FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 35, No 5 (2020), 1417–1438
https://doi.org/10.22190/FUMI2005417I

MULTIPLE USE OF BACKTRACKING LINE SEARCH IN UNCONSTRAINED OPTIMIZATION

Branislav Ivanov, Bilall I. Shaini and Predrag S. Stanimirović

Abstract. The class of gradient methods is a very efficient iterative technique for solving unconstrained optimization problems. Motivated by recent modifications of some variants of the SM method, this study proposed two methods that are globally convergent as well as computationally efficient. Each of the methods is globally convergent under the influence of a backtracking line search. Results obtained from the numerical implementation of these methods and performance profiling show that the methods are very competitive with respect to well-known traditional methods.
Keywords: unconstrained optimization; gradient methods; line search.

1. Introduction

The following unconstrained optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \tag{1.1}$$

is ubiquitous in all areas of science and practical engineering applications. In (1.1), the function $f:\mathbb{R}^n\to\mathbb{R}$ is uniformly convex (UC) and twice continuously differentiable (TCD).

The most frequent iterations for solving (1.1) is the gradient descent (GD) iterative scheme

$$\mathbf{x}_{k+1}^{GD} = \mathbf{x}_k^{GD} - t_k\mathbf{g}_k, \tag{1.2}$$

where $t_k > 0$ is the stepsize and $\mathbf{g}_k := \nabla f(\mathbf{x}_k)$ corresponds to the gradient of f. The step length t_k is mainly calculated using the backtracking line search (BLS).

The Newton iterations stabilized by the line search are defined as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k G_k^{-1}\mathbf{g}_k, \tag{1.3}$$

Received November 10, 2020; accepted November 23, 2020
2020 *Mathematics Subject Classification.* Primary 65K05; Secondary 90C30

wherein G_k^{-1} means the inverse of the Hessian matrix $G_k := \nabla^2 f(\mathbf{x}_k)$. In order to avoid time consuming computation of the Hessian and its inverse, practical numerical methods for solving unconstrained optimization problem are derived from the usage of appropriate approximations H_k of G_k^{-1}. The general scheme of quasi-Newton type with line search [16] is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k H_k \, \mathbf{g}_k. \tag{1.4}$$

In order to define efficient class of quasi-Newton methods, we use the simplest scalar approximation of the Hessian with respect to known classifications from [5, 8]:

$$B_k := \gamma_k I, \ \ \gamma_k > 0, \tag{1.5}$$

where I is an identity matrix of appropriate order and $\gamma_k > 0$ is a real parameter. The choice (1.5) leads to the iterative prototype

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k^{-1} t_k \mathbf{g}_k, \tag{1.6}$$

where t_k denotes the basic step size and γ_k^{-1} is an additional step size which should be defined appropriately. Clearly, the value $\gamma_k^{-1} t_k$ can be considered as a composite step size, so that iterations (1.6) are GD methods. The iterations (1.6) are known as *improved gradient descent* (IGD) methods.

Andrei in [1, 3] originated so called *Accelerated Gradient Descent* (AGD) iterations in the form

$$\mathbf{x}_{k+1}^{AGD} = \mathbf{x}_k^{AGD} - \theta_k^{AGD} t_k \mathbf{g}_k. \tag{1.7}$$

The AGD process (1.7) was improved into the *Modified AGD* (MAGD) method [7] as

$$\mathbf{x}_{k+1}^{MAGD} = \mathbf{x}_k^{MAGD} - \theta_k (t_k + t_k^2 - t_k^3) \mathbf{g}_k. \tag{1.8}$$

A few variants of the IGD class (1.6) were proposed in [7, 10, 11, 14, 15]. The *SM* method belongs to the class IGD methods. It was originated in [14] by the iterative process

$$\mathbf{x}_{k+1}^{SM} = \mathbf{x}_k^{SM} - t_k \left(\gamma_k^{SM}\right)^{-1} \mathbf{g}_k, \tag{1.9}$$

where $t_k > 0$ is the basic step size and $\gamma_k^{SM} > 0$ is the gain parameter determined as in

$$\gamma_{k+1}^{SM} = 2\gamma_k^{SM} \frac{\gamma_k^{SM} \left[f(\mathbf{x}_{k+1}^{SM}) - f(\mathbf{x}_k^{SM}) \right] + t_k \|\mathbf{g}_k\|^2}{t_k^2 \|\mathbf{g}_k\|^2}.$$

The ADSS model from [10] is defined as

$$\mathbf{x}_{k+1}^{ADSS} = \mathbf{x}_k^{ADSS} - \left(t_k \left(\gamma_k^{ADSS}\right)^{-1} + l_k \right) \mathbf{g}_k, \tag{1.10}$$

where t_k and l_k are determined by BLSs. The TADSS method [15] is defined by the iterative rule

$$\mathbf{x}_{k+1}^{TADSS} = \mathbf{x}_k^{TADSS} - \left(t_k\left((\gamma_k^{TADSS})^{-1} - 1\right) + 1\right)\mathbf{g}_k.$$

The next scheme was proposed as the modified SM (MSM) method in [7]:

$$\mathbf{x}_{k+1}^{MSM} = \mathbf{x}_k^{MSM} - (t_k + t_k^2 - t_k^3)(\gamma_k^{MSM})^{-1}\mathbf{g}_k. \tag{1.11}$$

The acceleration parameters in ADD, ADSS, TADSS and MSM methods are summarized in Table 1.1.

Table 1.1: Acceleration parameters γ_{k+1} in variants SM method.

Method	**Acceleration parameter** γ_{k+1}	**Reference**
ADD	$\gamma_{k+1}^{ADD} = 2\frac{f(\mathbf{x}_{k+1}^{ADD}) - f(\mathbf{x}_k^{ADD}) - t_k(\mathbf{g}_k^{ADD})^{\mathrm{T}}\left(t_k\mathbf{d}_k^{ADD} - (\gamma_k)^{-1}\mathbf{g}_k\right)}{\left(t_k\mathbf{d}_k^{ADD} - \gamma_k^{-1}\mathbf{g}_k\right)^{\mathrm{T}}\left(t_k\mathbf{d}_k^{ADD} - (\gamma_k^{ADD})^{-1}\mathbf{g}_k\right)}$	(2014) [11]
ADSS	$\gamma_{k+1}^{ADSS} = 2\frac{f(\mathbf{x}_{k+1}^{ADSS}) - f(\mathbf{x}_k^{ADSS}) + \left(t_k(\gamma_k)^{-1} + l_k\right)\|\mathbf{g}_k\|^2}{\left(t_k(\gamma_k^{ADSS})^{-1} + l_k\right)^2\|\mathbf{g}_k\|^2}$	(2015) [10]
TADSS	$\gamma_{k+1}^{TADSS} = 2\frac{f(\mathbf{x}_{k+1}^{TADSS}) - f(\mathbf{x}_k^{TADSS}) + \psi_k\|\mathbf{g}_k\|^2}{\psi_k^2\|\mathbf{g}_k\|^2}$, $\psi_k = t_k((\gamma_k^{TADSS})^{-1} - 1) + 1$	(2015) [15]
MSM	$\gamma_{k+1}^{MSM} = 2\gamma_k\frac{\gamma_k\left[f(\mathbf{x}_{k+1}^{MSM}) - f(\mathbf{x}_k^{MSM})\right] + (t_k + t_k^2 - t_k^3)\|\mathbf{g}_k\|^2}{(t_k + t_k^2 - t_k^3)^2\|\mathbf{g}_k\|^2}$	(2019) [7]

The main goal of this research is to study the impact of multiple usage of backtracking line search in modified SM method [7] and practical computational performance of two new methods. Our intention is to propose and investigate improvements of the MSM method. Globally, we investigate possibility to multiple use backtracking line search in the modified MSM method.

Main results of the present study can be highlighted as follows:

(1) A novel iterative scheme is proposed using the idea of computing the step parameters t_k, t_k^2 and t_k^3 in the MSM method by means of multiple BLS procedures. The resulting iterations will be denoted as TMSM and DMSM.

(2) Convergence behavior of the proposed iterations are investigated on appropriate quadratic functions.

(3) Numerical experiments compare introduced methods with existing iterations and analyze three main performances: number of iterative steps and function evaluations and CPU time.

The remainder of the paper is developed according to the following hierarchy of sections. Two modifications of the MSM methods, termed as TMSM and DMSM methods, are introduced in Section 2. Section 3. investigates the convergence of the presented TMSM and DMSM methods. In Section 4., we perform a number of numerical experiments and compare main performances of the novel methods with similar available methods. Final remarks are presented in Section 5.

2. Multiple use of backtracking line search in modified SM method

The MSM method is based on the iteration

$$\mathbf{x}_{k+1}^{MSM} = \mathbf{x}_k^{MSM} - t_k^{MSM}(\gamma_k^{MSM})^{-1}\mathbf{g}_k, \tag{2.1}$$

where $t_k^{MSM} = t_k + t_k^2 - t_k^3$. The leading idea in defining t_k^{MSM} arises from the observation $t_k + t_k^2 > t_k^{MSM} > t_k$, which means that the MSM method proposes a slightly greater step size with respect to the SM iterations. Since t_k arises from the BLS procedure, which ensures $t_k \in (0,1)$, it implies

$$t_k \le t_k^{MSM} \le t_k + t_k^2.$$

Our intention in current research is to improve behaviour of iterations (2.1) using two or three appropriately defined step-parameters. Following this idea, a method based on triple usage of the BLS in the MSM method is obtained when t_k^2 is substituted with l_k^2 and t_k^3 is substituted with j_k^3 in (2.1), where t_k, l_k and j_k are defined by independent LS procedures: the first BLS (Algorithm 1) calculates t_k, another BLS (Algorithm 2) calculates l_k, while the third BLS (Algorithm 3) determines j_k.

Replacing the above changes gives the expression of the TMSM iteration:

$$\mathbf{x}_{k+1}^{TMSM} = \mathbf{x}_k^{TMSM} - t_k^{TMSM}\left(\gamma_k^{TMSM}\right)^{-1}\mathbf{g}_k, \tag{2.2}$$

where

$$t_k^{TMSM} = \begin{cases} t_k + l_k^2 - j_k^3, & t_k + l_k^2 - j_k^3 > t_k \\ t_k, & t_k + l_k^2 - j_k^3 \le t_k. \end{cases} \tag{2.3}$$

The second order Taylor development of $f(\mathbf{x}_{k+1}^{TMSM})$ gives

$$\begin{aligned} f(\mathbf{x}_{k+1}^{TMSM}) \approx{}& f(\mathbf{x}_k^{TMSM}) - t_k^{TMSM}\left(\gamma_k^{TMSM}\right)^{-1}\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k \\ &+ \frac{1}{2}\left(t_k^{TMSM}\right)^2\left((\gamma_k^{TMSM})^{-1}\mathbf{g}_k\right)^{\mathrm{T}}\nabla^2 f(\xi)\left(\gamma_k^{TMSM}\right)^{-1}\mathbf{g}_k. \end{aligned} \tag{2.4}$$

The parameter ξ in (2.4) fulfills the condition $\xi \in [\mathbf{x}_k^{TMSM}, \mathbf{x}_{k+1}^{TMSM}]$. One possible choice is

$$\begin{aligned} \xi &= \mathbf{x}_k^{TMSM} + \delta(\mathbf{x}_{k+1}^{TMSM} - \mathbf{x}_k^{TMSM}) \\ &= \mathbf{x}_k^{TMSM} - \varphi\, t_k^{TMSM}\left(\gamma_k^{TMSM}\right)^{-1}\mathbf{g}_k, \quad 0 \le \varphi \le 1. \end{aligned} \tag{2.5}$$

According to [14], $\nabla^2 f(\xi)$ is approximated as $\gamma_{k+1}^{TMSM} I$. So, (2.4) reduces to

$$\begin{aligned} f(\mathbf{x}_{k+1}^{TMSM}) \approx{}& f(\mathbf{x}_k^{TMSM}) - t_k^{TMSM}\left(\gamma_k^{TMSM}\right)^{-1}\|\mathbf{g}_k\|^2 \\ &+ \frac{1}{2}\left(t_k^{TMSM}\right)^2\gamma_{k+1}^{TMSM}(\gamma_k^{TMSM})^{-2}\|\mathbf{g}_k\|^2. \end{aligned} \tag{2.6}$$

Then γ_{k+1}^{TMSM} can be obtained from (2.6) as

$$\gamma_{k+1}^{TMSM} = 2\gamma_k^{TMSM} \frac{\gamma_k^{TMSM}\left[f(\mathbf{x}_{k+1}^{TMSM}) - f(\mathbf{x}_k^{TMSM})\right] + t_k^{TMSM}\|\mathbf{g}_k\|^2}{\left(t_k^{TMSM}\right)^2\|\mathbf{g}_k\|^2}. \tag{2.7}$$

The improper situation $\gamma_{k+1}^{TMSM} < 0$ can be resolved by taking $\gamma_{k+1}^{TMSM} = 1$.

The BLS method is implemented in the Algorithm 1 from [14]. Algorithm 1 defines t_k starting from $t = 1$ and subsequently decreases values of t so that it reduces the value of the objective f enough.

Algorithm 1 The backtracking line search calculates t_k.

Require: A real function $f(\mathbf{x})$, appropriate search direction $\mathbf{d}_k$ at the point $\mathbf{x}_k$ and the positive real numbers $0 < \sigma < 0.5$ and $\beta \in (0, 1)$.
1: $t = 1$.
2: While $f(\mathbf{x}_k + t\mathbf{d}_k) > f(\mathbf{x}_k) + \sigma t \mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k$, do $t := t\beta$.
3: Output $t_k := t$.

Algorithm 2 The second backtracking line search calculates l_k.

Require: Objective function $f(\mathbf{x})$, the search direction $\mathbf{d}_k$ at the point $\mathbf{x}_k$ and positive real numbers $0 < \sigma_l < 0.5$ and $\beta_l \in (0, 1)$.
1: $l = 1$.
2: While $f(\mathbf{x}_k + l\mathbf{d}_k) > f(\mathbf{x}_k) + \sigma_l l \mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k$, take $l := l\beta_l$.
3: Return $l_k = l$.

Algorithm 3 The third backtracking line search calculates j_k.

Require: Objective function $f(\mathbf{x})$, the search direction $\mathbf{d}_k$ at the point $\mathbf{x}_k$ and positive real numbers $0 < \sigma_j < 0.5$ and $\beta_j \in (0, 1)$.
1: $j = 1$.
2: While $f(\mathbf{x}_k + j\mathbf{d}_k) > f(\mathbf{x}_k) + \sigma_j j \mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k$, take $j := j\beta_j$.
3: Return $j_k = j$.

Finally, the TMSM method is described in Algorithm 4.

It is expectable that the total number of iterations (NofI) required by the TMSM method will be smaller than the number of iterations of the MSM method, but an increase in the number of function evaluations (NofFE) and the CPU time (CPUT)is expectable. Based on these indicators, we came up with the idea to omit one line search in the TMSM method. This would drastically reduce the CPUT and the NofFE. Following this idea, a method of double use backtracking line search in modified SM method is obtained. In this way, we get a new expression of the DMSM iteration:

$$\mathbf{x}_{k+1}^{DMSM} = \mathbf{x}_k^{DMSM} - t_k^{DMSM}\left(\gamma_k^{DMSM}\right)^{-1}\mathbf{g}_k, \tag{2.8}$$

Algorithm 4 Triple use of backtracking line search in the MSM method (the TMSM method)

Require: Objective function $f(\mathbf{x})$, initial point $\mathbf{x}_0^{TMSM} \in \mathrm{dom}(f)$ and parameters $0 < \lambda < 1$, $0 < \nu < 1$.

1: Put $k = 0$, evaluate $f(\mathbf{x}_0^{TMSM})$, $\mathbf{g}_0 = \nabla f(\mathbf{x}_0^{TMSM})$, and put $\gamma_0^{TMSM} = 1$.

2: If
$$\|\mathbf{g}_k\| \le \lambda \quad \text{and} \quad \frac{|f(\mathbf{x}_{k+1}^{TMSM}) - f(\mathbf{x}_k^{TMSM})|}{1 + |f(\mathbf{x}_k^{TMSM})|} \le \nu,$$
STOP; else go to Step 3.

3: (The first backtracking) Compute $t_k \in (0, 1]$ using Algorithm 1.

4: (The second backtracking) Compute $l_k \in (0, 1]$ using Algorithm 2.

5: (The third backtracking) Compute $j_k \in (0, 1]$ using Algorithm 3.

6: Determine t_k^{TMSM} using (2.3).

7: Compute $\mathbf{x}_{k+1}^{TMSM} = \mathbf{x}_k^{TMSM} - (\gamma_k^{TMSM})^{-1} t_k^{TMSM} \mathbf{g}_k$.

8: Compute $f(\mathbf{x}_{k+1}^{TMSM})$ and $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1}^{TMSM})$.

9: Determine γ_{k+1}^{TMSM} using (2.7).

10: If $\gamma_{k+1}^{TMSM} < 0$, then take $\gamma_{k+1}^{TMSM} = 1$.

11: Set $k := k + 1$, go to the step 2.

12: Return $\mathbf{x}_{k+1}^{TMSM}$ and $f(\mathbf{x}_{k+1}^{TMSM})$.

where

$$t_k^{DMSM} = \begin{cases} t_k + t_k^2 - j_k^3, & t_k + t_k^2 - j_k^3 > t_k \\ t_k, & t_k + t_k^2 - j_k^3 \le t_k. \end{cases} \tag{2.9}$$

In exactly the same way as for the TMSM method, we arrive at

$$\gamma_{k+1}^{DMSM} = 2\gamma_k^{DMSM} \frac{\gamma_k^{DMSM} \left[f(\mathbf{x}_{k+1}^{DMSM}) - f(\mathbf{x}_k^{DMSM}) \right] + t_k^{DMSM} \|\mathbf{g}_k\|^2}{\left(t_k^{TMSM}\right)^2 \|\mathbf{g}_k\|^2}. \tag{2.10}$$

The difficulty $\gamma_{k+1}^{DMSM} < 0$ can be resolved using $\gamma_{k+1}^{DMSM} = 1$.

The DMSM method is presented in Algorithm 5:

Algorithm 5 Double use backtracking line search in the MSM method (the DMSM method)

Require: Function $f(\mathbf{x})$, chosen initial point $\mathbf{x}_0^{DMSM} \in \mathrm{dom}(f)$ and parameters $0 < \lambda < 1$, $0 < \nu < 1$.

1: Put $k = 0$, evaluate $f(\mathbf{x}_0^{DMSM})$, $\mathbf{g}_0 = \nabla f(\mathbf{x}_0^{DMSM})$ and take $\gamma_0^{DMSM} = 1$.

2: If

$$\|\mathbf{g}_k\| \le \lambda \quad \text{and} \quad \frac{|f(\mathbf{x}_{k+1}^{DMSM}) - f(\mathbf{x}_k^{DMSM})|}{1 + |f(\mathbf{x}_k^{DMSM})|} \le \nu,$$

STOP; else go to Step 3.

3: (The first backtracking) Compute $t_k \in (0, 1]$ using Algorithm 1.

4: (The second backtracking) Compute $j_k \in (0, 1]$ using Algorithm 3.

5: Determine t_k^{DMSM} using (2.9).

6: Compute $\mathbf{x}_{k+1}^{DMSM} = \mathbf{x}_k^{DMSM} - (\gamma_k^{DMSM})^{-1} t_k^{DMSM} \mathbf{g}_k$.

7: Compute $f(\mathbf{x}_{k+1}^{DMSM})$ and $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1}^{DMSM})$.

8: Determine the scalar approximation $\gamma_{k+1}^{DMSM} I$ of the Hessian of f at the point $\mathbf{x}_{k+1}^{DMSM}$ using (2.10).

9: If $\gamma_{k+1}^{DMSM} < 0$, then take $\gamma_{k+1}^{DMSM} = 1$.

10: Put $k := k + 1$, go to the step 2.

11: Return $\mathbf{x}_{k+1}^{DMSM}$ and $f(\mathbf{x}_{k+1}^{DMSM})$.

3. Convergence analysis

The content of this section is the convergence analysis of the TMSM and DMSM methods. In the following part, we restate and derive some basic statements which will be used in the convergence analysis of Algorithms 4 and 5. The proofs can be found in [1, 9, 12, 13, 14] and have been omitted:

(H_1) the function f is bounded below on $B_0 = \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \le f(\mathbf{x}_0)\}$;

(H_2) the gradient $\mathbf{g}$ is Lipschitz continuous in an open convex set $B \supseteq B_0$:

$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|, \quad \forall\, \mathbf{x}, \mathbf{y} \in B,\ L > 0. \tag{3.1}$$

Proposition 3.1. [1, 13] *Let $\mathbf{d}_k$ be a descent direction and the gradient $\mathbf{g}_k$ satisfies the Lipschitz condition* (3.1). *If t_k is determined by the BLS in Algorithm 1, then*

$$t_k \ge \min\left\{1, -\frac{\beta(1-\sigma)}{L} \frac{\mathbf{g}_k^{\mathrm{T}} \mathbf{d}_k}{\|\mathbf{d}_k\|^2}\right\}. \tag{3.2}$$

Lemma 3.1. *If the function f is UC and TCD on $\mathbb{R}^n$ then there exist m, M such that*

$$0 < m \le 1 \le M, \tag{3.3}$$

then $f(\mathbf{x})$ *possesses a minimizer* $\mathbf{x}^*$ *and*

$$m\|\mathbf{y}\|^2 \le \mathbf{y}^{\mathrm{T}}\nabla^2 f(\mathbf{x})\mathbf{y} \le M\|\mathbf{y}\|^2, \quad \forall\ \mathbf{x},\mathbf{y} \in \mathbb{R}^n; \tag{3.4}$$

$$\frac{1}{2}m\|\mathbf{x}-\mathbf{x}^*\|^2 \le f(\mathbf{x}) - f(\mathbf{x}^*) \le \frac{1}{2}M\|\mathbf{x}-\mathbf{x}^*\|^2, \quad \forall\ \mathbf{x} \in \mathbb{R}^n; \tag{3.5}$$

$$m\|\mathbf{x}-\mathbf{y}\|^2 \le (\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y}))^{\mathrm{T}}(\mathbf{x}-\mathbf{y}) \le M\|\mathbf{x}-\mathbf{y}\|^2, \quad \forall\ \mathbf{x},\mathbf{y} \in \mathbb{R}^n. \tag{3.6}$$

Lemma 3.2. [14] *The following inequality holds for a TCD and UC function* f *and for the IGD sequence* $\{\mathbf{x}_k\}$ *generated by* (1.6)*:*

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \ge \mu\|\mathbf{g}_k\|^2, \tag{3.7}$$

with

$$\mu = \min\left\{\frac{\sigma}{M}, \frac{\sigma(1-\sigma)}{L}\beta\right\}. \tag{3.8}$$

In further, it is assumed in this section that $\mathbf{d}_k$ is a descent direction. Further, the scalar approximation of Hessian is TCD. Moreover, instead of (3.4) and (3.3) it is sufficient to assume:

$$m \le \gamma_k \le M, \quad 0 < m \le 1 \le M, \quad m, M \in \mathbb{R}. \tag{3.9}$$

So, all values $\gamma_k < 0$ will be replaced by $\gamma_k = 1$, while the cases $\gamma_k > M$ will be resolved by $\gamma_k = M$.

Theorem 3.1. *Let* (H_1) *and* (H_2) *and* (3.9) *be true and the mapping* f *is UC. Then the sequence* $\{\mathbf{x}_k^{DMSM}\}$ *fulfils* (3.7)–(3.8).

Proof. From (2.8), it can be concluded

$$\begin{aligned}\mathbf{x}_{k+1}^{DMSM} &= \mathbf{x}_k^{DMSM} - t_k^{DMSM}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k \\ &= \mathbf{x}_k^{DMSM} - t_k\frac{t_k^{DMSM}}{t_k}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k \\ &= \mathbf{x}_k^{DMSM} + t_k\mathbf{d}_k,\end{aligned}$$

where $\mathbf{d}_k = -\frac{t_k^{DMSM}}{t_k}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k$.

Based on the stopping condition of the backtracking line search (Algorithm 1), we conclude

$$f(\mathbf{x}_k^{DMSM}) - f(\mathbf{x}_{k+1}^{DMSM}) \ge -\sigma t_k \mathbf{g}_k{}^{\mathrm{T}}\mathbf{d}_k. \quad \forall\ k \in \mathbb{N}. \tag{3.10}$$

In the situation $t_k < 1$, by putting expression for $\mathbf{d}_k$ into (3.10), the following inequalities can be derived:

$$\begin{aligned}f(\mathbf{x}_k^{DMSM}) - f(\mathbf{x}_{k+1}^{DMSM}) &\ge -\sigma t_k \mathbf{g}_k{}^{\mathrm{T}}\mathbf{d}_k \\ &= -\sigma t_k \mathbf{g}_k{}^{\mathrm{T}}\left(-\frac{t_k^{DMSM}}{t_k}\left(\gamma_k^{DMSM}\right)^{-1}\mathbf{g}_k\right) \\ &= \sigma t_k \frac{t_k^{DMSM}}{t_k}\left(\gamma_k^{DMSM}\right)^{-1}\|\mathbf{g}_k\|^2.\end{aligned} \tag{3.11}$$

Now, from (3.2), it follows that

$$
\begin{aligned}
t_k &\geq -\frac{\beta(1-\sigma)}{L}\cdot\frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k}{\|\mathbf{d}_k\|^2}\\
&= -\frac{\beta(1-\sigma)}{L}\cdot\frac{\mathbf{g}_k^{\mathrm{T}}\left(-\frac{t_k^{DMSM}}{t_k}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k\right)}{\left\|-\frac{t_k^{DMSM}}{t_k}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k\right\|^2}\\
&= \frac{\beta(1-\sigma)}{L}\cdot\frac{\mathbf{g}_k{}^{\mathrm{T}}\frac{t_k^{DMSM}}{t_k}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k}{\left(\frac{t_k^{DMSM}}{t_k}\right)^2(\gamma_k^{DMSM})^{-2}\|\mathbf{g}_k\|^2}\\
&= \frac{\beta(1-\sigma)}{L}\cdot\frac{\|\mathbf{g}_k\|^2}{\frac{t_k^{DMSM}}{t_k}(\gamma_k^{DMSM})^{-1}\|\mathbf{g}_k\|^2}\\
&= \frac{(1-\sigma)\beta}{L}\cdot\frac{t_k\gamma_k^{DMSM}}{t_k^{DMSM}}.
\end{aligned}
\tag{3.12}
$$

By applying inequality (3.12) to (3.11), we obtain

$$
\begin{aligned}
&f(\mathbf{x}_k^{DMSM}) - f(\mathbf{x}_{k+1}^{DMSM}) \geq \sigma t_k\frac{t_k^{DMSM}}{t_k}(\gamma_k^{DMSM})^{-1}\|\mathbf{g}_k\|^2\\
&\geq \sigma\frac{(1-\sigma)\beta}{L}\cdot\frac{\gamma_k^{DMSM}}{\frac{t_k^{DMSM}}{t_k}}\frac{t_k^{DMSM}}{t_k}(\gamma_k^{DMSM})^{-1}\|\mathbf{g}_k\|^2\\
&\geq \sigma\frac{(1-\sigma)\beta}{L}\|\mathbf{g}_k\|^2.
\end{aligned}
\tag{3.13}
$$

In the case $t_k = 1$, based on (3.9) and (3.10) the following inequality holds

$$
\begin{aligned}
f(\mathbf{x}_k^{DMSM}) - f(\mathbf{x}_{k+1}^{DMSM}) &\geq -\sigma\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k\\
&= -\sigma\mathbf{g}_k^{\mathrm{T}}\left(-\frac{t_k^{DMSM}}{t_k}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k\right)\\
&= \frac{\sigma}{\gamma_k^{DMSM}}\frac{t_k^{DMSM}}{t_k}\|\mathbf{g}_k\|^2.
\end{aligned}
\tag{3.14}
$$

According to (2.9), it follows that $t_k^{DMSM} \geq t_k$, which implies

$$
\begin{aligned}
f(\mathbf{x}_k^{DMSM}) - f(\mathbf{x}_{k+1}^{DMSM}) &\geq \frac{\sigma}{\gamma_k^{DMSM}}\|\mathbf{g}_k\|^2\\
&\geq \frac{\sigma}{M}\|\mathbf{g}_k\|^2.
\end{aligned}
\tag{3.15}
$$

Finally, from (3.13) and (3.15) we get (3.8). □

Theorem 3.2. *Let* (H_1) *and* (H_2) *are valid in conjunction with* (3.9) *and* f *be a UC function.*
(a) *The sequence* $\{\mathbf{x}_k^{DMSM}\}$ *satisfies* $\lim\limits_{k\to\infty}\|\mathbf{g}_k\|=0$, *and* $\{\mathbf{x}_k^{DMSM}\}$ *converges to* $\mathbf{x}^*$.
(b) *The sequence* $\{\mathbf{x}_k^{TMSM}\}$ *satisfies* $\lim\limits_{k\to\infty}\|\mathbf{g}_k\|=0$, *and* $\{\mathbf{x}_k^{TMSM}\}$ *converges to* $\mathbf{x}^*$.

Proof. Analogously as the proof of [14, Theorem 4.1]. □

Lemma 3.3 confirms the convergence of the DMSM method on the strictly convex quadratic (SCQ) functions

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}A\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x}, \tag{3.16}$$

where $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix and $\mathbf{b} \in \mathbb{R}^n$. The eigenvalues of A are ordered as $\lambda_1 \leq \cdots \leq \lambda_n$.

Lemma 3.3. *The DMSM iterations* (2.8) *applied on a SCQ function* f *given by the expression* (3.16) *satisfy the inequality*

$$\lambda_1 \leq \frac{\gamma_{k+1}^{DMSM}}{t_{k+1}} \leq \frac{2\lambda_n}{\sigma}, k \in \mathbb{N}. \tag{3.17}$$

Proof. Simple verification gives

$$\begin{aligned} f(\mathbf{x}_{k+1}^{DMSM}) - f(\mathbf{x}_k^{DMSM}) =& \frac{1}{2}(\mathbf{x}_{k+1}^{DMSM})^{\mathrm{T}}A\mathbf{x}_{k+1}^{DMSM} - \mathbf{b}^{\mathrm{T}}\mathbf{x}_{k+1}^{DMSM} \\ &- \frac{1}{2}(\mathbf{x}_k^{DMSM})^{\mathrm{T}}A\mathbf{x}_k^{DMSM} + \mathbf{b}^{\mathrm{T}}\mathbf{x}_k^{DMSM}. \end{aligned} \tag{3.18}$$

The substitute of (2.8) in (3.18) gives

$$\begin{aligned} f(\mathbf{x}_{k+1}^{DMSM})-f(\mathbf{x}_k^{DMSM}) =& \frac{1}{2}\left[\mathbf{x}_k^{DMSM}-t_k^{DMSM}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k\right]^{\mathrm{T}} \\ &\times A\left[\mathbf{x}_k^{DMSM}-t_k^{DMSM}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k\right] \\ &- \mathbf{b}^{\mathrm{T}}[\mathbf{x}_k^{DMSM}-t_k^{DMSM}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k] \\ &- \frac{1}{2}(\mathbf{x}_k^{DMSM})^{\mathrm{T}}A\mathbf{x}_k^{DMSM}+\mathbf{b}^{\mathrm{T}}\mathbf{x}_k^{DMSM} \\ =& -\frac{1}{2}t_k^{DMSM}(\gamma_k^{DMSM})^{-1}(\mathbf{x}_k^{DMSM})^{\mathrm{T}}A\mathbf{g}_k \\ &- \frac{1}{2}t_k^{DMSM}(\gamma_k^{DMSM})^{-1}\,\mathbf{g}_k^{\mathrm{T}}A\mathbf{x}_k^{DMSM} \\ &+ \frac{1}{2}\left(t_k^{DMSM}\right)^2(\gamma_k^{DMSM})^{-2}\mathbf{g}_k^{\mathrm{T}}A\mathbf{g}_k \\ &+ t_k^{DMSM}(\gamma_k^{DMSM})^{-1}\mathbf{b}^{\mathrm{T}}\mathbf{g}_k. \end{aligned} \tag{3.19}$$

Since the gradient of the function (3.16) corresponding to DMSM is equal to

$$\mathbf{g}_k = A\mathbf{x}_k^{DMSM} - \mathbf{b}, \tag{3.20}$$

one can verify

$$\begin{aligned} f(\mathbf{x}_{k+1}^{DMSM}) - f(\mathbf{x}_k^{DMSM}) &= t_k^{DMSM}(\gamma_k^{DMSM})^{-1}[\mathbf{b}^{\mathrm{T}}\mathbf{g}_k - (\mathbf{x}_k^{DMSM})^{\mathrm{T}}A\mathbf{g}_k] \\ &\quad + \frac{1}{2}\left(t_k^{DMSM}\right)^2(\gamma_k^{DMSM})^{-2}\mathbf{g}_k^{\mathrm{T}}A\mathbf{g}_k \\ &= t_k^{DMSM}(\gamma_k^{DMSM})^{-1}[\mathbf{b}^{\mathrm{T}} - (\mathbf{x}_k^{DMSM})^{\mathrm{T}}A]\mathbf{g}_k \\ &\quad + \frac{1}{2}\left(t_k^{DMSM}\right)^2(\gamma_k^{DMSM})^{-2}\mathbf{g}_k^{\mathrm{T}}A\mathbf{g}_k \\ &= -t_k^{DMSM}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k \\ &\quad + \frac{1}{2}\left(t_k^{DMSM}\right)^2(\gamma_k^{DMSM})^{-2}\mathbf{g}_k^{\mathrm{T}}A\mathbf{g}_k. \end{aligned} \tag{3.21}$$

After substitute (3.21) into (2.10), the parameter γ_{k+1}^{DMSM} becomes

$$\begin{aligned} \gamma_{k+1}^{DMSM} &= 2\gamma_k^{DMSM} \frac{\gamma_k^{DMSM}\left[f(\mathbf{x}_{k+1}^{DMSM}) - f(\mathbf{x}_k^{DMSM})\right] + t_k^{DMSM}\|\mathbf{g}_k\|^2}{\left(t_k^{DMSM}\right)^2\|\mathbf{g}_k\|^2} \\ &= \frac{\mathbf{g}_k^{\mathrm{T}}A\mathbf{g}_k}{\|\mathbf{g}_k\|^2}. \end{aligned} \tag{3.22}$$

Therefore, the following inequalities are valid:

$$\lambda_1 \le \gamma_{k+1}^{DMSM} \le \lambda_n, \; k \in \mathbb{N}. \tag{3.23}$$

The inequality in (3.17) follows from (3.23) in conjunction with $0 < t_{k+1} \le 1$. In order to verify the right hand side inequality in (3.17), it suffices to observe the upper bound caused by the BLS

$$t_k \ge \frac{\beta(1-\sigma)\gamma_k}{L},$$

which implies

$$\frac{\gamma_{k+1}^{DMSM}}{t_{k+1}} < \frac{L}{\beta(1-\sigma)}. \tag{3.24}$$

Using $\mathbf{g}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ in common with the fact that A symmetric, it follows that

$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| = \|A\mathbf{x} - A\mathbf{y}\| = \|A(\mathbf{x} - \mathbf{y})\| \le \|A\|\|\mathbf{x} - \mathbf{y}\| = \lambda_n\|\mathbf{x} - \mathbf{y}\|. \tag{3.25}$$

The Lipschitz constant L in (3.24) can be equal to the largest eigenvalue λ_n. Using $\sigma \in (0, 0.5)$, $\beta \in (\sigma, 1)$ one obtains

$$\frac{\gamma_{k+1}^{DMSM}}{t_{k+1}} < \frac{L}{\beta(1-\sigma)} = \frac{\lambda_n}{\beta(1-\sigma)} < \frac{2\lambda_n}{\sigma}. \tag{3.26}$$

So, the right inequality in (3.17) is verified. □

Theorem 3.3. *Let f be a SCQ function defined in* (3.16)*. In the case $\lambda_n < 2\lambda_1$ the DMSM method* (2.8) *satisfies*

$$(d_i^{k+1})^2 \le \delta^2 (d_i^k)^2, \tag{3.27}$$

where

$$\delta = \max\left\{1 - \frac{\sigma\lambda_1}{2\lambda_n}, \frac{\lambda_n}{\lambda_1} - 1\right\}. \tag{3.28}$$

In addition,

$$\lim_{k\to\infty} \|\mathbf{g}_k\| = 0. \tag{3.29}$$

Proof. Let $\{v_1, \ldots, v_n\}$ be orthonormal eigenvectors of A. On the basis of (3.20), there exist real quantities $d_1^k, d_2^k, \ldots, d_n^k$ satisfying

$$\mathbf{g}_k = \sum_{i=1}^{n} d_i^k v_i. \tag{3.30}$$

Now, using (2.8) one can simply deduce

$$\begin{aligned}
\mathbf{g}_{k+1} &= A\mathbf{x}_{k+1}^{DMSM} - \mathbf{b} \\
&= A(\mathbf{x}_k^{DMSM} - t_k^{DMSM}(\gamma_k^{DMSM})^{-1}\mathbf{g}_k) - \mathbf{b} \\
&= \mathbf{g}_k - t_k^{DMSM}(\gamma_k^{DMSM})^{-1} A\mathbf{g}_k \\
&= \left(I - t_k^{DMSM}(\gamma_k^{DMSM})^{-1} A\right)\mathbf{g}_k.
\end{aligned}$$

Using the simple linear approximation of $\mathbf{g}_{k+1}$ as in (3.30), we get

$$\mathbf{g}_{k+1} = \sum_{i=1}^{n} d_i^{k+1} v_i = \sum_{i=1}^{n} \left(1 - t_k^{DMSM}(\gamma_k^{DMSM})^{-1}\lambda_i\right) d_i^k v_i. \tag{3.31}$$

To prove (3.27), it is enough to show that $\left|1 - \frac{\lambda_i}{\gamma_k^{DMSM}\left(t_k^{DMSM}\right)^{-1}}\right| \le \delta$. Two cases can be observed. First, if $\lambda_i \le \frac{\gamma_k^{DMSM}}{t_k^{DMSM}}$ implying (3.17), we can conclude the following:

$$1 > \frac{\lambda_i}{\gamma_k^{DMSM}\left(t_k^{DMSM}\right)^{-1}} \ge \frac{\sigma\lambda_1}{2\lambda_n} \Longrightarrow 1 - \frac{\lambda_i}{\gamma_k^{DMSM}\left(t_k^{DMSM}\right)^{-1}} \le 1 - \frac{\sigma\lambda_1}{2\lambda_n} \le \delta. \tag{3.32}$$

Now, let us examine another case $\frac{\gamma_k^{DMSM}}{t_k^{DMSM}} < \lambda_i$. Since

$$1 < \frac{\lambda_i}{\gamma_k^{DMSM}\left(t_k^{DMSM}\right)^{-1}} \le \frac{\lambda_n}{\lambda_1}, \tag{3.33}$$

it follows that

$$(3.34)\qquad \left|1-\frac{\lambda_i}{\gamma_k^{DMSM}\left(t_k^{DMSM}\right)^{-1}}\right| \le \frac{\lambda_n}{\lambda_1}-1\le\delta.$$

Now, in order to prove $\lim\limits_{k\to\infty}\|\mathbf{g}_k\| = 0$, it suffices to use the orthonormality of $\{v_1,\dots,v_n\}$ in common with (3.30) and conclude

$$(3.35)\qquad \|\mathbf{g}_k\|^2=\sum_{i=1}^{n}(d_i^k)^2.$$

Since (3.27) is valid and $0<\delta<1$ holds, (3.35) initiates that (3.30). □

4. Numerical results

All the considered methods are coded in Matlab R2017a programming language and executed on the notebook with Intel Core i3 2.0 GHz CPU, 8 GB RAM and Windows 10 operating system. The number of iterations (NofI), number of function evaluations (NofFE) and the CPU time (CPUT) are analyzed in numerical experiments.

Numerical testing is based on 24 test functions from [2], where a lot of the problems are taken over from CUTEr collection [4]. For each of tested functions in Tables 4.1, 4.2 and 4.3, 12 numerical testings are performed with 100, 200, 300, 500, 1000, 2000, 3000, 5000, 7000, 8000, 10000 and 15000 unknowns. Tables 4.1, 4.2 and 4.3 arrange summary numerical results for AGD, MAGD, MSM, SM, DMSM and TMSM, tested on 24 functions.

For each of six tested methods (AGD, MAGD, SM, MSM, DMSM and TMSM), the same stopping criteria are used:

$$\|\mathbf{g}_k\|\le 10^{-6}\quad\text{and}\quad \frac{|f(\mathbf{x}_{k+1})-f(\mathbf{x}_k)|}{1+|f(\mathbf{x}_k)|}\le 10^{-16}.$$

The BLS parameters for AGD, MAGD, MSM and SM methods are $\sigma=0.0001$ and $\beta=0.8$. The backtracking procedures in the DMSM method are implemented using $\sigma=0.0001$ and $\beta=0.8$ for Algorithm 1 and $\sigma_j=0.00015$ and $\beta_j=0.85$ for Algorithm 3.

The backtracking procedures in the TMSM method are developed using $\sigma=0.0001$ and $\beta=0.8$ for Algorithm 1, $\sigma_l=0.0002$ and $\beta_l=0.9$ for Algorithm 2 and $\sigma_j=0.00015$ and $\beta_j=0.85$ for Algorithm 3.

Table 4.4 contains average values of NofI, the NofFE and the CPUT for all 288 numerical experiments.

Based on the values for NofI given in Table 4.4, it can be concluded that the DMSM and TMSM methods gives superior results with respect to MAGD, AGD, MSM and SM methods.

Table 4.1: Numerical results of the AGD, MAGD, MSM, SM, DMSM and TMSM methods for the NofI.

Test function	*MAGD*	*TMSM*	*MSM*	*DMSM*	*SM*	*AGD*
Perturbed Quadratic	352325	31269	34828	31386	59908	353897
Raydan 1	58504	30148	26046	17238	14918	22620
Diagonal 3	119719	6767	7030	7077	12827	120416
Generalized Tridiagonal 1	647	332	346	350	325	670
Extended Tridiagonal 1	692219	685	1370	728	4206	3564
Extended TET	455	191	156	156	156	443
Diagonal 4	8084	96	96	96	96	120
Diagonal 5	48	72	72	72	72	48
Extended Himmelblau	302	312	260	264	196	396
Perturbed quadratic diagonal	1060824	36640	37454	31662	44903	2542050
Quadratic QF1	362896	32099	36169	33138	62927	366183
Extended quadratic penalty QP1	229	338	369	298	271	210
Extended quadratic penalty QP2	356357	1735	1674	990	3489	395887
Quadratic QF2	71647	31745	32727	30642	64076	100286
Extended Tridiagonal 2	1665	694	659	583	543	1657
ARWHEAD (CUTE)	12834	328	430	302	270	5667
Almost Perturbed Quadratic	354369	30790	33652	32902	60789	356094
LIARWHD (CUTE)	925138	1257	3029	1726	18691	1054019
ENGVAL1 (CUTE)	822	623	461	434	375	743
QUARTC (CUTE)	177	302	217	220	290	171
Generalized Quartic	229	191	181	186	189	187
Diagonal 7	159	144	147	111	108	72
Diagonal 8	154	120	120	109	118	60
Full Hessian FH3	63	63	63	63	63	45

Performance profiles from [6] are used in comparing the selected methods. As usual, the NofI, NofFE and CPUT profiles are used. All numerical results are represented in Figures 4.1 and 4.2. Figure 4.1 (left) shows the performances of compared methods related to NofI. Figure 4.1 (right) illustrates the performance of these methods relative to NofFE. Graphs in Figure 4.2 illustrate the behavior of considered methods with respect to to CPUT.

From the results displayed in Tables 4.1, 4.2 and 4.3 and according to graphs in Figures 4.1 and Figure 4.2, the following can be observed.

(1) The DMSM and TMSM methods give better results compared to other methods when we compare the number of iterations.

(2) The SM, MSM, DMSM and TMSM exhibit better performances than the AGD and MAGD methods.

From Figure 4.1 (left), it is observable that the graph of the $DMSM$ method comes first to the top, which signifies that the DMSM outperforms other considered methods with respect to the NofI.

Table 4.2: Numerical results of the AGD, MAGD, MSM, SM, DMSM and TMSM methods for the NofFE.

Test function	$MAGD$	$TMSM$	MSM	$DMSM$	SM	AGD
Perturbed Quadratic	13855459	645704	200106	370595	337910	13916515
Raydan 1	1282162	1305952	311260	326766	81412	431804
Diagonal 3	4244404	131307	38158	80193	69906	4264718
Generalized Tridiagonal 1	9057	2934	1191	2061	1094	9334
Extended Tridiagonal 1	2077341	14797	10989	9147	35621	14292
Extended TET	4130	1689	528	948	528	3794
Diagonal 4	133440	2316	636	1320	636	1332
Diagonal 5	108	300	156	228	156	108
Extended Himmelblau	5192	3636	976	1908	668	6897
Perturbed quadratic diagonal	38728371	1309740	341299	629088	460028	94921578
Quadratic QF1	13192789	661661	208286	392426	352975	13310016
Extended quadratic penalty QP1	2939	6400	2196	5421	2326	2613
Extended quadratic penalty QP2	8846145	44962	11491	14058	25905	9852040
Quadratic QF2	2810965	642829	183142	364257	353935	3989239
Extended Tridiagonal 2	9613	9779	2866	4951	2728	8166
ARWHEAD (CUTE)	468970	15416	5322	8503	3919	214284
Almost Perturbed Quadratic	13936462	639129	194876	393591	338797	14003318
LIARWHD (CUTE)	41619197	39788	27974	33271	180457	47476667
ENGVAL1 (CUTE)	8332	10120	2285	4319	2702	6882
QUARTC (CUTE)	414	1412	494	780	640	402
Generalized Quartic	1244	1311	493	836	507	849
Diagonal 7	745	930	504	696	335	333
Diagonal 8	740	805	383	546	711	304
Full Hessian FH3	1955	2160	566	1263	631	1352

Figure 4.1 (right) confirms that all six methods are able to solve all test cases. Further, the MSM method is superior in 58.33% of all tests with respect to MAGD (4.17%), TMSM(0%), DMSM(4.17%), SM(29.17%) and AGD(16.67%).

Graphs in Figure 4.2 again confirm that all the methods are able to solve test problems, and the MSM is winer in 54.17% of the tests with respect to MAGD (4.17%), TMSM(0%), DMSM(4.17%), SM(37.50%) and AGD(4.17%).

According to individual data arranged in the tables 4.1-4.3, generated average values as well as the presented graphs, the conclusion is that the DMSM method is winer concerning the NofI.

Compared to the previous numerical results obtained during the testing of AGD, MAGD, MSM, SM, DMSM and TMSM methods, in the next test for parameter values in the second and third backtracking line search we take the values that are less than the values in primary backtracking. The aim of this test is to answer the question: Does the choice of higher or lower parameter values in the second and third backtracking line search in relation to the primary backtracking line search

Table 4.3: Numerical results of the AGD, MAGD, MSM, SM, DMSM and TMSM methods for the CPUT.

Test function	$MAGD$	$TMSM$	MSM	$DMSM$	SM	AGD
Perturbed Quadratic	6049.531	344.172	116.281	198.328	185.641	6756.047
Raydan 1	334.266	388.156	31.906	67.344	36.078	158.359
Diagonal 3	6401.969	199.547	52.609	120.406	102.875	5527.844
Generalized Tridiagonal 1	7.781	4.641	1.469	3.625	1.203	11.344
Extended Tridiagonal 1	8853.172	26.828	29.047	17.297	90.281	55.891
Extended TET	2.766	1.703	0.516	1.203	0.594	3.219
Diagonal 4	16.172	0.719	0.203	0.359	0.141	0.781
Diagonal 5	0.313	0.750	0.344	0.734	0.328	0.391
Extended Himmelblau	1.031	1.094	0.297	0.703	0.188	1.953
Perturbed quadratic diagonal	22820.172	534.750	139.625	273.188	185.266	44978.750
Quadratic QF1	6846.453	258.938	81.531	168.453	138.172	12602.563
Extended quadratic penalty QP1	1.063	2.234	1.000	3.516	0.797	1.266
Extended quadratic penalty QP2	1872.797	12.578	3.516	8.063	6.547	3558.734
Quadratic QF2	768.563	243.938	73.438	153.109	132.703	1582.766
Extended Tridiagonal 2	2.531	4.938	1.047	2.375	1.031	3.719
ARWHEAD (CUTE)	138.000	6.422	1.969	4.609	1.359	95.641
Almost Perturbed Quadratic	7086.563	285.563	73.047	153.891	133.516	13337.125
LIARWHD (CUTE)	15372.625	10.203	9.250	12.641	82.016	27221.516
ENGVAL1 (CUTE)	2.641	4.328	1.047	2.375	1.188	3.906
QUARTC (CUTE)	2.078	4.531	1.844	3.297	2.313	2.469
Generalized Quartic	0.500	0.734	0.281	0.375	0.188	0.797
Diagonal 7	0.688	0.953	0.547	1.469	0.375	0.625
Diagonal 8	0.656	0.781	0.469	1.078	0.797	0.438
Full Hessian FH3	1.188	1.672	0.391	1.234	0.391	1.438

Table 4.4: Average numerical outcomes for 24 test functions tested on 12 numerical experiments.

Average performances	MAGD	TMSM	MSM	DMSM	SM	AGD
Number of iterations	182494.42	8622.54	9064.83	7947.21	14575.25	221896.04
No. of fun.evaluation	5885007.25	228961.54	64424.04	110298.83	93938.63	8434868.21
CPU time (sec)	3190.98	97.51	25.90	49.99	46.00	4829.4

directly affect the numerical results of DMSM and TMSM methods?

The primary BLS uses the same parameters $\sigma = 0.0001$ and $\beta = 0.8$ as in the first test for AGD, MAGD, MSM and SM methods. The BLS procedures in the DMSM method are implemented using $\sigma = 0.0001$ and $\beta = 0.8$ for Algorithm 1 and $\sigma_j = 0.00005$ and $\beta_j = 0.7$ for Algorithm 3. Also, the BLS in the TMSM method are implemented using $\sigma = 0.0001$ and $\beta = 0.8$ for Algorithm 1, $\sigma_l = 0.00001$ and $\beta_l = 0.6$

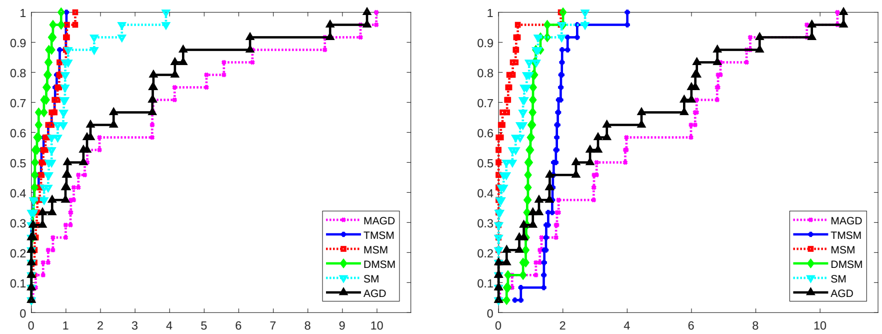

Fig. 4.1: Performance profiles based on the NofI (left) and NofFE (right).

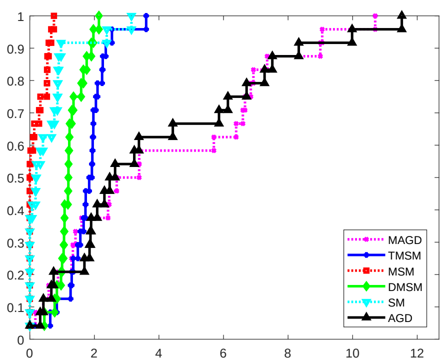

Fig. 4.2: Performance profiles based upon CPUT.

for Algorithm 2 and $\sigma_j = 0.00005$ and $\beta_j = 0.7$ for Algorithm 3.

All other conditions (stop criteria and number of variables) remain the same as in the first numerical experiment.

The obtained numerical results are shown in the Tables 4.5, 4.6 and 4.7.

Table 4.8 includes the average values of NofI, the NofFE and the CPUT in a second numerical experiment.

According to the NofI values given in Table 4.8, it can be notified that the DMSM method gives better results and in the second numerical experiment compared to MAGD, AGD, MSM, SM and TMSM methods.

All numerical results from Tables 4.5, 4.6 and 4.7 are represented in Figures 4.3 and 4.4. Figure 4.3 (left) shows the NofI performances of compared methods. Figure 4.3 (right) demonstrates the NofFE profile of these methods. Figure 4.4

Table 4.5: Numerical results of the AGD, MAGD, MSM, SM, DMSM and TMSM methods for the NofI.

Test function	$MAGD$	MSM	SM	AGD	$TMSM$	$DMSM$
Perturbed Quadratic	352325	34828	59908	353897	35697	28487
Raydan 1	58504	26046	14918	22620	9801	17594
Diagonal 3	119719	7030	12827	120416	8372	6409
Generalized Tridiagonal 1	647	346	325	670	342	348
Extended Tridiagonal 1	692219	1370	4206	3564	907	760
Extended TET	455	156	156	443	156	156
Diagonal 4	8084	96	96	120	96	96
Diagonal 5	48	72	72	48	72	72
Extended Himmelblau	302	260	196	396	288	294
Perturbed quadratic diagonal	1060824	37454	44903	2542050	31031	37331
Quadratic QF1	362896	36169	62927	366183	39619	26585
Extended quadratic penalty QP1	229	369	271	210	303	362
Extended quadratic penalty QP2	356357	1674	3489	395887	2047	1908
Quadratic QF2	71647	32727	64076	100286	39452	28651
Extended quadratic exponential EP1	67	100	73	48	107	107
Extended Tridiagonal 2	1665	659	543	1657	528	615
ARWHEAD (CUTE)	12834	430	270	5667	304	281
Almost Perturbed Quadratic	354369	33652	60789	356094	35755	26274
LIARWHD (CUTE)	925138	3029	18691	1054019	1340	3543
ENGVAL1 (CUTE)	822	461	375	743	418	482
QUARTC (CUTE)	177	217	290	171	289	275
Generalized Quartic	229	181	189	187	197	195
Full Hessian FH3	63	63	63	45	63	63
Diagonal 9	325609	10540	13619	329768	10219	11229

shows the performance CPUT.

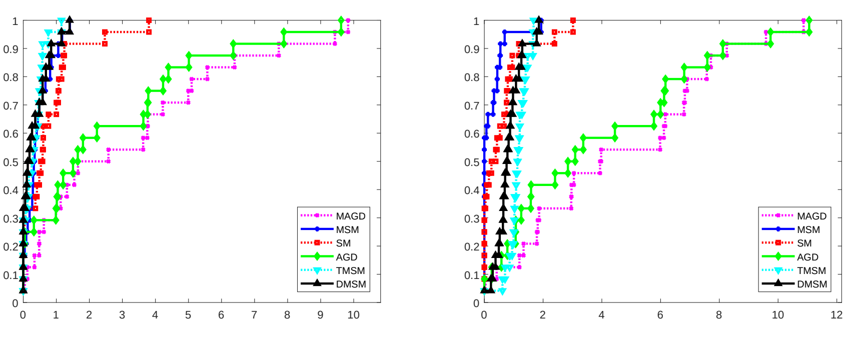

FIG. 4.3: Performance profiles based on the NofI (left) and NofFE (right).

Table 4.6: Numerical results of the AGD, MAGD, MSM, SM, DMSM and TMSM methods for the NofFE.

Test function	$MAGD$	MSM	SM	AGD	$TMSM$	$DMSM$
Perturbed Quadratic	13855459	200106	337910	13916515	423496	260678
Raydan 1	1282162	311260	81412	431804	124905	280011
Diagonal 3	4244404	38158	69906	4264718	95962	54865
Generalized Tridiagonal 1	9057	1191	1094	9334	2408	2153
Extended Tridiagonal 1	2077341	10989	35621	14292	13562	6800
Extended TET	4130	528	528	3794	1080	828
Diagonal 4	133440	636	636	1332	1284	996
Diagonal 5	108	156	156	108	300	228
Extended Himmelblau	5192	976	668	6897	2136	2418
Perturbed quadratic diagonal	38728371	341299	460028	94921578	619938	529154
Quadratic QF1	13192789	208286	352975	13310016	472273	243573
Extended quadratic penalty QP1	2939	2196	2326	2613	5073	3895
Extended quadratic penalty QP2	8846145	11491	25905	9852040	29847	21345
Quadratic QF2	2810965	183142	353935	3989239	444580	257674
Extended quadratic exponential EP1	1513	894	661	990	2083	1617
Extended Tridiagonal 2	9613	2866	2728	8166	4446	4456
ARWHEAD (CUTE)	468970	5322	3919	214284	9038	6761
Almost Perturbed Quadratic	13936462	194876	338797	14003318	424470	237534
LIARWHD (CUTE)	41619197	27974	180457	47476667	22254	53306
ENGVAL1 (CUTE)	8332	2285	2702	6882	6064	4442
QUARTC (CUTE)	414	494	640	402	1264	909
Generalized Quartic	1244	493	507	849	1043	798
Full Hessian FH3	1955	566	631	1352	1152	957
Diagonal 9	12984028	68189	89287	13144711	131327	125119

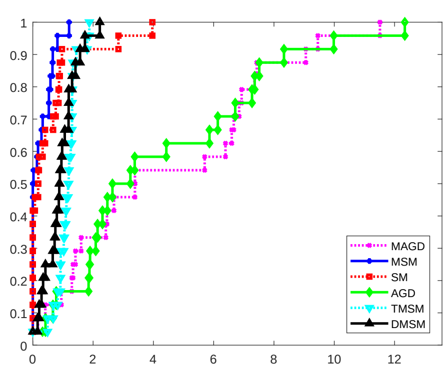

FIG. 4.4: Performance profiles arising from CPUT.

Table 4.7: Numerical results of the AGD, MAGD, MSM, SM, DMSM and TMSM methods for the CPUT.

Test function	$MAGD$	MSM	SM	AGD	$TMSM$	$DMSM$
Perturbed Quadratic	6049.531	116.281	185.641	6756.047	219.328	134.781
Raydan 1	334.266	31.906	36.078	158.359	44.828	66.484
Diagonal 3	6401.969	52.609	102.875	5527.844	129.734	96.688
Generalized Tridiagonal 1	7.781	1.469	1.203	11.344	2.969	2.969
Extended Tridiagonal 1	8853.172	29.047	90.281	55.891	25.672	12.609
Extended TET	2.766	0.516	0.594	3.219	1.234	0.938
Diagonal 4	16.172	0.203	0.141	0.781	0.344	0.172
Diagonal 5	0.313	0.344	0.328	0.391	0.594	0.516
Extended Himmelblau	1.031	0.297	0.188	1.953	0.688	0.875
Perturbed quadratic diagonal	22820.172	139.625	185.266	44978.750	263.953	220.719
Quadratic QF1	6846.453	81.531	138.172	12602.563	173.953	91.047
Extended quadratic penalty QP1	1.063	1.000	0.797	1.266	2.781	1.813
Extended quadratic penalty QP2	1872.797	3.516	6.547	3558.734	8.750	5.906
Quadratic QF2	768.563	73.438	132.703	1582.766	169.266	98.141
Extended quadratic exponential EP1	0.844	0.688	0.438	0.750	1.000	0.859
Extended Tridiagonal 2	2.531	1.047	1.031	3.719	1.828	1.922
ARWHEAD (CUTE)	138.000	1.969	1.359	95.641	2.813	2.625
Almost Perturbed Quadratic	7086.563	73.047	133.516	13337.125	158.156	92.578
LIARWHD (CUTE)	15372.625	9.250	82.016	27221.516	5.250	17.406
ENGVAL1 (CUTE)	2.641	1.047	1.188	3.906	2.578	2.391
QUARTC (CUTE)	2.078	1.844	2.313	2.469	4.625	3.203
Generalized Quartic	0.500	0.281	0.188	0.797	0.422	0.500
Full Hessian FH3	1.188	0.391	0.391	1.438	1.063	0.891
Diagonal 9	6662.984	43.609	38.672	6353.172	61.984	114.703

Table 4.8: Average numerical results in the second numerical experiment.

Average performances	$MAGD$	MSM	SM	AGD	$TMSM$	$DMSM$
Number of iterations	196051.21	9497.04	15136.33	235632.88	9058.46	8004.88
No. of fun.evaluation	6426009.58	67265.54	97642.88	8982579.21	118332.71	87521.54
CPU time (sec)	3468.58	27.71	47.58	5094.18	53.49	40.45

In accordance with obtained numerical data generated in the second numerical experiment, we can give an answer to the question, that independently of the choice of parameter values in the second and third backtracking line search, the DMSM iterations has the best results in relation to NofI. Also, if we compare the average results obtained in Tables 4.4 and 4.8, we can see that there is a slight percentage decrease in the average numerical results of the NofFE and CPUT, the DMSM method compared to the MSM method in the second numerical experiment.

5. Conclusion

Multiple usage of the backtracking line search in the modified SM (MSM) method lead to two improvements of the MSM scheme, denoted as the TMSM and DMSM methods. Proposed iterations are investigated both theoretically and numerically. The linear convergence of the defined model is proved for UC and for a subset of SCQ functions. Numerical experiments confirm that the derived TMSM and DMSM methods outperform the SM, AGD, MAGD and the MSM with respect to the number of iterations. Numerical values arranged in Tables 4.1-4.8 confirm the better performance of presented accelerated gradient descent method. Finally, the obtained TMSM and DMSM methods can be used as a motivation for different possibilities of deriving new efficient schemes for unconstrained optimization.

REFERENCES

1. N. Andrei: *An acceleration of gradient descent algorithm with backtracking for unconstrained optimization*. Numer. Algor. **42** (2006), 63–73.
2. N. Andrei: *An unconstrained optimization test functions collection*. Adv. Model. Optim. **10** (2008), 147—161.
3. N. Andrei: *Relaxed gradient descent and a new gradient descent methods for unconstrained optimization*. https://camo.ici.ro/neculai/newgrad.pdf, Visited October 29, 2020.
4. I. Bongartz, A. R. Conn, N. I. M. Gould and Ph. L. Toint: *CUTEr: constrained and unconstrained testing environments*. ACM Trans. Math. Softw. **21** (1995), 123-–160.
5. C. Brezinski: *A classification of quasi-Newton methods*. Numer. Algor. **33** (2003), 123-–135.
6. E. D. Dolan and J. J. Moré: *Benchmarking optimization software with performance profiles*. Math. Program. **91** (2002), 201—213.
7. B. Ivanov, P. S. Stanimirović, G. V. Milovanović, S. Djordjević and I. Brajević: *Accelerated multiple step-size methods for solving unconstrained optimization problems*. Optimization Methods and Software (2019), https://doi.org/10.1080/10556788.2019.1653868.
8. J. Nocedal and S. J. Wright: *Numerical Optimization*. Springer-Verlag New York, Inc, 1999.
9. J. M. Ortega and W. C. Rheinboldt: *Iterative Solution of Nonlinear Equation in Several Variables*. Academic Press, New York, London, 1970.
10. M. J. Petrović: *An accelerated Double Step Size method in unconstrained optimization*. Applied Math. Comput. **250** (2015), 309–319.
11. M. J. Petrović and P. S. Stanimirović: *Accelerated Double Direction method for solving unconstrained optimization problems*. Mathematical Problems in Engineering **2014** (2014), Article ID 965104, 8 pages.
12. R. T. Rockafellar: *Convex Analysis*. Princeton University Press, Princeton, 1970.
13. Z. -J. Shi: *Convergence of line search methods for unconstrained optimization*. App. Math. Comput. **157** (2004), 393–405.
14. P. S. Stanimirović and M. B. Miladinović: *Accelerated gradient descent methods with line search*. Numer. Algor. **54** (2010), 503–520.

15. P. S. Stanimirović, G. V. Milovanović and M. J. Petrović: *A transformation of accelerated double step size method for unconstrained optimization*. Mathematical Problems in Engineering **2015** (2015), Article ID 283679, 8 pages.

16. W. Sun and Y. -X. Yuan: *Optimization Theory and Methods: Nonlinear Programming*. Springer, Berlin, 2006.

Branislav Ivanov
University of Belgrade, Technical Faculty in Bor
Department of Management
Vojske Jugoslavije 12, 19210 Bor, Serbia
ivanov.branislav@gmail.com

Bilall I. Shaini
State University of Tetova
Rr. e Ilindenit, p.n., Tetovo
R. Macedonia
bilall.shaini@unite.edu.mk

Predrag S. Stanimirović
University of Niš, Faculty of Sciences and Mathematics
Department of Computer Science
Višegradska 33, 18000 Niš, Serbia
pecko@pmf.ni.ac.rs