A quasisecant method for minimizing nonsmooth functions

Adil M. Bagirov and Asef Nazari Ganjehlou

Centre for Informatics and Applied Optimization, School of Information Technology and Mathematical Sciences, University of Ballarat, Victoria 3353, Australia

Abstract

In this paper a new algorithm to locally minimize nonsmooth, nonconvex functions is developed. We introduce the notion of secants and quasisecants for nonsmooth functions. The quasisecants are applied to find descent directions of locally Lipschitz functions. We design a minimization algorithm which uses quasisecants to find descent directions. We prove that this algorithm converges to Clarke stationary points. Numerical results are presented demonstrating the applicability of the proposed algorithm in wide variety of nonsmooth, nonconvex optimization problems. We also, compare the proposed algorithm with the bundle method using numerical results.

Keywords: nonsmooth optimization, nonconvex optimization, subdifferential, bundle method.

Mathematical Subject Classification (2000): 65K05, 90C25.

1 Introduction

In this paper we develop an algorithm for solving the following unconstrained minimization problem:

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

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

Numerical methods for solving Problem (1), with different assumptions on the objective function f, have been studied extensively. Subgradient methods [24], bundle methods and its variations [10, 12, 13, 14, 15, 16, 17, 18, 21, 22, 25, 27], a discrete gradient method [1, 2, 4], a gradient sampling method [7] and methods based on smoothing techniques [23] are among them.

Subgradient methods are quite simple, however, they are not effective to solve many nonsmooth optimization problems. Algorithms based on smoothing techniques are applicable only to the special class of nonsmooth functions such as the maximum functions, max-min type functions. Bundle type algorithms are more general algorithms and they build up information about the subdifferential of the objective function using ideas known as bundling and aggregation. One of the important features of bundle methods is that they use subgradients from previous iterations which are still relevant at the current iteration. Relevancy here means that subgradients from those iterations are still good underestimator at the current iteration. In the case of convex functions, it is easy to identify relevant subgradients from the previous iterations, and bundle type algorithms are known to be very efficient for nonsmooth, convex problems. For nonconvex functions subgradient information is meaningful only locally and must be discounted when no longer relevant. Special rules have to be designed to identify those irrelevant subgradients. As a result bundle type algorithms are more complicated in the nonconvex case [7].

The gradient sampling method proposed in [7] does not require any rules for discounting irrelevant subgradients. It is a stochastic method which relies on the fact that locally Lipschitz functions are almost everywhere differentiable. At each iteration of this method a given number of gradients are evaluated from some neighborhood of a current point.

In this paper, we propose a new algorithm for solving Problem (1). It is efficient for minimization of nonsmooth nonconvex functions. We introduce the notions of a secant and a quasisecant for locally Lipschitz functions. The new minimization algorithm uses quasisecants to compute descent directions. There are some similarities between the proposed method on one side and the bundle and gradient sampling methods on the other side. In the new method we build up information about the approximation of the subdifferential using bundling idea which makes it similar to bundle type methods. We compute subgradients from a given neighborhood of a current point which makes the new method similar to the gradient sampling method.

We present results of numerical experiments. In these experiments the new algorithm is applied to solve well known nonsmooth optimization test problems. We also compare the proposed algorithm with the bundle method using numerical results.

The structure of the paper is as follows: The notions of a secant and a quasisecant are introduced in Section 2. In Section 3, we describe an algorithm for the computation of a descent direction. A secant method is introduced and its convergence is studied in Section 4. Results of numerical experiments are given in Section 5. Section 6 concludes the paper.

We use the following notation in this paper. $\mathbb{R}^n$ is an n-dimensional Euclidean space, $\langle u, v\rangle = \sum_{i=1}^n u_i v_i$ is an inner product in $\mathbb{R}^n$ and $\|\cdot\|$ is the associated Euclidean norm, $\partial f(x)$ is the Clarke subdifferential of the Lipschitz function f at a point x [8], co denotes the convex hull of a set, $(x, y) = \{z \in R^n : \ z = \alpha x + (1-\alpha)y, \ \alpha \in (0,1)\}$ is an open line segment joining points x and y, $S_1 = \{x \in \mathbb{R}^n : \ \|x\| = 1\}$ is the unit sphere, $B_\varepsilon(x) = \{y \in \mathbb{R}^n : \ \|y - x\| < \varepsilon\}, \ \varepsilon > 0$.

2 Secants and quasisecants

The concept of secants is widely used in optimization. For example the secants have been used to design quasi-Newton methods. In this section we introduce the notion of secants and quasisecants for locally Lipschitz functions. We start with univariate functions.

Consider a function $\varphi : \mathbb{R}^1 \to \mathbb{R}^1$ and assume that it is locally Lipschitz. A secant is a line passing through at least two points on the graph of the function φ. A straight line passing through points $(x, \varphi(x))$ and $(x+h, \varphi(x+h))$, where $h > 0$, is given by:

$$l(x) = cx + d$$

where

$$c = \frac{\varphi(x+h) - \varphi(x)}{h}, \quad d = \varphi(x) - cx.$$

The equation

$$\varphi(x+h) - \varphi(x) = ch \tag{2}$$

is called a secant equation (see Fig. 1). For smooth functions the secant line is close to the tangent line if h is sufficiently small.

We can give another representation of the number c in the secant equation using the Lebourg's mean value theorem (see [8]). This theorem implies that there exist $y \in (x, x+h)$ and $u \in \partial\varphi(y)$ such that

$$\varphi(x+h) - \varphi(x) = uh,$$

and one can take $c = u$. However, it is not easy to compute the point y and consequently the subgradient u.

Now let us consider a locally Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}^1$. For given $x \in \mathbb{R}^n, g \in S_1$ and $h > 0$ consider the following function $\varphi(t) = f(x + tg), \ t \in \mathbb{R}^1$. It is obvious that

$$f(x + hg) - f(x) = \varphi(h) - \varphi(0).$$

One can give the following definition of secants for the function f by generalizing the definition for the univariate function φ.

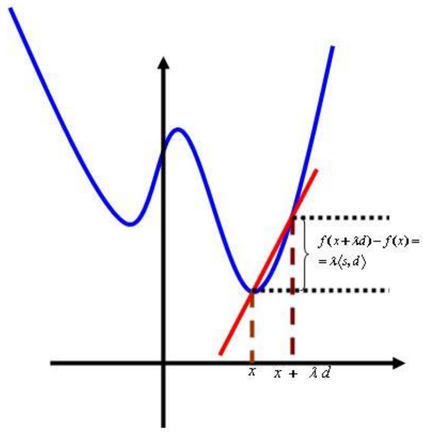

Figure 1: Secants for a univariate function

Definition 1 *A vector $u \in \mathbb{R}^n$ is called a secant of the function f at the point x in the direction $g \in S_1$ with the length $h > 0$ if*

$$f(x + hg) - f(x) = h\langle u, g\rangle. \tag{3}$$

From now on we will use the notation $u(x, g, h)$ for any secant of the function f at a point x in the direction $g \in S_1$ with the length $h > 0$.

For a given $h > 0$, consider a set-valued mapping $x \mapsto Sec(x, h)$:

$$Sec(x, h) = \{w \in \mathbb{R}^n : \ \exists (g \in S_1), \ w = u(x, g, h)\}.$$

Note that a secant with respect to a given direction is not unique, and there are many vectors u satisfying the equality (3). Consequently, the mapping $x \mapsto Sec(x, h)$ can be defined in many different ways. However, only secants approximating subgradients of the function f are of interest. One such secant is given by the Lebourg's mean value theorem when the function f is locally Lipschitz. The Lebourg's theorem implies that there exist $y \in (x, x + hg)$ and $u \in \partial f(y)$ such that

$$f(x + hg) - f(x) = h\langle u, g\rangle.$$

Then it is clear that the subgradient u is a secant at a point x. However it is not easy to compute such subgradients.

Consider the following set at a point x:

$$SL(x) = \left\{ w \in \mathbb{R}^n : \ \ \exists (g \in S_1, \{h_k\}, \ h_k \downarrow 0 \text{ as } k \to \infty) : \ w = \lim_{k\to\infty} u(x, g, h_k) \right\}.$$

A mapping $x \mapsto Sec(x, h)$ is called a subgradient-related (SR)-secant mapping if the corresponding set $SL(x) \subseteq \partial f(x)$ for all $x \in \mathbb{R}^n$.

Computation of secants is not always easy task. For this reason, we introduce the notion of quasisecants by replacing strict equality in the definition of secants by inequality.

Definition 2 *A vector $v \in \mathbb{R}^n$ is called a quasisecant of the function f at the point x in the direction $g \in S_1$ with the length $h > 0$ if*

$$f(x + hg) - f(x) \leq h\langle v, g\rangle.$$

Fig. 2 presents examples of quasisecants in univariate case.

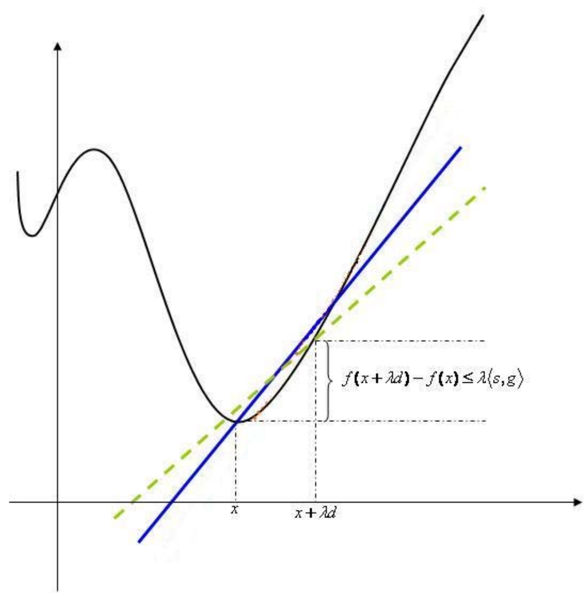

Figure 2: Quasisecants for a univariate function

We will use the notation $v(x, g, h)$ for any quasisecant of the function f at the point x in the direction $g \in S_1$ with the length $h > 0$. It is clear that any secant is also quasisecant. Therefore, the computation of quasisecants must be easier than the computation of secants.

For a given $h > 0$ consider the set of quasisecants of the function f at a point x:

$$QSec(x, h) = \{w \in \mathbb{R}^n : \quad \exists(g \in S_1), \quad w = v(x, g, h)\}$$

and the set of limit points of quasisecants as $h \downarrow 0$:

$$QSL(x) = \left\{ w \in \mathbb{R}^n : \quad \exists (g \in S_1, \{h_k\}, \; h_k \downarrow 0 \text{ as } k \to \infty) : \quad w = \lim_{k \to \infty} v(x, g, h_k) \right\}.$$

A mapping $x \mapsto QSec(x, h)$ is called a subgradient-related (SR)-quasisecant mapping if the corresponding set $QSL(x) \subseteq \partial f(x)$ for all $x \in \mathbb{R}^n$. In this case elements of $QSec(x, h)$ are called SR-quasisecant. In the sequel, we will consider sets $QSec(x, h)$ which contain only SR-quasisecant. Next we will present classes of functions for which SR-quasisecants can be efficiently computed.

2.1 Quasisecants of smooth functions

Assume that the function f is continuously differentiable. Then

$$v(x, g, h) = \nabla f(x + hg) + \alpha g, \; g \in S_1, \; h > 0$$

where

$$\alpha = \frac{f(x + hg) - f(x) - h\langle \nabla f(x + hg), g \rangle}{h}$$

is a secant (also quasisecant) at a point x with respect to the direction $g \in S_1$. Since the function f is continuously differentiable, it is clear that $v(x, g, h) \to \nabla f(x)$ as $h \downarrow 0$, which means that $v(x, g, h)$ is SR-quasisecant at the point x.

2.2 Quasisecants of convex functions

Assume that the function f is proper convex. Since

$$f(x + hg) - f(x) \leq h\langle v, g \rangle, \quad \forall v \in \partial f(x + hg),$$

any $v \in \partial f(x + hg)$ is a quasisecant at the point x. Then we have

$$QSec(x, h) = \bigcup_{g \in S_1} \partial f(x + hg).$$

The upper semicontinuity of the mapping $x \mapsto \partial f(x)$ implies that the set $QSL(x) \subset \partial f(x)$. This means that any $v \in \partial f(x + hg)$ is a SR-quasisecant at the point x.

2.3 Quasisecants of maximum functions

Consider the following maximum function:

$$f(x) = \max_{i=1,\ldots,m} f_i(x)$$

where the functions $f_i, \ i = 1, \ldots, m$ are continuously differentiable. Let $v^i \in \mathbb{R}^n$ be a SR-quasisecant of the function f_i at a point x. For any $g \in S_1$ consider the following set

$$R(x + hg) = \{i \in \{1, \ldots, m\} : \ f_i(x + hg) = f(x + hg)\}.$$

The set $QSec(x, h)$ of quasisecants at a point x is defined as

$$QSec(x, h) = \bigcup_{g \in S_1} \left\{ v^i(x, g, h), \quad i \in R(x + hg) \right\}.$$

Since the mapping $x \mapsto \partial f(x)$ is upper semicontinuous, the set $QSL(x) \subset \partial f(x)$, and quasisecants, which are defined above, are also SR-quasisecants.

2.4 Quasisecants of d.c. functions

Consider the function

$$f(x) = f_1(x) - f_2(x)$$

where functions f_1 and f_2 are proper convex. Take any subgradients $v^1 \in \partial f_1(x + hg), \ v^2 \in \partial f_2(x)$. Then the vector $v = v^1 - v^2$ is a quasisecant of the function f at a point x. However, such quasisecants need not to be SR-quasisecants. Since d.c. functions are quasidifferentiable ([9]) and if additionally subdifferentials $\partial f_1(x)$ and $\partial f_2(x)$ are polytopes, one can use an algorithm from [3, 4] to compute subgradients v^1 and v^2 such that their difference will converge to a subgradient of the function f at the point x. Thus, we can use this algorithm to compute SR-quasiseacnts of the function f.

The following important nonsmooth functions are d.c. functions and their subdifferential and superdifferential are polytopes:

$$F_1(x) = \sum_{i=1}^{m} \min_{j=1,\ldots,p} f_{ij}(x),$$

$$F_2(x) = \max_{i=1,\ldots,m} \ \min_{j=1,\ldots,p} f_{ij}(x).$$

Here functions f_{ij} are continuously differentiable and proper convex. Both functions F_1 and F_2 can be represented as difference of two convex function. One can compute their SR-quasisecants using their d.c. representation. Many interesting functions belong to this class of functions. For example, the error function in cluster analysis is of this type ([6]). Continuous piecewise linear functions can be represented as a max-min of linear functions.

Results of this section demonstrate that SR-quasisecants can be efficiently computed for a large class of nonsmooth convex and nonconvex functions. However, the computation of SR-quasisecants for d.c. functions is more costly than for other functions considered in this section.

SR-quasisecants defined in Subsections 2.1 - 2.4 satisfy the following condition: for any $\varepsilon > 0$ there exists $\delta > 0$ such that

$$QSec(y,h) \subset \partial f(x) + B_\varepsilon(0) \tag{4}$$

for all $x \in B_\delta(x)$ and $h \in (0,\delta)$. This can be easily proved taking into account upper semicontinuity of the subdifferential.

3 Computation of a descent direction

From now on, we will assume that for any bounded subset $X \subset \mathbb{R}^n$ and any $h_0 > 0$ there exists $K > 0$ such that

$$\|v\| \leq K$$

for all $v \in QSec(x,h),\ x \in X$ and $h \in (0,h_0]$. It is obvious that this assumption is true for all functions considered in the previous section. Given $x \in \mathbb{R}^n$ and $h > 0$ we consider the following set:

$$W(x,h) = \overline{\text{co}}\ QSec(x,h).$$

where $\overline{\text{co}}$ is a closed convex hull of a set. It is clear that the set $W(x,h)$ is compact and convex.

Proposition 1 *Assume that* $0_n \notin W(x,h),\ h > 0$ *and*

$$\|v^0\| = \min\left\{\|v\| :\ v \in W(x,h)\right\} > 0.$$

Then

$$f(x+hg^0) - f(x) \leq -h\|v^0\|$$

where $g^0 = -\|v^0\|^{-1}v^0$.

Proof: Since $W(x,h)$ is compact and convex set we get

$$\max\left\{\langle v, g^0\rangle :\ v \in W(x,h)\right\} = -\|v^0\|.$$

Then it follows from the definition of quasisecants that

$$\begin{aligned} f(x+hg^0) - f(x) &\leq h\langle v(x,g^0,h), g^0\rangle \\ &\leq h\max\left\{\langle v, g^0\rangle : v \in W(x,h)\right\} \\ &= -h\|v^0\|. \end{aligned}$$

△

Proposition 1 implies that quasisecants can be used to find descent directions of a function f. Furthermore, this can be done for any $h > 0$. However it is not always possible to apply Proposition 1 since it assumes the entire set $W(x,h)$ to be known. However, the computation of the entire set $W(x,h)$ is not always possible. Even it cannot be computed for functions whose subdifferentials are polytopes at any point. Therefore, we propose the following algorithm for computation of descent directions and this algorithm uses only a few elements from $W(x,h)$.

Let the numbers $h > 0,\ c_1 \in (0,1)$ and a small enough number $\delta > 0$ be given.

Algorithm 1 Computation of the descent direction.

Step 1. Choose any $g^1 \in S_1$ and compute a quasisecant $v^1 = v(x, g^1, h)$ in the direction g^1. Set $V_1(x) = \{v^1\}$ and $k = 1$.

Step 2. Compute $\|\bar{v}^k\|^2 = \min\{\|v\|^2 : v \in \operatorname{co} V_k(x)\}$. If

$$\|\bar{v}^k\| \leq \delta, \tag{5}$$

then stop. Otherwise go to Step 3.

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

Step 4. If

$$f(x + hg^{k+1}) - f(x) \leq -c_1 h \|\bar{v}^k\|, \tag{6}$$

then stop. Otherwise go to Step 5.

Step 5. Compute a quasisecant $v^{k+1} = v(x, g^{k+1}, h)$ in the direction g^{k+1}, construct the set $V_{k+1}(x) = \operatorname{co}\left\{V_k(x) \bigcup \{v^{k+1}\}\right\}$, set $k = k+1$ and go to Step 2.

Some explanations on Algorithm 1 follow. In Step 1 we select any direction $g^1 \in S_1$ and compute the initial quasisecant in this direction. The least distance between the convex hull of all computed quasisecants and the origin is found in Step 2. This is a quadratic programming problem and algorithms from [11, 26] can be applied to solve it. In numerical experiments we use the algorithm from [26]. If the least distance is less than a given tolerance $\delta > 0$, then the point x is an approximate stationary point; otherwise, we compute a new search direction in Step 3. If it is the descent direction satisfying (6) then the algorithm stops (Step 4). Otherwise, we compute a new quasisecant in the direction g^{k+1} in Step 5 which improves the approximation of the set $W(x,h)$.

It should be noted that there are some similarities between the ways descent directions are computed in the bundle-type algorithms and in Algorithm 1. The latter algorithm is close to the version of the bundle method proposed in [25]. However in the new algorithm we use quasisecants instead of subgradients.

In the next proposition using standard technique we prove that Algorithm 1 is a terminating.

Proposition 2 *Assume that f is a locally Lipschitz function, $h > 0$ and there exists $K, 0 < K < \infty$ such that*

$$\max\{\|v\| : v \in W(x,h)\} \leq K. \tag{7}$$

If $c_1 \in (0,1)$ and $\delta \in (0,K)$, then Algorithm 1 terminates after at most m steps, where

$$m \leq 2\log_2(\delta/K)/\log_2 K_1 + 2, \quad K_1 = 1 - [(1-c_1)(2K)^{-1}\delta]^2.$$

Proof: In order to prove the proposition it is sufficient to estimate an upper bound for the number of steps m when the condition (5) is satisfied. If both stopping criteria (5) and (6) are not satisfied, then a new quasisecant v^{k+1} computed in Step 5 does not belong to the set $V_k(x)$: $v^{k+1} \notin V_k(x)$. Indeed, in this case $\|\bar{v}^k\| > \delta$ and

$$f(x + hg^{k+1}) - f(x) > -c_1 h \|\bar{v}^k\|.$$

It follows from the definition of the quasisecants that

$$f(x + hg^{k+1}) - f(x) \leq h\langle v^{k+1}, g^{k+1}\rangle,$$

and we have

$$\langle v^{k+1}, \bar{v}^k\rangle < c_1\|\bar{v}^k\|^2. \tag{8}$$

Since $\bar{v}^k = \operatorname{argmin}\{\|v\|^2 : v \in V_k(x)\}$, the necessary condition for a minimum implies that

$$\langle \bar{v}^k, v\rangle \geq \|\bar{v}^k\|^2$$

for all $v \in V_k(x)$. The inequality along with (8) means that $v^{k+1} \notin V_k(x)$. Thus, if both stopping criteria are not satisfied then the algorithm allows one to improve the approximation of the set $W(x,h)$.

It is clear that $\|\bar{v}^{k+1}\|^2 \leq \|tv^{k+1} + (1-t)\bar{v}^k\|^2$ for all $t \in [0,1]$ which means

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

It follows from (7) that

$$\|v^{k+1} - \bar{v}^k\| \leq 2K.$$

Hence, taking into account the inequality (8), we have

$$\|\bar{v}^{k+1}\|^2 \leq \|\bar{v}^k\|^2 - 2t(1-c_1)\|\bar{v}^k\|^2 + 4t^2K^2.$$

Let $t_0 = (1-c_1)(2K)^{-2}\|\bar{v}^k\|^2$. It is clear that $t_0 \in (0,1)$ and therefore

$$\|\bar{v}^{k+1}\|^2 \leq \left\{1 - [(1-c_1)(2K)^{-1}\|\bar{v}^k\|]^2\right\}\|\bar{v}^k\|^2. \tag{9}$$

Since $\|\bar{v}^k\| > \delta$ for all $k = 1, \ldots, m-1$, it follows from (9) that

$$\|\bar{v}^{k+1}\|^2 \leq \{1 - [(1-c_1)(2K)^{-1}\delta]^2\}\|\bar{v}^k\|^2.$$

Let $K_1 = 1 - [(1-c_1)(2K)^{-1}\delta]^2$. Then $K_1 \in (0,1)$ and we have

$$\|\bar{v}^m\|^2 \leq K_1\|\bar{v}^{m-1}\|^2 \leq \ldots \leq K_1^{m-1}\|\bar{v}^1\|^2 \leq K_1^{m-1}K^2.$$

Thus, the inequality (5) is satisfied if $K_1^{m-1}K^2 \leq \delta^2$. This inequality must happen after at most m steps where

$$m \leq 2\log_2(\delta/K)/\log_2 K_1 + 2.$$

△

Definition 3 *A point $x \in \mathbb{R}^n$ is called a (h,δ)-stationary point if*

$$\min_{v \in W(x,h)} \|v\| \leq \delta$$

4 A secant method

In this section we describe the secant method for solving problem (1). Let $h > 0,\ \delta > 0,\ c_1 \in (0,1), c_2 \in (0, c_1]$ be given numbers.

Algorithm 2 The secant method for finding (h,δ)-stationary points.

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

Step 2. Apply Algorithm 1 for the computation of the descent direction at $x = x^k$ for given $\delta > 0$ and $c_1 \in (0,1)$. This algorithm terminates after a finite number of iterations $m > 0$. As a result, we get the set $V_m(x^k)$ and an element v^k such that

$$\|v^k\|^2 = \min\left\{\|v\|^2 : \ v \in V_m(x^k)\right\}.$$

Furthermore, either $\|v^k\| \le \delta$ or for the search direction $g^k = -\|v^k\|^{-1}v^k$

$$f(x^k + hg^k) - f(x^k) \le -c_1 h\|v^k\|. \tag{10}$$

Step 3. If

$$\|v^k\| \le \delta \tag{11}$$

then stop. Otherwise go to Step 4.

Step 4. Compute $x^{k+1} = x^k + \sigma_k g^k$, where σ_k is defined as follows

$$\sigma_k = \text{argmax}\left\{\sigma \ge 0 : \ f(x^k + \sigma g^k) - f(x^k) \le -c_2\sigma\|v^k\|\right\}.$$

Set $k = k + 1$ and go to Step 2.

Theorem 1 *Assume that the function f is bounded below*

$$f_* = \inf\{f(x) : x \in \mathbb{R}^n\} > -\infty. \tag{12}$$

Then Algorithm 2 terminates after finite many iterations $M > 0$ and produces (h, δ)-stationary point x^M where

$$M \le M_0 \equiv \left\lfloor \frac{f(x^0) - f_*}{c_2 h\delta} \right\rfloor + 1$$

Proof: Assume the contrary. Then the sequence $\{x^k\}$ is infinite and points x^k are not (h, δ)-stationary points. This means that

$$\min\{\|v\| : \ v \in W(x^k, h)\} > \delta, \quad \forall k = 1, 2, \ldots.$$

Therefore, Algorithm 1 will find descent directions and the inequality (10) will be satisfied at each iteration k. Since $c_2 \in (0, c_1]$, it follows from (10) that $\sigma_k \ge h$. Therefore, we have

$$\begin{aligned} f(x^{k+1}) - f(x^k) &< -c_2\sigma_k\|v^k\| \\ &\le -c_2 h\|v^k\|. \end{aligned}$$

Since $\|v^k\| > \delta$ for all $k \ge 0$, we get

$$f(x^{k+1}) - f(x^k) \le -c_2 h\delta,$$

which implies

$$f(x^{k+1}) \le f(x^0) - (k + 1)c_2 h\delta$$

and therefore, $f(x^k) \to -\infty$ as $k \to +\infty$ which contradicts (12). It is obvious that the upper bound for the number of iterations M necessary to find the (h, δ)-stationary point is M_0. △

Remark 1 Since $c_2 \leq c_1$, always $\sigma_k \geq h$, and therefore $h > 0$ is a lower bound for σ_k. This leads to the following rule for the estimation of σ_k. We define a sequence:

$$\theta_l = 2^l h, \quad l = 1, 2, \dots,$$

and σ_k is defined as the largest θ_l satisfying the inequality in Step 4 of Algorithm 2.

Algorithm 2 can be applied to compute stationary points of the function f that is points x where $0 \in \partial f(x)$. Let $\{h_k\}$, $\{\delta_k\}$ be sequences such that $h_k \to +0$ and $\delta_k \to +0$ as $k \to \infty$.

Algorithm 3 The secant method.

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

Step 2. If $0 \in \partial f(x^k)$, then stop.

Step 3. Apply Algorithm 2 starting from the point x^k for $h = h_k$ and $\delta = \delta_k$. This algorithm terminates after finite many iterations $M > 0$, and as a result, it computes (h_k, δ_k)-stationary point x^{k+1}.

Step 4. Set $k = k + 1$ and go to Step 2.

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

Theorem 2 *Assume that the function f is locally Lipschitz, the set $W(x, h)$ is constructed using SR-quasisecants, the condition (4) is satisfied and the set $\mathcal{L}(x^0)$ is bounded for starting points $x^0 \in \mathbb{R}^n$. Then every accumulation point of the sequence $\{x^k\}$ belongs to the set $X^0 = \{x \in \mathbb{R}^n : 0 \in \partial f(x)\}$.*

Proof: Since the function f is locally Lipschitz and the set $\mathcal{L}(x^0)$ is bounded, $f_* - \infty$. Therefore, conditions of Theorem 1 are satisfied, and Algorithm 2 generates a sequence of (h_k, δ_k)-stationary points after the finite number of points for all $k \geq 0$. Since for any $k > 0$ the point x^{k+1} is (h_k, δ_k)-stationary, it follows from the definition of the (h_k, δ_k)-stationary points that

$$\min\left\{\|v\| : v \in W(x^{k+1}, h_k)\right\} \leq \delta_k. \tag{13}$$

It is obvious that $x^k \in \mathcal{L}(x^0)$ for all $k \geq 0$. The boundedness of the set $\mathcal{L}(x^0)$ implies that the sequence $\{x^k\}$ has at least one accumulation point. Let x^* be an accumulation point and $x^{k_i} \to x^*$ as $i \to +\infty$. The inequality in (13) implies that

$$\min\left\{\|v\| : v \in W(x^{k_i}, h_{k_i})\right\} \leq \delta_{k_i - 1}. \tag{14}$$

The mapping $QSec(\cdot,\cdot)$ satisfies the condition (4), therefore, at the point x^* for any $\varepsilon > 0$ there exists $\eta > 0$ such that

$$W(y,h) \subset \partial f(x^*) + B_\varepsilon \tag{15}$$

for all $y \in B_\eta(x^*)$ and $h \in (0,\eta)$. Since the sequence $\{x^{k_i}\}$ converges to x^* there exists $i_0 > 0$ such that $x^{k_i} \in B_\eta(x^*)$ for all $i \geq i_0$. On the other hand since $\delta_k, h_k \to +0$ as $k \to +\infty$ there exists $k_0 > 0$ such that $\delta_k < \varepsilon$ and $h_k < \eta$ 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 (14) and (15) that

$$\min\{\|v\| : \ v \in \partial f(x^*)\} \leq 2\varepsilon.$$

Since $\varepsilon > 0$ is arbitrary and the mapping $x \mapsto \partial f(x)$ is upper semicontinuous, $0 \in \partial f(x^*)$. △

5 Results of numerical experiments

The efficiency of the proposed algorithm was verified by applying it to some academic test problems with nonsmooth objective functions. We consider three types of problems:

1. Problems with nonsmooth convex objective functions;

2. Problems with nonsmooth nonconvex regular objective functions;

3. Problems with nonsmooth, nonconvex and nonregular objective functions.

Test Problems 2.1-7, 2.9-12, 2.14-16, 2.18-25 from [19] and Problems 1-3 from [1] have been used in numerical experiments.

The brief description of test problems are given in Table 1, where the following notation is used:

- n - number of variables;

- n_m - the total number of functions under maximum and minimum (if a function contains maximum and minimum functions);

- f_{opt} - optimal value.

In our experiments, we use two bundle algorithms for comparisons: PMIN - a recursive quadratic programming variable metric algorithm for minimax optimization and PBUN - a proximal bundle algorithm. The description of these algorithms can be found in [20]. PMIN is applied to minimize maximum functions and PBUN is applied to solve problems with nonregular objective functions. We compute subgradients in problems with maximum objective functions, and in problems with nonregular objective functions, we approximate subgradients using the scheme from [3, 4]. In Algorithm 3 $c_1 = 0.2,\ c_2 = 0.05,\ \delta_k = 10^{-7},\ h_{k+1} = 0.5h_k,\ k \geq 1$ and $h_1 = 1$.

First we applied both PMIN and Algorithm 3 for solving problems P1-P22 using starting points from [19]. Results are presented in Table 2. In this table we present the value of the objective function at the final point (f), the number function and subgradient evaluations (n_f and n_{sub}, respectively). These results demonstrate that the bundle method performs better than the secant method. The latter method uses significantly less function and subgradient evaluations. The secant method fails to find the best known solutions for problems P9 and P10 whereas the bundle method finds those solutions for all problems.

At the next step we applied both methods to solve all problems using 20 randomly generated starting points. In order to compare the performance of the algorithms, we use two indicators: n_b - the number of successful runs considering the best known solution and n_s - the number of successful runs considering the best found solution by these two algorithms. Assume that f_{opt} and $\bar{f}$ are the values of the objective function at the best known solution and at the best found solution, respectively. Then we say that an algorithm finds the best solution with respect to a tolerance $\varepsilon > 0$ if

$$\frac{f_* - f_0}{1 + |f_*|} \leq \varepsilon,$$

where f_* is equal either to f_{opt} (for n_b) or to $\bar{f}$ (for n_s) and f_0 is the optimal value of the objective function found by an algorithm. In our experiments $\varepsilon = 10^{-5}$.

The results of numerical experiments are presented in Tables 3 and 4. In Table 3 we present f_{av} - the average objective value over 20 runs of algorithms and also numbers n_b and n_s for each method.

Results presented in Table 3 show that both the secant method and the bundle method (PMIN) are effective methods for the minimization of nonsmooth convex functions. However, the bundle method is faster and more accurate than the secant method. Both algorithm perform similarly for nonconex nonsmooth, regular functions. However, results for f_{av} show that the bundle method is more sensitive to the choice of starting points than the secant method. Overall the secant method performs better than the bundle method (PBUN) on nonconvex, nonsmooth nonregular functions.

Table 4 presents the average number of objective function (n_f) and subgradient (n_{sub}) evaluations as well as the average CPU time t (in seconds) over 20 runs of algorithms. These results demonstrate that the bundle method uses significantly less function and subgradient evaluations to minimize convex functions. Results for nonconvex regular functions show that the bundle method is sensitive to the choice of starting points. The numbers n_f and n_{sub} are large for some starting points and therefore their average values are very large for some problems (Problems P9, P10, P12, P16, P22). The secant method is much less sensitive to the choice of starting points.

6 Conclusions

In this paper we developed the secant method for minimizing nonsmooth nonconvex functions. We introduced the notion of an secant and quasisecant and demonstrated how they can be computed for some classes of nonsmooth functions. We proposed an algorithm for the computation of descent directions using quasisecants.

We presented results of numerical experiments and compared the secant method with the bundle method. The computational results show that the bundle method performs significantly better than the secant method for minimizing nonsmooth convex functions whereas the secant method outperforms the bundle method when the objective function is nonsmooth, nonconvex nonregular. Results for nonsmooth, nonconvex regular functions are mixed. These results show that the secant method is mush less sensitive to the choice of starting points than the bundle method.

Acknowledgements

Dr. Adil Bagirov is the recipient of an Australian Research Council Australian Research Fellowship (Project number: DP 0666061).

References

[1] A.M. Bagirov, A method for minimizing of quasidifferentiable functions, Optimization Methods and Software, 17(1)(2002), pp. 31-60.

[2] A.M. Bagirov, Continuous subdifferential approximations and their applications, Journal of Mathematical Sciences, 115(5)(2003), pp. 2567-2609.

[3] A.M. Bagirov, M. Ghosh and D. Webb, A derivative-free method for linearly constrained nonsmooth optimization, *Journal of Industrial and Management Optimization,* 2(3)(2006), pp. 319-338.

[4] Bagirov A.M., Karasozen B., and Sezer M., (2007) Discrete gradient method: a derivative free method for nonsmooth optimization, *Journal of Optimization Theory and Applications,* accepted for publication.

[5] A.M. Bagirov, A.M. Rubinov, N.V. Soukhoroukova and J. Yearwood, Supervised and unsupervised data classification via nonsmooth and global optimisation, TOP: Spanish Operations Research Journal, 11(1)(2003), pp. 1-93.

[6] A.M. Bagirov and J. Yearwood, A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems, European Journal of Operational Research, 170(2)(2006), pp. 578-596.

[7] J.V. Burke, A.S. Lewis and M.L. Overton, A robust gradient sampling algorithm for nonsmooth, nonconvex optimization, SIAM Journal on Optimization, 15(3)(2005), pp. 751-779.

[8] F.H. Clarke, Optimization and Nonsmooth Analysis, New York, John Wiley, 1983.

[9] V.F. Demyanov, and A.M. Rubinov, Constructive Nonsmooth Analysis, Peter Lang, Frankfurt am Main, 1995.

[10] A. Frangioni, Generalized bundle methods, SIAM Journal on Optimization, 113(1)(2002), pp. 117-156.

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

[12] M. Fukushima and L. Qi, A globally and superlinearly convergent algorithm for nonsmooth convex minimization, SIAM Journal on Optimization, 6(1996), pp. 1106-1120.

[13] M. Gaudioso and M.F. Monaco, A bundle type approach to the unconstrained minimization of convex nonsmooth functions, Mathematical Programming, 23(1982), pp. 216-226.

[14] J.B. Hiriart-Urruty and C. Lemarechal, Convex Analysis and Minimization Algorithms, Springer Verlag, Heidelberg, Vol. 1 and 2, 1993.

[15] K.C. Kiwiel, Methods of Descent for Nondifferentiable Optimization, Lecture Notes in Mathematics, Springer-Verlag, Berlin, Volume 1133, 1985.

[16] K.C. Kiwiel, Proximal control in bundle methods for convex nondifferentiable minimization, Mathematical Programming, 29(1990), pp. 105-122.

[17] C. Lemarechal and C.A. Sagastizabal, Practical aspects of the Moreau-Yosida regularization I: Theoretical preliminaries, *SIAM Journal on Optimization,* 7(1997), pp. 367-385.

[18] L. Luksan and J. Vlcek, A bundle Newton method for nonsmooth unconstrained minimization, Mathematical Programming, 83(1998), pp. 373-391.

[19] L. Luksan and J. Vlcek, Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization, Technical Report No. 78, Institute of Computer Science, Academy of Sciences of the Czech Republic, 2000.

[20] L. Luksan and J. Vlcek, Algorithm 811: NDA: Algorithms for nondifferentiable optimization, *ACM Transaction on Mathematical Software,* 27(2), 2001, pp. 193-213.

[21] M.M. Makela and P. Neittaanmaki, Nonsmooth Optimization, World Scientific, Singapore, 1992.

[22] R. Mifflin, A quasi-second-order proximal bundle algorithm, Mathematical Programming, 73(1996), pp. 51-72.

[23] E. Polak and J.O. Royset, J.O., Algorithms for finite and semi-infinite min-max-min problems using adaptive smoothing techniques, Journal of Optimization Theory and Applications, 119(3)(2003), pp. 421-457.

[24] N.Z. Shor, Minimization Methods for Non-Differentiable Functions, Springer-Verlag, Heidelberg, 1985.

[25] P.H. Wolfe, A method of conjugate subgradients of minimizing nondifferentiable convex functions, Mathematical Programming Study, 3(1975), pp. 145 - 173.

[26] P.H. Wolfe, Finding the nearest point in a polytope, Mathematical Programming, 11(2)(1976), pp. 128-149.

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

Table 1: The description of test problems

Function type	Problems	n	n_m	f_{opt}
Nonsmooth convex	P1 (Problem 2.1 [19])	2	3	1.9522245
	P2 (Problem 2.5 [19])	4	4	-44
	P3 (Problem 2.22 [19])	10	2	54.598150
	P4 (Problem 2.23 [19])	11	10	3.7034827173
Nonsmooth nonconvex regular	P5 (Problem 2.2 [19])	2	3	0
	P6 (Problem 2.3 [19])	2	2	0
	P7 (Problem 2.4 [19])	3	6	3.5997193
	P8 (Problem 2.6 [19])	4	4	-44
	P9 (Problem 2.7 [19])	3	21	0.0042021
	P10 (Problem 2.9 [19])	4	11	0.0080844
	P11 (Problem 2.10 [19])	4	20	115.70644
	P12 (Problem 2.11 [19])	4	21	0.0026360
	P13 (Problem 2.12 [19])	4	21	0.0020161
	P14 (Problem 2.14 [19])	5	21	0.0001224
	P15 (Problem 2.15 [19])	5	30	0.0223405
	P16 (Problem 2.16 [19])	6	51	0.0349049
	P17 (Problem 2.18 [19])	9	41	0.0061853
	P18 (Problem 2.19 [19])	7	5	680.63006
	P19 (Problem 2.20 [19])	10	9	24.306209
	P20 (Problem 2.21 [19])	20	18	93.90525
	P21 (Problem 2.24 [19])	20	31	0.0000000
	P22 (Problem 2.25 [19])	11	65	0.047700
Nonsmooth nonconvex nonregular	P23 (Problem 1 [1])	2	6	2
	P24 (Problem 2 [1])	2	-	0
	P25 (Problem 3 [1])	4	-	0

Table 2: Results of numerical experiments with given starting points

Prob.	Secant			Bundle		
	f	n_f	n_{sub}	f	n_f	n_{sub}
P1	1.95222	288	134	1.95222	8	8
P2	-44	436	238	-44	16	12
P3	54.60361	469	137	54.59815	88	36
P4	3.70348	509	266	3.70348	18	17
P5	0	244	164	0	8	8
P6	0	511	241	0	180	94
P7	3.59972	709	208	3.59972	15	14
P8	-44	379	285	-44	21	13
P9	0.05061	2639	881	0.00420	9	9
P10	0.01983	455	254	0.00808	12	11
P11	115.70644	432	310	115.70644	11	11
P12	0.00264	3092	1439	0.00264	113	36
P13	0.00202	1633	995	0.00202	86	35
P14	0.00012	1214	892	0.00012	8	7
P15	0.02234	777	537	0.02234	57	17
P16	0.03490	868	681	0.03490	53	22
P17	0.00619	1194	960	0.00619	107	20
P18	680.63006	431	282	680.63006	45	20
P19	24.30621	836	679	24.30621	19	14
P20	93.90525	1991	1703	93.90525	30	21
P21	0.00000	15104	11149	0.00000	20	19
P22	0.04803	3187	2880	0.04803	286	68

Table 3: Results of numerical experiments

Prob.	Secant			Bundle		
	f_{av}	n_b	n_s	f_{av}	n_b	n_s
P1	1.95222	20	20	1.95222	20	20
P2	-44	20	20	-44	20	20
P3	54.59890	20	20	54.59815	20	20
P4	3.70348	20	20	3.70348	20	20
P5	1.21000	16	17	0.90750	17	18
P6	0	20	20	0	20	20
P7	3.59972	20	20	3.59972	20	20
P8	-44	20	20	-28.14011	12	12
P9	0.04646	4	7	0.03051	9	18
P10	0.01789	0	9	0.01520	2	17
P11	115.70644	20	20	115.70644	20	20
P12	0.00264	20	20	0.00264	20	20
P13	0.00627	19	19	0.02752	14	14
P14	0.10662	3	16	0.30582	3	15
P15	0.27411	7	17	0.32527	5	10
P16	0.09631	17	19	0.30572	12	13
P17	0.09312	0	10	0.39131	2	10
P18	680.63006	20	20	680.63006	20	20
P19	24.30621	20	20	24.30621	20	20
P20	93.90525	20	20	93.90525	20	20
P21	0.00000	20	20	0.00000	20	20
P22	0.24743	0	9	0.22057	1	16
P23	2	20	20	2	20	20
P24	0	20	20	0.07607	17	17
P25	1.5	8	17	2.30008	2	8

Table 4: The number of function and subgradient evaluations and CPU time

Prob.	Secant			Bundle		
	n_f	n_{sub}	t	n_f	n_{sub}	t
P1	269	132	0.00	10	10	0.00
P2	347	207	0.00	12	11	0.00
P3	304	121	0.00	66	35	0.00
P4	544	327	0.01	65	55	0.00
P5	242	146	0.00	22	9	0.00
P6	1729	771	0.00	493	251	0.00
P7	425	181	0.00	20	16	0.00
P8	603	408	0.00	146	56	0.00
P9	608	232	0.00	1626	171	0.01
P10	1060	372	0.00	57891	4843	0.19
P11	443	280	0.00	29	15	0.00
P12	3170	1487	0.01	26081	1904	0.12
P13	1807	1091	0.01	372	173	0.01
P14	1086	775	0.00	61	26	0.00
P15	785	529	0.01	51	23	0.00
P16	856	669	0.02	4985	445	0.11
P17	2460	1791	0.10	60331	4761	1.10
P18	445	272	0.00	58	33	0.00
P19	990	801	0.01	18	15	0.00
P20	2035	1745	0.03	35	26	0.00
P21	14427	11108	0.67	160	52	0.03
P22	1171	818	0.03	66280	4974	2.97
P23	185	104	0.00	32	32	0.00
P24	228	106	0.00	22	22	0.00
P25	459	251	0.00	37	37	0.00