A generalized worst-case complexity analysis for non-monotone line searches

Geovani N. Grapiglia & Ekkehard W. Sachs

Numerical Algorithms

ISSN 1017-1398
Volume 87
Number 2

Numer Algor (2021) 87:779-796
DOI 10.1007/s11075-020-00987-6

Springer

Numerical Algorithms (2021) 87:779–796
https://doi.org/10.1007/s11075-020-00987-6

ORIGINAL PAPER

A generalized worst-case complexity analysis for non-monotone line searches

Geovani N. Grapiglia[1] · Ekkehard W. Sachs[2]

Received: 8 November 2019 / Accepted: 17 July 2020 / Published online: 6 August 2020

Abstract
We study the worst-case complexity of a non-monotone line search framework that covers a wide variety of known techniques published in the literature. In this framework, the non-monotonicity is controlled by a sequence of nonnegative parameters. We obtain complexity bounds to achieve approximate first-order optimality even when this sequence is not summable.

Keywords Nonlinear optimization · Unconstrained optimization · Non-monotone line search · Worst-case complexity

1 Introduction

The worst-case complexity analysis of algorithms for non-convex optimization has become a very active research area [31, 32, 34]. This type of analysis aims at an estimate for the maximum number of iterations that an algorithm needs to generate an ϵ-approximate critical point of the objective function. The numerical schemes for smooth unconstrained optimization considered so far include line search algorithms [7, 12, 19, 32, 36], trust-region algorithms [14, 17, 18, 21], and regularization algorithms [4, 8–11, 16, 20, 27, 33, 40].

In most of these studies, the algorithms that were analyzed are monotone, that is, they do not allow an increase in the values of the objective function in successive iterations. In this paper, we consider a whole family of non-monotone step-size rules and analyze their complexity. This is carried out by using a general algorithmic

✉ Geovani N. Grapiglia
grapiglia@ufpr.br

Ekkehard W. Sachs
sachs@uni-trier.de

[1] Departamento de Matemática, Universidade Federal do Paraná, Centro Politécnico, Cx. Postal 19.081, Curitiba, Paraná 81531-980, Brazil

[2] Department of Mathematics, University of Trier, Trier, 54286, Germany

framework, extending the work in [37]. The framework is built upon a generalized Armijo rule in which the non-monotonicity is controlled by a sequence $\{\nu_k\}$ of non-negative real numbers. It was shown in [19] that, if the sequence $\{\nu_k\}$ is summable, the algorithms in the class take at most $\mathcal{O}(\epsilon^{-2})$ iterations to find ϵ-approximate critical points. Here, we relax the summability assumption and provide complexity estimates for the resulting non-monotone schemes. As a by-product, we obtain a unified liminf-type global convergence result for non-monotone schemes in which $\nu_k \to 0$, covering the non-monotone rules in [23] and [41]. Compared with these approaches, the analysis presented here is remarkably simple and our generalized results allow more freedom for the development of new non-monotone line search algorithms. As an example, we design a non-monotone step size rule related to the Metropolis rule.

The paper is organized as follows. In Section 2, we present worst-case complexity estimates. We use these estimates to derive in a new way global convergence results as outlined in Section 3. In Section 4, a Metropolis-based non-monotone rule is motivated and defined. We report preliminary numerical experiments in Section 5.

2 Worst-case complexity analysis

Given a Hilbert space $(X, \langle\,.\,,\,.\,\rangle)$, we consider the minimization problem:

$$\min_{x\in X} f(x), \tag{1}$$

where $f : X \to \mathbb{R}$ is Fréchet differentiable. We shall denote the gradient of f at $x \in X$ by $\nabla f(x)$. Furthermore, given $x_k \in X$, we call $d_k \in X$ a descent direction for f at x_k if $\langle \nabla f(x_k), d_k\rangle < 0$. Finally, we shall denote the norm induced by the inner product $\langle\,.\,,\,.\,\rangle$ by $\|\cdot\|$.

We will consider the following general descent algorithm with a non-monotone Armijo line search, which is a slight modification of the scheme proposed by Sachs and Sachs [37].

Algorithm 1 General non-monotone descent algorithm.

Step 0 Given $x_0 \in X$, $\alpha_0 > 0$ and $\beta, \rho \in (0, 1)$, set $k := 0$.
Step 1 Compute a descent direction $d_k \in X$ for x_k.
Step 2.1 Set $l := 0$.
Step 2.2 Choose $\nu_{k,l} \geq 0$. If

$$f(x_k + \alpha_k\beta^l d_k) \leq f(x_k) + \rho\alpha_k\beta^l \langle \nabla f(x_k), d_k\rangle + \nu_{k,l} \tag{2}$$

set $l_k = l$, $\nu_k = \nu_{k,l_k}$ and go to Step 3. Otherwise, set $l := l + 1$ and repeat Step 2.2.
Step 3 Set $x_{k+1} = x_k + \alpha_k\beta^{l_k} d_k$, $\alpha_{k+1} = \alpha_k\beta^{l_k-1}$, $k := k + 1$ and go to Step 1.

Remark 1 The root of the non-monotone term $f(x_k) + \nu_{k,l}$ in Algorithm 1 can be traced back to [25] and [13]. In addition, a trust region with line search using a similar non-monotone term has been proposed in [39].

Remark 2 The difference between Algorithm 1 and the general scheme in [37] is that at any given iteration k, instead of using a fixed non-monotone term ν_k, we allow it to change within the line search procedure. This flexibility allows to cover the non-monotone rule described in Section 4.

To analyze the worst-case complexity of Algorithm 1, we shall consider the following assumptions:

A1 The objective function $f : X \to \mathbb{R}$ is Fréchet differentiable and its gradient $\nabla f : X \to X$ is Lipschitz continuous with Lipschitz constant $L > 0$.

A2 There exists $f_{low} \in \mathbb{R}$ such that $f(x) \geq f_{low}$ for all $x \in X$.

A3 For all k,

$$\langle \nabla f(x_k), d_k \rangle \leq -c_1 \|\nabla f(x_k)\|^2 \quad \text{and} \quad \|d_k\| \leq c_2 \|\nabla f(x_k)\|$$

for some constants $c_1, c_2 > 0$.

Lemma 1 *Suppose $f : X \to \mathbb{R}$ is Fréchet differentiable and its gradient $\nabla f : X \to X$ is Lipschitz continuous with Lipschitz constant $L > 0$:*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in X.$$

Then,

$$f(y) \leq f(x) - \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \quad \forall x, y \in X. \tag{3}$$

Proof See, for example, Theorem 1.2.22 in [38]. □

The next lemma provides a lower bound on α_k.

Lemma 2 *Suppose that A1 and A3 hold. Then, for all k,*

$$\alpha_k \geq \min \left\{ \alpha_0, \frac{2(1-\rho)c_1}{Lc_2^2} \right\} \equiv \bar{\alpha}. \tag{4}$$

Proof Since $\nu_{k,l} \geq 0$ for all k and l, the result can be shown as in the proof of Lemma 2 in [19]. □

The first theorem gives an upper bound on the total number of function evaluations after $k \geq 1$ iterations.

Theorem 1 *Suppose that A1 and A3 hold and let N_k be the total number of function evaluations up to the k-th iteration of Algorithm 1. Then,*

$$N_k \leq 2(k+1) + \frac{1}{\log(\beta)} \left[\log(\bar{\alpha}) - \log(\alpha_0) \right], \tag{5}$$

where $\bar{\alpha}$ is defined in Lemma 2.

Proof Theorem 3 in [19] applies here, since the proof only uses $\alpha_{k+1} = \beta^{\ell_k - 1}\alpha_k$ and the bound $\alpha_k \geq \bar{\alpha}$ for all k. □

Remark 3 From (5), we see that in Algorithm 1 the average number of function evaluations per iteration, up to the k-th iteration, is asymptotically bounded by 2:

$$\frac{N_k}{k} \leq 2\left(1 + \frac{1}{k}\right) + \frac{1}{k}\frac{\log(\bar{\alpha}) - \log(\alpha_0)}{\log(\beta)}.$$

Now, define:

$$\kappa_c = \min\left\{\rho\beta\alpha_0 c_1, \frac{2\beta\rho(1-\rho)c_1^2}{Lc_2^2}\right\}. \tag{6}$$

With respect to sequence $\{\nu_k\}_{k=0}^{+\infty}$ that controls the amount of the non-monotonicity, we shall consider the following assumption:

A4 $\lim_{T\to+\infty} \frac{1}{T}\sum_{k=0}^{T-1} \nu_k = 0.$

Note that, if $\sum_{k=0}^{+\infty}\nu_k < +\infty$, then A4 is satisfied. However, A4 also may be satisfied for sequences that are not summable. An example is $\nu_k = M/(k+1)$, with $M > 0$, for which $\sum_{k=0}^{+\infty}\nu_k = +\infty$ but:

$$\lim_{T\to+\infty} \frac{1}{T}\sum_{k=0}^{T-1} \nu_k \leq \lim_{T\to+\infty} \frac{M}{T}\left[\ln(T) + 1\right] = 0.$$

Therefore, the complexity analysis presented here includes non-monotone terms that were not covered by the analysis in [19].

Given $\epsilon > 0$, under the assumption A4, we shall denote by $T_0(\epsilon)$ any non-negative integer such that:

$$T \geq T_0(\epsilon) \Longrightarrow \frac{1}{T}\sum_{k=0}^{T-1} \nu_k \leq \frac{\kappa_c \epsilon^2}{2}, \tag{7}$$

where κ_c is given by (6).

Our next theorem establishes an upper bound on the number of iterations necessary for Algorithm 1 generate x_k such that $\|\nabla f(x_k)\| \leq \epsilon$. Using (7), the proof follows by adapting the proof of Theorem 1 in [19].

Theorem 2 *Suppose that A1–A4 hold and let the sequence $\{x_k\}_{k=0}^{+\infty}$ be generated by Algorithm 1. If:*

$$T \geq \max\left\{T_0(\epsilon), \frac{2(f(x_0) - f_{low})}{\kappa_c \epsilon^2}\right\}, \tag{8}$$

then

$$\min_{k=0,\ldots,T-1} \|\nabla f(x_k)\| \leq \epsilon. \tag{9}$$

Proof It follows from (2), A3 and Lemma 2 that:

$$\begin{aligned} \nu_k + f(x_k) - f(x_{k+1}) &\geq \rho\alpha_k\beta^{l_k}\left(-\langle\nabla f(x_k), d_k\rangle\right) \\ &\geq \rho\beta\alpha_{k+1}c_1\|\nabla f(x_k)\|^2 \\ &\geq \kappa_c\|\nabla f(x_k)\|^2, \end{aligned} \tag{10}$$

where κ_c is defined in (6). Summing up these inequalities for $k = 0, \ldots, T-1$, and using A2, we get:

$$\sum_{k=0}^{T-1}\kappa_c\|\nabla f(x_k)\|^2 \leq f(x_0) - f(x_T) + \sum_{k=0}^{T-1}\nu_k \leq f(x_0) - f_{low} + \sum_{k=0}^{T-1}\nu_k.$$

Consequently,

$$\kappa_c T \min_{k=0,\ldots,T-1}\|\nabla f(x_k)\|^2 \leq f(x_0) - f_{low} + \sum_{k=0}^{T-1}\nu_k,$$

which gives

$$\min_{k=0,\ldots,T-1}\|\nabla f(x_k)\|^2 \leq \frac{f(x_0) - f_{low}}{\kappa_c T} + \frac{1}{\kappa_c T}\sum_{k=0}^{T-1}\nu_k. \tag{11}$$

Since (8) holds, we have $T \geq T_0(\epsilon)$, and so it follows from (7) that:

$$\frac{1}{\kappa_c T}\sum_{k=0}^{T-1}\nu_k \leq \frac{\epsilon^2}{2}. \tag{12}$$

On the other hand, also by (8) we have

$$\frac{f(x_0) - f_{low}}{\kappa_c T} \leq \frac{\epsilon^2}{2}. \tag{13}$$

Combining (11), (12), and (13), we have:

$$\min_{k=0,\ldots,T-1}\|\nabla f(x_k)\|^2 \leq \frac{\epsilon^2}{2} + \frac{\epsilon^2}{2} = \epsilon^2,$$

which gives (9). □

An important class of non-monotone schemes is the one that corresponds to $\{\nu_k\}$ summable. As mentioned in "Introduction", it includes, for example, the non-monotone rule of Zhang and Hager [41] and the non-monotone rule of Ahookhosh, Amini, and Bahrami [1] (for details, see Section 6 in [37]). For this class, Theorem 2 has the following consequence.

Corollary 1 *Suppose that A1–A3 hold and that* $\sum_{k=0}^{+\infty}\nu_k < +\infty$. *Let* $\{x_k\}_{k=0}^{+\infty}$ *be a sequence generated by Algorithm 1. Given* $\epsilon \in (0, 1)$, *if:*

$$T \geq 2\max\left\{\sum_{k=0}^{+\infty}\nu_k,\ f(x_0) - f_{low}\right\}\kappa_c^{-1}\epsilon^{-2}, \tag{14}$$

then

$$\min_{k=0,\ldots,T-1} \|\nabla f(x_k)\| \leq \epsilon. \tag{15}$$

Proof Note that

$$0 \leq \frac{1}{T}\sum_{k=0}^{T-1} \nu_k \leq \frac{1}{T}\sum_{k=0}^{+\infty} \nu_k, \quad \text{for all } T \geq 1.$$

Since $\{\nu_k\}$ is summable, it follows that A4 is satisfied. Moreover,

$$T \geq \frac{2\left(\sum_{k=0}^{+\infty}\nu_k\right)}{\kappa_c\epsilon^2} \Longrightarrow \frac{\kappa_c\epsilon^2}{2} \geq \frac{1}{T}\left(\sum_{k=0}^{+\infty}\nu_k\right) \geq \frac{1}{T}\left(\sum_{k=0}^{T-1}\nu_k\right).$$

Therefore, (7) holds for:

$$T_0(\epsilon) = \frac{2\sum_{k=0}^{+\infty}\nu_k}{\kappa_c\epsilon^2},$$

and (14) can be rewritten as:

$$T \geq \max\left\{T_0(\epsilon), \frac{2(f(x_0) - f_{low})}{\kappa_c\epsilon^2}\right\}.$$

Thus, by Theorem 2, (15) must be true. □

When $\sum_{k=0}^{+\infty}\nu_k < +\infty$, Corollary 1 gives a worst-case complexity bound of $\mathcal{O}(\epsilon^{-2})$ iterations, which agrees with the bound established in [19]. The next result allows us to obtain worst-case complexity estimates even when $\{\nu_k\}$ is not summable.

Corollary 2 *Suppose that A1–A3 hold and that $\nu_k \to 0$. Let constant $C > 0$ such that $\nu_k \leq C$ for all k and, given $\delta > 0$, let $k_0(\delta)$ be a positive integer such that $\nu_k \leq \delta$ if $k \geq k_0(\delta)$. Then, for any sequence $\{x_k\}_{k=0}^{+\infty}$ generated by Algorithm 1, if:*

$$T \geq \max\left\{\frac{2k_0(\delta/2)C}{\delta}, 1 + k_0(\delta/2), \frac{2(f(x_0) - f_{low})}{\kappa_c\epsilon^2}\right\} \tag{16}$$

for $\delta = \kappa_c\epsilon^2/2$, it follows that

$$\min_{k=0,\ldots,T-1} \|\nabla f(x_k)\| \leq \epsilon. \tag{17}$$

In particular, if $\nu_k = M/k$ for all k, with $M > 0$ constant, then (17) holds if:

$$T \geq \max\left\{\frac{16M^2}{\kappa_c^2\epsilon^4}, 1 + \frac{4M}{\kappa_c\epsilon^2}, \frac{2(f(x_0) - f_{low})}{\kappa_c\epsilon^2}\right\}. \tag{18}$$

Proof Given $\delta > 0$, if:

$$T \geq \max\left\{\frac{2k_0(\delta/2)C}{\delta}, 1 + k_0(\delta/2)\right\}$$

we have:

$$\begin{aligned}\frac{1}{T}\sum_{k=0}^{T-1}\nu_k &= \frac{1}{T}\left(\sum_{k=0}^{k_0(\delta/2)-1}\nu_k\right)+\frac{1}{T}\left(\sum_{k=k_0(\delta/2)}^{T-1}\nu_k\right)\\ &\leq \frac{1}{T}\left(\sum_{k=0}^{k_0(\delta/2)-1}C\right)+\frac{1}{T}\left(\sum_{k=k_0(\delta/2)}^{T-1}\frac{\delta}{2}\right)\\ &\leq \frac{1}{T}\left(\sum_{k=0}^{k_0(\delta/2)-1}C\right)+\frac{1}{T}\left(\sum_{k=0}^{T-1}\frac{\delta}{2}\right)\\ &\leq \frac{k_0(\delta/2)C}{T}+\frac{\delta}{2}\\ &\leq \delta.\end{aligned}$$

Therefore, the assumption A4 is satisfied and (7) holds for:

$$T_0(\epsilon)=\max\left\{\frac{2k_0(\delta/2)C}{\delta},\,1+k_0(\delta/2)\right\},$$

with $\delta = \kappa_c\epsilon^2/2$. Consequently, if (16) holds, then (8) is true and the conclusion comes directly from Theorem 2. Finally, suppose that $\nu_k = M/k$ for all k. Then, $\nu_k \to +\infty$, $\nu_k \leq M$ for all k and, given $\delta > 0$:

$$\nu_k = \frac{M}{k} \leq \delta \iff k \geq \frac{M}{\delta}.$$

Hence, in this case, we have:

$$k_0(\delta) = \frac{M}{\delta} \quad \text{and} \quad C = M.$$

Therefore, the condition (16) becomes (18). □

Remark 4 Consider $\nu_k = \epsilon/k$ for all $k \geq 1$, with $\epsilon \in (0, 1)$. In this case, even though $\sum_{k=0}^{+\infty}\nu_k = +\infty$, it follows from Corollary 2 (with $M = \epsilon$) that Algorithm 1 takes at most $\mathcal{O}(\epsilon^{-2})$ iterations to generate x_k such that $\|\nabla f(x_k)\| \leq \epsilon$.

A worst-case complexity bound of $\mathcal{O}\left(\epsilon^{-2}\right)$ also can be obtained for variants of Algorithm 1 characterized by the following assumption:

A4' $\nu_k = \mathrm{o}\left(\|\nabla f(x_k)\|^2\right)$ as $k \to +\infty$, i.e., for all $\delta > 0$, there exists $n_0 \in \mathbb{N}$ such that

$$\nu_k \leq \delta\|\nabla f(x_k)\|^2, \quad \forall k \geq n_0.$$

Under assumption A4', for any $\delta > 0$, we can define the number:

$$n_0(\delta) = \min\left\{n_0 \in \mathbb{N} \mid \nu_k \leq \delta\|\nabla f(x_k)\|^2,\ \forall k \geq n_0\right\}. \tag{19}$$

One example of sequence $\{\nu_k\}$ satisfying A4' is:

$$\nu_0 = 0 \quad \text{and} \quad \nu_k = k^{-1}\|\nabla f(x_k)\|^2 \quad \forall k \geq 1.$$

The next lemma gives a finite upper bound of $\mathcal{O}(\epsilon^{-2})$ for the *total number of iterations* of Algorithm 1 in which $\|\nabla f(x_k)\| > \epsilon$ for a given $\epsilon > 0$.

Lemma 3 *Suppose that A1–A3 hold and let sequence* $\{x_k\}_{k=0}^{+\infty}$ *be generated by Algorithm 1. Given* $\epsilon > 0$*, if A4' holds, then the number of elements of the set:*

$$\Omega_\epsilon = \{k \mid \|\nabla f(x_k)\| > \epsilon\} \tag{20}$$

is bounded as follows:

$$|\Omega_\epsilon| \leq k_1 + \left[\frac{2\left(f(x_0) - f_{low} + \sum_{i=0}^{k_1-1} v_i\right)}{\kappa_c}\right]\epsilon^{-2}, \tag{21}$$

where $k_1 = n_0(\frac{\kappa_c}{2})$ *is independent of* ϵ*, with* κ_c *and* $n_0(\cdot)$ *defined in* (6) *and* (19)*, respectively.*

Proof By A4' and (19), k_1 is well-defined and

$$v_k \leq \frac{\kappa_c}{2}\|\nabla f(x_k)\|^2, \quad \forall k \geq k_1.$$

Thus, it follows from (10) that:

$$\begin{aligned}\kappa_c\|\nabla f(x_k)\|^2 &\leq f(x_k) - f(x_{k+1}) + v_k \\ &\leq f(x_k) - f(x_{k+1}) + \frac{\kappa_c}{2}\|\nabla f(x_k)\|^2, \ \forall k \geq k_1,\end{aligned}$$

which implies that

$$\frac{\kappa_c}{2}\|\nabla f(x_k)\|^2 \leq f(x_k) - f(x_{k+1}), \quad \forall k \geq k_1. \tag{22}$$

Given $0 \leq s < t$, let us define:

$$\Omega_\epsilon(s,t) = \{s \leq k \leq t \mid \nabla f(x_k)\| > \epsilon\}. \tag{23}$$

For all $t > k_1$, it follows from (22) that

$$\frac{\kappa_c}{2}\epsilon^2 \leq f(x_k) - f(x_{k+1}), \quad \forall k \in \Omega_\epsilon(k_1, t).$$

Therefore,

$$\begin{aligned}|\Omega_\epsilon(k_1,t)|\frac{\kappa_c\epsilon^2}{2} &= \sum_{k\in\Omega_\epsilon(k_1,t)} \frac{\kappa_c\epsilon^2}{2} \\ &\leq \sum_{k\in\Omega_\epsilon(k_1,t)} f(x_k) - f(x_{k+1}) \\ &\leq \sum_{k=k_1}^{t} f(x_k) - f(x_{k+1}) \\ &= f(x_{k_1}) - f(x_{t+1}) \\ &\leq f(x_{k_1}) - f_{low},\end{aligned}$$

and so

$$|\Omega_\epsilon(k_1, t)| \leq \left[\frac{2\left(f(x_{k_1}) - f_{low}\right)}{\kappa_c}\right]\epsilon^{-2}. \tag{24}$$

Since $t > k_1$ is arbitrary, by (23), (24), and (20), we get:

$$|\Omega_\epsilon| \leq k_1 + |\Omega_\epsilon(k_1, +\infty)| \leq k_1 + \left[\frac{2(f(x_{k_1}) - f_{low})}{\kappa_c}\right]\epsilon^{-2}. \tag{25}$$

Finally, notice that:

$$f(x_{k_1}) \leq f(x_0) + \sum_{i=0}^{k_1-1} \nu_i. \tag{26}$$

Thus, (21) follows directly from (25) and (26). □

3 Global convergence results

The next theorem comes as a by-product from the previous complexity estimates and yields a convergence result which simplifies known proofs substantially and generalizes other non-monotone step size rules.

Theorem 3 *Suppose that A1–A3 hold and let the sequence* $\{x_k\}_{k=0}^{+\infty}$ *be generated by Algorithm 1. If* $\nu_k \to 0$ *as* $k \to +\infty$*, then either there exists* $\bar{k}$ *such that* $\nabla f(x_{\bar{k}}) = 0$ *or*

$$\liminf_{k\to+\infty} \|\nabla f(x_k)\| = 0. \tag{27}$$

Proof Let $\epsilon > 0$. Since $\nu_k \to 0$ as $k \to +\infty$, there exist constants C and $k_0(\frac{\kappa_c\epsilon^2}{4}) > 0$ such that $\nu_k \leq C$ for all k, and $\nu_k \leq \kappa_c\epsilon^2/4$ for all $k \geq k_0(\frac{\kappa_c\epsilon^2}{4})$. Thus, from Corollary 2, if

$$T \geq \max\left\{\frac{4k_0(\frac{\kappa_c\epsilon^2}{4})C}{\kappa_c\epsilon^2}, 1 + k_0\left(\frac{\kappa_c\epsilon^2}{4}\right), \frac{2(f(x_0) - f_{low})}{\kappa_c\epsilon^2}\right\} \tag{28}$$

then

$$\min_{k=0,\ldots,T-1} \|\nabla f(x_k)\| \leq \epsilon.$$

As $\epsilon > 0$ is arbitrary, this proves that:

$$\lim_{T\to+\infty}\left(\min_{k=0,\ldots,T-1} \|\nabla f(x_k)\|\right) = 0.$$

Therefore, either there exists $\bar{k}$ for which $\|\nabla f(x_{\bar{k}})\| = 0$ or (27) is true. □

More importantly, our analysis provides a unified global convergence proof for many non-monotone methods based on the method proposed in [23], which is one of

the most used non-monotone line search algorithms. It corresponds to the modified Armijo rule:

$$f(x_k + \alpha_k \beta^{l_k} d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \rho \alpha_k \beta^{l_k} \langle \nabla f(x_k), d_k \rangle, \tag{29}$$

for a suitable choice of $m(k)$. Notice that this rule can be written in the form (2) with:

$$v_{k,l} \equiv v_k = \max_{0 \leq j \leq m(k)} f(x_{k-j}) - f(x_k).$$

Corollary 3 *Suppose that A1–A3 hold and let sequence* $\{x_k\}_{k=0}^{+\infty}$ *be generated by Algorithm 1 where* (2) *is replaced by:*

$$f(x_k + \alpha_k \beta^{l_k} d_k) \leq R_k + \rho \alpha_k \beta^{l_k} \langle \nabla f(x_k), d_k \rangle \tag{30}$$

with

$$f(x_k) \leq R_k \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) \tag{31}$$

where $m(0) = 0$ *and* $0 \leq m(k) \leq \min\{m(k-1)+1, N\}$, *for a user-defined* $N \in \mathbb{N}$. *If*

$$\left\{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\right\}$$

is bounded, then either exists $\bar{k}$ *such that* $\nabla f(x_{\bar{k}}) = 0$ *or*

$$\liminf_{k \to +\infty} \|\nabla f(x_k)\| = 0.$$

Proof As in the proof of Lemma 2 in [3], by (31), one can show that

$$\lim_{k \to +\infty} f(x_k) = \lim_{k \to +\infty} \max_{0 \leq j \leq m(k)} f(x_{k-j}). \tag{32}$$

Hence for $v_k = R_k - f(x_k)$, we obtain

$$\lim_{k \to +\infty} v_k = \lim_{k \to +\infty} R_k - f(x_k) \leq \lim_{k \to +\infty} \max_{0 \leq j \leq m(k)} f(x_{k-j}) - f(x_k) = 0.$$

Therefore, the result follows directly from Theorem 3. □

This generalized convergence result includes, for example, the non-monotone methods in [1–3, 35]. The worst-case complexity of these methods, however, depends on how fast $v_k = R_k - f(x_k)$ converges to zero. Due to the max $\{\cdot\}$ on the right-hand side of (16), the best iteration-complexity bound that one can get from Corollary 2 is $\mathcal{O}(\epsilon^{-2})$. This is exactly the complexity obtained by Cartis, Sampaio, and Toint [12] for a non-monotone method based on rule (29).

Notice that Theorem 3 gives a liminf-type convergence result. An improved lim-type result can be obtained for variants of Algorithm 1 characterized by assumption A4'. Indeed, from the complexity estimate given in Lemma 3, we can establish the global convergence of Algorithm 1 with the same argument used to prove Corollary 2.1 in [27].

Theorem 4 *Suppose that A1–A3 hold and let the sequence* $\{x_k\}_{k=0}^{+\infty}$ *be generated by Algorithm 1. If A4' also holds, then:*

$$\lim_{k\to+\infty} \|\nabla f(x_k)\| = 0. \tag{33}$$

Proof Suppose that (33) does not hold. Then, there exists $\epsilon > 0$ and a subsequence $\{x_{k_j}\}_{j=0}^{+\infty}$ of $\{x_k\}_{k=0}^{+\infty}$ such that:

$$\|\nabla f(x_{k_j})\| > \epsilon, \quad \forall j \in \mathbb{N}.$$

This means that the corresponding set $\Omega_\epsilon = \{k \mid \|\nabla f(x_k)\| > \epsilon\}$ is infinite, contradicting Lemma 3. □

4 A metropolis-based non-monotone rule

One of the core ideas of non-monotone rules is to allow the iterates to escape from local minimizers and to increase the probability of finding global minimizers. In the context of derivative-free heuristics for global optimization, simulated annealing [24, 26, 28] is one of the most efficient schemes. At the kth iteration of a simulated annealing algorithm, the acceptance or rejection of a candidate point x_k^+ is usually done by the *Metropolis rule*: given a uniform random number $p_k \in [0, 1]$, the next iterate is set as:

$$x_{k+1} = \begin{cases} x_k^+, & \text{if } p_k \le \min\left\{1, \exp\left(-\frac{f(x_k^+)-f(x_k)}{\tau_k}\right)\right\}, \\ x_k, & \text{otherwise,} \end{cases} \tag{34}$$

where $\tau_k > 0$ for all k, with $\tau_k \to 0$. By rule (34), if $f(x_k^+) \le f(x_k)$ then x_k^+ is always accepted, i.e., $x_{k+1} = x_k^+$. However, the candidate point x_k^+ also can be accepted when $f(x_k^+) > f(x_k)$, allowing the iterates to escape from local minimizers. The larger the difference $f(x_k^+) - f(x_k) > 0$ is, the smaller is the probability to accept x_k^+. Since $\tau_k \to 0$, the probability of accepting x_k^+ when $f(x_k^+) > f(x_k)$ also goes to zero when $k \to +\infty$.

Back to Algorithm 1, notice that the bigger is the non-monotone parameter $\nu_{k,l}$, the bigger is the chance to accept a candidate point $x_{k,l}^+ = x_k + \alpha_k \beta^l d_k$ with $f(x_{k,l}^+) > f(x_k)$. Thus, we can try to mimic the Metropolis acceptance rule by choosing $\nu_{k,l}$ as follows:

Step 2.1 Set $l := 0$.
Step 2.2 Compute $x_{k,l}^+ = x_k + \alpha_k \beta^l d_k$ and define:

$$\nu_{k,l} = \sigma \exp\left(-\frac{\max\left\{\theta, f(x_{k,l}^+) - f(x_k)\right\}}{\tau_k}\right) \tag{35}$$

for some constants $\sigma, \theta > 0$ independent of k and l, with $\tau_k = 1/\ln(k+1)$. If

$$f(x_{k,l}^+) \le f(x_k) + \rho\alpha_k\beta^l\langle\nabla f(x_k), d_k\rangle + \nu_{k,l}$$

set $l_k = l$ and $\nu_k = \nu_{k,l_k}$. Otherwise, set $l := l + 1$ and repeat Step 2.2.

The next two theorems establish complexity bounds of $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-\frac{2(1+\theta)}{\theta}})$ for Algorithm 1, when $\theta > 1$ and $\theta \in (0, 1]$, respectively.

Theorem 5 *Suppose that A1–A3 hold and let the sequence $\{x_k\}_{k=0}^{+\infty}$ be generated by Algorithm 1 with $\nu_{k,l}$ defined by (35). Given $\epsilon > 0$, if $\theta > 1$, and:*

$$T \geq 2 \max \left\{ \sigma \sum_{k=0}^{+\infty} \frac{1}{(k+1)^{\theta}},\ f(x_0) - f_{low} \right\} \kappa_c^{-1} \epsilon^{-2}, \tag{36}$$

then

$$\min_{k=0,\ldots,T-1} \|\nabla f(x_k)\| \leq \epsilon. \tag{37}$$

Proof By (35), for all k we have:

$$\nu_k = \sigma e^{-\max\{\theta, f(x_{k+1}) - f(x_k)\} \ln(k+1)} \leq \sigma \left(\frac{1}{k+1} \right)^{\theta}. \tag{38}$$

Thus,

$$\sum_{k=0}^{+\infty} \nu_k = \sigma \sum_{k=0}^{+\infty} \left(\frac{1}{k+1} \right)^{\theta} < +\infty,$$

and Corollary 1 yields the result. □

Theorem 6 *Suppose that A1–A3 hold and let sequence $\{x_k\}_{k=0}^{+\infty}$ be generated by Algorithm 1 with $\nu_{k,\ell}$ defined by (35). Given $\epsilon \in (0, 1)$, if $\theta \in (0, 1]$ and:*

$$T \geq \max \left\{ \left(\frac{4}{\kappa_c} \right)^{\frac{1+\theta}{\theta}} \sigma^{\frac{1}{\theta}},\ 1 + \left(\frac{4\sigma}{\kappa_c} \right)^{\frac{1}{\theta}},\ \frac{2(f(x_0) - f_{low})}{\kappa_c} \right\} \epsilon^{-\frac{2(1+\theta)}{\theta}}, \tag{39}$$

then (37) holds.

Proof By (38), we have $\nu_k \to 0$. Moreover, $\nu_k \leq \sigma$ and given $\delta > 0$,

$$\nu_k \leq \delta \quad \text{if} \quad k \geq \left(\frac{\sigma}{\delta} \right)^{\frac{1}{\theta}}.$$

Denote:

$$C = \sigma \quad \text{and} \quad k_0(\delta) = \left(\frac{\sigma}{\delta} \right)^{\frac{1}{\theta}}.$$

Taking $\delta = \kappa_c \epsilon^2 / 2$, it follows from (39) that:

$$\begin{aligned} T &\geq \max \left\{ \left(\frac{4}{\kappa_c} \right)^{\frac{1+\theta}{\theta}} \sigma^{\frac{1}{\theta}} \epsilon^{-\frac{2(1+\theta)}{\theta}},\ 1 + \left(\frac{4\sigma}{\kappa_c} \right)^{\frac{1}{\theta}} \epsilon^{-\frac{2}{\theta}},\ \frac{2(f(x_0) - f_{low})}{\kappa_c} \epsilon^{-2} \right\} \\ &= \max \left\{ \frac{2k_0(\delta/2)C}{\delta},\ 1 + k_0(\delta/2),\ \frac{2(f(x_0) - f_{low})}{\kappa_c \epsilon^2} \right\}. \end{aligned}$$

Thus, by Corollary 2, (37) must be true. □

Remark 5 The smaller is θ, the bigger is the chance to accept $x_{k,\ell}^+$ with $f(x_{k,\ell}^+) > f(x_k)$. Thus, the higher level of non-monotonicity obtained with $\theta \in (0, 1]$ may lead to better local minimizers. However, this has a price: by Theorem 6, the number of iterations that Algorithm 1 needs to find approximate stationary points may be significantly bigger in comparison with the case $\theta > 1$.

5 Preliminary numerical experiments

We performed some numerical experiments comparing Octave implementations of six instances of Algorithm 1. Specifically, we considered the following codes:

(i) The monotone algorithm obtained from Algorithm 1 by setting $\nu_{k,l} = 0$ for all k and l. We shall refer to this code as "M1".

(ii) The non-monotone algorithm in [23] obtained from Algorithm 1 by setting $\nu_{k,l} = \max_{0\le j\le m_k}\left[f(x_{k-j})\right] - f(x_k)$ for all k and l, where $m(0) = 0$ and $m(k) = \min[m(k-1)+1, 10]$. We shall refer to this code as "NM1".

(iii) The non-monotone algorithm in [41] obtained from Algorithm 1 by setting $\nu_{k,l} = C_k - f(x_k)$ for all k and l, where $C_0 = f(x_0)$ and, for all $k \ge 1$,

$$C_k = \frac{\eta_{k-1} Q_{k-1} C_{k-1} + f(x_k)}{Q_k} \quad \forall k \ge 1,$$

$Q_k = \eta_{k-1} Q_{k-1} + 1$ and $\eta_{k-1} = 0.85/k$, with $Q_0 = 1$. We shall refer to this code as "NM2".

(iv) The non-monotone algorithm obtained from Algorithm 1 by setting $\nu_{0,l} = 0$, and $\nu_{k,l} = \epsilon/k$ for $k \ge 1$, where ϵ is the desired precision for the norm of the gradient. We will refer to this code as "NM3".

(v) The non-monotone algorithm obtained from Algorithm 1 by setting $\nu_{0,l} = 0$, and $\nu_{k,l} = \gamma_k \|\nabla f(x_k)\|_2^2$ with $\gamma_k = \|\nabla f(x_0)\|_2^{-2}\left(\frac{1}{k}\right)$, for $k \ge 1$. We will refer to this code as "NM4".

(vi) The non-monotone algorithm obtained from Algorithm 1 by setting $\nu_{k,l}$ as in (35), with user-define positive parameters σ and θ. We shall refer to this code as "NM5(σ, θ)".

In all implementations, we consider the parameters $\alpha_0 = 1$ and $\beta = \rho = 0.5$. The search directions were generated as $d_k = -H_k \nabla f(x_k)$, where H_k is computed using the BFGS update whenever it is possible, namely:

$$H_{k+1} = \begin{cases} \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) H_k \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k}, & \text{if } s_k^T y_k > 0, \\ H_k, & \text{otherwise.} \end{cases}$$

where $H_0 = I$, $s_k = x_{k+1} - x_k$, and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. All the experiments were performed with Octave 4.2.2 on a PC with a 2.70 GHz Intel(R) i5 microprocessor.

Table 1 Numerical results for problems from the Moré-Garbow-Hillstrom collection [30]

PROBLEM (n_p^*)	M1	NM1	NM2	NM3	NM4	NM5(ϵ, 2)	NM5(ϵ, 1)
1. (43)	1.000	1.162	1.000	1.558	1.000	1.581	1.534
2. (27)	1.037	4.629	1.037	1.666	1.000	1.444	1.629
3. (173)	1.017	1.034	1.017	F	1.000	F	F
4. (50)	1.480	4.680	1.140	1.260	1.000	1.300	1.260
5. (17)	1.058	1.529	1.058	1.764	1.000	2.235	1.705
6. (36)	1.000	5.666	1.000	3.083	1.000	1.027	3.055
7. (34)	1.000	1.323	1.000	1.205	1.000	1.676	1.176
8. (19)	1.000	2.473	1.000	2.421	1.000	1.105	2.368
9. (3)	1.333	1.333	1.333	1.333	1.333	1.000	1.000
10. (305)	F	F	F	F	F	1.000	F
11. (9)	1.000	1.111	1.000	10.000	1.000	1.111	10.000
12. (25)	1.000	3.240	1.200	4.440	1.000	1.120	4.120
13. (39)	1.000	1.102	1.000	6.000	1.000	1.025	6.461
14. (85)	1.000	1.729	1.000	1.741	1.000	1.023	1.729
15. (24)	1.000	1.166	1.000	1.666	5.416	1.083	1.625
16. (83)	F	1.108	F	1.698	F	1.000	1.686
17. (79)	1.000	1.075	1.000	1.050	1.000	F	F
18. (37)	1.054	1.081	1.054	1.000	1.054	1.432	5.648
19. (49)	1.081	1.000	1.081	1.612	1.040	1.428	1.591
20. (57)	1.000	1.263	1.000	1.666	1.000	2.368	1.649
21. (67)	1.000	1.910	1.000	1.835	1.000	1.194	1.820
22. (39)	1.000	1.102	1.000	6.000	1.000	1.025	6.461
23. (63)	1.158	4.031	1.158	2.857	1.158	1.000	2.857
24. (16)	1.187	1.187	1.187	5.562	1.000	1.375	5.500
25. (60)	1.000	3.350	1.000	1.650	1.000	1.083	1.633
26. (58)	F	F	F	1.000	F	F	1.000
27. (10)	1.900	1.000	1.900	3.300	F	1.100	1.400
28. (8)	1.750	1.500	1.000	2.500	1.125	4.125	2.750
29. (31)	1.032	1.709	1.000	1.032	1.032	2.193	1.032
30. (38)	1.078	1.473	1.000	5.078	1.078	2.342	5.052
31. (1)	4.000	3.000	3.000	3.000	3.000	1.000	1.000
32. (20)	1.000	1.000	1.000	1.000	1.000	1.000	1.000
33. (18)	1.055	1.000	1.055	1.000	1.000	1.000	1.000
$\rho_s(1)$	0.454	0.121	0.515	0.121	0.606	0.212	0.151

In our first experiment, we applied the referred codes to the set of problems from [30].[1] We used the stopping rules:

$$\|\nabla f(x_k)\|_2 \leq 10^{-5}, \tag{40}$$

[1] We considered the same dimensions as in [19].

and

$$k = k_{\max} \equiv 500. \tag{41}$$

We declare that a problem p was solved by a solver s when s stopped due to (40). In this case, let:

$$n_{p,s} = \text{the number of iterations required to solve problem } p \text{ by solver } s.$$

As proposed in [15], the relative performance of solver s on problem p can be measured in terms of the *performance ratio*:

$$r_{p,s} = \frac{n_{p,s}}{n_p^*}, \quad \text{where } n_p^* = \min\left\{n_{p,s} \,:\, s \in \mathcal{S}\right\}.$$

Using $r_{p,s}$, the *performance profile* for each code s is defined as:

$$\rho_s(\tau) = \frac{\text{no. of problems s.t. } r_{p,s} \leq \tau}{\text{total no. of problems}}.$$

Note that $\rho_s(1)$ is the percentage of problems for which the solver s wins over the rest of the solvers (i.e., $n_{p,s} = n_p^*$). The usual graphs of performance profiles are not very informative in this case because of the superposition of a large number of lines (one for each code), which makes the interpretation of the results difficult. Therefore, we summarize the relevant information at Table 1. Specifically, we report the performance ratio $r_{p,s}$ for each pair (p, s) in our test set, and $\rho_s(1)$ for each solver s.

An entry "F" indicates that the corresponding code stopped due to (41). As we can see, solver NM4 was the most efficient (winning on 60.6% of the problems), while solvers NM1 and NM3 were the most robust (failing in only two problems). The superior performance of NM2 and NM4 is in line with the empirical evidence that it is better to start with a bigger non-monotone term far from a critical point and to have a smaller one close to it (see, e.g., [1, 35]). Moreover, the fact that NM5(ϵ, 2) was more efficient than NM5(ϵ, 1) is in accordance with Theorems 5 and 6 (recall Remark 5). These numerical results illustrate the ability of our new non-monotone rules for finding approximate critical points.

Table 2 Results for the Griewank function: distributions of the best function values found by each code within 500 iterations

	M1	NM1	NM2	NM3	NM4	NM5(ϵ, 2)
Maximum	179.8002	136.3502	179.8002	179.8002	179.8002	179.8002
75th percentile	119.1955	89.9534	119.1955	119.1955	119.1955	119.1955
Median	82.7324	25.2736	82.7324	82.7324	78.1701	82.7324
25th percentile	34.0983	9.7496	28.9691	34.0983	34.0983	34.0983
Minimum	10.1014	0.3353	10.1014	10.1014	10.1014	10.1014

Table 3 Results for code NM5 with $M = |f(x_0)|$ and different values of θ

NM5($\|f(x_0)\|, \theta$)	$\theta = 4$	$\theta = 2$	$\theta = 1$	$\theta = 0.5$	$\theta = 0.25$	$\theta = 0.125$
Maximum	136.3843	124.3656	70.2559	19.2514	10.1188	18.2874
75th percentile	99.6332	96.3803	29.7006	6.8724	4.9171	2.6889
Median	62.0849	70.6839	19.2193	2.1444	1.6082	0.9238
25th percentile	34.0983	19.2036	10.1014	0.7201	0.3102	0.2367
Minimum	10.1014	5.8595	1.1145	0.0377	0.0404	0.0609

In order to investigate the ability of non-monotone methods for finding better local optima, we applied all codes to minimize the two-dimensional Griewank function [22]:

$$f(x) = 1 + \frac{x_1^2}{4000} + \frac{x_2^2}{4000} - \cos(x_1)\cos\left(x_2/\sqrt{2}\right). \tag{42}$$

This function has a huge number of local minimizers but only one global minimizer, namely $x^* = (0, 0)$ with $f(x^*) = 0$. We considered 60 initial points generated in the box $[-600, 600] \times [-600, 600]$:

$$\left(-600 + \frac{1200(i-1)}{3}, -600 + \frac{1200(j-1)}{14}\right), \quad i = 1, \ldots, 4, \;\; j = 1, \ldots, 15.$$

For each starting point, we recorded the best function value found by each code within 500 iterations. The distributions of these values are summarized in Table 2.

From Table 2, we see that code NM1 (with more aggressive non-monotone behavior) was much better than the other codes in terms of the best function values found. Since in NM5 the non-monotonicity can be increased by increasing σ or decreasing θ, we set $\sigma = |f(x_0)|$ and tested several values of θ. The distributions of the best function values found by each variant of NM5 within 500 iterations are summarized on Table 3.

As expected, the best function values were obtained with small values of θ (see Remark 5). Moreover, the function values obtained with $\theta \leq 0.5$ were significantly better than the values obtained with NM1. These preliminary results confirm the ability of non-monotone methods of escaping from the closest local minimizers. Moreover, they suggest that non-monotone line searches based on the Metropolis rule (as in NM5) may be competitive with standard non-monotone methods on difficult problems with many non-global local minimizers.

6 Conclusion

In this paper, we investigated the worst-case complexity of a generalized version of the non-monotone line search framework proposed in [37] for smooth unconstrained optimization problems. In this framework, the level of non-monotonicity is controlled by a sequence $\{\nu_k\}$ of non-negative parameters. In a previous paper [19], we proved that the algorithms in the referred framework take at most $\mathcal{O}(\epsilon^{-2})$ iterations to find ϵ-critical points, when the objective f is nonconvex. For that, we had to assume that

$\sum_{k=0}^{+\infty} \nu_k < +\infty$. Now, by refining our analysis, we were able to obtain bounds of the same order even when $\sum_{k=0}^{+\infty} \nu_k = +\infty$. Our generalized results include a unified global convergence proof for non-monotone schemes in which $\nu_k \to 0$, allowing more freedom for the design of new non-monotone line search algorithms. As a topic for future research, it would be interesting to investigate the possible extension of our results to inexact subsampled methods for minimizing finite sums [5, 6].

Acknowledgments We are very grateful to three anonymous referees, whose comments helped improve significantly the paper. We are also grateful to Masoud Ahookhosh for his insightful comments on the first version of this work.

Funding information G. N. Grapiglia was partially supported by the National Council for Scientific and Technological Development - Brazil (grants 401288/2014-5 and 406269/2016-5).

References

1. Ahookhosh, M., Amini, K., Bahrami, S.: A class of nonmonotone Armijo-type line search method for unconstrained optimization. Optimization **61**, 387–404 (2012)
2. Ahookhosh, M., Ghaderi, S.: On efficiency of nonmonotone Armijo-type line searches. Appl. Math. Model. **43**, 170–190 (2017)
3. Amini, K., Ahookhosh, M., Nosratipour, H.: An inexact line search approach using modified nonmonotone strategy for unconstrained optimization. Numerical Algorithms **60**, 49–78 (2014)
4. Bellavia, S., Gurioli, G., Morini, B.: Adaptive cubic regularization methods with dynamic inexact Hessian information and applications to finite-sum minimization. IMA Journal of Numerical Analysis, drz076. https://doi.org/10.1093/imanum/drz076 (2020)
5. Bellavia, S., Krejić, N., Jerinkić, N.K.: Subsampled inexact Newton methods for minimizing large sums of convex functions. IMA Journal of Numerical Analysis, drz027. https://doi.org/10.1093/imanum/drz027 (2020)
6. Bellavia, S., Jerinkić, N.K., Malaspina, G.: Subsampled nonmonotone spectral gradient methods. Communications in Applied and Industrial Mathematics **11**, 19–34 (2020)
7. Bergou, E., Diouane, Y., Gratton, S.: A line-search algorithm inspired by the adaptive cubic regularization framework and complexity analysis. Journal on Optimization Theory and Applications **178**, 885–913 (2018)
8. Birgin, E.G., Gardenghi, J.L., Martínez, J.M., Santos, S.A., Toint, P.L.: Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. Math. Program. **163**, 359–368 (2017)
9. Birgin, E.G., Martínez, J.M.: The use of quadratic regularization with a cubic descent condition for unconstrained optimization. SIAM J. Optim. **27**, 1049–1074 (2017)
10. Cartis, C., Gould, N.I.M., Toint, P.L.: On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. SIAM J. Optim. **20**, 2833–2852 (2010)
11. Cartis, C., Gould, N.I.M., Toint, P.L.: Adaptive cubic regularization methods for unconstrained optimization. Part II: worst-case function - and derivative - evaluation complexity. Math. Program. **130**, 295–319 (2011)
12. Cartis, C., Sampaio, P.R., Toint, P.L.: Worst-case evaluation complexity of first-order non-monotone gradient-related algorithms for unconstrained optimization. Optimization **64**, 1349–1361 (2015)
13. La Cruz, W., Noguera, G.: Hybrid spectral gradient method for the unconstrained minimization problem. J. Glob. Optim. **44**, 193–212 (2009)
14. Curtis, F.E., Robinson, D.P., Samadi, M.: A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. Math. Program. **162**, 1–32 (2017)
15. Dolan, E., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–2013 (2002)

16. Dussault, J.-P.: ARCq: a new adaptive regularization by Cubics variant. Optimization Methods and Software **33**, 322–335 (2018)
17. Grapiglia, G.N., Yuan, J., Yuan, Y.: On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization. Math. Program. **152**, 491–520 (2015)
18. Grapiglia, G.N., Yuan, J., Yuan, Y.: Nonlinear stepsize control algorithms: complexity bounds for first-and second-order optimality. J. Optim. Theory Appl. **171**, 980–997 (2016)
19. Grapiglia, G.N., Sachs, E.W.: On the worst-case evaluation complexity of non-monotone line search algorithms. Comput. Optim. Appl. **68**, 555–577 (2017)
20. Grapiglia, G.N., Nesterov, Y.: Regularized Newton methods for minimizing functions with Hölder continuous Hessians. SIAM J. Optim. **27**, 478–506 (2017)
21. Gratton, S., Sartenaer, A., Toint, P.L.: Recursive trust-region methods for multiscale nonlinear optimization. SIAM J. Optim. **19**, 414–444 (2008)
22. Griewank, A.O.: Generalized descent for global optimization. J. Optim. Theory Appl. **34**, 11–39 (1981)
23. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
24. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
25. Li, D.H., Fukushima, M.: A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. Optimization Methods & Software **13**, 181–201 (2000)
26. Locatelli, M.: Simulated annealing algorithms for continuous global optimization: convergence conditions. J. Optim. Theory Appl. **104**, 121–133 (2000)
27. Martínez, J.M., Raydan, M.: Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. J. Glob. Optim. **68**, 367–385 (2017)
28. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H.: Equation of state calculations by fast computer machines. J. Chem. Phys. **21**, 1087–1092 (1953)
29. Mo, J., Liu, C., Yan, S.: A nonmonotone trust-region method based on nonincreasing technique of weighted average of the sucessive function value. J. Comput. Appl. Math. **209**, 97–108 (2007)
30. Moré, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**, 17–41 (1981)
31. Nemirovsky, A.S., Yudin, D.B.: Problem Complexity and Method Efficiency in Optimization. Wiley, New York (1983)
32. Nesterov, Y.: Introductory Lectures on Convex Optimization: a Basic Course. Kluwer Academic Publishers, Dordrecht (2004)
33. Nesterov, Y., Polyak, B.T.: Cubic regularization of Newton method and its global performance. Math. Program. **108**, 177–205 (2006)
34. Nesterov, Y.: Lectures on Convex Optimization. Springer, Berlin (2018)
35. Nosratipour, H., Borzabadi, A.H., Fard, O.S.: On the nonmonotonicity degree of nonmonotone line searches. Calcolo **54**, 1217–1242 (2017)
36. Royer, C.W., Wright, S.J.: Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. SIAM J. Optim. **28**, 1448–1477 (2018)
37. Sachs, E.W., Sachs, S.M.: Nonmonotone line searches for optimization algorithms. Control. Cybern. **40**, 1059–1075 (2011)
38. Sun, W., Yuan, Y.: Optimization Theory and Methods: Nonlinear Programming. Springer, Berlin (2006)
39. Tarzanagh, D.A., Peyghami, M.R., Mesgarani, H.: A new nonmonotone trust region method for unconstrained optimization equipped by an efficient adaptive radius. Optimization Methods & Software **29**, 819–836 (2014)
40. Xu, P., Roosta, F., Mahoney, M.W.: Newton-type methods for non-convex optimization under inexact Hessian information. Mathematical Programming. https://doi.org/10.1007/s10107-019-01405-z (2020)
41. Zhang, H.C., Hager, W.W.: A nonmonotone line search technique for unconstrained optimization. SIAM Journal on Optimization **14**, 1043–1056 (2004)