Iranian Journal of Operations Research
Vol. 12, No. 2, 2021, pp. 73-82

A nonmonotone extension of the line search method for minimization of locally Lipschitz functions

Z. Akbari[1,*], Z. Saeidian[2]

In this paper, a nonmonotone line search strategy is presented for minimization of the locally Lipschitz continuous function. First, the Armijo condition is generalized along a descent direction at the current point. Then, a step length is selected along a descent direction satisfying the generalized Armijo condition. We show that there exists at least one step length satisfying the generalized Armijo condition. Next, the nonmonotone line search algorithm is proposed and its global convergence is proved. Finally, the proposed algorithm is implemented in the MATLAB environment and compared with some methods in the subject literature. It can be seen that the proposed method not only computes the global optimum also reduces the number of function evaluations than the monotone line search method.

Keywords: Lipschitz functions, nonmonotone line search method, Armijo condition, minimization algorithm, Global convergence.

Manuscript was received on 02/03/2022, revised on 03/07/2022 and accepted for publication on 05/23/2022.

1. Introduction

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

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

where $f:\mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz continuous function. There exist several iterative methods for solving the problem (1) where they use the monotone iterative techniques [1-9]. If the sequence $\{x_k\}$ be generated by a monotone iterative algorithm, then $f(x_{k+1}) \le f(x_k)$. These methods may not converge to the global optimal point and converge to the local optimal point when the initial point is selected near to that point. This is a disadvantage of monotone methods, while nonmonotone methods do not dependent on the initial point.

In this paper, we try to extend a nonmonotone method for solving (1). When $f:\mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function, then one of the most important methods, for solving the problem (1), is the line search method [10]. The basis of the line search method is finding a step length α_k along a descent direction d_k. The line search method is divided into two classes: nonmonotone and monotone. The Computing of the step length is done by the exact and inexact techniques.
In the exact search method, α_k is calculated from solving the following problem:

$$\min_{\alpha>0} f(x_k + \alpha\, d_k)\,.$$

As can be seen, the above problem is an optimization problem, and it has a high computational cost in solving Large-scale problems. In the inexact line search method, α_k is the largest number satisfying the Armijo condition:

* Corresponding Author.
1 Department of Mathematical Sciences, University of Mazandaran, Babolsar, Iran, Email: z.akbari@umz.ac.ir.
2 Department of Mathematical Sciences, University of Kashan, Kashan, Iran, Email: saeidian@kashanu.ac.ir.

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T d_k,$$

where $c_1 \in (0,1)$ and $\nabla f(x_k)$ is the gradient of f at the point x_k [10]. The Armijo condition is also called the sufficient reduction condition. α_k is usually calculated from the Backtracking method. For the first time, the nonmonotone line search technique was introduced by Grippo et al. [11] for solving the smooth one of the problem (1). In this method, d_k is the Newton descent direction and $\alpha_k \in (0,1]$ is the biggest nonnegative number such that

$$f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq m(k)} \left[f\left(x_{k-j}\right) \right] + c_1 \alpha_k \nabla f(x_k)_k^T d, \tag{2}$$

where $m(0) = 0$, $0 \leq m(k) \leq min[m(k-1)+1, M]$, for $k \geq 1$, and M is a nonnegative integer. Afterwards, many researchers studied the nonmonotone line search technique and its global convergence [12,13]. Due to the good numerical results of the nonmonotone line search method, the researchers combined this method with other popular methods in nonlinear optimization [14,15].

There is not any nonmonotone technique for minimizing the locally Lipschitz continuous function. In this work, we get the nonmonotone idea and propose a new method for solving the nonsmooth optimization problem (1). We review some preliminary concepts of the nonsmooth analysis in Section 2. In Section 3, the NN line search is proposed for computing a step length along a given descent direction. Then, the global convergence property of the presented minimization algorithm is proved. The presented algorithm is generalized to find a descent direction and step length satisfying the N Armijo condition. Next, the global convergence property of the presented algorithm is shown. The numerical results are reported in Section 5. Section 6 states the conclusion and the future research.

2. Preliminaries

In this section, we state the basic concepts and definitions of the nonsmooth analysis [16]. The Clarke generalized directional derivative of the locally Lipschitz function f at the point x in the direction d is defined by:

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

The Clarke generalized subdifferential at point x is given

$$\partial f(x) = \{\xi \in \mathbb{R}^n \mid f^\circ(x,d) \geq \xi^T d \quad \forall\, d \in \mathbb{R}^n\},$$

where each vector $\xi \in \partial f(x)$ is called the subgradient of f at x. For $\varepsilon > 0$, the Goldstein ε-subdifferential of f at the point x is the set

$$\partial_\varepsilon f(x) := \text{cl con}\{\partial f(y), \|x - y\|_2 \leq \varepsilon\},$$

where "cl con" is the closure convex hull of a set. Each vector $\xi \in \partial_\varepsilon f(x)$ is called an ε-subgradient of the function f at x [16]. It can be seen that $f_\varepsilon^\circ(x,d) = \sup_{\xi \in \partial_\varepsilon f(x)} \xi^T d$ for all $d \in \mathbb{R}^n$. If f be differentiable at x, then $\nabla f(x) \in \partial f(x)$. Furthermore, if f is continuously differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$. x^* is called as an ε-stationary point of f if $0 \in \partial_\varepsilon f(x)$ or $f_\varepsilon^\circ(x,d) \geq 0$ for all $d \in \mathbb{R}^n$.

3. Nonsmooth nonmonotone line search technique and its convergence

In this section, we present a new method for the problem (1) and we show its global convergence property. First of all, we define a descent direction by $\partial_\varepsilon f(\cdot)$, for a given $\varepsilon > 0$. Next, by this definition, we generalize the Armijo condition and show that it is well-defined. Then, we extend the Nonmonoton (N) line search algorithm and demonstrate its convergence property. Let v_k is a vector of $\partial_\varepsilon f(x_k)$, with the least l_2-norm:

$$v_k = \text{argmin}\,\{\|\xi\| \;\; |\xi \in \partial_\varepsilon f(x_k)\}. \tag{3}$$

If $v_k^T d < 0$, then d is a descent direction for f at x_k. Now, for a decreasing direction, the Nonmonotone Armijo condition is defined for the continuous locally Lipschitz functionas as follows:

Definition 3.1. Suppose that d_k is a descent direction for the function f at x_k.We say that the step length $\alpha > 0$ satisfies the Nonmonotone Armijo condition, if the following inequality holds:

$$f(x_k + \alpha_k d_k) - f\left(x_{l(k)}\right) \leq -c_1 \alpha_k v_k^T d_k \ , \tag{4}$$

where v_k is the solution of (3), $c_1 \in (0,1], m(0) = 0,\ f\left(x_{l(k)}\right) = \max_{0\leq\, j \leq\, m(k)} f\left(x_{k-j}\right)$, for $k \geq 1$, $0 \leq m(k) \leq min\{m(k-1)+1, M\}$ and M is a nonnegative integer.

Now, we show that there exists at least one step length for each descent direction such that the N Armijo condition (4) holds.

Proposition 3.2. Suppose that $\varepsilon > 0$ and v_k is the solution of the problem (3), then

$$f\left(x_k + \frac{\varepsilon}{||d_k||}\ d_k\right) - f\left(x_{l(k)}\right) \leq\ -c_1 \varepsilon \ \|v_k\|, \tag{5}$$

where $d_k = -v_k$.

Proof. Let $t = \frac{\varepsilon}{||d_k||}$. Since f is a locally Lipschitz continuous function, then according to Mean-Value Theorem, Theorem 3.18 in [16], there exists $z \in (x_k, x_k + td_k)$, where $[x_k, x_k + t\ d_k]$ is the line segment, such that

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

Therefore, there exits $\xi \in \partial f(z)$

$$f(x_k + td_k) - f(x_k) = t\xi^T d_k. \tag{6}$$

Since $\left||z - x_k|\right| \leq \varepsilon$, then $\xi \in \partial_\varepsilon f(x_k)$. As respects $f_\varepsilon^\circ(x_k, d_k) = max\{\xi^T d_k\ |\xi \in \partial_\varepsilon f(x_k)\}$, we have

$$f(x_k + td_k) - f(x_k) = t\xi^T d_k \leq t\ f_\varepsilon^\circ(x_k, d_k).$$

Hence

$$f(x_k + td_k) \leq\ f(x_k) + tf_\varepsilon^\circ(x_k, d_k).$$

On the other hand $f\left(x_{l(k)}\right) = \max_{0\leq\, j \leq\, m(k)} f(x_{k-j})$. Thus

$$f(x_k + td_k) \leq f(x_k) + tf_\varepsilon^\circ(x_k, d_k) \leq f(x_{l(k)}) + tf_\varepsilon^\circ(x_k, d_k).$$

Also $c_1 \in (0,1)$ and $f_\varepsilon^\circ(x_k, d_k) < 0$, so

$$f(x_k + t\ d_k) \leq f(x_{l(k)}) + t\ f_\varepsilon^\circ(x_k, d_k) \leq f(x_{l(k)}) + c_1 t\ f_\varepsilon^\circ(x_k, d_k),$$

and the proof is complete.

Now, we are ready to present the new N line search method algorithmically as follows:

Algorithm 3.1 (Nonmonotone line search technique)
Step 1. Set $\varepsilon, \sigma, c_1 \in (0,1), x_0 \in \mathbb{R}^n, k = 0$, and a positive integer M.
Step 2. Consider v_k as a solution of the problem (3)
    If $\|v_k\| \leq \varepsilon$, **then** stop, **else** set $d_k = -v_k$ and got to the Step 3.
Step 3 Set $\alpha = \sigma$ and $\theta = min\{\varepsilon, \frac{\varepsilon}{\|d_k\|}\}$
    While $f(x_k + \alpha d_k) > f(x_{l(k)}) - c_1 \alpha\ \|v_k\|^2$ and $\alpha > \theta$
        $\alpha: = \sigma * \alpha;$
    End(While)
Step 4 If $\alpha > \varepsilon_k$, **then** $\alpha_k: = \alpha$, **else** $\alpha_k := \theta$. Set $x_{k+1} = x_k + \alpha_k d_k, k = k+1$ and go to Step 2.

The following lemma shows that the generalized sequence $\{f(x_{l(k)}\}$, by Algorithm 3.1, is nonincreasing.

Lemma 3.3 Suppose that f is a locally Lipschitz function and the level set $\mathcal{L} = \{x \,| f(x) \leq f(x_0)\}$ is bounded. Then the following inequalities hold,

$$f(x_{k+1}) \leq f\big(x_{l(k+1)}\big) \leq f\big(x_{l(k)}\big),$$

where the sequence $\{f\big(x_{l(k)}\big)\}$ is generated by Algorithm 3.1. Also $\{f\big(x_{l(k)}\big)\}$ has at least one limit point.

Proof. According to the definition of $m(k)$, we have $m(k+1) \leq m(k) + 1$,

$$\begin{aligned} f(x_{l(k+1)}) &= \max_{0\leq j\leq m(k+1)} f(x_{k+1-j}) \\ &\leq \max_{0\leq j\leq m(k)+1} f(x_{k+1-j}) \\ &= \max\{f\big(x_{l(k)}\big), f(x_{k+1})\} \\ &= f\big(x_{l(k)}\big). \end{aligned}$$

Hence $f(x_k) \leq f\big(x_{l(k)}\big) \leq f(x_0)$, so $\{x_k\} \subset \mathcal{L}$. On the other hand, $\mathcal{L}$ is bounded, therefore $\{x_k\}$ has at least one convergent subsequence. Since the function f is locally Lipschitz function, so the sequence $\{f(x_{l(k)})\}$ has at least a limit point.

Now, we are ready to prove the global convergence of Algorithm 3.1. In the following, theorem, we show that $0 \in \partial f(x^*)$, for each accumulation point x^* of the generated sequence $\{x_k\}$ by Algorithm 3.1.

Theorem **3.4** Let $f: \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function and the level set $\mathcal{L}$ be bounded. If Algorithm 3.1 does not terminate after finitely many iterations, then $0 \in \partial f(x^*)$, where x^* is a limit point of $\{x_k\}$.

Proof. Suppose that $\varepsilon > 0$ and Algorithm 3.1 does not terminate after finitely many iterations. From (5), we have

$$f\big(x_{l(k)}\big) - f\big(x_{l(l(k)-1)}\big) \leq -c_1 \alpha_{l(k)-1} \big\|v_{l(k)-1}\big\|^2, \quad \text{for} \quad k > M \tag{7}$$

Lemma 3.2 shows that the sequence $\{f(x_{l(k)})\}$ is convergent, then taking limit, $k \to \infty$, (7) implies

$$\lim_{k\to\infty} \alpha_{l(k)-1} \big\|v_{l(k)-1}\big\|^2 = 0. \tag{8}$$

Since $d_k = -v_k$ and $\alpha_k \leq \sigma$, (8) implies

$$\lim_{k\to\infty} \alpha_{l(k)-1} \big\|d_{l(k)-1}\big\| = 0.$$

Let $\hat{l}(k) = l(k + M + 2)$, we prove by induction that

$$\lim_{k\to\infty} \alpha_{\hat{l}(k)-j} \big\|d_{l(k)-j}\big\| = 0 \tag{9}$$

for $k \geq j - 1$ and $j \geq 1$. If $j = 1$,)8) implies (9). Now, we assume that (9) holds for a given j, and we prove (9) for $j + 1$. Consider (5) at $k = \hat{l}(k) - (j + 1)$ as follows:

$$f\left(x_{\hat{l}(k)-j}\right) \leq f(x_{l\left(\hat{l}(k)-(j+1)\right)}) - c_1\alpha_{\hat{l}(k)-(j+1)}\left|\left|v_{\hat{l}(k)-(j+1)}\right|\right|^2.$$

Using the same technique in converting the equation (7) to (8), we have

$$\lim_{k\to\infty} \alpha_{\hat{l}(k)-(j+1)}\left|\left|d_{\hat{l}(k)-(j+1)}\right|\right| = 0.$$

Then (9) is correct for all $k \geq j-1$ and $j \geq 1$. Thus $||x_{\hat{l}(k)} - x_{\hat{l}(k)-(j+1)}|| \to 0$ and since f is locally Lipschitz, we have

$$\lim_{k\to\infty} f(x_{\hat{l}(k)-(j+1)}) = \lim_{k\to\infty} f(x_{\hat{l}(k)-j}) = \lim_{k\to\infty} f(x_{l(k)}),\ \forall j \geq 1. \tag{10}$$

On the other hand, we have

$$x_{k+1} = x_{\hat{l}(k)} - \sum_{j-1}^{\hat{l}(k)-k-1} \alpha_{x_{\hat{l}(k)-j}} d_{x_{\hat{l}(k)-j}},$$

then

$$\lim_{k\to\infty} ||x_{k+1} - x_{\hat{l}(k)}|| = 0.$$

Since f is locally Lipschitz and by (10), then we have

$$\lim_{k\to\infty} f(x_{k+1}) = \lim_{k\to\infty} f(x_{\hat{l}(k)}) = \lim_{k\to\infty} f(x_{l(k)}).$$

We have the N Armijo condition (5) as follows:

$$f(x_{k+1}) \leq f\left(x_{\hat{l}(k)}\right) - c_1\alpha_k\|v_k\|^2.$$

Taking limits for $k \to \infty$

$$\lim_{k\to\infty} \alpha_k \|v_k\|^2 = 0,$$

and since $\alpha_k \leq \sigma$ is bounded, we have

$$\lim_{k\to\infty} \|v_k\| = 0. \tag{16}$$

Since $\partial f(\cdot)$ is upper semicontinuous, so we have $0 \in \partial f(x^*)$, where x^* is an accumulation point of the sequence $\{x_k\}$ and proof is complete.

4. Numerical results

In this section, the numerical results are reported to show the performance of the N line search technique. The proposed algorithm is compared with the steepest descent approximation algorithm in [18] on the nonsmooth optimization problems in [16,19,20]. The algorithms are implemented in the MATLAB *2019b* environment. When we compute the function value, a subgradient is computed. Thus the number of function and subgradient evaluations are equal. So, we just report the number of function evaluations for comparing these two algorithms.

Table 1 contains the test problems with their optimal values in $n = 10, n = 100$ and $n = 1000$, where n indicates the dimension of the problem. Problems 8 and 10 are nonconvex and the rest of the problems are convex.

Table 1. The test problems with their optimum values

No.	Problems	$n = 10$	$n = 100$	$n = 1000$
1	MAXQ	0	0	0
2	MAXHILB	0	0	0
3	LQ	-1.272792e+01	-1.400071e+02	−1.41279e+003
4	CB3I	18	198	1998
5	CB3II	18	198	1998
6	problem 2 from TEST29	0	0	0
7	problem 5 from TEST29	0	0	0
8	problem 6 from TEST29	0	0	0
9	problem11from TEST29	1.019614e+02	1.186324e+03	1.20312e+004
10	problem 13 from TEST29	4.537978e+00	5.559023e+01	5.66131e+002

Since the problem (3) is not practical in many cases, instead of v_k, the approximate solution for the problem (3) is used, i.e. w_k. w_k is the approximated solution to the problem (3) which is obtained from the reference [18]. The N Armijo condition is also replaced by the following condition:

$$f(x_k + \alpha_k d_k) - f\big(x_{l(k)}\big) \leq -c_1 \alpha_k \|w_k\|^2,$$

where $d_k = -w_k$.

By using the above condition instead of the condition (5), as shown in the proof of Theorem 1, we can show:

$$\lim_{k\to\infty} \alpha_k \|w_k\|^2 = 0.$$

Since $\|v_k\|^2 \leq \|w_k\|^2$, so we have

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

The alternative condition results in the correctness of Algorithm 3.1. We investigate the efficiency of the proposed algorithm in solving the test problems for $M = 1,2,5,10$. In the case $M = 1$, the proposed algorithm is converted to the steepest descent approximation algorithm in [18]. Nonsmooth nonmonotone line search algorithm and steepest descent approximation algorithm are shown with NLS and MY, respectively. We consider the abbreviations n_{NLS} and n_{MY} for the number of function evaluations of the NLS algorithm and the MY algorithm, respectively. The parameters are initialized similar to MY algorithm in [18] as follows:

$$\varepsilon_0 = 10^{-3}, \delta_0 = 10^{-4}, c_1 = 10^{-4}, \sigma = .5.$$

In references [19,20], the starting points are chosen so that a test algorithm may fall in the local minimum, hence we set the starting point as the one selected in the references. We say that an algorithm solves a test problem if the following inequality holds:

$$\frac{|f^* - \hat{f}|}{|f^*| + 1} \leq \gamma,$$

where γ, f^*, and $\hat{f}$ are the number of digits means of the optimal solution, the optimal value, and the computed optimal value, respectively. Now, we report how many problems are solved by the proposed algorithm for different M and γ in Table 2.

Table 2. The number of problems solved by the proposed algorithm for difference M and γ.

	$n = 10$		$n = 100$		$n = 1000$	
M	$\gamma = 10^{-4}$	$\gamma = 10^{-5}$	$\gamma = 10^{-4}$	$\gamma = 10^{-5}$	$\gamma = 10^{-4}$	$\gamma = 10^{-5}$
1	9	7	6	4	5	5
2	9	8	8	5	7	6
5	7	7	6	5	6	6
10	9	7	7	4	7	6

According to Table 2, the case $M = 2$ is the best choice because the proposed algorithm solves more problems in this case than in other cases. The performance of the algorithms is roughly similar for $n = 10$. When the dimension increases, more problems are solved for $m \geq 2$. This demonstrates that the proposed method is more efficient than MY method for large-scale problems. Also, the most test problems are solved by the proposed algorithm in the case $M = 2$. In Table 3, we report the optimal value of problems where obtained by NLS method in the cases $M = 1,2,5,10$.
Since better results were obtained for the case $M=2$, then we report the ratio of the number of function evaluations by the MY method to the number of function evaluations by NLS method for $M=2$, i.e. $\frac{n_{MY}}{n_{NLS}}$. This ratio indicates how increasing inaccuracy is related to the increase in the number of function evaluations. In Table 4, the symbols ``+" and ``-" are used to show NLS algorithm solves a problem successfully or unsuccessfully, respectively. In the case $M=2$ and the large scale dimensions, the optimal solution of some problems is calculated with the much lower number of function evaluations. In other problems, there is no significant difference between cases, but this is negligible given that the proposed method can solve more problems.

5. Conclusion

In this paper, we presented the nonsmooth nonmonotone line search technique for solving nonsmooth optimization problems for the first time. We generalized the Armijo condition for the locally Lipschitz function, where called the N Armijo condition. Then, we showed that there exists at least one step length for each descent direction satisfying the N Armijo condition. The minimization algorithm was proposed and its global convergence was proved. Afterward, the proposed algorithm and the steepest descent approximation method were implemented and compared. The reported numerical results showed that the proposed algorithm has better implementation than the steepest descent approximation method. In future work, we want to combine the nonsmooth trust region method with the nonmonotone line search technique. We guess, if the trust region method is combined with the N line search technique, then we will get better numerical results.

Table 3 The obtained optimal values for $M = 1,2,5,10$ for sizes $n = 10,100,1000$.

No.	$n = 10$				$n = 100$				$n = 100$			
	$M = 1$	$M = 2$	$M = 5$	$M = 10$	$M = 1$	$M = 2$	$M = 5$	$M = 10$	$M = 1$	$M = 2$	$M = 5$	$M = 10$
1	2.93e-13	2.93e-13	2.93e-13	2.93e-13	3.18e-13	3.18e-13	3.18e-13	3.18e-13	4.97E-13	4.97e-13	4.97e-13	4.97e-13
2	3.25e-06	2.08e-06	3.34e-06	2.79e-06	8.80e-06	9.27e-06	7.75e-06	3.93e-05	0.000953301	0.000828806	0.000352293	0.000135647
3	-12.7277	-12.7279	-12.7279	-12.7279	-140.0048	-140.005	-140.0042	-140.0047	-1412.798	1412.798	-1412.798	-1412.799
4	18.00001	18.00001	18.00001	18.00001	198.0209	198	198	198	1999.577	1998.002	1998.001	1998.005
5	18.00001	18.00006	18.00536	18.00055	198.0161	198.0107	198.0211	198.0215	1998.721	1998.035	1998.231	1998.062
6	5.66e-07	5.64e-07	5.72e-07	5.72e-07	3.40e-07	5.28e-07	4.96e-07	4.96e-07	4.92e-07	4.78e-07	4.79e-07	4.81e-07
7	1.97e-05	1.06e-05	8.49e-06	1.15e-05	0.000124626	6.05e-05	0.000221875	0.000157379	0.0213502	0.000722257	0.000624773	0.002145679
8	3.35e-06	5.80e-07	9.03e-07	8.32e-07	3.35e-06	5.80e-07	8.32e-07	5.80e-07	3.35e-06	5.80e-07	9.03e-07	8.32e-07
9	106.0591	106.0593	106.0591	106.0593	1187.64	1187.643	1186.985	1186.422	12031.34	12031.32	12031.32	12031.25
10	4.538056	4.538013	4.540724	4.538025	55.67039	55.65863	55.67198	55.66436	566.3342	566.2807	566.3198	566.3238

Table 4. Success and failure in solving problems

	$n = 10$			$n = 100$			$n = 1000$		
No.	MY	NLS	$\frac{n_{MY}}{n_{NLS}}$	MY	NLS	$\frac{n_{MY}}{n_{NLS}}$	MY	NLS	$\frac{n_{MY}}{n_{NLS}}$
1	+	+	1.34	+	+	1.01	+	+	1
2	+	+	1.62	+	+	2.78	-	-	-
3	+	+	1.86	+	+	0.77	+	+	0.86
4	+	+	0.55	-	+	-	-	+	-
5	+	+	2.99	+	+	1.36	-	+	-
6	+	+	2.57	+	+	1.15	+	+	1.44
7	+	+	1.27	-	+	-	-	-	-
8	+	+	0.079	+	+	0.079	+	+	0.07
9	-	-	-	-	-	-	+	-	1.64
10	+	+	0.001	-	-	-	-	-	-

References

[1] Burke, J.V., Lewis, A.S. and Overton, M.L. (2005). A robust gradient sampling algorithm fornonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15:571–779,

[2] Akbari, Z., Yousefpour, R. and Peyghami, M. R., (2015). A new nonsmooth trust region algorithm for locally Lipschitz unconstrained optimization problems. *SIAM J. Optim.,* 164, 733-754.

[3] Shor, N. Z., (1985). Minimization methods for non-differentiable functions. Berlin, Springer.

[4] Frangioni, A., (2002). Generalized bundle methods. *SIAM J. on Optimization*, 13, 117-156.

[5] Mäkelä, M. M., (2002). Survey of bundle methods for nonsmooth optimization. *Optimization Methods and Software*, 17, 1-29.

[6] Hiriart-Urruty, J., and Lemaréchal, C., (1993). Convex analysis and minimization algorithms II: advanced theory and bundle method. Berlin, Springer.

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

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

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

[10] Nocedal, J., Wright, S.J., (1999). Numerical optimization. Berlin, Springer.

[11] Grippo, L., Lampariello, F., Lucidi, S., (1986). A nonmonotone line search technique for newton's method. *SIAM J. Numer. Anal.*, 23(4), 707-716.

[12] Dai, Y., (2002). On the nonmonotone line search. *J. Optim. Theory Appl.*, 112(2), 315-330.

[13] Zhang, H., Hager, W., (2004). A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. Optim.*, 14, 1043-1056.

[14] Sun., W., (2004). Nonmonotone trust region method for solving optimization problems. *Appl. Math. Comput*., 156(1), 159-174.

[15] Ataee Tarzanagh, D., Peyghami, M. and Mesgarani, H., (2014). A new nonmonotone trust region method for unconstrained optimization equipped by an efficient adaptive radius. *Optimization Methods and Software*, 29(4), 819-836.

[16] Bagirov, A., Karmitsa, N., Mäkelä, M. M., (2014). Introduction to nonsmooth optimization: theory, *practice and software.* Berlin, Springer.

[17] Dolan, E.D., More, J.J., (2002). Benchmarking optimization software with performance profiles. *Math. Program*., 91, 201-213.

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

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

[20] Luksan, L., Tuma, M., Siska, M., Vlcek, J., Ramesova, N., (2002). Ufo 2002. Interactive system for universal functional optimization. *Tech. rep., Academy of Sciences of the Czech Republic*.