is established in Section 3. Section 4 describes randomized curvilinear searches. Finally, we have a conclusion section

R^N denotes the N-dimensional Euclidean space with the usual inner product $\langle \cdot,\cdot \rangle$ and the associated norm $|\cdot|$. Superscripts are used to denote different vectors, e.g. x^1 and x^2. All vectors are row vectors.

2. Derivation of the method

In order to make the paper more self-contained, we shall now review the method of Kiwiel (1984b).

The heart of the method is the model of $f(x+td)-f(x)$ for predicting the effect of moving from a point $x \in R^N$ to the next

point $x+td$ along a direction $d \in R^N$ with a stepsize $t > 0$. We start, therefore, by recalling the properties of $f'(x;d)$ (see, e.g. Demyanov and Rubinov (1983) for details). We shall use the following notation

$$h_i(x) = \max_{j \in J_i} h_{ji}(x) \quad \text{for} \quad i \in I, \tag{2.1}$$

$$h(x) = (h_1(x), \ldots, h_M(x)),$$

$$f(x) = g(x, h(x)).$$

For $z=(x,y) \in R^N \times R^M$ we denote by $\nabla g(x,y)$ the N-vector $(\frac{\partial g}{\partial z_1}(z), \ldots, \frac{\partial g}{\partial z_N}(z))$, while $\frac{\partial g}{\partial y_i}(x,y)$ denotes $\frac{\partial g}{\partial z_{i+N}}(z)$, $i \in I$. Let

$$a_i(x) = \frac{\partial g}{\partial y_i}(x, h(x)) \quad \text{for al} \quad x \in R^N, \; i \in I,$$

$$b(x) = \nabla g(x, h(x)) \quad \text{for all } x.$$

Then from Taylor's expansion

$$f'(x;d) = \langle b(x), d \rangle + \sum_{i \in I} a_i(x) h_i'(x;d) =$$

$$= \langle b(x), d \rangle + \sum_{i \in I} a_i(x) \max_{j \in J_i(x)} \langle \nabla h_{ji}(x), d \rangle,$$

so that

$$f'(x;d) = \langle b(x), d \rangle + \sum_{i \in I_+(x)} \max_{j \in J_i(x)} \langle a_i(x) \nabla h_{ji}(x), d \rangle +$$

$$+ \sum_{i \in I_-(x)} \min_{j \in J_i(x)} \langle a_i(x) \nabla h_{ji}(x), d \rangle,$$

where

$$J_i(x) = \{j \in J_i : h_{ji}(x) = h_i(x)\}, \; i \in I,$$

$$I_+(x) = \{i \in I : a_i(x) > 0\},$$

$$I_-(x) = \{i \in I : a_i(x) < 0\}$$

and the summation over an empty index set yields zero. Therefore

$$f'(x;d) = \max_{v \in A(x)} \langle v, d \rangle + \min_{w \in B(x)} \langle w, d \rangle,$$

where

$$A(x) = \{v : v=b(x)+ \sum_{i \in I_+(x)} a_i(x)\nabla h_{ji}(x) \quad \text{for some} \quad j \in J_i(x)\},$$

$$B(x) = \{w : w= \sum_{i \in I_-(x)} a_i(x)\nabla h_{ji}(x) \quad \text{for some} \quad j \in J_i(x)\}. \quad (2.2)$$

Observe that, in general, $f'(\cdot,d)$ is discontinuous because $A(\cdot)$ and $B(\cdot)$ may change abruptly if so do $J_i(\cdot)$. Changes in $I_+(\cdot)$ and $I_-(\cdot)$ do not introduce discontinuities in $f'(\cdot;d)$, since each i may enter or leave $I_+(\cdot)$ or $I_-(\cdot)$ only with $a_i(\cdot)=0$, whereas $b(\cdot)$, $a_i(\cdot)$ and $\nabla h_{ji}(\cdot)$ are continuous.

Let us now analyze algorithmic implications of the discontinuity of $f'(\cdot;d)$. Suppose that our algorithm has arrived at some point x close to a non-stationary point $\bar{x}$ satisfying

$$f(\bar{x};\bar{d}) < 0 \quad \text{for some} \quad \bar{d}. \quad (2.3)$$

In order for the algorithm not to jam up around $\bar{x}$, it should be able to find a direction d ("close" to $\bar{d}$, say) and a stepsize $t > 0$ such that it can move away from $\bar{x}$ to the next point $x+td$ with a significantly lower objective value. To this end, since (2.3) is equivalent to

$$\max_{v \in A(\bar{x})} \langle v,\bar{d} \rangle + \langle \bar{w},\bar{d} \rangle < 0 \quad \text{for some} \quad \bar{d} \in R^N, \ \bar{w} \in B(\bar{x}), \quad (2.4)$$

the algorithm needs at x some model for approximating the value of

$$\max_{v \in A(\bar{x})} \langle v,d \rangle + \langle w,d \rangle \quad \text{for} \quad w \in B(\bar{x}) \quad (2.5)$$

as a function of $d \in R^N$. Clearly, $f'(x;\cdot)$ can hardly serve as such a model, since it depends only on $A(x)$ and $B(x)$, which may represent only part of $A(\bar{x})$ and $B(\bar{x})$ even when x is close to $\bar{x}$.

For these reasons, the algorithm of Kiwiel (1984b) approximates (2.5) with the family of functions

$$\hat{f}(d;x,w,\delta) = \langle b(x),d \rangle + \sum_{i \in I_+(x)} a_i(x) \max_{j \in J_i(x,\delta)} [h_{ji}(x)-h_i(x) + \langle \nabla h_{ji}(x),d \rangle] + \langle w,d \rangle \quad \text{for all } d$$

parametrized by w in

$$B(x,\delta) = \{w : w= \sum_{i \in I_-(x)} a_i(x)\nabla h_{ji}(x), \ j \in J_i(x,\delta)\}, \quad (2.6)$$

where the use of

$$J_i(x,\delta) = \{j \in J_i : h_{ji}(x) \geq h_i(x)-\delta\}$$

with a fixed "anticipation" tolerance $\delta > 0$ may predict changes of $J_i(\cdot)$ around x. Indeed, by continuity, we have $J_i(\bar{x}) \subset J_i(x,\delta)$ if x is close to $\bar{x}$. Note that each $\hat{f}(d;x,w,\delta)$ with $w \in B(x)$ approximates $f'(x;d)$ from above. Also the models $\hat{f}(d;x,w,\delta)$ yield correct approximations to (2.5) when x is close to $\bar{x}$ and $|d|$ is small, since for such d the terms involving $j \in J_i(\bar{x}) \setminus J_i(x,\delta)$ may be neglected.

In order to "anticipate" (2.4), the algorithm finds for each $w \in B(x,\delta)$ a direction $d(w)$ to

$$\text{minimize } \hat{f}(d;x,w,\ ) + \tfrac{1}{2}|d|^2 \quad \text{over all } d \in R^N, \tag{2.7}$$

where the term $|d|^2/2$ ensures that $d(w)$ stays in the region where $\hat{f}(\cdot;x,w,\delta)$ may be close to $f(x+\cdot)-f(x)$. Indeed, $|d(w)|$ cannot be very large, since

$$d(w) = -\Big[b(x) + \sum_{i \in I_+(x)} a_i(x) \sum_{j \in J_i(x,\delta)} \lambda_{ji}(w) \nabla h_{ji}(x)\Big] \tag{2.8a}$$

for some

$$\lambda_{ji}(w) \geq 0 \quad \text{for } j \in J_i(x,\delta), \quad \sum_{j \in J_i(x,\delta)} \lambda_{ji}(w) = 1, \text{ for } i \in I_+(x) \tag{2.8b}$$

(see, e.g. Kiwiel (1984a)).

Note that each $d(w)$ with $w \in B(x)$ is a descent direction for f at x if $d(w) \neq 0$, since

$$\hat{f}(d(w);x,w,\delta) + \tfrac{1}{2}|d(w)|^2 \leq f(0;x,w,\delta) + \tfrac{1}{2}|0|^2 = 0,$$

so that $f'(x;d(w)) \leq \hat{f}(d(w);x,w,\delta) < 0$. Of course, for $w \in B(x,\delta) \setminus B(x)$ we may have $f(x+td(w)) > f(x)$ for all small $t > 0$. However, for larger t it may happen that $f(x+td(w)) < f(x)$ when w becomes close to $B(x+td(w))$. Therefore, the method of Kiwiel (1984b) searches for a stepsize t by computing $f(x+td(w))$ for all $w \in B(x,\delta)$. We shall now describe a modification which uses only two search directions.

Algorithm 2.1.

Step 0 (Initialization). Select a starting point $x^1 \in R^N$, an anti-

cipation tolerance $\delta > 0$ and a line search parameter $m > 0$. Set $k=1$.

Step 1 (Descent direction finding). For each $w \in B(x^k)$, find $d(w)$ from the solution $(d(w);u_i(w),\ i \in I_+(x^k))$ to the quadratic programming subproblem with $x=x^k$

$$\min_{d,u_i} \tfrac{1}{2}|d|^2 + \langle b(x),d\rangle + \sum_{i \in I_+(x)} a_i(x)u_i + \langle w,d\rangle, \tag{2.9}$$

$$\text{s.t.}\quad h_{ji}(x)-h_i(x)+\langle \nabla h_{ji}(x),d\rangle \le u_i \quad \text{for} \quad j \in J_i(x,\delta),\ i \in I_+(x).$$

Step 2 (Stopping criterion). If $d(w)=0$ for all $w \in B(x^k)$, terminate. Otherwise, set $B^k=\{w\}$ for some w such that $d(w)\neq 0$, and continue.

Step 3 (Additional direction finding). Draw w at random from $B(x^k,\delta) \setminus B^k$ according to a uniform distribution. Find $d(w)$ by solving (2.9). Augment B^k with w and set

$$u^k = -\max\{|d(w)|^2 : w \in B^k\}. \tag{2.10}$$

Step 4 (Stepsize selection). (i) Set $t=1$.

(ii) Find w in B^k that yields the smallest value of $f(x^k + td(w))$.

(iii) If

$$f(x^k+td(w)) \le f(x^k)+m(t)^2u^k,$$

set $t^k=t$, $x^{k+1}=x^k+td(w)$ and go to Step 5; otherwise, replace t by $t/2$ and go to Step 4(ii).

Step 5. Increase k by 1 and go to Step 1.

The algorithm cannot cycle infinitely at Step 4, since Step 4 is always entered with $\tilde{w} \in B(x^k)$ such that $f'(x^k;d(\tilde{w})) < 0$. Hence $t \downarrow 0$ would lead to

$$f'(x^k;d(\tilde{w})) \ge \liminf_{t\downarrow 0} \left[\min_{w \in B^k} f(x^k+td(w))-f(x^k)\right]/t \ge \lim_{t\downarrow 0} mtu^k = 0,$$

a contradiction.

If we computed $d(w)$ for all $w \in B(x^k,\delta)$ and replaced B^k by $B(x^k,\delta)$ in the algorithm, we would obtain the method of

Kiwiel (1984b). Since

$$|B(x,\delta)| = \prod_{i \in I_-(x)} |J_i(x,\delta)|$$

can be large even when each $|J_i(x,\delta)|$ is small, using only two search directions may decrease the computational effort by a large factor.

In order to better understand the algorithm, consider the example

$$f(x) = (x)^3 - \max\{0,-x\} \quad \text{for} \quad x \in R$$

with $x^1=0.1$, $\delta=+\infty$ and $m=0.1$. If the algorithm used only $B^k=\{0\}$ for all k (as it would if δ were zero), then we would have $d(0)=-3(x^k)^2$ with x^k converging to $\bar{x}=0$, which is nonstationary. However, even one occurance of $B^k=\{0,1\}$ produces $d(1)=-(1+3(x^k)^2)$, which enables the algorithm to "jump" over $\bar{x}=0$ to $x^{k+1}<0$, and then continue with $x^k \to -\infty$.

3. Convergence

In this section we shall establish global convergence of the algorithm w.p.1. In the absence of convexity, we will content ourselves with finding an inf-stationary point for f.

We start by recalling from Kiwiel (1984b) the properties of search directions generated around nonstationary points.

Lemma 3.1. Suppose that $\bar{x} \in R^N$, $\bar{w} \in B(\bar{x})$ and $\bar{d} \in R^N$ are such that $\hat{f}(\bar{d};\bar{x},\bar{w},0) < 0$. Then there exist $\bar{\varepsilon} > 0$ and neighborhoods $S(\bar{x})$ and $S(\bar{w})$ of $\bar{x}$ and $\bar{w}$, respectively, such that

$$f'(\bar{x};d(x,w)) \le -\bar{\varepsilon} \quad \text{for all} \quad x \in S(\bar{x}),\ w \in S(\bar{w}), \tag{3.1}$$

$$|d(x,w)| \ge \bar{\varepsilon} \quad \text{for all} \quad x \in S(\bar{x}),\ w \in S(\bar{w}), \tag{3.2}$$

where $d(x,w)$ denotes the solution of (2.7).

In particular, since $f'(\bar{x};\bar{d}) \le \hat{f}(\bar{d};\bar{x},w,0)$ for $w \in B(\bar{x})$, the above lemma shows that the algorithm finds at least one descent direction for f at x^k if and only if x^k is nonstationary. Hence we have

Lemma 3.2. Algorithm 2.1 terminates at the k-th iteration if and only if x^k is inf-stationary for f.

Our main result is

Theorem 3.3. Every accumulation point of an infinite sequence $\{x^k\}$ generated by Algorithm 2.1 is inf-stationary for f w.p.1.

Proof. Strictly speaking, each sequence $\{x^k\}$ generated by the algorithm should be considered as a realization (trajectory) of a random process with discrete time defined on a suitable probability space. For brevity, we shall, however, supress the dependence of $\{x^k\}$ on elementary events.

Suppose that there exist $\bar{x} \in R^N$ and an infinite set $K \subset \{1,2,\ldots\}$ such that $x^k \xrightarrow{K} \bar{x}$. For contradiction purposes, assume that $\bar{x}$ is nonstationary. By Lemma 3.1, there exist $\bar{w} \in B(\bar{x})$ and $\bar{\varepsilon} > 0$ such that (3.1) and (3.2) hold for some $S(\bar{x})$ and $S(\bar{w})$. Since $x^k \xrightarrow{K} \bar{x}$ and $\delta > 0$ is fixed, an elementary continuity argument based on (2.6) implies that

$$B(x^k,\delta) \cap S(\bar{w}) \neq 0 \quad \text{for all large} \quad k \in K,$$

so there exist $w^k \in B(x^k,\delta)$ and $d^k = d(x^k,w^k)$ such that

$$f'(\bar{x};d^k) \leq -\bar{\varepsilon} \quad \text{for all large} \quad k \in K, \tag{3.3}$$

$$|d^k| \geq \bar{\varepsilon} \quad \text{for all large} \quad k \in K. \tag{3.4}$$

Let n_B be such that $|B(x,\delta)| \leq n_B$ for all x. Since n_B is finite and $x^k \xrightarrow{K} \bar{x}$, (2.8) implies the existence of $\bar{u} < 0$ such that for all $k \in K$ one has $\bar{u} \leq -|d(x^k,w)|^2 \leq 0$ for all $w \in B(x^k,\delta)$. Then $\bar{u} \leq u^k \leq 0$ for all $k \in K$ from (2.10). Moreover, $\{d^k\}_{k \in K}$ is bounded, so one may use Taylor's expansion as in Demyanov et al. (1983) to show that

$$f(\bar{x} + td^k) \leq f(\bar{x}) + tf'(\bar{x};d^k) + o(t,k),$$

where $o(t,k)/t \to 0$ as $t \downarrow 0$ uniformly with respect to $k \in K$. Hence, by (3.3), for any fixed $\hat{\varepsilon} \in (0,\bar{\varepsilon})$ there is $t(\hat{\varepsilon}) > 0$ such that

$$f(\bar{x}+td^k) \leq f(\bar{x}) - \hat{\varepsilon}t \quad \text{for all} \quad t \in [0,t(\hat{\varepsilon})] \quad \text{and large} \quad k \in K. \tag{3.5}$$

Next, since $x^k \xrightarrow{K} \bar{x}$, $\{d^k\}_{k\in K}$ is bounded and f is continuous, for any $\varepsilon > 0$ we have

$$f(x^k + td^k) - f(x^k) \le f(\bar{x} + td^k) - f(\bar{x}) + \varepsilon \tag{3.6}$$

for all $t \in [0, t(\hat{\varepsilon})]$ and large $k \in K$. Let us choose ε such that the interval $[\underline{t}(\varepsilon), \bar{t}(\varepsilon)]$ of solutions to the inequality

$$\varepsilon - \hat{\varepsilon}t \le m(t)^2\bar{u} \tag{3.7}$$

contains $1/2^i$ for some $i > 0$. This is possible, since $[\underline{t}(\varepsilon), \bar{t}(\varepsilon)] \to [0, -\varepsilon/m\bar{u}]$ as $\varepsilon \downarrow 0$. Then $\bar{t} = 1/2^i$ satisfies, by (3.5)-(3.7) and the fact that $\bar{u} \le u^k$ for $k \in K$,

$$f(x^k + \bar{t}d^k) \le f(x^k) + m(\bar{t})^2 u^k \quad \text{for all large } k \in K. \tag{3.8}$$

Suppose that $w^k \in B^k$ for infinitely many $k \in K$. For such k, (3.4) and (2.10) yield

$$-u^k \ge |d^k|^2 \ge \bar{\varepsilon}^2, \tag{3.9}$$

whereas (3.8) and the construction of $t^k > \bar{t}$ imply

$$f(x^{k+1}) \le f(x^k + t^k d^k) \le f(x^k) + m(t^k)^2 u^k \le f(x^k) + m(\bar{t})^2 u^k. \tag{3.10}$$

Clearly, (3.9) and (3.10) cannot hold simultaneously for infinitely many k, since $f(x^k) \downarrow f(\bar{x})$ from the continuity of f and the fact that $x^k \xrightarrow{K} \bar{x}$ with $f(x^{k+1}) < f(x^k)$ for all k.

Thus we need only consider the case when $w^k \in B(x^k, \delta) \setminus B^k$ for all large $k \in K$. But this event has probability 0, since for each $k \in K$ the probability that w^k enters B^k at Step 3 is not less than $1/n_B$. Therefore, $\bar{x}$ is inf-stationary w.p.1.

4. Modifications

Step 1 of Algorithm 2.1 requires the solution of $|B(x^k)|$ quadratic programming subproblems in order to find just one descent direction. Since $|B(x^k)|$ may be large, in general, we shall now show how to reduce this effort. To this end, we need the following result.

Lemma 4.1. Let $X_B = \{x \in R^N : |B(x)| = 1\}$. Then X_B is of full Lebesque measure in R^N.

Proof. General properties of functions of the form (2.1)(see, e.g. Rockafellar,(1982)) imply that the set $\{\nabla h_{ji}(x): j \in J_i(x)\}$ is a singleton for almost all x, for each $i \in I$. Hence (2.2) yields the desired conclusion.

We conclude from the above lemma that if $\{x^k\} \subset X_B$ then $|B(x^k)|=1$ for all k. We proceed, therefore, to show how to ensure that $\{x^k\} \subset X_B$ w.p. 1.

For any x and d in R^N, consider the family of arcs

$$C^{\tilde{}}_{\tilde{d}} = \{y \in R^N : y=x+td+(t)^2\tilde{d},\ t \in [0,1]\}$$

parametrized by auxiliary directions $\tilde{d}$ in

$$D(r) = \{\tilde{d} \in R^N : |\tilde{d}| \leq r\},$$

where $r > 0$. Let a subset E of R^N have Lebesgue measure zero. Then it is not difficult to see that almost arcs $C^{\tilde{}}_{\tilde{d}}$ meet E in a set of zero one-dimensional measure. Applying this fact in the case where E is the complement of X_B, we deduce that for almost all $\tilde{d}$ in D(r) we have $|B(x+td+(t)^2\tilde{d})|=1$ for almost all t in $[0,1]$. Hence we propose the following randomized modification of Step 4, in which $r^k \in (0,0.1)$ is a small pertubation parameter.

Step 4' (Randomized stepsize selection). (i) Find $\tilde{d}^k=(\tilde{d}^k_1,\ldots,\tilde{d}^k_N)$ by drawing each $\tilde{d}^k_i$ from $[-r^k,r^k]$ according to a uniform distribution. Set t=1.

(ii) Draw t at random from $[-r^k,r^k]$ according to a uniform distribution. Replace t by $t(1+\tilde{t})$.

(iii) Find w in B^k that yields the smallest value of $f(x^k+td(w)+(t)^2\tilde{d}^k)$.

(iv) If $f(x^k+td(w)+(t)^2\tilde{d}^k) \leq f(x^k)+m(t)^2u^k$, set $t^k=t$, $\hat{d}^k=d(w)$, $x^{k+1}=x^k+t^k\hat{d}^k+(t)^2\tilde{d}^k$ and go to Step 5; otherwise, replace t by t/2 and go to Step 4'(ii).

In order to analyze Step 4', we note that f is locally Lipschitz continuous, since so are h_i (see, e.g. Rackafellar (1982)). Thus for each bounded neighborhood S(x) of a point $x \in R^N$ there exists a Lipschitz constant $L < \infty$ such that

$$|f(x')-f(x")| \leq L|x'-x"| \quad \text{for all } x',x" \in S(x).$$

Letting $x=x^k$ and recalling that $f'(x;d(\tilde{w}))<0$ for some $\tilde{w}\in B^k$ at Step 4, we see that the algorithm cannot cycle infinitely at Step 4, since $t\downarrow 0$ would give for $d=d(\tilde{w})$ and $\tilde{d}=\tilde{d}^k$

$$f'(x;d)=\lim_{t\downarrow 0}[f(x+td)-f(x)]/t =$$

$$=\lim_{t\downarrow 0}[f(x+td+(t)^2\tilde{d})-f(x)]/t+\lim_{t\downarrow 0}[f(x+td)-f(x+td+(t)^2\tilde{d})]/t =$$

$$\geq \lim_{t\downarrow 0} mtu^k + \lim_{t\downarrow 0} Lt|\tilde{d}| = 0,$$

a contradiction. Thus we conclude from the preceding results that Step 4′ produces $x^{k+1}\in X_B$ w.p. 1.

We may now establish convergence of the resulting method.

Theorem 4.2. Suppose that Algorithm 2.1 with Step 4′ generates an infinite sequence $\{x^k\}$ with perturbation parameters $r^k\downarrow 0$, starting from a point x^1 chosen at random according to some positive probability density on some ball in R^N. Then $|B(x^k)|=1$ for all k w.p. 1, and every accumulation point of $\{x^k\}$ is inf-stationary for f w.p. 1.

Proof. Of course, $x^1\in X_B$ w.p. 1 and hence, by the preceding results, $\{x^k\}\subset X_B$ w.p. 1. Thus the assertion can be established by introducing the following modifications in the last three paragraphs of the proof of Theorem 3.3.

Since $x^k \xrightarrow{K} \bar{x}$, $\{d^k\}_{k\in K}$ is bounded, $\tilde{d}^k\to 0$ and f is locally Lipschitz continuous, for any $\varepsilon>0$ we have

$$f(x^k+td^k+(t)^2\tilde{d}^k) - f(x^k) \leq f(\bar{x}+td^k) - f(\bar{x}) + \varepsilon$$

for all $t\in[0,t(\hat{\varepsilon})]$ if $k\in K$ is large enough, because

$$|f(x^k + td^k + (t)^2\,\tilde{d}^k) - f(\bar{x} + td^k) + f(\bar{x}) - f(x^k)| \leq$$

$$\leq L(|x^k - \bar{x}| + t(\hat{\varepsilon})^2|\tilde{d}^k| + |\bar{x} - x^k|),$$

where L is a Lipschitz constant of f around $\bar{x}$. Next, choose ε such that (3.7) holds for all $t\in T$, where $T=[1/2^{i+2},\ 1/2^i]$ for some $i>0$, and replace (3.8) by

$$f(x^k+td^k+(t)^2\tilde{d}^k) \leq f(x^k)+m(t)^2u^k \quad \text{for all } t\in T \text{ and large } k\in K.$$

Then for $\bar{t}=1/2^{i+2}$ we may replace (3.10) by

$$f(x^{k+1}) \leq f(x^k+t^k d^k+(t^k)^2\tilde{d}^k) \leq f(x^k)+m(\bar{t})^2 u^k,$$

since Step 4' decreases trial stepsizes by a factor of at most $2/(1+r^k)$ with $r^k \neq 0$. Hence the proof may be completed as before.

We conclude that in practice the modified algorithm will typically generate only two search directions at each iteration.

5. Conclusions

We have presented a randomized version of the method of Kiwiel (1984b) for minimizing smooth compositions of max-type functions. Our modifications may decrease significantly the work involved in quadratic programming and line searches.

A few words about possible extensions are in order. The first of our ideas, i.e. the random choice of only two search directions at each iteration, may be easily incorporated in the methods of Demyanov et al. (1983) and Kiwiel (1984c) for solving constrained problems with functions of the form (1.1) or with pointwise maxima of such functions, and in the algorithm of Kiwiel (1984d) for constrained maxminmax problems. The second concept, i.e. the use of only two randomized curvilinear searches at each iteration, is readily applicable to the algorithms of Kiwiel (1984c,1984d). Its use in the methods of Demyanov et al. (1983) would involve either introducing approximate minimizations along arcs, or employing the curvilinear searches of Section 4.

Of course, efficient and robust implementations of all these methods will require much work. We intend to pursue this subject in the near future.

References

Auslender A. (1981). Minimisation de fonctions localement Lipschitziennes: applications a la programmation mi-convexe, mi-differentiable. In: Nonlinear Programming 4 (O.L. Mangasarian, R.R. Mayer, and S.M. Robinson, eds.), pp.429-460, Academic Press, New York.

Ben-Tal A. and J. Zowe (1982). Necessary and sufficient optimality conditions for a class of nonsmooth minimization problems. Math. Programming 24, 70-91.

Bertsekas D. (1977). Approximation procedures based on the method of multipliers. J. Optim. Theory Appl. 23, 487-510.

Demyanov V.F., S. Gamidov and T.I. Sivelina (1983). An algorithm for minimizing a certain class of quasidifferentiable functions. WP-83-122, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Demyanov V.F. and A.M. Rubinov (1983). On quasidiffrentiable mappings. Math. Operat. Statistic, Ser. Optim. 14, 3-21.

Fletcher R. (1981). Practical Methods of Optimization, Vol.II, Constrained Optimization. Wiley, New York.

Kiwiel K.C. (1983). A phase I - phase II method for inequality constrained minimax problems. Control Cyb. 12, 55-75.

Kiwiel K.C. (1984a). A quadratic approximation method for minimizing a class of quasidifferentiable functions. Numer. Math. (to appear).

Kiwiel K.C. (1984b). A method of linearizations for minimizing certain quasidifferentiable functions. In: Quasidifferentiable Functions and Optimization (V.F. Demyanov and L.C. W. Dixon, eds.), pp. - , Mathematical Programming Study , North-Holland, Amsterdam (to appear).

Kiwiel K.C. (1984c). A method of feasible directions for certain quasidifferentiable inequality constrained minimization problems. Collaborative Paper, International Institute for Applied Systems Analysis, Laxenburg, Austria (to appear).

Kiwiel K.C. (1984d). An algorithm for maxminmax problems. Collaborative Paper, International Institute for Applied Systems Analysis, Laxenburg, Austria (to appear).

Papavassilopoulos G. (1981). Algorithms for a class of nondifferentiable problems. J. Optim. Theory Appl. 34, 31-82.

Rockafellar R.T. (1982). Favorable classes of Lipschitz continuous functions in subgradient optimization. CP-82-S8, Progress in Nondifferentiable Optimization (E. Nurminski, ed.), pp. 125-144, International Institute for Applied Systems Analysis, Laxenburg, Austria.