Iranian Journal of Numerical Analysis and Optimization
Vol. 9, No. 2, (2019), pp 123–139
DOI:10.22067/ijnao.v9i2.75991

Research Article

An extension of the quasi-Newton method for minimizing locally Lipschitz functions

Z. Akbari *

Abstract

We present a method to minimize locally Lipschitz functions. At first, a local quadratic model is developed to approximate a locally Lipschitz function. This model is constructed by using the ϵ-subdifferential. We minimize this local model and compute a search direction. It is shown that this direction is descent. We generalize the Wolfe conditions for finding an adequate step length along this direction. Next, the method is equipped with a quasi-Newton approach to update the local model and its globally convergence is proposed. Finally, the proposed algorithm is implemented in MATLAB environment on some standard nonsmooth optimization test problems and compared with some algorithms in the literature.

AMS(2010): 49J52; 90C56; 90C26.

Keywords: Quasi-Newton method; Quadratic model; Line search algorithm; Locally Lipschitz functions.

1 Introduction

In this paper, we consider the following unconstrained nonsmooth optimization problem:

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz continuous function. There exist several methods for solving this problem, for example, the subgradient method [25], the bundle methods [7,10,19,24], algorithms based on smoothing

*Corresponding author

Received 16 September 2018; revised 6 April 2019; accepted 10 April 2019

Z. Akbari
Faculty of Mathematical Sciences, University of Mazandaran, Babolsar, Iran. e-mail: z.akbari@umz.ac.ir

techniques [22], the derivative free methods [3], the bundle and quasi-Newton combining methods [13,15,17], the gradient sampling method [5,12], the limited memory gradient bundle method [11], and the trust region method [2,23].

In the most smooth optimization problems, the objective function is approximated by a quadratic model as follows:

$$Q(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{d^T B_k d}{2}, \tag{2}$$

where $\nabla f(x_k)$ and B_k are the gradient and Hessian of f at x_k; see [21]. In fact, the objective function is locally approximated at x_k. In quasi-Newton methods, the quadratic model (2) is minimized and its solution is a descent direction at x_k. Afterward a line search method is applied along this direction. This process is repeated until the optimal solution is achieved. Also, in trust region methods, the model (2) is minimized on a given region, called trust region, and a search direction is obtained. If this direction does not decrease the objective function adequately, then the trust region is updated.

In the nonsmooth optimization same as the smooth one, the objective function is approximated. Han et al. [9], presented a quadratic model to approximate the locally Lipschitz function f as follows:

$$Q(x_k + d) = f(x_k) + \phi(x_k, d) + \frac{d^T B_k d}{2}, \tag{3}$$

where $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a given iteration function and B_k is a given $n \times n$ symmetric matrix. The iteration function ϕ must have some properties to guarantee the convergence of produced minimization algorithm.

The presented iteration function in [9] is not practical. So, in this paper, we propose a practical iteration function and solve the following problem:

$$\min_{d \in \mathbb{R}^n} Q_k(x_k + d) = f(x_k) + \phi(x_k, d) + \frac{d^T B_k d}{2}. \tag{4}$$

Suppose that d_k is the solution of (4). We show that d_k is a descent direction and develop a line search method to reduce objective function along this direction.

In this paper, we propose a computable quadratic model to approximate the locally Lipschitz function and develop a minimization algorithm based on this model. In Section 2, some preliminary concepts of the nonsmooth analysis are reviewed. In Section 3, an iteration function is introduced for the locally Lipschitz objective function. After that, a descent algorithm is proposed. In Section 4, the global convergence of the presented algorithm is proved. The line search algorithm is described in Section 5. Numerical results are given in Section 6.

2 Preliminaries

In this section, we state the basic concepts and definitions of nonsmooth analysis from [4]. The Clarke generalized directional derivative of the locally Lipschitz function f at the point x in the direction d is defined by

$$f^{\circ}(x,d) := \limsup_{y\to x, t\downarrow 0} \frac{f(y+td)-f(x)}{t}.$$

The Clarke generalized gradient of f at x is the set $\partial f(x)$ defined by

$$\partial f(x) = \{\xi \in \mathbb{R}^n | f^{\circ}(x;d) \geq \xi^T d \text{ for all } d \in \mathbb{R}^n\}.$$

Each vector $\xi \in \partial f(x)$ is called a subgradient of f at x. For $\epsilon > 0$, The Clarke generalized ϵ-directional derivative of f at x in the direction d is defined by

$$f^{\circ}_{\epsilon}(x,d) := \limsup_{y\to x, t\downarrow 0} \frac{f(y+td)-f(x)+\epsilon}{t},$$

and the Goldstein ϵ-subdifferential of f at the point x is the set

$$\partial_{\epsilon} f(x) := \mathsf{cl}\ \mathsf{conv}\{\partial f(y),\ \|x-y\|_2 \leq \epsilon\},$$

where conv and cl are used to denote the convex hull and the closure, respectively. Each element $\xi \in \partial_{\epsilon} f(x)$ is called an ϵ-subgradient of the function f at x; see [4]. It can be seen that $f^{\circ}_{\epsilon}(x,d) = \sup_{\xi \in \partial_{\epsilon} f(x)} \xi^T d$. Let L be the Lipschitz constant of f in a neighborhood of x. Then, we have

$$\|v\|_2 \leq L, \quad \text{for all } v \in \partial f(x).$$

Throughout the paper, the used norm is the Euclidean norm. If f is differentiable at x, then $\nabla f(x) \in \partial f(x)$. Furthermore, if f is continuously differentiable at x, then

$$\partial f(x) = \{\nabla f(x)\}.$$

If $0 \in \partial_{\epsilon} f(x)$, then x is called as an ϵ-stationary point.

3 Generalized quasi-Newton method

In this section, we develop a computable iteration function and suppose that B_k is positive definite and has bounded norm, $m \leq \|B_k\| \leq M$. Since B_k^{-1} is positive definite, then we consider its Cholesky decomposition as follows:

$$B_k^{-1} = L^T L,$$

where L is an upper triangular matrix with real and positive diagonal entries, and L^T denotes the transpose of L. To define the iteration function, we consider the following problem:

$$v_k = \arg \min_{\xi \in \partial_\epsilon f(x_k)} \|L\xi\|^2, \tag{5}$$

where ϵ is a small enough positive scalar, and define $\phi(x_k, d) = v_k^T d$. Therefore the problem (4) is rewritten as follows:

$$\min_{d \in \mathbb{R}^n} q_k(d) = v_k^T d + \frac{d^T B_k d}{2}. \tag{6}$$

By this modification, $-B_k^{-1} v_k$ is the unique solution of (6). Let $d_k = -\frac{B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2}$ be a search direction. The line search is applied along this direction and the step length α_k is returned. The next iteration is computed as follows:

$$x_{k+1} = x_k + \alpha_k d_k.$$

At the point x_{k+1}, either B_k is updated by one of the quasi-Newton methods or is set the identity matrix for all k. Now, we are ready to express the generalized quasi-Newton algorithm for solving the problem (1):

Algorithm 1 Generalized quasi-Newton algorithm

Step 0 (Initialization): Let $B_1 = I_{n\times n}$, $\epsilon > 0$, $c_1 \in (0, 1)$, $x_1 \in \mathbb{R}^n$, $k = 1$.

Step 1 (Creating an iteration function): Set L as an upper triangular matrix of Cholesky decomposition for the matrix B_k^{-1}. Solve (5) at the point x_k and denote its solution by v_k.

Step 2 (Stopping condition): **If** $\|v_k\|_2 = 0$, then **stop**, **else** set $d_k = -\frac{B_k^{-1} v_k}{\|B_k^{-1} v_k\|}$ and compute $q_k = q_k(d_k)$.

Step 3 (Line search method): Apply the line search algorithm and find the step length $\alpha_k \in [\epsilon, 1]$ such that

$$\alpha_k = \max \left\{ \alpha \mid f(x_k + \alpha d_k) - f(x_k) \le c_1 \alpha v_k^T d_k \right\}.$$

Set $x_{k+1} = x_k + \alpha_k d_k$.

Step 4 (Updating): **If** it is necessary, **then** update B_k, set $k = k + 1$ and **go to step 1**.

As mentioned before, we can set $B_k = I$ for all k in Step 4, which means that the matrix B_k^{-1} is a fixed and identity matrix. Therefore, Algorithm 1 is converted to the steepest descent method for minimizing the locally Lipschitz function [18]. If B_k is updated by the BFGS formula (see [21]), then Algorithm 1 is a generalization of the quasi-Newton method for minimizing the locally Lipschitz continuous function. In Section 5, we describe a line search algorithm how to find a step length along the search direction such that the Wolfe conditions are satisfied.

The following lemma shows that the optimal value of (6) is nonpositive. Also it ensures that if the optimal value of (6) is nonzero, then the corresponding solution provides a descent direction for the function f at the point x_k. Else x_k is the optimal solution.

Lemma 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function and let $c_1 \in (0, 1)$. Suppose that v_k is the solution of problem (5) at x_k such that $v_k \neq 0$. If $d_k = -\frac{B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2}$, then*

(a) $v_k^T d_k \le 0$ and $q_k \le 0$.

(b) if $v_k^T d_k > 0$, then there exists a scalar $\bar{\alpha} > 0$ such that for all $\alpha \in [0, \bar{\alpha}]$, we have

$$f(x_k + \alpha d_k) - f(x_k) \le c_1 \alpha v_k^T d_k.$$

Proof. Part (a) is clear. To prove part (b), at the contrary, we suppose there exists a positive sequence $\{\alpha_i\}$ such that $\alpha_i \to 0$ and we have

$$f(x_k + \alpha_i d_k) - f(x_k) > c_1 \alpha_i v_k^T d_k.$$

Both sides of the above inequality are divided by α_i, then, with passing to the $\limsup_{i\to\infty}$, we deduce

$$f_\epsilon^\circ(x_k, d_k) \ge f^\circ(x_k, d_k) \ge \limsup_{i\to\infty} \frac{f(x_k + \alpha_i d_k) - f(x_k)}{\alpha_i} \ge c_1 v_k^T d_k.$$

Since $d_k = -\frac{B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2}$, then $v_k^T d_k = -\frac{v_k^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2} = -\frac{\|L v_k\|_2^2}{\|B_k^{-1} v_k\|_2}$. Thus we have

$$f_\epsilon^\circ(x_k, d_k) \ge -c_1 \frac{\|L v_k\|^2}{\|B_k^{-1} v_k\|_2}. \tag{7}$$

On the other hand, we have

$$\begin{aligned} f_\epsilon^\circ(x_k, d_k) &= \max_{\xi \in \partial_\epsilon f(x_k)} \xi^T d_k = \max_{\xi \in \partial_\epsilon f(x_k)} \frac{-\xi^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2}, \\ &= \frac{-1}{\|B_k^{-1} v_k\|_2} \min_{\xi \in \partial_\epsilon f(x_k)} \xi^T L^T L v_k = \frac{-1}{\|B_k^{-1} v_k\|_2} \min_{\xi \in \partial_\epsilon f(x_k)} (L\xi)^T L v_k. \end{aligned}$$

According to (5), we have $\|Lv_k\|^2 \leq \|L\xi\|^2$, for all $\xi \in \partial_\epsilon f(x)$, and since the set $\partial_\epsilon f(x_k)$ is a convex compact set, referring to [20, Lemma 5.2.6.], reader can easily find out

$$f^\circ_\epsilon(x_k, d_k) = -\frac{\|Lv_k\|_2^2}{\|B_k^{-1}v_k\|_2}. \tag{8}$$

By (7) and (8), we have

$$-\frac{\|Lv_k\|_2^2}{\|B_k^{-1}v_k\|_2} = f^\circ_\epsilon(x_k, d_k \geq -c_1\frac{\|Lv_k\|_2^2}{\|B_k^{-1}v_k\|_2}.$$

Since $c_1 \in (0,1)$, the above equation is false. Thus, part (b) is proved. □

The following lemma shows that the step length $\alpha_k = \epsilon$ is satisfied the Arimijo condition along d_k for all k.

Lemma 2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function and let $c_1 \in (0,1)$. Then the step length ϵ satisfies the Armijo condition along d_k at x_k.*

Proof. Since f is a locally Lipschitz continuous function, then according to the mean-value theorem in [20], there exists $\eta \in (0,1)$ such that $z = x_k + \eta\epsilon d_k$ and

$$f(x_k + \epsilon d_k) - f(x_k) \in \partial f(z)^T(\epsilon d_k).$$

So, there exists $u \in \partial f(z)$ such that

$$f(x_k + \epsilon d_k) - f(x_k) = \epsilon u^T d_k = -\frac{\epsilon u^T B_k^{-1} v_k}{\|B_k^{-1}v_k\|_2} = -\frac{\epsilon (Lu)^T (Lv_k)}{\|B_k^{-1}v_k\|_2}. \tag{9}$$

Since $v_k \in \partial_\epsilon f(x_k)$ is the solution of problem (5), and $\partial_\epsilon f(x_k)$ is a convex compact set, then by [20, Lemma 5.2.6], we have

$$(Lu)^T(Lv_k) \geq \|Lv_k\|_2^2. \tag{10}$$

So, by (9) and (10), we have

$$f(x_k + \epsilon d_k) - f(x_k) \leq -\frac{\epsilon\|Lv_k\|_2^2}{\|B_k^{-1}v_k\|_2} = -\frac{\epsilon v_k^T B_k^{-1} v_k}{\|B_k^{-1}v_k\|_2} = \epsilon v_k^T d_k.$$

Since $v_k^T d_k < 0$ and $c_1 \in (0,1)$, then

$$f(x_k + \epsilon d_k) - f(x_k) \leq c_1 \epsilon v_k^T d_k.$$

Thus the proof is completed. □

4 Optimality condition and global convergence

In this section, we provide the necessary optimality condition and prove the global convergence of Algorithm 1. If $\|v_k\|_2 = 0$ at some iteration k, then $0 \in \partial_\epsilon f(x_k)$. This shows that x_k is an ϵ-stationary point. Now, we show that if Algorithm 1 dose not terminate after finite many iterations, then each accumulation point of the sequence $\{x_k\}$ is an ϵ-stationary point of f. To prove this, we first prove the following lemma.

Lemma 3. *Suppose that the level set $\mathcal{L} := \{x : f(x) \leq f(x_1)\}$ is bounded, and there exist positive constants M and m such that $m\|d\|_2^2 \leq d^T B_k d \leq M\|d\|_2^2$, for all k and $d \in \mathbb{R}^n$. If Algorithm 1 does not terminate after finite many iterations, then*

$$\lim_{k\to\infty} \|v_k\| = 0.$$

Proof. Suppose that Algorithm 1 does not terminate after finite many iterations. Since $\|v_k\| \neq 0$ for each k, then by Lemma 1, we have

$$f(x_k + \alpha_k d_k) - f(x_k) \leq c_1 \alpha_k v_k^T d_k. \tag{11}$$

Since $\{f(x_k)\}$ is a strictly decreasing and bounded below sequence, then it converges and we have

$$f(x_k + \alpha_k d_k) - f(x_k) \to 0.$$

So by (11), we have

$$0 \leq c_1 \lim_{k\to\infty} \alpha_k v_k^T d_k.$$

By Lemma 1, we know $v_k^T d_k = -\frac{v_k^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2} \leq 0$, so

$$0 \leq c_1 \lim_{k\to\infty} \alpha_k v_k^T d_k = -c_1 \lim_{k\to\infty} \frac{\alpha_k v_k^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2} \leq 0.$$

On the other hand, by Lemma 2, $\alpha_k \geq \epsilon$ Thus

$$\lim_{k\to\infty} \frac{v_k^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2} = 0. \tag{12}$$

Let $0 < \lambda_1^k \leq \lambda_2^k \leq \cdots \leq \lambda_n^k$ be the eigenvalues of B_k^{-1}. Since B_k^{-1} is a positive definite matrix, then we have $\lambda_1^k \geq m$ and $\lambda_n^k \leq M$. Therefore

$$\frac{m}{M}\|v_k\|_2 \leq \frac{\lambda_1^k}{\lambda_n^k}\|v_k\|_2 \leq \frac{v_k^T B_k^{-1} v_k}{\|B_k^{-1} v_k\|_2} \leq \frac{\lambda_n^k}{\lambda_1^k}\|v_k\| \leq \frac{M}{m}\|v_k\|_2. \tag{13}$$

Then, by (12) and (13), we conclude $\|v_k\|_2 \to 0$. □

In the following theorem, we show that each accumulation point of the sequence $\{x_k\}$ is an ϵ-stationary point.

Theorem 1. *Suppose that the level set $\mathcal{L} := \{x : f(x) \leq f(x_1)\}$ is bounded and that $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz function on $\mathcal{L}$. If x^* is an accumulation point of the sequence $\{x_k\}$, then x^* is an ϵ-stationary point of f.*

Proof. Since $\{x_k\} \subset \mathcal{L}$, then the sequence $\{x_k\}$ is bounded and it has a convergent subsequence. Let $\{x_{k_i}\}$ be a subsequence of the sequence $\{x_k\}$ convergent to x^*. We have $v_{k_i} \in \partial_\epsilon f(x_{k_i})$ and by Lemma 3, $v_{k_i} \to 0$. On the other hand, $\partial_\epsilon f(\cdot)$ is upper semicontinuous. Thus $0 \in \partial_\epsilon f(x^*)$ and this shows that x^* is an ϵ-stationary point of f. □

5 Line search algorithm

In this section, we present a line search algorithm to update the matrix B_k by the BFGS method. To do this, we need two subgradients $\xi_k \in \partial f(x_k)$ and $\xi_{k+1} \in \partial f(x_{k+1})$, such that the curvature condition is satisfied, that is,

$$\xi_{k+1}^T d_k \geq c_2 \xi_k^T d_k,$$

where $c_2 \in (c_1, 1)$. To find ξ_{k+1}, we use the line search strategy same as [21,26]. The step length α must satisfy the Wolfe conditions at x_k along d_k. The Wolfe conditions are the Armijo and curvature conditions, as

$$\begin{aligned} & f(x_k + \alpha d_k) - f(x_k) \leq c_1 \alpha f_\epsilon^\circ(x_k, d_k), \ \text{(Armijo condition)} \\ & \exists \xi_{k+1} \in \partial f(x_k + \alpha d_k), \ \text{s.t. } \xi_{k+1}^T d_k \geq c_2 f_\epsilon^\circ(x_k, d_k). \ \text{(curvature condition)} \end{aligned}$$

In this paper, we present an algorithm to find a step length, which satisfies the Armijo condition. Now we present the following algorithm. This algorithm finds an interval including a step length satisfies the Wolfe conditions at the point x_k along d_k. Algorithm 2 is started with $\alpha_0 = \epsilon$, because the step length ϵ satisfies the Armijo condition. The following proposition shows that Algorithm 2 terminates after finite many iterations.

Proposition 1. *If $f_\epsilon^\circ(x_k, d_k) < 0$ and f is bounded below at x_k along d_k, then Algorithm 2 terminates after finite many iterations.*

Proof. Since $\phi(\alpha) = f(x_k + \alpha d_k)$ is bounded below and the line $l(\alpha) = f(x_k) + c_1 \alpha f_\epsilon^\circ(x_k, d_k)$ is unbounded below, then there exists $\overline{\alpha}$ such that $\phi(\alpha) > l(\alpha)$ for all $\alpha > \overline{\alpha}$. Thus Algorithm 2 terminates when $\alpha_1 > \overline{\alpha}$. □

Either Algorithm 2 finds a step length that it satisfies the Wolfe conditions or it returns an interval. We prove that there exists a step length in this interval, which satisfies the Wolfe conditions.

Algorithm 2 Line search algorithm

```
set $\alpha_1 = 1, \alpha_0 = \epsilon$;
repeat
    if $f(x_k + \alpha_1 d_k) - f(x_k) > c_1 \alpha_1 f^\circ_\epsilon(x_k, d_k)$ then
        $\alpha^*$=wolfe($\alpha_0, \alpha_1$)
        return $\alpha^*$
    else if $\exists \xi \in \partial f(x_k + \alpha_1 d_k)$ s.t. $\xi^T d_k \geq c_2 f^\circ_\epsilon(x_k, d_k)$ then
        $\alpha^* = \alpha_1$
        return $\alpha^*$
    end if
    $\alpha_0 = \alpha_1$;
    $\alpha_1 = 2\alpha_1$
until
```

Proposition 2. *In Algorithm 2, suppose that $f(x_k + \alpha_1 d_k) - f(x_k) > c_1 \alpha_1 f^\circ_\epsilon(x_k, d_k)$. Then the interval (α_0, α_1) contains step lengths satisfying the Wolfe conditions.*

Proof. Let $\psi(\alpha) = f(x_k + \alpha d_k) - f(x_k) - c_2 \alpha f^\circ_\epsilon(x_k, d_k)$. It is obvious that $\psi(\alpha_0) \leq \psi(\alpha_1)$. Since the curvature condition is not satisfied at α_0, we have

$$\xi^T d_k < c_2 f^\circ f(x_k, d_k), \ \forall \xi \in \partial f(x_k + \alpha_0 d_k).$$

Thus $0 \notin \partial\phi(\alpha_0)$. Therefore α_0 is not the local minimizer of ψ on $[\alpha_0, \alpha_1]$. So the minimum point of ψ must be in (α_0, α_1). Suppose that $\alpha^* \in (\alpha_0, \alpha_1)$ is the minimizer of ψ. Thus $0 \in \partial\phi(\alpha^*)$. Since

$$\partial\phi(\alpha^*) \subset \partial f(x_k + \alpha^* d_k)^T d_k + c_2 f^\circ_\epsilon(x_k, d_k),$$

there exists $\xi \in \partial f(x_k + \alpha^* d_k)$ such that $\xi^T d_k - c_2 f^\circ_\epsilon(x_k, d_k) = 0$. On the other hand, $\psi(\alpha_0) \geq \psi(\alpha^*)$ and α_0 satisfies the Armijo condition. Thus α^* satisfies the Armijo condition. Therefore, α^* is satisfied in the Wolfe conditions. □

Now we present an algorithm to find a step length, which satisfies the Wolfe conditions on $[\alpha_0, \alpha_1]$. The following proposition indicates the convergence of Algorithm 3.

Proposition 3. *If Algorithm 3 does not terminate after finite many iterations, then it converges to α^* such that $0 \in \partial f(x_k + \alpha^* d_k)^T d_k - c_2 f^\circ_\epsilon(x_k, d_k)$.*

Proof. Let ψ be the function defined in Proposition 2. Since $\psi(a_i) < \psi(b_i)$ and $0 \notin \partial f(x_k + a_i d_k)^T d_k - c_2 f^\circ_\epsilon(x_k, d_k)$, then $0 \notin \partial\psi(a_i)$. Thus, ψ takes its minimum on (a_i, b_i). Suppose r_i is the minimum of ψ on (a_i, b_i). So, we have $0 \in \partial\psi(r_i)$. On the other hand, $\{a_i\}$ and $\{b_i\}$ are monotone and bounded sequences. Therefore, these sequences are convergent. Also, we have

Algorithm 3 Wolfe Algorithm

```
$i = 1$
$a_i = \alpha_1$
$b_i = \alpha_0$
$t_i = \frac{a_i+b_i}{2}$
repeat
  if $f(x_k + t_i d_k) - f(x_k) \geq c_1 t_i f_\epsilon^\circ(x_k, d_k)$ then
    $a_{i+1} = t_i$ and $b_{i+1} = b_i$
  else if $\exists \xi \in \partial f(x_k + t_i d_k)$ such that $\xi^T d_k \geq c_2 f_\epsilon^\circ(x_k, d_k)$ then
    return $t_i$
  else
    $b_{i+1} = t_i$ and $a_{i+1} = a_i$.
  end if
  $t_i = \frac{a_i+b_i}{2}$
  $i = i + 1$
until
```

$$\lim_{i\to\infty} a_i - b_i = 0,$$

thus

$$\lim_{i\to\infty} a_i = \lim_{i\to\infty} b_i = \lim_{i\to\infty} r_i = \alpha^*.$$

Since $\partial\psi(\cdot)$ is upper semicontinuous and $0 \in \partial\psi(r_i)$, then $0 \in \partial\psi(\alpha^*)$. This shows that $0 \in \partial f(x_k + \alpha^* d_k)^T d_k - c_2 f_\epsilon^\circ(x_k, d_k)$. □

Now, we go back to problem (5). Solving this problem is impractical, because the structural of the set $\partial_\epsilon f(x)$ is unclear. So the set $\partial_\epsilon f(x)$ is approximated and problem (5) is approximately solved. In [1], the generalized h-increasing point algorithm is proposed for computing an approximation solution of problem (5). Here, we summarize the generalized h-increasing point algorithm. Let W_l be a finite subset of $\partial_\epsilon f(x_k)$. The convex hull of W_l is considered as an approximation of $\partial_\epsilon f(x_k)$. We solve the following problem instead of (5):

$$\min_{v \in \mathsf{conv} W_k} v^T B_k v,$$

and consider w_k as its solution. Let $g_k = -\frac{w_k}{\|w_k\|}$. If the inequality

$$f(x_k + \epsilon g_k) - f(x_k) \leq -c_1 \epsilon \|w_k\|, \tag{14}$$

is satisfied, then $\mathsf{conv} W_k$ is an acceptable approximation of $\partial_\epsilon f(x_k)$. In this case, we consider g_k and $-\|w_k\|$ as an approximation of d_k and $f_\epsilon^\circ(x_k, d_k)$. Else a new element from $\partial_\epsilon f(x_k)$, such as v_l, must be added into W_k such that $v_l \notin \mathsf{conv} W_k$. To find such an element, we apply the generalized h-increasing point algorithm in [1]. In [1], it is proved that this procedure is terminated after finite many iterations. Either the direction g_k is returned such that (14) is satisfied or we have

$$\|w_k\| \leq \delta, \tag{15}$$

where δ is a predefined threshold. If (14) is satisfied, then the Armijo condition is satisfied along direction g_k at x_k with step length ϵ. Also when (15) holds, we can assume that x_k is an ϵ-stationary point. Finding such a direction is given in Algorithm 3.1 in [1].

6 Numerical experiments

In the numerical experiments, as computing d_k by (5) is impractical, we use Algorithm 3.1 in [1] and approximate it with g_k. In this case, $f_\epsilon^\circ(x_k, d_k)$ is approximated with $-\|w_k\|$. We set parameters as follows: $\epsilon = 10^{-6}$, $\delta = 10^{-6}$, $c_1 = 10^{-4}$, and $c_2 = 0.9$. In Algorithms 2 and 3, the curvature condition is checked just by one subgradient from $\partial_\epsilon f(x_k + td_k)$, say ξ, that is, the following condition is checked:

$$\xi^T d_k \geq c_2 f_\epsilon^\circ(x_k, d_k).$$

In this section, we used some test problems in [8, 14] and report the computational and numerical results of Algorithm 1 denoted by ”NQSN”. Also Algorithm 1 is compared with some algorithms existing in the literatures. Some of the selected algorithms are implemented in MATLAB environment and other ones in Fortran. The measurement of algorithm efficiency is the number of function evaluations. To have a better comparison of the implemented algorithms, the performance profile of Dolan and More in [6] is used.

Two classes of test functions are applied. The first class is taken from [8] and the second one is the TEST29 in [14]. The test problems are introduced in Table 1. We compare the smooth BFGS method [21], the variable metric bundle method (PVAR) [16,17], a method presented in [18] (MY), the limited-memory BFGS method (LBFGS) [21], the limited memory bundle method (LMBM) [8], and the gradient sampling method (GS) with the presented method.

Each algorithm is terminated after 10000 iterations. In the performance profile, an algorithm solves a problem successfully when

$$\frac{|f_{\min} - f^*|}{1 + |f^*|} \leq thr,$$

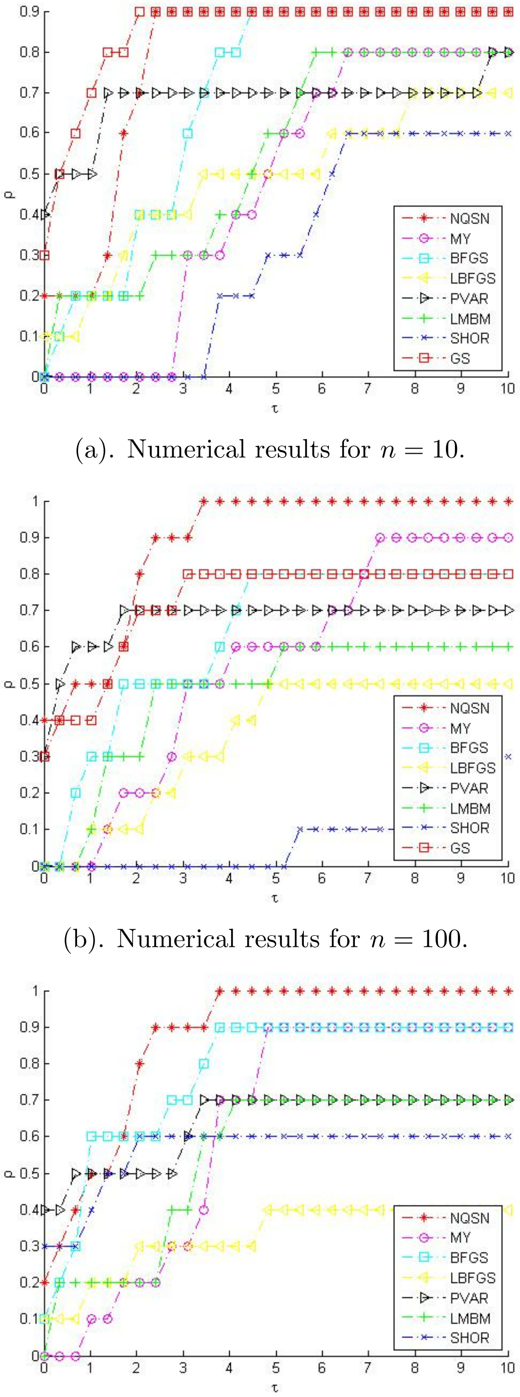
where $thr \in (0, 1)$ is a predefined threshold and $f_{\min}$ and f^* are the optimal value and achieved optimal value, respectively. In this paper, we set thr equal to 10^{-4}.

Table 1: Test problems and their optimal value for $n = 1000$.

First class of problems			Second class of problems		
No.	problem	optimal value	No.	problem	optimal value
1	MAXQ	0	11	problem 2 from TEST29	0
2	MAXHILB	0	12	problem 5 from TEST29	0
3	LQ	-1.412799e+003	13	problem 6 from TEST29	0
4	CB3I	1998	14	problem 11 from TEST29	1.203128e+004
5	CB3II	1998	15	problem 13 from TEST29	5.661313e+002
6	NACTFACES	0	16	problem 17from TEST29	0
7	Brown 2	0	17	problem 19 from TEST29	0
8	Mifflin 2	-7.065034e+002	18	problem 20 from TEST29	0
9	Crescent I	0	19	problem 22 from TEST29	0
10	Crescent II	0	20	problem 24 from TEST29	0

Let n be the size of the test problem. The algorithms are tested with 3 sizes of test problems, the small size $n = 10$, middle size $n = 100$, and large scale $n = 1000$. Since the GS algorithm is very time consuming for large scale problems, then we do not run this algorithm for large scale problems. In Figures 1 and 2, we report the results for the first and second class of test problems as the performance profile, respectively.

Figure 1 shows that the first class of the test problems is simpler than the second one. The proposed algorithm is more efficient than other algorithms for middle and large scale problems and is the only algorithm can solve all these problems. Also, the number of function evaluations of NQSN method is less than other methods. In the second class of test problems, for small size problems, the GS method is more efficient than other ones. But the BFGS and NQSN have similar results. But in the middle size, the MY method can just solve all problems. Performance profiles show that the NQSN method solves problems with significantly less number of function evaluations. For large scale test problems, just MY and NQSN can solve 6 problems. While the number of function evaluations in the NQSN method is very less than the number of function evaluations in MY method. The results show that the NQSN method is more efficient than other algorithms for large scale problems. Also, this algorithm solves problems with less number of function evaluations.

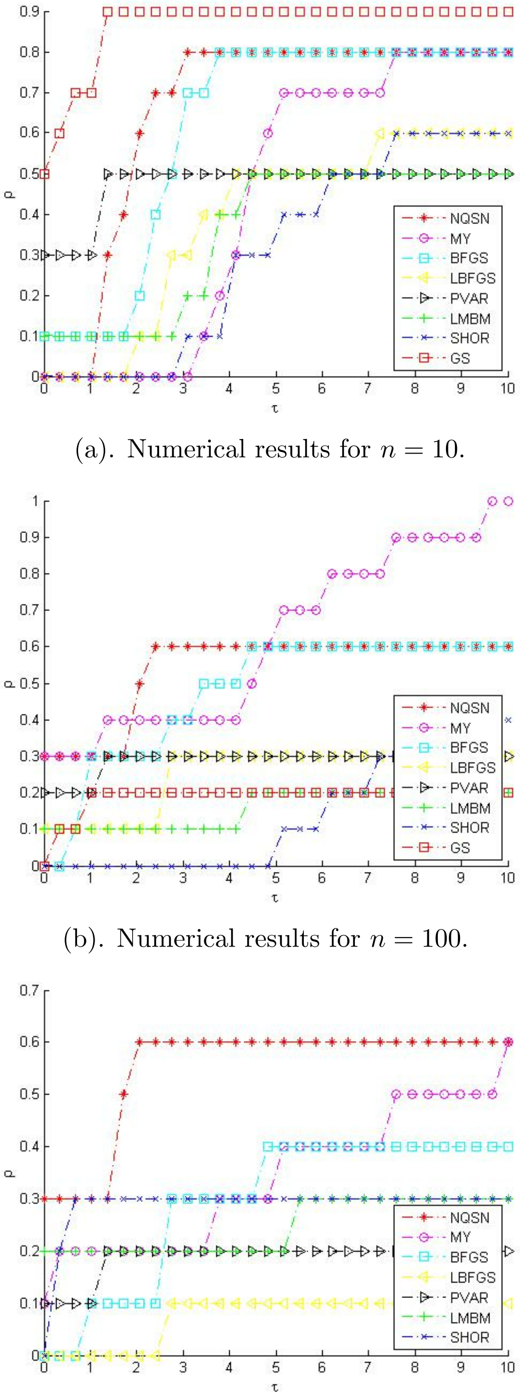


(a). Numerical results for $n = 10$.

(b). Numerical results for $n = 100$.

(c). Numerical results for $n = 1000$.

Figure 1: The performance profiles of the first class of test problems



(a). Numerical results for $n = 10$.

(b). Numerical results for $n = 100$.

(c). Numerical results for $n = 1000$.

Figure 2: The performance profiles of the second class of test problems

7 Conclusion

In this paper, the generalized quasi-Newton algorithm is presented for minimizing locally Lipschitz functions. Same as the iterative optimization methods, first we introduced a descent direction for the locally Lipschitz function in each iteration. Then, we computed an adequate step length along this direction satisfies the generalized Wolfe conditions. After that, we showed that the generalized quasi-Newton algorithm is convergent. The presented algorithm was applied to some standard test functions and compared with the MY method, the smooth BFGS method, the gradient sampling method, the limited-memory BFGS method and the limited memory bundle method. In future work, we will develop this algorithm for minimizing locally Lipschitz functions with nonlinear constraints.

Acknowledgements

Authors are grateful to their anonymous referees and editor for their constructive comments.

References

1. Akbari, Z. and Yousefpour, R. *A trust region method for solving linearly constrained locally Lipschitz optimization problems*, Optimization, 66(9) (2017), 1519–1529.

2. Akbari, Z., Yousefpour, R. and Peyghami, M.R.*A new Nonsmooth trust region Algorithm for locally Lipschitz unconstrained optimization problems*, J. Optim. Theory Appl. 164(3) (2015), 733–754.

3. Bagirov, A.M. *Continuous subdifferential approximations and their applications*, Optimization and related topics, 2. J. Math. Sci. (N.Y.) 115 (2003), 2567–2609.

4. Bagirov, A.M., Karmitsa, N. and Mäkelä, M. M. *Introduction to nonsmooth optimization: Theory, practice and software*, Springer Publishing Company, Incorporated, 2014.

5. Burke, J.V. and Overton, M.L. *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM J. Optim. 15 (2005), 571–779.

6. Dolan, E.D. and More, J.J. *Benchmarking optimization software with performance profiles*, Math. Program. 91 (2002), 201–213.

7. Frangioni, A. *Generalized bundle methods*, SIAM J. Optim. 113 (2003), 117–156.

8. Haarala, N. and Miettinen, K. and Mäkelä, M.M. *Globally convergent limited memory bundle method for large-scale nonsmooth optimization*, Math. Program. 109 (2007), 181–205.

9. Han, S., Pang, J. and Rangaraj, N. *Globally Convergent Newton Methods for Nonsmooth Equations*, Math. Oper. Res.,17(3) (1992), 586–607.

10. Hiriart-Urruty, J-B. and Lemaréchal, C. *Convex analysis and minimization algorithms II: Advanced theory and bundle methods*, Springer-Verlag, 1993.

11. Karmitsa, N. and Bagirov, A.M. *Limited memory discrete gradient bundle method for nonsmooth derivative-free optimization*, Optimization, 61(12) (2012), 1491-1509.

12. Kiwiel, K.C. *Convergence of the Gradient Sampling Algorithm for Nonsmooth Nonconvex Optimization*, SIAM J. Optim. 18(2) (2007), 379–388.

13. Lewis, A.S. and Overton, M.L. *Nonsmooth optimization via quasi-Newton methods*, Math. Program. 141(1-2) (2012) Ser. A, 135–163. .

14. Luksan, L. , Tuma, M. , Siska, M., Vlcek, J. and Ramesova, N. *UFO 2002: interactive system for universal functional optimization*, Academy of Sciences of the Czech Republic, (2012). Also available as http://www.cs.cas.cz/luksan/ufo.pdf.

15. Luksan, L. and Vlcek, J. *A bundle-Newton method for nonsmooth unconstrained minimization*, Math. Program. 83 (1998), 373–391.

16. Luksan, L. and Vlcek, J. *Globally convergent variable metric method for convex nonsmooth unconstrained minimization*, J. Optim. Theory Appl. 102 (1999), 593–613.

17. Luksan, L. and Vlcek, J. *Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization*, J. Optim. Theory Appl., 111 (2001), 407–430.

18. Mahdavi-Amiri, M. and Yousefpour, R. *An effective nonsmooth optimization algorithm for locally Lipschitz functions*, J. Optim. Theory Appl. 155 (2012), 180–195.

19. Mäkelä, M. M. *Survey of bundle methods for nonsmooth optimization*, Optim. Methods Softw. 17 (1) (2002), 1–29.

20. Mäkelä, M.M. and Neittaanmaki, P. *Nonsmooth optimization: Theory and algorithms with applications to optimal control*, World Scientific, 1992.

21. Nocedal, J. and Wright, S.J. *Numerical Optimization*, Springer, 1999.

22. Polak, E. and Royset, J.O. *Algorithms for finite and semi-infinite min-max-min problems using adaptive smoothing techniques*, J. Optim. Theory Appl. 119(421) (2003), 421–457.

23. Qi, L. and Sun, J. *A trust region algorithm for minimization of locally Lipschitzian functions*, Math. Program. 66 (1994), 25–43.

24. Schramm, H. and Zowe, J. *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM J. Optim. 2 (1992), 121–152.

25. Shor, N. Z. *Minimization methods for non-differentiable functions*, Springer-Verlag, 1985.

26. Yousefpour, R. *Combination of steepest descent and BFGS methods for nonconvex nonsmooth optimization*, Numer. Algorithms, 72(1) (2016), 57–90.