A GENERALIZED WORST-CASE COMPLEXITY ANALYSIS FOR NON-MONOTONE LINE SEARCHES

G.N. GRAPIGLIA* AND EKKEHARD W. SACHS†

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.

Key words. Nonlinear optimization, Unconstrained optimization, Non-monotone line search, Worst-case complexity

1. Introduction.

The worst-case complexity analysis of algorithms for nonconvex optimization has become a very active research area. 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 objection function. The numerical schemes for smooth unconstrained optimization considered so far include line search algorithms [7, 12, 17, 27, 30], trust-region algorithms [13, 15, 16, 19] and regularization algorithms [4, 8, 9, 10, 11, 14, 18, 24, 28, 33].

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 framework, extending the work in [31]. The framework is built upon a generalized Armijo rule in which the non-monotonicity is controlled by a sequence $\{\nu_k\}$ of nonnegative real numbers. It was shown in [17] 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 [21] and [34]. Compared to 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 stepsize 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

$$\min_{x\in X}\, f(x), \tag{2.1}$$

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

[*] Departamento de Matemática, Universidade Federal do Paraná, Centro Politécnico, Cx. postal 19.081, 81531-980, Curitiba, Paraná, Brazil (grapiglia@ufpr.br). This author was supported by the National Council for Scientific and Technological Development - Brazil (grants 401288/2014-5 and 406269/2016-5).

[†] Department of Mathematics, University of Trier, 54286, Trier, Germany (sachs@uni-trier.de).

if $\langle \nabla f(x_k), d_k \rangle < 0$. Finally, we shall denote the norm induced by the inner product $\langle \cdot\,, \cdot \rangle$ by $\| \cdot \|$.

In what follows 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 [31].

```
**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
    (2.2)    $f(x_k + \alpha_k \beta^l d_k) \leq f(x_k) + \rho \alpha_k \beta^l \langle \nabla f(x_k), d_k \rangle + \nu_{k,l}$
    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 2.1. *The difference between Algorithm 1 and the general scheme in [31] 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 2.2. *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,

$$f(y) \leq f(x) - \langle \nabla f(x), y - x \rangle + \frac{L}{2} \| y - x \|^2 \quad \forall x, y \in X. \tag{2.3}$$

Proof. See, for example, Theorem 1.2.22 in [32]. □

The next lemma provides a lower bound on α_k. Its proof is similar to the proof of Lemma 2 in [17], and we give it here for completeness.

Lemma 2.3. *Suppose that A1 and A3 hold. Then, for all k,*

$$\alpha_k \geq \min \left\{ \alpha_0, \frac{2(1-\rho)c_1}{L c_2^2} \right\} \equiv \bar{\alpha}. \tag{2.4}$$

Proof. Obviously, (2.4) holds for $k = 0$. Assume that (2.4) holds for some $k \geq 0$. If $\ell_k = 0$, then

$$\alpha_{k+1} = \beta^{-1}\alpha_k > \alpha_k \geq \bar{\alpha},$$

that is, (2.4) also holds for $k+1$. If $\ell_k \geq 1$, by Step 2.2 and the definition of α_{k+1} we have

$$\begin{aligned} f(x_k + \alpha_k d_k) - f(x_k) &> \rho\alpha_{k+1}\langle \nabla f(x_k), d_k\rangle + \nu_{k,\ell_k - 1} \\ &\geq \rho\alpha_{k+1}\langle \nabla f(x_k), d_k\rangle. \end{aligned} \tag{2.5}$$

On the other hand, from A1 and Lemma 2.2, it follows that

$$f(x_k + \alpha_{k+1} d_k) - f(x_k) \leq \alpha_{k+1}\langle \nabla f(x_k), d_k\rangle + \frac{L\alpha_{k+1}^2}{2}\|d_k\|^2. \tag{2.6}$$

Combining (2.5), (2.6) and A3, we obtain

$$\rho\alpha_{k+1}\langle \nabla f(x_k), d_k\rangle \leq \alpha_{k+1}\langle \nabla f(x_k), d_k\rangle + \frac{L\alpha_{k+1}^2}{2}\|d_k\|^2$$

$$\Longrightarrow \alpha_{k+1} \geq \frac{2(1-\rho)}{L}\left(-\frac{\langle \nabla f(x_k), d_k\rangle}{\|d_k\|^2}\right) \geq \frac{2(1-\rho)c_1}{Lc_2^2},$$

and so, (2.4) also holds for $k+1$. □

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

THEOREM 2.4. *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,*

$$N_k \leq 2(k+1) + \frac{1}{\log(\beta)}\left[\log(\bar{\alpha}) - \log(\alpha_0)\right], \tag{2.7}$$

where $\bar{\alpha}$ is defined in Lemma 2.3.

Proof. Theorem 3 in [17] 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 2.5. *From (2.2) we see that Algorithm 1 the number of function evaluation on average*

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

can be bounded in the long run by two.

Now, define

$$\kappa_c = \min\left\{\rho\beta\alpha_0 c_1, \frac{2\beta\rho(1-\rho)c_1^2}{Lc_2^2}\right\}. \tag{2.8}$$

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

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

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

$$T \geq T_0(\epsilon) \Longrightarrow \frac{1}{T}\sum_{k=0}^{T-1} \nu_k \leq \frac{\kappa_c \epsilon^2}{2}, \tag{2.9}$$

where κ_c is given by (2.8).

Our next theorem establishes an upper bound on the number of iterations necessary for Algorithm 1 generate x_k such that $\|\nabla f(x_k)\| \leq \epsilon$.

Theorem 2.6. *Suppose that A1-A4 hold and let the sequence $\{x_k\}_{k=0}^{+\infty}$ be generated by Algorithm 1. If*

$$T \geq \max\left\{T_0(\epsilon), \frac{2(f(x_0) - f_{low})}{\kappa_c \epsilon^2}\right\}, \tag{2.10}$$

then

$$\min_{k=0,\ldots,T-1} \|\nabla f(x_k)\| \leq \epsilon. \tag{2.11}$$

Proof. It follows from (2.2), A3 and Lemma 2.3 that

$$\begin{aligned} \nu_k + f(x_k) - f(x_{k+1}) &\geq \rho \alpha_k \beta^{l_k} \left(-\langle \nabla f(x_k), d_k \rangle\right) \\ &\geq \rho \beta \alpha_{k+1} c_1 \|\nabla f(x_k)\|^2 \\ &\geq \kappa_c \|\nabla f(x_k)\|^2, \end{aligned} \tag{2.12}$$

where κ_c is defined in (2.8). 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

$$\min_{k=0,\ldots,T-1} \|\nabla f(x_k)\|^2 \leq \frac{f(x_0) - f_{low}}{\kappa_c T} + \frac{1}{\kappa_c T}\sum_{k=0}^{T-1} \nu_k. \tag{2.13}$$

Since (2.10) holds, we have $T \geq T_0(\epsilon)$, and so it follows from (2.9) that

$$\frac{1}{\kappa_c T}\sum_{k=0}^{T-1} \nu_k \leq \frac{\epsilon^2}{2}. \tag{2.14}$$

On the other hand, also by (2.10) we have

$$\frac{f(x_0) - f_{low}}{\kappa_c T} \leq \frac{\epsilon^2}{2}. \tag{2.15}$$

Combining (2.13), (2.14) and (2.15), 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 (2.11). □

An important class of non-monotone schemes is the one that corresponds to $\{\nu_k\}$ summable. As mentioned in the Introduction, it includes, for example, the non-monotone rule of Zhang and Hager [34] (for details, see Section 6 in [31]). For this class, Theorem 2.6 gives the following result.

COROLLARY 2.7. *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*

$$T\geq 2\max\left\{\sum_{k=0}^{+\infty}\nu_k,\,f(x_0)-f_{low}\right\}\kappa_c^{-1}\epsilon^{-2},\tag{2.16}$$

then

$$\min_{k=0,\ldots,T-1}\|\nabla f(x_k)\|\leq\epsilon.\tag{2.17}$$

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, (2.9) holds for

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

and (2.16) 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 2.6, (2.17) must be true. □

When $\sum_{k=0}^{+\infty}\nu_k<+\infty$, Corollary 2.7 gives a worst-case complexity bound of $\mathcal{O}(\epsilon^{-2})$ iterations, which agrees with the bound established in [17]. The next result allows us to obtain worst-case complexity estimates even when $\{\nu_k\}$ is not summable.

COROLLARY 2.8. *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*

$$T\geq\max\left\{\frac{2k_0(\delta/2)C}{\delta},1+k_0(\delta/2),\frac{2(f(x_0)-f_{low})}{\kappa_c\epsilon^2}\right\}\tag{2.18}$$

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

$$\min_{k=0,\ldots,T-1} \|\nabla f(x_k)\| \leq \epsilon. \tag{2.19}$$

In particular, if $\nu_k = M/k$ for all k, with $M > 0$ constant, then (2.19) holds if

$$T \geq \max\left\{\frac{16M^2}{\kappa_c^2 \epsilon^4}, 1 + \frac{4M}{\kappa_c \epsilon^2}, \frac{2(f(x_0) - f_{low})}{\kappa_c \epsilon^2}\right\}. \tag{2.20}$$

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, assumption A4 is satisfied and (2.9) 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 (2.18) holds, then (2.10) is true and the conclusion comes directly from Theorem 2.6. Finally, suppose that $\nu_k = M/k$ for all k. Then, $\nu_k \to 0$, $\nu_k \leq M$ for all k and, given $\delta > 0$,

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

Hence, in this case, we have

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

Therefore, condition (2.18) becomes (2.20). □

Remark 2.9. *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 2.8 (with $M = \epsilon$) that Algorithm 1 also takes at most $\mathcal{O}(\epsilon^{-2})$ iterations to generate x_k such that $\|\nabla f(x_k)\| \leq \epsilon$.*

3. Global Convergence Results. The following 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 3.1. *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*

$$\liminf_{k\to+\infty} \|\nabla f(x_k)\| = 0. \tag{3.1}$$

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 2.8, if

$$T \geq \max\left\{\frac{4k_0(\frac{\kappa_c\epsilon^2}{4})C}{\kappa_c\epsilon^2}, 1 + k_0\left(\frac{\kappa_c\epsilon^2}{4}\right), \frac{2(f(x_0) - f_{low})}{\kappa_c\epsilon^2}\right\} \tag{3.2}$$

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 (3.1) is true. □ More importantly, our analysis provides a unified global convergence proof for many non-monotone methods based on the method proposed in [21], which is one of the most used non-monotone line search algorithms. It corresponds to the modified Armijo rule

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

for a suitable choice of $m(k)$. Notice that this rule can be written in the form (2.2) with

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

We show how the convergence can be obtained as a simple corollary from the previous theorem. We generalize the stepsize rule to include also those that have been proposed in recent years (see, for example, [1, 2, 3, 29]).

COROLLARY 3.2. *Suppose that A1-A3 hold and let sequence* $\{x_k\}_{k=0}^{+\infty}$ *be generated by Algorithm 1 where (2.2) is replaced by*

$$f(x_k + \alpha_k\beta^{l_k}d_k) \leq R_k + \rho\alpha_k\beta^{l_k}\langle\nabla f(x_k), d_k\rangle \tag{3.3}$$

with

$$f(x_k) \leq R_k \leq \max_{0\leq j\leq m(k)} f(x_{k-j}) \tag{3.4}$$

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*

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

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

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

Proof. Under the compactness assumption on the level set, it was shown in [21] that

$$\lim_{k\to+\infty} f(x_k) = \lim_{k\to+\infty} \max_{0\leq j\leq m(k)} f(x_{k-j}). \tag{3.5}$$

Hence for $\nu_k = R_k - f(x_k)$ we obtain

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

Therefore, the result follows directly from Theorem 3.1. □

Remark 3.3. *It follows from Corollary 3.2 that any line search algorithm built upon the non-monotone rule (3.3) with* R_k *satisfying (3.4) is globally convergent.*

An improved lim-type convergence result can be obtained for variants of Algorithm 1 characterized by te following assumption:

A4' For all $k \geq 0$,

$$\nu_k = \theta_k \|\nabla f(x_k)\|^2,$$

with $\theta_k \geq 0$ and $\lim_{k\to+\infty} \theta_k = 0$.

Under assumption A4', the next lemma gives a finite upper bound of $\mathcal{O}(\epsilon^{-2})$ for the total number of iteration of Algorithm 1 in which $\|\nabla f(x_k)\| > \epsilon$ for a given $\epsilon > 0$.

Lemma 3.4. *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*

$$\Omega_\epsilon = \{k \mid \|\nabla f(x_k)\| > \epsilon\} \tag{3.6}$$

is bounded as follows

$$|\Omega_\epsilon| \leq k_1 + \left[\frac{2\left(f(x_0) - f_{low} + \sum_{i=0}^{k_1-1} \theta_i \|\nabla f(x_i)\|^2\right)}{\kappa_c}\right]\epsilon^{-2}, \tag{3.7}$$

where k_1 *is any positive integer such that* $\theta_k \leq \kappa_c/2$ *if* $k \geq k_1$, *for* κ_c *defined in (2.8).*

Proof. Since $\lim_{k\to+\infty} \theta_k = 0$ (by A4'), there exists $k_1 \in \mathbb{N}$ such that $\theta_k \leq \kappa_c/2$ if $k \geq k_1$. Thus, it follows from (2.12) that

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

which implies that

$$\frac{\kappa_c}{2}\|\nabla f(x_k)\|^2 \leq f(x_k) - f(x_{k+1}), \quad \forall k \geq k_1. \tag{3.8}$$

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

$$\Omega_{\epsilon}(s,t) = \left\{ s \leq k \leq t \,|\, \nabla f(x_k)\| > \epsilon \right\}. \tag{3.9}$$

For all $t > k_1$, it follows from (3.8) 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

$$|\Omega_{\epsilon}(k_1,t)| \leq \left[\frac{2\left(f(x_{k_1}) - f_{low}\right)}{\kappa_c}\right]\epsilon^{-2}. \tag{3.10}$$

Since $t > k_1$ is arbitrary, by (3.9), (3.10) and (3.6), we get

$$|\Omega_{\epsilon}| \leq k_1 + |\Omega_{\epsilon}(k_1,+\infty)| \leq k_1 + \left[\frac{2(f(x_{k_1}) - f_{low})}{\kappa_c}\right]\epsilon^{-2}. \tag{3.11}$$

Finally, notice that

$$f(x_{k_1}) \leq f(x_0) + \sum_{i=0}^{k_1-1} \nu_i = f(x_0) + \sum_{i=0}^{k_1-1} \theta_i \|\nabla f(x_i)\|^2. \tag{3.12}$$

Thus, (3.7) follows directly from (3.11) and (3.12). □

From the complexity estimate given in Lemma 3.4, we can establish the global convergence of Algorithm 1. The proof follows the same argument used to prove Corollary 2.1 in [24].

THEOREM 3.5. *Suppose that A1-A3 hold and let the sequence $\{x_k\}_{k=0}^{+\infty}$ be generated by Algorithm 1. If A4' also holds, then*

$$\lim_{k\to+\infty} \|\nabla f(x_k)\| = 0. \tag{3.13}$$

Proof. Suppose that (3.13) does not hold. Then, there exists $\epsilon > 0$ and a subsequence $\left\{x_{k_j}\right\}_{j=0}^{+\infty}$ 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 $\Omega_{\epsilon} = \{k \,|\, \|\nabla f(x_k)\| > \epsilon\}$ is infinity, contradictiong Lemma 3.4. □

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 [25, 22, 23] 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

$$x_{k+1} = \begin{cases} x_k^+, & \text{if } p_k \leq \min\left\{1, \exp\left(-\dfrac{f(x_k^+) - f(x_k)}{\tau_k}\right)\right\}, \\ x_k, & \text{otherwise,} \end{cases} \tag{4.1}$$

where $\tau_k > 0$ for all k, with $\tau_k \to 0$. By rule (4.1), if $f(x_k^+) \leq f(x_k)$ 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

$$\nu_{k,l} = M\exp\left(-\frac{max\left\{\theta, f(x_{k,l}^+) - f(x_k)\right\}}{\tau_k}\right) \tag{4.2}$$

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

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

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 4.1. *Suppose that A1-A3 hold and let sequence $\{x_k\}_{k=0}^{+\infty}$ be generated by Algorithm 1 with $\nu_{k,l}$ defined by (4.2). Given $\epsilon > 0$, if $\theta > 1$ and*

$$T \geq 2\max\left\{M\sum_{k=0}^{+\infty}\frac{1}{(k+1)^\theta}, f(x_0) - f_{low}\right\}\kappa_c^{-1}\epsilon^{-2}, \tag{4.3}$$

then

$$\min_{k=0,\ldots,T-1} \|\nabla f(x_k)\| \leq \epsilon. \tag{4.4}$$

Proof. By (4.2), for all k we have

$$\nu_k = Me^{-\max\{\theta, f(x_{k+1}) - f(x_k)\}\ln(k+1)} \leq M\left(\frac{1}{k+1}\right)^\theta. \tag{4.5}$$

Thus,

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

and Corollary 1 yields the result. □

THEOREM 4.2. *Suppose that A1-A3 hold and let sequence $\{x_k\}_{k=0}^{+\infty}$ be generated by Algorithm 1 with $\nu_{k,\ell}$ defined by (4.2). Given $\epsilon \in (0,1)$, if $\theta \in (0,1]$ and*

$$T \geq \max\left\{\left(\frac{4}{\kappa_c}\right)^{\frac{1+\theta}{\theta}} M^{\frac{1}{\theta}}, 1+\left(\frac{4M}{\kappa_c}\right)^{\frac{1}{\theta}}, \frac{2(f(x_0)-f_{low})}{\kappa_c}\right\}\epsilon^{-\frac{2(1+\theta)}{\theta}}, \tag{4.6}$$

then (4.4) holds.

Proof. By (4.5), we have $\nu_k \to 0$. Moreover, $\nu_k \leq M$ and given $\delta > 0$,

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

Denote

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

Taking $\delta = \kappa_c\epsilon^2/2$, it follows from (4.6) that

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

Thus, by Corollary 2.8, (4.4) must be true. □

REMARK 4.3. *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 4.2, the number of iterations that Algorithm 1 needs to find approximate stationary points may be significantly bigger in comparison to the case $\theta > 1$.*

5. Preliminary Numerical Experiments. We performed some numerical experiments comparing MATLAB implementations of four 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 "M".

(ii) the non-monotone algorithm in [34] 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 $k \geq 1$,

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

$Q_k = \eta_{k-1}Q_{k-1} + 1$ and $\eta_{k-1} = 0.85/k$, with $Q_0 = 1$. We shall refer to this code as "NM1".

(iii) the non-monotone algorithm in [21] obtained from Algorithm 1 by setting $\nu_{k,l} = \max_{0\leq j\leq m_k}[f(x_{k-j})] - f(x_k)$ 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 "NM2".

(iv) the non-monotone algorithm obtained from Algorithm 1 by setting $\nu_{k,l}$ as in (4.2), with $M = 50 + |f(x_0)|$ and $\theta = 1.01$. We shall refer to this code as "NM3".

In all implementations, we consider the parameters $\alpha_0 = 1$ and $\beta = \rho = 0.5$. The search directions were generated as $d_k = -\lambda_k \nabla f(x_k)$, where

$$\lambda_{k+1} = \begin{cases} \max\left\{\lambda_{min}, \min\left\{\frac{s_k^T s_k}{s_k^T y_k}, \lambda_{max}\right\}\right\}, & \text{if } s_k^T y_k > 0, \\ \lambda_{max}, & \text{otherwise,} \end{cases}$$

with $\lambda_0 = 1$, $s_k = x_{k+1} - x_k$, $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, $\lambda_{min} = 10^{-30}$ and $\lambda_{max} = 10^{30}$. We report here some results obtained for the two-dimensional Griewank function [20]:

$$f(x) = 1 + \frac{x_1^2}{4000} + \frac{x_2^2}{4000} - cos(x_1)cos(x_2/\sqrt{2}). \tag{5.1}$$

This function has a huge number of local minimizers but only one global minimizer, namely $x^* = (0,0)$ with $f(x^*) = 0$. We applied all the codes to minimize (5.1) considering 60 initial points generated in the box $[-600, 600] \times [-600, 600]$:

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

Within a budget of 500 function evaluations, code M found the best function value for 3, 33% of the starting points, while codes NM1, NM2 and NM3 found the best function value for 13, 33%, 20, 00% and 63, 33% of the starting points, respectively. The distribution of the best function values found by each code is summarized in Figure 5.1.

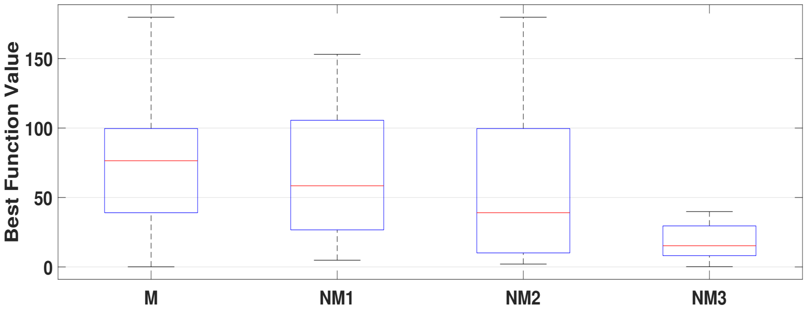

Fig. 5.1. *Box-plot of the best objective function values obtained by the four codes.*

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 NM3) 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 [31] 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 [17], 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 $\nu_k \rightarrow 0$, 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].

REFERENCES

[1] Ahookhosh, M., Amini, K., Bohrami, 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. Applied Mathematical Modelling **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.: Theoretical study of an adaptive cubic regularization method with dynamic inexact Hessian information. arXiv: 1808.06239 (2018)

[5] Bellavia, S., Krejić, N., Jerinkić, N.K.: Subsampled inexact Newton methods for minimizing large sums of convex functions. arXiv:1811.05730 (2018)

[6] Bellavia, S., Jerinkić, N.K., Malaspina, G.: Subsampled Nonmonotone Spectral Gradient Methods. arXiv:1812.06822 (2019)

[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, Ph.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 Journal on Optimization **27**, 1049–1074 (2017)

[10] Cartis, C., Gould, N.I.M., Toint, Ph.L.: On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. SIAM Journal on Optimization **20**, 2833-2852 (2010)

[11] Cartis, C., Gould, N.I.M., Toint, Ph.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, Ph.R., Toint, Ph.L.: Worst-case evaluation complexity of first-order nonmonotone gradient-related algorithms for unconstrained optimization. Optimization **64**, 1349–1361 (2015)

[13] 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. Mathematical Programming **162**, 1–32 (2017)

[14] Dussault, J-P.: ARCq: a new Adaptive Regularization by Cubics variant. Optimization Methods and Software **33**, 322–335 (2018)

[15] 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)

[16] Grapiglia, G.N., Yuan, J., Yuan, Y.: Nonlinear Stepsize Control Algorithms: Complexity Bounds for First-and Second-Order Optimality. Journal of Optimization Theory and Applications **171**, 980-997 (2016)

[17] Grapiglia, G.N., Sachs, E.W.: On the worst-case evaluation complexity of non-monotone line search algorithms. Computational Optimization and Applications **68**, 555-577 (2017)

[18] Grapiglia, G.N., Nesterov, Yu.: Regularized Newton Methods for Minimizing Functions with Hölder Continuous Hessians. SIAM J. Optim. **27**, 478-506 (2017)

[19] Gratton, S., Sartenaer, A., Toint, Ph.L.: Recursive trust-region methods for multiscale nonlinear optimization. SIAM J. Optim. **19**, 414–444 (2008)

[20] Griewank, A.O.: Generalized Descent for Global Optimization. Journal of Optimization Theory and Applications **34**, 11-39 (1981)

[21] Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newtons method, SIAM Journal on Numerical Analysis **23**, 707-716 (1986)

[22] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)

[23] Locatelli, M.: Simulated Annealing Algorithms for Continuous Global Optimization: Convergence Conditions. Journal of Optimization Theory and Applications **104**, 121–133 (2000)

[24] Martínez, J.M., Raydan, M.: Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. Journal of Global Optimization **68**, 367–385 (2017)

[25] Metropolis, N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H.: Equation of state calculations by fast computer machines. Journal of Chemical Physics **21**, 1087–1092 (1953)

[26] Mo, J., Liu, C., Yan, S.: A nonmonotone trust-region method based on nonincreasing technique of weighted average of the sucessive function value. Journal of Computational and Applied Mathematics **209**, 97–108 (2007)

[27] Nesterov, Yu.: Introductory lectures on convex optimization: A basic course. Kluwer, Academic Publishers, Dordrecht, The Netherlands (2004)

[28] Nesterov, Yu., Polyak, B.T.: Cubic regularization of Newton method and its global performance. Mathematical Programming **108**, 177-205 (2006)

[29] Nosratipour, H., Borzabadi, A.H., Fard, O.S.: On the nonmonotonicity degree of nonmonotone line searches. Calcolo **54**, 1217-1242 (2017)

[30] Royer, C.W., Wright, S.J.: Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. SIAM Journal on Optimization **28**, 1448-1477 (2018)

[31] Sachs, E.W., Sachs, S.M.: Nonmonotone line searches for optimization algorithms. Control and Cybernetics **40**, 1059–1075 (2011)

[32] Sun, W., Yuan, Y.: Optimization Theory and Methods: Nonlinear Programming. Springer (2006)

[33] Xu, P., Roosta-Khorasani, F., Mahoney, M.W.: Newton-type methods for non-convex optimization under inexact Hessian information. arXiv: 1708.07164v2 (2017)

[34] Zhang, H.C., Hager, W.W.: A nonmonotone line search technique for unconstrained optimization, SIAM journal on Optimization **14**, 1043-1056 (2004)