Iranian Journal of Mathematical Sciences and Informatics
Vol. 16, No. 1 (2021), pp 15-33
DOI: 10.29252/ijmsi.16.1.15

A Trust-region Method Using Extended Nonmonotone Technique for Unconstrained Optimization

Morteza Kimiaei[a*], Hamid Esmaeili[b], Farzad Rahpeymaii[c]

[a]Fakultät für Mathematik, Universität Wien, Oskar-Morgenstern-Platz 1, A-1090 Wien.
[b]Department of Mathematics, Bu-Ali Sina University, Hamedan, Iran.

[c]Department of Mathematics, Payame Noor University, PO BOX 19395–3697, Tehran, Iran.

E-mail: kimiaeim83@univie.ac.at
E-mail: esmaeili47@yahoo.com
E-mail: rahpeyma_83@yahoo.com

Abstract. In this paper, we present a nonmonotone trust-region algorithm for unconstrained optimization. We first introduce a variant of the nonmonotone strategy proposed by Ahookhosh & Amini [1] and incorporate it into the trust-region framework to construct a more efficient approach. Our new nonmonotone strategy combines the current function value with the maximum function values in some prior successful iterates. For iterates far away from the optimizer, we give a very strong nonmonotone strategy. In the vicinity of the optimizer, we have a weaker nonmonotone strategy. It leads to a medium nonmonotone strategy when iterates are not far away from or close to the optimizer. Theoretical analysis indicates that the new approach converges globally to a first-order critical point under classical assumptions. In addition, the local convergence is studied. Extensive numerical experiments for unconstrained optimization problems are reported showing that the new algorithm is robust and efficient.

Keywords: Unconstrained optimization, Trust-region framework, Nonmonotone technique, Theoretical convergence.
2010 Mathematics subject classification: 90-08, 90C26, 90C06.

*Corresponding Author

Received 22 October 2017; Accepted 07 December 2019

1. Introduction

This paper considers the unconstrained nonlinear optimization problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathbb{R}^n, \end{array} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. Many techniques are available to solve the problem (1.1). Two important classes of these methods are line-search methods and trust-region methods. In the simplest form, line search methods produce the new point $x_{k+1} := x_k + \alpha_k d_k$ for where α_k is a step-size and d_k is a search direction, whereas trust-region methods generate a trial step d_k by computing an exact or an approximate solution of the following subproblem

$$\begin{array}{ll} \min & m_k(x_k + d) := f_k + g_k^T d + \frac{1}{2} d^T B_k d \\ \text{s.t.} & d \in \mathbb{R}^n \text{ and } \|d\| \leq \delta_k. \end{array} \tag{1.2}$$

Here $\|\cdot\|$ denotes the Euclidean norm, $f_k := f(x_k)$, $g_k := \nabla f(x_k)$, B_k is Hessian $G_k := \nabla^2 f(x_k)$ or its symmetric approximation, and δ_k is a trust-region radius. The ratio

$$r_k := \frac{f_k - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)}, \tag{1.3}$$

plays a key role in the traditional trust-region framework. The model matches the original problem better at the current iterate x_k whenever r_k is sufficiently close to 1, which means there is a good agreement between the model and the objective function and we can expand the trust-region for the next step. Otherwise, there is not a good agreement between the model and the objective function, so we shrink the trust-region and the subproblem (1.2) is solved in the reduced region, cf. [31].

It is well-known that the traditional optimization approaches generally need to use a globalization technique such as line search or trust-region to guarantee the global convergence of the algorithm. These globalization techniques mostly enforce a monotonicity $f_{k+1} \leq f_k$ to the produced sequence of the objective function values, usually leading to slow convergence, see [1, 2, 5, 6, 11, 14, 17, 18, 19, 21, 23, 35]. To avoid this drawback of globalization techniques, Grippo et al. [17] introduced a nonmonotone strategy for unconstrained optimization problems. In particular, they modified the Armijo rule as

$$f(x_k + \alpha_k d_k) \leq f_{l(k)} + \delta \alpha_k g_k^T d_k, \tag{1.4}$$

where $\delta \in (0, 1)$ and

$$f_{l(k)} = \max_{0 \leq j \leq n(k)} \{f_{k-j}\}, \quad k \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}, \tag{1.5}$$

in which $n(0) = 0$ and $0 \leq n(k) \leq \min\{n(k-1) + 1, N\}$ with $N \geq 0$. The theoretical and numerical results show that the new technique has remarkable positive effects on Armijo-type line searches to get a faster global convergence especially for highly nonlinear problems. These excellent results attract many researchers to investigate more about the effects of these strategies in a wide variety of optimization procedures, see [1, 5, 6, 35]. As a prominent example, the first use of nonmonotone techniques in trust-region framework was introduced and analyzed by Deng et al. in [11]. Recently, Ahookhosh & Amini [1] and Ahookhosh et al. [5] introduced a new nonmonotone strategy and applied it to both the trust-region and line search schemes for unconstrained optimization. These techniques employ the nonmonotone term

$$R_k = \eta_k f_{l(k)} + (1 - \eta_k) f_k, \tag{1.6}$$

where $\eta_k \in [\eta_{\min}, \eta_{\max}]$, $\eta_{\min} \in [0, 1)$ and $\eta_{\max} \in [\eta_{\min}, 1]$. It is clear that the nonmonotonicity of R_k can be adjusted by selecting an adaptive process for η_k such that it makes R_k more relaxed for practical usage.

Although the nonmonotone technique (1.6) has good convergence results, it suffers from some difficulties as follow:

- Whenever η_k is close to 1, for some $k \in \mathbb{N}_0$, and iterates are far away from the optimizer, R_k augments the effect of $f_{l(k)}$ and can not prevent resolving the trust-region subproblem;
- Whenever iterates are not close to the optimizer, it is possible for the sequence $\{\eta_k\}_{k\geq 0}$ to quickly converge to a very small positive number. This means that R_k augments the effect of f_k and may lead to reject the current trial step;
- Regarding the above disadvantages, computational cost for solving the problem will be increased.

In this paper, we propose a new method to solve the problem (1.1), based on a new nonmonotone technique, and establish its global convergence to first-order critical points together with local superlinear and quadratic convergence rates. The preliminary numerical results exhibit the efficiency of the proposed method for unconstrained optimization problems.

This work is organized as follows. In Section 2, we describe a new nonmonotone trust-region algorithm and explain some of its properties. In Section 3, we prove that the proposed algorithm is globally convergent. The numerical results are reported in Section 4. Finally, some conclusions are expressed in Section 5.

2. Motivation and Algorithmic Structure

In this section, a novel nonmonotone trust-region strategy is presented. After proposing a new nonmonotone technique, we incorporate it into trust-region framework to construct a more effective procedure to solve the problem (1.1).

It is well-known that the best convergence results are obtained by stronger nonmonotone strategy whenever iterates are far away from the optimizer and weaker nonmonotone strategy for iterates close to the optimizer, see [1, 2, 3, 4, 5, 6, 7, 35]. We believe that the nonmonotone strategy (1.6) does not show an appropriate behavior when iterates are far away from the optimizer because it does not permit the parameter η_k to be greater than 1. Therefore, we define

$$\widehat{R}_k := \widehat{\eta}_k f_{l(k)} + (1 - \widehat{\eta}_k) f_k, \tag{2.1}$$

where

$$\widehat{\eta}_k := \begin{cases} \eta_k \Big| \dfrac{f_{l(k)}}{f_k} \Big|, & \text{if } f_k \neq 0, \\ \eta_k, & \text{otherwise}, \end{cases} \tag{2.2}$$

$\eta_k \in [\eta_{\min}, \eta_{\max}]$, $\eta_{\min} \in [0, 1)$ and $\eta_{\max} \in [\eta_{\min}, 1]$. On the basis of this nonmonotone strategy, we can replace the ratio (1.3) by

$$\widehat{r}_k := \frac{\widehat{R}_k - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)}, \tag{2.3}$$

which has three advantages:

- Whenever iterates are far away from the optimizer, for some $k \in \mathbb{N}_0$, the ratio $|f_{l(k)}/f_k|$ may be greater than 1, i.e., some elements of $\{\widehat{\eta}_k\}_{k\geq 0}$ may be greater than 1, too. Hence, the elements of $\{\widehat{R}_k\}_{k\geq 0}$, for some $k \in \mathbb{N}_0$, may be greater than those of $\{f_{l(k)}\}_{k\geq 0}$. Consequently, if iterates are far away from the optimizer, the sequence $\{\widehat{R}_k\}_{k\geq 0}$, for some $k \in \mathbb{N}_0$, provides a very stronger nonmonotone strategy that can augment the effect of $f_{l(k)}$.

- Whenever iterates are close to the optimizer, the ratio $|f_{l(k)}/f_k|$, for some $k \in \mathbb{N}_0$, may be smaller than 1, i.e., the elements of $\{\widehat{\eta}_k\}_{k\geq 0}$, for some $k \in \mathbb{N}_0$, may be smaller than 1, too. Therefore, the elements of $\{\widehat{R}_k\}_{k\geq 0}$, for some $k \in \mathbb{N}_0$, may be smaller than those of $\{R_k\}_{k\geq 0}$ providing a weaker nonmonotone strategy.

- Whenever iterates are not very close to the optimizer, η_k may lead to $\widehat{\eta}_k < 1$, for some $k \in \mathbb{N}_0$. Consequently, the elements of $\{\widehat{R}_k\}_{k\geq 0}$ locate between the elements of $\{R_k\}_{k\geq 0}$ and $\{f_{l(k)}\}_{k\geq 0}$, for some $k \in \mathbb{N}_0$, and we obtain a medium nonmonotone strategy.

- If iterates are close to the optimizer, $\{\widehat{\eta}_k\}_{k\geq 0}$, for some $k \in \mathbb{N}_0$, is equal to $\{\eta_k\}_{k\geq 0}$, i.e., $\widehat{R}_k$ is equal to R_k, for some $k \in \mathbb{N}_0$. Therefore, we obtain a weaker nonmonotone strategy that can augment the effect of f_k when iterates are close to the optimizer.

On the basis of the above considerations, the algorithmic framework of our approach can be outlined as follows:

In Algorithm 1, if $\widehat{r}_k \geq \mu_3 > 0$, the iterates are called very successful, leading to $\delta_{k+1} \geq \delta_k$. The iterates are called successful if $\widehat{r}_k \geq \mu_2$ and the trust-region

(Algorithm 1: Nonmonotone trust-region algorithm (**NMTRN**))

(S.0) An initial point $x_0 \in \mathbb{R}^n$, a symmetric positive-definite matrix $B_0 \in \mathbb{R}^{n\times n}$, $0 < \eta_0 < 1$, $0 < \mu_1 \le \mu_2 \le \mu_3 < 1$, $0 < \gamma_1 \le \gamma_2 < 1$, $\gamma_3 \ge 1$, $N > 0$ and $\epsilon > 0$ and set $n(0) := 0$; $\widehat{R}_0 := f_0$; $k := 0$.

(S.1) If $\|g_k\| \le \epsilon$ holds, STOP.

(S.2) Specify the trial point d_k by solving the subproblem (1.2).

(S.3) Determine the trust-region ratio $\widehat{r}_k$ using (2.3).
If $\widehat{r}_k \ge \mu_1$ holds, set $x_{k+1} := x_k + d_k$; otherwise, set $x_{k+1} := x_k$.

(S.4) Select $n(k+1)$ in $[0, \min\{n(k)+1, N\}]$, $f_{l(k+1)}$ using (1.5), η_{k+1} by an adaptive formula, generate $\widehat{\eta}_{k+1}$ using (2.2) and $\widehat{R}_{k+1}$ using (2.1).

(S.5) Update the radius of trust-region by using

$$\delta_{k+1} := \begin{cases} \gamma_1\delta_k, & \text{if } \widehat{r}_k < \mu_1 \\ \gamma_2\delta_k, & \text{if } \widehat{r}_k \in [\mu_1, \mu_2) \\ \delta_k, & \text{if } \widehat{r}_k \in [\mu_2, \mu_3) \\ \min\{\gamma_3\delta_k, \delta_0\}, & \text{if } \widehat{r}_k \ge \mu_3. \end{cases} \tag{2.4}$$

(S.6) Update B_{k+1} by a quasi-Newton formula.

(S.7) Set $k := k+1$, and go to (S.1).

radius does not change. Moreover, the iterates for which $\widehat{r}_k \ge \mu_1$ are called successful, so that $\delta_{k+1} \le \delta_k$. Otherwise, the iterates for which $\widehat{r}_k < \mu_1$ are said to be unsuccessful, so that $\delta_{k+1} \le \delta_k$. For cases when iterates are successful or very successful, the new point is generated by $x_{k+1} := x_k + d_k$; otherwise, we set $x_{k+1} := x_k$, cf. [31].

3. Convergence Theory

In this section, we will investigate the global and local quadratic convergence results of the proposed algorithm given in Section 2. We here consider the following assumptions:

(H0) $m_k(x_k) - m_k(x_k + d_k) \ge \beta\|g_k\| \min\left\{\delta_k, \dfrac{\|g_k\|}{\|B_k\|}\right\}, \quad \forall\, k.$

(H1) $f \in \mathcal{C}^2$ and has a lower bound on the level set

$$L(x_0) := \Big\{x \in \mathbb{R}^n \mid f(x) \le f(x_0)\Big\}.$$

(H2) There exists a constant $M > 0$ such that $\|B_k\| \le M$ for all k.

(H3) There exists a constant $\sigma > 0$ such that the trial step d_k satisfies $\|d_k\| \le \sigma\|g_k\|$.

Suppose that the objective function f is a twice continuously differentiable function and the level set $L(x_0)$ is bounded. Then, (H1) implies that $\|\nabla^2 f(x)\|$ is uniformly continuous and bounded above on the open bounded convex set

Ω, containing $L(x_0)$. As a result, there exists a constant $L_g > 0$ such that $\|\nabla^2 f(x)\| \leq L_g$, for all $x \in \Omega$. Therefore, using the mean value theorem, we can conclude that, for all $x, y \in \Omega$,

$$\|g(x) - g(y)\| \leq L_g \|x - y\|,$$

which leads to this fact that the objective function f is Lipschitz continuous in the open bounded convex set Ω.

Lemma 3.1. *Suppose that (H1) and (H2) hold. Then, there exists a constant $\kappa > 0$ such that*

$$|m_k(x_k + d_k) - f(x_k + d_k)| \leq \kappa \|d_k\|^2.$$

Proof. See [10, 13]. □

Consider the following sets

$$\begin{aligned}
\mathcal{I}_1 &:= \{k \in \mathbb{N}_0 \mid \widehat{\eta}_k \in (0,1)\},\\
\mathcal{I}_2 &:= \{k \in \mathbb{N}_0 \mid \widehat{\eta}_k \geq 1 \ \text{ and } \ f_{l(k)} \geq f_{k+1}\},\\
\mathcal{I}_3 &:= \{k \in \mathbb{N}_0 \mid \widehat{\eta}_k \geq 1 \ \text{ and } \ f_{l(k)} < f_{k+1}\},
\end{aligned}$$

that have a key role for proving the next lemmas.

Lemma 3.2. *Suppose that (H1) holds and the sequence $\{x_k\}_{k\geq 0}$ is generated by Algorithm 1. Then, the following statements are true:*
(a) If $f_k \neq 0$ and $f_{l(k)} > 0$, then $R_k \leq \widehat{R}_k < f_{l(k)}$, for $k \in \mathcal{I}_1$.
(b) If $f_k \neq 0$ and $f_{l(k)} > 0$, then $\widehat{R}_k \geq f_{l(k)}$, for $k \in \mathcal{I}_2 \cup \mathcal{I}_3$.
(c) If $f_k < 0$ and $f_{l(k)} < 0$, $f_k \leq \widehat{R}_k \leq R_k$, for $k \in \mathcal{I}_1$
(d) If $f_k = 0$, then $\widehat{R}_k = R_k$.
(e) If $f_{l(k)} = 0$, then $\widehat{R}_k = 0$.

Proof. (a) Suppose that $f_k \neq 0$ and $f_{l(k)} > 0$. Then, we have $\left|\dfrac{f_{l(k)}}{f_k}\right| \geq 1$ and consequently $\widehat{\eta}_k \geq \eta_k$. This inequality, together with $f_k \leq f_{l(k)}$, for $k \in \mathcal{I}_1$, gives

$$R_k = \eta_k(f_{l(k)} - f_k) + f_k \leq \widehat{\eta}_k(f_{l(k)} - f_k) + f_k = \widehat{R}_k = (1 - \widehat{\eta}_k)(f_k - f_{l(k)}) + f_{l(k)} < f_{l(k)}.$$

(b) Assume that $f_k \neq 0$ and $f_{l(k)} > 0$. From (2.1) and $f_k \leq f_{l(k)}$, for $k \in \mathcal{I}_2 \cup \mathcal{I}_3$, we obtain

$$\begin{aligned}
\widehat{R}_k &= \widehat{\eta}_k f_{l(k)} + (1 - \widehat{\eta}_k) f_k = (\widehat{\eta}_k - 1) f_{l(k)} + (1 - \widehat{\eta}_k) f_k + f_{l(k)}\\
&= (\widehat{\eta}_k - 1)(f_{l(k)} - f_k) + f_{l(k)} \geq f_{l(k)}.
\end{aligned}$$

(c) Assume that $f_k < 0$ and $f_{l(k)} < 0$. Then, we have $\left|\dfrac{f_{l(k)}}{f_k}\right| \leq 1$ and consequently $\widehat{\eta}_k \leq \eta_k$. Therefore, this inequality, along with $f_k \leq f_{l(k)}$, for $k \in \mathcal{I}_1$, gives

$$f_k \leq \widehat{\eta}_k(f_{l(k)} - f_k) + f_k = \widehat{R}_k \leq \eta_k(f_{l(k)} - f_k) + f_k = R_k.$$

For (d) and (e), the proof is clear. □

Lemma 3.3. *Suppose that (H0)-(H3) hold and the sequence* $\{x_k\}_{k\geq 0}$ *is generated by Algorithm 1.*

(a) $\{f_{l(k)}\}_{k\in\mathcal{I}_1}$ *is a convergent subsequence of* $\{f_k\}_{k\geq 0}$ *and* $\{x_k\}_{k\in\mathcal{I}_1} \subset L(x_0)$.

(b) *If* $\mathcal{I}_1$ *is not finite, then* $\lim\limits_{k\to\infty, k\in\mathcal{I}_1} f_{l(k)} = \lim\limits_{k\to\infty, k\in\mathcal{I}_1} f_k$.

(c) *If* $\mathcal{I}_1$ *is not finite, then* $\lim\limits_{k\to\infty, k\in\mathcal{I}_1} \widehat{R}_k = \lim\limits_{k\to\infty, k\in\mathcal{I}_1} f_k$.

Note that

Proof. (a) Assume that x_{k+1} is accepted by Algorithm 1. This fact, along with Lemma 3.2, implies

$$f_{l(k)} - f_{k+1} > \widehat{R}_k - f_{k+1} \geq \mu_1(m_k(x_k) - m_k(x_k + d_k)) > 0,$$

leading to

$$f_{k+1} \leq f_{l(k)}, \quad \forall\, k \in \mathcal{I}_1. \tag{3.1}$$

To prove that the subsequence $\{f_{l(k)}\}_{k\in\mathcal{I}_1}$ is decreasing, we consider the two following cases:

Case 1. If $k \geq N$, then we have $n(k+1) \leq n(k)+1$ for all $k \in \mathcal{I}_1$. Therefore, (3.1) results in

$$f_{l(k+1)} = \max_{0\leq j\leq n(k+1)}\{f_{k+1-j}\} \leq \max\{\max_{0\leq j\leq n(k)}\{f_{k-j}\}, f_{k+1}\} = \max\{f_{l(k)}, f_{k+1}\} = f_{l(k)}, \tag{3.2}$$

for all $k \in \mathcal{I}_1$.

Case 2. If $k < N$, then $n(k) = k$, for all $k \in \mathcal{I}_1$. By using $f_k \leq f_0$, we get

$$f_{l(k)} = f_0, \quad \forall k \in \mathcal{I}_1. \tag{3.3}$$

We show that $x_k \in L(x_0)$, for all $k \in \mathcal{I}_1$. The proof is given by the induction. The definition of $f_{l(k)}$ indicates that $f_{l(0)} = f_0$. Next, assume that $x_k \in L(x_0)$, for some $k \in \mathcal{I}_1$ (the induction hypothesis), holds. By using the induction hypothesis and the decreasing sequence $f_{l(k)}$, we get

$$f_{k+1} \leq f_{l(k+1)} \leq f_{l(k)} \leq f_0,$$

which shows that $x_{k+1} \in L(x_0)$. Thus, the subsequence $\{x_k\}_{k\in\mathcal{I}_1}$ is contained in $L(x_0)$.

Finally, (H1) and $x_k \in L(x_0)$, for all $k \in \mathcal{I}_1$, imply that the subsequence $\{f_{l(k)}\}_{k\in\mathcal{I}_1}$ is bounded. Thus, the subsequence $\{f_{l(k)}\}_{k\in\mathcal{I}_1}$ is convergent.

(b) The proof can be done in the same way as Lemma 7 in [1].

(c) By $f_k \leq \widehat{R}_k \leq f_{l(k)}$ and $\lim\limits_{k\to\infty, k\in\mathcal{I}_1} f_{l(k)} = \lim\limits_{k\to\infty, k\in\mathcal{I}_1} f_k$, the result is valid. □

Lemma 3.4. *Suppose that (H0)-(H3) hold and the sequence* $\{x_k\}_{k\geq 0}$ *is generated by Algorithm 1.*

(a) $\{f_{l(k)}\}_{k\in\mathcal{I}_2}$ *is a convergent subsequence of* $\{f_k\}_{k\geq 0}$ *and* $\{x_k\}_{k\in\mathcal{I}_2} \subset L(x_0)$.

(b) *If* $\mathcal{I}_2$ *is not finite, then* $\lim\limits_{k\to\infty, k\in\mathcal{I}_2} f_{l(k)} = \lim\limits_{k\to\infty, k\in\mathcal{I}_2} f_k$.

(c) *If* $\mathcal{I}_2$ *is not finite, then* $\lim\limits_{k\to\infty, k\in\mathcal{I}_2} \widehat{R}_k = \lim\limits_{k\to\infty, k\in\mathcal{I}_2} f_k$.

Proof. (a) Assume that x_{k+1} is accepted by Algorithm 1. Hence, we have

$$\widehat{R}_k - f_{k+1} \geq \mu_1(m_k(x_k) - m_k(x_k + d_k)) > 0,$$

implying

$$f_{k+1} \leq \widehat{R}_k, \quad \forall\, k \in \mathcal{I}_2.$$

Now, similar to Lemma 3.3, we can simply obtain that the subsequence $\{f_{l(k)}\}_{k\in\mathcal{I}_2}$ is convergent and $\{x_k\}_{k\in\mathcal{I}_2} \subset L(x_0)$ since $f_{l(k)} \geq f_{k+1}$, for $k \in \mathcal{I}_2$.

(b) Since x_{k+1} is accepted by Algorithm 1, we obtain

$$\begin{aligned} \widehat{R}_k - f_{k+1} &= \widehat{\eta}_k f_{l(k)} + (1 - \widehat{\eta}_k) f_k - f_{k+1} \\ &= (\widehat{\eta}_k - 1)(f_{l(k)} - f_k) + f_{l(k)} - f_{k+1} \\ &\geq \mu_1(m_k(x_k) - m_k(x_k + d_k)) > 0. \end{aligned} \tag{3.4}$$

By the definition of $\{f_{l(k)}\}_{k\geq 0}$, it is clear that $l(k) \leq k$. Hence $l(k) - 1$ can be considered as a successful iterate preceding kth successful iterate. By substituting k by $l(k) - 1$ in (3.4), we obtain

$$(\widehat{\eta}_{l(k)-1} - 1)(f_{l(l(k)-1)} - f_{l(k)-1}) + f_{l(l(k)-1)} - f_{l(k)} \geq \mu_1(m_k(x_{l(k)-1}) - m_k(x_{l(k)})) > 0.$$

By recalling item (a) and taking limits from both sides of the above inequality, we get

$$\lim_{k\to\infty, k\in\mathcal{I}_2} (m_k(x_{l(k)-1}) - m_k(x_{l(k)})) = 0.$$

The reminding of the proof follows from Lemma 7 in [1].

(c) By combining item (b) and the definition of $\widehat{\eta}_k$, we get

$$\lim_{k\to\infty, k\in\mathcal{I}_2} \widehat{\eta}_k = \lim_{k\to\infty, k\in\mathcal{I}_2} \eta_k \lim_{k\to\infty, k\in\mathcal{I}_2} \left|\frac{f_{l(k)}}{f_k}\right| = \lim_{k\to\infty, k\in\mathcal{I}_2} \eta_k = \eta_*.$$

This expression, together with item (b), leads to

$$\begin{aligned} \lim_{k\to\infty, k\in\mathcal{I}_2} \widehat{R}_k &= \eta_* \lim_{k\to\infty, k\in\mathcal{I}_2} f_{l(k)} + (1 - \eta_*) \lim_{k\to\infty, k\in\mathcal{I}_2} f_k \\ &= \eta_* \lim_{k\to\infty, k\in\mathcal{I}_2} f_k + (1 - \eta_*) \lim_{k\to\infty, k\in\mathcal{I}_2} f_k \\ &= \lim_{k\to\infty, k\in\mathcal{I}_2} f_k, \end{aligned}$$

giving the results. □

Lemma 3.5. *Suppose that (H0)-(H3) hold and the sequence* $\{x_k\}_{k\geq 0}$ *is generated by Algorithm 1.*

(a) $\{\widehat{R}_k\}_{k\in\mathcal{I}_3}$ *is a convergent subsequence of* $\{\widehat{R}_k\}_{k\geq 0}$ *and* $\{x_k\}_{k\in\mathcal{I}_3} \subset L(x_0)$.

(b) *If* $\mathcal{I}_3$ *is not finite, then* $\lim_{k\to\infty, k\in\mathcal{I}_3} f_{l(k)} = \lim_{k\to\infty, k\in\mathcal{I}_3} f_k$.

(c) *If* $\mathcal{I}_3$ *is not finite, then* $\lim_{k\to\infty, k\in\mathcal{I}_3} \widehat{R}_k = \lim_{k\to\infty, k\in\mathcal{I}_3} f_k$.

Proof. (a) Since x_{k+1} is accepted by Algorithm 1, we can write

$$\widehat{R}_k - f_{k+1} \geq \mu_1(m_k(x_k) - m_k(x_k + d_k)) > 0,$$

leading to

$$f_{k+1} \leq \widehat{R}_k, \quad \forall\, k \in \mathcal{I}_3. \tag{3.5}$$

Since $f_{l(k)} < f_{k+1}$ for $k \in \mathcal{I}_3$, the definition of $f_{l(k)}$ implies that $f_{l(k+1)} \leq f_{k+1}$, for $k \in \mathcal{I}_3$. By combining this expression with (3.5), we get

$$\widehat{R}_{k+1} \leq f_{k+1} \leq \widehat{R}_k, \tag{3.6}$$

which shows that $\{\widehat{R}_k\}_{k\in\mathcal{I}_3}$ is decreasing.

We now use the induction to show that $x_k \in L(x_0)$, for all $k \in \mathcal{I}_3$. If $k = 0$, the definition of $\widehat{R}_k$ indicates that $\widehat{R}_0 = f_0$. Let us the induction hypothesis is satisfied, i.e., $x_k \in L(x_0)$, for some $k \in \mathcal{I}_3$. From (3.6), the definition of $\widehat{R}_k$, and the induction hypothesis, we obtain

$$f_{k+1} = \widehat{R}_{k+1} \leq \widehat{R}_k \leq f_0.$$

Thus, $x_{k+1} \in L(x_0)$ giving the result.

(H1) and $x_k \in L(x_0)$, for all $k \in \mathcal{I}_3$, imply that the subsequence $\{\widehat{R}_k\}_{k\in\mathcal{I}_3}$ is bounded. Thus, the subsequence $\{\widehat{R}_k\}_{k\in\mathcal{I}_3}$ is convergent, too.

(b) From the definition of $\mathcal{I}_3$, we obtain

$$\lim_{k\to\infty, k\in\mathcal{I}_3} f_{l(k)} < \lim_{k\to\infty, k\in\mathcal{I}_3} f_{k+1} = \lim_{k\to\infty, k\in\mathcal{I}_3} f_k. \tag{3.7}$$

It follows from $f_k \leq f_{l(k)}$ that

$$\lim_{k\to\infty, k\in\mathcal{I}_3} f_k \leq \lim_{k\to\infty, k\in\mathcal{I}_3} f_{l(k)}. \tag{3.8}$$

From (3.7) and (3.8), we obtain

$$\lim_{k\to\infty, k\in\mathcal{I}_3} f_{l(k)} = \lim_{k\to\infty, k\in\mathcal{I}_3} f_k.$$

(c) The result is proven in the same way as item (c) of Lemma 3.4. □

The next result is the direct consequence of Lemmas 3.3-3.5.

Corollary 3.6. *Suppose that (H0)-(H3) hold and the sequence* $\{x_k\}_{k\geq 0}$ *is generated by Algorithm 1.*

(a) $\{f_{l(k)}\}_{k\in\mathcal{I}_1\cup\mathcal{I}_2}$ *is a convergent subsequence of* $\{f_k\}_{k\geq 0}$ *and* $\{x_k\}_{k\in\mathcal{I}_1\cup\mathcal{I}_2} \subset L(x_0)$.

(b) $\{\widehat{R}_k\}_{k\in\mathcal{I}_3}$ *is a convergent subsequence of* $\{R_k\}_{k\geq 0}$ *and* $\{x_k\}_{k\in\mathcal{I}_3} \subset L(x_0)$.

(c) *If $\mathcal{I}_i$ is not finite, for $i = 1, 2, 3$, then* $\lim_{k\to\infty} f_{l(k)} = \lim_{k\to\infty} f_k$.

(d) *If $\mathcal{I}_i$ is not finite, for $i = 1, 2, 3$,* $\lim_{k\to\infty} \widehat{R}_k = \lim_{k\to\infty} f_k$.

The following result is essential for giving the global convergence.

Lemma 3.7. *Suppose that (H0)-(H3) hold. If there exists $\epsilon > 0$ such that*

$$\|g_k\| \geq \epsilon, \quad \forall\ k, \tag{3.9}$$

then there exists a constant $\tau > 0$ such that

$$\delta_k \geq \tau, \quad \forall\ k. \tag{3.10}$$

Proof. It can be shown by induction over k that (3.10) holds with

$$\tau := \min\Big\{\delta_0, \frac{\gamma_1\epsilon}{M}, \frac{\gamma_1\beta(1-\mu_1)\epsilon}{\kappa}\Big\}.$$

Indeed, (3.10) is clearly true for $k = 0$. Assuming that (3.10) is true for iterate k, we establish the inequality for iterate $k+1$. By (H0), (H2) and Lemma 3.1, we have

$$1 - r_k \leq \frac{\kappa\delta_k^2}{\beta\epsilon\min\Big\{\delta_k, \dfrac{\epsilon}{M}\Big\}}.$$

If

$$\delta_k \leq \min\Big\{\frac{\epsilon}{M}, \frac{\beta(1-\mu_2)\epsilon}{\kappa}\Big\}, \tag{3.11}$$

it follows that $1 - r_k \leq 1 - \mu_2$, that is, $r_k \geq \mu_2$. As $\widehat{r}_k \geq r_k$, we obtain $\widehat{r}_k \geq \mu_2$. Thus, the update rule for δ_k and the induction assumption provide the bound $\delta_{k+1} \geq \delta_k \geq \tau$, and so (3.10) holds for $k+1$.

Now, suppose that (3.11) is not true. Then, the update rule of δ_k and the definition of τ imply that

$$\delta_{k+1} \geq \gamma_1\delta_k \geq \min\Big\{\frac{\gamma_1\epsilon}{M}, \frac{\gamma_1\beta(1-\mu_2)\epsilon}{\kappa}\Big\} \geq \tau.$$

This shows that (3.10) holds for $k+1$ and completes the induction argument. □

Theorem 3.8. *Suppose that (H0)-(H3) hold. Then*

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

Proof. Suppose by contradiction that there exists $\epsilon > 0$ such that $\|g_k\| \geq \epsilon$ for all k. In this case, Lemma 3.7 provides the bound

$$\delta_k \geq \tau, \quad \forall\ k, \tag{3.12}$$

where τ is defined in (3.10). Let us consider the set $\mathcal{K} := \{k \in \mathbb{N} \mid \widehat{r}_k \geq \mu_1\}$. For $k \in \mathcal{K}$, (H0) and (3.12) imply that

$$\widehat{R}_k - f_{k+1} \geq \mu_1\beta\epsilon\min\Big\{\frac{\epsilon}{M}, \tau\Big\}. \tag{3.13}$$

On the other hand, as $k \to \infty$, by the definition of $\mathcal{I}_i$'s, there exists at least $i \in \{1,2,3\}$ so that $\mathcal{I}_i$ is not finite. Then, by Lemma 3.3-3.5, we get $\widehat{R}_k - f_{k+1} \to 0$ as $k \to \infty$ and $k \in \mathcal{K} \subseteq \mathcal{I}_i$. Thus, from (3.13) we see that $\mathcal{K}$ is finite. Therefore, $\widehat{r}_k < \mu_1$ for all k sufficiently large. Consequently, by the rule of updating δ_k, we get $\delta_k \to 0$, which contradicts (3.12). □

Theorem 3.9. *Suppose that (H0)-(H3) hold. Then*

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

Proof. Thanks to Theorem 3.8 and to Corollary 3.6, it follows as in the proof of Theorem 11 in [1]. □

In the sequel, we will show the local superlinear and quadratic convergence rates of Algorithm 1 under some classical assumptions that have been widely used in the nonlinear optimization literatures.

Theorem 3.10. *Suppose that (H0)-(H3) hold, the sequence $\{x_k\}_{k\geq 0}$ is generated by Algorithm 1 converging to x^*, the matrix $G(x) := \nabla^2 f(x)$ is continuous in a neighborhood $N(x^*, \epsilon)$ of x^*, and B_k satisfies*

$$\lim_{k\to\infty} \frac{\|[B_k - G(x^*)]d_k\|}{\|d_k\|} = 0.$$

If x_0 is close enough to x_*, then, the sequence $\{x_k\}_{k\geq 0}$ converges to x^* superlinearly. Moreover, if $B_k := G(x_k)$ and $G(x)$ is Lipschitz continuous in a neighborhood $N(x^*, \epsilon)$, then the sequence $\{x_k\}_{k\geq 0}$ converges to x^* quadratically.

Proof. The proof is similar to the proof of Theorems 4.1 and 4.2 in [2] and therefore the details are omitted. □

4. Preliminary numerical experiments

We now firstly report the results obtained by running Algorithm 1 (NMTRN) in comparison with the nonmonotone trust-region algorithm of Ahookhosh et al. in [1] (NMTRA) and the nonmonotone trust-region algorithm from Zhang et al. in [35] (NMTRZ) on 219 standard unconstrained test problems; see Appendix A. All tests were written in double precision format in MATLAB 2011a on a laptop Asus with a 1.7 GHz Intel Core i3-4010U CPU and 4 GB of memory under ubuntu 10.04 Linux.

For all of these codes, the trust-region subproblems are solved by Steihaug & Toint procedure, see [10, 30]. Such an algorithm ends up at $x_k + d$ if

$$\|\nabla m(x_k+d)\| \leq \min\left\{0.01, \|\nabla m_k(x_k)\|^{\frac{1}{2}}\right\} \|\nabla m_k(x_k)\| \text{ or } \|d\| = \delta_k,$$

holds. In our numerical experiments, the algorithms are stopped whenever

$$\|g_k\| \leq 10^{-6}\sqrt{n},$$

or the total number of iterates exceeds 20000. During our implementation, we verified whether the different codes converge to the same point. Therefore, we only provided data for problems in which all algorithms converged to the same point. In all algorithms, the matrix B_k is updated by the compact limited memory BFGS formula

$$B_k := B_k^{(0)} - \begin{bmatrix} Y_k & B_k^{(0)} S_k \end{bmatrix} \begin{bmatrix} -D_k & L_k^T \\ L_k & S_k^T B_k^{(0)} S_k \end{bmatrix}^{-1} \begin{bmatrix} Y_k^T \\ S_k^T B_k^{(0)} \end{bmatrix},$$

where the basic matrix $B_k^{(0)}$ is defined as $B_k^{(0)} := \lambda I$, for some positive scalar λ. S_k, Y_k, D_k and L_k are defined as follows:

$$S_k := [s_{k-m}, \ldots, s_{k-1}], \quad Y_k = [y_{k-m}, \ldots, y_{k-1}],$$
$$D_k := \operatorname{diag}\left[s_{k-m}^T y_{k-m}, \ldots, s_{k-1}^T y_{k-1}\right],$$
$$(L_k)_{i,j} := \begin{cases} s_{k-m+i-1}^T y_{k-m+j-1}, & \text{if } i > j, \\ 0, & \text{otherwise,} \end{cases}$$

where

$$s_k := x_{k+1} - x_k, \qquad y_k := g_{k+1} - g_k,$$

and $m := \min\{k, m_1\}$ in where $m_1 = 5$. In our implementation, we use

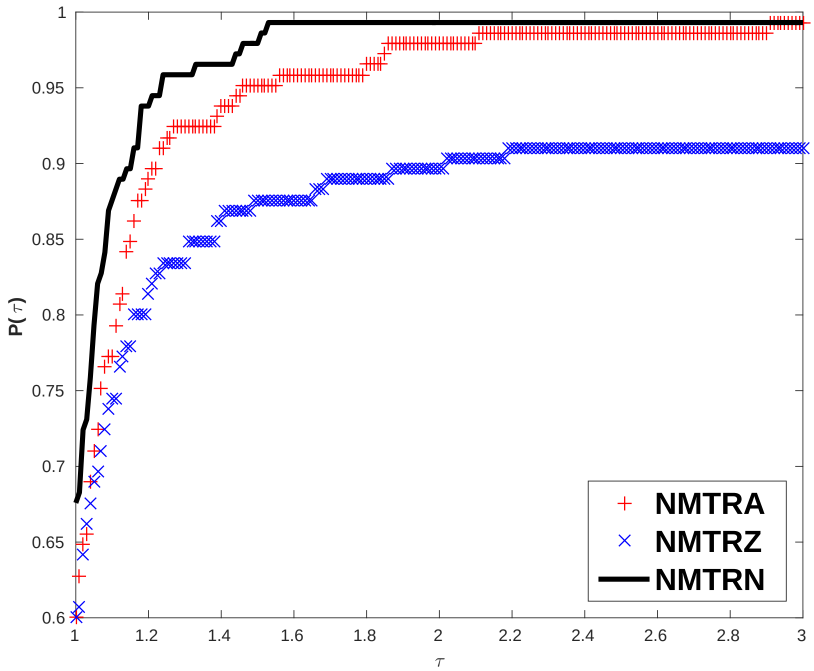
$$\lambda := \frac{\|y_{k_m}\|^2}{{y_{k_m}}^T s_{k_m}},$$

suggested by Shanno & Plau in [32]. However, we do not update B_k whenever the curvature condition, i.e., $s_{k_i}^T y_{k_i} > 0$ for $i = 1, \ldots, m$, does not hold, cf. [9]. The code of the compact limited memory BFGS updating formula is rewritten based on ASTRAL code in [34]. For all algorithms, the trust-region radius is updated by (2.4) and we set $\mu_1 = 10^{-5}$, $\mu_2 = 0.2$, $\mu_3 = 0.8$, $\gamma_1 = 0.25$, $\gamma_2 = 0.5$, $\gamma_3 = 2$, $\delta_0 = 10$ and $N = 10$, see [34]. Furthermore, for all algorithms, the parameter η_k is updated by

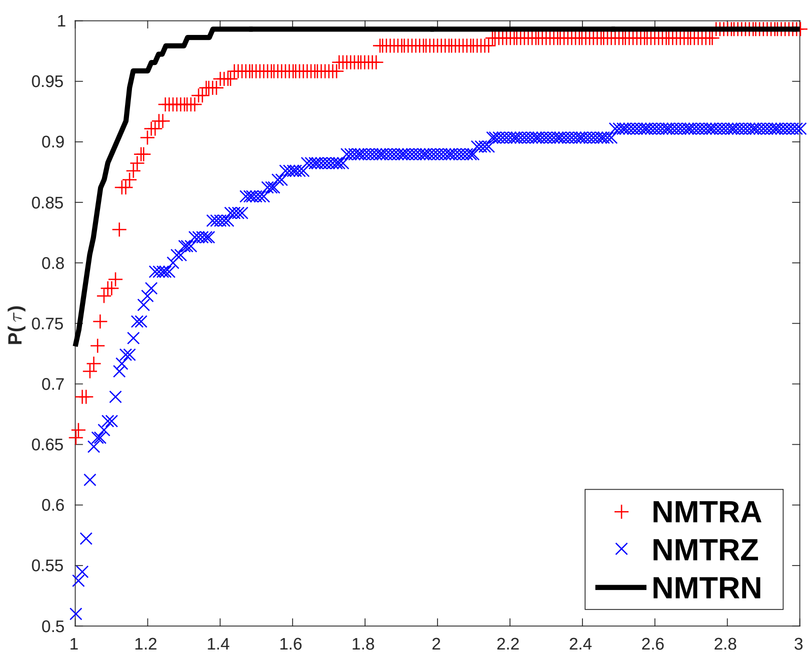
$$\eta_k := \begin{cases} \frac{2}{3}\eta_{k-1} + 0.01 & \text{if } \|g_k\| \le \xi, \\ \max\{0.99\eta_{k-1}, 0.5\} & \text{otherwise,} \end{cases}$$

where $\eta_0 = 0.2$ and $\xi = 10^{-2}$, see [4].

Tables 1-3 indicate the names and dimensions of the test problems considered. To demonstrate the overall behavior of the presented algorithms and get more insight about the performance of the considered codes, the performance of all codes, based on N_i and N_f, has been assessed by applying the performance profile proposed from Dolan & Moré by [12]. In this procedure, the profile of each code is measured based on the ratio of its computational outcome versus the best numerical outcome of all codes. This profile offers a tool for comparing the performance of iterative processes in statistical structure. In the figures, P designates the percentage of problems, which are solved within a factor τ



(a)



(b)

FIGURE 1. A comparison among NMTRA, NMTRZ and NMTRN by performance profiles using the measures N_i and N_f: (a) displays the number of iterations (left); (b) shows the number of function evaluations (right), respectively

of the best solver. The results are illustrated in Figures 1 with respect to the total number of iterates and the total number of function evaluations.

Subfigure (a) of Figure 1 shows that NMTRN outperforms NMTRA and NMTRZ regarding the total number of iterates. In particular, NMTRN has most wins in nearly 68% score of the tests with the greatest efficiency. Meanwhile, in the sense of the ability of completing a run successfully, it is the best among considered algorithms because it grows up faster than others and reaches 1 more rapidly. As illustrated in Subfigure (b) of Figure 1, NMTRN implements remarkably better than others where it has most wins in approximately 74% score of performed tests concerning the total number of function evaluations. Furthermore, Figure 1 shows similar patterns in the sense of the ability of completing a run successfully. As a result, this fact directly implies that the total number of solving the trust-region subproblems is notably decreased for NMTRN.

5. Concluding Remarks

This paper is concerned with introducing and analyzing a trust-region-based algorithm for unconstrained optimization using a new effective nonmonotone strategy. To overcome some disadvantages of the nonmonotone strategy (1.6), our new nonmonotone strategy has been constructed based on a combination of the current function value with the maximum function values in some prior successful iterate. We showed that a suitable adaptive process can increase effectiveness of the new nonmonotone strategy compared with some stat-of-the-art nonmonotone strategies [1, 35]. The global convergence and local convergence rates of the proposed algorithm are established. Preliminary numerical results on a large set of unconstrained optimization problems indicate the promising behavior of the proposed method.

Acknowledgement

The first author acknowledges financial support of the Doctoral Program "Vienna Graduate School on Computational Optimization" funded by Austrian Science Foundation under Project No W1260-N35.

References

1. M. Ahookhosh, K. Amini, An efficient nonmonotone trust-region method for unconstrained optimization,*Numerical Algorithms* **59**(4), (2012), 523–540.
2. M. Ahookhosh, K. Amini, A nonmonotone trust region method with adaptive radius for unconstrained optimization problems, *Computers and Mathematics with Applications*, **60**, (2010), 411–422.
3. M. Ahookhosh, K. Amini, M. Kimiaei, A globally convergent trust-region method for large-scale symmetric nonlinear systems, *Numerical Functional Analysis and Optimization*, **36**, (2015), 830–855.

4. M. Ahookhosh, K. Amini, H., Nosratipour, An inexact line search approach using modified nonmonotone strategy for unconstrained optimization, *Numerical Algorithms*, **66**, (2014), 49–78.
5. M. Ahookhosh, K. Amini, M.R. Peyghami, A nonmonotone trust-region line search method for large-scale unconstrained optimization, *Applied Mathematical Modelling*, **36**, (2012), 478–487.
6. M. Ahookhosh, H. Esmaeili, M. Kimiaei, An effective trust-region-based approach for symmetric nonlinear systems, *International Journal of Computer Mathematics*, **90**, (2013), 671–690.
7. M. Ahookhosh, S. Ghaderi, Two globally convergent nonmonotone trust-region methods for unconstrained optimization, *Journal of Applied Mathematics and Computing*, **50**(1-2), (2016), 529–555.
8. N. Andrei, An unconstrained optimization test functions collection, *Advanced Modeling and Optimization*, **10**(1), (2008), 147–161.
9. R. Byrd, J. Nocedal, R. Schnabel, Representation of quasi-Newton matrices and their use in limited memory methods, *Mathematical Programming*, **63**, (1994), 129–156.
10. A.R. Conn, N.I.M. Gould, Ph.L. Toint, *Trust-Region Methods*, Society for Industrial and Applied Mathematics SIAM, Philadelphia, 2000.
11. N.Y. Deng, Y. Xiao, F.J. Zhou, Nonmonotonic trust region algorithm, *Journal of Optimization Theory and Applications*, **26**, (1993), 259–285.
12. E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming*, **91**, (2002), 201–213.
13. H. Esmaeili, M. Kimiaei, An improved adaptive trust-region method for unconstrained optimization, *Mathematical Modelling and Analysis*, **19**, (2014), 469–490.
14. Fasano, G., Lampariello, F., Sciandrone, M.: A truncated nonmonotone Gauss-Newton method for large-scale nonlinear least-squares problems, *Computational Optimization and Applications*, **34**(3), 343–358 (2006)
15. A. Fischer, P.K. Shukla, M. Wang, On the inexactness level of robust Levenberg-Marquardt methods, *Optimization*, **59**(2), (2010), 273–287.
16. N.I.M Gould, D. Orban, Ph.L. Toint, CUTEst: a Constrained and Unconstrained Testing Environment with safe threads for mathematical optimization, *Computational Optimization and Applications*, **60**(3), (2015), 545–557.
17. L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton's method, *SIAM Journal on Numerical Analysis*, **23**, (1986), 707–716.
18. L. Grippo, F. Lampariello, S. Lucidi, A truncated Newton method with nonmonotone linesearch for unconstrained optimization, *Journal of Optimization Theory and Applications*, **60**(3), (1989), 401–419.
19. L. Grippo, F. Lampariello, S. Lucidi, A class of nonmonotone stabilization method in unconstrained optimization, *Numerische Mathematik*, **59**, (1991), 779–805.
20. L. Kaufman, Reduced storage quasi-Newton trust region approaches to function optimization, *SIAM Journal on Optimization*, **10**(1), 56–69 (1999)
21. M. Kimiaei, A new class of nonmonotone adaptive trust-region methods for nonlinear equations with box constraints, *Calcolo*, **54**(3), 769–812 (2017)
22. Kimiaei, M., Ghaderi, S.: A new restarting adaptive Trust-Region method for unconstrained optimization, *Journal of the Operations Research Society of China*, **5**(4), (2017), 487–507.
23. Kimiaei, M., Rahpeymaii, F.: A new nonmonotone line-search trust-region approach for nonlinear systems, *TOP*, **27**(2), (2019), 199–232.
24. L. Lukšan, C. Matonoha, J. Vlček, Modified CUTE problems for sparse unconstrained optimization. *Techical Report*, **1081**, ICS AS CR, November 2010.

25. L. Lukšan, J. Vlček, Sparse test problems for unconstrained optimization, *Techical Report*, **1064**, ICS AS CR, November 2003.
26. YU. Nesterov, Modified Gauss-Newton scheme with worst case guarantees for global performance, *Optimization Methods and Software*, **22**(3), (2007) 469–483.
27. J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer, NewYork, (2006).
28. M.J.D. Powell, Convergence properties of a class of minimization algorithms. in Nonlinear Programming, O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, eds., Academic Press, NewYork, 1–27 (1975)
29. D.F. Shanno, K.H. Phua, Matrix conditioning and non-linear optimization, *Mathematical Programming*, **14**, (1978), 149–160.
30. T. Steihaug, The conjugate gradient method and trust regions in large scale optimization, *SIAM Journal on Numerical Analysis*, **20**, (1983), 626–637
31. W. Sun, Y. Yuan, *Optimization Theory and Methods*: Nonlinear Programming. Springer, Berlin (2006).
32. S.W. Thomas, *Sequential estimation techniques for quasi-Newton algorithms*, Cornell University, 1975.
33. Ph.L. Toint, Numerical solution of large sets of algebraic nonlinear equations, *Mathematics of Computation*, **46**(173), (1986), 175–189.
34. L. Xu, J.V. Burke, An active set $\ell_\infty-$trust region algorithm for box constrained optimization. Technical Report preprint, Departeman Mathematics, University of Washington, Seattle, WA 98195, U.S.A,
35. H.C. Zhang, W.W. Hager, A nonmonotone line search technique for unconstrained optimization, *SIAM journal on Optimization*, **14**(4), (2004), 1043–1056.

Appendix A. Appendix: The list of test problems

In Table 1, the problems are discussed from the CUTEst unconstrained test problems proposed by Gould et al. in [16] while the test problems of Table 2 are taken from Lukšan & Vlček in [25]. In addition, the problems of Table 3 are selected from Andrei in [8].

Table 1. List of `CUTEst` test problems

Problem name	Dim	Problem name	Dim	Problem name	Dim
AIRCRFTB	8	DIXMAANL	9000	MEXHAT	2
ALLINITU	4	DIXMAANM	9000	MINSURF	64
ARGLINA	200	DIXMAANN	9000	MOREBV	5000
ARWHEAD	5000	DIXMAANO	9000	MSQRTALS	1024
BARD	3	DIXMAANP	9000	MSQRTBLS	1024
BDQRTIC	100	DJTL	2	NLMSURF	5625
BEALE	2	DQDRTIC	5000	NCB20	5010
BIGGS3	6	DQRTIC	1000	NCB20B	5000
BIGGS5	6	EDENSCH	2000	NONCVXU2	1000
BIGGS6	6	EG2	1000	NONDIA	5000

Table 1. List of `CUTEst` test problems (*continued*)

BOX2	3	EIGENALS	110	NONDQUAR	5000
BOX3	3	EIGENBLS	110	OSBORNEA	5
BROWNAL	200	EIGENCLS	110	OSBORNEB	11
BRKMCC	2	ENGVAL1	5000	PENALTY1	1000
BRYBND	5000	ENGVAL2	3	PENALTY2	10
CHAINWOO	1000	ERRINROS	50	PENALTY3	50
CHNROSNB	50	EXPFIT	2	POWELLSG	500
CLIFF	2	FMINSRF2	5625	POWER	10000
COSINE	10000	FMINSURF	5625	RAYBENDL	40
CRAGGLVY	5000	FREUROTH	5000	ROSENBR	2
CUBE	2	GENROSE	500	S308	2
CURLY10	10	GROWTHLS	3	SCHMVETT	5000
DECONVU	63	GULF	3	SENSORS	100
DENSCHNA	2	HAIRY	2	SINEVAL	2
DENSCHNB	2	HATFLDD	3	SINQUAD	1000
DENSCHNC	2	HATFLDE	3	SISSER	2
DENSCHND	3	HEART6LS	6	SNAIL	2
DENSCHNE	3	HEART8LS	8	SPARSQUR	10000
DENSCHNF	2	HELIX	3	SPMSRTLS	4900
DIXMAANA	9000	HILBERTA	2	SROSENBR	5000
DIXMAANB	9000	HILBERTB	10	TESTQUAD	5000
DIXMAANC	9000	HIMMELBB	2	TOINTGOR	50
DIXMAAND	9000	HIMMELBF	4	TOINTGSS	5000
DIXMAANE	9000	HIMMELBG	2	TRIDIA	5000
DIXMAANF	9000	HIMMELBH	2	VARDIM	200
DIXMAANG	9000	KOWOSB	4	VAREIGVL	50
DIXMAANH	9000	LIARWHD	5000	WATSON	12
DIXMAANI	9000	LMINSURF	5625	WOODS	4000
DIXMAANJ	9000	MANCINO	100	YFITU	3
DIXMAANK	9000	MARATOSB	2	ZANGWIL2	2

Table 2. List of Lukšan and Vlček's test problems

Problem name	Dim	Problem name	Dim
Allgower and Georg b.v	15	Generalization of the Brown 2	10000
Another trigonometric	5000	Modified discrete b.v	100
Ascher and Russel b.v	30	Potra and Rheinboldt b.v	20
Attracting-Repelling	400	Problem 201	200
Banded trigonometric	3000	Problem 202	20000
Brent	9	Problem 206	500
Broyden tridiagonal (problem 36)	40000	Problem 207	20000
Broyden tridiagonal (problem 62)	1000	Problem 208	25000
Chained and modified prolem HS47	8	Problem 212	30000
Chained and modified prolem HS48	8	Problem 213	5000
Chained cragg and levy	10000	Problem 214	30000
Chained exponential	1000	Seven-diagonal system	2000
Chained Freudenstein and Roth	10000	Seven-diag. gen. of the broyden trid.	5000
Chained powell singular	12000	Singular Broyden	5000
Chained Rosenbrock	1000	Sparse modifi. of the Nazareth trig.	8
Chained serpentine	1000	Sparse signomial	1200
Chained wood	1000	Sparse trigonometric	4
Countercurrent reactors 1	8	Structured Jacobian	2000
Countercurrent reactors 2	800	Toint quadratic merging	10000
Discrete boundary value	5000	Toint trigonometric	100
Extended Freudenstein and Roth	5000	Tridiagonal exponential	15
Extended Gragg and Levy	30000	Tridiagonal system	1000
Extended Powell badly scaled	4	Trigexp 1	25000
Extended Powell Singular	28000	Trigexp 2	100
Extended Rosenbrock	40000	Troesch	50
Extended Wood	30000	Variational 1	1000
Five-diagonal system	2000	Variational 2	1000
Flow in a channel	20	Variational 3	1000
Generalized Broyden Banded	30000	Variational 4	1000
Generalized Broyden tridiagonal	30000	Variational Calvar 2	500
Generalization of the Brown 1	1000	Wrigth and Holst zero residual	200

Table 3. List of Andrei's test problems

Problem name	Dim	Problem name	Dim
Almost Perturbed Quadratic	10000	Extended Tridiagonal 2	20000
Diagonal 1	10	Extended White and Holst	20000
Diagonal 2	10000	Fletcher	500
Diagonal 3	100	Generalized PSC1	30000
Diagonal 4	30000	Genaralized Tridiagonal 1	30000
Diagonal 5	30000	Generalized Tridiagonal 2	200
Extended Beale	30000	Generalized Rosenbrock	1000
Extended BD1	30000	Generalized White and Holst	500
Extended Cliff	30000	Hager	2000
Extended Himmelblau	30000	Perturbed Quadratic	3000
Extended Maratos	30000	Perturbed Quadratic diagonal	5000
Extended Penalty	1000	Purterbed Tridiagonal quadratic	5000
Extended Powell	20000	Quadratic QF1	10000
Extended PSC1	30000	Quadratic QF2	10000
Extended quadratic exponential EP1	10000	Raydan 1	1000
Extended quadratic penalty QP1	10000	Raydan 2	1000
Extended quadratic penalty QP2	10000	Staircase 1	4
Extended TET	30000	Staircase 1	3
Extended Tridiagonal 1	30000		