Numer Algor (2016) 72:57–90
DOI 10.1007/s11075-015-0034-2

ORIGINAL PAPER

Combination of steepest descent and BFGS methods for nonconvex nonsmooth optimization

Rohollah Yousefpour[1]

Received: 7 August 2014 / Accepted: 3 August 2015 / Published online: 19 August 2015

Abstract In this paper, a method is developed for solving nonsmooth nonconvex minimization problems. This method extends the classical BFGS framework. First, we generalize the Wolfe conditions for locally Lipschitz functions and prove that this generalization is well defined. Then, a line search algorithm is presented to find a step length satisfying the generalized Wolfe conditions. Next, the Goldstein ε-subgradient is approximated by an iterative method and a descent direction is computed using a positive definite matrix. This matrix is updated using the BFGS method. Finally, a minimization algorithm based on the BFGS method is described. The algorithm is implemented in MATLAB and numerical results using it are reported.

Keywords Lipschitz functions · Wolfe conditions · Nonsmooth line search method · Nonsmooth BFGS method

Mathematics Subject Classification (2010) 49J52 · 90C26

1 Introduction

Nonsmooth unconstrained optimization problems arise in many applications including control theory, discrete minimax problems, complementarity problems and image denoising. Several methods for solving nonsmooth nonconvex optimization problems have been developed based on the Clarke subdifferential. The subgradient-type methods are the simplest methods for solving convex optimization problems [1]. The

✉ Rohollah Yousefpour
yousefpour@umz.ac.ir

[1] Department of Mathematical Sciences, University of Mazandaran, Babolsar, Iran

bundle-type methods are developed for convex and nonconvex optimization problems [2–10]. The algorithms based on smoothing techniques are presented in [11, 12]. The discrete gradient (DG) algorithm is considered as a derivative free method [13, 14]. In the most recent works in [15, 16], the gradient sampling (GS) algorithm is proposed to solve nonconvex optimization problems.

In methods in which only first order derivative information is employed, the BFGS method is one of the most efficient methods to solve smooth optimization problems. However, in the literature, few modifications of quasi-Newton methods have been developed to solve the nonsmooth nonconvex problems and we review some of them. In this paper, a new nonsmooth version of BFGS algorithm is developed for minimizing locally Lipschitz functions.

Since the Moreau-Yosida regularization of a convex function is differentiable [6], the a Quasi-Newton method can be applied to the Moreau-Yosida regularization. In fact, a Quasi-Newton method is combined with the bundle method [6, 8]. Under second-order smoothness assumption, these algorithms converge superlinearly [22]. However this assumption is strong and quite restrictive. Another drawback of this class is that the derivative of the regularized function is computed by convex nonsmooth minimization. More specifically, at each iteration of the Quasi-Newton method, the bundle method is applied for approximating the gradient of the regularized function, which is time consuming for the large scale functions.

In [28], the Quasi-Newton method is combined with the bundle method. The search direction is computed by the aggregated subgradients and by this search direction, the inverse of Hessian matrix approximation is updated by the BFGS or SR1 method. This class of algorithms was improved for large scale problems in [29, 30] by using the BFGS limited memory method. Recently, the behavior of the BFGS method on nonsmooth functions was studied without any modification with exact and inexact line search [31–33]. The numerical experiments show that the BFGS method with inexact line search enjoys good behavior for some nonsmooth functions.

When the classical BFGS method is used to minimize a nonsmooth function, the search direction is selected from the Clarke generalized gradient set. So, there is not any guarantee that this direction is descent. In [28] and [29, 30], the search direction is computed by using 3 subgradients and this does not guarantee that the computed direction is descent. Thus, instead of the smooth BFGS method, the search direction in the genralized nonsmooth BFGS may be not descent. So this is the main reason that these methods have poor performance in some nonconvex minimization problems. For increasing the performance of a generalized BFGS method for nonconvex minimization problems, the search direction is computed such that it is descent. In this paper, we propose a minimization algorithm where a descent direction is computed using Goldstein ϵ-subgradients and a positive definite matrix. Using an idea similar to those from [13, 36], an algorithm is developed to iteratively approximate Goldstein ϵ-subgradients. This procedure computes a descent direction after finite many iterations. This is the first step for generalized the BFGS method for nonconvex functions. Instead of other generaliztion, the developed algorithm in this paper computes a descent direction.

In the second step, a line search algorithm must be applied along the computed direction such that the Wolfe conditions are satisfied. In this paper, the Wolfe conditions are generalized based on the Goldstein ε-subgradient and it will be proven that there exist step lengths satisfying this generalization for each descent direction. We modify the smooth line search algorithm [38] for finding a step length satisfying the generalized Wolfe conditions. The modified line search algorithm also returns two subgradients for updating the approximation of Hessian matrix by the BFGS method. The proposed algorithm is implemented in MATLAB and the results are compared with those obtained using other methods.

In Section 2 some preliminaries are provided. In Section 3, the generalized Wolfe conditions and a line search algorithm are presented. A procedure for computing a descent direction is discussed in Section 4. Next, based on BFGS algorithm, a minimization algorithm is presented. In Section 5, the numerical results are reported.

2 Preliminaries

In this section, some preliminaries are given which will be used throughout the paper.

2.1 Notations

In this paper, $\mathbb{R}^n$ is the n-dimensional Euclidian Space. We use $B(x, r)$ as the open ball around x with radius r. The inner product is denoted by $\langle ., . \rangle$. $\|x\|$ is the norm of vector x and defined by $\sqrt{\langle x, x \rangle}$ and for the positive definite matrix H, we define $\|x\|_H = \langle Hx, x \rangle$. Since H is a positive definite matrix, then all of its eigenvalues are positive [46]. Let λ_n and λ_1 be the largest and smallest eigenvalue of H. Then, we have [46]

$$\lambda_1 \|x\|^2 \leq \|x\|_H \leq \lambda_n \|x\|^2 . \tag{1}$$

$\mathsf{conv}(A)$ is convex hull of a set A.

2.2 Nonsmooth analysis

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function and L be the Lipschitz constant in some neighborhood of x. The Clarke generalized directional derivative of f at x in the direction of v, denoted by $f^\circ(x, v)$, is defined by

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

and based on this generalization, in [39], the Clarke generalized subdifferential is defined as follows:

$$\partial f(x) := \left\{ \xi \in \mathbb{R}^n \ : \ f^\circ(x, v) \geq \ \langle \xi, v \rangle, \ \forall v \in \mathbb{R}^n \right\}.$$

We have [39]

$$\|\xi\| \leq L, \quad \forall \xi \in \partial f(x), \tag{2}$$

and

$$f^{\circ}(x,g) = \sup_{v\in\partial f(x)} \langle g, v\rangle. \tag{3}$$

If f is differentiable at x, then $\nabla f(x) \in \partial f(x)$ [39]. Furthermore, if F is continuously differentiable at x, then we have $\partial F(x) = \{\nabla F(x)\}$. By Rademacher's Theorem [39], a Lipschitz function is differentiable almost everywhere and, thus the gradient exists almost everywhere. If x is a minimal point of f, then $0 \in \partial f(x)$. For each $\varepsilon > 0$, in [35], the Goldstein ε-subdifferential is defined by:

$$\partial_{\varepsilon} f(x) := \mathrm{conv}\,\{\xi \ : \ \xi \in \partial f(y), \ \ y \in B(x,\varepsilon)\}.$$

If $0 \in \partial_{\varepsilon} f(x)$, then x is said to be an ε-stationary point. Moreover, let $f_{\varepsilon}^{\circ}(x,g) = \sup_{v\in\partial_{\varepsilon} f(x)}\langle g, v\rangle$ and thus we have $f^{\circ}(x,g) \le f_{\varepsilon}^{\circ}(x,g)$. If $f^{\circ}(x,g) < 0$, then g is a descent direction, i.e., there exists $\alpha > 0$, such that

$$f(x+tg) - f(x) < 0, \quad \forall t \in (0,\alpha).$$

Suppose that $f_{\varepsilon}^{\circ}(x,g) < 0$ for some $\varepsilon > 0$ and $\|g\| \le 1$. Then, by the Lebourg's Mean Value Theorem [39], for all $t \in (0,\varepsilon]$ there exist $\theta \in (0,1)$ and $\xi \in \partial f(x+t\theta g)$ such that

$$f(x+tg) - f(x) = t\langle \xi, g\rangle \le t f_{\varepsilon}^{\circ}(x,g). \tag{4}$$

This inequality shows that, when $f_{\varepsilon}^{\circ}(x,g)$ is positive or slightly negative, then it cannot be guaranteed to reduce f along g. If $0 \notin \partial_{\varepsilon} f(x)$, then we can find a descent direction. Now, consider the following problem

$$\min_{\|g\|\le 1} f_{\varepsilon}^{\circ}(x,g) = \min_{\|g\|\le 1}\max_{\xi\in\partial_{\varepsilon} f(x)} \langle \xi, g\rangle. \tag{5}$$

This problem has a solution, which can be computed by solving the following problem [7],

$$\min_{\xi\in\partial_{\varepsilon} f(x)} \|\xi\|. \tag{6}$$

If ξ_0 is the solution of (6), then $g = -\frac{\xi_0}{\|\xi_0\|}$ is the solution of (5) and $f_{\varepsilon}^{\circ}(x,g) = -\|\xi_0\|$. Let H be a positive definite matrix. In (6), we replace $\|x\|$ with $\|x\|_H$ and thus problem (6) is reduced to:

$$\min_{\xi\in\partial_{\varepsilon} f(x)} \|\xi\|_H. \tag{7}$$

The following proposition shows that a descent direction can be computed by solving Problem (7).

Proposition 2.1 *Suppose that ξ_0 be the solution of (7) and $g = -H\xi_0$. Then, we have $f_{\varepsilon}^{\circ}(x,g) = -\|\xi_0\|_H$, and there exists $\alpha > 0$ such that*

$$f(x+tg) - f(x) \le t f_{\varepsilon}^{\circ}(x,g), \quad \forall t \in (0,\alpha).$$

Proof Let $H = R^T R$ be the Cholesky factorization of H. For all $v \in R\partial_{\varepsilon} f(x)$, there exists $\xi \in \partial_{\varepsilon} f(x)$ such that $v = R\xi$. So we have $\|v\|^2 = \langle R\xi, R\xi\rangle = \langle \xi, H\xi\rangle$. Thus, the following problem

$$\min \|v\|^2 \ \ s.t. \ \ v \in R\partial_{\varepsilon} f(x), \tag{8}$$

is equivalent to (7). If ξ_0 is the solution of (7), then $R\xi_0$ is the solution of (8). Since $R\partial_\varepsilon f(x)$ is convex, then we have [6]

$$\langle R\xi, R\xi_0\rangle \geq \langle R\xi_0, R\xi_0\rangle, \ \forall \xi \in \partial_\varepsilon f(x),$$

and thus

$$\langle \xi, H\xi_0\rangle \geq \langle \xi_0, H\xi_0\rangle, \ \forall \xi \in \partial_\varepsilon f(x). \tag{9}$$

Since $\xi_0 \in \partial_\varepsilon f(x)$, then (9) shows that $f_\varepsilon^\circ(x, g) = -\|\xi_0\|_H$. Now, suppose that $\alpha = \frac{\varepsilon}{\|g\|}$. So, we have $x + tg \in B(x, \varepsilon)$ for all $t \in (0, \alpha]$. The rest of theorem follows from the Lebourge's mean value Theorem. □

In subgradient based algorithms, it is assumed that at each point one subgradient is available. In this paper, we use the same assumption to design an algorithm. If it is not possible to compute such a subgradient then the algorithm terminates. Let $\phi(\alpha) := f(x + \alpha g)$. We have [39] $\partial\phi(\alpha) \subset \langle \partial f(x + \alpha g), g\rangle$, where

$$\langle \partial f(x + \alpha g), g\rangle = \{\langle \xi, g\rangle \ : \ \xi \in \partial f(x + \alpha g)\}.$$

The equality holds if f is regular and in general the equality does not hold if f is not regular. If for all $v \in \mathbb{R}^n$, the one-side directional derivative $f'(x, v)$ exists and $f'(x, v) = f^\circ(x, v)$, then f is regular at x. In such a case the above formula cannot be used to compute subgradients of ϕ. However, according to the above mentioned assumption we assume that one subgradient of ϕ is available at any $\alpha \geq 0$.

2.3 The BFGS method

Suppose that F is continuously differentiable. Let H_k be a positive definite matrix and the approximation of its inverse Hessian and $\nabla F(x_k)$ be its gradient at x_k. We know that $g = -H_k\nabla F(x_k)$ is a descent direction. The approximation of inverse Hessian can be updated by the BFGS method, when the computed step length satisfies in the Wolfe conditions. For given constants $0 < c_1 < c_2 < 1$ the Wolfe conditions are formulated as:

$$F(x + \alpha g) - F(x) \leq c_1\alpha\langle \nabla F(x_k), g_k\rangle \tag{10}$$

$$\langle \nabla F(x_k + \alpha_k g_k), g_k\rangle \geq c_2\langle \nabla F(x_k), g_k\rangle. \tag{11}$$

Consider the following notations:

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla F(x_{k+1}) - \nabla F(x_k).$$

If the step length α_k satisfies in the Wolfe conditions, then we have $\langle y_k, s_k\rangle > 0$. This inequality is known as secant inequality. If the secant inequality holds, then H_k can be updated by the BFGS method

$$H_{k+1} = (I - \rho_k s_k y^T)H_k(I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T,$$

where $\rho_k = \frac{1}{\langle y_k, s_k\rangle}$.

In nonsmooth case, suppose

$$w_k = \arg\min_{v\in\partial_\varepsilon f(x_k)} \|v\|_{H_k}, \tag{12}$$

and $g_k = -H_k w_k$, where H_k is a positive definite matrix. Let α_k be a step length and $x_{k+1} = x_k + \alpha_k g_k$ be the next iteration. To update H_k by the BFGS method, we must select $v_k \in \partial f(x_k)$ and $v_{k+1} \in \partial f(x_{k+1})$ such that $\langle y_k, s_k \rangle > 0$, where $y_k = v_{k+1} - v_k$. We use an approach proposed in [40] where the authors generalized the Wolfe conditions for nonsmooth convex functions using subgradients. For a convex function f, since f is regular, then the step length α satisfies the Wolfe conditions if the following inequalities hold for constants $c_1 \in (0, 1)$ and $c_2 \in (c_1, 1)$

$$f(x + \alpha g) - f(x) \leq c_1 \alpha \sup_{v \in \partial f(x)} \langle v, g \rangle = c_1 \alpha f'(x, g),$$

and

$$\sup_{v \in \partial f(x+\alpha g)} \langle v, g \rangle \geq c_2 \sup_{v \in \partial f(x)} \langle v, g \rangle.$$

In this paper, we generalize the Wolfe conditions for the locally Lipschitz functions by the ε-subdifferential. If $\varepsilon = 0$, then this generalization is the Yu et al.'s generalization [40]. Then, a line search algorithm is developed based on the generalized Wolfe conditions which coincide with the Wolfe conditions for smooth optimization when the objective function is smooth. The proposed line search algorithm require computation of only one subgradient from $\partial f(x)$ and the full computation of this set is not necessary.

In this paper, we use the following lemmas from [36].

Lemma 2.1 *Let $h : \mathbb{R} \to \mathbb{R}$ be a locally Lipschitz function around r. If h is decreasing in a neighborhood of r, then $\xi \leq 0$ for all $\xi \in \partial h(r)$, and if $\xi < 0$ for all $\xi \in \partial h(r)$, then h is decreasing in a neighborhood of r.*

Lemma 2.2 *If $h : \mathbb{R} \to \mathbb{R}$ is locally Lipschitz and $h(b) > h(a)$ such that $a < b$, then there exists $\theta_0 \in [a, b]$ such that h is increasing in a neighborhood of θ_0.*

3 Line search algorithm

In this section, an approximation of Goldstein ε-subdifferential is used to generalize the Wolfe conditions for locally Lipschitz functions. Then, we show that there exist step lengths satisfying these conditions. Finally, an algorithm is presented to find such step lengths.

3.1 Generalized Wolfe conditions

In this paper, we suppose that H is a positive definite matrix and $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz function. Let $W \subset \partial_\varepsilon f(x)$ and

$$w = \arg \min_{v \in \mathrm{conv} W} \|v\|_H . \tag{13}$$

Define $g = -Hw$. Since $\mathrm{conv}W$ is an approximation of $\partial_\varepsilon f(x)$, then w can be considered as approximation of (7). Suppose that ξ_0 is a solution of (7). Thus by

Proposition 2.1, $-\|w\|_H$ can be considered as an approximation of $f_\varepsilon^\circ(x, -H\xi_0)$. Based on an approximation of $\partial_\varepsilon f(x)$, we generalize the Armijo condition.

Definition 3.1 Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz, $W \subset \partial_\varepsilon f(x)$, w be the solution of (13) and $g = -Hw$. Iff the following inequality holds for a step length α and fixed constant $c_1 \in (0, 1)$

$$f(x + \alpha g) - f(x) \leq -c_1 \alpha \|w\|_H ,$$

then α satisfies in the generalized Armijo condition (GAC).

If $f^\circ(x+\alpha g, g)$ is only slightly negative or even positive, then by (4) f cannot sufficiently be decreased along g with lager step length. This leads to the generalization of the curvature condition for locally Lipschitz functions as follows.

Definition 3.2 Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz, $W \subset \partial_\varepsilon f(x)$, w be the solution of (13) and $g = -Hw$. The step length α satisfies in the nonsmooth curvature inequality, iff the following inequality holds for constant $c_2 \in (c_1, 1)$,

$$f^\circ(x + \alpha g, g) \geq -c_2 \|w\|_H . \tag{14}$$

Lemma 3.1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz, $W \subset \partial_\varepsilon f(x)$, w be the solution of (13) and $g = -Hw$. If there exists $\xi \in \partial f(x + \alpha g)$ such that $\langle \xi, g\rangle \geq -c_2 \|w\|_H$, then the nonsmooth curvature inequality holds.*

Proof By (3), we have $f^\circ(x + \alpha g, g) \geq -c_2 \|w\|_H$. □

Now, we present the generalized Wolfe Conditions (GWC).

Definition 3.3 Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz, $W \subset \partial_\varepsilon f(x)$, w be the solution of (13) and $g = -Hw$. For constants $0 < c_1 < c_2 < 1$, iff there exist $\xi \in \partial f(x+\alpha g)$ such that $\langle \xi, g\rangle \geq -c_2 \|w\|_H$ and the GAC satisfies in α along direction g at x, then we say that the step length α satisfies the GWC along direction g at x.

Now, we prove that for each descent direction, there exist intervals containing step lengths satisfying the GWC. Before its proof, we define the following function,

$$W(\alpha) := f(x + \alpha g) - f(x) + c_2 \alpha \|w\|_H , \tag{15}$$

and prove the following lemma.

Lemma 3.2 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz, $W \subset \partial_\varepsilon f(x)$, w be the solution of (13) and $g = -Hw$. If $W(\cdot)$ is not decreasing on neighborhood of α, then the nonsmooth curvature inequality holds at α.*

Proof Suppose that $W(\cdot)$ is not decreasing in the neighborhood of α, by Lemma 2.1, there exists $v \in \partial W(\alpha)$ such that $v \geq 0$. Since, we have

$$\partial W(\alpha) \subseteq \langle \partial f(x + \alpha g), g\rangle + c_2 \|w\|_H , \tag{16}$$

then there exists $\xi \in \partial f(x+\alpha g)$ such that $v = \langle \xi, g\rangle + c_2 \|w\|_H$. So, $\langle \xi, g\rangle \geq -c_2 \|w\|_H$. Thus, by Lemma 3.1, the nonsmooth curvature inequality holds for all step lengths which $W(\cdot)$ is not decreasing on their neighborhood. □

Proposition 3.1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz, $W \subset \partial_\varepsilon f(x)$, w be the solution of (13) and $g = -Hw$. Suppose that f be bounded below along the ray $\{x+\alpha g | \alpha > 0\}$ and there exists a step which satisfied the GAC along the direction g at x. If $0 < c_1 < c_2 < 1$, then there exist open intervals of step lengths satisfying the GWC.*

Proof Let $\phi(\alpha) = f(x+\alpha g)$ and $l(\alpha) = f(x) - c_1\alpha \|w\|_H$. By (4), we have

$$\phi(\alpha) < l(\alpha), \quad \text{for all } \alpha \in (0, \varepsilon). \tag{17}$$

Since $-\|w\|_H < 0$ and $c_1 > 0$, then $l(\alpha)$ is unbounded below. On the other hand, $\phi(\alpha)$ is bounded below for all $\alpha > 0$. Thus, there exists $\alpha_0 > 0$ such that

$$\phi(\alpha) > l(\alpha), \quad \text{for all } \alpha \geq \alpha_0. \tag{18}$$

By (17) and (18), there exists some $\alpha > 0$ such that $\phi(\alpha) = l(\alpha)$. Let $\alpha_1 > 0$ be the smallest value such that $\phi(\alpha_1) = l(\alpha_1)$. Therefore, we have, $f(x+\alpha_1 g) = f(x) - c_1\alpha_1 \|w\|_H$. Thus, the GAC is satisfied for all $\alpha \in (0, \alpha_1)$ and we have $f(x+\alpha g) - f(x) \leq -c_1\alpha \|w\|_H$. By the similar way, let $\alpha_2 > 0$ be the smallest value such that $W(\alpha_2) = 0$. Since $c_1 < c_2$, then we have $\alpha_2 < \alpha_1$. Since $W(t) < 0$ for all $t \in (0, \alpha_2)$ and $W(0) = W(\alpha_2) = 0$, then W takes its minimum on $(0, \alpha_2)$. Therefore, there exist some open subintervals in $(0, \alpha_2)$ such that W is increasing and thus $v \geq 0$ for all $v \in \partial W(\cdot)$ on them. By Lemma 3.2, for some $\xi \in \partial f(x+\alpha g)$, we have $\langle \xi, g\rangle \geq -c_2 \|w\|_H$. Since $\alpha_2 < \alpha_1$, then all step length in that subinterval satisfy the GAC and GWC. □

3.2 Nonsmooth line search algorithm

The main idea of the nonsmooth line search algorithm is similar to the smooth version [38] and it converts to the smooth version, when f is continuously differentiable. The first stage starts with a trial estimate of α_1 and keeps increasing until it finds either an acceptable step length or an interval that contains acceptable step lengths. The second stage is started by calling Algorithm 2. This algorithm reduces the size of the interval until a step length satisfying the GWC is found. Now, we present the nonsmooth line search algorithm. In the presented algorithms in this subsection, let $W \subset \partial_\varepsilon f(x)$, w be the solution of (13) and $g = -Hw$. Also assume that ε is satisfied the GAC along direction g at x. In Section 4, we present Algorithm 4 to compute such an approximation. To simplify the notations, we define the following one variable function,

$$A(\alpha) := f(x+\alpha g) - f(x) + c_1\alpha \|w\|_H .$$

In Algorithm 1, the step lengths $\{\alpha_i\}$ are monotonically increasing, until a step length or an interval containing a step length satisfying the GWC is found. Suppose that $A(\alpha_i) \geq 0$. Since $A(\alpha_{i-1}) \leq 0$, then $A(\alpha_{i-1}) \leq A(\alpha_i)$. So, by Lemma 2.2,

Algorithm 1 Line Search Algorithm

$\alpha_0 \leftarrow 0$
$\alpha_1 \leftarrow 1$
$i \leftarrow 1$
repeat
  if $A(\alpha_i) \geq 0$ **then**
    $\alpha_* \leftarrow \text{Wolfe}(\alpha_i, \alpha_{i-1})$
    STOP
  end if
  compute $\xi \in \partial f(x + \alpha_i g)$
  such that $\langle \xi_k, g \rangle + c_2 \|w\|_H \in \partial W(t_k)$.
  if $\langle \xi, g \rangle + c_2 \|w\|_H \geq 0$ **then**
    $\alpha_* \leftarrow \alpha_i$
    STOP
  else
    $\alpha_{i+1} \leftarrow 2\alpha_i$
  end if
end repeat

$[\alpha_{i-1}, \alpha_i]$ contains an interval such that A is negative and increasing on this interval. So, in this interval, the GWC is satisfied and Algorithm 1 invokes Wolfe algorithm. The following proposition shows that Algorithm 1 is terminated after finitely many iterations.

Proposition 3.2 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz, $W \subset \partial_\varepsilon f(x)$, w be the solution of (13) and $g = -Hw$. Also assume that ε is satisfied the GAC along direction g at x. If f is bounded below along the ray $\{x + \alpha g | \alpha > 0\}$, then Algorithm 1 terminates after finitely many iterations.*

Proof Since $f(x+\alpha g)$ is bounded below and $f(x)-c_1\alpha \|w\|_H$ is unbounded below, then there exits $\bar{\alpha}$ such that

$$f(x + \alpha g) > f(x) - c_1\alpha \|w\|_H , \ \forall \alpha > \bar{\alpha},$$

i.e. $A(\alpha) > 0$ for all $\alpha > \bar{\alpha}$. Thus, Algorithm 1 terminates after finitely many iterations. □

Now we present the second stage of algorithm.
The following proposition describes the behavior of Algorithm 2.

Proposition 3.3 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz, $W \subset \partial_\varepsilon f(x)$, w be the solution of (13) and $g = -Hw$. Also assume that ε is satisfied the GAC along direction g at x. Either Algorithm 2 terminates after finitely many iterations, or it generates a sequence of intervals $[a_k, b_k]$, such that each one contains some subintervals satisfying the GWC and a_k and b_k converge to a step length $t_0 > 0$. Also,*

Algorithm 2 Wolfe Algorithm

$k \leftarrow 1$
$a_k \leftarrow \alpha_{i-1}$
$b_k \leftarrow \alpha_i$
repeat
  $t_k \leftarrow \frac{a_k+b_k}{2}$
  if $A(t_k) \geq 0$ **then**
    $b_{k+1} \leftarrow t_k$
    $a_{k+1} \leftarrow a_k$
  else
    compute $\xi_k \in \partial f(x + t_k g)$
    such that $\langle \xi_k, g \rangle + c_2 \|w\|_H \in \partial W(t_k)$.
    if $\langle \xi_k, g \rangle + c_2 \|w\|_H \geq 0$ **then**
      $\alpha_* \leftarrow t$
      STOP
    else
      $b_{k+1} \leftarrow b_k$
      $a_{k+1} \leftarrow t_k$
    end if
  end if
  $k \leftarrow k + 1$
end repeat

there exist $\zeta_1, \zeta_2, \zeta_3 \in \partial f(x + t_0 g)$ *such that* $\langle \zeta_1, g \rangle \leq -c_2 \|w\|_H$, $\langle \zeta_2, g \rangle \geq -c_2 \|w\|_H$ *and* $\langle \zeta_3, g \rangle \geq -c_1 \|w\|_H$.

Proof If the algorithm terminates after finitely many iterations, then there is nothing to prove. Suppose that the algorithm does not terminate after finitely many iterations. Since a_k and b_k are monotone sequence thus they are convergent. On the other hand, we have $b_k - a_k = \frac{b_1 - a_1}{2^{k-1}}$, thus $\lim_{k\to\infty} b_k - a_k = 0$. Therefore, these sequences are convergent to same point such as t_0. We suppose that L is the Lipshitz constance for f over $\{x + \alpha g : \alpha \in [a_1, b_1]\}$.

We prove that a_k will be positive after finitely many iterations. Suppose that $a_1 = 0$. Since g is a descent direction, then there exists $\alpha > 0$, such that $A(s) < 0$ for all $s \in (0, \alpha)$. If $a_m = 0$, then we must have $A(t_k) > 0$ for all $k = 1, \ldots, m$. In these iterations, we have $b_{k+1} = t_k$, $a_{k+1} = a_k = 0$ and $t_{k+1} = \frac{b_k}{2} = \frac{\alpha_i}{2^k}$ for all $k = 1, \ldots, m$. Therefore, after finitely many iterations $t_k \leq \alpha$. In this iteration, since $A(t_k) \leq 0$, then $a_{k+1} = t_k$.

Let S be all the iterations such that $a_{k+1} = t_k$. Therefore, $\langle \xi_k, g \rangle < -c_2 \|w\|_H$ for all $k \in S$. We have $\|\xi_k\| \leq L$ for all $k \in S$. Thus, the sequence $\{\xi_k\}$ contains a convergent subsequence. Therefore, without loss of generality, we can assume this

sequence is convergent and $\zeta_1 = \lim_{k\in S, k\to\infty} \xi_k$. By the upper semicontinuously of $\partial f(\cdot)$ and $t_k \to t_0$, we have $\zeta_1 \in \partial f(x + t_0 g)$ and $\langle \zeta_1, g\rangle \leq -c_2 \|w\|_H$.

Since $a_k < b_k$, $A(a_k) < 0$ and $A(a_k) < A(b_k)$, then by Lemma 2.2, $A(\cdot)$ contains a step length r_k such that $A(\cdot)$ is increasing on its neighborhood and $A(r_k) < 0$. On the other hand, $c_1 < c_2$, therefore $W(\cdot)$ is also increasing in a neighborhood of r_k. Therefore, the GWC is satisfied at r_k. If $\langle \kappa_k, g\rangle + c_2 \|w\|_H \in \partial W(r_k)$ for some $\kappa_k \in \partial f(x + r_k g)$, then by lemma 2.1 $\langle \kappa_k, g\rangle + c_2 \|w\|_H \geq 0$. Since $\{\kappa_k\}$ has a convergent subsequence, then without loss of generality, suppose that $\zeta_2 = \lim_{k\to\infty} \kappa_k$. Since $\partial f(\cdot)$ is upper semicontinuous and $r_k \to t_0$, then $\zeta_2 \in \partial f(x + t_0 g)$ and we have $\langle \zeta_2, g\rangle \geq -c_2 \|w\|_H$. Since $A(\cdot)$ is increasing on a neighborhood of r_k, then $v \geq 0$ for all $v \in \partial A(r_k)$. Thus, we have $\langle \eta_k, g\rangle \geq -c_1 \|w\|_H$, for all $\eta_k \in \partial f(x + r_k g)$ such that $\langle \eta_k, g\rangle + c_1 \|w\|_H \in \partial A(r_k)$. Since $\{\eta_k\}$ has a convergent subsequence, then without loss of generality, suppose that $\zeta_3 = \lim_{k\to\infty} \eta_k$. Since $r_k \to t_0$, then $\zeta_3 \in \partial f(x + t_0 g)$ and we have $\langle \zeta_3, g\rangle \geq -c_1 \|w\|_H$. □

Corollary 3.1 *Assume that all assumptions of Proposition 3.3 are satisfied and f is continuously differentiable almost everywhere. If Algorithm 2 does not terminate after finitely many iterations and the sequence $\{t_k\}$ converges the step length t_0, then t_0 belongs to a set with zero measure.*

Proof Suppose that Algorithm 2 does not terminate after finitely many iterations and converges to t_0. If f is continuously differentiable at t_0, then

$$\partial f(x + t_0 g) = \{\langle \nabla f(x + t_0 g)\}\,. \tag{19}$$

So, by Proposition 3.3 and (19), we have

$$\langle \nabla f(x + t_0 g), g\rangle \geq -c_1 \|w\|_H \text{ and } \langle \nabla f(x + t_0 g), g\rangle \leq -c_2 \|w\|_H\,.$$

These relations are in contradiction with $c_1 < c_2$. So, if Algorithm 2 does not terminate after finitely many iterations, then f is not continuously differentiable at t_0. On the other hand, f is not continuously differentiable in a set with zero measure. □

Now, we show that Wolfe algorithm terminates after finitely many iterations for semismooth functions.

Corollary 3.2 *Assume that all assumptions of Proposition 3.3 are satisfied and f is a semismooth function. Then Wolfe algorithm terminates after finitely many iterations.*

Proof Suppose that Wolfe algorithm does not terminate after finitely many iterations and converges to t_0. In each iterations, we have $A(a_k) < 0$ and $A(b_k) \geq 0$. On the other hand, a_k and b_k converge to t_0. Thus

$$A(t_0) = 0. \tag{20}$$

Let S be the set of all indices such that $a_{k+1} = t_k$. We show that S is an infinite set. By contrary, if S is finite, then there exist k_0 such that $a_{k_0+1} = t_{k_0}$ and $b_{k+1} = t_k$ for all $k > k_0$. Thus, we have $t_0 = \lim_{k\to\infty} t_k = a_{k_0+1}$. Therefore, $A(t_0) = A(a_{k_0+1}) < 0$ and this is in contradiction with (20). Therefore, S is an infinite set. Since f is semismooth, then f is directionally differentiable and $f'(x, g) = f^\circ(x, g)$ [9]. Also, we have

$$\lim_{\substack{k\to\infty \\ v_k \in \partial f(x+l_k g) \\ l_k \downarrow 0}} \langle v_k, g\rangle = f'(x, g). \tag{21}$$

The nonsmooth curvature does not satisfy at t_k. Thus, we have $\langle \xi_k, g\rangle < -c_2 \|w\|_H$ for all $k \in S$ where $\xi_k \in \partial f(x + t_k g)$. Since $a_k \le t_0$, then by (21) we have $\lim_{k\in S, k\to\infty} \langle \xi_k, g\rangle = f'(x + t_0 g, -g)$. Thus

$$f'(x + t_0 g, -g) \le -c_2 \|w\|_H . \tag{22}$$

Since $A(a_k) < 0$ and $A(t_0) = 0$, then by Lemma 2.2 there exists $r_k \in (a_k, t_0)$ such that $A(\cdot)$ is increasing on its neighborhood. Therefore, by Lemma 2.1, $\xi > 0$ for all $\xi \in \partial A(r_k)$. Hence, for some $\eta_k \in \partial f(x + r_k g)$, we have $\langle \eta_k, g\rangle \ge c_1 \|w\|_H$. Since $r_k \le t_0$, then by (21) we have $\lim_{k\to\infty} \langle \eta_k, g\rangle = f'(x + t_0 g, -g)$. Thus

$$f'(x + t_0 g, -g) \ge -c_1 \|w\|_H . \tag{23}$$

This is in contradiction with (22). Since $c_1 < c_2$. Therefore, Wolfe algorithm must be terminated after finitely many iterations. □

If Wolfe algorithm does not terminate after finitely many iterations, then $W(\cdot)$ has infinite number of extremum point in $[a_1, b_1]$. Otherwise by Proposition 3.5, Wolfe algorithm terminates after finitely many iterations. Similar situation may happen in the smooth case, for example [41, Theorem 2.3] and [42, Theorem 2.1]. Let $F : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function and g be a descent direction at x. Define $\psi(\alpha) = F(x + \alpha g) - F(x) - c_2\alpha\langle \nabla F(x), g\rangle$. If the line search algorithm does not terminate after finitely many iterations, then the sign of $\psi'(\alpha)$ is changed in infinite number of times, i.e., $\psi(\alpha)$ has infinite number of the local extremum. The line search algorithm converges to α^* such that $\psi'(\alpha^*) = 0$. A similar result for locally Lipschitz functions is proven in Proposition 3.4. But here by Corollary 3.1, we prove that in smooth functions, the line search algorithm must be terminated after finitely many iterations.

Proposition 3.4 *Assume that all assumptions of Proposition 3.3 are satisfied and Algorithm 2 does not terminate after finitely many iterations and converges to* t_0. *Then* $0 \in \langle \partial f(x + t_0 g), g\rangle + c_2 \|w\|_H$.

Proof By Proposition 3.3 there exist $\eta_1, \eta_2 \in \partial f(x + t_0 g)$ such that $\langle \zeta_1, g\rangle \le -c_2 \|w\|_H$ and $\langle \zeta_2, g\rangle \ge -c_2 \|w\|_H$. Let $s_1 = \langle \zeta_2, g\rangle + c_1 \|w\|_H$ and $s_2 = \langle \zeta_3, g\rangle +$

$c_1 \|w\|_H$. We have $s_1, s_2 \in \langle \partial f(x + t_0 g), g\rangle + c_2 \|w\|_H$. Since $s_1 \leq 0$, $s_2 \geq 0$ and $\langle \partial f(x + t_0 g), g\rangle + c_2 \|w\|_H$ is convex, then $0 \in \langle \partial f(x + t_0 g), g\rangle + c_2 \|w\|_H$. □

If f does not has an infinite number of local extremal points in any bounded set, then there exist finitely many subintervals such that $W(\cdot)$ is increasing on there. Commonly, functions have this property. By this assumption, we can prove that this algorithm terminates after finitely many iterations for any locally Lipschitz function.

Proposition 3.5 *Assume that all assumptions of Proposition 3.3 are satisfied and $W(\cdot)$ has a finite number of local extremal points in any bounded set, then Algorithm 2 terminates after finitely many iterations.*

Proof Since W has a finite number of local extremal points on $(\alpha_i, \alpha_{i+1}]$, then it is increasing on a finite number of subintervals in $(\alpha_i, \alpha_{i+1}]$. Let $d_\varepsilon > 0$ be the shortest length of subinterval in $(\alpha_i, \alpha_{i+1}]$ on which W is increasing. In each iteration, the length of $[a_k, b_k]$ is divided in two, so after k iterations we have $b_k - a_k = \frac{\alpha_i - \alpha_{i-1}}{2^{k-1}}$. Since $W(b_k) > W(a_k)$ and $b_k > a_k$, then (a_k, b_k) includes at least a subinterval on which W is increasing. After finitely many iterations, such as k, we have $b_k - a_k \leq d_\varepsilon$. On the other hand, d_ε is the shortest length of subinterval on which W is increasing. Therefore, at iteration k, W is increasing on t_k. Thus, by Lemma 2.1, $\langle \xi_k, g\rangle + c_2 \|w\|_H \geq 0$. Hence, the algorithm terminates after k iterations. □

In [9] based on the smooth line search algorithm, a line search algorithm is developed for semismooth functions. In this algorithm, two parameters are updated in each iterations. If the step length satisfies the Armijo condition the first parameter, t_L, is updated. The second parameter, t_R, is updated, when the step length satisfied the curvature condition. Thus t_R and t_L will be equal if the step length satisfied the Wolfe conditions.This algorithm terminates, when $t_R - t_L$ is less than a threshold. By this condition, the finite termination of the line search algorithm is proved for semismooth functions. In this paper, we show that the smooth line search algorithm can be applied to locally Lipschitz functions by replacing the directional derivative by the generalized directional derivative. Also, the Wolfe conditions are generalized and we show that at each iteration of line search algorithm there is a step length, which satisfies this generalization. In this paper by Corollary 3.2, the finite termination of the line search algorithm is proved for semismooth functions.

In the finite precision arithmetic, if the length of interval $[a_k, b_k]$ is too small, then two function values $f(x + a_k g)$ and $f(x + b_k g)$ may be indistinguishable. So, in practice, Algorithm 2 must be terminated after finitely many iterations [38]. In our numerical experiments, Algorithm 2 terminates after 30 iterations. If Algorithm 2 does find a step length satisfying the GWC, then we select a step satisfying the GAC. Since $A(a_k) < 0$, then we set a_k as a step length. In this case, the step length does not satisfy the GWC. Thus, the approximation of inverse Hessian cannot be updated.

In such iterations, the approximation of inverse Hessian is initialized by the identity matrix.

4 Combining the steepest descent method by the BFGS algorithm

In Proposition 2.1, we show that a descent direction can be computed by the solution of (7). Since it is not always possible to compute the whole set $\partial_\epsilon f(\cdot)$ then, $\partial_\varepsilon f(x)$ is approximated and based on its approximation a descent direction is computed. In this section, we present an algorithm for computing a descent direction based on a positive definite matrix and $\partial_\varepsilon f(x)$. At each iteration, an approximation of $\partial_\varepsilon f(x)$ is improved by adding a new element. We prove that this algorithm terminates after finitely many iterations. The nonsmooth line search algorithm is applied along this direction and a step length satisfying the GWC is computed. We discuss how the positive definite matrix is updated by the BFGS method and subgradients. Finally, the convergence of the minimization algorithm is proven.

4.1 Computing descent direction

We approximate $\partial_\varepsilon f(x)$ by the convex hull of the finite number of ε-subgradients. More exactly, if $W_k = \{v_1, v_2, \ldots, v_k\} \subset \partial_\varepsilon f(x)$ then we consider $\mathsf{conv}W_k$ as an approximation of $\partial_\varepsilon f(x)$. Therefore, the solution of the following problem is an approximation of the solution of (7),

$$w_k = \arg \min_{v \in \mathsf{conv}W_k} \|v\|_H . \tag{24}$$

Since $\mathsf{conv}W_k$ is an approximation of $\partial_\varepsilon f(x)$, then w_k is an approximation of ξ_0 in (7). On the other hand, we have $f_\varepsilon^\circ(x, g) = -\|\xi_0\|_H$, therefore $-\|w_k\|_H$ can be considered as an approximation of $f_\varepsilon^\circ(x, g)$. Equation (24) is equivalent to a quadratic programming problem and there exist several efficient methods for computing its solution [43–45]. We use the method described in [45]. Set $g = -Hw_k$; if we have

$$f(x + \varepsilon g) - f(x) \leq -c_1 \varepsilon \|w_k\|_H , \tag{25}$$

for some constant $c_1 \in (0, 1)$, then $\mathsf{conv}W_k$ is an acceptable approximation of $\partial_\varepsilon f(x)$. If the sufficient decrease (25) is not satisfied, then $\mathsf{conv}W_k$ (which is an approximation of $\partial_\varepsilon f(x)$) must be improved by adding a new element of $\partial_\varepsilon f(x)$ into W_k, such that this element does not belongs to $\mathsf{conv}W_k$. How to select such an element is described in the following proposition.

Proposition 4.1 *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is a Locally Lipschitz function, $0 \notin \partial_\varepsilon f(x)$, W_k is a collection of k elements of $\partial_\varepsilon f(x)$, $w_k = \arg\min_{v \in \mathsf{conv}W_k} \|v\|_H$ and $g = -Hw_k$. If (25) is not satisfied, then there exists $v \in \partial f(x + \varepsilon g)$ such that $v \notin \mathsf{conv}W_k$.*

Proof Since (25) is not satisfied, the we have $A(\varepsilon) > 0$. On the other hand, $A(0) = 0$. Therefore, by Lemma 2.2, there exists $\alpha \in (0, \varepsilon)$ such that A is increasing on its neighborhood. Since A is increasing on a neighborhood of α, then $\xi \geq 0$ for all $\xi \in \partial A(\alpha)$. We have

$$\partial A(\alpha) \subseteq \langle \partial f(x + \alpha g), g \rangle + c_1 \|w_k\|_H ,$$

thus

$$\langle -v, Hw_k \rangle \geq -c_1 \|w_k\|_H \tag{26}$$

for all $v \in \partial f(x + \alpha g)$, where $\langle v, g \rangle + c_1 \|w_k\|_H \in \partial A(\alpha)$. Consider the Cholesky factorization of H, i.e., $H = R^T R$. Since $R\mathsf{conv}W_k$ is convex and Rw_k is its element with minimum norm, then

$$\langle Hw_k, v \rangle = \langle Rw_k, Rv \rangle \geq \|w_k\|_H , \quad \text{for all } v \in \mathsf{conv}W_k. \tag{27}$$

Therefore, by (27) and (26) we have $v \notin \mathsf{conv}W_k$, for all $v \in \partial f(x + \alpha g)$ such that $\langle v, g \rangle + c_1 \|w_k\|_H \in \partial A(\alpha)$. □

To improve $\mathsf{conv}W_k$ as an approximation of $\partial_\varepsilon f(x)$, by Proposition 4.1, we need to find a point in $(0, \varepsilon]$ such that A is increasing on that neighborhood. Now, an algorithm is presented to find such a point. The idea of this algorithm is similar to Algorithm 2. Let W_k be a finite subset of $\partial_\varepsilon f(x)$,

$$w_k = \arg \min_{v \in \mathsf{conv}W_k} \|v\|_H ,$$

and suppose that (25) is not satisfied. Similar to Algorithm 2, in each iteration of the following algorithm, we have an interval, which contains a point which A is increasing on its neighborhood and the interval length is halved. The algorithm starts with the initial interval, $[0, \varepsilon]$ such that $A(0) = 0$ and $A(\varepsilon) > 0$. Thus, by Lemma 2.2, $[0, \varepsilon]$ contains a point, which A is increasing on its neighborhood. In each iteration, we have an interval $[a, b]$ such that $A(a) < A(b)$ and $a < b$, therefore it contains a point, which A is increasing on its neighborhood. When $A(\cdot)$ is increasing on a neighborhood of t, then by Lemma 2.1 $\langle v, g \rangle + c_1 \|w_k\|_H \geq 0$ for all $v \in \partial f(x + tg)$ such that $\langle v, g \rangle + c_1 \|w_k\|_H \in \partial A(t)$. If at any iteration $\langle v, g \rangle + c_1 \|w_k\|_H \geq 0$, then by Proposition 4.1 $v \notin \mathsf{conv}W_k$ and Algorithm 3 is terminated. Now, we present an algorithm to find a point belonging to $(0, \varepsilon]$ such that A is increasing on its neighborhood.

Remark 4.1 If $\|g\| \leq 1$, then $v \in \partial_\varepsilon f(x)$. Otherwise, we define

$$\overline{g} = \frac{g}{\|g\|}, \ \overline{c} = \frac{c_1}{\|g\|}, \tag{28}$$

and $\overline{A}(t) = f(x + t\overline{g}) - f(x) + t\overline{c} \|w_k\|_H$. If $\overline{A}$ is increasing on a neighborhood α, then

$$\langle v, \overline{g} \rangle + \overline{c} \|w_k\|_H \geq 0, \tag{29}$$

Algorithm 3 Finding increasing point

$b \leftarrow \varepsilon$
$a \leftarrow 0$
$t \leftarrow b$
repeat
  compute $v \in \partial f(x + tg)$ such that $\langle v, g\rangle + c_1 \|w_k\|_H \in \partial A(t)$
  if $\langle v, g\rangle + c_1 \|w_k\|_H \geq 0$ **then**
    STOP
  else if $A(b) > A(t)$ **then**
    $a \leftarrow t$
  else
    $b \leftarrow t$
  end if
  $t \leftarrow \frac{a+b}{2}$
end repeat

for all $v \in \partial f(x+\alpha\overline{g})$ and some $\alpha \in (0, \varepsilon]$. By (28) and (29), $\langle v, g\rangle + c_1 \|w_k\|_H \geq 0$. This inequality shows that $v \notin \mathsf{conv} W_k$ and $v \in \partial_\varepsilon f(x)$. Thus, without loss of generality, we can assume that $v \in \partial_\varepsilon f(x)$ for all $v \in \partial f(x + \alpha g)$ and $\alpha \in (0, \varepsilon]$.

This algorithm is similar to Algorithm 3 in [36]. The following proposition shows the behavior of algorithm.

Proposition 4.2 [36] *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz. Either Algorithm 3 terminates after finitely many iterations, or it generates a sequence of intervals $[a_k, b_k]$, each one containing some subintervals on which A is increasing. These intervals converge to a step length $t_0 > 0$ such that $0 \in \partial A(t_0)$.*

If $A(\cdot)$ does not have an infinite number of local extremal points in $[0, \varepsilon]$, then similar to Proposition 3.5, we can prove that this algorithm terminates after finitely many iterations. Practically, for small $\varepsilon > 0$, applying Algorithm 3 is not costly. We have observed in our numerical experiments that since $A(\cdot)$ does not usually have a local extremum on $(0, \varepsilon]$, then most often the algorithm terminates after one iteration [36].

Now, we present an algorithm to find a descent direction. In each iteration of this algorithm, the approximation of $\partial_\varepsilon f(x)$ is improved by adding a new element and the algorithm terminates, when (25) is satisfied. If H is the identity matrix, then Algorithm 4 converts to Algorithm 4.1 in [36]. Now, we present the algorithm as following, Now, the following proposition shows that Algorithm 4 terminates after finitely many iterations and its proof is similar to [13, Theorem 6.1.].

Algorithm 4 Computing descent direction

Step 0: (Initialize)
Let $v_1 \in \partial f(x)$ and $\delta, c_1, \varepsilon \in (0,1)$. Set $W_1 = \{v_1\}$ and let $l = 1$.

Step 1: (Compute a descent direction)
Solve the following minimization problem and let

$$w_l = \arg \min_{v \in \mathrm{conv} W_l} \|v\|_H \,, \tag{30}$$

If $\|w_l\|_H \leq \delta$ **then stop else** let $g_{l+1} = -Hw_l$.

Step 2: (Stopping condition)
If

$$f(x + \varepsilon g_{l+1}) - f(x) \leq -c_1 \varepsilon \|w_l\|_H \,, \tag{31}$$

then Stop.

Step 3: (Improve upon the approximation of $\partial_\varepsilon f(x)$)
Apply Algorithm 3 at point x along direction g_{l+1} and interval $(0, \varepsilon]$. Suppose that Algorithm 3 returns $\alpha \in (0, \varepsilon]$ and $v \in \partial f(x + \alpha g)$ such that $\langle v, g\rangle + c_1 \|w_k\|_H \geq 0$. We have $v \notin \mathrm{conv} W_l$. Set $v_{l+1} = v$, $W_{l+1} = W_l \cup \{v_{l+1}\}$ and $l = l + 1$. **Go to** Step 1.

Proposition 4.3 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function and for the point $x_1 \in \mathbb{R}^n$, the level set $M = \{x : f(x) \leq f(x_1)\}$ be bounded. Then, for each $x \in M$, Algorithm 4 terminates after finitely many iterations.*

Proof Let L be a Lipschitz constant for f in M. If the stoping condition is not satisfied, based on Algorithm 3, we have $v_{k+1} \notin \mathrm{conv} W_k$ and

$$\langle v_{k+1}, Hw_k\rangle \leq c_1 \|w_k\|_H \,. \tag{32}$$

Now, condition (31) is not satisfied after finitely many iterations, for some m we have $\|w_m\|_H \leq \delta$ and therefore the algorithm terminates. Let $R^T R$ be the Cholesky factorization of H. Since w_{k+1} is an element of $R\mathrm{conv} W_{k+1}$ with minimal norm, then for all $t \in (0,1)$ and $v_{k+1} \in \mathrm{conv} W_{k+1}$ we have,

$$\begin{aligned}
\|w_{k+1}\|_H &\leq \|tv_{k+1} + (1-t)w_k\|_H \\
&= \|w_k + t(v_{k+1} - w_k)\|_H \\
&= \langle w_k + t(v_{k+1} - w_k), H(w_k + t(v_{k+1} - w_k))\rangle \\
&= \langle w_k, Hw_k\rangle + t^2 \langle v_{k+1} - w_k, H(v_{k+1} - w_k)\rangle - 2t\langle w_k, H(v_{k+1} - w_k)\rangle \\
&= \|w_k\|_H + 2t\langle w_k, H(v_{k+1} - w_k)\rangle + t^2 \|v_{k+1} - w_k\|_H \\
&= \|w_k\|_H + 2t(\langle w_k, Hv_k\rangle - \|w_k\|_H) + t^2 \|v_{k+1} - w_k\|_H \,.
\end{aligned}$$

By (1), we have $\|v_{k+1} - w_k\|_H \leq \lambda_n \|v_{k+1} - w_k\|^2$. Since $v_{k+1}, w_k \in \partial_\varepsilon f(x)$, then by (2) we have $\|v_{l+1} - w_k\| \leq 2L$. Thus, $\|v_{k+1} - w_k\|_H \leq 4\lambda_n L^2$. Now, by (32),

$$\|w_{k+1}\|_H \leq \|w_k\|_H - 2t(1-c_1)\|w_k\|_H + 4t^2\lambda_n L^2.$$

By (1) and (2), we have $\|w_k\|_H \leq \lambda_n L^2$. Let $t = \frac{(1-c_1)}{\lambda_n(2L)^2}\|w_k\|_H \in (0,1)$. For given $\delta \in (0, \min(L, \lambda_n))$ and for all $\|w_k\|_H > \delta$, we have,

$$\begin{aligned}\|w_{k+1}\|_H &\leq \left(1 - \|w_k\|_H \frac{(1-c_1)^2}{\lambda_n(2L)^2}\right)\|w_k\|_H \\ &\leq \left(1 - \frac{(1-c_1)^2\delta}{\lambda_n(2L)^2}\right)\|w_k\|_H .\end{aligned}$$

Define $r = 1 - \frac{(1-c_1)^2\delta}{\lambda_n(2L)^2}$. Since $\lambda_n L^2 > \delta$, then $r \in (0,1)$. So, for all $\|w_k\|_H > \delta$, we have,

$$\|w_{k+1}\|_H \leq r\|w_k\|_H \leq \cdots \leq r^k\|w_1\|_H \leq \lambda_n r^k L^2.$$

Therefore, after finitely many iterations, we have $\|w_{k+1}\|_H \leq \delta$ and the algorithm terminates. □

4.2 Minimization algorithm

To update the approximation of inverse Hessian, we need a pair of subgradients from $\partial f(x)$ and $\partial f(x + \alpha g)$ such that the secant equation is satisfied. How to select these subgradients is described in the following proposition.

Proposition 4.4 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function and $\mathrm{conv}W_k$ be an approximation of $\partial_\varepsilon f(x)$. Suppose that $w_k = \arg\min_{v \in \mathrm{conv}W_k}\|v\|_H$ and $g = -Hw_k$ satisfying in (25). If the line search algorithm returns the step length α and subgradient ξ, then $\langle \xi, g\rangle \geq c_1\langle v, g\rangle$ for all $v \in \mathrm{conv}W_k$ and*

$$\langle \xi - v, g\rangle > 0.$$

Proof Let $H = R^T R$ be the Cholesky factorization of H. Since Rw_k is the element of $R\mathrm{conv}W_k$ with the minimum norm, then for all $v \in \mathrm{conv}W_k$ we have $\langle Rv, -Rw_k\rangle \leq \langle Rw_k, -Rw_k\rangle$. Thus

$$\begin{aligned}\langle v, -Hw_k\rangle &= -v^T Hw_k \\ &= v^t R^T Rw_k \\ &= \langle Rv, -Rw_k\rangle \\ &\leq \langle Rw_k, -Rw_k\rangle \\ &= -\|w_k\|_H ,\end{aligned}$$

On the other hand, we have $\langle \xi, -Hw_k \rangle \geq -c_1 \|w_k\|_H$, thus $\langle \xi, -Hw_k \rangle \geq c_1 \langle v, -Hw_k \rangle$ for all $v \in \mathrm{conv}W_k$. This gives

$$\langle \xi - v, -Hw_k \rangle \geq (c_1 - 1)\langle v, -Hw_k \rangle \geq (1 - c_1)\|w_k\|_H > 0.$$

□

Suppose that Algorithm 4 is applied at point x and it returns the descent direction g and $\mathrm{conv}W_l$ as an approximation of $\partial_\varepsilon f(x)$. We apply the line search algorithm at x along direction g. The line search algorithm returns a step length α and a subgradient $\xi \in \partial f(x + \alpha g)$. Proposition 4.4 shows that ξ and each element of $\mathrm{conv}W_k$ satisfies the secant equation. Thus, H can be updated using the BFGS method with ξ and an element of $\mathrm{conv}W_k$. In the implementation of the minimization algorithm, we use the ξ and $v_1 \in W_l$, which Algorithm 4 is initialized v_1, for updating the approximation of inverse Hessian. Now, we present the nonsmooth version of BFGS algorithm.

In Algorithm 5, we have two loops, outer and inner loop. In the inner loop, $\partial_{\varepsilon_k} f(x_k^m)$ is approximated by Algorithm 4 and descent direction is computed. When $\|w_k^m\| \leq \delta_k$, then parameters must be updated. In Theorem 3, we show that the inner loop is terminated after finitely many iterations. The outer loop iterates infinitely. But in the practice, the outer loop terminates when δ_k is less than a threshold.

Now, we show that secant equation is satisfied in Step 4 of Algorithm 5. The line search algorithm returns v_k^{m+1} such that $\langle v_k^{m+1}, g_k^m \rangle \geq -c_2 \|w_k^m\|_{H_k^m}$ and since Rw_k^m is an element of $R\mathrm{conv}W_k^m$ with minimal norm, then we have

$$\langle v_k^m, g_k^m \rangle = \langle v_k^m, H_k^n w_k^m \rangle \leq -\|w_k^m\|_{H_k^m}.$$

Thus,

$$\begin{aligned} \langle y, s \rangle = \alpha \langle v_k^{m+1} - v_k^m, g_k^m \rangle &\geq \alpha \left(-c_2 \|w_k^m\|_{H_k^m} + \|w_k^m\|_{H_k^m} \right) \\ &= \alpha(1 - c_2)\|w_k^m\|_{H_k^m} > 0. \end{aligned}$$

This equation shows that the approximation of inverse Hessian can be updated by the BFGS method.

Remark 4.2 To prove the global convergence of the minimization algorithm, it is required that H_k is bounded. Thus, if the following equation is satisfied,

$$\langle w_k^m, H_k^m w_k^m \rangle < \sigma \|w_k^m\|^2,$$

then we set $H_k^m = H_k^m + \sigma I_n$ and $g_k^m = g_k^m - \sigma w_k^m$, where $\sigma \in (0, 1)$. By this modification, we have $\langle w_k^m, H_k^m w_k^m \rangle \geq \sigma \|w_k^m\|^2$.

The following theorem shows that every accumulation point of the sequence $\{x_k\}$, generated by Algorithm 5, belongs to the set $X = \{x \in \mathbb{R}^n : 0 \in \partial f(x)\}$. The proof is similar to [13, Theorem 6.2].

Algorithm 5 Minimization algorithm

Step 0 (Initialization)
Let $x_1 \in \mathbb{R}^n$, $v_1^1 \in \partial f(x_1)$, $\theta_\varepsilon, c_1, \theta_\delta, \varepsilon_1, \delta_1 \in (0,1)$, $H_1 = I_{n\times n}$, $c_2 \in (c_1, 1)$ and set $k = 1$, where $I_{n\times n}$ is the identity matrix.

Step 1 (Set new parameters)
Set $m = 1$, $H_k^m = H_k$ and $x_k^m = x_k$.

Step 2 (Compute descent direction)
Apply Algorithm 4 at point x_k^m, with $H = H_k^m$, $v_1 = v_k^m$, $\delta = \delta_k$ and $\varepsilon = \varepsilon_k$. Let n_k^m be the number of iterations needed for termination of Algorithm 4 and let $\left\|w_k^m\right\|_{H_k^m} = \min\left\{\|w\|_{H_k} : w \in \mathrm{conv}W_k^m\right\}$. **If** $\left\|w_k^m\right\|_{H_k} = 0$ **then Stop else** let $g_k^m = -H_k^m w_k^m$ be the descent direction.

Step 3 (Line search)
If the stopping condition (31), as given in Algorithm 4, is not satisfied **then go to** Step 5, **else** apply Algorithm 1. **If** Algorithm 2 terminates successfully, **then** α is the line search parameter satisfying the GWC and $v_k^{m+1} \in \partial_\varepsilon f(x_k^m + \alpha g_k^m)$ is a vector such that $\langle v_k^{m+1}, g_k^m\rangle + c_2\left\|w_k^m\right\|_{H_k^m} > 0$, **else** α is the line search parameter satisfying the GAC. Construct the next iterate $x_k^{m+1} = x_k^m + \alpha g_k^m$ and **go to** Step 4.

Step 4 (BFGS update)
If Algorithm 2 terminates successfully, **then** set $s = \alpha g_k^m$, $y = v_k^{m+1} - v_k^m$ and

$$H_k^{m+1} = H_k^m - \frac{H_k^m y y^T H_k^m}{\langle y, H_k^m y\rangle} + \frac{ss^T}{\langle y, y\rangle},$$

else set $H_k^{m+1} = I$. Set $m = m + 1$ and **go to** Step 2.

Step 5 (Update parameters)
Set $\varepsilon_{k+1} = \varepsilon_k \times \theta_\varepsilon$, $\delta_{k+1} = \delta_k \times \theta_\delta$, $x_{k+1} = x_k^m$, $H_{k+1} = H_k^m$ and let $k = k + 1$. **Go to** Step 1.

Theorem 4.1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function. If the level set*

$$M = \{x : f(x) \le f(x_1)\},$$

is bounded, then either Algorithm 5 terminates finitely at some k_0 and m_0 with $\left\|w_{k_0}^{m_0}\right\| = 0$ or every cluster point of the sequence $\{x_k\}$, generated by Algorithm 5, belongs to the set $X = \{x \in \mathbb{R}^n : 0 \in \partial f(x)\}$.

Proof If the algorithm terminates after finitely many iterations at some k_0 and m_0, then we have $0 \in \partial_{\varepsilon_k} f(x_{k_0}^{m_0})$. Therefore, $x_{k_0}^{m_0}$ is an ϵ-subdifferential stationary point. Now, suppose that the algorithm does not terminate after finitely many

iterations. Since M is bounded and f is locally Lipschitz, then we have, $f^* = \inf\{f(x) : x \in \mathbb{R}^n\} > -\infty$. At each point x_k^m, we apply Algorithm 4. By Proposition 4.3, this algorithm is terminated after finitely many iterations. Either it returns the descent direction $g_k^m = -H_k^m w_k^m$ such that (31) is satisfied, or $\left\|w_k^m\right\| \leq \delta_k$. First, for each k, we show that there exists $m_k > 0$ such that $\left\|w_k^{m_k}\right\|_{H_k^{m_k}} \leq \delta_k$. By the contradiction, suppose that $\left\|w_k^m\right\|_{H_k^m} > \delta_k$ for all m, then Proposition 4.3 shows that Algorithm 4 returns the descent direction $g_k^m = -H_k^m w_k^m$, such that (31) is satisfied for all m. So, the line search algorithm is applied for all m and a step length α is computed. Since $A(\varepsilon_k) \leq 0$, then $\alpha \geq \varepsilon_k$. Thus, we have

$$\begin{aligned} f(x_k^{m+1}) - f(x_k^m) &\leq -c_1\alpha \left\|w_k^m\right\|_{H_k^m} \\ &\leq -c_1\varepsilon_k \left\|w_k^m\right\|_{H_k^m}. \end{aligned} \tag{33}$$

On the other hand, $\left\|w_k^m\right\|_{H_k^m} > \delta_k$, thus $\lim_{m\to\infty} f(x_k^m) = -\infty$. This is in contradiction with the fact that $f(x_k^m) \geq f^* > -\infty$. Therefore, for each k, there is $m_k > 0$ such that $\left\|w_k^{m_k}\right\|_{H_k^{m_k}} \leq \delta_k$. Hence, after finitely many iterations, $x_{k+1} = x_k^{m_k}$ and we have

$$\min\left\{\|v\|_{H_k^{m_k}} : v \in \operatorname{conv} W_k^{m_k}\right\} \leq \delta_k. \tag{34}$$

Also, in the iteration m_k, parameters are updated. Thus, we have $\delta_{k+1} = \delta_k \times \theta_\delta$ and $\varepsilon_{k+1} = \varepsilon_k \times \theta_\varepsilon$. Since $\theta_\delta, \theta_\varepsilon \in (0, 1)$, then δ_k and ε_k converge to 0, when $k \to \infty$. Since $\{x_k\} \subseteq M$ and M is bounded, then $\{x_k\}$ has an accumulation point, namely x^* and there exists a subsequence $\left\{x_{k_i}\right\}$ such that $x_{k_i} \to x^*$ as $k_i \to \infty$. On the other hand, we have,

$$\operatorname{conv} W_{k_i}^{m_{k_i}} \subseteq \partial_{\varepsilon_{k_i}} f(x_{k_i}^{m_{k_i}}). \tag{35}$$

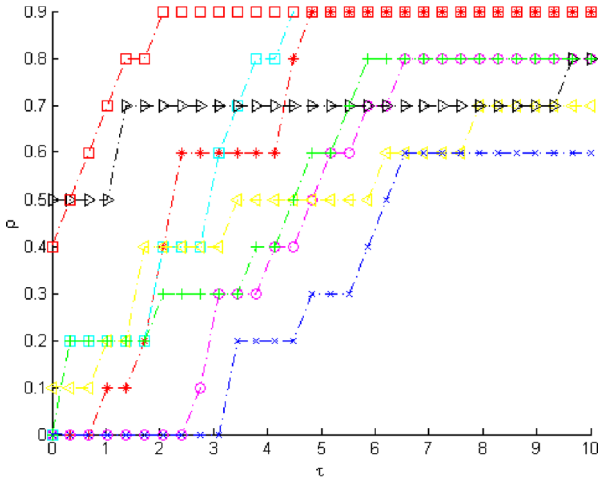
Now, by (34) and (35), we have,

$$\left\|w_{k_i}^*\right\|_{H_{k_i}^{m_{k_i}}} = \min\left\{\|v\|_{H_{k_i}^{m_{k_i}}} : v \in \partial_{\varepsilon_{k_i}} f(x_{k_i})\right\} \leq \delta_{k_i}.$$

Since $\delta_{k_i} \to 0$ as $k_i \to \infty$, then we have $\lim_{k_i\to\infty} \left\|w_{k_i}^*\right\|_{H_{k_i}^{m_{k_i}}} = 0$. By Remark 4.2, we have $\left\|w_{k_i}^*\right\|_{H_{k_i}^{m_{k_i}}} \geq \sigma \left\|w_{k_i}^*\right\|^2$. Therefore $\lim_{k_i\to\infty} \left\|w_{k_i}^*\right\| = 0$. Thus, $\lim_{k_i\to\infty} w_{k_i}^* = 0$. On the other hand, there exists $y_{k_i} \in B(x_{k_i}, \varepsilon_{k_i})$ such that $w_{k_i}^* \in \partial f(y_{k_i})$. Since $\partial f(\cdot)$ is upper semicontinuous and $y_{k_i} \to x^*$, then $0 \in \partial f(x^*)$ and this completes the proof. □

5 Numerical experiments

In this section, we implement Algorithm 5, denoted by "NBFGS", and compare the results with some other nonsmooth optimization algorithms. All the algorithms are



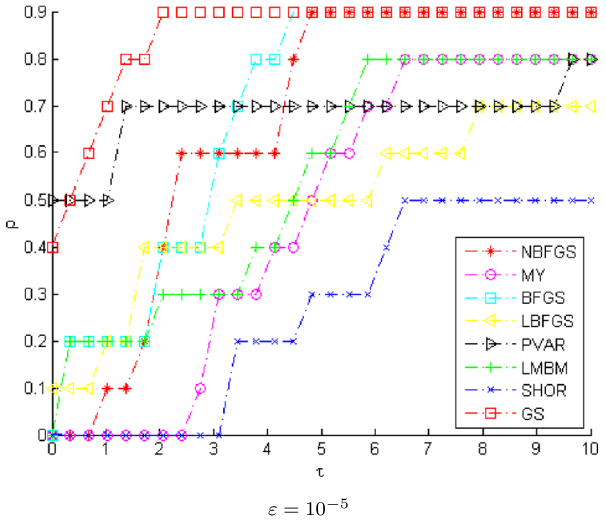
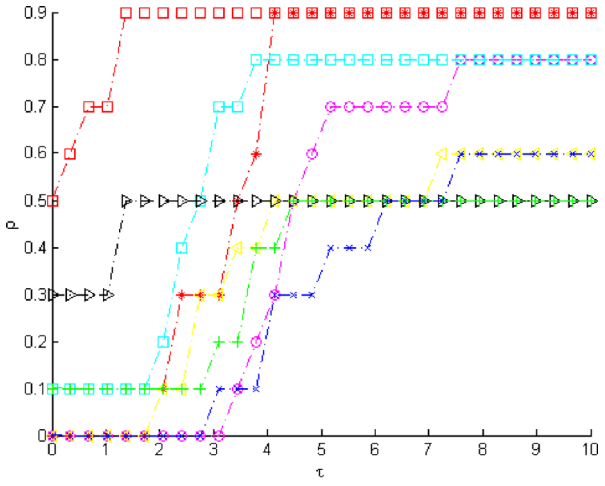
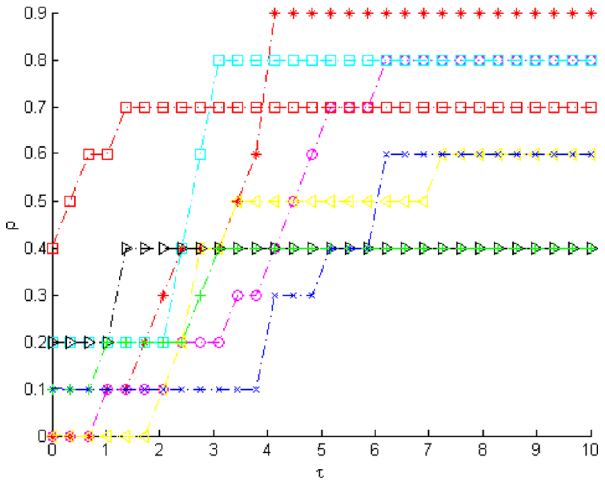


Fig. 1 Performance profiles for tested algorithms in dimensions $n = 10$, for first classes of problems

being implemented in MATLAB R2007b. The number of subgradient evaluations is considered as the measure of an algorithm efficiency. Also, we take the advantages of the performance profile of Dolan and More in [48] to have a better comparison between the implemented algorithms. Two classes of test functions are used to measure the efficiency of the considered algorithms. The first class of test problems is taken from [29] and the second one is the TEST29 taken from [47]. In the numerical experiments, we see that the second class of problems is harder than the first class.



$\varepsilon = 10^{-4}$



$\varepsilon = 10^{-5}$

Fig. 2 Performance profiles for tested algorithms in dimensions $n = 10$, for second classes of problems

Thus, we compare the performance of algorithms for each class of problems separately. The test problems are introduced in Table 1. 10 randomly generated starting points were used for each test problem and we report the number of successful runs of each algorithm.

In the smooth line search, c_1 and c_2 are initialized with 10^{-4} and 0.9. Here, we also set these parameters with the same values. The algorithm is tested with different

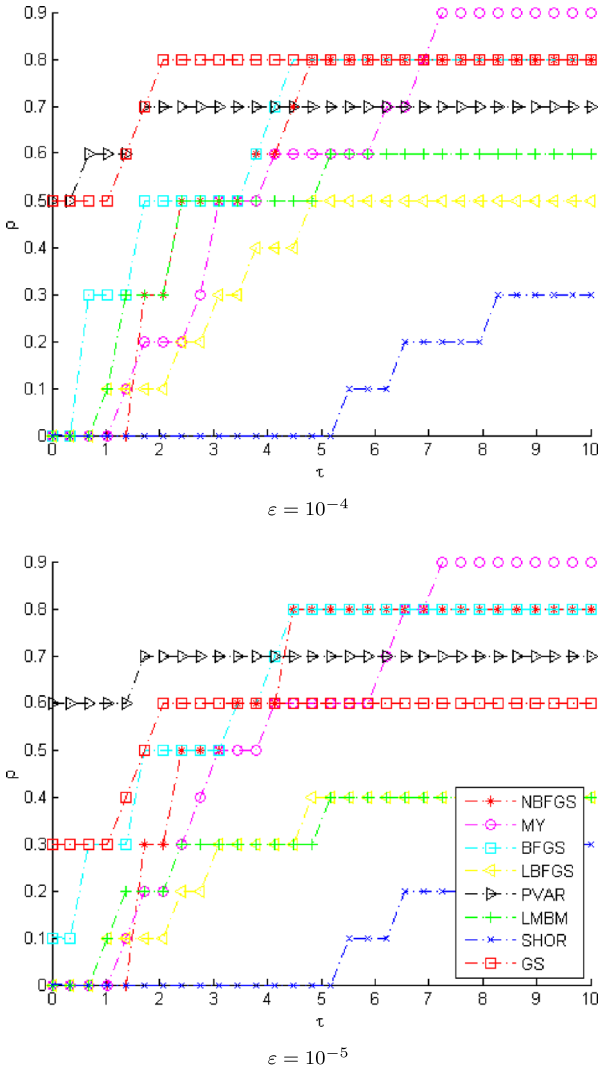


Fig. 3 Performance profiles for tested algorithms in dimensions $n = 100$, for first classes of problems in the same column

values of parameters and the values are chosen that give the best results for the all test problems. We set $\varepsilon = 10^{-6}$, $\theta_\varepsilon = .1$, $\delta_1 = 10^{-6}$, $\theta_\delta = 1$, and $\sigma = 10^{-12}$. If any the following condition satisfies,

- $\varepsilon_k <= 10^{-15}$,
- $\left\| w_k^m \right\| \leq 10^{-6}$,
- the number of the function evaluation exceeds 1000000,

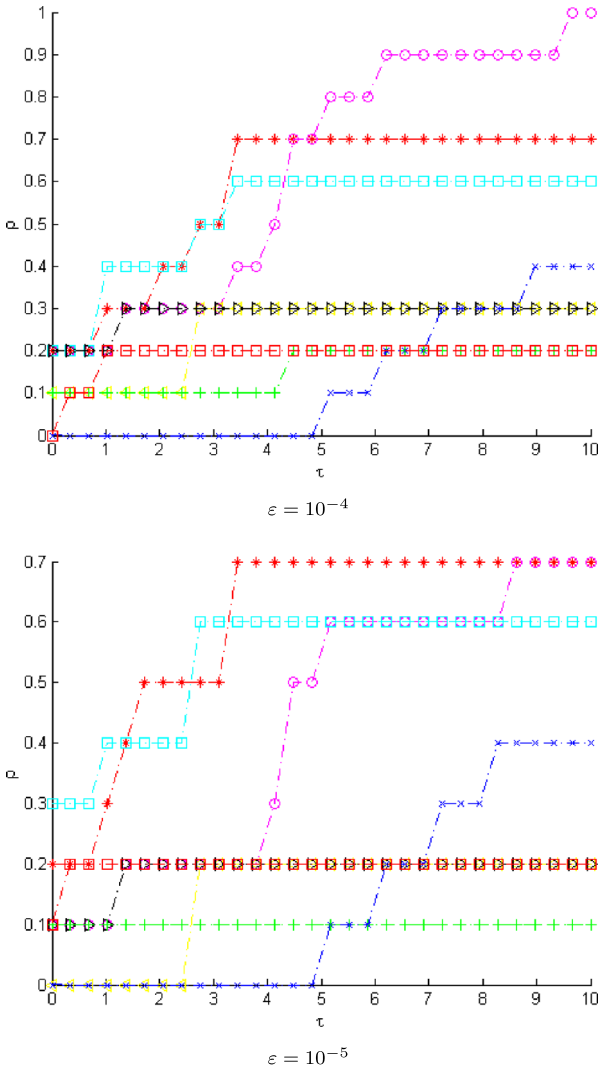


Fig. 4 Performance profiles for tested algorithms in dimensions $n = 100$, for second classes of problems in the same column

then Algorithm 5 terminates.

We compare the presented algorithm with the variable metric bundle method (PVAR) [27, 28], Limited-Memory Bundle Method (LMBM) [29], MY method [36], gradient sampling method (GS) [15], Shor-R algorithm [1], smooth BFGS method and Limited-Memory BFGS method (LBFGS) [38].

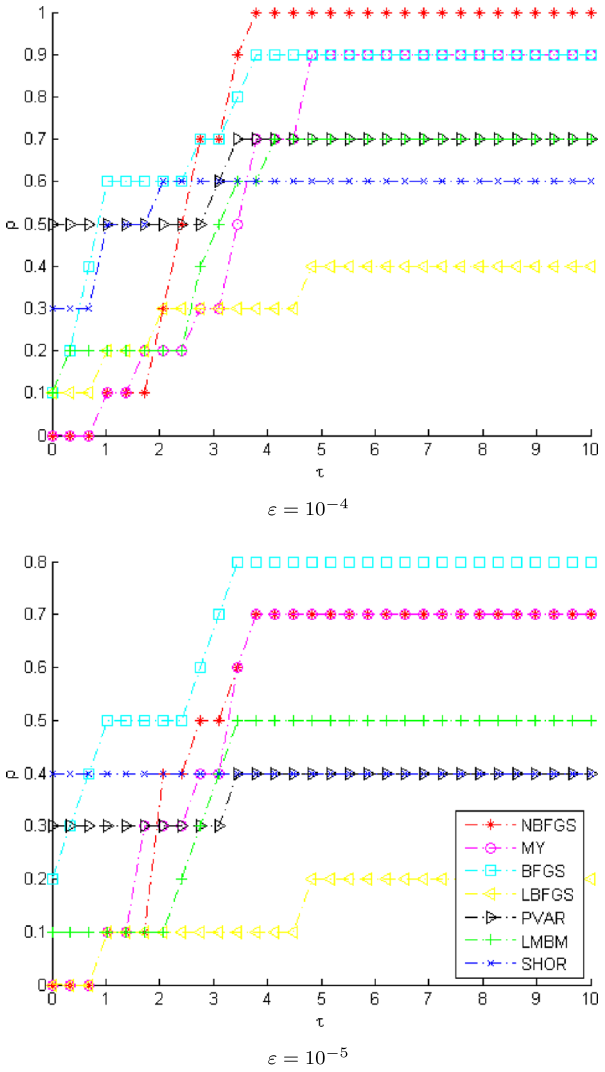


Fig. 5 Performance profiles for tested algorithms in dimensions $n = 1000$, for first classes of problems in the same column

In the performance profile, we say an algorithm is successfully solve a problem, if the following inequality satisfies,

$$\frac{|f_{\min} - f_*|}{|f_{\min}| + 1} \leq \epsilon,$$

where, $f_{\min}$ is the global minimizing value and Algorithm 3 returns f_* as approximation of $f_{\min}$. We use three tolerances $\epsilon = 10^{-4}, 10^{-5}$ and the number of

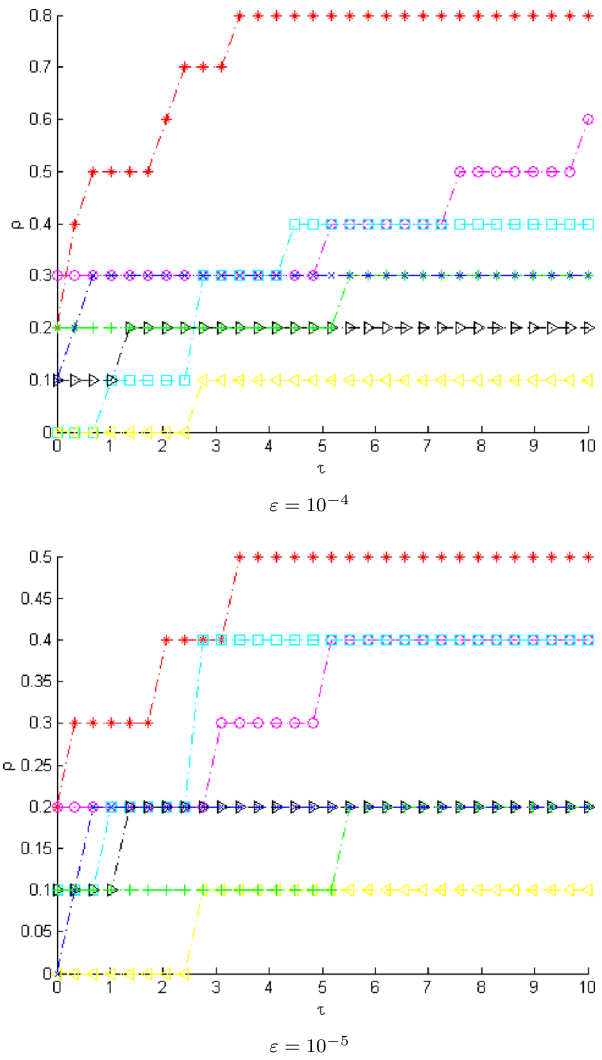


Fig. 6 Performance profiles for tested algorithms in dimensions $n = 1000$, for second classes of problems in the same column

subgradient evaluations as a performance measure. Since the number of function evaluations is equal to the number of subgradient evaluations in the all methods, then we use the number of subgradient evaluations and show the results in Figs. 1, 2, 3, 4, 5 and 6. In these figures, each row shows the different optimality thresholds, $\varepsilon = 10^{-3}$, 10^{-4} and 10^{-5} for the first, second and third rows respectively and

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

No.	problem	convex	optimal value
First class of problems			
1	MAXQ	+	0
2	MAXHILB	+	0
3	LQ	+	−1.41279e+003
4	CB3I	+	1998
5	CB3II	+	1998
6	NACTFACES	−	0
7	Brown 2	−	0
8	Mifflin 2	−	−7.06503e+002
9	Crescent I	−	0
10	Crescent II	-	0
Second class of problems			
11	problem 2 from `TEST29`	+	0
12	problem 5 from `TEST29`	+	0
13	problem 6 from `TEST29`	+	0
14	problem 11 from `TEST29`	+	1.20312e+004
15	problem 13 from `TEST29`	+	5.66131e+002
16	problem 17from `TEST29`	−	0
17	problem 19 from `TEST29`	−	0
18	problem 20 from `TEST29`	−	0
19	problem 22 from `TEST29`	−	0
20	problem 24 from `TEST29`	−	0

each column shows performance profiles for the first and second class of problems respectively. We set $\epsilon = 10^{-4}$ and run each algorithm for 10 fixed random starting points for dimension 10, 50 and 100. For each algorithm, the number of number of unsuccessful runs are reported in Table 2 for each problem and the average number of subgradient evaluations for each successful implementation are reported in Table 3.

For small dimensions, the numerical experiments show that all of the tested algorithm have an acceptable behavior. But, some of these algorithms are not efficient for high-dimensional problems. Specially, this is more evident for nonconvex problems. The numerical results shown that the NBFGS and BFGS methods have the same behavior for some test problems and the NBFGS method has better efficiency in nonconvex problems. The computed search direction is the main reason for the

Table 2 Each column of table shows the number of unsuccessful runs for each algorithm in dimension 10, 50 and 100 respectively. The starting points are fixed for all algorithms

prob	NBFGS	MY	BFGS	LBFGS	PVAR	LMBM	SHOR	GS
1	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	2,10,10	0,0,0
2	0,0,0	0,1,5	0,0,0	10,10,10	0,1,0	4,3,6	10,10,10	0,0,0
3	0,0,0	0,2,3	0,0,0	7,7,3	0,0,0	0,8,3	10,10,10	0,1,6
4	0,0,0	0,3,3	0,0,0	0,8,6	0,4,5	1,8,7	9,10,10	0,2,5
5	5,5,4	0,0,0	7,5,6	4,7,5	6,10,10	2,2,3	10,10,10	0,0,0
6	0,0,0	0,0,0	0,0,0	0,0,0	4,10,10	0,0,6	10,10,10	0,4,8
7	0,0,0	0,1,1	0,0,0	0,3,5	0,7,6	0,8,7	1,10,10	0,0,1
8	5,0,0	2,2,2	10,10,0	9,7,10	10,10,0	10,7,10	10,10,10	10,10,0
9	0,0,0	0,0,0	0,0,0	0,0,0	8,5,7	0,0,0	10,10,10	0,0,0
10	0,0,0	0,0,0	0,0,0	10,10,10	0,8,6	5,10,10	10,10,10	0,4,7
11	0,0,0	0,0,0	0,0,0	0,0,0	0,1,0	10,10,10	10,10,10	0,6,8
12	0,0,0	2,6,6	0,0,0	10,10,10	6,3,3	6,6,10	10,10,10	0,0,0
13	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	2,10,10	0,0,0
14	10,10,10	10,10,10	10,10,10	10,10,10	10,10,10	10,10,10	10,10,10	10,10,10
15	5,10,10	4,10,10	5,10,10	5,10,10	7,10,10	4,10,10	9,10,10	3,10,10
16	0,3,4	0,0,3	0,3,4	0,9,8	5,10,10	0,7,10	5,10,10	1,3,2
17	10,10,10	10,10,10	10,10,10	10,10,10	10,10,10	10,10,10	10,10,10	10,10,10
18	5,10,10	4,10,10	5,10,10	10,10,10	10,10,10	8,10,10	10,10,10	7,10,10
19	0,0,0	0,10,10	0,0,0	10,10,10	4,9,10	10,10,10	10,10,10	0,0,0
20	0,0,0	0,10,10	0,0,0	0,10,10	10,10,10	5,10,10	10,10,10	0,0,0

behavior of these methods. In the NBFGS, the serach direction is a descent direction depending on positive definite matrix. This is the main reason that NBFGS gives better results for nonconvex problems. Since the available Matlab code for the GS algorithm does not return the number of subgradient evaluations, then we do not report the subgradient evaluations of this algorithm.

By increasing the optimality threshold, a solver can successfully solve a problem, when it approximates the optimal solution accurately. In dimension $n = 10$, the performance profiles of the NBFGS, MY, BFGS and GS methods are fixed and do not vary. Thus these methods approximate the optimal value with enough currency. But, in the higher dimensions, the performance of each solver is changed. The performance profiles of several solvers show that the performance of NBFGS method has the least variation than other methods. Also, the NBFGS has the least number of unsuccessful runs, when starting points are randomly generated. In some test functions, the number of subgradient evaluations of the BFGS method is less than the NBFGS methods. Computing cost for the descent direction is its main result.

Table 3 Each column of table shows the average number of subgradient evaluations for each algorithm in dimension 10, 50 and 100 respectively

	NBFGS			MY			BFGS			LBFGS			PVAR			LMBM			SHOR		
1	565.5	2121.3	3733.2	703.6	3520.1	6902.3	657.2	3298.1	6219.4	221.4	1597.2	3697.3	55.7	182.6	287.2	61.3	330.1	662.2	1000.0	6980.0	13617.0
2	2224.5	4164.8	4135.8	2244.8	2949.1	3544.2	1037.6	1863.1	2309.8	–	–	–	45.3	60.9	63.5	520.8	995.7	908.3	3858.0	9943.0	11066.0
3	2040.6	14282.4	7542.8	10929.5	77107.4	100064.7	577.8	2761.5	4659.5	1127.7	5912.3	2635.3	78.2	272.3	500.0	2016.4	2870.5	2434.4	3138.0	5318.0	906.0
4	2662.3	6056.9	8428.0	3702.7	46519.6	8382.3	630.0	2813.9	5244.4	10739.6	14813.0	5135.3	145.4	751.0	1752.2	3321.3	644.5	1462.7	1288.0	4333.0	4480.0
5	626.0	989.0	573.7	6196.2	11612.4	19904.8	268.0	241.0	247.0	335.5	436.3	404.8	116.8	–	–	488.4	771.4	624.0	1225.0	8602.0	9490.0
6	1272.0	4706.4	7651.5	3204.9	17459.0	35286.0	592.6	2499.6	4686.5	549.9	2915.7	6849.4	101.0	–	–	993.5	3328.2	5903.3	1015.0	4571.0	361.0
7	2661.5	8801.4	12340.3	2636.7	4402.1	8343.7	637.5	2859.2	5543.4	12805.0	8052.0	6829.8	102.6	477.7	709.3	2468.5	832.5	604.3	1021.0	7840.0	10800.0
8	116.6	504.0	17747.4	2946.9	12311.1	90917.8	–	–	4961.9	837.0	5402.3	–	–	–	585.3	–	1416.3	–	0.0	0.0	9230.0
9	336.2	380.4	327.1	1032.7	1166.4	1452.7	200.8	187.7	183.7	350.0	339.1	366.0	72.0	105.4	86.0	534.9	440.8	563.7	1151.0	7467.0	11994.0
10	2521.9	9460.8	14212.9	2573.0	3094.9	3376.1	680.5	3063.2	5392.3	–	–	–	101.8	342.0	635.5	4310.8	–	–	3577.0	4443.0	2507.0
11	2938.6	5864.9	11117.2	2756.6	13969.4	27732.5	628.1	2901.5	5610.3	10051.0	17932.3	17629.9	64.4	244.4	402.4	–	–	–	1558.0	2438.0	632.0
12	2123.9	4060.9	5780.4	13649.5	26026.5	27843.0	840.2	1597.0	1946.6	–	–	–	57.0	76.6	88.0	943.8	1635.3	–	1608.0	6215.0	12232.0
13	103.2	105.2	94.5	305.4	279.7	317.6	73.3	75.2	71.1	69.3	73.7	67.9	22.3	37.5	90.4	49.7	36.3	42.0	1514.0	6129.0	10388.0
14	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
15	1257.2	–	–	1665.5	–	–	293.2	–	–	361.4	–	–	134.7	–	–	1283.3	–	–	1084.0	–	–
16	1011.5	4561.1	7840.8	2833.7	18826.9	35975.4	576.9	2666.7	4917.5	464.9	5327.0	8707.0	84.2	–	–	624.3	5637.0	–	1464.0	4285.0	8564.0
17	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
18	4997.0	–	–	33536.3	–	–	3955.4	–	–	–	–	–	–	–	–	3207.0	–	–	505.0	–	–
19	2132.4	9871.0	21635.1	24444.7	–	–	599.2	2706.0	5187.5	–	–	–	79.0	297.0	–	–	–	–	1492.0	4756.0	10525.0
20	2396.7	13472.6	29359.1	10208.6	–	–	814.7	4178.9	10036.8	2342.9	–	–	–	–	–	5257.2	–	–	1616.0	5611.0	9739.0

The starting points are fixed for all algorithms

Table 4 Test results, the number of subgradient evaluations and final value of function with $n = 1000$

	NBFGS		MY		BFGS		PVAR		LMBM		SHOR	
No.	f_*	nfeval	f_*	nfeval	f_*	nfeval	f_*	nfeval	f_*	nfeval	f_*	nfeval
1	1.118e−07	36406	8.231e−10	9886	5.396e−08	24093	7.430e+02	17439	8.363e−06	3708	4.988e−06	22808
2	4.048e−05	696	3.426e−05	3614	3.683e−09	1909	4.538e−02	721	3.685e−05	143	6.166e−03	861
3	−1.413e+03	20147	−1.413e+03	12240	-1.413e+03	14074	-1.413e+03	6537	-1.413e+03	1569	-1.413e+03	1824
4	1.998e+03	10531	1.998e+03	3061	1.998e+03	1893	1.998e+03	2142	1.998e+04	4	1.998e+03	1788
5	1.998e+03	334	1.998e+03	694	1.998e+03	68	2.001e+03	85	1.998e+03	65	1.998e+03	1125
6	1.481e−09	107	7.033e−11	382	1.221e−11	59	0.000e+00	107	2.516e−06	530	1.377e−14	569
7	5.075e−08	398	1.463e−09	1423	6.563e−07	194	1.774e−20	2367	1.927e+03	68	9.132e−04	1636
8	−7.065e+02	5439	−7.065e+02	14189	−7.065e+02	1085	−7.065e+02	597	-7.065e+02	4115	-7.065e+02	3892
9	−2.665e−15	452	6.895e−10	1149	5.551e−15	154	2.777e−01	185	1.968e−06	120	3.681e−08	817
10	8.203e−08	1940	1.203e−09	4678	3.839e−06	522	7.427e−02	5717	1.172e+00	40	1.369e−04	9626
11	3.757e−07	28293	3.290e−09	26478	2.292e−14	48134	2.217e−02	17579	5.703e−04	1644	9.815e-01	63
12	6.069e−06	391	3.001e−05	100018	2.679e−10	682	1.642e−03	2797	6.292e−06	106	6.434e−06	4563
13	6.661e−16	147	1.301e−11	488	6.661e−16	88	2.220e−16	90	4.816e−08	32	1.943e−16	14
14	1.203e+04	15180	1.203e+04	13688	1.204e+04	1399	1.203e+04	5678	1.622e+04	24	1.204e+04	1336
15	5.661e+02	39891	5.684e+02	100073	6.218e+02	29426	5.690e+02	7477	3.753e+03	912	5.661e+02	9429
16	2.240e−03	4551	3.046e−09	21368	2.388e−03	40	2.388e−03	40	1.675e−03	1390	2.258e−03	9262
17	2.894e−10	27602	4.564e−03	100033	1.000e+00	1099	1.000e+00	458	1.495e−04	3607	1.000e+00	872
18	1.195e 07	31223	5.005e−01	8571	5.000e−01	1030	5.000e−01	1296	5.027e−01	135	5.000e−01	3087
19	4.618e−05	873	6.437e−06	100009	1.537e−06	13286	1.119e−03	391	4.795e−04	565	8.288e−04	433
20	1.099e−01	2806	7.653e−02	21896	1.099e−01	592	1.099e-01	691	1.106e-01	123	1.359e-01	2034

6 Conclusions

The numerical results shown that the smooth BFGS and LBFGS method have good behavior to solve nonsmooth optimization problems such that they give better solutions than the famous nonsmooth optimization algorithms. By combination of the BFGS and a descent method, the efficiency of the BFGS method is increased especially for high dimensional problems. In the combination method, the number of subgradient evolutions are more than the BFGS method in some cases. In fact, computing the descent direction increases the number of subgradient evolutions. In this paper, a descent direction is computed for locally Lipschitz function. If the descent direction computing algorithm is modified based on a special functions, for example convex functions, semismooth functions and ..., we except that the number of subgradient evolutions are reduced significantly. In future works, we try to modify the NBFGS method for special functions such that its efficiency is increased. Also, by combination of the LBFGS and a descent method, an efficient method will be construct for large scale problems.

Acknowledgments The author is grateful to three anonymous referees for their valuable comments and suggestions that improved the presentation of the paper.

References

1. Shor, N.Z.: Minimization Methods for Non-differentiable Functions. Springer, Berlin (1985)
2. Frangioni, A.: Generalized bundle methods. SIAM J. Optim. **13**(1), 117–156 (2002)
3. Fuduli, A., Gaudioso, M., Giallombardo, G.: Minimizing nonconvex nonsmooth functions via cutting planes and proximity control, vol. 14, pp. 743–756 (2003). (electronic)
4. Fuduli, A., Gaudioso, M., Giallombardo, G.: A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. Optimization Methods & Software **19**(1), 89–102 (2004)
5. Gaudioso, M., Gorgone, E., Monaco, M.F.: Piecewise linear approximations in nonconvex nonsmooth optimization. Numer. Math. **113**(1), 73–88 (2009)
6. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms. Springer, Berlin (1993)
7. Makela, M.M., Neittaanmaki, P.: Nonsmooth Optimization. World Scientific (1992)
8. Kiwiel, K.: Methods of Descent for Nondifferentiable Optimization. Lecture Notes in Mathematics 1133. Springer, Berlin (1985)
9. Mifflin, R.: An algorithm for constrained optimization with semismooth functions. Math. Oper. Res. **2**(2), 191–207 (1977)
10. Wolfe, P.H.: A method of conjugate subgradients of minimizing nondifferentiable convex functions. Math. Program. Study **3**, 145–173 (1975)
11. Mayne, D.Q., Polak, E.: Nondifferential optimization via adaptive smoothing. J. Optim. Theory Appl. **43**, 19–30 (1984)
12. Polak, E., Royset, J.O.: Algorithms for finite and semi-infinite min-max-min problems using adaptive smoothing techniques. J. Optim. Theory Appl. **119**(3), 421–457 (2003)
13. Bagirov, A.M.: Continuous subdifferential approximations and their applications. J. Math. Sci. **115**, 2567–2609 (2003)
14. Bagirov, A.M., Karasözen, B., Sezer, M.: Discrete gradient method: derivative-free method for nonsmooth optimization. J. Optim. Theory Appl. **137**(2), 317–334 (2008)
15. Burke, J.V., Lewis, A.S., Overton, M.L.: A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM J. Optim. **15**, 571–779 (2005)

16. Kiwiel, K.C.: Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization. SIAM J. Optim. **18**(2), 379–388 (2007)
17. Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.: A family of variable metric proximal methods. Math. Program. **68**, 15–47 (1995)
18. Burke, J., Qian, M.: On the superlinear convergence of the variable metric proximal point algorithm using broyden and bfgs matrix secant updating. Math. Program. **88**, 157–181 (1997)
19. Chen, X., Fukushima, M.: Proximal quasi-newton methods for nondifferentiable convex optimization. Math. Program. **88**, 313–334 (1999)
20. Fukushima, M., Qi, L.: A globally and superlinearly convergent algorithm for nonsmooth convex minimization. SIAM J. Optim. **6**, 1106–1120 (1996)
21. Hare, W., Sagastizábal, C.: Computing proximal points of nonconvex functions. Math. Program. **116**, 221–258 (2009)
22. Lemaréchal, C., Sagastizábal, C.: Variable metric bundle methods: From conceptual to implementable forms. Math. Program. **76**, 393–410 (1996)
23. Mifflin, R.: A quasi-second-order proximal bundle algorithm. Math. Program. **73**, 51–72 (1996)
24. Mifflin, R., Sun, D., Qi, L.: Quasi-newton bundle-type methods for nondifferentiable convex optimization. SIAM J. Optim. **8**, 583–603 (1998)
25. Zhu, C.: Asymptotic convergence analysis of some inexact proximal point algorithms for minimization. SIAM J. Optim. **6**, 626–637 (1996)
26. Luksan, L., Vlcek, J.: A bundle-newton method for nonsmooth unconstrained minimization. Math. Program. **83**, 373–391 (1998)
27. Luksan, L., Vlcek, J.: Globally convergent variable metric method for convex nonsmooth unconstrained minimization. J. Optim. Theory Appl. **102**, 593–613 (1999)
28. Luksan, L., Vlcek, J.: Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization. J. Optim. Theory Appl. **111**, 407–430 (2001)
29. Haarala, N., Miettinen, K., Mäkelä, M.M.: Globally convergent limited memory bundle method for large-scale nonsmooth optimization. Math. Program. **109**, 181–205 (2007)
30. Karmitsa, N., Mäkelä, M.M.: Limited memory bundle method for large bound constrained nonsmooth optimization: convergence analysis. Optimization Methods Software **25**, 895–916 (2010)
31. Lewis, A., Overton, M.: Behavior of bfgs with an exact line search on nonsmooth examples. Tech. rep (2008). http://www.cs.nyu.edu/overton/papers/pdffiles/bfgs_exactLS.pdf
32. Lewis, A., Overton, M.: Nonsmooth optimization via bfgs. Tech. rep. http://www.cs.nyu.edu/overton/papers/pdffiles/bfgs_inexactLS.pdf (2008)
33. Lewis, A., Overton, M.: Nonsmooth optimization via quasi-newton methods. Math. Program., 1–29 (2012). doi:10.1007/s10107-012-0514-2
34. Bertsekas, D.P., Mitter, S.K.: A descent numerical method for optimization problems with nondifferentiable cost functionals. SIAM J. Control. **11**, 637–652 (1973)
35. Goldstein, A.A.: Optimization of Lipschitz continuous functions. Math. Program. **13**, 14–22 (1977)
36. Mahdavi-Amiri, N., Yousefpour, R.: An effective nonsmooth optimization algorithm for locally Lipschitz functions. J. Optim. Theory Appl. **155**, 180–195 (2012)
37. Wolfe, P.: A method of conjugate subgradients for minimizing non-differentiable functions. Nondifferentiable Optimization. In: Balinski, M., Wolfe, P. (eds.), vol. 3, pp. 145–173. Mathematical Programming Study, North-Holland (1975)
38. Nocedal, J., Wright, S.J.: Numerical optimization, 2nd edn. Springer Series in Operations Research and Financial Engineering. Springer, New York (2006)
39. Clarke, F.H.: Optimization and Nonsmooth Analysis. Society for Industrial and Applied Mathematics (1987)
40. Yu, J., Vishwanathan, S., Günter, S., Schraudolph, N.N.: A quasi-newton approach to nonsmooth convex optimization problems in machine learning. J. Mach. Learn. Res. **11**, 1145–1200 (2010)
41. More, J.J., Thuente, D.J., Mcs-p, P.: Line search algorithms with guaranteed sufficient decrease. ACM Trans. Math. Software **20**, 286–307 (1992)
42. Al-Baali, M., Fletcher, R.: An efficient line search for nonlinear least squares. J. Optim. Theory Appl. **48**(3), 359–377 (1986)
43. Daugavet, V.A.: Modification of the Wolfe method. Zh. Vychisl. Mat. Mat. Fiz. **21**, 504–508 (1981)
44. Mitchell, B.F., Demyanov, V.F., Malozemov, V.N.: Finding the point of a polyhedron closest to the origin. SIAM J. Control. Optim. **12**, 19–26 (1974)
45. Wolfe, P.: Finding the nearest point in a polytope. Math. Program. **11**, 128–149 (1976)

46. Golub, G.H., Van Loan, C.F.: Matrix Computations (3rd Edn.) Johns Hopkins University Press (1996)
47. Luksan, L., Tuma, M., Siska, M., Vlcek, J., Ramesova, N.: Ufo 2002. interactive system for universal functional optimization. Tech. rep., Academy of Sciences of the Czech Republic (2002)
48. Dolan, E., More, J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)