Journal of Global Optimization (2006) 35: 163–195
DOI 10.1007/s10898-005-3834-4

Piecewise Partially Separable Functions and a Derivative-free Algorithm for Large Scale Nonsmooth Optimization

ADIL M. BAGIROV and JULIEN UGON
Centre for Informatics and Applied Optimization, School of Information Technology and Mathematical Sciences, University of Ballarat, Victoria 3353, Australia
(e-mail: a.bagirov@ballarat.edu.au)

(Accepted 6 October 2005)

Abstract. This paper introduces the notion of piecewise partially separable functions and studies their properties. We also consider some of many applications of these functions. Finally, we consider the problem of minimizing of piecewise partially separable functions and develop an algorithm for its solution. This algorithm exploits the structure of such functions. We present the results of preliminary numerical experiments.

Mathematics Subject Classifications. 65K05, 90C25

Key words: discrete gradient, large scale optimization, nonsmooth optimization, piecewise partially separable functions, subdifferential

1. Introduction

Some important practical problems can be reduced to nonsmooth optimization problems which contain hundreds or thousands of variables. The cluster analysis problem and the problem of calculation of piecewise linear function separating two sets are such problems (see, [6–10, 18, 21]).

Currently available general-purpose nonsmooth optimization methods are not efficient to solve such problems. To our best knowledge the paper [17] presents the first algorithm for dealing with large scale nonsmooth optimization problems. In this paper variable metric bundle algorithm with limited memory has been developed.

Large-scale optimization problems, as a rule, have a special structure. This structure is exploited to design efficient algorithms. Last two decades different algorithms have been developed for solving large scale optimization problems where both objective and constraint functions are twice continuously differentiable (see, for example, [12, 13, 16]). These algorithms strongly rely on the structure of large scale optimization problems, specifically the sparsity of Hessians of the objective and constraint functions.

In this paper we study large scale nonsmooth optimization problems. We introduce a class of piecewise partially separable functions and develop an algorithm for their minimization. This algorithm is based on the so-called discrete gradient method (see [4, 5]). We present preliminary results of numerical experiments which demonstrate that the proposed algorithm is efficient for minimization of piecewise partially separable functions with several thousand variables.

The paper has the following structure. In Section 2 we introduce the new class of nonsmooth functions. Section 3 presents some properties of piecewise partially separable functions. We describe some of many applications of these functions in Section 4. We discuss an algorithm for minimizing of one subclass of piecewise partially separable functions in Section 5. Section 6 presents the results of preliminary numerical experiments and Section 7 concludes the paper.

2. Piecewise Partially Separable Functions: Definition and Examples

Let f be a scalar function defined on an open set $D_0 \subseteq \mathbb{R}^n$ containing a closed set $D \subseteq \mathbb{R}^n$. Here $\mathbb{R}^n$ is an n-dimensional Euclidean space.

DEFINITION 1. The function f is called partially separable if there exists a family of $n \times n$ diagonal matrices $U_i, i = 1, \ldots, M$ such that the function f can be represented as follows:

$$f(x) = \sum_{i=1}^{M} f_i(U_i x).$$

Without loss of generality we assume that the matrices U_i are binary, that is they contain only 0 and 1. It is also assumed that the number m_i of nonzero elements in the diagonal of the matrix U_i is much smaller than n.

In other terms, the function f is called partially separable if it can be represented as the sum of functions of a much smaller number of variables. If $M = n$ and $diag(U_i) = e_i$ where e_i is the i-th orth vector, then the function f is separable.

Remark 1. Any function f can be considered as partially separable if we take $M = 1$ and $U_1 = I$, where I is the identity matrix. However, we consider situations where $M > 1$ and $m_i \ll n, i = 1, \ldots, M$.

EXAMPLE 1. Consider the following function

$$f(x) = \sum_{i=1}^{n} \min\{|x_i|, |x_1|\}.$$

This function is partially separable. Indeed, in this case $M = n, m_i = 2$, $U_i^{11} = 1, U_i^{ii} = 1$, all other elements of U_i are zeros for all $i = 1, \ldots, n$ and $f_i(U_i x) = \min\{|x_i|, |x_1|\}$.

DEFINITION 2. The function f is said to be piecewise partially separable if there exists a finite family of closed sets $D_1, \ldots, D_m$ such that $\cup_{i=1}^m D_i = D$ and the function f is partially separable on each set $D_i, i = 1, \ldots, m$.

EXAMPLE 2. All partially separable functions are piecewise partially separable.

EXAMPLE 3. Consider the following function

$$f(x) = \max_{j=1,\ldots,n} \sum_{i=1}^{n} |x_i - x_j|.$$

The function f is piecewise partially separable. It is clear that the functions

$$\varphi_j(x) = \sum_{i=1}^{n} |x_i - x_j|, \quad j = 1, \ldots, n$$

are partially separable with $M = n, m_i = 2$ and $U_i^{ii} = U_i^{jj} = 1$ for all $i = 1, \ldots, n$. In this case the sets $D_i, i = 1, \ldots, n$ are defined as follows:

$$D_i = \{x \in \mathbb{R}^n : \varphi_i(x) \geqslant \varphi_j(x), \quad j = 1, \ldots, n, \quad j \neq i\}.$$

The piecewise partially separability of the function f follows from the fact that the maximum of partially separable functions is piecewise partially separable, which will be proved later on in Proposition 7.

2.1. CHAINED AND PIECEWISE CHAINED FUNCTIONS

One of the interesting and important classes of partially separable functions is the one of the so-called chained functions.

DEFINITION 3. The function f is said to be k-chained, $k \leqslant n$, if it can be represented as follows:

$$f(x) = \sum_{i=1}^{n-k+1} f_i(x_i, \ldots, x_{i+k-1}).$$

For example, if $k=2$, the function f is:

$$f(x)=\sum_{i=1}^{n-1} f_i(x_i, x_{i+1}).$$

PROPOSITION 1. *Any k-chained function is partially separable.*

Proof. Indeed for k-chained functions $M=n-k+1$, $m_i=k$ and the matrices $U_i, i=1,\dots,M$ are defined as follows:

$$U_i^{jj}=1,\ j=i,\dots,i+k-1$$

and all other elements of U_i are zeros. □

PROPOSITION 2. *Any separable function is* 1*-chained.*

DEFINITION 4. The function f is said to be piecewise k-chained if there exists a finite family of closed sets $D_1,\dots,D_m$ such that $\cup_{i=1}^m D_i = D$ and the function f is k-chained on each set D_i, $i=1,\dots,m$.

PROPOSITION 3. *Any piecewise k-chained function is piecewise partially separable.*
The proof directly follows from Proposition 1. □

The following is an example of piecewise 2-chained function.

EXAMPLE 4. (Chained Crescent I function ([24]).

$$f(x)=\max\{f_1(x), f_2(x)\},$$

where

$$f_1(x)=\sum_{i=1}^{n-1}\left(x_i^2+(x_{i+1}-1)^2+x_{i+1}-1\right),$$
$$f_2(x)=\sum_{i=1}^{n-1}\left(-x_i^2-(x_{i+1}-1)^2+x_{i+1}+1\right).$$

Both f_1 and f_2 are 2-chained functions. We define two sets as follows:

$$D_1=\{x\in\mathbb{R}^n : f_1(x)\geqslant f_2(x)\},$$
$$D_2=\{x\in\mathbb{R}^n : f_2(x)\geqslant f_1(x)\}.$$

It is clear that the sets D_1, D_2 are closed, $f(x) = f_1(x)$ for $x \in D_1$ and $f(x) = f_2(x)$ for $x \in D_2$. Furthermore $D_1 \bigcup D_2 = D$. Thus the function f is piecewise 2-chained.

2.2. PIECEWISE SEPARABLE FUNCTIONS

DEFINITION 5. The function f is said to be piecewise separable if there exists a finite family of closed sets $D_1, \ldots, D_m$ such that $\bigcup_{i=1}^{m} D_i = D$ and the function f is separable on each set $D_i, i = 1, \ldots, m$.

PROPOSITION 4. *Any piecewise separable function is piecewise* 1*-chained.*

Proof. Since any separable function is 1-chained (Proposition 2) the proof is straightforward. □

COROLLARY 1. *Any piecewise separable function is piecewise partially separable.*

PROPOSITION 5. All separable functions are piecewise separable. In this case $m = 1$.

EXAMPLE 5. All piecewise linear functions are piecewise separable. A function $f : D \to \mathbb{R}^1$ is said to be piecewise linear if there exists a finite family of closed sets $Q_1, \ldots, Q_p$ such that $\bigcup_{i=1}^{p} Q_i = D$ and the function f is linear on each set $Q_i, i = 1, \ldots, p$. Since any linear function is separable the function f is piecewise separable and in this case $m = p$.

EXAMPLE 6. One of the simplest piecewise separable functions is the following maximum function:

$$f(x) = \max_{i=1,\ldots,n} x_i^2.$$

Here $m = n$ and

$$D_i = \{x \in \mathbb{R}^n : x_i^2 \geqslant x_j^2, \quad j = 1, \ldots, n, \quad j \neq i\}.$$

$f(x) = x_i^2$ for any $x \in D_i$. It is clear that $\bigcup_{i=1}^{m} D_i = \mathbb{R}^n$. It should be noted that the function f is neither separable nor piecewise linear.

3. Properties of Piecewise Partially Separable Functions

In this section we study some properties of piecewise partially separable functions.

PROPOSITION 6. *Let f_1 and f_2 be partially separable functions on the closed D. Then the function $f(x)=f_1(x)+f_2(x)$ is also partially separable on D.*

Proof. Since the functions f_1 and f_2 are partially separable there exist families of matrices $U_i^1,\ i=1,\dots,M_1$ and $U_j^2, j=1,\dots,M_2$ such that

$$f_1(x)=\sum_{i=1}^{M_1} f_{1i}(U_i^1 x),$$

$$f_2(x)=\sum_{j=1}^{M_2} f_{2j}(U_j^2 x).$$

Consider the following sets:

$$I=\left\{i\in\{1,\dots,M_1\}: U_i^1\neq U_j^2,\ \ \forall j\in\{1,\dots,M_2\}\right\},$$
$$J=\left\{j\in\{1,\dots,M_2\}: U_j^2\neq U_i^1,\ \ \forall i\in\{1,\dots,M_1\}\right\},$$
$$H=\left\{(i,j),\ \ i\in\{1,\dots,M_1\},\ \ j\in\{1,\dots,M_2\}: U_i^1=U_j^2\right\}.$$

It is clear that for any $i\in I$ there is no $j\in\{1,\dots,M_2\}$ such that $(i,j)\in H$ and similarly for any $j\in J$ there is no $i\in\{1,\dots,M_1\}$ such that $(i,j)\in H$. Then the function f can be represented as follows

$$f(x)=\sum_{(i,j)\in H}\left(f_{1i}(U_i^1 x)+f_{2j}(U_j^2 x)\right)+\sum_{i\in I} f_{1i}(U_i^1 x)+\sum_{j\in J} f_{2j}(U_j^2 x).$$

This function is partially separable, that is

$$f(x)=\sum_{k=1}^{M}\bar{f}_k(V_k x),$$

where $M=M_1+M_2-card(H)$, the matrices $V_k, k=1,\dots,M$ can be defined as follows:

$$V_k=\begin{cases} U_i^1=U_j^2 & k=1,\dots,card(H) & (i,j)\in H,\\ U_i^1 & k=card(H)+1,\dots,M_1 & i\in I,\\ U_j^2 & k=M_1+1,\dots,M_1+M_2-card(H) & j\in J,\end{cases}$$

and

$$\bar{f}_k(V_k x) = \begin{cases} (f_{1i}(U_i^1 x) + f_{2j}(U_j^2 x)) & k = 1, \ldots, card(H) \quad (i,j) \in H \\ f_{1i}(U_i^1 x) & k = card(H)+1, \ldots, M_1 \quad i \in I \\ f_{2j}(U_j^2 x) & k = M_1+1, \ldots, M_1+M_2-card(H) \quad j \in J. \end{cases}$$

Here $card(H)$ stands for the cardinality of the set H. □

We say that two partially separable functions f_1 and f_2 have the same structure if $I = J = \emptyset$. These functions are more interesting from a practical point of view. In this case the function f has the same structure as f_1 and f_2 and

$$f(x) = \sum_{(i,j)\in H} (f_{1i}(U_i^1 x) + f_{2j}(U_j^2 x)).$$

For example, if f_1 and f_2 are k-chained then the function f is also k-chained.

PROPOSITION 7. *If f and g are piecewise partially separable (piecewise k-chained, piecewise separable) continuous functions on the closed set D, then*

(1) *$h(x) = \alpha f(x), \alpha \in \mathbb{R}^1$ is piecewise partially separable (piecewise k-chained, piecewise separable);*
(2) *$h(x) = f(x) + g(x)$ is piecewise partially separable (piecewise k-chained, piecewise separable);*
(3) *$h(x) = max(f(x), g(x))$, $h(x) = min(f(x), g(x))$ and $h(x) = |f(x)|$ are piecewise partially separable (piecewise k-chained, piecewise separable).*

Proof.(1) The proof is straightforward.

(2) Since the functions f and g are piecewise partially separable there exist families of closed sets

$$D_i^f, \quad i = 1, \ldots, m_1, \quad \bigcup_{i=1}^{m_1} D_i^f = D$$

and

$$D_j^g, \quad j = 1, \ldots, m_2, \quad \bigcup_{j=1}^{m_2} D_j^g = D$$

such that the function f is partially separable on the sets D_i^f and the function g is partially separable on the sets D_j^g. We define a family of sets $Q_{ij}, i=1,\dots,m_1, j=1,\dots,m_2$ where

$$Q_{ij} = D_i^f \cap D_j^g.$$

It is clear that

$$\bigcup_{i,j} Q_{ij} = D$$

and the sets Q_{ij} are closed. Since the sum of partially separable functions is partially separable we get that $f+g$ is partially separable on each set Q_{ij}.

The proof for piecewise k-chained and piecewise separable functions is similar.

(3) Consider the following two sets:

$$P_1 = \{x \in D : f(x) \geqslant g(x)\}, \qquad P_2 = \{x \in D : g(x) \geqslant f(x)\}.$$

It is clear that $P_1 \cup P_2 = D$. Since the functions f and g are continuous the sets P_1 and P_2 are closed. We define the following families of sets:

$$Q_i^1 = P_1 \cap D_i^f, \quad i=1,\dots,m_1, \qquad Q_j^2 = P_2 \cap D_j^g, \quad j=1,\dots,m_2.$$

These sets are closed. It can be easily shown that

$$\left(\bigcup_i^{m_1} Q_i^1\right) \cup \left(\bigcup_j^{m_2} Q_j^2\right) = D.$$

$h(x) = f(x), x \in Q_i^1, i=1,\dots,m_1$ and f is partially separable on each set Q_i^1. Similarly $h(x) = g(x), x \in Q_j^2, j=1,\dots,m_2$ and g is partially separable on each set Q_j^2. Then we get that the function h is piecewise partially separable.

Since $h(x) = \min(f(x), g(x)) = -\max(-f(x), -g(x))$ then we get that h is piecewise partially separable. $h(x) = |f(x)| = \max(f(x), -f(x))$ and both f and $-f$ are piecewise partially separable it follows that the function h is also piecewise partially separable.

Again the proof for piecewise k-chained and piecewise separable functions is similar. □

The problem of computation of Hessians of twice continuously differentiable partially separable functions was discussed by many authors (see, for example, [1, 12]).

In order to describe some differential properties of piecewise partially separable functions we recall some definitions from nonsmooth analysis.

We consider a locally Lipschitz continuous function f defined on $\mathbb{R}^n$. This function is differentiable almost everywhere and one can define for it a Clarke subdifferential (see [11]), by

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

here $D(f)$ denotes the set where f is differentiable and co is a convex hull of a set.

The function f is differentiable at the point $x \in \mathbb{R}^n$ with respect to the direction $g \in \mathbb{R}^n$ if the limit

$$f'(x, g) = \lim_{\alpha \to +0} \frac{f(x + \alpha g) - f(x)}{\alpha}$$

exists. The number $f'(x, g)$ is said to be the derivative of the function f with respect to the direction $g \in \mathbb{R}^n$ at the point x.

The Clarke upper derivative $f^0(x, g)$ of the function f at the point x with respect to the direction $g \in \mathbb{R}^n$ is defined as follows:

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

The following is true (see [11])

$$f^0(x, g) = \max\{\langle v, g\rangle : v \in \partial f(x)\}.$$

Here $\langle \cdot, \cdot \rangle$ stands for an inner product in $\mathbb{R}^n$. It should be noted that the Clarke upper derivative always exists for locally Lipschitz continuous functions. The function f is said to be regular at the point $x \in \mathbb{R}^n$ if

$$f'(x, g) = f^0(x, g)$$

for all $g \in \mathbb{R}^n$. For Clarke regular functions there exists a calculus (see [11, 14]). However in general for nonregular functions such a calculus does not exist.

Now let us assume that the function f is partially separable and the functions $f_i, i = 1, \ldots, M$ are directionally differentiable. Then the function f is also directionally differentiable and

$$f'(x, g) = \sum_{i=1}^{M} f_i'(U_i x, U_i g). \tag{1}$$

It follows from this formula that if f separable then

$$f'(x, g) = \sum_{i=1}^{n} f_i'(x_i, g_i), \tag{2}$$

where

$$f_i'(x_i, g_i) = \begin{cases} f_{i+}'(x_i) & \text{if } g_i > 0, \\ 0 & \text{if } g_i = 0, \\ -f_{i-}'(x_i) & \text{if } g_i < 0. \end{cases}$$

and $f_{i+}'(x_i)$, $f_{i-}'(x_i)$ are the right and left side derivatives of the function f_i at the point x_i.

Below we study the Lipschitz continuity and directional differentiability of piecewise partially separable functions.

Let f be a piecewise partially separable function on the closed convex set $D \subset \mathbb{R}^n$, that is there exists a family of closed sets D_j, $j = 1, \ldots, m$ such that $\bigcup_{j=1}^{m} D_j = D$, $f(x) = f_j(x), x \in D_j$ and the functions f_j are partially separable on D_j.

PROPOSITION 8. *Let f be continuous and each function f_j be locally Lipschitz continuous on D_j, $j = 1, \ldots, m$. Then the function f is locally Lipschitz continuous on D.*

Proof. We take any bounded subset $\overline{D} \subset D$. Then there exists a subset of indices $\{j_1, \ldots, j_p\} \subset \{1, \ldots, m\}$ such that

$$\operatorname{co} \overline{D} \cap D_{j_k} \neq \emptyset, \quad k = 1, \ldots, p.$$

Let $L_{j_k} > 0$ be a Lipschitz constant of the function f_{j_k} on the set $\operatorname{co} \overline{D} \cap D_{j_k}, k = 1, \ldots, p$. Let

$$L_0 = \max_{k=1,\ldots,p} L_{j_k}.$$

Now we take any two points $x, y \in \overline{D}$. Then there exist indices $j_{k_1}, j_{k_2} \in \{j_1, \ldots, j_p\}$ such that $x \in D_{j_{k_1}}$ and $y \in D_{j_{k_2}}$. If $k_1 = k_2 = k$ then it is clear that

$$|f(x) - f(y)| = |f_k(x) - f_k(y)| \leqslant L_k \|x - y\| \leqslant L_0 \|x - y\|.$$

Otherwise we consider the segment $[x, y] = \alpha x + (1 - \alpha) y, \alpha \in [0, 1]$ joining these two points and define the following set:

$$Z_{[x,y]} = \left\{ z \in [x, y] : \exists l_1, l_2 \in \{1, \ldots, p\} : z \in D_{j_{l_1}} \cap D_{j_{l_2}} \right\}.$$

It is clear that in this case the set $Z_{[x,y]}$ is not empty. Then there exists a sequence of points $\{z_1, \ldots, z_N\} \subset Z_{[x,y]}$, $N \leqslant p$ such that

- $\{x, z_1\} \subset D_{j_{k_1}}, \quad l_0 = k_1;$
- $\{z_N, y\} \subset D_{j_{k_2}}, \quad l_N = k_2;$
- $\forall i \in \{1, \ldots, N-1\}, \exists l_i \in \{1, \ldots, p\} : \{z_i, z_{i+1}\} \subset D_{j_{l_i}}.$

Then taking into account the continuity of the function f we have:

$$
\begin{aligned}
|f(y) - f(x)| &= \left| f(y) + \sum_{i=1}^{N} (f(z_i) - f(z_i)) - f(x) \right| \\
&= \left| f_{j_{k_2}}(y) + \sum_{i=1}^{N} (f_{j_{l_{i-1}}}(z_i) - f_{j_{l_i}}(z_i)) - f_{j_{k_1}}(x) \right| \\
&\leqslant |f_{j_{k_2}}(y) - f_{j_{k_2}}(z_N)| + \sum_{i=1}^{N-1} |f_{j_{l_i}}(z_{i+1}) - f_{j_{l_i}}(z_i)| \\
&\quad + |f_{j_{k_1}}(z_1) - f_{j_{k_1}}(x)| \\
&\leqslant L_{j_1} \|y - z_N\| + \sum_{i=1}^{N-1} L_{j_i} \|z_i - z_{i+1}\| + L_{j_{k_1}} \|z_1 - x\| \\
&\leqslant L_0 (\|y - z_N\| + \sum_{i=1}^{N-1} \|z_i - z_{i+1}\| + \|z_1 - x\|).
\end{aligned}
$$

Then, as all z_i are aligned on the segment $[x, y]$, we get

$$|f(y) - f(x)| \leqslant L_0 \|y - x\|.$$

Since points x and y are arbitrary it follows that the function f is locally Lipschitz continuous. □

COROLLARY 2. *Assume that all conditions of Proposition 8 are satisfied. Then the function f is Clarke subdifferentiable.*

PROPOSITION 9. *Assume that for any two points $x, y \in D$ the set $Z_{[x,y]}$ is finite and all functions f_j, $j = 1, \ldots, m$ are directionally differentiable. Then the function f is also directionally differentiable.*

Proof. We take any point $x \in D$ and any direction $g \neq 0$ such that $x + \alpha g \in D, \alpha \in [0, \bar{\alpha}]$ for some $\bar{\alpha} > 0$. By the definition

$$f'(x, g) = \lim_{\alpha \to +0} \frac{f(x + \alpha g) - f(x)}{\alpha}.$$

Assume that $x \in \bigcap_{k \in K} D_k$, where $K \subset \{1, \ldots, m\}$. Let $y = x + \bar{\alpha} g \in D$. Since the set $Z_{[x,y]}$ is finite there exists a finite sequence of numbers $\alpha_1, \ldots, \alpha_l$ such that $\alpha_i \in (0, \bar{\alpha})$ and $x + \alpha_j g \in D_{k_j} \cap D_{k_{j+1}}$, $j = 1, \ldots, l$ and

- $[x, x + \alpha_1 g] \subset D_{k_1}, \quad k_1 \in K;$
- $[x + \alpha_l g, y] \subset D_{k_{l+1}};$
- $\forall i \in \{1, \ldots, l-1\} : [x + \alpha_i g, x + \alpha_{i+1} g] \subset D_{k_{i+1}}.$

This implies that the segment $[x, x + \alpha_1 g] \subset D_{k_1}$. Thus

$$f'(x, g) = f'_{k_1}(x, g).$$

It follows that if the function f is piecewise partially separable then its directional derivative can be calculated using (1) and if this function is piecewise separable then its directional derivative is calculated using (2). □

In general piecewise partially separable functions are not regular. The following example demonstrates it.

EXAMPLE 7. Consider the function

$$f(x_1, x_2) = \max\{|x_1| - |x_2|, -|x_1| + |x_2|\}, \quad (x_1, x_2) \in \mathbb{R}^2.$$

This function is piecewise separable. However it is not regular. Indeed, for the direction $g = (1, 1)$ at the point $x = (0, 1)$ we have

$$f'(x, g) = 0 \quad \text{and} \quad f^0(x, g) = 2,$$

that is $f'(x, g) < f^0(x, g)$.

This example shows that in general for the subdifferential of piecewise partially separable functions a full calculus does not exist. Therefore in many cases the computation of their subgradients is quite difficult task.

4. Motivation: Examples from Applications

In this section we present two very important applications of piecewise partially separable functions.

4.1. CLUSTERING FUNCTION

Cluster analysis has found many applications, including information retrieval, medicine etc. Clustering is also known as the unsupervised classification of patterns. The clustering problem has been studied by many authors and different algorithms have been developed for its solution (see [18, 21]).

In cluster analysis we assume that we have been given a finite set of points A in the n-dimensional space $\mathbb{R}^n$, that is

$$A = \{a^1, \ldots, a^M\}, \quad \text{where } a^i \in \mathbb{R}^n, \quad i = 1, \ldots, M.$$

The cluster analysis deals with the problems of organization of a collection of patterns a^i into clusters based on similarity. As a measure of similarity different distances can be used. Here for the sake of simplicity we consider Euclidean distance. We consider partition clustering, that is the distribution of the points of the set A into a given number q of disjoint subsets $A^i, i = 1, \ldots, q$ with respect to predefined criteria such that:

(1) $A^i \neq \emptyset, \quad i = 1, \ldots, q;$
(2) $A^i \cap A^j = \emptyset, \quad i, j = 1, \ldots, q, \quad i \neq j;$
(3) $A = \cup_{i=1}^{q} A^i.$

The sets $A^i, i = 1, \ldots, q$ are called clusters. We can assume that each cluster $A^i, i = 1, \ldots, q$ can be identified by its center (or centroid). Then the clustering problem can be reduced to the following nonsmooth optimization problem (see [6]):

$$\text{minimize } f(x^1, \ldots, x^q) \text{ subject to } \quad (x^1, \ldots, x^q) \in \mathbb{R}^{n \times q}, \tag{3}$$

where

$$f(x^1, \ldots, x^q) = \frac{1}{M} \sum_{i=1}^{M} \min_{s=1,\ldots,q} \|x^s - a^i\|^2. \tag{4}$$

x^i is the center of the cluster $A^i, i = 1, \ldots, q$. If $q > 1$, the function (4) is nonconvex and nonsmooth. The problem (3) is also known as the sum-of-squares clustering problem.

It is clear that the function

$$\psi(y) = \|y - a\|^2, \quad y \in \mathbb{R}^n$$

is separable and therefore the function

$$\varphi_i(x) = \min_{s=1,\ldots,q} \|x^s - a^i\|^2$$

is piecewise separable. Then it follows from Proposition 7 that the function (4) is piecewise separable.

4.2. MAX–MIN SEPARABILITY

The problems of supervised data classification arise in many areas including management science, medicine, chemistry. The aim of supervised data classification is to establish rules for the classification of some observations assuming that the classes of data are known. To find these rules, known training subsets of the given classes are used. This problem can be reduced to a number of set separation problems. For each class, the training points belonging to this class have to be separated from the other training points using a certain, not necessarily linear, function. In the paper [8] an algorithm for calculation of piecewise linear functions separating two sets is developed. This problem is formulated as a nonsmooth optimization problem with max–min-type objective function. We will briefly describe this problem.

Let A and B be given disjoint sets containing m and p n-dimensional vectors, respectively:

$$A=\{a^1,\dots,a^m\},\quad a^i\in\mathbb{R}^n,\ \ i=1,\dots,m,$$
$$B=\{b^1,\dots,b^p\},\ \ b^j\in\mathbb{R}^n,\quad j=1,\dots,\ \ A\cap B=\emptyset.$$

Let $H=\{h_1,\dots,h_l\}$, where $h_j=\{x^j,y_j\}$, $j=1,\dots,l$ with $x^j\in\mathbb{R}^n$, $y_j\in\mathbb{R}^1$, be a finite set of hyperplanes. Let $J=\{1\dots,l\}$. Consider any partition of this set $J^r=\{J_1,\dots,J_r\}$ such that

$$J_k\neq\emptyset,\ \ k=1,\dots,r,\quad J_k\cap J_j=\emptyset,\quad \cup_{k=1}^r J_k=J.$$

Let $I=\{1,\dots,r\}$. A particular partition $J^r=\{J_1,\dots,J_r\}$ of the set J defines the following max–min-type function:

$$\varphi(z)=\max_{i\in I}\min_{j\in J_i}\left\{\langle x^j,z\rangle-y_j\right\},\quad z\in\mathbb{R}^n. \tag{5}$$

DEFINITION 6. (*see* [8]). The sets A and B are max–min separable if there exist a finite number of hyperplanes $\{x^j,y_j\}$ with $x^j\in\mathbb{R}^n$, $y_j\in\mathbb{R}^1$, $j\in J=\{1,\dots,l\}$ and a partition $J^r=\{J_1,\dots,J_r\}$ of the set J such that

(1) for all $i\in I$ and $a\in A$

$$\min_{j\in J_i}\left\{\langle x^j,a\rangle-y_j\right\}<0;$$

(2) for any $b\in B$ there exists at least one $i\in I$ such that

$$\min_{j\in J_i}\left\{\langle x^j,b\rangle-y_j\right\}>0.$$

Remark 2. It follows from Definition 6 that if the sets A and B are max–min separable then $\varphi(a) < 0$ for any $a \in A$ and $\varphi(b) > 0$ for any $b \in B$, where the function φ is defined by (5). Thus the sets A and B can be separated by a function represented as a max–min of linear functions. Therefore this kind of separability is called a max–min separability.

The problem of the max–min separability is reduced to the following mathematical programming problem (see [8]):

$$\text{minimize } f(x, y) \quad \text{subject to } (x, y) \in \mathbb{R}^{ln} \times \mathbb{R}^{l}, \tag{6}$$

where the objective function f has the following form:

$$f(x, y) = f_1(x, y) + f_2(x, y)$$

and

$$f_1(x, y) = \frac{1}{m} \sum_{k=1}^{m} \max \left[0, \max_{i \in I} \min_{j \in J_i} \left\{ \langle x^j, a^k \rangle - y_j + 1 \right\} \right], \tag{7}$$

$$f_2(x, y) = \frac{1}{p} \sum_{t=1}^{p} \max \left[0, \min_{i \in I} \max_{j \in J_i} \left\{ -\langle x^j, b^t \rangle + y_j + 1 \right\} \right]. \tag{8}$$

One can see that both functions f_1 and f_2 are piecewise linear, therefore the resulting function f is piecewise linear and consequently piecewise separable.

5. Minimization of Piecewise Partially Separable Functions

In this section we will develop an algorithm for minimizing one class of piecewise partially separable functions.

We will consider the following unconstrained minimization problem

$$\text{minimize } f(x) \quad \text{subject to } x \in \mathbb{R}^{n}, \tag{9}$$

where the objective function f is as follows

$$f(x) = \sum_{i=1}^{M} \max_{j \in J_i} \min_{k \in K_j} f_{ijk}(x) \tag{10}$$

and functions $f_{ijk}, i=1,\ldots,M, j\in J_i, k\in K_j$ are partially separable, that is there exists a family of $n\times n$ matrices $U_{ijkt}, t=1,\ldots,M_{ijk}$ such that

$$f_{ijk}(x)=\sum_{t=1}^{M_{ijk}} f_{ijk}^t(U_{ijkt}x).$$

The function f is piecewise partially separable. If all functions f_{ijk} are l-chained (separable) then the function f is piecewise l-chained (piecewise separable).

Particular cases of this function are the following:

1. The case when the sets $J_i, i=1,\ldots,M$ are singletons

$$f(x)=\sum_{i=1}^{M}\min_{k\in K_i} f_{ik}(x). \tag{11}$$

   The clustering function serves as an example for this type of functions when $K_i=\{1,\ldots,K\}, \forall i\in\{1,\ldots,M\}$ and the functions f_{ik} are separable.
2. The case when $M=1$

$$f(x)=\max_{j\in J}\min_{k\in K_j} f_{jk}(x). \tag{12}$$

As we can see from Example 7 even for very simple cases this type of functions may not be regular and therefore sometimes the computation of their subgradients is quite difficult. Therefore, methods based on function evaluations only seem better alternatives to solve problem (9). However the existing direct search methods, including Powell method (see [27]) and Nelder–Mead simplex method [26], become inefficient when the number of variables increases.

We will develop a new modified version of the discrete gradient method for solving problem (9). This is a derivative-free method. The description of this method can be found in [3, 5] (see, also, [4]). The discrete gradient method can be considered as a version of the bundle method ([19, 20, 24]), where subgradients of the objective function are replaced by its discrete gradients. This method consists of three main steps: the calculation of discrete gradients, the calculation of descent directions and line search. Numerical experiments have shown that for large scale problems the first step takes most of the CPU time used by the method. We will introduce a new scheme for the calculation of discrete gradients of piecewise partially separable functions represented as a sum of max–min functions. To calculate the discrete gradients we use only values of the objective function.

Since the calculation of the objective function in the problem (9) can be expensive, such a scheme will allow one to significantly reduce the number of objective function evaluations.

In order to describe a new scheme for the calculation of the discrete gradient we recall here its definition.

5.1. DISCRETE GRADIENT

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

$$
\begin{aligned}
&S_1 = \{g \in \mathbb{R}^n : \|g\| = 1\},\\
&G = \{e \in \mathbb{R}^n : e = (e_1, \dots, e_n), |e_j| = 1, j = 1, \dots, n\},\\
&P = \{z(\lambda) : z(\lambda) \in \mathbb{R}^1, z(\lambda) > 0, \lambda > 0, \lambda^{-1} z(\lambda) \to 0, \lambda \to 0\},\\
&I(g, \alpha) = \{i \in \{1, \dots, n\} : |g_i| \geqslant \alpha\},
\end{aligned}
$$

where $\alpha \in (0, n^{-1/2}]$ is a fixed number.

Here S_1 is the unit sphere, G is the set of vertices of the unit hypercube in $\mathbb{R}^n$ and P is the set of univariate positive infinitesimal functions.

We define operators $H_i^j : \mathbb{R}^n \to \mathbb{R}^n$ for $i = 1, \dots, n$, $j = 0, \dots, n$ by the formula

$$
H_i^j g = \begin{cases} (g_1, \dots, g_j, 0, \dots, 0) & \text{if } j < i, \\ (g_1, \dots, g_{i-1}, 0, g_{i+1}, \dots, g_j, 0, \dots, 0) & \text{if } j \geqslant i. \end{cases} \tag{13}
$$

We can see that

$$
H_i^j g - H_i^{j-1} g = \begin{cases} (0, \dots, 0, g_j, 0, \dots, 0) & \text{if } j = 1, \dots, n, j \neq i, \\ 0 & \text{if } j = i. \end{cases} \tag{14}
$$

Let $e(\beta) = (\beta e_1, \beta^2 e_2, \dots, \beta^n e_n)$, where $\beta \in (0, 1]$. For $x \in \mathbb{R}^n$ we consider vectors

$$
x_i^j \equiv x_i^j(g, e, z, \lambda, \beta) = x + \lambda g - z(\lambda) H_i^j e(\beta), \tag{15}
$$

where $g \in S_1$, $e \in G$, $i \in I(g, \alpha)$, $z \in P$, $\lambda > 0$, $j = 0, \dots, n$, $j \neq i$.

It follows from (14) that

$$
x_i^{j-1} - x_i^j = \begin{cases} (0, \dots, 0, z(\lambda) e_j(\beta), 0, \dots, 0) & \text{if } j = 1, \dots, n, j \neq i, \\ 0 & \text{if } j = i. \end{cases} \tag{16}
$$

It is clear that $H_i^0 g = 0$ and $x_i^0(g, e, z, \lambda, \beta) = x + \lambda g$ for all $i \in I(g, \alpha)$.

DEFINITION 7. (*see* [2]). The discrete gradient of the function f at the point $x \in \mathbb{R}^n$ is the vector $\Gamma^i(x, g, e, z, \lambda, \beta) = (\Gamma_1^i, \ldots, \Gamma_n^i) \in \mathbb{R}^n$, $g \in S_1$, $i \in I(g, \alpha)$, with the following coordinates:

$$\Gamma_j^i = [z(\lambda) e_j(\beta)]^{-1} \left[f(x_i^{j-1}(g, e, z, \lambda, \beta)) - f(x_i^j(g, e, z, \lambda, \beta)) \right],$$
$$j = 1, \ldots, n, \; j \neq i,$$
$$\Gamma_i^i = (\lambda g_i)^{-1} \left[f(x_i^n(g, e, z, \lambda, \beta)) - f(x) - \sum_{j=1, j \neq i}^{n} \Gamma_j^i (\lambda g_j - z(\lambda) e_j(\beta)) \right].$$

A more detailed description of the discrete gradient and examples can be found in [3].

Remark 3. It follows from Definition 7 that for the calculation of the discrete gradient $\Gamma^i(x, g, e, z, \lambda, \beta), i \in I(g, \alpha)$ we define a sequence of points

$$x_i^0, \ldots, x_i^{i-1}, x_i^{i+1}, \ldots, x_i^n.$$

For the calculation of the discrete gradient it is sufficient to evaluate the function f at each point of this sequence.

Remark 4. The discrete gradient is defined with respect to a given direction $g \in S_1$. We can see that for the calculation of one discrete gradient we have to calculate $(n+1)$ values of the function f: at the point x and at the points $x_i^j(g, e, z, \lambda, \beta)$, $j = 0, \ldots, n$, $j \neq i$. For the calculation of the next discrete gradient at the same point with respect to any other direction $g^1 \in S_1$ we have to calculate this function n times, because we have already calculated f at the point x.

Remark 5. One can see from (16) that two successive points of the sequence

$$x_i^0, \ldots, x_i^{i-1}, x_i^{i+1}, \ldots, x_i^n$$

differ by one coordinate only. More precisely, the point x^k can be obtained from the point x^{k-1} by changing only the k-th coordinate.

5.2. CALCULATION OF THE DISCRETE GRADIENTS OF THE FUNCTION (10)

We take any point $x \in \mathbb{R}^n$ and any direction $g \in S_1$. Remark 3 implies that for the calculation of the discrete gradient of f at x with respect to the direction g first we have to define the sequence

$$x_i^0, \ldots, x_i^{i-1}, x_i^{i+1}, \ldots, x_i^n.$$

It follows from Remark 5 that each new point x^p differs from x^{p-1} by one coordinate only. In order to calculate the discrete gradient we have to evaluate the function f at all of these points.

The functions f_{ijk} are partially separable and they can be represented as

$$f_{ijk}(x) = \sum_{t=1}^{M_{ijk}} f_{ijk}^t(U_{ijkt}x).$$

We will call f_{ijk}^t term functions. The total number of these functions is

$$N_0 = \sum_{i=1}^{M} \sum_{j \in J_i} \sum_{k \in K_j} M_{ijk}.$$

For one evaluation of the function f we have to compute these functions N_0 times. Since for one evaluation of the discrete gradient we compute $n+1$ times the function f, the total number of computation of term functions for one evaluation of the discrete gradient is

$$N_t = (n+1)N_0.$$

For $p \in \{1, \ldots, n\}$ we introduce

$$Q_p^{ijk} = \left\{ t \in \{1, \ldots, M_{ijk}\} : U_{ijkt}^{pp} = 1 \right\},$$
$$\overline{Q}_p^{ijk} = \left\{ t \in \{1, \ldots, M_{ijk}\} : U_{ijkt}^{pp} = 0 \right\}.$$

It is clear that $M_{ijk} = card(Q_p^{ijk}) + card(\overline{Q}_p^{ijk})$. One can assume that $card(Q_p^{ijk}) \ll card(\overline{Q}_p^{ijk})$. For example, if all functions f_{ijk} are l-chained then

$$card(Q_p^{ijk}) \leqslant l \text{ and } card(\overline{Q}_p^{ijk}) \geqslant n - l - 1.$$

If these functions are separable then

$$card(Q_p^{ijk}) = 1 \text{ and } card(\overline{Q}_p^{ijk}) = n - 1.$$

Then the function f_{ijk} can be calculated at the point x^p using the following *simplified scheme*:

$$f_{ijk}(x^p) = \sum_{t \in Q_p^{ijk}} f_{ijk}^t(U_{ijkt}x^p) + \sum_{t \in \overline{Q}_p^{ijk}} f_{ijk}^t(U_{ijkt}x^{p-1}) \tag{17}$$

that is we compute only functions $f_{ijk}^t, t \in Q_p^{ijk}$ at the point x^p and all other functions remain the same as at the point x^{p-1}. Thus in order to calculate the function f at the point x^p we compute

$$N_s = \sum_{i=1}^{M} \sum_{j \in J_i} \sum_{k \in K_j} card(Q_p^{ijk})$$

times the term functions at this point. Since $card(Q_p^{ijk}) \ll M_{ijk}$ one can expect that $N_s \ll N_0$.

If all functions $f_{ijk}, i = 1, \ldots, M, j \in J_i, k \in K_j$ are l-chained then

$$N_s \leqslant l \sum_{i=1}^{M} \sum_{j \in J_i} card(K_j).$$

If all these functions are separable then

$$N_s = \sum_{i=1}^{M} \sum_{j \in J_i} card(K_j).$$

Thus in order to compute one discrete gradient at the point x with respect to the direction $g \in S_1$ we have to compute the function f at the points x and $x + \lambda g$ using formula (10) and at all other points $x_i^p, p = 1, \ldots, n, p \neq i$ it can be computed using simplified scheme (17). In this case the total number of computation of term functions is

$$N_{ts} = 2N_0 + (n-1)N_s$$

which is significantly less than N_t when n is large.

Now we consider one special case of functions (10).

5.2.1. *Functions Represented as a Sum of Minimum functions*

We consider the following functions:

$$f(x) = \sum_{i=1}^{M} \min_{k \in \bar{K}} f_{ik}(x^k), \tag{18}$$

where $\bar{K} = \{1, \ldots, K\}, x^k \in \mathbb{R}^n, x = (x^1, \ldots, x^K) \in \mathbb{R}^{K \times n}$ and the functions f_{ik} are separable

$$f_{ik}(x) = \sum_{j=1}^{n} f_{ijk}(x_j^k).$$

The function (18) can be derived from the function (10) when

$$J_i = \{1\}, \quad i = 1, \ldots, M, \quad K_j = \{1, \ldots, K\}.$$

In order to calculate one discrete gradient of the function (18) we have to evaluate $MK(n+1)$ times the functions f_{ijk}. However the use of the simplified scheme reduces this number to $2MK + n - 1$.

One of the special cases of functions (18) is the cluster function (4). This function can be rewritten as follows

$$f(x) = \sum_{i=1}^{M} \min_{k=1,\ldots,K} \|x^k - a^i\|^2, \quad x = (x^1, \ldots, x^K) \in \mathbb{R}^{K \times n}.$$

Here

$$f_{ik}(x^k) = \|x^k - a^i\|^2 \text{ and } f_{ijk}(x_j^k) = (x_j^k - a_j^i)^2.$$

For the computation of one discrete gradient without using simplified scheme we have to compute $MK(n+1)$ the very simple functions f_{ijk}, however the use of the simplified scheme allows one to reduce this number to $2MK + n - 1$. Since in the cluster analysis the number M is large we can assume that $MK \gg n$ and therefore

$$\frac{MK(n+1)}{2MK + n - 1} \approx \frac{n+1}{2}.$$

If n is large then we can significantly reduce computational efforts using the simplified scheme.

5.2.2. *Discrete Gradient Method*

In this subsection we briefly describe the discrete gradient method. A more detailed description of this method can be found in [3, 5].

We consider the following unconstrained minimization problem:

$$\text{minimize } f(x) \text{ subject to } x \in \mathbb{R}^n, \tag{19}$$

where the function f is assumed to be semismooth. An important step in the discrete gradient method is the calculation of a descent direction of the objective function f.

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

ALGORITHM 1. An algorithm for the computation of the descent direction.
Step 1. Choose any $g^1 \in S_1, e \in G, i \in I(g^1, \alpha)$ and compute a discrete gradient $v^1 = \Gamma^i(x, g^1, e, z, \lambda, \beta)$. Set $\overline{D}_1(x) = \{v^1\}$ and $k = 1$.
Step 2. Calculate the vector $\|w^k\|^2 = \min\{\|w\|^2 : w \in \overline{D}_k(x)\}$. If

$$\|w^k\| \leqslant \delta, \tag{20}$$

then stop. Otherwise go to Step 3.
Step 3. Calculate the search direction by $g^{k+1} = -\|w^k\|^{-1} w^k$.
Step 4. If

$$f(x + \lambda g^{k+1}) - f(x) \leqslant -c\lambda\|w^k\|, \tag{21}$$

then stop. Otherwise go to Step 5.
Step 5. Calculate a discrete gradient

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

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

EXPLANATIONS TO ALGORITHM 1. In Step 1 we calculate the first discrete gradient. The distance between the convex hull of all calculated discrete gradients and the origin is calculated in Step 2. 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 calculate 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 calculated, otherwise we calculate another discrete gradient with respect to this direction in Step 5 and add it to the set $\overline{D}_k$.

It is proved that Algorithm 1 is terminating (see [3, 5]).

Let numbers $c_1 \in (0, 1), c_2 \in (0, c_1]$ be given.

ALGORITHM 2. 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 1 for the calculation of the descent direction at $x = x_s^k, \delta = \delta_k, z = z_k, \lambda = \lambda_k, \beta = \beta_k, c = c_1$. This algorithm terminates after a finite number of iterations $m > 0$. As a result we get the set $D_m(x_s^k)$ and an element v_s^k such that

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

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

$$f(x_s^k + \lambda_k g_s^k) - f(x_s^k) \leqslant -c_1 \lambda_k \|v_s^k\|. \tag{22}$$

Step 4. If

$$\|v_s^k\| \leqslant \delta_k \tag{23}$$

then set $x^{k+1} = x_s^k$, $k = k+1$ and go to Step 2. Otherwise go to Step 5.
Step 5. Construct the following iteration $x_{s+1}^k = x_s^k + \sigma_s g_s^k$, where σ_s is defined as follows

$$\sigma_s = \operatorname{argmax}\{\sigma \geqslant 0 : f(x_s^k + \sigma g_s^k) - f(x_s^k) \leqslant -c_2 \sigma \|v_s^k\|\}.$$

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

The convergence of the discrete gradient method is studied in [3, 5].

6. Results of Numerical Experiments

A number of numerical experiments have been carried out using large scale nonsmooth optimization problems.

6.1. TEST PROBLEMS

The following test problems have been used in numerical experiments. The description of chained functions can be also found in [17, 22, 23]. We consider unconstrained minimization problems. Below f_* stands for the minimum value of a function f.

6.1.1. *Piecewise Chained Problems*

Problem 1. Chained LQ function

$$f(x) = \sum_{i=1}^{n-1} \max\left\{-x_i - x_{i+1}, -x_i - x_{i+1} + (x_i^2 + x_{i+1}^2 - 1)\right\},$$
$$f_* = -(n-1)\sqrt{2}.$$

Problem 2. Chained CB3 I function

$$f(x) = \sum_{i=1}^{n-1} \max\left\{x_i^4 + x_{i+1}^2, (2-x_i)^2 + (2-x_{i+1})^2, 2e^{-x_i + x_{i+1}}\right\},$$
$$f_* = 2(n-1).$$

Problem 3. Chained CB3 II function

$$f(x) = \max\left\{\sum_{i=1}^{n-1}(x_i^4 + x_{i+1}^2), \sum_{i=1}^{n-1}((2-x_i)^2 + (2-x_{i+1})^2), 2\sum_{i=1}^{n-1} e^{-x_i+x_{i+1}}\right\},$$

$f_* = 2(n-1)$.

Problem 4. Nonsmooth generalization of Brown function 2

$$f(x) = \sum_{i=1}^{n-1}\left(|x_i|^{x_{i+1}^2+1} + |x_{i+1}|^{x_i^2+1}\right),$$

$f_* = 0$.

Problem 5. Chained Mifflin 2 function

$$f(x) = \sum_{i=1}^{n-1}\left(-x_i + 2(x_i^2 - x_{i+1}^2 - 1) + 1.75|x_i^2 + x_{i+1}^2 - 1|\right),$$

f_* varies.

Problem 6. Chained Crescent I function

$$f(x) = \max\left\{\sum_{i=1}^{n-1}\left(x_i^2 + (x_{i+1}-1)^2 + x_{i+1} - 1\right),\right.$$
$$\left.\sum_{i=1}^{n-1}\left(-x_i^2 - (x_{i+1}-1)^2 + x_{i+1} + 1\right)\right\},$$

$f_* = 0$.

Problem 7. Chained Crescent II function

$$f(x) = \sum_{i=1}^{n-1}\max\left\{x_i^2 + (x_{i+1}-1)^2 + x_{i+1} - 1, -x_i^2 - (x_{i+1}-1)^2 + x_{i+1} + 1\right\},$$

$f_* = 0$.

Problem 8. Chained Wood function

$$f(x)=\sum_{j=1}^{k}[100(x_{2j-1}^2-x_{2j})^2+(x_{2j-1}-1)^2+90(x_{2j+1}^2-x_{2j+2})^2+(x_{2j+1}-1)^2$$
$$+10(x_{2j}+x_{2j+1}-2)^2+(x_{2j}-x_{2j+2})^2/10],\quad k=(n-2)/2,\quad f_*=0.$$

Problem 9. Chained Powell singular function

$$f(x)=\sum_{j=1}^{k}\left[(x_{2j-1}+10x_{2j})^2+5(x_{2j+1}-x_{2j+2})^2+(x_{2j}-2x_{2j+1})^4\right.$$
$$\left.+10(x_{2j-1}-x_{2j+2})^4\right],\quad k=(n-2)/2,\quad f_*=0.$$

6.1.2. *Piecewise Partially Separable Problems*

Problem 10. PPSF CB3 I function

$$f(x)=\sum_{i=1}^{n}\max\left\{x_i^4+x_1^2,(2-x_i)^2+(2-x_1)^2,2e^{-x_i+x_1}\right\},\quad f_*=2n.$$

Problem 11. PPSF CB3 II function

$$f(x)=\max\left\{\sum_{i=1}^{n}(x_i^4+x_1^2),\sum_{i=1}^{n}((2-x_i)^2+(2-x_1)^2),2\sum_{i=1}^{n}(e^{-x_i+x_1})\right\},$$
$$f_*=2n.$$

Problem 12. PPSF Nonsmooth generalization of Brown function 2

$$f(x)=\sum_{i=1}^{n}\left(|x_i|^{x_1^2+1}+|x_1|^{x_i^2+1}\right),\quad f_*=0.$$

Problem 13. PPSF Broyden function

$$f(x)=\sum_{i=1}^{n}|(3-2x_i)x_i-x_1-x_2+1|^{7/3}, \quad f_*=0.$$

It should be noted that Problems 8, 9 have smooth objective functions. The objective functions in Problems 10–13 are piecewise partially separable and they are modification of corresponding test functions from [22].

The code has been written in C++ and numerical experiments have been carried out on a PC Intel Pentium 4, 1.6 MHz. Their results are presented in Tables 1–3. In these tables we use the following notations:

- n is the number of variables;
- t the CPU time in seconds;
- N_f the number of evaluations of term functions when the simplified scheme is applied;
- N_S the number of objective function evaluations when the simplified scheme is applied;
- N_g the number of objective function evaluations without application of the simplified scheme;
- x^0 and x^* are the initial point and the minimizer, respectively.

We consider that starting from the point x^0 the algorithm succeeds if for the final point $\bar{x}$ the inequality

$$\frac{f(\bar{x})-f_*}{|f_*|+1}<\epsilon$$

is true. Otherwise we say that it fails. Here the tolerance $\epsilon=10^{-4}$.

In the numerical experiments for each problem and n we ran the algorithm starting from 100 randomly chosen points. In the tables we present average values of t, N_f, N_S and N_g/N_S. In the column "Failed" we present the number of failures of the algorithm. We also present the minimum and maximum values of the difference $f(x^0)-f(x^*)$ in order to demonstrate how far the initial points are from the solution.

Figures 1 and 3 show the dependence of N_S on the number of variables n for piecewise chained and piecewise partially separable functions, respectively. Figures 2 and 4 show the dependence of N_g/N_S on the number of variables n for these functions.

As one can see from Tables 1–3 the proposed algorithm allows us to solve all problems with a given accuracy except Problem 4 (with $n=2000$), Problem 6 (with $n=1000, 2000$), Problem 7 (with $n=100$–2000) and Problem 13 (with $n=800, 1000, 1500, 2000$). However, it should be noted that all problems, except Problem 7, have been solved with rougher accuracy. In the numerical

Table 1. Results for piecewise chained functions

							$f(x_0) - f(x^*)$	
	n	t	N_f	N_S	N_g/N_S	Failed	Min	Max
Chained LQ	2000	44.30	3.23e7	1.62e4	1800.0	0	1.27e05	1.43e05
	1500	23.60	1.84e7	1.23e4	1310.0	0	9.34e04	1.07e05
	1000	11.40	9.28e6	9.29e3	859.0	0	6.32e04	7.27e04
	800	8.31	6.66e6	8.34e3	686.0	0	4.84e04	5.80e04
	500	5.18	4.10e6	8.21e3	416.0	0	2.99e04	3.76e04
	300	3.26	2.38e6	7.95e3	244.0	0	1.79e04	2.22e04
	100	1.76	7.58e5	7.66e3	79.0	0	5.24e03	8.65e03
	50	1.31	3.02e5	6.17e3	39.9	0	2.33e03	4.47e03
	10	0.34	1.10e4	1.22e3	5.6	0	2.54e02	1.00e03
Chained CB3 I	2000	81.20	3.26e7	1.63e4	1380.0	0	2.73e09	7.80e09
	1500	49.00	2.13e7	1.42e4	963.0	0	1.86e09	6.71e09
	1000	25.60	1.14e7	1.14e4	633.0	0	9.85e08	4.39e09
	800	18.70	8.29e6	1.04e4	511.0	0	2.28e08	4.23e09
	500	9.25	4.20e6	8.42e3	309.0	0	3.29e08	3.13e09
	300	3.56	1.81e6	6.04e3	163.0	0	1.17e08	2.26e09
	100	0.75	3.93e5	3.97e3	49.2	0	2.95e06	9.50e08
	50	0.37	1.69e5	3.46e3	24.6	0	9.48e04	6.97e08
	10	0.03	1.44e4	1.60e3	3.0	0	3.96e03	3.55e08
Chained CB3II	2000	40.20	2.07e7	1.03e4	1250.0	0	2.92e09	7.65e09
	1500	18.2	1.06e7	7.04e3	806.0	0	2.01e09	5.76e09
	1000	8.22	5.19e6	5.19e3	488.0	0	9.61e08	4.42e09
	800	5.72	3.62e6	4.53e3	385.0	0	7.16e08	3.84e09
	500	2.93	1.87e6	3.76e3	243.0	0	3.73e08	2.99e09
	300	1.50	9.44e5	3.16e3	150.0	0	5.76e07	2.39e09
	100	0.50	2.37e5	2.39e3	54.0	0	1.05e07	1.40e09
	50	0.29	9.83e4	2.01e3	26.2	0	1.19e05	1.06e09
	10	0.02	1.03e4	1.15e3	3.9	0	4.38e03	2.42e08
Chained generalized Brown 2	2000	76.60	1.79e7	8.94e3	1850.0	5	8.86e02	9.37e02
	1500	32.30	7.76e6	5.18e3	1360.0	0	6.56e02	7.09e02
	1000	15.30	3.83e6	3.84e3	874.0	0	4.35e02	4.77e02
	800	9.75	2.50e6	3.12e3	689.0	0	3.49e02	3.77e02
	500	4.91	1.15e6	2.30e3	425.0	0	2.14e02	2.42e02
	300	3.24	5.78e5	1.93e3	250.0	0	1.28e02	1.45e02
	100	1.57	1.38e5	1.39e3	82.1	0	4.02e01	4.99e01
	50	2.48	5.56e4	1.14e3	40.3	0	1.86e01	2.64e01
	10	0.03	4.53e3	5.03e2	6.6	0	2.39e00	5.87e00

experiments we restricted the maximum number of discrete gradients which can be calculated at each iteration to 100. In all these problems in order to calculate solutions with higher accuracy we have to significantly increase this number. But in this case the CPU time may increase substantially.

Results for CPU time reported in the tables demonstrate that the algorithm is quite fast to find solutions with the given accuracy in problems up to 2000 variables.

The numbers presented in columns for the minimum and maximum values of the difference $f(x^0) - f_*$ show that randomly chosen initial points are not close to the solutions for all experiments. Therefore one can assert

Table 2. Results for piecewise chained functions

							$f(x_0) - f(x^*)$	
	n	t	N_f	N_S	N_g/N_S	Failed	Min	Max
Chained Crescent I	2000	10.70	8.84e6	4.42e3	1840.0	5	1.27e05	1.43e05
	1500	5.51	4.72e6	3.15e3	1350.0	3	9.48e04	1.09e05
	1000	2.45	2.23e6	2.23e3	875.0	0	6.26e04	7.05e04
	800	1.71	1.56e6	1.95e3	692.0	0	4.87e04	5.72e04
	500	0.99	8.62e5	1.73e3	428.0	0	2.93e04	3.60e04
	300	0.62	4.54e5	1.52e3	256.0	0	1.78e04	2.30e04
	100	0.30	1.19e5	1.20e3	82.9	0	5.37e03	8.04e03
	50	0.23	5.18e4	1.06e3	40.5	0	2.33e03	4.47e03
	10	0.03	5.78e3	6.42e2	6.7	0	2.83e02	9.95e02
Chained Crescent II	2000	25.80	2.15e7	1.08e4	1760.0	100	1.27e05	1.41e05
	1500	11.20	1.07e7	7.17e3	1250.0	99	9.40e04	1.06e05
	1000	4.45	5.14e6	5.15e3	779.0	100	6.20e04	7.23e04
	800	2.83	3.62e6	4.53e3	606.0	98	4.96e04	5.86e04
	500	1.30	1.84e6	3.69e3	361.0	99	3.01e04	3.64e04
	300	0.70	8.95e5	2.99e3	207.0	97	1.76e04	2.25e04
	100	2.48	2.53e5	2.55e3	71.2	61	5.11e03	8.56e03
	50	0.50	1.25e5	2.55e3	39.1	0	2.31e03	4.46e03
	10	0.26	7.51e3	8.34e2	6.0	0	2.61e02	9.23e02
Chained Mifflin	2000	85.20	7.18e7	3.59e4	1700.0	0	4.69e05	5.16e05
	1500	63.50	6.73e7	4.49e4	1140.0	0	3.48e05	3.91e05
	1000	22.50	2.83e7	2.83e4	708.0	0	2.29e05	2.68e05
	800	14.00	1.83e7	2.29e4	549.0	0	1.84e05	2.11e05
	500	6.13	8.08e6	1.62e4	331.0	0	1.13e05	1.33e05
	300	3.15	3.73e6	1.25e4	198.0	0	6.61e04	8.37e04
	100	1.60	7.97e5	8.05e3	75.0	0	1.76e04	3.10e04
	50	1.65	3.37e5	6.88e3	40.0	0	8.59e03	1.81e04
	10	0.91	1.39e4	1.54e3	6.0	0	9.15e02	3.84e03
Chained Wood	2000	62.00	4.27e7	2.14e4	1290.0	0	3.31e08	3.86e08
	1500	34.20	2.44e7	1.63e4	844.0	0	2.49e08	3.05e08
	1000	19.20	1.26e7	1.26e4	466.0	0	1.62e08	1.99e08
	800	14.00	8.77e6	1.10e4	345.0	0	1.22e08	1.72e08
	500	13.20	4.39e6	8.79e3	201.0	0	7.79e07	1.11e08
	300	10.40	2.25e6	7.51e3	117.0	0	4.20e07	6.59e07
	100	11.40	6.19e5	6.25e3	40.6	0	1.08e07	2.77e07
	50	11.10	2.73e5	5.57e3	22.7	0	4.35e06	1.28e07
	10	0.29	2.75e4	3.05e3	3.7	0	5.70e04	3.18e06
Chained Powell singular	2000	24.10	1.53e7	7.64e3	1040.0	0	1.48e08	1.88e08
	1500	13.90	9.13e6	6.09e3	658.0	0	1.10e08	1.52e08
	1000	8.89	5.29e6	5.30e3	402.0	0	7.19e07	1.01e08
	800	13.00	4.13e6	5.17e3	325.0	0	5.08e07	8.58e07
	500	6.42	2.52e6	5.04e3	219.0	0	3.36e07	5.68e07
	300	5.72	1.67e6	5.58e3	142.0	0	1.78e07	3.67e07
	100	5.69	5.40e5	5.45e3	44.4	0	4.33e06	1.36e07
	50	5.44	2.24e5	4.58e3	21.6	0	1.26e06	7.62e06
	10	0.21	3.79e4	4.22e3	3.3	0	2.49e04	1.77e06

Table 3. Results for piecewise partially separable functions

							$f(x_0) - f(x^*)$	
	n	t	N_f	N_S	N_g/N_S	Failed	Min	Max
PPSF CB3 I	2000	51.20	3.87e7	1.94e4	854.0	0	3.82e06	8.07e10
	1500	27.50	2.19e7	1.46e4	605.0	0	2.80e06	7.70e10
	1000	12.90	1.07e7	1.07e4	384.0	0	1.88e06	2.93e10
	800	9.26	7.85e6	9.82e3	298.0	0	1.49e06	3.99e10
	500	4.93	4.26e6	8.53e3	178.0	0	9.11e05	2.00e10
	300	2.71	2.28e6	7.63e3	103.0	0	5.17e05	1.89e10
	100	1.57	5.24e5	5.29e3	30.5	0	1.33e05	5.91e09
	50	2.26	2.01e5	4.09e3	15.5	0	6.04e04	3.07e09
	10	0.04	1.50e4	1.67e3	3.3	0	3.00e03	9.64e07
PPSF CB3 II	2000	24.40	2.12e7	1.06e4	777.0	0	3.74e06	9.08e10
	1500	11.70	1.08e7	7.20e3	540.0	0	2.76e06	6.83e10
	1000	5.62	5.35e6	5.36e3	341.0	0	1.84e06	2.67e10
	800	4.14	3.98e6	4.98e3	261.0	0	1.47e06	3.80e10
	500	2.24	2.17e6	4.35e3	163.0	0	9.04e05	2.08e10
	300	1.29	1.31e6	4.36e3	97.1	0	5.08e05	1.93e10
	100	0.86	9.97e5	1.01e4	29.8	0	1.36e05	3.98e09
	50	0.25	1.99e5	4.05e3	15.6	0	7.76e04	2.09e09
	10	0.02	1.08e4	1.19e3	3.4	0	3.68e03	4.93e08
PPSF generalized Brown 2	2000	79.60	3.74e7	1.87e4	975.0	3	5.00e02	1.32e03
	1500	39.80	1.94e7	1.30e4	724.0	0	4.01e02	9.97e02
	1000	16.30	8.22e6	8.23e3	473.0	0	2.47e02	6.59e02
	800	10.70	5.38e6	6.74e3	376.0	0	1.97e02	5.27e02
	500	5.10	2.40e6	4.81e3	232.0	0	1.24e02	3.23e02
	300	3.60	1.24e6	4.13e3	138.0	0	7.74e01	1.98e02
	100	2.98	2.74e5	2.77e3	45.9	0	2.28e01	6.68e01
	50	2.89	1.01e5	2.07e3	23.0	0	1.23e01	3.41e01
	10	0.02	6.21e3	6.89e2	4.7	0	1.50e00	6.79e00
PPSF Broyden	2000	54.50	6.18e7	3.09e4	525.0	93	1.25e07	2.21e07
	1500	19.10	2.44e7	1.63e4	329.0	82	8.85e06	1.72e07
	1000	8.41	1.16e7	1.16e4	199.0	32	6.25e06	1.10e07
	800	5.59	7.99e6	1.00e4	164.0	8	5.05e06	8.75e06
	500	2.48	3.32e6	6.66e3	109.0	0	2.62e06	5.73e06
	300	1.32	1.58e6	5.29e3	70.9	0	1.75e06	3.35e06
	100	0.57	2.96e5	2.99e3	26.5	0	5.27e05	1.36e06
	50	0.62	1.26e5	2.56e3	14.0	0	2.39e05	6.85e05
	10	0.06	1.60e4	1.78e3	3.1	0	1.05e04	1.95e05

that the number of objective function evaluations N_S is moderate for all problems and n. We can also see from Figures 1 and 3 that the number N_S seems a linear function of the number of variables for all problems for which the algorithm was successful.

The ratio N_g/N_S increases as the number of variables increases. Figures 2 and 4 demonstrate that this ratio is a linear function of the number of variables and $N_g/N_S \approx \alpha n$ where $\alpha = 0.30$–0.95.

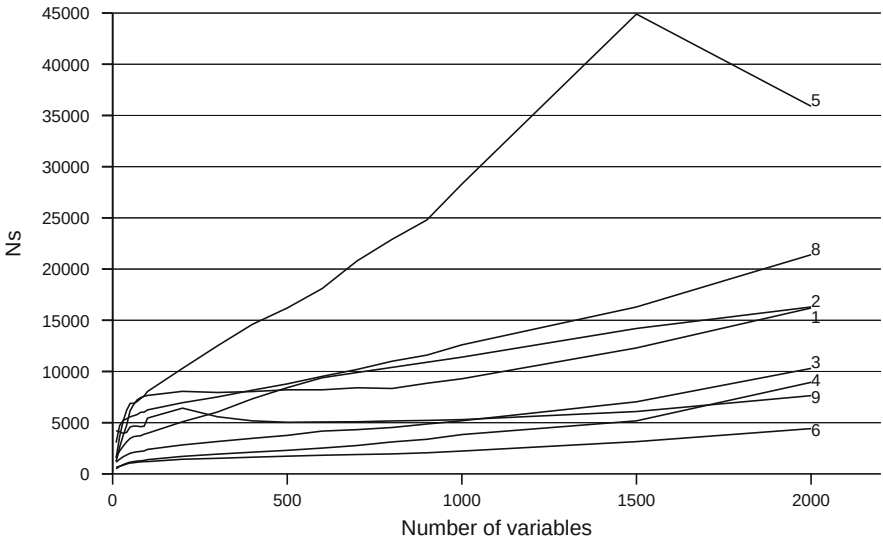

Figure 1. Average number of function evaluations for piecewise chained functions.

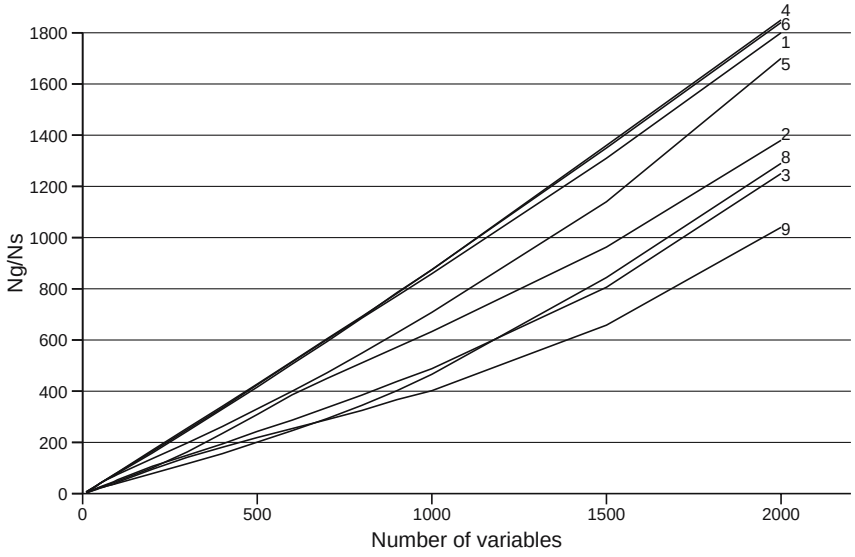

Figure 2. Average ratio of the number of function evaluations for general scheme to simplified scheme for piecewise chained functions.

The numerical results show that the number of term function evaluations is small in all cases, and therefore from a more practical viewpoint, almost all problems were solved in less than one minute on a normal PC.

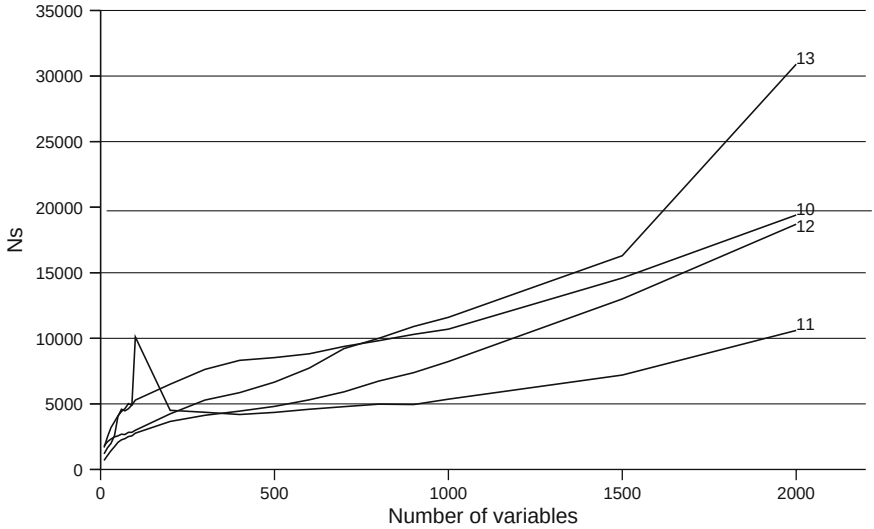

Figure 3. Average number of function evaluations for PPS functions.

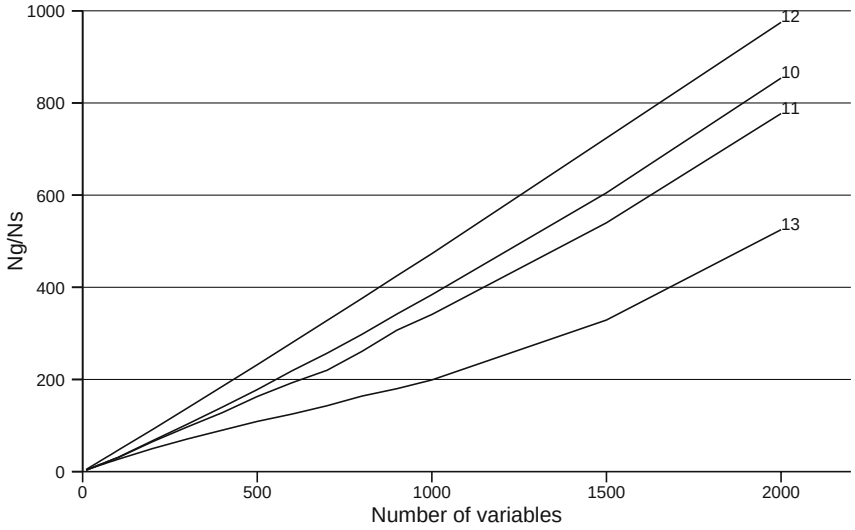

Figure 4. Average ratio of the number of function evaluations for general scheme to simplified scheme for PPS functions.

7. Conclusions

In this paper we have developed an algorithm for solving one class of large scale nonsmooth optimization. This class contains piecewise partially separable functions. These functions have many practical applications including applications in data mining and information retrieval. An algorithm for minimization of these functions is the modification of the discrete gradient

method. It has been shown that the calculation of discrete gradients can be significantly accelerated. We present results of preliminary numerical experiments which demonstrate that the proposed algorithm is efficient for solving many large scale nonsmooth optimization problems up to 2000 variables.

As it was pointed out above in this paper the discrete gradient method consists of three major steps: the computation of the discrete gradients, the computation of a descent direction by solving a certain quadratic programming problem and a line search. The simplified scheme proposed in this paper allows one to significantly accelerate the computation of the discrete gradients. However, the acceleration of the two other steps taking into account the structure of problems may lead to more efficient algorithms to solve a broad class of large scale nonsmooth optimization problems. This will be the subject of our further research.

Acknowledgement

This research was supported by the Australian Research Council.

References

1. Aberick, B.M., Bicshof, C.H., Carle, A., More, J. and Griewank, A. (1994), Computing large sparse Jacobian matrices using automatic differentiation, *SIAM Journal on Scientific and Statistical Computing* 15, 285–294.
2. Bagirov, A.M. and Gasanov, A.A. (1995), A method of approximating a quasidifferential, *Russian Journal of Computational Mathematics and Mathematical Physics* 35(4), 403–409.
3. Bagirov, A.M. (1999), Minimization methods for one class of nonsmooth functions and calculation of semi-equilibrium prices, In: Eberhard, A. et al. (eds.) *Progress in Optimization: Contribution from Australasia*, Kluwer Academic Publishers, pp. 147–175.
4. Bagirov, A.M. (2002), A method for minimzation of quasidifferentiable functions, *Optimization Methods and Software* 17(1), 31–60.
5. Bagirov, A.M. (2003), Continuous subdifferential approximations and their applications, *Journal of Mathematical Sciences* 115(5), 2567–2609.
6. Bagirov, A.M., Rubinov, A.M., Soukhoroukova, N.V. and Yearwood, J. (2003), Unsupervised and supervised data classification via nonsmooth and global optimization, *TOP: Spanish Journal of Operations Research* 1–93.
7. Bagirov, A.M. and Ugon, J. (2005), An algorithm for minimizing clustering functions, *Optimization* 54(4–5), 351–368.
8. Bagirov, A.M. Max-min separability, *Optmization Methods and Software* 20(2–3), 271–290.
9. Bagirov, A.M. and Yearwood, J. A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems, *European Journal of Operational Research* 170(2), 578–596.
10. Bock, H.H. (1974), *Automatische Klassifikation*, Vandenhoeck & Ruprecht, Gottingen.
11. Clarke, F.H. (1983), *Optimization and Nonsmooth Analysis*, Wiley, New York.
12. Colson, B. and Toint, Ph.L. (2002), A derivative-free algorithm for sparse unconstrained optimization problems, In: Siddiqi, A.H., and Kocvara, M. (eds.), *Trends in Industrial and Applied Mathematics*, Kluwer Academic Publishers, Dordrecht, pp. 131–147.

13. Conn, A.R., Gould, N. and Toint, Ph.L. (1994), Improving the decomposition of partially separable functions in the context of large-scale optimization: a first approach, In: Hager, W.W., Hearn, D.W., and Pardalos, P.M. (eds.), *Large Scale Optimization: State of the Art*, Kluwer Academic Publishers, Dordrecht, pp. 82–94.
14. Demyanov, V.F. and Rubinov, A.M. (1995), *Constructive Nonsmooth Analysis*, Peter Lang, Frankfurt am Main.
15. Evtushenko, Yu.G. (1972), A numerical method for finding best guaranteed estimates, *USSR Journal of Computational Mathematics and Mathematical Physics* 12, 109–128.
16. Griewank, A. and Toint, Ph.L. (1982), On the unconstrained optimization of partially separable functions, In: Powell, M.J.D. (ed.), *Nonlinear Optimization*, Academic Press, pp. 301–312.
17. Haarala, M., Miettinen, K. and Makela, M.M. (2004), New-limited memory bundle method for large-scale nonsmooth optimization, *Optimization Methods and Software* 19(6), 673–692.
18. Hansen, P. and Jaumard, B. (1997), Cluster analysis and mathematical programming, *Mathematical Programming* 79(1–3), 191–215.
19. Hiriart-Urruty, J.-P. and Lemarechal, C. (1993), *Convex Analysis and Minimization Algorithms,* Vol. 1 and 2, Springer-Verlag, Berlin, New York.
20. Kiwiel, K.C. (1985), *Methods of Descent for Nondifferentiable Optimization, Lecture Notes in Mathematics*, 1133, Springer-Verlag, Berlin.
21. Jain, A.K., Murty, M.N. and Flynn, P.J. (1999), Data clustering: a review, *ACM Computing Surveys* 31(3), 264–323.
22. Luksan, L. and Vlcek, J. (1999), Sparse and partially separable test problems for unconstrained and equality constrained optimization, Technical Report 767, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague.
23. Luksan, L. and Vlcek, J. (2000), Test problems for nonsmooth unconstrained and linearly constrained optimization, Technical Report 798, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague.
24. Makela, M.M. and Neittaanmaki, P. (1992), *Nonsmooth Optimization*, World Scientific, Singapore.
25. Mifflin, R. (1977), Semismooth and semiconvex functions in constrained optimization, *SIAM Journal on Control and Optimization* 15(6), 959–972.
26. Nelder, J.A. and Mead, R. (1965), A simplex method for function minimization, *Computer Journal* 7, 308–313.
27. Powell, M.J.D. (2002), UOBYQA: unconstrained optimization by quadratic approximation, *Mathematical Programming* Series B, 92(3), 555–582.