Journal of Mathematical Extension
Vol. 11, No. 3, (2017), 31-42
ISSN: 1735-8299
URL: http://www.ijmex.com

A Subgradient Method for Unconstrained Nonconvex Nonsmooth Optimization

M. Maleknia
University of Isfahan

Abstract. We propose an iterative method that solves an unconstrained nonconvex nonsmooth optimization problem. The proposed method is a descent method that uses subgradients at each iteration and contains very simple procedures for finding descent directions and for solving line search subproblems. The convergence of the algorithms is studied.

AMS Subject Classification: 65K05; 90C30; 90C26
Keywords and Phrases: Nonsmooth optimization, Subgradient method.

1. Introduction

In this paper, we study an unconstrained nonsmooth minimization problem with locally Lipschitz and directionally differentiable objective function. There have been proposed various methods for solving these problems when the objective function is continuously differentiable (see, for example,[5, 8, 14, 16]). The subgradient method is one of the effective methods for solving such problems. It was originally developed by N.shor and then was modified by many authors(see[4, 13, 15] and more recent papers [1, 6, 12, 10, 11]). The subgradient method uses one subgradient and function evaluation at each iteration. It does not involve any subproblems either for finding search directions or for computation of step lengths. Therefore, it is easy to implement this method. Although the

Received: August 2016; Accepted: January 2016

subgradient method is very slow it is well known that some of its modifications might be more successful for solving large scale problems than other nonsmooth optimization methods (for some of these modifications see [10, 11]).

Bundle methods and their various modifications are known to be among the most efficient methods in nonsmooth optimization (see, [9]). These methods involve a quadratic programming subproblem to find search directions. The size of subproblem may increase significantly as the number of variables increases which makes the bundle-type methods not suitable for large scale nonsmooth optimization problems. The implementation of bundle-type methods, which require the use of quadratic programming solvers, is not as easy as the implementation of the subgradient method.

In this work, we introduce a method for solving an unconstrained nonsmooth minimization problem. The method is based on the subgradient method introduced in [2, 3]. We design a descent method which contains a simple line search procedure and the convergence of the method will be studied.

The paper is structured as follows. Section 2 provides some necessary preliminaries. Section 3 presents an algorithm for finding descent directions. The minimization method is presented in Section 4. Section 5 concludes the paper.

2. Preliminaries

We use the following notations in this paper. $\mathbb{R}^n$ is an n-dimensional Euclidean space, $\langle u, v\rangle := \sum_{i=1}^n u_i v_i$ is an inner product in $\mathbb{R}^n$ and $\|.\|$ is the associated Euclidean norm. $S_1 := \{x \in \mathbb{R}^n \ : \ \|x\| = 1\}$ is the unit sphere, $B_\varepsilon(x) := \{y \in \mathbb{R}^n \ : \ \|x - y\| < \varepsilon\}$ is the open ball centered at x with radius $\varepsilon > 0$. Furthermore, $B_\varepsilon := B_\varepsilon(0_n)$.

Let f be a function defined on $\mathbb{R}^n$. The function f is called locally Lipschitz if for any bounded subset $X \subset \mathbb{R}^n$ there exists an $L > 0$ such that

$$|f(x) - f(y)| \leqslant L\|x - y\| \qquad \forall x, y \in X.$$

We recall that a locally Lipschitz function f is differentiable almost everywhere and that we can define for it a Clarke subdifferential [7] by

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

here $D(f)$ denotes the set where f is differentiable, co denotes the convex hull of a set. It is shown in [7] that the mapping $\partial f(x)$ is upper semicontinuous and bounded on bounded sets. The generalized directional derivative of f at x in the direction d is defined as

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

If the function f is locally Lipschitz the generalized directional derivative exists and

$$f^\circ(x,d) = \max\left\{ \langle v, d \rangle \ : \ v \in \partial f(x) \right\}.$$

A function f is called a Clarke regular on $\mathbb{R}^n$, if it is differentiable with respect to any direction $d \in \mathbb{R}^n$ and $f'(x,d) = f^\circ(x,d)$ for all $x, d \in \mathbb{R}^n$ where $f'(x,d)$ is a derivative of the function f at the point x with respect to the direction d :

$$f'(x,d) = \lim_{\alpha\to +0} \alpha^{-1} \left[f(x+\alpha d) - f(x) \right].$$

It is clear that directional derivative $f'(x,d)$ of the Clarke regular function f is upper semicontinuous with respect to x for all $d \in \mathbb{R}^n$. Let f be a locally Lipschitz function defined on $\mathbb{R}^n$. For point x to be a minimum point of the function f on $\mathbb{R}^n$, it is necessary that

$$0_n \in \partial f(x).$$

3. Computation of a Descent Direction

Consider the following problem

$$\min \ h(y) \qquad \text{s.t} \qquad y \in \mathbb{R}^n \tag{1}$$

Where h is a locally Lipschitz and directionally differentiable function. In this section we design an algorithm for finding descent directions of the objective function. Let $y \in \mathbb{R}^n$ be a given point, $c_1 \in (0,1)$ and $\delta > 0$ be given numbers.

Algorithm 3.1.[Computation of the descent direction]

Step 1. Select any direction $d_1 \in S_1$ and compute a subgradient $v_1 \in \partial h(y)$ such that

$$h'(y, d_1) = \langle v_1, d_1 \rangle.$$

Set $\widetilde{v_1} := v_1$ and $k := 1$.

Step 2. Solve the following problem:

$$\text{minimize } \phi_k(\lambda) := \|\lambda v_k + (1-\lambda)\widetilde{v_k}\|^2 \qquad \text{s.t} \qquad \lambda \in [0,1]. \tag{2}$$

Let $\bar{\lambda_k}$ be a solution to this problem. Set

$$\bar{v_k} := \bar{\lambda_k} v_k + (1 - \bar{\lambda_k})\widetilde{v_k}.$$

Step 3. (Stopping criterion) If

$$\|\bar{v_k}\| < \delta, \tag{3}$$

then stop. Otherwise go to Step 4.

Step 4. Compute the search direction by $\bar{d_k} = -\|\bar{v_k}\|^{-1}\bar{v_k}$.

Step 5. If

$$h'(y, \bar{d_k}) \leqslant -c_1\|\bar{v_k}\|, \tag{4}$$

then stop. Otherwise go to step 6.

Step 6. Compute a subgradient $u \in \partial h(y)$ such that

$$h'(y, \bar{d_k}) = \langle u, \bar{d_k} \rangle$$

Set $v_{k+1} := u, \widetilde{v}_{k+1} := \bar{v_k}, k := k+1$ and go to step 2. It is worth to mention that at the kth iteration $(k > 1)$ of the algorithm we use the aggregate subgradient $\widetilde{v}_k$ computed at the previous $(k-1)$th iteration. This aggregate subgradient gives the least distance between the segment

$$S_{k-1} = \{v \in \mathbb{R}^n \ : \ v = \lambda v_{k-1} + (1-\lambda)\widetilde{v}_{k-1}, \, \lambda \in [0,1]\},$$

and the origin at the $(k-1)$th iteration and the distance can be calculated explicitly.

In the next theorem we prove that Algorithm (3.1) is finitely convergent.

Theorem 3.2. *Suppose that h is a locally Lipschitz and directionally differentiable function and at $y \in \mathbb{R}^n$*

$$C := \max\{\|v\| \ : \ v \in \partial h(y)\} < +\infty. \tag{5}$$

If $c_1 \in (0,1)$ and $\delta \in (0, C)$, then Algorithm (3.1) stops after $m > 0$ iterations, where

$$m \leqslant 2\log_2^{(\delta/C)} / \log_2^{C_1} + 1, \qquad C_1 = 1 - \left[(1-c_1)(2C)^{-1}\delta\right]^2.$$

Proof. Algorithm (3.1) terminates when either the condition (3) or the condition (4) is met. Hence, it is sufficient to estimate the upper bound of the number of iterations m when the condition (3) occurs. If none of conditions (3) and (4) hold then new subgradient v_{k+1} computed in step 6 does not belong to the segment S_k. Since $\bar{v_k}$ is the solution to problem (2) it follows from necessary condition for a minimum that

$$\langle v, \bar{v_k}\rangle \geqslant \|\bar{v_k}\|^2, \qquad \forall v \in S_k. \tag{6}$$

On the other hand since the condition (4) does not hold, we obtain

$$h'(y, \bar{d_k}) > -c_1\|\bar{v_k}\|.$$

Since $h'(y, \bar{d_k}) = \langle v_{k+1}, \bar{d_k}\rangle$ therefore

$$\langle v_{k+1}, \bar{d_k}\rangle > -c_1\|\bar{v}_k\|.$$

Hence,

$$\langle v_{k+1}, \bar{v_k}\rangle < c_1\|\bar{v_k}\|^2. \tag{7}$$

From (6) we have $v_{k+1} \notin S_k$. Therefore, it is proved that newly calculated subgradient v_{k+1} is not on the segment between v_k and $\widetilde{v_k}$, and by updating subgradients in this manner, we change the set S_k at every iteration of Algorithm (3.1). Now we will prove that if stopping criteria do not apply then the new minimizer $\bar{v}_{k+1}$ is better than $\bar{v_k}$ in the sense that $\phi_{k+1}(\bar{\lambda}_{k+1}) < \phi_k(\bar{\lambda}_k)$.
Since ϕ_{k+1} is strictly convex function, $\bar{v}_{k+1} = \bar{\lambda}_{k+1}v_{k+1} + (1-\bar{\lambda}_{k+1})\widetilde{v}_{k+1}$ is its only minimizer. In addition, since $\widetilde{v}_{k+1} = \bar{v_k}$, we have

$$\phi_{k+1}(\bar{\lambda}_{k+1}) = \|\bar{v}_{k+1}\|^2 < \phi_{k+1}(0) = \|\bar{v_k}\|^2 = \phi_k(\bar{\lambda}_k).$$

Furthermore, we will show that at each iteration the optimal value of the function ϕ_k over the sets $S_k, k = 1, 2, \ldots$ will be decreased sufficiently. We observe that

$$\|\bar{v}_{k+1}\|^2 \leqslant \|tv_{k+1} + (1-t)\bar{v_k}\|^2,$$

for all $t \in [0, 1]$. Hence,

$$\|\bar{v}_{k+1}\|^2 \leqslant t^2\|v_{k+1} - \bar{v_k}\|^2 + 2t\langle v_{k+1} - \bar{v_k}, \bar{v_k}\rangle + \|\bar{v_k}\|^2. \tag{8}$$

It follows from (5) that

$$\|v_{k+1} - \bar{v_k}\| \leqslant 2C. \tag{9}$$

From (7), (8) and (9) we have

$$\begin{aligned}\phi_{k+1}(\bar{\lambda}_{k+1}) = \|\bar{v}_{k+1}\|^2 &< 4t^2C^2 + (1 - 2t(1-c_1))\,\|\bar{v_k}\|^2 \\ &= 4t^2C^2 + (1 - 2t(1-c_1))\,\phi_k(\bar{\lambda_k}).\end{aligned}$$

Let $t_0 = (1-c_1)\|\bar{v_k}\|^2(4C^2)^{-1}$. It is clear that $t_0 \in (0, 1)$. Therefore, for $t = t_0$ we have

$$\|\bar{v}_{k+1}\|^2 < \left[1 - (1-c_1)^2(4C^2)^{-1}\|\bar{v_k}\|^2\right]\|\bar{v_k}\|^2. \tag{10}$$

Since $\|\bar{v_k}\| > \delta$ for all $k = 1, \ldots, m-1$, it follows that

$$\|\bar{v}_{k+1}\|^2 < \left[1 - (1-c_1)^2(4C^2)^{-1}\delta^2\right]\|\bar{v_k}\|^2.$$

Let $C_1 = 1 - (1-c_1)^2(4C^2)^{-1}\delta^2$. It follows that $C_1 \in (0,1)$. Then we have

$$\phi_{k+1}(\bar{\lambda}_{k+1}) < C_1 \phi_k(\bar{\lambda}_k).$$

Since $\phi_1(\bar{\lambda_1}) = \|v_1\|^2 \leqslant C^2$ we get

$$\phi_k(\bar{\lambda_k}) < (C_1)^{k-1} C^2.$$

Therefore, the inequality (3) is satisfied if $C_1^{m-1}C^2 \leqslant \delta^2$. This inequality must happen after at most m iterations where

$$m \leqslant \frac{2\log_2^{\delta/C}}{\log_2^{C_1}} + 1.$$

This completes the proof. □

Based on Algorithm (3.1) one can find descent directions because it allows to design a very simple line search procedure.

4. The Method

In this section we will describe two algorithms for solving the problem (1). The first algorithm can find the so-called δ-stationary points of problem (1) whereas the second algorithm finds its Clarke stationary points. We start with the definition of the δ-stationary points.

Definition 4.1. *A point y is called the δ-stationary point for problem (1) iff:*

$$\min\{\|v\| \ : \ v \in \partial h(y)\} \leqslant \delta.$$

It follows from definition that if y be a δ-stationary point then

$$0_n \in \partial h(y) + B_\delta.$$

Let $c_1 \in (0,1)$, $c_2 \in (0, c_1)$ be given numbers. An algorithm for finding δ-stationary points is presented as follows.

Algorithm 4.2.[Computation of δ-stationary points]

Step 1. Select any starting point $y_0 \in \mathbb{R}^n$ and set $k := 0$.

Step 2. Apply Algorithm (3.1) for the computation of the descent direction at $y = y_k$ for given $\delta > 0$ and $c_1 \in (0,1)$. This algorithm terminates after finite number of iterations and as a result, we get the subgradient v_k, the aggregated subgradient $\widetilde{v_k}$ and an element $\bar{v_k}$ such that

$$\bar{v}_k = \bar{\lambda}_k v_k + (1 - \bar{\lambda}_k)\widetilde{v_k}$$

where

$$\bar{\lambda}_k := \operatorname{argmin}_{\lambda \in [0,1]} \|\lambda v_k + (1-\lambda)\widetilde{v_k}\|^2.$$

Furthermore, either $\|\bar{v}_k\| < \delta$ or for the search direction $\bar{d}_k = -\|\bar{v}_k\|^{-1}\bar{v}_k$,

$$h'(y_k, \bar{d}_k) \leqslant -c_1\|\bar{v}_k\|. \tag{11}$$

Step 3. If

$$||\bar{v}_k|| < \delta,$$

then stop. Otherwise go to Step 4.

Step 4. Compute $y_{k+1} = y_k + \sigma_k \bar{d}_k$, where σ_k is defined as follows:

$$\sigma_k = \operatorname{argmax}\left\{\sigma \geqslant 0 \ : \ h(y_k + \sigma \bar{d}_k) - h(y_k) \leqslant -c_2\sigma\|\bar{v}_k\|\right\}.$$

Set $k := k+1$ and go to Step 2.
In the next theorem we prove that Algorithm (4.2) is finitely convergent to the set of δ-stationary points of problem (1)

Theorem 4.3. *Suppose that function h is bounded below:*

$$h_* := \inf\{h(y) \ : \ y \in \mathbb{R}^n\} > -\infty. \tag{12}$$

Then Algorithm (4.2) terminates after finite many iterations $M > 0$ and produces the δ-stationary point y_M where

$$M \leqslant M_0 := \left\lfloor \frac{h(y_0) - h_*}{c_2 z \delta} \right\rfloor + 1. \tag{13}$$

in which z is a positive number such that $\sigma_k \geqslant z$, $k \in \mathbb{N}$.

Proof. Suppose on the contrary, that is the sequence $\{y_k\}$ generated by Algorithm (4.2) is infinite and the points y_k are not δ-stationary points for any $k = 1, 2, \ldots$. This means that

$$\min\{\|v\| \ : \ v \in \partial h(y_k)\} > \delta, \qquad \forall k = 1, 2, \ldots.$$

Then a descent direction $\bar{d}_k$ is obtained at y_k so that the condition (4) will be satisfied:

$$h'(y_k, \bar{d}_k) \leqslant -c_1\|\bar{v}_k\| < -c_2\|\bar{v}_k\|.$$

Hence there exists $z > 0$ such that for all $\alpha \in (0, z]$

$$h(y_k + \alpha\bar{d}_k) - h(y_k) < -c_2\alpha\|\bar{v}_k\|$$

It follows from the definition of σ_k that $\sigma_k \geqslant z$. Therefore, we have

$$h(y_{k+1}) - h(y_k) = h(y_k + \sigma_k\bar{d}_k) - h(y_k) < -c_2\sigma_k\|\bar{v}_k\| \leqslant -c_2 z\|\bar{v}_k\|.$$

In view of condition $\|\bar{v}_k\| > \delta$ we have

$$h(y_{k+1}) - h(y_k) < -c_2 z\delta.$$

Thus,

$$h(y_{k+1}) \leqslant h(y_0) - (k+1)c_2 z\delta.$$

So, $h(y_k) \to -\infty$ as $k \to \infty$ which contradicts (12). Clearly, the upper bound for the number of iterations M necessary to find the δ-stationary point is M_0 given by (13). □

Now we are ready to state an algorithm for finding stationary points of problem (1), that is, point y satisfying the condition $0_n \in \partial h(y)$. Assume that $\varepsilon > 0$ is a tolerance.

Algorithm 4.4. [Computation stationary point]

Step 1 Select sequence $\{\delta_j\}$ such that $\delta_j > 0$ and $\delta_j \to 0$ as $j \to \infty$. Choose any starting point $y_0 \in \mathbb{R}^n$, and set $k := 0$.

Step 2 If $\delta_k \leqslant \varepsilon$, then stop with y_k as the final solution.

Step 3 Apply Algorithm (4.2) starting from the point y_k with $\delta = \delta_k$. This algorithm finds an δ_k-stationary point y_{k+1} after finitely many iterations $M > 0$.

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

Next we state the convergence of Algorithm (4.4). For the point $y_0 \in \mathbb{R}^n$, we consider the level set

$$L(y_0) := \{y \in \mathbb{R}^n \ : \ h(y) \leqslant h(y_0)\}.$$

Theorem 4.5 *Suppose that objective function h in problem (1) is locally Lipschitz and directionally differentiable, the set $L(y_0)$ is bounded and $\varepsilon = 0$ in Algorithm (4.4). Then any accumulation point of the sequence $\{y_k\}$ generated by this algorithm is a stationary point of problem (1).*

Proof. According to theorem (4.3) the sequence of δ_k-stationary points will be generated after a finite number of iterations for all $k > 0$. Since for any $k > 0$, the point y_{k+1} is the δ_k-stationary point, it follows from the definition of the δ_k-stationary point that

$$\min\{\|v\| \ : \ v \in \partial h(y_{k+1})\} \leqslant \delta_k.$$

It is obvious that $y_k \in L(y_0)$ for all $k > 0$. The boundedness of the set $L(y_0)$ implies that the sequence $\{y_k\}$ has at least one accumulation point. Let y^* be an accumulation point and $y_{k_i} \to y^*$ as $i \to \infty$. Hence

$$\min\{\|v\| \ : \ v \in \partial h(y_{k_i})\} \leqslant \delta_{k_i - 1} \tag{14}$$

Now suppose that $\tau > 0$ is arbitrary. Since the sequence $\{\delta_j\}$ converges to 0, it follows that there exists i_1 such that $\delta_{k_i} < \tau$ for all $i > i_1$. In view of (14) we have

$$\min\{\|v\| \ : \ v \in \partial h(y_{k_i})\} \leqslant \tau, \qquad \forall i > i_1 + 1$$

Now let

$$\|v^*_{k_i}\| = \min\{\|v\| \ : \ v \in \partial h(y_{k_i})\} \leqslant \tau, \qquad \forall i > i_1 + 1$$

Then $\|v^*_{k_i}\| \leqslant \tau$ for all $i > i_1 + 1$ and without loss of generality we assume that $v^*_{k_i} \to v^*$ as $i \to \infty$. It is clear that $\|v^*\| \leqslant \tau$ and by upper semicontinuity of subdifferential mapping, we have

$$v^* \in \partial h(y^*),$$

Therefore

$$\min\{\|v\| \ : \ v \in \partial h(y^*)\} \leqslant \tau.$$

Since τ is arbitrary $0_n \in \partial h(y^*)$. This completes the proof. □

5. conclusion

In this paper we have proposed an algorithm for solving unconstrained nonsmooth optimization problems. This algorithm can be applied to a broad class of nonsmooth optimization problems including problems with non regular objective functions. It contains simple procedure for finding descent directions and step lengths.

References

[1] K. M. Anstreicher and L. A. Wolsey, Two well-known properties of subgradient optimization, *Math. Program*, 120 (2009), 213-220.

[2] A. M. Bagirov and A. N. Ganjehlou, A quasisecant method for minimizing nonsmooth functions, *Optim. Methods Softw.*, 25 (2010), 3-18.

[3] A. M. Bagirov, L. Jin, N. Karmitsa, A. Al Nuaimat, and N. Sultanova, Subgradient method for nonconvex nonsmooth optimization, *J. Optim. Theory Appl.*, 157 (2013), 416-435.

[4] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, New York (1999).

[5] P. R. Brent, *Algorithms for Minimization Without Derivatives*, Prentice Hall-Englewood Cliffs, New Jersey (1973).

[6] C. Beltran and F. J. Heredia, An effective line search for saddle-point problems, *J. Optim. Theory Appl.*, 125 (2005), 1-18.

[7] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York (1983).

[8] A. R. Conn and Ph. L. Toint, *An Algorithm Using Quadratic Interpolation for Unconstrained Derivative Free Optimization, In Nonlinear Optimization and Applications*, Plenum Publishing, 27-47, New York (1996).

[9] A. Fuduli, M. Gaudioso, and G. Giallombardo, A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization, *Optim. Methods Softw.*, 19 (2004), 89-102.

[10] A. Judistky, G. Lan, A. Nemirovski, and A. Shapiro, Robust stochastic approximation approach to stochastic programming, *SIAM. J. Optim.*, 19 (2009), 1574-1609.

[11] Y. Nesterov, Primal-dual subgradient method for convex problems, *Math. Program*, 120 (2009), 221-259.

[12] A. Nedic and A. Ozdaglar, Subgradient method for saddle-point problems, *J. Optim. Theory Appl.*, 149 (2009), 205-228.

[13] T. Polyak, *Introduction to Optimization*, Optimization Software, New York (1987).

[14] M. J. D Powell, Direct search algorithms for optimization calculations, *Acta Numerica*, 7 (1998), 287-336.

[15] N. Z. Shor, *Minimization Method for Non-differentiable Functions*, Springer, Berlin (1985).

[16] M. H. Wright, *Direct Search Methods:once scorned, now respectable, in numerical analysis*, Addison-Wesley Longman, 191-208, Harlow (1995).

Morteza Maleknia
Department of Mathematics
Ph.D Student of Mathematics
University of Isfahan
Isfahan, Iran
E-mail: m.maleknia@sci.ui.ac.ir