An adaptive high order method for finding third-order critical points of nonconvex optimization

Xihua Zhu 1 · Jiangze Han 2 · Bo Jiang 3

Received: 29 January 2021 / Accepted: 20 February 2022 / Published online: 16 March 2022 © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Recently, the optimization methods for computing higher-order critical points of nonconvex problems attract growing research interest (Anandkumar Conference on Learning Theory 81-102, 2016), (Cartis Found Comput Math 18:1073-1107, 2018), (Cartis SIAM J Optim 30:513-541, 2020), (Chen Math Program 187:47-78, 2021), as they are able to exclude the so-called degenerate saddle points and reach a solution with better quality. Despite theoretical developments in (Anandkumar Conference on Learning Theory 81-102, 2016), (Cartis Found Comput Math 18:1073-1107, 2018), (Cartis SIAM J Optim 30:513-541, 2020), (Chen Math Program 187:47-78, 2021), the corresponding numerical experiments are missing. This paper proposes an implementable higher-order method, named adaptive high order method (AHOM), to find the third-order critical points. AHOM is achieved by solving an “easier” subproblem and incorporating the adaptive strategy of parameter-tuning in each iteration of the algorithm. The iteration complexity of the proposed method is established. Some preliminary numerical results are provided to show that AHOM can escape from the degenerate saddle points, where the second-order method could possibly get stuck.

Keywords Continuous optimization · Nonconvex optimization · Adaptive algorithm · Higher order method · Third-order critical points

Research supported by NSFC Grants 72171141, NSFC Grants 71971132, NSFC Grants 72150001, NSFC Grants 11831002, GIFSUFE Grants CXJJ-2019-391, and Program for Innovative Research Team of Shanghai University of Finance and Economics.

Bo Jiang
isyebojiang@gmail.com

Xihua Zhu
zhuxihua@163.sufe.edu.cn

Jiangze Han
jiangze.han@sauder.ubc.ca

1 School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, People’s Republic of China

2 UBC Sauder School of Business, the University of British Columbia, Vancouver, BC V6T 1Z2, Canada

3 Research Institute for Interdisciplinary Sciences, School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, People’s Republic of China
1 Introduction

In this paper, we consider the following unconstrained optimization problem

\[ f^* := \min_{x \in \mathbb{R}^n} f(x), \tag{1} \]

where \( f \) is nonconvex and \( p \)-times differentiable. In recent years there has been a surge of research interest in nonconvex optimization (see, for instance, \([2,4,7,8,10–16,18–20,26,29,30,33,34,44]\)). However, it is well known that optimizing a nonconvex problem is a notoriously challenging task. Even a less challenging work of finding a local optimum is computationally hard in the worst case \([42]\), and it is even NP-hard to check whether a critical point is a local minimizer \([37]\). Therefore, this paper focuses on optimization methods of finding some “good” critical points. In fact, the concept of critical point can be divided into a few subcategories. Classical gradient descent type method may be stuck at a first-order critical point, i.e., \( \nabla f(x) = 0 \). While algorithms incorporating second-order differentiable information \([41]\) may converge to a second-order critical point, i.e., \( \nabla f(x) = 0 \) and \( \nabla^2 f(x) \succeq 0 \), which could exclude some first-order critical points that are not local optima. However, it is still possible that the second-order method could get stuck at the so-called degenerate saddle point (Hessian matrix has nonnegative eigenvalues with some eigenvalues equal to 0). To see this, let us consider problem (1) with two concrete objective functions:

\( (i) \) monkey problem: \( f(x) = x_0^3 - 3x_0x_1^2; \)

\( (ii) \) nonconvex coercive function: \( f(x) = \frac{1}{3}x_0^3 + \frac{1}{4}x_1^4 - \frac{1}{2}x_1^2. \)

There are two degenerate saddle points \((0, 0)\) and \((0, 1)\) in these two problems, respectively. As shown in Fig. 1, the gradient descent method (GD) and the adaptive cubic regularization of Newton’s method (ARC) will get stuck at these two saddle points after a few iterations by selecting \((1, 0)\) and \((3, 3)\) as the initial points for the two problems, respectively. In order to get away from the degenerate saddle points, the notion of a higher-order critical point was proposed in \([2,14]\), and the corresponding optimization algorithms were designed as well to find such higher-order critical points. We implement these ideas to solve the monkey problem and minimize the nonconvex coercive function. We set the same initial points as before. As shown in Fig. 1, our proposed algorithm (AHOM) could escape from the aforementioned two degenerate saddle points.

Prior to our work, there are several papers \([2,14,16,18,33]\) concerning optimization methods for computing higher-order critical points. In particular, Anandkumar and Ge \([2]\) proposed a third-order algorithm that utilizes third-order derivative information and converges to a third-order critical point, i.e., it is a second-order critical point and satisfies the third-order condition additionally (see (11) for its definition). Lucchi et al. \([33]\) proposed a stochastic third-order algorithm that uses sub-sampled derivatives rather than the exact ones, and they showed that the proposed algorithm also converges to a third-order critical point. Cartis et al. \([14]\) presented a trust-region method with the \( p \)-th order derivative for convexly constrained problems, and it computes an \( \epsilon \)-approximate \( q \)-th \( (q \geq 2) \) order critical point within at most \( O(\epsilon^{-(q+1)}) \) iterations. Later on, such iteration bound was improved to \( O\left(\epsilon^{-\frac{q+1}{p-q+1}}\right)\) in \([16,18]\). Despite those theoretical developments, the corresponding numerical experiments
are absent, and the practical issue regarding the implementation of their algorithms remains to be addressed. Specifically, a nonconvex subproblem, which is NP-hard in general, needs to be globally solved in each iteration of the algorithms in [14,16,18,33]. Moreover, there is no remedy provided in Cartis et al. [14,16,18] or Lucchi et al. [33] regarding the subproblem solving. In contrast, we propose a subroutine that can solve our subproblem approximately and such approximation does not affect the convergence of the whole algorithm. Anandkumar and Ge’s [2] method assumes the knowledge of problem parameters such as the Lipschitz constants of the second-order and the third-order derivatives, which are hard to estimate in practice. As a matter of fact, an algorithm that does not depend on problem parameters is often desirable in optimization. Therefore, various adaptive strategies [4,8,12,13,16–19,22,31,33,44] have been adopted to adjust the parameters in the process of iteration. In this paper, we propose an adaptive high order method (AHOM) for problem (1), which incorporates Anandkumar and Ge’s approach [2] by some adaptive strategies. In particular, we use a single iteration of the adaptive regularized \( p \)-th order method (ARp) [8,15] in each step of AHOM to tune the high-order regularization term adaptively and introduce a new criterion to estimate the third-order Lipschitz constant dynamically. It turns out our AHOM can solve some nonconvex machine learning problems and escape from the degenerate saddle points (see Sect. 5 for details).

Another merit of introducing high-order derivative information to optimization algorithms is the associated iteration complexity bounds could be improved. A few recent papers [3,9,23,27,28,32,39,40] indicate that high-order derivatives indeed accelerate classical algorithms in the context of convex optimization. A similar phenomenon was also observed in nonconvex optimization. For the unconstrained case, in contrast with the evaluation complexity of \( O(\epsilon^{-2}) \) in the first-order method [38], Nesterov [41] showed that the cubic regularization of Newton method could find an \( \epsilon \)-approximate first-order critical point with at most \( O(\epsilon^{-3/2}) \) evaluations of the objective function (and its derivatives). By using up to \( p \)-th (\( p \geq 1 \)) order derivatives, Birgin et al. [8] first proposed ARp method, whose evaluation complexity of finding first-order critical points is improved to \( O(\epsilon^{-(p+1)/p}) \). Later on, Birgin et al. [6] further proposed a reliable algorithm with some numerical experiments for the case \( p = 3 \). Moreover, Cartis et al. [15] managed to adapt the ARp method to reach second-order critical points. In the meanwhile, the high-order method was also extended to accommodate nonconvex optimization with constraints [7,16,34] and non-Lipschitz nonconvex optimization [18,19]. Since our AHOM algorithm also belongs to the category of high-order methods, we show that its iteration bound improves that of the algorithm in [2] when \( p \geq 3 \). It is worth
mentioning that as we perform adaptations on both the high-order regularization term and estimator of the third-order Lipschitz constant, the corresponding iteration analysis becomes more technically involved than that in [2].

The rest of the paper is organized as follows. In Sect. 2, we introduce some preliminaries and the assumptions used throughout this paper. In Sect. 3, we propose our adaptive high order method (AHOM) for problem (1). Section 4 is devoted to analyzing the iteration bound of AHOM. In Sect. 5, we present some preliminary numerical results on solving some nonconvex polynomial optimization problems and $\ell_2$-regularized nonconvex logistic regression problems, where AHOM is able to escape from degenerate saddle points and even occasionally converge to a point satisfying second-order sufficient condition.

2 Preliminaries

In this section, we introduce notations, various approximate critical measures and present some basic assumptions that will be used in the paper.

2.1 Notations

Recall that a high-order tensor is a multidimensional array. In particular, first-order and second-order tensors are vectors and matrices, respectively. Throughout, we use lower-case letters to denote vectors (e.g., $v \in \mathbb{R}^n$), capital letters to denote matrices (e.g., $M \in \mathbb{R}^{n \times n}$), and capital calligraphy letters to denote high-order tensors (e.g., $T \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_p}$), with subscripts of indices being their entries (e.g., $v_i$, $M_{ij}$, $T_{j_1,j_2,\ldots,j_p}$). For a $p$-times differentiable function $f$, the associated $p$-th order derivative tensor is given by

$$\nabla^p f(x) = \left[ \frac{\partial^p f(x)}{\partial x_{i_1} \cdots \partial x_{i_p}} \right]_{i_j \in [n], \forall j},$$  

where $[n]$ denotes $\{1, \ldots, n\}$.

The operations between tensor $T$ and vectors $v_1, \ldots, v_p$ yields a multi-linear form

$$T(v_1, \ldots, v_p) = \sum_{i_1,\ldots,i_p} T_{i_1\cdots i_p}[v_1]_{i_1} \cdots [v_p]_{i_p}.$$  

We say a tensor is symmetric if $T_{j_1,j_2,\ldots,j_p} = T_{\pi(j_1,j_2,\ldots,j_p)}$ for any permutation $\pi$ of the indices $(j_1, j_2, \ldots, j_p)$. When $T$ is a symmetric tensor, we may make $v_1, \ldots, v_p$ identical to the same vector $v$ in the above multi-linear form, yielding that

$$T(v)^p = \sum_{i_1,\ldots,i_p} T_{i_1\cdots i_p}v_1 \cdots v_p.$$  

Similarly, the multi-linear form with respect to matrices $U_1 \in \mathbb{R}^{n \times n_1}, \ldots, U_p \in \mathbb{R}^{n \times n_p}$ is defined as

$$[T(U_1, \ldots, U_p)]_{i_1,i_2,\ldots,i_p} = \sum_{j_1,j_2,\ldots,j_p \in [n]} T_{j_1,j_2,\ldots,j_p}[U_1]_{j_1,i_1} \cdots [U_p]_{j_p,i_p}.$$  

\copyright Springer
where $\mathcal{T}(U_1, \cdots, U_p)$ itself is a $p$-th order tensor with $n_j$ being the dimension of $j$-th direction. Suppose $S$ is the projection matrix associated with subspace $S$. We call

$$\text{Proj}_S \mathcal{T} \overset{\text{def}}{=} \mathcal{T}(S, \cdots, S)$$

the projection tensor of $T$ on subspace $S$. Note that any $p$ vectors $v_1, \cdots, v_p$ applied to the projection tensor is equivalent to the projections of $v_1, \cdots, v_p$ on $S$ applied to the original tensor:

$$[\mathcal{T}(S, \cdots, S)](v_1, \cdots v_p) = \mathcal{T}(Sv_1, \cdots, Sv_p).$$

The Frobenius norm of a $p$-th order tensor $T$ is:

$$\|T\|_F = \sqrt{\sum_{j_1, j_2, \cdots, j_p \in [n]} T^2_{j_1, j_2, \cdots, j_p}},$$

and the spectral norm of a $p$-th order tensor is defined as

$$\|T\|_p = \max_{\|v_1\| = \cdots = \|v_p\| = 1} |\mathcal{T}(v_1, \cdots, v_p)|.$$ (4)

For a symmetric tensor $T$, the spectral norm in (4) is equivalent to $\|T\|_p = \max_{\|v\| = 1} |\mathcal{T}(v, \cdots, v)|$. In particular, the spectral norm of a symmetric matrix $M \in \mathbb{R}^{n \times n}$ is equivalent to $\|M\|_2 = \max\{|\lambda_1(M)|, \cdots, |\lambda_n(M)|\}$, where $\lambda_i(M)$ denotes the $i$-th largest eigenvalue of $M$. Note that all the matrices considered in this paper are symmetric.

### 2.2 Lipschitz continuous assumption

We assume that the $p$-th order derivative (2) is globally Lipschitz continuous, i.e., there exits $L_p \geq 0$ such that

$$\|\nabla^p f(x) - \nabla^p f(y)\|_p \leq L_p \|x - y\|, \quad \text{for all } x, y \in \mathbb{R}^n,$$ (5)

where the $\|\|_p$ is the tensor spectral norm of $p$-th order tensor given by (4). In the rest of the paper, we let $L \overset{\text{def}}{=} \max \left\{ \frac{L_k}{(k-1)!}, \ k = 1, 2, \cdots, p \right\}$.

With tensor notations, the Taylor expansion of function $f(\cdot)$ at $x \in \mathbb{R}^n$ can be written as:

$$T_p(x, s) = f(x) + \sum_{j=1}^{p} \frac{1}{j!} \nabla^j f(x)(s)^j.$$ (6)

Then, there is a bound between $f(y)$ and its Taylor expansion:

$$|f(x+s) - T_p(x, s)| \leq \frac{L_p}{(p+1)!} \|s\|(p+1).$$ (7)

In our later analysis, we shall need the special case of $p = 3$:

$$\left| f(x+s) - f(x) - (\nabla f(x), s) - \frac{1}{2} s^T \nabla^2 f(x)s - \frac{1}{6} \nabla^3 f(x)(s, s, s) \right| \leq \frac{L_3}{24} \|s\|^4.$$ (8)
2.3 Approximate critical points

We define the first-order and second-order critical measures of problem (1) as

\[ \chi_{f,1}(x) \overset{\text{def}}{=} \| \nabla f(x) \| \]

and

\[ \chi_{f,2}(x) \overset{\text{def}}{=} \max \{ 0, -\lambda_n(\nabla^2 f(x)) \} \]

respectively, where \( \lambda_n(\nabla^2 f(x)) \) is the smallest eigenvalue of Hessian matrix \( \nabla^2 f(x) \). Then, a point \( x \) satisfying \( \chi_{f,1}(x) \leq \epsilon_1 \) is an \( \epsilon_1 \)-approximate first-order critical point, and we call it an \( (\epsilon_1, \epsilon_2) \)-approximate second-order critical point if it further satisfies \( \chi_{f,2}(x) \leq \epsilon_2 \).

Recall it was demonstrated in [2] that \( x \) is a third-order critical point if it is a second-order critical point and

\[ \nabla^3 f(x)(u, u, u) = 0 \]

holds for any \( u \) that satisfies \( u^\top \nabla^2 f(x) u = 0 \). (11)

Following the idea in [2], we consider the eigen-subspace of the Hessian matrix below.

**Definition 1** For any symmetric matrix \( M \) with an eigen-decomposition \( M = \sum_{i=1}^n \lambda_i v_i v_i^\top \), we adopt \( S_x(M) \) to denote the span of eigenvectors with eigenvalue at most \( \tau \). That is

\[ S_x(M) = \text{span}\{ v_i | \lambda_i \leq \tau \} . \]

Now we are able to define the third-order critical measure of the objective function.

**Definition 2** ((\( \beta, \kappa \))-competitive subspace and third-order critical measure) Given any \( \beta > 0 \) and \( \kappa > 0 \), let \( (\beta, \kappa) \)-competitive subspace \( S^x \) at point \( x \) be the largest eigen-subspace \( S_x(\nabla^2 f(x)) \) such that \( \tau \leq \frac{\chi_{f,3}(x)^2}{12 \beta^2} \), where

\[ \chi_{f,3}(x) = \| \text{Proj}_{S^x} \nabla^3 f(x) \|_F \]

is the norm of the third-order derivatives projected in this subspace. We call \( \chi_{f,3}(x) \) is the third-order critical measure of \( f \).

Note that our notation above is slightly different from that proposed by Anandkumar and Ge [2], where the adaptive estimator \( \kappa \) is fixed as \( L_3 \). In contrast, we consider the \( (\tau, \kappa) \)-competitive subspace and third-order critical measure to exclude the dependence on the third-order Lipschitz parameter \( L_3 \). In fact, the reason to let \( \chi_{f,3}(x) = \| \text{Proj}_{S^x} \nabla^3 f(x) \|_F \) as a third-order critical measure is that condition (11) is implied by \( \| \text{Proj}_{S^x} \nabla^3 f(x) \|_F = 0 \).

To see this, suppose \( \| \text{Proj}_{S^x} \nabla^3 f(x) \|_F = 0 \). We observe that

\[ \text{span}\{ u | u^\top \nabla^2 f(x) u = 0 \} = S_0(\nabla^2 f(x)) \subseteq S_x(\nabla^2 f(x)) \] for any \( \tau > 0 \)

according to Definition 1. Then, for any \( u \in S_0(\nabla^2 f(x)) \subseteq S_x(\nabla^2 f(x)) \), we have \( u \in S^x \), that is \( S^x u = u \), where \( S^x \) is the projection matrix associated with the subspace \( S^x \). Combining this fact with \( \| \text{Proj}_{S^x} \nabla^3 f(x) \|_F = 0 \), we conclude that for any \( u \in S_0(\nabla^2 f(x)) \subseteq S^x \)

\[ \nabla^3 f(x)(u, u, u) = \nabla^3 f(x)(S^x u, S^x u, S^x u) = 0, \]

which is exactly the condition (11).

With the above consideration, we define the \( (\epsilon_1, \epsilon_2, \epsilon_3) \)-approximate critical point as follows.
Definition 3 We call \( x \in \mathbb{R}^n \) an \((\epsilon_1, \epsilon_2, \epsilon_3)\)-approximate critical point of problem (1) if it satisfies
\[
(i) \; \chi_{f,1}(x) \leq \epsilon_1, \quad (ii) \; \chi_{f,2}(x) \leq \epsilon_2, \quad (iii) \; \chi_{f,3}(x) \leq \epsilon_3.
\]

We end this section by presenting an algorithm named ACCS that can find a \((\tau, \kappa)\)-competitive subspace efficiently.

Algorithm 1 ACCS (Algorithm for computing the \((\tau, \kappa)\)-competitive subspace)

**Input**: Hessian matrix \( M = \nabla^2 f(z) \), third order derivative \( T = \nabla^3 f(z) \), approximation ratio \( \beta \), adaptive parameter \( \kappa \).

**Output**: Competitive subspace \( S \) and \( \chi_{f,3}(z) \).

Perform the eigen-decomposition of \( M = \sum_{i=1}^{n} \lambda_i v_i v_i^\top \). [\( \lambda_i \) is \( i \)-th largest eigenvalue of \( M \)]

for \( i = 1 \) to \( n \) do
  Let \( S = \text{span}\{v_i, v_{i+1}, \ldots, v_n\} \).
  Let \( \chi_{f,3}(z) = \|\text{Proj}_S T\|_F \)
  if \( \frac{\chi_{f,3}(z)^2}{12\kappa \beta^2} \geq \lambda_i \) then
    terminate and return: \( S \) and \( \chi_{f,3}(z) \).
  end if
end for

return \( S = \emptyset, \; \chi_{f,3}(z) = 0 \).

3 Adaptive high order method (AHOM)

This section aims to design an adaptive high order method (AHOM) that can find a third-order critical point.

3.1 The single iteration of adaptive regularized \( p \)-th order method (SARp)

Before introducing the AHOM algorithm, we first present a subroutine (SARp) in Algorithm 2 that is invoked in every iteration of AHOM. SARp is a single iteration of adaptive regularized \( p \)-th order method (ARp) in [15] and requires an approximate minimization of
\[
m(x_k, s, \sigma_k) \overset{\text{def}}{=} T_p(x_k, s) + \frac{\sigma_k}{p+1} \|s\|^{p+1},
\]
where \( \sigma_k \) is the adaptive coefficient of the \((p+1)\)-th order regularization term.

Algorithm 2 Single iteration of ARp (SARp)

**Input**: Objective function \( f \), last iterate \( x_k \), regularization parameter \( \sigma_k \).

**Output**: Generated point \( z_k \) and next regularization parameter \( \sigma_{k+1} \).

Step 0: Initialization. Give the constants \( \theta, \eta_1, \eta_2, \gamma_1, \gamma_2, \gamma_3, \sigma_{\text{min}} \) are also given and satisfy
\[
\theta > 0, \; \sigma_{\text{min}} \in (0, \sigma_0], \; 0 < \eta_1 \leq \eta_2 < 1, \; 0 < \gamma_1 < 1 < \gamma_2 < \gamma_3.
\]

Compute \( f(x_k) \).
Step 1: Step calculation. Compute the step $s_k$ by approximately minimizing the model $m(x_k, s, \sigma_k)$ with respect to the $s$ satisfying the following conditions
\[
m(x_k, s_k, \sigma_k) < m(x_k, 0, \sigma_k) \\
\chi_{m,i}(x_k, s_k, \sigma_k) \leq \theta \|s_k\|^{(p+1-i)}, \quad (i = 1, 2).
\]

Step 2: Acceptance of the trial point. Compute $f(x_k + s_k)$ and define
\[
\rho_k = f(x_k + s_k) - f(x_k - s_k).
\]
If $\rho_k \geq \eta_1$, then let $z_k = x_k + s_k$; otherwise $z_k = x_k$.

Step 3: Regularization parameter update. Set
\[
\sigma_{k+1} \in \begin{cases}
\max\{\sigma_{\min}, \gamma_1 \sigma_k\}, \sigma_k & \text{if } \rho \geq \eta_2, \\
[\sigma_k, \gamma_2 \sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\
[\gamma_2 \sigma_k, \gamma_3 \sigma_k] & \text{if } \rho_k < \eta_1.
\end{cases}
\]

Return point $z_k$ and regularization parameter $\sigma_{k+1}$.

We remark that the conditions in Step 1 of SARp are easily achievable by applying some existing algorithms like the ARC method [12,13]. As SARp is a single step of ARp, many useful properties of ARp can be carried over to SARp, which are summarized in the following lemma.

Lemma 1 ([15], Lemma 3.1, Lemma 3.3, Lemma 3.4) Given $x_k$, the mechanism of SARp guarantees the following properties of the approximate minimizer $s_k$ of $m(x_k, s, \sigma_k)$.

(i) $T_p(x_k, 0) - T_p(x_k, s_k) \geq \frac{\sigma_k}{p+1} \|s_k\|^{p+1}$,

(ii) $\|s_k\| \geq \left(\frac{\chi_{f,1}(x_k + s_k)}{L + \theta + \sigma_k}\right)^{\frac{1}{p}}$,

(iii) $\|s_k\| \geq \left(\frac{\chi_{f,2}(x_k + s_k)}{(p-1)L + \theta + p\sigma_k}\right)^{\frac{1}{p-1}}$.

With the lemma above, we are able to prove some bounds for the critical measures and the sufficient decrease on the objective function in terms of the distance between $z_k$ and $x_k$.

Proposition 1 Suppose that $(z_k, \sigma_{k+1}) = \text{SARp}(f, x_k, \sigma_k)$, then for all successful SARp ($\rho_k \geq \eta_1$), there have

(i) $\chi_{f,1}(z_k) \leq (L + \theta + \sigma_k)\|z_k - x_k\|^p$,

(ii) $\chi_{f,2}(z_k) \leq ((p-1)L + \theta + p\sigma_k)\|z_k - x_k\|^{p-1}$,

(iii) $f(z_k) \leq f(x_k) - \eta_1 \sigma_{\min} \frac{\sigma_k}{p+1} \|z_k - x_k\|^{p+1}$,

where $\eta_1$ and $\sigma_{\min}$ are defined in SARp algorithm.

Proof If SARp is successful, we have $z_k = x_k + s_k$. Moreover, we have (i) and (ii) hold, which are just reformulations of (ii) and (iii) in Lemma 1. To prove (iii), we note that
\[ \rho_k = \frac{f(x_k) - f(z_k)}{T_p(x_k, 0) - T_p(x_k, s_k)} \geq \eta_1 \]

in successful SARp, which combined with (i) in Lemma 1 yields that

\[ f(x_k) - f(z_k) \geq \eta_1 (T_p(x_k, 0) - T_p(x_k, s_k)) \geq \frac{\eta_1 \sigma_{\text{min}}}{p + 1} \| z_k - x_k \|^{p+1}. \]

\[ \square \]

### 3.2 The AHOM algorithm

Now we are ready to present our AHOM algorithm in Algorithm 3.

**Algorithm 3** Adaptive High Order Method (AHOM)

**Input**: An initial point \( x_0 \), objective function \( f \), accuracy levels \( \epsilon_1, \epsilon_2 \) and \( \epsilon_3 \) of critical measures.

**Output**: Solution \( x_{\epsilon_3} \) that satisfies third-order critical measure.

**Initialization**. Set regularization parameters \( \sigma_0 > 0, \kappa_0 > 0 \), and constants \( 0 < \xi_1 < 1, \xi > 1, \beta > 0 \).

**for** \( k = 0, 1, 2, \ldots \)

**Step 1: Step calculation.**
1. Find \( z_k \) that can decrease the objective function value by computing \( (z_k, \sigma_{k+1}) = \text{SARp}(f, x_k, \sigma_k) \).
2. Search for competitive subspace \( S^{z_k} \) and third-order tensor norm \( \chi_{f,3}(z_k) \) by computing \( (S^{z_k}, \chi_{f,3}(z_k)) = \text{ACC}(\nabla^2 f(z_k), \nabla^3 f(z_k), \beta, \kappa_k) \).
3. **Test for termination.** Evaluate \( \chi_{f,i}(z_k) \),
   - if \( \chi_{f,i}(z_k) \leq \epsilon_i \), for \( i = 1, 2, 3 \),
   - **terminate** with a solution \( x_\epsilon = x_{k+1} \).
4. If \( \chi_{f,3}(z_k) \geq \beta (24 \cdot \chi_{f,1}(z_k) \cdot \kappa_k^3)^{1/3} \), go to Step 2,
   - else let \( x_{k+1} = z_k \) and go to Step 3.

**Step 2: Acceptance of the trial point.**
Compute \( u = \text{ATN}(\nabla^3 f(z_k), S^{z_k}, \beta) \) such that

\[ \nabla^3 f(z_k)(u, u, u) \geq \frac{\chi_{f,3}(z_k)}{\beta}, \]

where \( \text{ATN} \) is described in Algorithm 4. Let \( \Delta_k = \frac{\chi_{f,3}(z_k)^4}{24 \beta^4 \kappa_k^4} \), compute \( f(z_k - \frac{\chi_{f,3}(z_k)}{\beta \kappa_k} u) \) and define

\[ \Phi_k = \frac{f(z_k) - f(z_k - \frac{\chi_{f,3}(z_k)}{\beta \kappa_k} u)}{\Delta_k}. \]  

(13)

If \( \Phi_k \geq \xi_1 \), let \( x_{k+1} = z_k - \frac{\chi_{f,3}(z_k)}{\beta \kappa_k} u \); otherwise let \( x_{k+1} = z_k \).

**Step 3: Regularization parameter update.** Set
\[ \kappa_{k+1} = \begin{cases} \zeta \kappa_k, & \text{if } \chi_{f,3}(z_k) \geq \beta (24 \cdot \chi_{f,1}(z_k) \cdot \kappa_k^2)^{1/3} \text{ and } \Phi_k < \xi_1, \\ \kappa_k, & \text{otherwise.} \end{cases} \]

End for

AHOM algorithm takes the SARp as a subroutine, and further utilizes the first-order, second-order and third-order derivatives to make progress. AHOM stops when all the three critical measures are sufficiently small, i.e., \( \chi_{f,i}(z_k) \leq \epsilon_i \), for some given \( \epsilon_i \) with \( i = 1, 2, 3 \). The decrease of the first two order critical measures is achieved by iteratively performing SARp in Step 1. When the third-order critical measure \( \chi_{f,3} \) on the trial point \( z_k \) is large, a descent direction \( u \) is constructed. Then a nontrivial update will be performed if the sufficient relative decrease on the objective (i.e., \( \Phi_k \geq \xi_1 \)) further occurs. On the other hand, a step resulting in an insufficient relative decrease is rejected by the algorithm, and the adaptive estimator \( \kappa \) is increased by a factor of \( \zeta \). It is also possible that the third-order critical measure \( \chi_{f,3} \) is already below the given tolerance, but either \( \chi_{f,1} \) or \( \chi_{f,2} \) is still large. In this case, Step 2 is skipped and \( x_{k+1} = z_k \). Furthermore, if \( z_k \) is obtained by an unsuccessful SARp, \( x_{k+1} \) is actually equal to \( x_k \) (that is, \( x_{k+1} \) is not updated). However, the cubic regularizer \( \sigma_{k+1} \) is updated in this case, which may lead to an update on the next trial point \( z_{k+1} \).

Finally, we would like to mention that algorithm ATN in Step 2 was proposed in [2] with the detailed descriptions given as follows:

**Algorithm 4** Approximate Tensor Norms (ATN)

**Input**: Tensor \( T \), subspace \( S \), constant \( \beta \).

**Output**: Unit vector \( u \in S \) such that \( T(u, u, u) \geq \| \text{Proj}_S T \|_F / \beta \).

repeat

Let \( u \) be a random standard Gaussian in subspace \( S \).

until \( |T(u, u, u)| \geq \| \text{Proj}_S T \|_F / \beta \).

return \( u \) if \( T(u, u, u) > 0 \) and \(-u\) otherwise.

This algorithm aims to find a vector \( u \in S \) such that the value \( T(u, u, u) \) is an approximation of \( \| \text{Proj}_S T \|_F \). The following result reveals that it converges by at most 2 iterations in expectation.

**Lemma 2** ([2] Theorem 7) There is a universal constant \( B \) such that the expected number of iterations of Algorithm ATN is at most 2, and the output of ATN is a unit vector \( u \) that satisfies \( T(u, u, u) \geq \| \text{Proj}_S T \|_F / \beta \) for \( \beta = Bn^{1.5} \).

### 4 Iteration complexity analysis of AHOM

To provide the iteration bound for AHOM, like in [8,15], we also need to define some “successful” iterations. Since there are two regularization parameters: \( \sigma \) and \( \kappa \), whether they are updated successfully or not defines two types of “successful” iterations accordingly. We first consider the cubic regularizer \( \sigma \) in SARp.

**Definition 4** We say an iteration in AHOM is “successful SARp” if the SARp called in Step 1 of this iteration is successful (i.e., \( \rho_j \geq \eta_1 \)); otherwise, it is an “unsuccessful SARp” iteration. Suppose \( T \) is the total number of iterations in AHOM, we denote by

\[ (i) \quad S_{\text{SARp}} = \{0 \leq j \leq T - 1 \mid \rho_j \geq \eta_1\} \]
the index set of all iterations such that the associated trial point \( z_k \) is successful in SARp, and the complementary set including all the “unsuccessful SARp” iterations is denoted as

\[(ii) \mathcal{U}_{\text{SARp}} = \{ 0 \leq j \leq T - 1 \mid \rho_j < \eta_1 \} .\]

Recall that in Lemma 3.5 of [15], the total number of iterations in the ARp for second-order critical points can be bounded by a function of the number of successful SARp (i.e., \(|\mathcal{S}_{\text{SARp}}|\)). At first glance, we do not expect such bound holds for AHOM as the iterate could possibly be updated at Step 2 of AHOM after performing SARp, resulting in a whole different sequence in contrast with that of ARp. However, we note that the universal bound in Lemma 3.2 of [15] for the cubic regularizer \( \sigma \) is still valid for \( \sigma_{k+1} \) in SARp. Therefore, the same relationship between the two iteration numbers in Lemma 3.5 of [15] is carried over to AHOM by a similar proof.

**Lemma 3** The mechanism of AHOM and its subroutine SARp guarantees that

\[
T \leq |\mathcal{S}_{\text{SARp}}| \left( 1 + \frac{|\log \gamma_1|}{\log \gamma_2} \right) + \frac{1}{\log \gamma_2} \log \left( \frac{\sigma_{\text{max}}}{\sigma_0} \right)
\]

where \( \sigma_{\text{max}} = \max \left\{ \sigma_0, \frac{\gamma_1 L (p+1)}{p (1-\eta_2)} \right\} \).

Next, we consider the successful iteration defined by the update of \( \kappa_{k+1} \).

**Definition 5** If an iteration in AHOM performs a nontrivial update \( x_{j+1} = z_j - \epsilon_j u \) in Step 2, we call it a “third-order successful” iteration. Suppose \( T \) is the total number of iterations in AHOM, we denote by

\[(i) \mathcal{S}_{\text{third}} = \{ 0 \leq j \leq T - 1 \mid \Phi_j \geq \xi_1 \text{ and } \chi_{f,3}(z_j) \geq \beta (24 \cdot \chi_{f,1}(z_k) \cdot \kappa_k^2)^{1/3} \} ,\]

the index set of all “third-order successful” iterations. While all the “third-order unsuccessful” iterations are categorized into two sets:

\[(ii) \mathcal{U}_{\text{third}1} = \{ 0 \leq j \leq T - 1 \mid \Phi_j < \xi_1 \text{ and } \chi_{f,3}(z_j) \geq \beta (24 \cdot \chi_{f,1}(z_k) \cdot \kappa_k^2)^{1/3} \} \]

and

\[(iii) \mathcal{U}_{\text{third}2} = \{ 0 \leq j \leq T - 1 \mid \chi_{f,3}(z_j) < \beta (24 \cdot \chi_{f,1}(z_k) \cdot \kappa_k^2)^{1/3} \} ,\]

due to the violation on the relative decrease \( \Phi \) or the third-order critical measure \( \chi_{f,3}(z_j) \).

According to Lemma 3, it suffices to bound \(|\mathcal{S}_{\text{SARp}}|\) to establish the overall iteration complexity of AHOM. From Definition 4 and 5, we have the following identity:

\[
T - 1 = |\mathcal{S}_{\text{SARp}}| + |\mathcal{U}_{\text{SARp}}| = |\mathcal{S}_{\text{third}}| + |\mathcal{U}_{\text{third}1}| + |\mathcal{U}_{\text{third}2}|.
\]

Consequently,

\[
|\mathcal{S}_{\text{SARp}}| = |\mathcal{S}_{\text{SARp}} \cap (\mathcal{S}_{\text{third}} \cup \mathcal{U}_{\text{third}1} \cup \mathcal{U}_{\text{third}2})|
\]

\[
= |\mathcal{S}_{\text{SARp}} \cap \mathcal{S}_{\text{third}}| + |\mathcal{S}_{\text{SARp}} \cap \mathcal{U}_{\text{third}1}| + |\mathcal{S}_{\text{SARp}} \cap \mathcal{U}_{\text{third}2}|
\]

\[
\leq |\mathcal{S}_{\text{SARp}} \cap \mathcal{S}_{\text{third}}| + |\mathcal{U}_{\text{third}1}| + |\mathcal{S}_{\text{SARp}} \cap \mathcal{U}_{\text{third}2}|. \quad (15)
\]

Next, we shall first bound the term: \(|\mathcal{U}_{\text{third}1}|\). Before doing so, we first provide the benefit of using third-order information.
Furthermore, the construction of the competitive subspace implies that
\[ \delta \] which amounts to
\[ S \] is a unit vector in \( S^z_k \) such that
\[ [\nabla^3 f(z_k)](u, u, u) \geq \chi_{f,3}(z_k)/\beta. \] Let \( x_{k+1} = z_k - \chi_{f,3}(z_k) \beta \kappa_k^2 u \), then we have
\[ f(x_{k+1}) \leq f(z_k) - \xi_1 \chi_{f,3}(z_k)^4/24\beta^4\kappa_k^3, \]
i.e., iteration \( k \) is third-order successful.

**Proof** Let \( \varepsilon = \chi_{f,3}(z_k)/\beta \kappa_k \), \( \delta_1 = \chi_{f,1}(z_k) \) and \( \delta_2 = \max_{y \in S^z_k} y^\top \nabla^2 f(z_k) y \), then by using (8) we have that
\[ f(x_{k+1}) \leq f(z_k) - \varepsilon \nabla f(z_k)^\top u + \frac{\varepsilon^2}{2} \nabla f(z_k) u - \frac{\varepsilon^3}{6} [\nabla^3 f(z_k)](u, u, u) + \frac{L_3 \varepsilon^4}{24} \|u\|^4 \]
\[ \leq f(z_k) + \delta_1 \varepsilon + \frac{\delta_2 \varepsilon^2}{2} - \frac{\|\text{Proj}_{S^z_k} \nabla^3 f(z_k)\|}{6\beta} \varepsilon^3 + \frac{L_3 \varepsilon^4}{24} \]
\[ = f(z_k) + \delta_1 \varepsilon + \frac{\delta_2 \varepsilon^2}{2} - \frac{\chi_{f,3}(z_k) \varepsilon^3}{6\beta} + \frac{L_3 \varepsilon^4}{24}. \]

From the assumption \( \chi_{f,3}(z_k) \geq \beta(24 \delta_1 \kappa_k^2)^{1/3} \), one has that
\[ \delta_1 \varepsilon \leq \frac{\chi_{f,3}(z_k)^3}{24\beta^3 \kappa_k^2} \cdot \varepsilon = \frac{\kappa_k \varepsilon^3}{24} \cdot \left( \frac{\chi_{f,3}(z_k)}{\beta \kappa_k} \right)^3 = \frac{\kappa_k \varepsilon}{24}. \]

Furthermore, the construction of the competitive subspace implies that \( \delta_2 \leq \frac{\chi_{f,3}(z_k)^2}{12\kappa_k \beta^2} \) and thus
\[ \frac{\delta_2 \varepsilon^2}{2} \leq \frac{\chi_{f,3}(z_k)^2 \varepsilon^2}{12 \kappa_k \beta^2} \leq \frac{\kappa_k \varepsilon^3}{12} \cdot \left( \frac{\chi_{f,3}(z_k)}{\beta \kappa_k} \right)^2 \cdot \frac{\varepsilon^2}{2} = \frac{\kappa_k \varepsilon^4}{24}. \]

Therefore, combining the above inequalities with the assumption \( \kappa_k \geq \frac{L_3}{2 - \xi_1} \), which is equivalent to \( \frac{2\kappa_k - L_3}{\kappa_k^2} \geq \xi_1 \), yields that
\[ f(x_{k+1}) \leq f(z_k) - (2 \kappa_k - L_3) \frac{\varepsilon^4}{24} \]
\[ = f(z_k) - \frac{2 \kappa_k - L_3}{\kappa_k} \cdot \frac{\chi_{f,3}(z_k)^4}{24\beta^4 \kappa_k^3} \]
\[ \leq f(z_k) - \xi_1 \frac{\chi_{f,3}(z_k)^4}{24\beta^4 \kappa_k^3}, \]
which amounts to
\[ \Phi_k = \frac{f(z_k) - f(z_k - \varepsilon u)}{\Delta_k} = \frac{f(z_k) - f(x_{k+1})}{\frac{\chi_{f,3}(z_k)^4}{24\beta^4 \kappa_k^3}} \geq \xi_1 \]
meaning that the iteration \( k \) is a third-order successful iteration. \( \square \)

Then it is easy to see that \( \kappa_k \) has an upper bound as shown below.
Lemma 5 For all iteration $k$ in AHOM, we have that
\[ k_k \leq \kappa_{\text{max}} \overset{\text{def}}{=} \max \left\{ \kappa_0, \frac{\zeta L}{2 - \xi_1} \right\} \]  
(17)
where $\zeta > 1$ and $0 < \xi_1 < 1$.

Proof We note that $k_k$ is increased by a factor of $\zeta$ only when $\chi_f,3(z_k) \geq \beta (24 \cdot \chi_f,1(z_k) \cdot k_k^2)^{1/3}$ and $\Phi_k < \xi_1$. However, we have shown in Lemma 4 that $\Phi_k \geq \xi_1$ as long as $\chi_f,3(z_k) \geq \beta (24 \cdot \chi_f,1(z_k) \cdot k_k^2)^{1/3}$ and $k_k \geq \frac{L_3}{2 - \xi_1}$. Therefore, $k_k$ will not be updated once it exceeds $\frac{L_3}{2 - \xi_1}$. We introduce the factor $\zeta > 1$ in $\kappa_{\text{max}}$ to accommodate case when $k_k$ is only slightly less than $\frac{L_3}{2 - \xi_1}$ in its last update. \hfill \Box

As a consequence, we are able to bound the number of type 1 unsuccessful iterations $|U_{\text{third}1}|$ in AHOM.

Lemma 6 It holds that
\[ |U_{\text{third}1}| \leq \left\lfloor \log \left( \frac{\kappa_{\text{max}}}{\kappa_0} \right) \right\rfloor \frac{\log \zeta}{\log \xi}. \]  
(18)

Proof The updating rule of $k_k$ in AHOM gives that
\[ k_{k+1} = \zeta k_k, \ k \in U_{\text{third}1}, \ and \ k_{k+1} = k_k, \ k \in (S_{\text{third}} \cup U_{\text{third}2}). \]
Thus we deduce inductively that
\[ \kappa_0 \zeta^{|U_{\text{third}1}| + |S_{\text{third}}|} \leq \kappa_{\text{max}}. \]
Therefore the conclusion follows by dividing by $\kappa_0$ and then taking log on both sides. \hfill \Box

With all the above results, we are now in position to state our main complexity result below.

Theorem 1 Suppose algorithm AHOM starts at $x_0$, and $f$ has global min $f^*$. Then, given $\epsilon_1 > 0$, $\epsilon_2 > 0$ and $\epsilon_3 > 0$, Algorithm 3 needs at most
\[ \left\lfloor \frac{2 \omega(f(x_0) - f^*)}{\epsilon_1} \right\rfloor \max \left\{ \frac{p + 1}{\eta_1 \sigma_{\text{min}} (L + \theta + \sigma_{\text{max}})} \left( \frac{p + 1}{p} \right), \frac{p + 1}{\eta_1 \sigma_{\text{min}}} ((p - 1)L + \theta + p \sigma_{\text{max}}) \right\}, \]
iterations in total to produce an iterate $x_\epsilon$ such that $\chi_{f,i}(x_\epsilon) \leq \epsilon_i$, $i = 1, 2, 3$, where
\[ \omega \overset{\text{def}}{=} \max \left\{ \frac{p + 1}{\eta_1 \sigma_{\text{min}} (L + \theta + \sigma_{\text{max}})} \left( \frac{p + 1}{p} \right), \frac{p + 1}{\eta_1 \sigma_{\text{min}}} ((p - 1)L + \theta + p \sigma_{\text{max}}) \right\} . \]  
(19)
\[ \tilde{\Delta}_1 = \left( 1 + \frac{|\log \gamma_1|}{\log \gamma_2} \right), \ \tilde{\Delta}_2 = \frac{1}{\log \gamma_2} \log \left( \frac{\sigma_{\text{max}}}{\sigma_0} \right), \ and \ \kappa_{\text{max}} \ is \ given \ by \ (17). \]

Proof According to Lemma 3 and inequality (15), it suffices to bound the three terms: $|S_{\text{SARp}} \cap S_{\text{third}}|$, $|U_{\text{third}1}|$ and $|S_{\text{SARp}} \cup U_{\text{third}2}|$. The upper bound of $|U_{\text{third}1}|$ can be found in Lemma 6. To bound the other two terms, we note that the algorithm AHOM will
not stop as long as either the first-order, the second-order or the third-order critical measure is still above the given tolerance, namely,

\[(a) \chi_{f,1}(x_{k+1}) > \epsilon_1 \text{ or } (b) \chi_{f,2}(x_{k+1}) > \epsilon_2 \text{ or } (c) \chi_{f,3}(x_{k+1}) > \epsilon_3.\]  \hspace{1cm} (20)

In the following, we shall bound $|\mathcal{S}_{SARp} \cap \mathcal{S}_{third}|$ and $|\mathcal{S}_{SARp} \cap \mathcal{U}_{third2}|$ under the three scenarios above.

We first consider the term $|\mathcal{S}_{SARp} \cap \mathcal{S}_{third}|$. Suppose the current iteration $k \in (\mathcal{S}_{SARp} \cap \mathcal{S}_{third})$. The successful criterion $\Phi_k \geq \xi_1$ for $\mathcal{S}_{third}$ in Step 2 of AHOM amounts to

\[f(z_k) - f(x_{k+1}) \geq \xi_1 \frac{\chi_{f,3}(z_k)^4}{24\beta_3^4k_3^4}.\]  \hspace{1cm} (21)

Since $k \in \mathcal{S}_{SARp}$, (iii) in Proposition 1 holds, which together with (21) implies that

\[f(x_k) - f(x_{k+1}) = f(x_k) - f(z_k) + f(z_k) - f(x_{k+1}) \geq \frac{\eta_1 \sigma_{min}}{p+1} \|z_k - x_k\|^{p+1} + \xi_1 \frac{\chi_{f,3}(z_k)^4}{24\beta_3^4k_3^4}.\]  \hspace{1cm} (22)

Next, we further bound the inequality (22) from below according to the three scenarios in (20).

1. In the case of condition (a) in (20) holds, we deduce from (22) and part (i) in Proposition 1 that

\[f(x_k) - f(x_{k+1}) \geq \frac{\eta_1 \sigma_{min}}{p+1} \|z_k - x_k\|^{p+1} > \omega_1 \epsilon_1^{p+1},\]  \hspace{1cm} (23)

where $\omega_1 \overset{\text{def}}{=} \frac{\eta_1 \sigma_{min}}{p+1} \left(\frac{1}{L+\theta+\sigma_{max}}\right)^{\frac{p+1}{p}}$.  

2. In the case of condition (b) in (20) holds, we deduce from (22) and part (ii) in Proposition 1 that

\[f(x_k) - f(x_{k+1}) \geq \frac{\eta_1 \sigma_{min}}{p+1} \|z_k - x_k\|^{p+1} > \omega_2 \epsilon_2^{p+1},\]  \hspace{1cm} (24)

where $\omega_2 \overset{\text{def}}{=} \frac{\eta_1 \sigma_{min}}{p+1} \left(\frac{1}{(p-1)L+\theta+p\sigma_{max}}\right)^{\frac{p+1}{p-1}}$.  

3. In the case of condition (c) in (20) holds, we deduce from (22) have that

\[f(x_k) - f(x_{k+1}) \geq \frac{\xi_1 \chi_{f,3}(z_k)^4}{24\beta_3^4k_3^4} > \omega_3 \epsilon_3^4,\]  \hspace{1cm} (25)

where $\omega_3 \overset{\text{def}}{=} \frac{\xi_1}{24\beta_3^4k_{max}}$.  

Therefore, for any iteration $k \in (\mathcal{S}_{SARp} \cap \mathcal{S}_{third})$, combining (23), (24) and (25) guarantees that:

\[f(x_k) - f(x_{k+1}) \geq \min\{\omega_1, \omega_2, \omega_3\} \min\left\{\epsilon_1^{p+1}, \epsilon_2^{p+1}, \epsilon_3^4\right\}.\]
Recalling \( f^* \) is a universal lower bound of \( f \), one has that

\[
f(x_0) - f^* \geq \sum_{k=0}^{T-1} (f(x_k) - f(x_{k+1}))
\]

\[
\geq \sum_{k \in (S_{SARp} \cap S_{hird})} (f(x_k) - f(x_{k+1}))
\]

\[
\geq \min\{\omega_1, \omega_2, \omega_3\} \min\{\epsilon_1, \epsilon_2, \epsilon_3\} \cdot |S_{SARp} \cap S_{hird}|
\]

and concludes the desired upper bound

\[
|S_{SARp} \cap S_{hird}| \leq \frac{f(x_0) - f^*}{\min\{\omega_1, \omega_2, \omega_3\}} \max\{\epsilon_1^{\frac{p+1}{p}}, \epsilon_2^{\frac{p+1}{p-1}}, \epsilon_3^{-4}\}. \tag{27}
\]

Then, we bound the number of \(|S_{SARp} \cap U_{hird2}|\). Suppose the current iteration \( k \in (S_{SARp} \cap U_{hird2}) \), from (iii) in Proposition 1 and the mechanism of AHOM, we know that

\[
f(x_k) - f(x_{k+1}) = f(x_k) - f(z_k) \geq \frac{\eta_1 \sigma_{\min}}{p+1} \|z_k - x_k\|^{p+1} \tag{28}
\]

and

\[
\chi_{f,3}(z_k) < \beta (24 \cdot \chi_{f,1}(z_k) \cdot \kappa_4^{2})^{1/3}. \tag{29}
\]

Similarly, we further bound the inequality (28) according to the three scenarios in (20). The same argument for (23) and (24) implies that they are still valid for \( k \in (S_{SARp} \cap U_{hird2}) \) under scenarios (a) and (b) in (20), respectively. In the case of third condition (c) in (20) holds, we deduce from (28), (29) and part (i) in Proposition 1 have that

\[
f(x_k) - f(x_{k+1}) \geq \frac{\eta_1 \sigma_{\min}}{p+1} \|z_k - x_k\|^{p+1}
\]

\[
\geq \frac{\eta_1 \sigma_{\min}}{p+1} \left(\frac{\chi_{f,1}(z_k)}{L + \theta + \sigma_{\max}}\right)^{\frac{p+1}{p}}
\]

\[
> \omega_4 \epsilon_3^{\frac{3(p+1)}{p}}
\]

where \( \omega_4 \overset{\text{def}}{=} \frac{\eta_1 \sigma_{\min}}{p+1} \left(\frac{1}{24 \beta \chi_{f,1}(z_k)}(L + \theta + \sigma_{\max})\right)^{\frac{p+1}{p}} \). Therefore, for any iteration \( k \in (S_{SARp} \cap U_{hird2}) \), combining (23), (24), (30) and the argument for (27), we have the following desired bound:

\[
|S_{SARp} \cap U_{hird2}| \leq \frac{f(x_0) - f^*}{\min\{\omega_1, \omega_2, \omega_3\}} \max\{\epsilon_1^{\frac{p+1}{p}}, \epsilon_2^{\frac{p+1}{p-1}}, \epsilon_3^{-\frac{3(p+1)}{p}}\}. \tag{31}
\]
Finally, by invoking Lemma 3, the total number of iterations $T$ can be upper bounded by

$$T \leq |S_{SAR_p}| \left(1 + \frac{\log \gamma_1}{\log \gamma_2}\right) + \frac{1}{\log \gamma_2} \log \left(\frac{\sigma_{\max}}{\sigma_0}\right)$$

$$\leq \left(2\omega(f(x_0) - f^*) \max\{\epsilon_1^{-p+1}, \epsilon_2^{p-1}, \epsilon_3^{-4}, \epsilon_3^{-3(p+1)/p}\} + \frac{\log(\frac{\sigma_{\max}}{\sigma_0})}{\log \zeta}\right)$$

$$\cdot \left(1 + \frac{\log \gamma_1}{\log \gamma_2}\right) + \frac{1}{\log \gamma_2} \log \left(\frac{\sigma_{\max}}{\sigma_0}\right),$$

where $\omega$ is defined in (19).

We remark that the above theorem indicates that our method finds the third-order critical points with iteration complexity $O(\epsilon^{-4})$. Compared to the general result $O(\epsilon^{-(q+1)/(p-q+1)}$ in [16,18] with $q = 3$, our complexity has the same bound when $p = 3$ but becomes weaker if $p > 3$. As we mentioned earlier, the algorithms in [16,18] to achieve the general bound require globally solving a non-convex subproblem, and the computational effort is to some degree hidden in the procedure of solving the subproblem. Thus, it is in some sense unfair to compare our bound with the general bound directly.

5 Numerical experiments

In order to justify the capability of AHOM, our numerical experiments are designed to observe the phenomenon that the solutions returned by AHOM are superior to that by the lower-order (i.e., second-order) method. To this end, the optimization problems to be solved in the experiments should contain third-order critical points and degenerate saddle points, where the second-order method could possibly get stuck. Therefore, in this section, we consider a few specific nonconvex polynomial optimization problems and a nonconvex logistic regression problem applied to some real classification data. We do not adopt the nonconvex optimization problems that are tested in the experiments for the adaptive regularized $p$-th order method (ARp) [6], as the numerical results in [6] seem to indicate that the third-order method (i.e., $p = 3$) and the second-order method (i.e., $p = 2$) converge to the same solution point for these problems.

We implement the AHOM algorithm with $p = 3$ in the experiments and the subroutine SARp in each iteration of AHOM reduces to the adaptive cubic regularization of Newton’s method (ARC). More specifically, we apply the so-called Lanczos [12] process to approximately solve the subproblem $\min_{x \in \mathbb{R}^n} m(x_k, s, \sigma_k)$ in SARp. The parameters of SARp are set as $\sigma_0 = 2.0, \sigma_{\min} = 10^{-16}, \gamma_1 = 0.5, \gamma_2 = 1.1, \gamma_3 = 2.0, \eta_1 = 0.1, \eta_2 = 0.9$, which are the same as that of the benchmark algorithm ARC. For the parameters in the main loop of AHOM, we set $\xi_1 = 10^{-9}, \zeta = 1.1, \kappa_0 = 10^{-6}$ and $\beta = 20$.

We compare our AHOM method ($p=3$) with two second-order methods: ARC and trust region method (TR). Specially, to illustrate the benefits of our proposed AHOM main framework, we apply Lanczos process [12] and GLTR process [25] to approximately solve the subproblems in ARC and TR, respectively. We adopt the ARC and TR in the public package¹ with the default parameters except that full batch rather than subsampled batch of component functions are taken. In ARC, the “initial_penalty_parameter” and “penalty_decrease_multiplier” are set to $\sigma_0 = 2.0$ and $\gamma_2 = 1.1$ (all parameters are

¹ https://github.com/dalab/subsampledcubicregularization.
the same as those in SARp). The initial radius and the max radius of the trust region in TR are 2 and $10^4$ respectively. Recall that the first-order, second-order, and third-order critical measures are given by $\chi_{f,1}(x) = \|\nabla f(x)\|$, $\chi_{f,2}(x) = \max[0, -\lambda_n(\nabla^2 f(x))]$, $\chi_{f,3}(x) = \|\text{Proj}_{S(x)} \nabla^3 f(x)\|_F$, respectively. We set equal error tolerances of $10^{-6}$ for these three measures. Then an approximate third-order critical point satisfies (i) $\chi_{f,1}(x) \leq 10^{-6}$, (ii) $\chi_{f,2}(x) \leq 10^{-6}$, (iii) $\chi_{f,3}(x) \leq 10^{-6}$, which is also the stopping criterion for AHOM. In addition, ARC and TR stop when (i) and (ii) are satisfied. Unless otherwise specified, all the experiments are conducted on a Lenovo laptop with Windows 7, an Intel(R) Xeon(R) CPU E3-1505M v5 @ 2.8GHz, and 16GB memory.

5.1 Numerical experiments for nonconvex polynomial optimization problems

In this subsection, we apply AHOM and the two benchmark algorithms to solve some nonconvex polynomial optimization problems. Those polynomial functions are selected or constructed on purpose to include degenerate saddle points and have a finite global minimum. To verify if AHOM is able to escape from the degenerate saddle points, the initial solutions are chosen to be close to the degenerate saddle points, and we require the algorithms to run at least 200 iterations. As we shall see below, AHOM indeed manages to get away from the degenerate saddle points and converge to a solution with a significantly smaller objective value.

**Example 1** Consider the problem

$$\min_x f(x) := x_1^4 + x_2^4 + x_3^4 - 4x_1x_2x_3, \quad (32)$$

where the objective function is modified from the polynomial $x_1^2 + x_2^4 + x_3^4 - 4x_1x_2x_3$ in [21]. It is easy to verify that the origin $0 = (0, 0, 0)$ is an isolated and degenerate critical point. We choose $0$, $0$, $2$ as the initial point to solve problem (32), then the ARC and the TR algorithm get stuck at $(0, 0, 2.9936e^{-11})$ and $(0, 0, 6.0352e^{-32})$ respectively, and the associated objective values are $f_{\text{ARC}}(x^*) = 8.0311e^{-43}$ and $f_{\text{TR}}(x^*) = 1.3267e^{-125}$. While the AHOM is able to converge to the point $x^* = (-1.0, -1.0, 1.0)$ with $f_{\text{AHOM}}(x^*) = -1.0$.

**Example 2** Consider the problem

$$\min_x f(x) := \sum_{i=1}^{n} x_i^{2m} - (n + 1) \sum_{1 \leq i < k \leq n} x_i x_j x_k x_l, \quad m \geq 3, \ n \geq 4 \quad (33)$$

which is an extension of problem (32). The original $0$ is also a degenerate saddle point. We apply AHOM, ARC, TR to solve problem (33) with the same random initial solutions that are close to the degenerate saddle point $0$. In particular, we randomly pick 2 out of the $n$ components and add random variables drawn of the uniform distribution with support $[0, 2]$. The results for different $m$ and $n$ in (33) are provided in Table 1, which verify our conclusion.

**Example 3** Consider the problem

$$\min_x f(x) = 1 + x_1^4 x_2^2 + x_1^2 x_2^4 - 3x_1^2 x_2^2, \quad (34)$$

where the objective function is the dehomogenized Motzkin polynomial [36]. Note that $(0, 0)$ is an isolated and degenerate critical point. When we choose $(0.0001, 0)$ as the initial point, both ARC and TR get stuck at $(0, 0)$. However, our AHOM is able to converge to the optimal
which is motivated by the term parameter $\alpha$. Consider the problem
\[
\begin{align*}
\min_{x} & \quad f(x) := \left( \sum_{i=1}^{n} x_i^2 \right)^2 \left( \sum_{i=1}^{n} x_i^2 - (n + 1) \right) - 6 \sum_{1 \leq i \neq j \leq n} x_i^2 x_j^2 x_k^2, \quad n \geq 3, \\
\text{subject to} & \quad x_i \geq 0, \quad i = 1, \ldots, n.
\end{align*}
\]

which is motivated by the term $x_1^4 x_2^2 + x_2^2 x_4^2 - 3 x_1^2 x_2^2 = x_1^2 x_2^2 (x_1^2 + x_2^2 - 4)$ in (34). Obviously, the original $0$ is also a degenerate saddle point. We randomly generate a point close to $0$ as the initial point to start ARC, TR, AHOM. In particular, we randomly pick 1 out of the $n$ components and add a random variable drawn from the uniform distribution with support $[0, 0.1]$. The results for different $n$ in (35) are provided in Table 2, which confirm our conclusion.

### 5.2 Numerical experiments for nonconvex logistic regression problems

In this subsection, we show the performance of our algorithm for solving the following nonconvex logistic regression problem:
\[
\min_{x \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^{n} \left( \frac{1}{1 + e^{-w^T x_i}} - y_i \right)^2 + \frac{\alpha}{2} \| w \|^2,
\]

where $\{(x_i, y_i)\}_{i=1}^{n}$ is a collection of data samples with $y_i$ labeled as 0 or 1, and the regularization parameter $\alpha = 10^{-5}$. In contrast with the standard logistic regression, the loss in

### Table 1 Numerical results of problem (33)

| $m$ | $n$ | $f(x)$ | $\| \nabla f(x) \|$ | $\| \nabla f(x) \|$ | $\| \nabla f(x) \|$ | $\| \nabla f(x) \|$ |
|-----|-----|--------|----------------|----------------|----------------|----------------|
| 3   | 4   | $-1.1574$ | $1.1640e^{-14}$ | $1.6568e^{-23}$ | $4.9416e^{-19}$ | $2.8378e^{-108}$ | $1.2911e^{-89}$ |
| 3   | 5   | $-160.0$ | $2.7150e^{-13}$ | $1.6596e^{-23}$ | $4.9418e^{-19}$ | $4.6355e^{-108}$ | $1.7090e^{-89}$ |
| 3   | 9   | $-672.2472$ | $7.7706e^{-13}$ | $1.6557e^{-23}$ | $4.9389e^{-19}$ | $3.8777e^{-105}$ | $5.8571e^{-87}$ |
| 3   | 15  | $-6.8592e^9$ | $6.5602e^{-7}$ | $1.6566e^{-23}$ | $4.9411e^{-19}$ | $7.4304e^{-109}$ | $3.7181e^{-90}$ |
| 5   | 4   | $-1.8899$ | $4.2130e^{-15}$ | $2.7043e^{-18}$ | $1.1707e^{-15}$ | $2.5412e^{-98}$ | $1.1069e^{-87}$ |
| 5   | 9   | $-1.1065e^4$ | $3.8369e^{-12}$ | $2.2666e^{-18}$ | $9.9878e^{-16}$ | $6.9333e^{-95}$ | $1.8182e^{-84}$ |
| 5   | 15  | $-9.1387e^5$ | $4.0376e^{-10}$ | $8.2416e^{-19}$ | $5.3017e^{-16}$ | $5.0729e^{-100}$ | $4.3125e^{-89}$ |

Example 4 Consider the problem
\[
\min_{x} f(x) := \left( \sum_{i=1}^{n} x_i^2 \right)^2 \left( \sum_{i=1}^{n} x_i^2 - (n + 1) \right) - 6 \sum_{1 \leq i \neq j \leq n} x_i^2 x_j^2 x_k^2, \quad n \geq 3, 
\]

which is motivated by the term $x_1^4 x_2^2 + x_2^2 x_4^2 - 3 x_1^2 x_2^2 = x_1^2 x_2^2 (x_1^2 + x_2^2 - 3)$ in (34). Obviously, the original $0$ is also a degenerate saddle point. We randomly generate a point close to $0$ as the initial point to start ARC, TR, AHOM. In particular, we randomly pick 1 out of the $n$ components and add a random variable drawn from the uniform distribution with support $[0, 0.1]$. The results for different $n$ in (35) are provided in Table 2, which confirm our conclusion.

### Table 2 Numerical results of problem (35)

| $n$ | $f(x)$ | $\| \nabla f(x) \|$ | $\| \nabla f(x) \|$ | $\| \nabla f(x) \|$ | $\| \nabla f(x) \|$ |
|-----|--------|----------------|----------------|----------------|----------------|
| 3   | $-15.6735$ | $2.2748e^{-14}$ | $-9.4815$ | $3.1960e^{-11}$ | $-9.4815$ | $1.4211e^{-14}$ |
| 5   | $-118.3432$ | $2.4480e^{-13}$ | $-32.0$ | $1.2960e^{-11}$ | $-32.0$ | $1.7053e^{-13}$ |
| 9   | $-1555.2$ | $3.3412e^{-12}$ | $-148.1481$ | $7.4307e^{-8}$ | $-148.1481$ | $2.2737e^{-13}$ |
| 16  | $-22542.7689$ | $1.4406e^{-9}$ | $-727.8519$ | $4.5475e^{-13}$ | $-727.8519$ | $9.0949e^{-13}$ |
Table 3 Statistics of data sets

| Dataset   | Number of Samples | Dimension |
|-----------|-------------------|-----------|
| a1a       | 1605              | 123       |
| phishing  | 11,055            | 68        |
| sonar     | 208               | 60        |
| splice    | 1000              | 60        |
| svmguide1 | 3089              | 4         |
| svmguide3 | 1243              | 22        |

(36) is quantified as the square of the difference between the logistic function \( \frac{1}{1+e^{-w^\top x_i}} \) and the observed outcome \( y_i \), which thus is a nonconvex function. In fact, there have been some similar nonconvex loss functions proposed and studied in [24,35]. It worths mentioning that both the loss functions in (36) and [24,35] belong to a broader function class named sigmoid function, and there is an optimization model tailored for sigmoid function called sigmoidal programming in [43].

Our experiments are conducted on 6 data sets all come from LIBSVM available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html. The summary of those data sets is shown in Table 3.

To test the performance of AHOM with the starting point from different regions of the search space, we randomly generate 10 initial points in each of the following two settings

1. \( w^0 = \text{numpy.random.normal}(0, 100, d) \),
2. \( w^0 = \text{numpy.random.normal}(0, 1000, d) \),

where \( \text{numpy.random.normal}(\cdot) \) is a python function to generate random vectors from a normal distribution. Then we apply AHOM and the two benchmark methods to solve problem (36), and report the min value, average value, the number of hitting the min value and average running time of the 10 trials in Table 4. According to this table, we can see that the average running time of AHOM is longer than that of ARC and TR in all the six data sets, but the AHOM algorithm is able to converge to a better solution point in data sets “a1a”, “sonar” and “splice”. In the other data sets (“phishing”, “svmguide1” and “svmguide3”), although the three algorithms occasionally converge to the point with the same minimum objective value, AHOM has more chance to hit the minimum value than ARC and TR. In addition, the results also indicate that AHOM is more robust than the other two algorithms in the sense that it obtains a lower average value in most data sets, especially in “a1a”, “splice” and “svmguide1”.

To visualize the performance of the three algorithms, we plot the function values versus iterations and function values versus time of the three algorithms in Figs. 2 and 3 respectively, where the initial points are randomly selected very far from the origin. From the two figures, we can see that AHOM can converge to a better solution for all six data sets, while the other two methods may get stuck at some lower-order critical point. This result indicates that using third-order information can indeed help escape some degenerate saddle points. The detailed information about function value and the three critical measures of the output points of the three algorithms is presented in Table 5, where we can see that AHOM even occasionally converges to a point satisfying second-order sufficient condition (see the rows for the data set “splice” and “svmguide1”).
Table 4 Convergence results of 10 random initial points

| dataset     | min_value | average_value | number_hit_min | average_time |
|-------------|-----------|---------------|----------------|--------------|
|              | AHOM      | ARC           | TR             | AHOM         | ARC           | TR             | AHOM         | ARC           | TR             | AHOM         | ARC           | TR             |
| w0 = numpy.random.normal(0, 10, d) |           |               |                |              |               |                |              |               |                |              |               |                |              |               |                |
| ala         | 74.98032  | 112.20091     | 98.87565       | 106.11467    | 138.85240     | 136.69510      | 4             | 0             | 0              | 325.1380     | 23.9695       | 23.8190        |
| phishing    | 236.17975 | 236.17975     | 236.17975      | 236.17975    | 236.17975     | 236.17975      | 10            | 10            | 10             | 193.4128     | 13.2040       | 109.5266       |
| sonar       | 3.06688   | 4.53584       | 3.56963        | 6.14674      | 6.55688       | 6.36364        | 1             | 0             | 0              | 19.9642      | 6.3038       | 4.3654         |
| splice      | 50.63831  | 56.25948      | 56.25948       | 81.59551     | 108.65312     | 121.77993      | 1             | 0             | 0              | 56.9272      | 24.7039      | 17.5769        |
| svmguide1   | 161.26063 | 161.26063     | 161.26063      | 338.23659    | 604.39644     | 576.80780      | 7             | 3             | 5              | 5.8509       | 2.7889       | 1.9360         |
| svmguide3   | 88.65415  | 88.65415      | 88.65415       | 103.35172    | 108.71049     | 104.12067      | 4             | 3             | 2              | 5.4441       | 2.9408       | 2.4557         |
| w0 = numpy.random.normal(0, 1000, d) |           |               |                |              |               |                |              |               |                |              |               |                |              |               |                |
| ala         | 74.98032  | 148.63951     | 153.68826      | 88.89885     | 180.19408     | 201.53741      | 6             | 0             | 0              | 310.5245     | 56.3087      | 67.5601        |
| phishing    | 236.17975 | 236.17975     | 236.17975      | 265.66310    | 275.87608     | 275.71424      | 5             | 4             | 2              | 487.3529     | 24.6143      | 52.4114        |
| sonar       | 3.54900   | 4.05460       | 3.56121        | 7.13876      | 8.29064       | 6.46351        | 1             | 0             | 0              | 21.6090      | 7.7091       | 6.3652         |
| splice      | 46.80230  | 56.25948      | 56.25948       | 84.11524     | 105.99312     | 104.50340      | 1             | 0             | 0              | 78.7510      | 24.3417      | 18.5465        |
| svmguide1   | 161.26063 | 161.26063     | 161.26063      | 185.82546    | 543.65323     | 379.45453      | 7             | 3             | 5              | 8.0380       | 1.8927       | 0.7993         |
| svmguide3   | 88.65415  | 88.65415      | 88.65415       | 96.61493     | 112.08527     | 116.57744      | 3             | 1             | 1              | 7.2454       | 6.2115       | 4.8244         |
6 Conclusion

There is usually a trade-off between the convergence rate of algorithms and the quality of the solution found by them. The trade-off also exists between the total number of iterations and the per-iteration costs of algorithms. Take the comparison between the first-order and the second-order optimization methods as an example. Steepest descent methods have much cheaper per-iteration cost than Newton methods; however, they usually need more iterations. Furthermore, methods that use second-order information tend to find solutions with better quality with the help of the curvature information of the objective function. However, in addition to the second derivatives, some additional effort is required to guarantee the convergence to the second-order critical point. Therefore, the user must decide which is the best option, depending on the specific problems in practice.

For problems where there are multiple second-order critical points, including some degenerate saddle points, this paper proposes AHOM method to find the third-order critical points by utilizing the higher-order derivative information. This method is implementable as the subproblem is solvable and the adaptive strategy is adopted to tune the parameter in each iteration of the algorithm. Some preliminary numerical results are provided to show how our strategy can be successful in escaping from the degenerate saddle points. Although the pro-
Table 5 Function value and the critical measures at the converged points

| Algorithm | $f(x)$ | $\|\nabla f(x)\|$ | $\lambda_{\text{min}}(\nabla^2 f(x))$ | $\|\text{Proj S} \nabla^3 f(x)\|_F$ |
|-----------|--------|-----------------|---------------------------------|---------------------------------|
| a1a       |        |                 |                                 |                                 |
| AHOM      | 74.9803| 2.6850e-6       | 1e-5                            | 1.8106e-12                      |
| ARC       | 174.4197| 4.0762e-7      | 1e-5                            |                                  |
| TR        | 225.6347| 9.4638e-7      | 4.5531e-6                       |                                  |
| phishing  |        |                 |                                 |                                 |
| AHOM      | 236.1797| 3.0359e-6      | 1e-5                            | 0                               |
| ARC       | 306.5812| 2.5308e-7      | 1e-5                            |                                  |
| TR        | 305.1456| 3.9543e-8      | 1e-5                            |                                  |
| sonar     |        |                 |                                 |                                 |
| AHOM      | 2.0802 | 6.7908e-10     | 1.1569e-5                       | 3.4330e-15                      |
| ARC       | 8.5420 | 2.8592e-7      | 1.0096e-5                       |                                  |
| TR        | 8.0405 | 1.6822e-8      | 9.8650e-6                       |                                  |
| splice    |        |                 |                                 |                                 |
| AHOM      | 56.2595| 3.2888e-14     | 0.3029                          | 3.9487e-12                      |
| ARC       | 154.5323| 4.3664e-7     | 8.4323e-5                       |                                  |
| TR        | 102.5719| 1.0337e-7      | 7.8632e-5                       |                                  |
| svmguide1 |        |                 |                                 |                                 |
| AHOM      | 161.2606| 8.7597e-12    | 4.0163                          | 0                               |
| ARC       | 982.5015| 8.0962e-7     | 4.3880e-5                       |                                  |
| TR        | 305.6594| 3.4369e-7     | 3.7653e-5                       |                                  |
| svmguide3 |        |                 |                                 |                                 |
| AHOM      | 89.1117| 2.5688e-6      | 1e-5                            | 5.4749e-15                      |
| ARC       | 141.6604| 7.3964e-7     | 8.0890e-6                       |                                  |
| TR        | 141.6604| 4.5309e-7     | 8.0868e-6                       |                                  |

posed method is much slower than second-order methods, the quality of the solution obtained is generally better in the tested problems. This phenomenon occurs even with variations of the starting point, which highlights its advantage when there is limited information of the minimizer.

However, there are other improvements that must be made in AHOM so that it can be competitive with simpler methods when applied to general problems. For instance, the worst-case complexity of our algorithm matches the general result when $p = 3$, and it becomes a weaker bound if $p > 3$. Therefore, refining such complexity to $p > 3$ may indicate some progress, even for very specific problems. In addition, a procedure to obtain the competitive subspace with deterministic convergence, rather than the one with expected value, is preferred. Another point that is worth better exploring is the difficulty of excessively small steps due to the use of high-order regularized models, in some situations. This is because of the fact that the minimizer of a function may not be the local minimizer of its Taylor expansion for $p \geq 3$, as shown at [5] and further explored at [1]. Understanding these issues well can motivate more efficient algorithms and possibly reformulate the theory about third-order critical points.
Acknowledgements

We would like to thank the two anonymous referees for their insightful comments, and we would like to also thank Professor Qi Deng at Shanghai University of Finance and Economics for the discussion on the numerical experiment of this paper.

References

1. Amaral, V., Andreani, R., Birgin, E., Marcondes, D., Martínez, J.: On complexity and convergence of high-order coordinate descent algorithms for smooth nonconvex box-constrained minimization. (2020) arXiv preprint arXiv:2009.01811
2. Anandkumar, A., Ge, R.: Efficient approaches for escaping higher order saddle points in non-convex optimization. In: Conference on Learning Theory, pp 81–102 (2016)
3. Baes, M.: Estimate sequence methods: extensions and approximations. Institute for Operations Research, ETH, Zürich, Switzerland (2009)
4. Bellavia, S., Gurioli, G., Morini, B., Toint, P.L.: Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. SIAM J. Optim. 29(4), 2881–2915 (2019)
5. Birgin, E., Krejić, N., Martínez, J.: Economic inexact restoration for derivative-free expensive function minimization and applications. (2020) arXiv preprint arXiv:2009.09062
6. Birgin, E.G., Gardenghi, J., Martínez, J.M., Santos, S.A.: On the use of third-order models with fourth-order regularization for unconstrained optimization. Optim. Lett. 14, 815–838 (2020)
7. Birgin, E.G., Gardenghi, J., Martínez, J.M., Santos, S.A., Toint, P.L.: Evaluation complexity for nonlinear constrained optimization using unscaled KKT conditions and high-order models. SIAM J. Optim. 26(2), 951–967 (2016)
8. Birgin, E.G., Gardenghi, J., Martínez, J.M., Santos, S.A., Toint, P.L.: Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. Math. Prog. 163(1–2), 359–368 (2017)
9. Bubeck, S., Jiang, Q., Lee, Y.T., Li, Y., Sidford, A.: Near-optimal method for highly smooth convex optimization. In: Conference on Learning Theory, pp. 492–507. PMLR (2019)
10. Carmon, Y., Duchi, J.C., Hinder, O., Sidford, A.: Lower bounds for finding stationary points I. Math. Prog. 184, 1–50 (2017)
11. Carmon, Y., Duchi, J.C., Hinder, O., Sidford, A.: Accelerated methods for nonconvex optimization. SIAM J. Optim. 28(2), 1751–1772 (2018)
12. Cartis, C., Gould, N.I., Toint, P.L.: Adaptive cubic regularisation methods for unconstrained optimization. part I: motivation, convergence and numerical results. Math. Program. 127(2), 245–295 (2011)
13. Cartis, C., Gould, N.I., Toint, P.L.: Adaptive cubic regularisation methods for unconstrained optimization. part II: worst-case function-and derivative-evaluation complexity. Math. Program. 130(2), 295–319 (2011)
14. Cartis, C., Gould, N.I., Toint, P.L.: Second-order optimality and beyond: characterization and evaluation complexity in convexly constrained nonlinear optimization. Found. Comput. Math. 18(5), 1073–1107 (2018)
15. Cartis, C., Gould, N.I., Toint, P.L.: A concise second-order complexity analysis for unconstrained optimization using high-order regularized models. Opt. Methods Softw. 35(2), 243–256 (2020)
16. Cartis, C., Gould, N.I., Toint, P.L.: Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints. SIAM J. Optim. 30(1), 513–541 (2020)
17. Chen, X., Jiang, B., Lin, T., Zhang, S.: Accelerating adaptive cubic regularization of Newton’s method via random sampling. J. Mach. Learn. Res. (2022)
18. Chen, X., Toint, P.L.: High-order evaluation complexity for convexly-constrained optimization with non-Lipschitzian group sparsity terms. Math. Program. 187(1), 47–78 (2021)
19. Chen, X., Toint, P.L., Wang, H.: Complexity of partially separable convexly constrained optimization with Non-Lipschitzian singularities. SIAM J. Optim. 29(1), 874–903 (2019)
20. Curtis, F.E., Robinson, D.P., Samadi, M.: An inexact regularized Newton framework with a worst-case iteration complexity of for nonconvex optimization. IMA J. Numer. Anal. 39(3), 1296–1327 (2018)
21. Cushing, J.M.: Extremal tests for scalar functions of several real variables at degenerate critical points. Aequationes Math. 13(1–2), 89–96 (1975)
22. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12, 2121–2159 (2011)
23. Gasnikov, A., Kovalev, D., Mohamed, A., Chernousova, E.: The global rate of convergence for optimal tensor methods in smooth convex optimization. (2018) arXiv preprint arXiv:1809.00382
24. Ghadimi, S., Lan, G., Zhang, H.: Generalized uniformly optimal methods for nonlinear programming. J. Sci. Comput. 79(3), 1854–1881 (2019)
25. Gould, N.I., Lucidi, S., Roma, M., Toint, P.L.: Solving the trust-region subproblem using the lanczos method. SIAM J. Optim. 9(2), 504–525 (1999)
26. Gould, N.I., Rees, T., Scott, J.A.: Convergence and evaluation-complexity analysis of a regularized tensor-Newton method for solving nonlinear least-squares problems. Comput. Optim. Appl. 73(1), 1–35 (2019)
27. Grapiglia, G.N., Nesterov, Y.: Tensor methods for finding approximate stationary points of convex functions. Optim. Methods Softw. pp. 1–34 (2020)
28. Grapiglia, G.N., Nesterov, Y.: Tensor methods for minimizing convex functions with hölder continuous higher-order derivatives. SIAM J. Optim. 30(4), 2750–2779 (2020)
29. Gratton, S., Simon, E., Toint, P.L.: An algorithm for the minimization of nonsmooth nonconvex functions using inexact evaluations and its worst-case complexity. Math. Program. 187(1), 1–24 (2021)
30. Jiang, B., Lin, T., Ma, S., Zhang, S.: Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis. Comput. Optim. Appl. 72, 115–157 (2019)
31. Jiang, B., Lin, T., Zhang, S.: A unified adaptive tensor approximation scheme to accelerate composite convex optimization. SIAM J. Optim. 30, 2897–2926 (2020)
32. Jiang, B., Wang, H., Zhang, S.: An optimal high-order tensor method for convex optimization. Math. Oper. Res. 46, 1390–1412 (2021)
33. Lucchi, A., Kohler, J.: A stochastic tensor method for non-convex optimization. (2019) arXiv preprint arXiv:1911.10367
34. Martínez, J.M.: On high-order model regularization for constrained optimization. SIAM J. Optim. 27(4), 2447–2458 (2017)
35. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent. In: Solla S., Leen T., Müller K. (eds.) Advances in neural information processing systems, vol 12, pp. 512–518 (1999)
36. Motzkin, T.S.: The arithmetic-geometric inequality. Inequalities (Proc. Sympos. Wright-Patterson Air Force Base, Ohio, 1965) pp. 205–224 (1967)
37. Murty, K., Kabadi, S.: Some np-complete problems in quadratic and nonlinear programming. Math. Program. 39, 117–129 (1987)
38. Nesterov, Y.: Introductory lectures on convex optimization a basic course. Appl. Optim. 87(5), 236 (2004)
39. Nesterov, Y.: Accelerating the cubic regularization of Newton’s method on convex problems. Math. Program. 112(1), 159–181 (2008)
40. Nesterov, Y.: Implementable tensor methods in unconstrained convex optimization. Math. Program. 186(1), 157–183 (2021)
41. Nesterov, Y., Polyak, B.T.: Cubic regularization of Newton method and its global performance. Math. Program. 108(1), 177–205 (2006)
42. Nie, J.: The hierarchy of local minimums in polynomial optimization. Math. Program. 151(2), 555–583 (2015)
43. Udell, M., Boyd, S.: Maximizing a sum of sigmoids. Optim. Eng. pp. 1–25 (2013)
44. Zheng, Y., Zheng, B.: A modified adaptive cubic regularization method for large-scale unconstrained optimization problem. (2019) arXiv preprint arXiv:1904.07440

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.