Discrete gradient method: Derivative-free method for nonsmooth optimization[1]

Adil M. Bagirov[2], Bülent Karasözen[3], Meral Sezer[4]

Communicated by F. Giannessi

[1]The authors are grateful to the anonymous referee for comments improving the quality of the paper. The first author is the recipient of an Australian Research Council Australian Research Fellowship (Project number: DP 0666061).

[2]Corresponding author, Research Fellow, a.bagirov@ballarat.edu.au, Centre for Informatics and Applied Optimization, School of Information Technology and Mathematical Sciences, University of Ballarat, Victoria 3353, Australia

[3]Professor, Department of Mathematics & Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey.

[4]PhD student, Department of Mathematics, Middle East Technical University, Ankara, Turkey.

Abstract

A new derivative-free method is developed for solving unconstrained nonsmooth optimization problems. This method is based on the notion of a discrete gradient. It is demonstrated that the discrete gradients can be used to approximate subgradients of a broad class of nonsmooth functions. It is also shown that the discrete gradients can be applied to find descent directions of nonsmooth functions. The preliminary results of numerical experiments with unconstrained nonsmooth optimization problems as well as the comparison of the proposed method with nonsmooth optimization solver DNLP from CONOPT-GAMS and derivative-free optimization solver CONDOR are presented.

Keywords: Nonsmooth optimization, Derivative-free optimization, Subdifferential, Discrete gradients.

1 Introduction

Consider the following unconstrained minimization problem:

$$\min \quad f(x) \quad \text{s. t.} \quad x \in \mathbb{R}^n \tag{1}$$

where the objective function f is assumed to be Lipschitz continuous.

Nonsmooth unconstrained optimization problems appear in many applications. Over more than four decades different methods have been developed to solve problem (1). The bundle-type methods (Refs. 1-8), algorithms based on smoothing techniques (Ref. 9) and the gradient sampling algorithm (Ref. 10) are among them.

In most of these algorithms at each iteration the computation of at least one subgradient or approximating gradient is required. However, there are many practical problems where the computation of even one subgradient is a difficult task. In such situations derivative free methods seem to be better choice since they do not use explicit computation of subgradients.

Among derivative free methods, the generalized pattern search methods are well-suited for nonsmooth optimization (Refs. 11,12). However, their convergence are proved under quite restrictive differentiability assumptions. It was shown in Ref. 12 that if the objective function f is continuously differentiable in $\mathbb{R}^n$ then the limit inferior of the norm of the gradient of the sequence of points generated by the generalized pattern search algorithm goes to zero. The paper Ref. 11 provides convergence analysis under less restrictive differentiability assumptions. It was shown that if f is strictly differentiable near the limit of any refining subsequence, the gradient at that point is zero. However, in many important practical problems the objective functions are not strictly differentiable at local minimizers.

In this paper, we develop a new derivative free method. First, we describe an algorithm for the approximation of the subgradients. Then we introduce the notion of a discrete gradient and prove that it can be used to approximate subdifferentials. We also describe an algorithm for the computation of the descent directions and study the convergence of the proposed method. Finally, we present the comparison of this method with the nonsmooth optimization solver, DNLP from GAMS and the derivative-free optimization solver CONDOR using results of numerical experiments.

The structure of the paper is as follows. Section 2 provides some preliminaries and Section 3 presents nonsmooth optimization formulation of the clustering problem. Approximation of subgradients is discussed in Section 4. Discrete gradients are introduced in Section 5. Section 6 presents an algorithm for the computation of a descent direction and Section 7 presents the discrete gradient method. Results of numerical experiments are given in Section 8. Section 9 concludes the paper.

2 Preliminaries

Let f be a locally Lipschitz continuous function defined on $\mathbb{R}^n$. It is differentiable almost everywhere and we can define for it a subdifferential (Ref. 13) by

$$\partial f(x) = \mathrm{co} \left\{ v \in \mathbb{R}^n : \exists (x^k \in D(f), x^k \to x, k \to +\infty) : v = \lim_{k \to +\infty} \nabla f(x^k) \right\},$$

here $D(f)$ denotes the set where f is differentiable, co denotes the convex hull of a set. The mapping $\partial f(x)$ is upper semicontinuous and bounded on bounded sets (Ref. 13). The generalized directional derivative of f at x in the direction g is defined as

$$f^0(x,g) = \limsup_{y\to x,\alpha\to+0} \alpha^{-1}[f(y+\alpha g) - f(y)].$$

For the locally Lipschitz function f this derivative exists and $f^0(x,g) = \max\{\langle v,g\rangle : v \in \partial f(x)\}$, where $\langle\cdot,\cdot\rangle$ stands for an inner product in $\mathbb{R}^n$. f is called a regular function on $\mathbb{R}^n$, if it is differentiable in any direction $g \in \mathbb{R}^n$ and $f'(x,g) = f^0(x,g)$ for all $x, g \in \mathbb{R}^n$, where $f'(x,g)$ is a derivative of the function f at the point x in the direction g. For a point x to be a local minimizer of a locally Lipschitz continuous function f on $\mathbb{R}^n$, it is necessary that $0 \in \partial f(x)$.

A function $f : \mathbb{R}^n \to \mathbb{R}^1$ is called semismooth at $x \in \mathbb{R}^n$, if it is locally Lipschitz continuous at x and for every $g \in \mathbb{R}^n$, the limit

$$\lim_{g'\to g,\alpha\to+0} \langle v,g\rangle, \quad v \in \partial f(x+\alpha g')$$

exists (Ref. 14). The semismooth function f is directionally differentiable. Consider the following set at a point $x \in \mathbb{R}^n$ with respect to a direction $g \in R^n$, $\|g\| = 1$:

$$R(x,g) = \mathrm{co}\left\{v \in \mathbb{R}^n : \exists (v^k \in \partial f(x+\lambda_k g),\ \lambda_k \to +0,\ k \to +\infty) :\ v = \lim_{k\to+\infty} v^k\right\}.$$

It follows from the semismoothness of f that $f'(x,g) = \langle v,g\rangle$ for all $v \in R(x,g)$.

Moreover, for any $\varepsilon > 0$ there exists $\lambda_0 > 0$ such that

$$\partial f(x + \lambda g) \subset R(x, g) + S_\varepsilon, \tag{2}$$

for all $\lambda \in (0, \lambda_0)$. Here $S_\varepsilon = \{v \in \mathbb{R}^n : \|v\| < \varepsilon\}$.

A function f is called quasidifferentiable at a point x, if it is locally Lipschitz continuous, directionally differentiable at this point and there exist convex, compact sets $\underline{\partial} f(x)$ and $\overline{\partial} f(x)$ such that:

$$f'(x, g) = \max\{\langle u, g\rangle : \ u \in \underline{\partial} f(x)\} + \min\{\langle v, g\rangle : \ v \in \overline{\partial} f(x)\}.$$

The set $\underline{\partial} f(x)$ is called a subdifferential, the set $\overline{\partial} f(x)$ is called a superdifferential and the pair $[\underline{\partial} f(x), \overline{\partial} f(x)]$ is called a quasidifferential of f at x (Ref. 15).

3 Data Clustering Problem

There are many problems where the objective and/or constraint functions are not regular. The cluster analysis problem is one of them. It is an important area in data mining. Clustering deals with the problems of organization of a collection of patterns into clusters based on similarity. In cluster analysis a finite set of points

$C = \{c^1, \dots, c^m\},\ c^i \in \mathbb{R}^n,\ i = 1, \dots, m$ is given. A partition clustering aims to distribute the points of the set C into a given number q of non-empty disjoint subsets $C^i,\ i = 1, \dots, q$ with respect to predefined criteria such that $C^i \bigcap C^j = \emptyset,\ i, j = 1, \dots, q,\ i \neq j$ and

$$C = \bigcup_{i=1}^{q} C^i.$$

The sets $C^i,\ i = 1, \dots, q$ are called clusters. The strict application of these rules is called hard clustering. We assume that each cluster $C^i,\ i = 1, \dots, q$ is identified by its center. In Refs. 16,17 the partition clustering is reduced to the following problem:

$$\min \ \ f(x^1, \dots, x^q) \quad \text{s. t. } (x^1, \dots, x^q) \in \mathbb{R}^{n \times q}, \tag{3}$$

where

$$f(x^1, \dots, x^q) = (1/m) \sum_{i=1}^{m} \min\{\|x^s - c^i\|^2 : \ s = 1, \dots, q\}. \tag{4}$$

Here $\|\cdot\|$ is the Euclidean norm and $x^s \in \mathbb{R}^n$ stands for s-th cluster center. If $q > 1$, the objective function (4) in problem (3) is nonconvex and nonsmooth. Moreover, it is non-regular. This function can be represented as the difference of two convex functions as follows: $f(x) = f_1(x) - f_2(x)$, where

$$f_1(x) = (1/m)\sum_{i=1}^{m}\sum_{s=1}^{q}\|x^s - c^i\|^2, \qquad f_2(x) = (1/m)\sum_{i=1}^{m}\max_{s=1,\dots,q}\sum_{k=1,k\neq s}^{q}\|x^k - c^i\|^2.$$

It is clear that the function f is quasidifferentiable and its subdifferential and superdifferential are polytopes at any point. This example demonstrates the importance of development of derivative-free methods for nonsmooth optimization.

4 Approximation of Subgradients

Consider a function f defined on $\mathbb{R}^n$ and assume that it is quasidifferentiable. Assume also that both sets $\underline{\partial} f(x)$ and $\overline{\partial} f(x)$ are polytopes at any $x \in \mathbb{R}^n$ that is at a point $x \in \mathbb{R}^n$ there exist non-empty sets $A = \{a^1, \dots, a^m\} \subset \mathbb{R}^n$, $B = \{b^1, \dots, b^p\} \subset \mathbb{R}^n$ such that $\underline{\partial} f(x) = \operatorname{co} A$, $\overline{\partial} f(x) = \operatorname{co} B$. We denote by $\mathcal{F}$ the class of all semismooth, quasidifferentiable functions whose subdifferential and superdifferential are polytopes at any $x \in \mathbb{R}^n$. This class contains, for example, functions represented as a maximum, minimum or max-min of a finite number of smooth functions.

Let $G = \{e \in \mathbb{R}^n : e = (e_1, \ldots, e_n), |e_j| = 1,\ j = 1, \ldots, n\}$ be a set of all vertices of the unit hypercube in $\mathbb{R}^n$. For $e \in G$ consider the sequence of n vectors $e^j = e^j(\alpha),\ j = 1, \ldots, n$ with $\alpha \in (0, 1]$, where $e^j = (\alpha e_1, \alpha^2 e_2, \ldots, \alpha^j e_j, 0, \ldots, 0)$.

We introduce the following sets:

$$\underline{R}_0(e) \equiv \underline{R}_0 = A, \quad \overline{R}_0(e) \equiv \overline{R}_0 = B,$$

$$\underline{R}_j(e) = \left\{v \in \underline{R}_{j-1}(e) : v_j e_j = \max\{w_j e_j :\ w \in \underline{R}_{j-1}(e)\}\right\},$$

$$\overline{R}_j(e) = \left\{v \in \overline{R}_{j-1}(e) : v_j e_j = \min\{w_j e_j :\ w \in \overline{R}_{j-1}(e)\}\right\}.$$

It is clear that

$$\underline{R}_j(e) \neq \emptyset,\ \forall j \in \{0, \ldots, n\}, \quad \underline{R}_j(e) \subseteq \underline{R}_{j-1}(e), \quad \forall j \in \{1, \ldots, n\},$$

$$\overline{R}_j(e) \neq \emptyset,\ \forall j \in \{0, \ldots, n\}, \quad \overline{R}_j(e) \subseteq \overline{R}_{j-1}(e), \quad \forall j \in \{1, \ldots, n\}.$$

Moreover

$$v_r = u_r \quad \forall v, u \in \underline{R}_j(e),\ w_r = z_r \quad \forall w, z \in \overline{R}_j(e),\ r = 1, \ldots, j, \tag{5}$$

Proposition 4.1 *Assume that $f \in \mathcal{F}$. Then $\underline{R}_n(e)$ and $\overline{R}_n(e)$ are singleton sets.*

The proof immediately follows from (5). □

Consider the following two sets:

$$\underline{R}(x, e^j(\alpha)) = \left\{ v \in A : \langle v, e^j \rangle = \max \ \{\langle u, e^j \rangle : \ u \in A\} \right\},$$

$$\overline{R}(x, e^j(\alpha)) = \left\{ w \in B : \langle w, e^j \rangle = \min \ \{\langle u, e^j \rangle : \ u \in B\} \right\}.$$

We take any $a \in A$. If $a \notin \underline{R}_n(e)$ then there exists $r \in \{1, \dots, n\}$ such that $a \in \underline{R}_t(e), \ t = 0, \dots, r-1$ and $a \notin \underline{R}_r(e)$. It follows from $a \notin \underline{R}_r(e)$ that $v_r e_r > a_r e_r$ for all $v \in \underline{R}_r(e)$. For $a \in A, \ a \notin \underline{R}_n(e)$ we define $d(a) = v_r e_r - a_r e_r > 0$ and then introduce the following number $d_1 = \min\{d(a) : \ a \in A \setminus \underline{R}_n(e)\}$. Since the set A is finite and $d(a) > 0$ for all $a \in A \setminus \underline{R}_n(e)$ it follows that $d_1 > 0$.

We also take any $b \in B$. If $b \notin \overline{R}_n(e)$ then there exists $r \in \{1, \dots, n\}$ such that $b \in \overline{R}_t(e), \ t = 0, \dots, r-1$ and $b \notin \overline{R}_r(e)$. Then we get $v_r e_r < b_r e_r$ for all $v \in \overline{R}_r(e)$. For $b \in B, \ b \notin \overline{R}_n(e)$ we define $d(b) = b_r e_r - v_r e_r > 0$ and introduce the number $d_2 = \min\{d(b) : \ b \in B \setminus \overline{R}_n(e)\}$. $d_2 > 0$ due to the fact that the set B is finite and $d(b) > 0$ for all $b \in B \setminus \overline{R}_n(e)$. Let $\bar{d} = \min\{d_1, \ d_2\}$. Since the subdifferential $\underline{\partial} f(x)$ and the superdifferential $\overline{\partial} f(x)$ are bounded on any bounded subset $X \subset \mathbb{R}^n$, there exists $D > 0$ such that $\|v\| \leq D$ and $\|w\| \leq D$ for all $v \in \underline{\partial} f(y),\ w \in \overline{\partial} f(y)$ and $y \in X$. We take any $r, j \in \{1, \ldots, n\},\ r < j$. Then for all $v, w \in \underline{\partial} f(x),\ x \in X$ and $\alpha \in (0, 1]$ we have

$$\left| \sum_{t=r+1}^{j} (v_t - w_t) \alpha^{t-r} e_t \right| < 2D\alpha n.$$

Let $\alpha_0 = \min\{1, \bar{d}/(4Dn)\}$. Then for any $\alpha \in (0, \alpha_0]$

$$\left| \sum_{t=r+1}^{j} (v_t - w_t) \alpha^{t-r} e_t \right| < \frac{\bar{d}}{2}. \tag{6}$$

In a similar way we can show that for all $v, w \in \overline{\partial} f(x),\ x \in X$ and $\alpha \in (0, \alpha_0]$

$$\left| \sum_{t=r+1}^{j} (v_t - w_t) \alpha^{t-r} e_t \right| < \frac{\bar{d}}{2}. \tag{7}$$

Proposition 4.2 *Assume that $f \in \mathcal{F}$. Then there exists $\alpha_0 > 0$ such that $\underline{R}(x, e^j(\alpha)) \subset \underline{R}_j(e)$ and $\overline{R}(x, e^j(\alpha)) \subset \overline{R}_j(e), \quad j = 1, \ldots, n$ for all $\alpha \in (0, \alpha_0]$.*

Proof: We will prove the first inclusion. The second inclusion can be proved in a similar way. Assume the contrary. Then there exists $y \in \underline{R}(x, e^j(\alpha))$ such that $y \notin \underline{R}_j(e)$. Consequently there exists $r \in \{1, \ldots, n\},\ r \leq j$ such that $y \notin \underline{R}_r(e)$ and $y \in \underline{R}_t(e)$ for any $t = 0, \ldots, r-1$. We take any $v \in \underline{R}_j(e)$. From (5) we have

$v_t e_t = y_t e_t, \quad t = 1, \ldots, r-1, \quad v_r e_r \geq y_r e_r + \bar{d}$. It follows from (6) that

$$\begin{aligned} \langle v, e^j \rangle - \langle y, e^j \rangle &= \sum_{t=1}^{j} (v_t - y_t) \alpha^t e_t \\ &= \alpha^r \left[v_r e_r - y_r e_r + \sum_{t=r+1}^{j} (v_t - y_t) \alpha^{t-r} e_t \right] > \alpha^r \bar{d}/2 > 0. \end{aligned}$$

Since $\langle y, e^j \rangle = \max\{\langle u, e^j \rangle : \; u \in \underline{\partial} f(x)\}$ and $v \in \underline{\partial} f(x)$ we get

$$\langle y, e^j \rangle \geq \langle v, e^j \rangle > \langle y, e^j \rangle + \alpha^r \bar{d}/2$$

which is the contradiction. □

Corollary 4.1 *Assume that the function $f \in \mathcal{F}$. Then there exits $\alpha_0 > 0$ such that*

$$f'(x, e^j(\alpha)) = f'(x, e^{j-1}(\alpha)) + v_j \alpha^j e_j + w_j \alpha^j e_j, \quad \forall v \in \underline{R}_j(e), \; \forall w \in \overline{R}_j(e), \quad j = 1, \ldots, n$$

for all $\alpha \in (0, \alpha_0]$.

Proof: Proposition 4.2 implies that $\underline{R}(x, e^j(\alpha)) \subset \underline{R}_j(e)$ and $\overline{R}(x, e^j(\alpha)) \subset \overline{R}_j(e), \; j = 1, \ldots, n$. Then there exist $v \in \underline{R}_j(e), \; w \in \overline{R}_j(e), \; v^0 \in \underline{R}_{j-1}(e), \; w^0 \in \overline{R}_{j-1}(e)$ such that $f'(x, e^j(\alpha)) - f'(x, e^{j-1}(\alpha)) = \langle v + w, e^j \rangle - \langle v^0 + w^0, e^{j-1} \rangle$ and the proof follows from (5). □

Let $e \in G$ and $\lambda > 0,\ \alpha > 0$ be given numbers. Consider the following points:

$$x^0 = x, \quad x^j = x^0 + \lambda e^j(\alpha), \quad j = 1, \ldots, n.$$

It is clear that $x^j = x^{j-1} + (0, \ldots, 0, \lambda \alpha^j e_j, 0, \ldots, 0), \quad j = 1, \ldots, n$. Let $v = v(e, \alpha, \lambda) \in \mathbb{R}^n$ be a vector with the following coordinates:

$$v_j = (\lambda \alpha^j e_j)^{-1} \left[f(x^j) - f(x^{j-1}) \right], \quad j = 1, \ldots, n. \tag{8}$$

For any fixed $e \in G$ and $\alpha > 0$ we introduce the set:

$$V(e, \alpha) = \left\{ w \in \mathbb{R}^n : \quad \exists (\lambda_k \to +0, \quad k \to +\infty), \quad w = \lim_{k \to +\infty} v(e, \alpha, \lambda_k) \right\}.$$

Proposition 4.3 *Assume that $f \in \mathcal{F}$. Then there exists $\alpha_0 > 0$ such that*

$$V(e, \alpha) \subset \partial f(x), \ \forall \ \alpha \in (0, \alpha_0].$$

Proof: It follows from the definition of vectors $v = v(e, \alpha, \lambda)$ that

$$\begin{aligned} v_j &= (\lambda \alpha^j e_j)^{-1} \left[f(x^j) - f(x^{j-1}) \right] \\ &= (\lambda \alpha^j e_j)^{-1} \left[f(x^j) - f(x) - (f(x^{j-1}) - f(x)) \right] \\ &= (\lambda \alpha^j e_j)^{-1} \left[\lambda f'(x, e^j) - \lambda f'(x, e^{j-1}) + o(\lambda, e^j) - o(\lambda, e^{j-1}) \right] \end{aligned}$$

where $\lambda^{-1}o(\lambda, e^i) \to 0, \quad \lambda \to +0, \;\; i = j-1, j$. We take $w \in \underline{R}_n(e)$ and $y \in \overline{R}_n(e)$. By Proposition 4.1 w and y are unique. Since $\underline{R}_n(e) = \underline{R}(x, e^n)$ and $\overline{R}_n(e) = \overline{R}(x, e^n)$ it follows from Proposition 4.2 in (Ref. 15, p. 146) that $w+y \in \partial f(x)$. The inclusions $w \in \underline{R}_n(e)$ and $y \in \overline{R}_n(e)$ imply that $w \in \underline{R}_j(e)$ and $y \in \overline{R}_j(e)$ for all $j \in \{1, \ldots, n\}$. It follows from Corollary 4.1 that there exists $\alpha_0 > 0$ such that

$$\begin{aligned} v_j(e, \alpha, \lambda) &= (\lambda \alpha^j e_j)^{-1} \left[\lambda \alpha^j e_j (w_j + y_j) + o(\lambda, e^j) - o(\lambda, e^{j-1})\right] \\ &= w_j + y_j + (\lambda \alpha^j e_j)^{-1} \left[o(\lambda, e^j) - o(\lambda, e^{j-1})\right] \end{aligned}$$

for all $\alpha \in (0, \alpha_0]$. Then for any fixed $\alpha \in (0, \alpha_0]$ we have

$$\lim_{\lambda \to +0} |v_j(e, \alpha, \lambda) - (w_j + y_j)| = 0.$$

Consequently, $\lim_{\lambda \to +0} v(e, \alpha, \lambda) = w + y \in \partial f(x)$. □

5 Computation of Subdifferentials

In this section we present an algorithm for the computation of subdifferentials. This algorithm is based on the notion of a discrete gradient. We start with the definition of the discrete gradient, which was introduced in Refs. 18, 19.

Let f be a locally Lipschitz continuous function defined on $\mathbb{R}^n$. Let

$$S_1 = \{g \in \mathbb{R}^n : \|g\| = 1\},$$

$$P = \{z : z(\lambda) \in \mathbb{R}^1,\ z(\lambda) > 0,\ \lambda > 0,\ \lambda^{-1} z(\lambda) \to 0,\ \lambda \to 0\}.$$

Here P is the set of univariate positive infinitesimal functions. We take any $g \in S_1,\ e \in G$, a positive number $\alpha \in (0, 1]$ and compute $i = \text{argmax}\,\{|g_k|,\ k = 1, \ldots, n\}$. Define vectors $e^j(\alpha),\ j = 1, \ldots, n$ as in Section 4 and consider the points:

$$x^0 = x + \lambda g, \quad x^j = x^0 + z(\lambda) e^j(\alpha),\ j = 1, \ldots, n.$$

Definition 5.1 *The discrete gradient of the function f at the point $x \in \mathbb{R}^n$ is the vector $\Gamma^i(x, g, e, z, \lambda, \alpha) = (\Gamma^i_1, \ldots, \Gamma^i_n) \in \mathbb{R}^n, g \in S_1$ with the following coordinates:*

$$\Gamma^i_j = [z(\lambda) \alpha^j e_j)]^{-1} \left[f(x^j) - f(x^{j-1}) \right], \quad j = 1, \ldots, n, \quad j \neq i,$$

$$\Gamma^i_i = (\lambda g_i)^{-1} \left[f(x + \lambda g) - f(x) - \lambda \sum_{j=1, j \neq i}^{n} \Gamma^i_j g_j \right].$$

It follows from Definition 5.1 that

$$f(x + \lambda g) - f(x) = \lambda \langle \Gamma^i(x, g, e, z, \lambda, \alpha), g \rangle \tag{9}$$

for all $g \in S_1,\ e \in G,\ z \in P,\ \lambda > 0,\ \alpha > 0$.

Remark 5.1 The discrete gradient is defined in a direction $g \in S_1$ and to compute it, first we define a sequence of points $x^0, \ldots, x^n$ and compute the values of the function f at these points that is we compute $n+2$ values of this function including the point x. $n-1$ coordinates of the discrete gradient are defined similar to those of the vector $v(e, \alpha, \lambda)$ from Section 4 and i-th coordinate is defined so that to satisfy the equality (9) which can be considered as some version of the mean value theorem.

Proposition 5.1 *Let f be a locally Lipschitz continuous function defined on $\mathbb{R}^n$ and let $L > 0$ be its Lipschitz constant. Then, for any $x \in \mathbb{R}^n,\ g \in S_1,\ e \in G,\ \lambda > 0,\ z \in P,\ \alpha > 0$*

$$\|\Gamma^i\| \leq C(n)L, \quad C(n) = (n^2 + 2n^{3/2} - 2n^{1/2})^{1/2}.$$

Proof: It follows from the definition of the discrete gradients that $|\Gamma^i_j| \leq L$ for all

$j = 1, \dots, n,\ j \neq i$. For $j = i$ we get

$$|\Gamma^i_i| \leq L \left(|g_i|^{-1} \|g\| + \sum_{j=1, j\neq i}^{n} |g_i|^{-1} |g_j| \right).$$

Since $|g_i| = \max\{|g_j|,\ j = 1, \dots, n\}$ we have $|g_i|^{-1}|g_j| \leq 1,\ j = 1, \dots, n$ and $|g_i|^{-1}\|g\| \leq n^{1/2}$. Consequently $|\Gamma^i_i| \leq L(n + n^{1/2} - 1)$. Thus, $\|\Gamma^i\| \leq C(n)L$. □

For a given $\alpha > 0$ we define the following set:

$$B(x, \alpha) = \{v \in \mathbb{R}^n : \ \exists (g \in S_1,\ e \in G,\ z_k \in P,\ z_k \to +0,\ \lambda_k \to +0,\ k \to +\infty),$$

$$v = \lim_{k \to +\infty} \Gamma^i(x, g, e, z_k, \lambda_k, \alpha)\}. \tag{10}$$

Proposition 5.2 *Assume that $f \in \mathcal{F}$. Then, there exists $\alpha_0 > 0$ such that*

$$co\, B(x, \alpha) \subset \partial f(x),\ \forall\ \alpha \in (0, \alpha_0].$$

Proof: Since the function f is semismooth it follows from (2) that for any $\varepsilon > 0$ there exists $\lambda_0 > 0$ such that $v \in R(x, g) + S_\varepsilon$ for all $v \in \partial f(x + \lambda g)$ and $\lambda \in (0, \lambda_0)$. We take any $\lambda \in (0, \lambda_0)$. It follows from Proposition 4.3 and the definition of the discrete gradient that there exist $\alpha_0 > 0$ and $z_0(\lambda) \in P$ such that for any

$\alpha \in (0, \alpha_0],\ z \in P, z(\lambda) < z_0(\lambda)$ can be found $v \in \partial f(x + \lambda g)$ so that $|\Gamma^i_j - v_j| < \varepsilon,\ \ j = 1, \ldots, n,\ j \neq i$. (2) implies that $\|v - w\| < \varepsilon$ for some $w \in R(x, g)$. Then

$$|\Gamma^i_j - w_j| < 2\varepsilon, \ \ j = 1, \ldots, n,\ j \neq i. \tag{11}$$

Since $w \in R(x, g)$ and the function f is semismooth $f'(x, g) = \langle w, g\rangle$ and

$$f(x + \lambda g) - f(x) = \lambda\langle w, g\rangle + o(\lambda, g) \tag{12}$$

where $\lambda^{-1}o(\lambda, g) \to 0$ as $\lambda \to +0$. It follows from (9) and (12) that

$$\Gamma^i_i - w_i = \sum_{j=1, j\neq i}^{n} (w_j - \Gamma^i_j)g_j g_i^{-1} + (\lambda g_i)^{-1}o(\lambda, g).$$

Taking into account (11) we get

$$|\Gamma^i_i - w_i| \leq 2(n - 1)\varepsilon + n^{1/2}\lambda^{-1}|o(\lambda, g)|. \tag{13}$$

Since $\varepsilon > 0$ is arbitrary it follows from (11) and (13) that $\lim_{k\to+\infty} \Gamma^i(x, g, e, z_k, \lambda_k, \alpha) = w \in \partial f(x)$. □

Remark 5.2 The discrete gradient contains three parameters: $\lambda > 0,\ z \in P$ and $\alpha > 0$. $z \in P$ is used to exploit semismoothness of the function f. If $f \in \mathcal{F}$ then for any $\delta > 0$ there exists $\alpha_0 > 0$ such that $\alpha \in (0, \alpha_0]$ for all $y \in S_\delta(x)$. In the sequel we assume that $z \in P$ and $\alpha > 0$ are sufficiently small.

Consider the following set at a point $x \in \mathbb{R}^n$:

$$D_0(x, \lambda) = \operatorname{cl}\operatorname{co}\left\{v \in \mathbb{R}^n : \exists (g \in S_1, e \in G, z \in P) : \quad v = \Gamma^i(x, g, e, \lambda, z, \alpha)\right\}.$$

Proposition 5.1 implies that the set $D_0(x, \lambda)$ is compact and convex for any $x \in \mathbb{R}^n$.

Corollary 5.1 *Assume that $f \in \mathcal{F}$ and in the equality*

$$f(x + \lambda g) - f(x) = \lambda f'(x, g) + o(\lambda, g), \; g \in S_1,$$

$\lambda^{-1} o(\lambda, g) \to 0$ *as* $\lambda \to +0$ *uniformly with respect to* $g \in S_1$. *Then, for any* $\varepsilon > 0$, *there exists* $\lambda_0 > 0$ *such that* $D_0(x, \lambda) \subset \partial f(x) + S_\varepsilon$ *for all* $\lambda \in (0, \lambda_0)$.

Proof: We take $\varepsilon > 0$ and set $\bar{\varepsilon} = \varepsilon / \bar{Q}$, where $\bar{Q} = \left(4n^2 + 4n\sqrt{n} - 6n - 4\sqrt{n} + 3\right)^{1/2}$.

It follows from the proof of Proposition 5.2 and upper semicontinuity of the subdifferential $\partial f(x)$ that for $\bar{\varepsilon} > 0$ there exists $\lambda_1 > 0$ such that

$$\min\left\{\sum_{j=1, j \neq i}^{n} \left(\Gamma^i_j(x, g, e, \lambda, z, \alpha) - v_j\right)^2 : \; v \in \partial f(x)\right\} < \bar{\varepsilon}, \quad j = 1, \ldots, n, \; j \neq i \quad (14)$$

for all $\lambda \in (0, \lambda_1)$. Let

$$A_0 = \text{Argmin}_{\,v \in \partial f(x)} \sum_{j=1, j\neq i}^{n} \left(\Gamma^i_j(x, g, e, \lambda, z, \alpha) - v_j \right)^2 .$$

It follows from (13) and the assumption of the proposition that for $\bar{\varepsilon} > 0$ there exists $\lambda_2 > 0$ such that

$$\min \left\{ \left| \Gamma^i_i(x, g, e, \lambda, z, \alpha) - v_i \right| : \; v \in A_0 \right\} \leq \left(2(n-1) + n^{1/2} \right) \bar{\varepsilon} \tag{15}$$

for all $g \in S_1$ and $\lambda \in (0, \lambda_2)$. Let $\lambda_0 = \min(\lambda_1, \lambda_2)$. Then (14) and (15) imply that

$$\min \left\{ \|\Gamma^i(x, g, e, \lambda, z, \alpha) - v_i\| : \; v \in \partial f(x) \right\} \leq \varepsilon$$

for all $g \in S_1$ and $\lambda \in (0, \lambda_0)$. □

Corollary 5.1 shows that the set $D_0(x, \lambda)$ is an approximation to the subdifferential $\partial f(x)$ for sufficiently small $\lambda > 0$. However it is true at a given point. In order to get convergence results for a minimization algorithm based on discrete gradients we need some relationship between the set $D_0(x, \lambda)$ and $\partial f(x)$ in some neighborhood of a given point x. We will consider functions satisfying the following assumption.

Assumption 5.1 *Let $x \in \mathbb{R}^n$ be a given point. For any $\varepsilon > 0$ there exist $\delta > 0$*

and $\lambda_0 > 0$ *such that* $D_0(y, \lambda) \subset \partial f(x + \bar{S}_\varepsilon) + S_\varepsilon$ *for all* $y \in S_\delta(x)$ *and* $\lambda \in (0, \lambda_0)$.

Here,

$$\partial f(x + \bar{S}_\varepsilon) = \bigcup_{y \in \bar{S}_\varepsilon(x)} \partial f(y), \quad \bar{S}_\varepsilon(x) = \{y \in \mathbb{R}^n : \ \|x - y\| \leq \varepsilon\}.$$

Consider problem (1) where $f : \mathbb{R}^n \to \mathbb{R}^1$ is arbitrary function.

Proposition 5.3 *Let* $x^* \in \mathbb{R}^n$ *be a local minimizer of the function* f. *Then there exists* $\lambda_0 > 0$ *such that* $0 \in D_0(x^*, \lambda)$ *for all* $\lambda \in (0, \lambda_0)$.

The proof follows from the fact that the set $D_0(x^*, \lambda)$ is compact and convex for any $\lambda > 0$. □

Proposition 5.4 *(Ref. 19) Let* $x \in \mathbb{R}^n$, $\lambda > 0$ *and* $0 \notin D_0(x, \lambda)$ *that is* $\|v^0\| = \min\{\|v\| : \ v \in D_0(x, \lambda)\} > 0$. *Then,* $g^0 = -\|v^0\|^{-1} v^0$ *is a descent direction at* x.

Thus, the set $D_0(x, \lambda)$ can be used to compute descent directions. However, the computation of this set is not easy. In the next section we propose an algorithm for the computation of descent directions using a few discrete gradients from $D_0(x, \lambda)$.

6 Computation of Descent Directions

Let $z \in P, \lambda > 0, \alpha \in (0,1]$, the number $c \in (0,1)$ and a tolerance $\delta > 0$ be given.

Algorithm 6.1 Algorithm for the computation of the descent direction.

Step 1. Choose any $g^1 \in S_1, e \in G$, compute $i = \operatorname{argmax}\{|g_j|,\ j = 1,\dots,n\}$ and a discrete gradient $v^1 = \Gamma^i(x, g^1, e, z, \lambda, \alpha)$. Set $\overline{D}_1(x) = \{v^1\}$ and $k = 1$.

Step 2. Compute the vector $\|w^k\|^2 = \min\{\|w\|^2 : w \in \overline{D}_k(x)\}$. If $\|w^k\| \le \delta$, then stop. Otherwise go to Step 3.

Step 3. Compute the search direction by $g^{k+1} = -\|w^k\|^{-1} w^k$.

Step 4. If $f(x + \lambda g^{k+1}) - f(x) \le -c\lambda\|w^k\|$, then stop. Otherwise go to Step 5.

Step 5. Compute $i = \operatorname{argmax}\{|g_j^{k+1}| :\ j = 1,\dots,n\}$ and a discrete gradient

$$v^{k+1} = \Gamma^i(x, g^{k+1}, e, z, \lambda, \alpha),$$

construct the set $\overline{D}_{k+1}(x) = \operatorname{co}\{\overline{D}_k(x) \bigcup \{v^{k+1}\}\}$, set $k = k+1$ and go to Step 2.

Some explanations to Algorithm 6.1 are necessary. In Step 1 we compute the discrete gradient with respect to an initial direction $g^1 \in \mathbb{R}^n$. The distance between the convex hull $\overline{D}_k(x)$ of all computed discrete gradients and the origin is computed in Step 2. This problem is solved using the algorithm from Ref. 20 (for more recent approaches to this problem, see Refs. 21, 22). If this distance is less than the tolerance $\delta > 0$ then we accept the point x as an approximate stationary point (Step 2), otherwise we compute another search direction in Step 3. In Step 4 we check whether this direction is a descent direction. If it is we stop and the descent direction has been computed, otherwise we compute another discrete gradient in this direction in Step 5 and update the set $\overline{D}_k(x)$. At each iteration the approximation of the subdifferential of the function f is improved.

Next we prove that Algorithm 6.1 terminates after a finite number of iterations.

Proposition 6.1 *Let f be a locally Lipschitz function defined on $\mathbb{R}^n$. Then, for $\delta \in (0, \bar{C})$ Algorithm 6.1 terminates after finite number of steps m, where*

$$m \leq 2(\log_2(\delta/\bar{C})/\log_2 r + 1), \quad r = 1 - [(1-c)(2\bar{C})^{-1}\delta]^2,$$

$\bar{C} = C(n)L$ and $C(n)$ is a constant from Proposition 5.1.

Proof: First, we will show that if both conditions for the termination of the algorithm do not satisfy, then a new discrete gradient $v^{k+1} \not\in \overline{D}_k(x)$. Indeed, in this case $\|w^k\| > \delta$ and $f(x+\lambda g^{k+1}) - f(x) > -c\lambda\|w^k\|$. It follows from (9) that

$$\begin{aligned} f(x+\lambda g^{k+1}) - f(x) &= \lambda\langle \Gamma^i(x, g^{k+1}, e, z, \lambda, \alpha), g^{k+1}\rangle \\ &= \lambda\langle v^{k+1}, g^{k+1}\rangle > -c\lambda\|w^k\|. \end{aligned}$$

Then we have

$$\langle v^{k+1}, w^k\rangle < c\|w^k\|^2. \tag{16}$$

On the other hand, since $w^k = \operatorname{argmin}\{\|w\|^2 : w \in \overline{D}_k(x)\}$, the necessary condition for a minimum implies that $\langle w^k, w - w^k\rangle \geq 0$ for any $w \in \overline{D}_k(x)$ or $\langle w^k, w\rangle \geq \|w^k\|^2$. The latter and (16) mean that $v^{k+1} \not\in \overline{D}_k(x)$.

Now we will show that the algorithm is a terminating. We will get an upper estimation for the number of the computed discrete gradients m, when $\|w^m\| \leq \delta$.

It is clear that $\|w^{k+1}\|^2 \leq \|tv^{k+1} + (1-t)w^k\|^2$ for all $t \in [0,1]$. Then

$$\|w^{k+1}\|^2 \leq \|w^k\|^2 + 2t\langle w^k, v^{k+1} - w^k\rangle + t^2\|v^{k+1} - w^k\|^2.$$

It follows from Proposition 5.1 that $\|v^{k+1} - w^k\| \leq 2\bar{C}$. Hence taking into account the inequality (16), we have $\|w^{k+1}\|^2 < \|w^k\|^2 - 2t(1-c)\|w^k\|^2 + 4t^2\bar{C}^2$. For $t = (1-c)(2\bar{C})^{-2}\|w^k\|^2 \in (0,1)$ we get

$$\|w^{k+1}\|^2 < \left\{1 - [(1-c)(2\bar{C})^{-1}\|w^k\|]^2\right\} \|w^k\|^2. \tag{17}$$

Let $\delta \in (0, \bar{C})$. It follows from (17) and the condition $\|w^k\| > \delta, k = 1, \ldots, m-1$ that $\|w^{k+1}\|^2 < \left\{1 - [(1-c)(2\bar{C})^{-1}\delta]^2\right\} \|w^k\|^2$. We denote by $r = 1 - [(1-c)(2\bar{C})^{-1}\delta]^2$. It is clear that $r \in (0,1)$. Then we have

$$\|w^m\|^2 < r\|w^{m-1}\|^2 < \ldots < r^{m-1}\|w^1\|^2 < r^{m-1}\bar{C}^2.$$

If $r^{m-1}\bar{C}^2 \leq \delta^2$, then $\|w^m\| \leq \delta$ and therefore, $m \leq 2(\log_2(\delta/\bar{C})/\log_2 r + 1)$. □

7 Discrete Gradient Method

Let sequences $\delta_k > 0,\ z_k \in P,\ \lambda_k > 0,\ \delta_k \to +0,\ z_k \to +0,\ \lambda_k \to +0,\ k \to +\infty$, sufficiently small number $\alpha > 0$ and numbers $c_1 \in (0,1), c_2 \in (0, c_1]$ be given.

Algorithm 7.1 Discrete Gradient Method

Step 1. Choose any starting point $x^0 \in \mathbb{R}^n$ and set $k = 0$.

Step 2. Set $s = 0$ and $x_s^k = x^k$.

Step 3. Apply Algorithm 6.1 for the computation of the descent direction at $x = x_s^k, \delta = \delta_k, z = z_k, \lambda = \lambda_k, c = c_1$. This algorithm terminates after a finite number of iterations $l > 0$. As a result we get the set $\overline{D}_l(x_s^k)$ and an element v_s^k such that

$$\|v_s^k\|^2 = \min\{\|v\|^2 : \ v \in \overline{D}_l(x_s^k)\}.$$

Furthermore either $\|v_s^k\| \leq \delta_k$ or for the search direction $g_s^k = -\|v_s^k\|^{-1} v_s^k$

$$f(x_s^k + \lambda_k g_s^k) - f(x_s^k) \leq -c_1 \lambda_k \|v_s^k\|. \tag{18}$$

Step 4. If $\|v_s^k\| \leq \delta_k$, then set $x^{k+1} = x_s^k, k = k + 1$ and go to Step 2. Otherwise go to Step 5.

Step 5. Compute $x_{s+1}^k = x_s^k + \sigma_s g_s^k$, where σ_s is defined as follows

$$\sigma_s = \text{argmax} \left\{\sigma \geq 0 : \ f(x_s^k + \sigma g_s^k) - f(x_s^k) \leq -c_2 \sigma \|v_s^k\| \right\}.$$

Step 6. Set $s = s + 1$ and go to Step 3.

For the point $x^0 \in \mathbb{R}^n$ we consider the set $M(x^0) = \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}$.

Theorem 7.1 *Assume that the function $f \in \mathcal{F}$, Assumption 5.1 is satisfied on $\mathbb{R}^n$ and the set $M(x^0)$ is bounded for any $x^0 \in \mathbb{R}^n$. Then, every accumulation point of $\{x^k\}$ belongs to the set $X^0 = \{x \in \mathbb{R}^n : 0 \in \partial f(x)\}$.*

Proof: Since the function f is continuous and the set $M(x^0)$ is bounded

$$f_* = \inf \{f(x) : x \in \mathbb{R}^n\} > -\infty. \tag{19}$$

First we will show that the loop between Steps 3 and 5 stops after a finite number of steps. In other words for any $k > 0$ there exists $s \geq 0$ such that $\|v_s^k\| \leq \delta_k$. Indeed, since $c_2 \in (0, c_1]$ it follows from (18) that $\sigma_s \geq \lambda_k$. Then we can write

$$f(x_{s+1}^k) - f(x_s^k) \;\leq\; -c_2\sigma_s\|v_s^k\| \;\leq\; -c_2\lambda_k\|v_s^k\|.$$

If $\|v_s^k\| > \delta_k$ for all $s \geq 0$ then we have $f(x_{s+1}^k) - f(x_s^k) \leq -c_2\lambda_k\delta_k$ or

$$f(x_{s+1}^k) \leq f(x_0^k) - (s+1)c_2\lambda_k\delta_k. \tag{20}$$

Since $\lambda_k > 0$ and $\delta_k > 0$ are fixed for any $k > 0$ it follows from (20) that $f(x_s^k) \to -\infty$ as $s \to +\infty$. This contradicts (19), that is the loop between Steps 3 and 5 terminates after a finite number of steps and we get a point x^{k+1}

where $\min\{\|v\| : \ v \in \overline{D}_l(x^{k+1})\} \leq \delta_k$. Since $\overline{D}_l(x^{k+1}) \subset D_0(x^{k+1}, \lambda_k)$, we have $\min\{\|v\| : \ v \in D_0(x^{k+1}, \lambda_k)\} \leq \delta_k$. Replacing $k+1$ by k we get

$$\min\{\|v\| : \ v \in D_0(x^k, \lambda_{k-1})\} \leq \delta_{k-1}. \tag{21}$$

Since $\{f(x^k)\}$ is a decreasing sequence $x^k \in M(x^0)$ for all $k > 0$. Then the sequence $\{x^k\}$ is bounded and therefore it has at least one accumulation point. Assume x^* is any accumulation point of the sequence $\{x^k\}$ and $x^{k_i} \to x^*$ as $i \to +\infty$. Then we have from (21)

$$\min\{\|v\| : \ v \in D_0(x^{k_i}, \lambda_{k_i-1})\} \leq \delta_{k_i-1}. \tag{22}$$

According to Assumption 5.1 at the point x^* for any $\varepsilon > 0$ there exist $\beta > 0$ and $\lambda_0 > 0$ such that

$$D_0(y, \lambda) \subset \partial f(x^* + \bar{S}_\varepsilon) + S_\varepsilon \tag{23}$$

for all $y \in S_\beta(x^*)$ and $\lambda \in (0, \lambda_0)$. Since the sequence $\{x^{k_i}\}$ converges to x^* for $\beta > 0$ there exists $i_0 > 0$ such that $x^{k_i} \in S_\beta(x^*)$ for all $i \geq i_0$. On the other hand since $\delta_k, \lambda_k \to 0$ as $k \to +\infty$ there exists $k_0 > 0$ such that $\delta_k < \varepsilon$ and $\lambda_k < \lambda_0$ for all $k > k_0$. Then there exists $i_1 \geq i_0$ such that $k_i \geq k_0 + 1$ for all $i \geq i_1$. Thus it follows from (22) and (23) that $\min\{\|v\| : \ v \in \partial f(x^* + \bar{S}_\varepsilon)\} \leq 2\varepsilon$. Since $\varepsilon > 0$ is arbitrary and the mapping $\partial f(x)$ is upper semicontinuous $0 \in \partial f(x^*)$. □

Remark 7.1 Since Algorithm 6.1 computes descent directions for any values of $\lambda > 0$ we take $\lambda_0 \in (0, 1)$, some $\beta \in (0, 1)$ and update $\lambda_k, \ k \geq 1$ by the formula $\lambda_k = \beta^k \lambda_0, \quad k \geq 1$. Thus, in the proposed method we use approximations to subgradients only at the final stage which guarantees convergence. In most of iterations we do not use approximations of subgradients. Therefore it is a derivative-free method.

Remark 7.2 There are similarities between the discrete gradient and bundle methods. More specifically, the method presented in this paper can be considered as a derivative-free version of the bundle method introduced in Ref. 8. Algorithms for the computation of descent directions in these two methods are similar. However, in the proposed method discrete gradients are used instead of subgradients.

Remark 7.3 It follows from (18) and the condition $c_2 \leq c_1$ that always $\sigma_s \geq \lambda_k$. In order to compute σ_s we define a sequence $\theta_m = m\lambda_k, \quad m \geq 1$ and σ_s is defined as the largest θ_m satisfying the inequality in Step 5.

8 Numerical Experiments

The efficiency of the proposed algorithm was verified by applying it to some unconstrained nonsmooth optimization problems. In numerical experiments we use 20 unconstrained test problems from Ref. 23: Problems 2.1-7 (P1-P7), 2.9-12 (P9-P12), 2.14-16 (P14-P16), 2.18-21 (P18-P21), 2.23-24 (P23, P24).

Objective functions in these problems are maximum functions and they are regular. Objective functions in Problems 2.1, 2.5, 2.23 are convex and they are nonconvex in all other problems. This means that the same algorithm may find different solutions starting from different initial points and/or different algorithms may find different solutions starting from the same initial point. The brief description of these problems is given in Table 1 where the following notation is used:

- n number of variables;
- n_m number of functions under maximum;
- f_{opt} optimum value (as reported in Ref. 23).

For the comparison we use the DNLP model of the CONOPT solver from The General Algebraic Modeling System (GAMS) and the CONDOR solver. DNLP is a nonsmooth optimization solver (Ref. 24). CONDOR is a derivative free optimization solver based on quadratic interpolation and trust region approach (see, Ref. 25 for more details).

Numerical experiments were carried out on PC Pentium 4 with CPU 1.6 MHz. We used 20 random initial points for each problem and initial points are the same for all algorithms. The results are presented in Table 2 where the following notation is used:

- f_{best} and f_{av} the best and average objective function values over 20 runs, respectively;
- nfc the average number of the objective function evaluations (for the discrete gradient method (DGM) and CONDOR);
- iter the average number of iterations (for DNLP);
- DN stands for DNLP and CR for CONDOR;

- F means that an algorithm failed for all initial points.

One can draw the following conclusions from Table 2:

(i) The DGM finds the best known solutions for all problems whereas the CONDOR solver could find the best known solutions only for Problems 1.1-3,7 and the DNLP solver only for Problems 1.1, 1.4.

(ii) Average results over 20 runs by the DGM are better than those by the DNLP and CONDOR solvers, except Problems 2.2, 2.3, 2.6, 2.7 and 2.24 where the CONDOR solver produces better results.

(iii) For convex problems 2.1, 2.5 and 2.23 the DGM always finds the best known solutions. However, this is not the case for the DNLP and CONDOR solvers.

(iv) For most of test problems results by the DNLP solver are worse than those by the CONDOR solver and the DGM. For some problems the values of objective functions and/or their gradients is too large and the DNLP solver fails to solve such problems. Results for Problems 2.6 and 2.23 demonstrate it. However the CONDOR solver and the DGM are quite effective to solve such problems.

(v) As it was mentioned above the most of the test problems are global optimization problems. Results presented demonstrate that the derivative-free methods are effective than Newton-like methods to solve global optimization problems.

(vi) One can see from Table 2 that the number of function calls by the CONDOR solver is significantly less than those by the DGM. However, there is no any significant difference in the CPU time used by different algorithms.

Since the most of test problems are nonconvex we suggest the following scheme to compare the performance of algorithms for each run. Let $\bar{f}$ be the best value obtained by all algorithms starting from the same initial point. Let f^1 be the value of the objective function at the final point obtained by an algorithm. If $f^1 - \bar{f} \le \varepsilon(|\bar{f}| + 1)$, then we say that this algorithm finds the best solution with respect to the tolerance $\varepsilon > 0$. Tables 3 and 4 present pairwise comparison and the comparison of three algorithms, respectively. The numbers in these tables show how many times an algorithm could find the best solution with respect to the tolerance

$\varepsilon = 10^{-4}$. Results presented in Table 3 demonstrate that the CONDOR produces better results than the DNLP in 90 % of runs, the DGM outperforms the DNLP in more than 95 % of runs and finally the DGM produces better results than the CONDOR in almost 80 % of runs.

Results from Table 4 show that the DGM produces better results than other two solvers for all problems except Problems 2.3 and 2.7 where the CONDOR is best.

9 Conclusions

In this paper we have proposed a derivative free algorithm, the discrete gradient method for solving unconstrained nonsmooth optimization problems. This algorithm can be applied to a broad class of nonsmooth optimization problems.

We have tested the new algorithm on some nonsmooth optimization problems. For comparison we used nonsmooth optimization algorithm: the DNLP solver from GAMS which is based on the smoothing of the objective function and the derivative free CONDOR solver which is based on the quadratic approximation of the objective function. Preliminary results of numerical experiments show that the DGM outperforms other two algorithms for the most of test problems considered in this paper. We can conclude that the discrete gradient method is a good alternative to existing derivative-free nonsmooth optimization algorithms.

References

[1] FRANGIONI, A., *Generalized bundle methods,* SIAM Journal on Optimization, Vol. 113, pp. 117-156, 2002.

[2] GAUDIOSO, M. and MONACO, M.F., *A bundle type approach to the unconstrained minimization of convex nonsmooth functions,* Mathematical Programming, Vol. 23, pp. 216-226, 1982.

[3] HIRIART-URRUTY, J.B. and LEMARECHAL, C., *Convex Analysis and Minimization Algorithms*, Springer Verlag, Heidelberg, Vols. 1 and 2, 1993.

[4] KIWIEL, K.C., *Methods of Descent for Nondifferentiable Optimization,* Lecture Notes in Mathematics, Springer-Verlag, Berlin, Vol. 1133, 1985.

[5] LEMARECHAL, C., *An extension of Davidon methods to nondifferentiable problems*, Nondifferentiable Optimization, Balinski, M.L. and Wolfe, P. (eds.), Mathematical Programming Study, Vol. 3, pp. 95-109, North-Holland, Amsterdam, 1975.

[6] MIFFLIN, R., *An algorithm for constrained optimization with semismooth functions*, Mathematics of Operations Research, Vol. 2, pp. 191-207, 1977.

[7] ZOWE, J., *Nondifferentiable optimization: A motivation and a short introduction into the subgradient and the bundle concept*, In: NATO SAI Series, Vol. 15, Computational Mathematical Programming, Schittkowski, K., (ed.), pp. 323-356, Springer-Verlag, New York, 1985.

[8] WOLFE, P.H., *A method of conjugate subgradients of minimizing nondifferentiable convex functions*, Mathematical Programming Study, Vol. 3, pp. 145-173, 1975.

[9] POLAK, E. and ROYSET, J.O., *Algorithms for finite and semi-infinite min-max-min problems using adaptive smoothing techniques*, Journal of Optimiza-

tion Theory and Applications, Vol. 119, pp. 421-457, 2003.

[10] BURKE, J.V., LEWIS, A.S. and OVERTON, M.L., *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM Journal on Optimization, Vol. 15, pp. 751-779, 2005.

[11] AUDET, C. and DENNIS, J.E., Jr., *Analysis of generalized pattern searches*, SIAM Journal on Optimization, Vol. 13, pp. 889-903, 2003.

[12] TORZCON, V., *On the convergence of pattern search algorithms*, SIAM Journal on Optimization, Vol. 7, pp. 1-25, 1997.

[13] CLARKE, F.H., *Optimization and Nonsmooth Analysis,* New York: John Wiley, 1983.

[14] MIFFLIN, R., *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control and Optimization, Vol. 15, pp. 959-972, 1977.

[15] DEMYANOV, V.F. and RUBINOV, A.M., *Constructive Nonsmooth Analysis*, Peter Lang, Frankfurt am Main, 1995.

[16] BAGIROV, A.M., RUBINOV, A.M., SOUKHOROUKOVA, A.V., and YEARWOOD, J., *Supervised and unsupervised data classification via nonsmooth and global optimisation*, TOP: Spanish Operations Research Journal, Vol. 11, pp. 1-93, 2003.

[17] BAGIROV, A.M. and YEARWOOD, J., *A new nonsmooth optimisation algorithm for minimum sum-of-squares clustering problems*, European Journal of Operational Research, Vol. 170, pp. 578-596, 2006.

[18] BAGIROV, A.M., *Minimization methods for one class of nonsmooth functions and calculation of semi-equilibrium prices*, Progress in Optimization: Contribution from Australasia, Edited by A. Eberhard et al., Kluwer Academic Publishers, Dordrecht, pp. 147-175, 1999.

[19] BAGIROV, A.M., *Continuous subdifferential approximations and their applications*, Journal of Mathematical Sciences, Vol. 115, pp. 2567-2609, 2003.

[20] WOLFE, P.H., *Finding the nearest point in a polytope*, Mathematical Programming, Vol. 11, pp. 128-149, 1976.

[21] FRANGIONI, A., *Solving semidefinite quadratic problems within nonsmooth optimization algorithms*, Computers & Operations Research, Vol. 23, pp. 1099-1118, 1996.

[22] KIWIEL, K.C., *A dual method for certain positive semidefinite quadratic programming problems*, SIAM J. Sci. Statist. Comput. Vol. 10, pp. 175-186, 1989.

[23] LUKŠAN, L. and VLČEK, J., *Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization*, Technical Report 78, Institute of Computer Science, Academy of Sciences of the Czech Republic, 2000.

[24] GAMS: *The Solver Manuals*, GAMS Development Corporation, Washington, D.C., 2004.

[25] BERGEN, F. V., *CONDOR: a constrained, non-linear, derivative-free parallel optimizer for continuous, high computing load, noisy objective functions*, PhD thesis, Université Libre de Bruxelles, Belgium, 2004.

List of tables

Table 1: The brief description of test problems

Prob.	n	n_m	f_{opt}	Prob.	n	n_m	f_{opt}
P1	2	3	1.95222	P12	4	21	0.00202
P2	2	3	0	P14	5	21	0.00012
P3	2	2	0	P15	5	30	0.02234
P4	3	6	3.59972	P16	6	51	0.03490
P5	4	4	-44	P18	9	41	0.00618
P6	4	4	-44	P19	7	5	680.63006
P7	3	21	0.00420	P20	10	9	24.30621
P9	4	11	0.00808	P21	20	18	133.72828
P10	4	20	115.70644	P23	11	10	261.08258
P11	4	21	0.00264	P24	20	31	0.00000

Table 2: Results of numerical experiments: best and average values

Pr.	f_{best}			f_{av}			iter	nfc	
	DN	CR	DGM	DN	CR	DGM	DN	CR	DGM
P1	1.9523	1.9522	1.9522	18.6631	1.9563	1.9522	44	88	314
P2	0.0014	0.00000	0.0000	12.6534	0.0021	0.9075	145	97	5018
P3	0.0000	0.0000	0.0000	66.6476	0.0000	0.2200	161	848	8943
P4	3.5998	3.6126	3.5997	9.7501	3.80643	3.5997	44	233	1079
P5	-43.9236	-43.9970	-44	-41.1455	-43.8039	-44	48	352	2862
P6	F	-43.9966	-44	F	-43.8198	-42.8657	F	433	10120
P7	0.0475	0.0042	0.0042	0.0586	0.0054	0.0416	6	215	1316
P9	0.0152	0.02581	0.0081	0.2142	0.0443	0.0179	47	279	5441
P10	116.4907	115.7644	115.7064	256.2033	116.7891	115.7064	51	283	2152
P11	0.0354	0.0103	0.0029	5.3347	0.1679	0.0032	46	343	2677
P12	0.0858	0.0398	0.0125	0.3957	0.1269	0.0628	48	314	2373
P14	2.1643	0.0350	0.0011	2.6490	0.2886	0.1826	34	937	3575
P15	0.7218	0.1405	0.0223	42.9887	0.3756	0.2787	40	592	4656
P16	0.3957	0.0548	0.0349	1.1809	0.4786	0.2872	25	796	7410
P18	0.3085	0.0814	0.0356	0.7024	0.2460	0.1798	43	1289	9694
P19	716.6131	686.0436	680.6301	803.8032	689.7816	680.6301	50	725	2654
P20	35.4217	24.9150	24.3062	60.0370	27.0316	24.3062	121	1892	12926
P21	118.5468	97.8171	93.9073	286.5167	102.7173	94.4516	120	9301	43633
P23	F	3.7053	3.7035	F	3.7107	3.7035	F	3054	3886
P24	0.8953	0.4278	0.3987	15.2643	0.6523	0.8337	58	7418	17928

Table 3: Pairwise comparison of algorithms

Prob.	First pair		Second pair		Third pair	
	DNLP	CONDOR	DNLP	DGM	CONDOR	DGM
P1	5	16	1	20	7	20
P2	1	19	3	17	8	17
P3	3	20	5	16	20	4
P4	10	10	2	20	0	20
P5	2	18	0	20	1	20
P6	0	20	0	20	3	18
P7	0	20	6	20	15	5
P9	4	16	1	19	0	20
P10	0	20	0	20	0	20
P11	5	15	0	20	0	20
P12	6	14	1	19	3	17
P14	0	20	0	20	2	18
P15	2	18	0	20	6	14
P16	1	19	0	20	4	16
P18	1	19	1	20	5	15
P19	0	20	0	20	0	20
P20	0	20	0	20	0	20
P21	0	20	0	20	1	19
P23	0	20	0	20	0	20
P24	0	20	1	19	8	12
Total	40	363	21	390	83	335

Table 4: Comparison of algorithms

Prob.	DNLP	CONDOR	DGM	Prob.	DNLP	CONDOR	DGM
P1	1	7	20	P12	1	3	16
P2	1	7	17	P14	0	2	18
P3	3	20	4	P15	0	6	14
P4	2	0	20	P16	0	4	16
P5	0	1	20	P18	0	5	15
P6	0	3	18	P19	0	0	20
P7	0	15	5	P20	0	0	20
P9	1	0	19	P21	0	1	19
P10	0	0	20	P23	0	0	20
P11	0	0	20	P24	0	8	12
Total	9	82	333				