SIAM J. OPTIM.
Vol. 30, No. 4, pp. 2897–2926

A UNIFIED ADAPTIVE TENSOR APPROXIMATION SCHEME TO ACCELERATE COMPOSITE CONVEX OPTIMIZATION*

BO JIANG†, TIANYI LIN‡, AND SHUZHONG ZHANG§

Abstract. In this paper, we propose a unified two-phase scheme to accelerate any high-order regularized tensor approximation approach on the smooth part of a composite convex optimization model. The proposed scheme has the advantage of not needing to assume any prior knowledge of the Lipschitz constants for the gradient, the Hessian, and/or high-order derivatives. This is achieved by tuning the parameters used in the algorithm *adaptively* in its process of progression, which has been successfully incorporated in high-order nonconvex optimization [C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Found. Comput. Math.*, 18 (2018), pp. 1073–1107; E. G. Birgin et al., *Math. Program.*, 163 (2017), pp. 359–368]. By adopting a similar approximate measure of the subproblem in [E. G. Birgin et al., *Math. Program.*, 163 (2017), pp. 359–368] for *nonconvex optimization*, we establish the overall iteration complexity bounds for three specific algorithms to obtain an ϵ-optimal solution for composite convex problems. In general, we show that the adaptive high-order method has an iteration bound of $O\left(1/\epsilon^{1/(p+1)}\right)$ if the first pth-order derivative information is used in the approximation, which has the same iteration complexity as in [M. Baes, *Estimate Sequence Methods: Extensions and Approximations*, Institute for Operations Research, ETH, Zürich, 2009; Y. Nesterov, *Math. Program.*, published online Nov. 21, 2019, https://doi.org/10.1007/s10107-019-01449-1], where the Lipschitz constants are assumed to be known, and the subproblems are assumed to be solved exactly. Thus, our results partially address the problem of incorporating adaptive strategies into the high-order *accelerated* methods raised by Nesterov in [*Math. Program.*, published online Nov. 21, 2019, https://doi.org/10.1007/s10107-019-01449-1], although our strategies cannot ensure the convexity of the auxiliary problem, and such adaptive strategies are already popular in high-order nonconvex optimization [C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Found. Comput. Math.*, 18 (2018), pp. 1073–1107; E. G. Birgin et al., *Math. Program.*, 163 (2017), pp. 359–368]. Specifically, we show that the gradient method achieves an iteration complexity on the order of $O\left(1/\epsilon^{1/2}\right)$, which is known to be best possible (cf. [Y. Nesterov, *Lectures on Convex Optimization*, 2nd ed., Springer, 2018]), while the adaptive cubic regularization methods with the exact/inexact Hessian matrix achieve an iteration complexity on the order of $O\left(1/\epsilon^{1/3}\right)$, which matches that of the original accelerated cubic regularization method presented in [Y. Nesterov, *Math. Program.*, 112 (2008), pp. 159–181]. The results of our numerical experiment clearly show the effect of the acceleration displayed in the adaptive Newton's method with cubic regularization on a set of regularized logistic regression instances.

Key words. convex optimization, tensor method, acceleration, adaptive method, iteration complexity

AMS subject classifications. 90C06, 90C60, 90C53

DOI. 10.1137/19M1286025

*Received by the editors September 9, 2019; accepted for publication (in revised form) July 2, 2020; published electronically October 8, 2020.

https://doi.org/10.1137/19M1286025

Funding: This work was supported in part by NSFC grants 11771269 and 11831002, in part by National Science Foundation grant CMMI-1462408, in part by Shenzhen Research Fund grant KQTD2015033114415450, and in part by the Program for Innovative Research Team of Shanghai University of Finance and Economics.

†Research Institute for Interdisciplinary Sciences, School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China (isyebojiang@gmail.com).

‡Department of Industrial Engineering and Operations Research, UC Berkeley, Berkeley, CA 94720 USA (darren_lin@berkeley.edu).

§Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, MN 55455 USA, and Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, China (zhangs@umn.edu).

1. Introduction. In this paper, we consider the following generic composite convex optimization model:

$$F^* := \min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) = f(\mathbf{x}) + r(\mathbf{x}), \tag{1.1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is *convex* and *smooth*, $r : \mathbb{R}^d \to \mathbb{R}$ is *convex* but possibly *non-smooth* with simple proximal mapping, and $F^* > -\infty$. During the past few decades, various classes of optimization algorithms for solving (1.1) (especially when $r(\mathbf{x}) = 0$ and $F(\mathbf{x})$ becomes smooth) have been developed and carefully analyzed; for relevant information see [40, 50, 48] and references therein. Despite the nice theoretical property of the existing solution methods, there remains a practical concern regarding the implementation, as many methods assume that some problem parameters, such as the first- and second-order Lipschitz constants, are available, and these parameters may be hard to estimate in practice. It would be ideal to come up with optimization algorithms that automatically estimate such parametric values; such algorithms not only would be easy to implement but also would maintain superior theoretical iteration bounds. In this case, we are demanding that an algorithm be less dependent on the knowledge of the problem structure at hand and therefore less prone to failures due to misinformation about such values. In this context, schemes that adaptively adjust the parameters used in the algorithms are often desirable and likely to lead to better numerical performances. For instance, researchers tend to train their deep learning models with an adaptive gradient method (see, e.g., AdaGrad in [23]) due to its robustness and effectiveness (cf. [36]). In fact, Adam [37] and RMSProp [53] are recognized as the default solution methods in the deep learning setting. Among the category of second-order methods, Cartis, Gould, and Toint [14, 15] proposed and analyzed an adaptive cubic regularized Newton's method, which soon became very popular due to its numerical efficiency. Nesterov proposed two implementable high-order methods in a recent paper [49], where he also commented that an unsolved issue in his approach was a dynamic adjustment scheme for the Lipschitz constant of the highest derivative to achieve practical efficiency.

Another fundamental issue in optimization (as well as in machine learning) is understanding how the classical algorithms (including the first-order, second-order, and high-order methods) can be accelerated. Nesterov [44] put forward the very first accelerated (optimal in its iteration counts) gradient-based algorithm for smooth convex optimization. Beck and Teboulle [4] successfully extended Nesterov's approach to accommodate the problem in the form of (1.1). Recently, accelerated algorithms were extended to incorporate second-order [46, 43] or high-order information [3, 49, 24] yielding a faster convergence rate. However, these algorithms do require the knowledge of some problem-specific parametric (Lipschitz) constants.

Overall, algorithms exhibiting traits of both *acceleration* and *adaptation* have been largely missing in the context of convex optimization. To the best of our knowledge, besides the current paper and two very recent reports [30, 31], no other paper has appeared on accelerated second-order methods (or any high-order methods) that are fully independent of the problem constants while maintaining superior theoretical iteration bounds. However, there are results on some combinations of the above flavors. For instance, the adaptive cubic regularized Newton's method [16] is Hessian-free and problem-parameter-free and allows a subproblem to be solved inexactly, but it merely achieves an iteration bound of $O\left(1/\epsilon^{1/2}\right)$ without acceleration. Thus, the following natural question arises: Can we develop an implementable accelerated second-order method with an iteration complexity lower than $O\left(1/\epsilon^{1/2}\right)$? One goal of this paper

is to present an affirmative answer to this question. It turns out that the resulting accelerated adaptive cubic regularization algorithm displays excellent numerical performance in solving a variety of large-scale machine learning models in our experiments.

1.1. Related work. Nesterov's seminal work [44] triggered a burst of research on accelerating first-order methods. Recently, a good deal of effort has been put into studying adaptive gradient methods with optimal convergence rate [23, 47, 39, 42] and those methods are widely used in training deep neural networks [37, 53]. In the case when second-order information is available, Nesterov accelerated the cubic regularized Newton's method [46] and obtained an improved iteration bound of $O\left(1/\epsilon^{1/3}\right)$. After that, Monteiro and Svaiter [43] managed to accelerate the Newton's proximal extragradient method with a faster convergence rate of $O\left(1/\epsilon^{2/7}\right)$. Very recently, Arjevani, Shamir, and Shiff [2] proved that $O\left(1/\epsilon^{2/7}\right)$ is actually a lower bound for the oracle complexity of the second-order methods for convex optimization, which implies that Monteiro and Svaiter's method is an optimal second-order method. In a recent work, Ghadimi, Liu, and Zhang [27] generalized the accelerated Newton's method with cubic regularization under inexact second-order information. However, the complexity bound is theoretically worse than that of its exact counterparts and is only as good as that of the optimal first-order method. Baes [3] extended the method from [46] to the high-order case and further improved the iteration complexity to $O(1/\epsilon^{1/(p+1)})$. This extension was recently revisited by Nesterov [49], who elaborated on an efficient implementation when $p = 3$. On the other hand, Arjevani, Shamir, and Shiff [2] showed that the worst-case iteration complexity of any high-order algorithm cannot be better than $O(1/\epsilon^{2/(3p+1)})$, and shortly after that an optimal high-order method was proposed in [24] with iteration complexity matching this lower bound. However, an additional bisection search is needed in each iteration of the method, and the number of bisection steps consumed is bounded by a logarithmic factor in the given precision [43, 35, 10]. Recently, Wilson, Mackey, and Wibisono [54] proved that a family of first-order rescaled gradient descent algorithms can achieve the same convergence rate as the optimal pth tensor algorithms for optimizing the so-called pth-order strongly smooth (see [54]) objective.

Although the parameter-free approach is well studied in the first-order case, all the aforementioned high-order (including second-order) accelerated methods assume that the Lipschitz constant for a certain degree of derivative is known, which may be unrealistic. To alleviate this, Cartis, Gould, and Toint [14, 15, 16] incorporated adaptive strategies into the method of Nesterov and Polyak [45] and further relaxed the criterion for solving each subproblem while maintaining the convergence properties for both convex [16] and nonconvex [14, 15] cases. However, as mentioned earlier, the iteration complexity established in [16] for convex optimization is merely $O\left(1/\epsilon^{1/2}\right)$.

In the context of nonconvex optimization, high-order information had already been proved to be useful to improve the convergence rate of the algorithms. In particular, Birgin et al. [8] first proposed a high-order regularization method similar to the cubic regularized algorithms in [14, 15] using adaptive parameter-tuning and inexact subproblem solving. Interestingly, high-order information enables finding high-order critical points [19] yielding a solution with better quality, and the method in [8] was improved by Cartis, Gould, and Toint [18] to converge to second-order critical points. Compared to the cubic regularized algorithm in [14, 15], the high-order regularization methods in [8, 18] have better iteration complexity for finding the first- and second-order critical points. Recently, the high-order methods were proposed to solve

nonsmooth and/or constrained optimization problems [7, 41, 20, 21, 22, 9].

In the literature, there are second-order methods which are efficient for solving (1.1), and they are referred to as proximal (quasi-)Newton methods. The global convergence and the local superlinear rate of convergence of those methods have been shown in [38] and more recently in [11]. Grapiglia and Nesterov [29] studied accelerated regularized Newton's methods for solving problem (1.1), where f is twice differentiable with a Hölderian continuous Hessian, and they showed that the iteration bound depends on the Hölderian parameter. As we were finalizing this paper, we noticed that Grapiglia and Nesterov [30, 31] extended their previous results to the high-order case including an adaptive variant with theoretical guarantees similar to ours, where the Hölderian parameter may be unknown. In comparison with the algorithm proposed in this paper, their algorithms only have a single phase, have a different acceptance condition, and use a different auxiliary function. In addition, the adaptive parameter in their auxiliary function is updated via computing a positive solution of a suitable univariate polynomial equation, which guarantees a key inequality that ensures acceleration. Such a parameter in our algorithm is dynamically adjusted. Consequently, different parameter choices lead to slightly different numerical performances (see section 5.2 for more details).

1.2. Contributions. The contributions of this paper can be summarized as follows. We present a unified adaptive accelerating scheme that can be specialized to several optimization algorithms, including gradient method, cubic regularized Newton's method with *exact/inexact* Hessian, and high-order method. For the gradient method, our adaptive algorithm achieves a convergence rate of $O\left(1/\epsilon^{1/2}\right)$ (Theorem 4.1), which matches the optimal rate for the first-order methods [48]. For the cubic regularized Newton's method we show that a global convergence rate of $O\left(1/\epsilon^{1/3}\right)$ holds (Theorem 4.2) without assuming any knowledge of the problem parameters. We further prove that, even without the exact Hessian information, the same $O\left(1/\epsilon^{1/3}\right)$ rate of convergence (Theorem 4.3) is still achievable for the cubic regularized approximative Newton's method. When our adaptive scheme reduces to the high-order method, the global rate of $O\left(1/\epsilon^{1/(p+1)}\right)$ is guaranteed by utilizing up to pth-order information, which achieves the same iteration bound as that in Baes [3] and Nesterov [49]. Therefore, all the algorithms developed in this paper are problem-parameter-free due to the adopted *fully adaptive* strategies, and they retain the same convergence rate. Note that the accelerated first-order methods proposed in [51, 12] shared the same characteristics, albeit that analysis is quite different. Similarly, the algorithms in [30, 31] by Grapiglia and Nesterov are parameter-independent, and their convergence rates also match those of the nonadaptive ones. In addition, the adaptivity enables an efficient implementation of the algorithm, while numerical experiments are largely missing in the literature of high-order methods. There are a few numerical results reported in [27]; however, the convergence rate is shown to be only as good as that of the accelerated first-order method. In this paper, we performed numerous numerical experiments which clearly showed the effect of acceleration of the proposed algorithms. Finally, our convergence rate, which attains the same order of magnitude as that in [46, 3, 49], is inferior to the rate for the optimal high-order method [24]. However, the gap between the two is small indeed; e.g., for $p = 2$ the gap amounts to $O\left(1/\epsilon^{1/3-2/7}\right) = O\left(1/\epsilon^{1/21}\right)$. Arguably, the additional logarithmic factors required by the optimal method [35, 10] could easily dominate the gap for practical ϵ values.

1.3. Notation and organization.

Notation. We denote vectors by bold lowercase letters, e.g., $\mathbf{x}$, and matrices by italic uppercase letters, e.g., X. The transpose of a real vector $\mathbf{x}$ is denoted as $\mathbf{x}^\top$. For a vector $\mathbf{x}$ and a matrix X, $\|\mathbf{x}\|$ and $\|X\|$ denote the ℓ_2 norm and the matrix spectral norm, respectively. We use $\lambda_{\min}(X)$ to denote the minimum eigenvalue of the matrix X, and $\nabla f(\mathbf{x})$, $\nabla^2 f(\mathbf{x})$, and $\nabla^d f(\mathbf{x})$ indicate the gradient, the Hessian, and the pth-order derivative tensor of f at $\mathbf{x}$, respectively. We denote

$$\nabla^d f(\mathbf{x})[\mathbf{x}^1, \ldots, \mathbf{x}^d] := \sum_{i_1,\ldots,i_d=1}^{n} \nabla^d f(\mathbf{x})_{i_1,\ldots,i_d} \mathbf{x}^1_{i_1} \ldots \mathbf{x}^d_{i_d},$$

and I denotes the identity matrix. For two symmetric matrices A and B, $A \succeq B$ indicates that $A - B$ is symmetric positive semidefinite. The $\log(x)$ denotes the natural logarithm of x for $x > 0$.

Organization. The rest of the paper is organized as follows. In section 2, we introduce some preliminaries and the assumptions used throughout this paper. In section 3, we propose our general framework for adaptively accelerating various optimization algorithms and present the main theoretical results on the iteration complexity. Section 4 is devoted to specializations of our framework to first-order, second-order, and high-order methods. In section 5, we present some preliminary numerical results on solving ℓ_2-regularzied and ℓ_1-regularized logistic regression problems, where acceleration of the method based on the adaptive cubic regularization for Newton's method is clearly observed. The details of all proofs can be found in the appendix.

2. Preliminaries.

Throughout this paper, we make the following assumptions for problem (1.1).

ASSUMPTION 2.1. *F is a proper, closed, and convex function in the domain*

$$\mathsf{dom}(F) := \left\{\mathbf{x} \in \mathbb{R}^d \mid F(\mathbf{x}) < +\infty\right\},$$

and the optimal set of problem (1.1) *is nonempty.*

ASSUMPTION 2.2. *The function f is pth continuously differentiable, and $\nabla^j f$ is Lipschitz continuous with $L_j > 0$ for $p - 1 \le j \le p$, i.e.,*

$$\left\|\nabla^j f(\mathbf{x}) - \nabla^j f(\mathbf{y})\right\| \le L_j \|\mathbf{x} - \mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in \mathsf{dom}(F), \tag{2.1}$$

where

$$\left\|\nabla^j f(\mathbf{x}) - \nabla^j f(\mathbf{y})\right\| = \max_{\|z^i\|=1,\, i=1,\ldots,j} \left(\nabla^j f(\mathbf{x}) - \nabla^j f(\mathbf{y})\right)[z^1 \ldots z^j]$$

is the operator norm associated with the tensor $\nabla^j f(\mathbf{x}) - \nabla^j f(\mathbf{y})$.

We remark that the pth-order Lipschitz continuity condition in Assumption 2.2 is standard in the convergence analysis of pth-order optimization methods for minimizing smooth functions [8, 49]. The Lipschitz continuous assumption on both pth- and $(p-1)$th-order derivatives is common in the derivative-free method with $p = 2$ [17] and is only needed in subsection 4.2.2 of this paper to deal with the second-order method with inexact Hessian information. We consider the following pth-order approximation

of $f(\mathbf{y})$ at point $\mathbf{x}$:

$$\begin{aligned}&\tilde{f}_p(\mathbf{y};\mathbf{x})\\&= f(\mathbf{x}) + (\mathbf{y}-\mathbf{x})^\top \nabla f(\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathbf{x})^\top \nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x})\\&\quad + \sum_{j=3}^{p} \frac{1}{j!} \nabla^j f(\mathbf{x}) \underbrace{[\mathbf{y}-\mathbf{x},\ldots,\mathbf{y}-\mathbf{x}]}_{j \text{ terms}}.\end{aligned}$$

Under Assumptions 2.1 and 2.2, the following two inequalities follow from residual analysis for the Taylor expansion (see also [8, 49]):

$$\left| f(\mathbf{y}) - \tilde{f}_p(\mathbf{y};\mathbf{x}) \right| \le \frac{L_p \|\mathbf{y}-\mathbf{x}\|^{p+1}}{(p+1)!} \tag{2.2}$$

and

$$\left\| \nabla f(\mathbf{y}) - \nabla \tilde{f}_p(\mathbf{y};\mathbf{x}) \right\| \le \frac{L_p \|\mathbf{y}-\mathbf{x}\|^{p}}{p!}. \tag{2.3}$$

Based on $\tilde{f}_p(\mathbf{y};\mathbf{x})$, we consider other approximations of $f(\mathbf{y})$. We call function $\overline{m}(\mathbf{y};\mathbf{x})$ an *effective approximation* of the smooth function $f(\mathbf{y})$ at point $\mathbf{x}$ if the following properties hold.

DEFINITION 2.3. *We call $\overline{m}(\mathbf{y};\mathbf{x})$ an effective approximation of $f(\mathbf{y})$ at a given point $\mathbf{x} \in \mathit{dom}(F)$ if it satisfies the following three properties:*

(i) *For any $\mathbf{y} \in \mathit{dom}(F)$, it holds that*

$$|f(\mathbf{y}) - \overline{m}(\mathbf{y};\mathbf{x})| \le \bar{\kappa}_p \|\mathbf{y}-\mathbf{x}\|^p + \kappa_p \|\mathbf{y}-\mathbf{x}\|^{p+1} \tag{2.4}$$

for some constants $\bar{\kappa}_p$ and κ_p.

(ii) *For any $\bar{\mathbf{x}} \approx \operatorname{argmin}_{\mathbf{y}\in\mathbb{R}^d} \ m(\mathbf{y};\mathbf{x},\sigma)$, it holds that*

$$|f(\bar{\mathbf{x}}) - \overline{m}(\bar{\mathbf{x}};\mathbf{x})| \le \beta_p \|\bar{\mathbf{x}}-\mathbf{x}\|^{p+1}, \tag{2.5}$$

$$\|\nabla f(\bar{\mathbf{x}}) - \nabla \overline{m}(\bar{\mathbf{x}};\mathbf{x})\| \le \rho_p \|\bar{\mathbf{x}}-\mathbf{x}\|^{p}, \tag{2.6}$$

or the above two inequalities hold for a pair $(h,\bar{\mathbf{x}})$ satisfying $\|\bar{\mathbf{x}}-\mathbf{x}\| \ge h$ when $\overline{m}(\bullet\,;\mathbf{x})$ is additionally dependent on some positive number h, where all the parameters are constants.

(iii) *$\overline{m}(\mathbf{y};\mathbf{x})$ is convex in $\mathbf{y}$.*

We remark that the dependence of $\overline{m}(\bullet\,;\mathbf{x})$ on h only occurs in subsection 4.2.2, where an approximated Hessian matrix is constructed based on step size h, leading to such a dependence. The pair $(h,\bar{\mathbf{x}})$ satisfying (2.5) and (2.6) for $\|\bar{\mathbf{x}}-\mathbf{x}\| \ge h$ can be found by a procedure similar to Algorithm 4.1 in [17]. The specific choices of $\overline{m}(\mathbf{y};\mathbf{x})$ and the corresponding values of $\bar{\kappa}_p, \kappa_p, \beta_p, \rho_p$ will be discussed in section 4 and summarized in Table 1. With an effective approximation $\overline{m}(\mathbf{y};\mathbf{x})$ of $f(\mathbf{y})$ in hand, the approximation model for the objective function $F(\mathbf{y})$ is now given by

$$m(\mathbf{y};\mathbf{x},\sigma) := \overline{m}(\mathbf{y};\mathbf{x}) + \frac{\sigma \|\mathbf{y}-\mathbf{x}\|^{p+1}}{p+1} + r(\mathbf{y}). \tag{2.7}$$

We end this section by specifying the definitions of *ε-optimality* and *proximal mapping,* which are frequently used in this paper.

DEFINITION 2.4 (ε-optimality). *Given $\varepsilon \in (0,1)$, $\mathbf{x} \in \mathbb{R}^d$ is said to be ε-optimal to problem* (1.1) *if*

$$F(\mathbf{x}) - F(\mathbf{x}^*) \leq \varepsilon,$$

where $\mathbf{x}^* \in \mathbb{R}^d$ is an optimal solution to problem (1.1).

DEFINITION 2.5 (proximal mapping). *The proximal mapping of r at $\mathbf{x} \in \mathbb{R}^d$ is*

$$\operatorname{prox}_r(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{z}\in\mathbb{R}^d} \; r(\mathbf{z}) + \frac{\|\mathbf{z}-\mathbf{x}\|^2}{2}.$$

3. Algorithmic framework. In this section, we propose a unified framework for accelerating the adaptive methods. This framework is composed of two subroutines: simple adaptive subroutine (SAS) and accelerated adaptive subroutine (AAS). Specifically, the framework starts with SAS, which terminates as soon as one successful iteration is identified. Then, the output of SAS is used as the initial point to run AAS until a sufficient number of successful iterations T_2 is observed. We also adopt the same auxiliary model as that used by Nesterov in [46, 49] except for the appearance of the subgradient ξ due to the additional nonsmooth regularization,

$$\psi_{j+1}(\mathbf{z}, \tau_{j+1}) = l_{j+1}(\mathbf{z}) + \tau_{j+1} R(\mathbf{z}), \tag{3.1}$$

where $R(\mathbf{z}) = \frac{1}{2(p+1)} \|\mathbf{z} - \bar{\mathbf{x}}_0\|^{p+1}$, $l_0(\mathbf{z}) = F(\bar{\mathbf{x}}_0)$, $l_{j+1}(\mathbf{z}) = l_j(\mathbf{z}) + \Delta l_j\left(\mathbf{z}; \bar{\mathbf{x}}_{j+1}, \bar{\xi}_{j+1}\right)$, and

$$\Delta l_j(\mathbf{z}, \mathbf{x}, \xi) = \frac{\Pi_{\ell=2}^{p+1}(j+\ell)}{p!} \left[F(\mathbf{x}) + (\mathbf{z}-\mathbf{x})^\top (\nabla f(\mathbf{x}) + \xi)\right].$$

The details of our algorithmic framework are summarized in Algorithm 3.1 (in the order "Main Procedure"–"SAS"–"AAS").

We remark that the two-phase scheme is necessary in our analysis to establish the accelerated rate of convergence while maintaining promising numerical performance. Some key ingredients of the framework are explained below:

Input: The input contains nine elements: $\mathbf{x}_0 \in \mathbb{R}^d$ is the initial point; σ_0 is the initial regularization parameter for the approximate model; $\sigma_{\min}$ is the safeguard level for the regularization parameter; τ_0 is the initial regularization parameter for the auxiliary model; $\gamma_1, \gamma_2, \gamma_3 \in (1, +\infty)$ are the ratios for adapting σ and τ; and $\eta > 0$ is the threshold for AAS. The approximation model $m(\cdot)$ is as given in (2.7).

In each iteration of our algorithm, we seek an approximate solution that minimizes $m(\cdot; \mathbf{x}, \sigma)$, which is defined as follows.

DEFINITION 3.1. *Let us call $\bar{\mathbf{x}} \approx \operatorname{argmin}_{\mathbf{y}\in\mathbb{R}^d} m(\mathbf{y}; \mathbf{x}, \sigma)$ with $\bar{\xi} \in \partial r(\bar{\mathbf{x}})$ if $m(\bar{\mathbf{x}}; \mathbf{x}, \sigma) \leq m(\mathbf{x}; \mathbf{x}, \sigma)$ and*

$$\left\|\nabla \overline{m}(\bar{\mathbf{x}}; \mathbf{x}) + \sigma \|\bar{\mathbf{x}} - \mathbf{x}\|^{p-1} (\bar{\mathbf{x}} - \mathbf{x}) + \bar{\xi}\right\| \leq \kappa_\theta \|\bar{\mathbf{x}} - \mathbf{x}\|^p, \qquad \kappa_\theta > 0. \tag{3.2}$$

Note that condition (3.2) without $\bar{\xi}$ was first proposed in [8] for smooth nonconvex optimization. In the case of $r(\mathbf{x}) = 0$ and $p = 2$, such an approximativeness measure does not include the condition

$$(\bar{\mathbf{x}} - \mathbf{x})^\top \nabla f(\mathbf{x}) + (\bar{\mathbf{x}} - \mathbf{x})^\top \nabla^2 f(\mathbf{x})(\bar{\mathbf{x}} - \mathbf{x}) + \sigma \|(\bar{\mathbf{x}} - \mathbf{x})\|^3 = 0 \tag{3.3}$$

and thus is weaker than the one used in [14]. This relaxation also suggests other approximations and implementable solution methods for (3.3). For instance, Carmon

Algorithm 3.1. A generic unified adaptive acceleration (UAA) framework.

Main Procedure:

Input: $\mathbf{x}_0 \in \mathbb{R}^d$, $\sigma_0 \geq \sigma_{\min} > 0$, $\tau_0 > 0$, $\gamma_2 > \gamma_1 > 1$, $\gamma_3 > 1$, $\eta > 0$, and approximate model $m(\cdot)$.

Phase I (SAS): $\left[\bar{\mathbf{x}}_0, \sigma^{\mathsf{SAS}}\right] = \mathsf{SAS}\left(\mathbf{x}_0, \sigma_0, \sigma_{\min}, \gamma_1, \gamma_2, m\right)$.

if $p \geq 3$ (p is the power index of the regularizer in $m(\cdot)$) **then**

$\sigma_0^{\mathsf{AAS}} = \max\{\sigma^{\mathsf{SAS}}, -\lambda_{\min}\left(\nabla^2 \overline{m}(\bar{\mathbf{x}}_0; \mathbf{x}_0)\right)/\|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|^{p-1}\}$.

else

$\sigma_0^{\mathsf{AAS}} = \sigma^{\mathsf{SAS}}$.

end if

Phase II (AAS): $[\mathbf{x}_{out}] = \mathsf{AAS}\left(\bar{\mathbf{x}}_0, \sigma_0^{\mathsf{AAS}}, \sigma_{\min}, \tau_0, \gamma_1, \gamma_2, \gamma_3, \eta, m\right)$.

Output: an ε-optimal solution $\mathbf{x}_{out}$.

Simple Adaptive Subroutine: $\mathsf{SAS}\left(\mathbf{x}_0, \sigma_0, \sigma_{\min}, \gamma_1, \gamma_2, m\right)$

Initialization: the total iteration count $i = 0$ and successful iteration count $j = 0$.

repeat

compute $\mathbf{x}_{i+1} \approx \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^d}\ m(\mathbf{x}; \mathbf{x}_i, \sigma_i)$.

if $F(\mathbf{x}_{i+1}) - m\left(\mathbf{x}_{i+1}; \mathbf{x}_i, \sigma_i\right) < 0$ **then**

update $\sigma_{i+1} \in [\sigma_{\min}, \sigma_i]$ and $j = j + 1$.

else

update $\mathbf{x}_{i+1} = \mathbf{x}_i$ and $\sigma_{i+1} \in [\gamma_1\sigma_i, \gamma_2\sigma_i]$.

end if

update $i = i + 1$.

until the successful iteration count $j = 1$.

Output: the total iteration number i, the iterate $\mathbf{x}_i$, and the regularization parameter σ_i.

Accelerated Adaptive Subroutine: $\mathsf{AAS}\left(\mathbf{x}_0, \sigma_0, \sigma_{\min}, \tau_0, \gamma_1, \gamma_2, \gamma_3, \eta, m\right)$

Initialization: the total iteration count $i = 0$ and successful iteration count $j = 0$.

Initial Step: construct the auxiliary model $\psi_0(\mathbf{z}, \tau_0) = l_0(\mathbf{z}) + \tau_0 R(\mathbf{z})$, update $\bar{\mathbf{x}}_0 = \mathbf{x}_0$, compute $\mathbf{z}_0 = \operatorname{argmin}_{\mathbf{z}\in\mathbb{R}^d}\ \psi_0(\mathbf{z}, \tau_0)$ and $\mathbf{y}_0 = \frac{1}{p+2}\bar{\mathbf{x}}_0 + \frac{p+1}{p+2}\mathbf{z}_0$.

for $i = 0, 1, 2, \ldots$ until convergence, **do**

compute $\mathbf{x}_{i+1} \approx \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^d}\ m(\mathbf{x}; \mathbf{y}_j, \sigma_i)$ and $\xi_{i+1} \in \partial r\left(\mathbf{x}_{i+1}\right)$.

if $\theta(\mathbf{x}_{i+1}, \mathbf{y}_j, \xi_{i+1}) \geq \eta$ **then**

update $\bar{\mathbf{x}}_{j+1} = \mathbf{x}_{i+1}$ and $\bar{\xi}_{j+1} = \xi_{i+1}$.

update $l_{j+1}(\mathbf{z}) = l_j(\mathbf{z}) + \Delta l_j\left(\mathbf{z}; \bar{\mathbf{x}}_{j+1}, \bar{\xi}_{j+1}\right)$ and $\tau_{j+1} = \tau_j$.

repeat

update $\tau_{j+1} = \gamma_3\tau_{j+1}$, and

$\mathbf{z}_{j+1} = \operatorname{argmin}_{\mathbf{z}\in\mathbb{R}^d}\ \{\psi_{j+1}(\mathbf{z}, \tau_{j+1}) = l_{j+1}(\mathbf{z}) + \tau_{j+1}R(\mathbf{z})\}$.

until $\psi_{j+1}(\mathbf{z}_{j+1}, \tau_{j+1}) \geq \frac{\Pi_{\ell=1}^{p+1}(j+1+\ell)}{(p+1)!}F(\bar{\mathbf{x}}_{j+1})$

update $\mathbf{y}_{j+1} = \frac{(j+1)+1}{(j+1)+p+2}\bar{\mathbf{x}}_{j+1} + \frac{p+1}{(j+1)+p+2}\mathbf{z}_{j+1}$, $\sigma_{i+1} \in [\sigma_{\min}, \sigma_i]$ and $j = j + 1$.

else

update $\mathbf{x}_{i+1} = \mathbf{x}_i$ and $\sigma_{i+1} \in [\gamma_1\sigma_i, \gamma_2\sigma_i]$.

end if

end for

Output: the total number of iterations i and the iterate $\mathbf{x}_i$.

and Duchi [13] proposed using the gradient descent method to solve (3.3), and they proved that it works well even when $m(\mathbf{y}; \mathbf{x}, \sigma)$ is nonconvex. However, the function $m(\mathbf{y}; \mathbf{x}, \sigma)$ in our case is strictly convex as long as $y \neq x$, and thus the gradient

descent method is likely to exhibit fast (linear) convergence behavior. When $r(\mathbf{x}) \neq 0$ and $p = 2$, we solve the subproblem with the accelerated proximal gradient descent (APGD) method as $m(\mathbf{y}; \mathbf{x}, \sigma)$ is guaranteed to be convex in this case. For the more general case of $r(\mathbf{x}) \neq 0$ and $p \geq 3$, we may resort to some existing algorithms [25, 26, 34] tailored for nonconvex composite optimization. In particular, we adopt the proximal gradient descent (PGD) method [25] with initialization $\mathbf{x}_0 = \mathbf{x}$ and step size $\alpha > 0$ with the kth iteration being

$$\mathbf{x}_{i,k+1} = \mathrm{prox}_{r/\alpha}\left(\mathbf{x}_{i,k} - \frac{\nabla \overline{m}(\mathbf{x}_{i,k}; \mathbf{x}) + \sigma \left\|\mathbf{x}_{i,k} - \mathbf{x}\right\|^{p-1}(\mathbf{x}_{i,k} - \mathbf{x})}{\alpha}\right)$$

until $\mathbf{x}_{i,k} \approx \mathrm{argmin}_{\mathbf{y}\in\mathbb{R}^d}\ m(\mathbf{y}; \mathbf{x}, \sigma)$.

Solving the auxiliary model: In this framework, we update $\mathbf{z}_{j+1}$ by solving the auxiliary problem as defined in (3.1): $\mathbf{z}_{j+1} = \mathrm{argmin}_{\mathbf{z}\in\mathbb{R}^d}\ \psi_{j+1}(\mathbf{z}, \tau_{j+1})$, where the parameter τ_{j+1} is tuned dynamically in the algorithm. This function is the bridge connecting the two terms on the left- and right-hand sides in (A.11) to establish the iteration bound. In fact, the above subproblem can be solved exactly. To see this, write out the optimality condition to get

$$\nabla l_{j+1}(\mathbf{z}_{j+1}) + \frac{\tau_{j+1}\left\|\mathbf{z}_{j+1} - \bar{\mathbf{x}}_0\right\|^{p-1}(\mathbf{z}_{j+1} - \bar{\mathbf{x}}_0)}{2} = 0,$$

which implies that

$$\left\|\mathbf{z}_{j+1} - \bar{\mathbf{x}}_0\right\| = \left(\frac{2\|\nabla \ell_{j+1}(\mathbf{z}_{j+1})\|}{\tau_{j+1}}\right)^{1/p}.$$

Moreover, we observe that $l_{j+1}(\mathbf{z})$ is a linear function of $\mathbf{z}$, and hence $\nabla \ell_{j+1}(\mathbf{z}_{j+1})$ is independent of $\mathbf{z}_{j+1}$. Consequently, we conclude that

$$\mathbf{z}_{j+1} = \bar{\mathbf{x}}_0 - \left(\frac{2}{\tau_{j+1}}\right)^{1/p} \frac{\nabla l_{j+1}(\mathbf{z}_{j+1})}{\|\nabla l_{j+1}(\mathbf{z}_{j+1})\|^{1-1/p}}.$$

Criterion: The criterion for determining the successful iteration in AAS is

$$\theta(\mathbf{x}_{i+1}, \mathbf{y}_j, \xi_{i+1}) \geq \eta.$$

In particular, for $p \geq 1$ we define $\theta(\mathbf{x}, \mathbf{y}, \xi)$ as

$$\theta(\mathbf{x}, \mathbf{y}, \xi) = \frac{(\mathbf{y} - \mathbf{x})^\top (\nabla f(\mathbf{x}) + \xi)}{\|\mathbf{y} - \mathbf{x}\|^{p+1}}.$$

Output: The output contains the total number of iterations i and the iterate $\mathbf{x}_i$. Note that $\mathbf{x}_i$ is an ε-optimal solution for problem (1.1).

3.1. Iteration complexity of the UAA. In this subsection, we first make the following assumption.

ASSUMPTION 3.2. *Suppose that $\mathbf{x}_0$ is the starting point of our algorithm and that $\mathbf{x}^*$ is an optimal solution of problem (1.1). The level set $\mathcal{L}(x_0, \sigma) := \{x \in \mathbb{R}^d \mid m(\mathbf{x}; \mathbf{x}_0, \sigma) \leq m(\mathbf{x}_0; \mathbf{x}_0, \sigma) = F(\mathbf{x}_0)\}$ of $m(\cdot)$ at $\mathbf{x}_0$ with regularization parameter σ is bounded when $\sigma = \sigma_{\min}$, and that is*

$$\max_{\mathbf{x}\in\mathcal{L}(x_0,\sigma_{\min})} \|\mathbf{x} - \mathbf{x}^*\| \leq D < \infty. \tag{3.4}$$

TABLE 1
Specific choices of $\overline{m}(\mathbf{y};\mathbf{x})$.

Derivative inf.	$\overline{m}(\mathbf{y};\mathbf{x})$	$\bar{\kappa}_p$	κ_p	β_p	ρ_p
up to 1st order	$\tilde{f}_1(\mathbf{y};\mathbf{x})$	0	$\frac{L_1}{2}$	$\frac{L_1}{2}$	L_1
up to 2nd order	$\tilde{f}_2(\mathbf{y};\mathbf{x})$	0	$\frac{L_2}{6}$	$\frac{L_2}{6}$	$\frac{L_2}{2}$
inexact Hessian	$\tilde{f}_1(\mathbf{y};\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathbf{x})^\top H(\mathbf{x})(\mathbf{y}-\mathbf{x})$	$L_1+\kappa$	$\frac{L_2}{6}$	$\frac{L_2+3\kappa}{6}$	$\frac{L_2+2\kappa}{2}$
up to pth order	$\tilde{f}_p(\mathbf{y};\mathbf{x})$	0	$\frac{L_p}{(p+1)!}$	$\frac{L_p}{(p+1)!}$	$\frac{L_p}{p!}$

Now we present the main theoretical results on the iteration complexity of UAA.

THEOREM 3.3. *Let the sequence of iterates* $\{\bar{\mathbf{x}}_j,\ j \geq 0\}$ *be generated by AAS in UAA, and let* $\mathbf{x}^*$ *be an optimal solution for* (1.1). *Denote*

$$C := (p+1)!\left(\frac{2(p+1)\kappa_p + 2\hat{\sigma}_1 + \hat{\sigma}_2}{2(p+1)}D^{p+1} + \bar{\kappa}_p D^p + (\kappa_\theta + \hat{\sigma}_1)(2D)^{p+1}\right),$$

where $\hat{\sigma}_1 := \max\left\{\bar{\sigma}_1, \frac{L_p}{(p-1)!}\right\}$ *and* $\hat{\sigma}_2 := \max\left\{\tau_0, \frac{2^p\gamma_3(\rho_p+\bar{\sigma}_2+\kappa_\theta)^{p+1}p^{p-1}}{\eta^p(p-1)!}\right\}$. *Then it holds that*

$$F(\bar{\mathbf{x}}_j) - F(\mathbf{x}^*) \leq \frac{C}{\Pi_{\ell=1}^{p+1}(j+\ell)},$$

which implies that the total iteration number required to reach the ε*-optimal solution can be bounded by*

$$j \leq 2 + \frac{2}{\log(\gamma_1)}\log\left(\frac{\bar{\sigma}_1}{\sigma_{\min}}\right) + \left\lceil \frac{1}{\log(\gamma_3)}\log\left(\frac{2^p(\rho_p+\bar{\sigma}_2+\kappa_\theta)^{p+1}p^{p-1}}{\eta^p(p-1)!\tau_0}\right)\right\rceil + \left(1 + \frac{2}{\log(\gamma_1)}\log\left(\frac{\bar{\sigma}_2}{\sigma_{\min}}\right)\right)\left[1 + \left(\frac{C}{\varepsilon}\right)^{\frac{1}{p+1}}\right].$$

The proof of the theorem is technically involved and hence postponed to the appendix.

4. Specializations of the UAA. In this section, we provide some concrete choices of $\overline{m}(\mathbf{y};\mathbf{x})$, which leads to different iteration complexities of the corresponding algorithms. To present a holistic picture of the results in this section, in Table 1 we summarize the forms of $\overline{m}(\mathbf{y};\mathbf{x})$ associated with different settings.

4.1. First-order adaptive accelerating method. The most popular choice of $\overline{m}(\mathbf{y};\mathbf{x})$ is the first-order approximation,

$$\overline{m}(\mathbf{y};\mathbf{x}) = \tilde{f}_1(\mathbf{y};\mathbf{x}) = f(\mathbf{x}) + (\mathbf{y}-\mathbf{x})^\top\nabla f(\mathbf{x}).$$

Obviously, it is convex, and by (2.2) and (2.3), (i) and (ii) in Definition 2.3 are satisfied with

$$\kappa_1 = \beta_1 = \frac{L_1}{2}, \quad \rho_1 = L_1, \quad \bar{\kappa}_1 = 0.$$

Moreover, the subproblem becomes $\min_{\mathbf{y}\in\mathbb{R}^d} f(\mathbf{x}) + (\mathbf{y}-\mathbf{x})^\top\nabla f(\mathbf{x}) + \frac{\sigma\|\mathbf{y}-\mathbf{x}\|^2}{2} + r(\mathbf{y})$, which has a closed form solution since $r(\cdot)$ has an easy proximal mapping. Therefore, $\kappa_\theta = 0$, and we have the following iteration bound.

THEOREM 4.1. *Letting* $\overline{m}(\mathbf{y};\mathbf{x}) = \tilde{f}_1(\mathbf{y};\mathbf{x})$ *in* UAA, *we obtain an adaptive accelerating first-order method, and the total iteration number for obtaining an* ϵ*-optimal solution is*

$$2 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_1}{\sigma_{\min}}\right) + \left\lceil \frac{1}{\log(\gamma_3)} \log\left(\frac{2\left(\frac{L_1}{2} + \bar{\sigma}_2\right)^2}{\eta\tau_0}\right)\right\rceil + \left(1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_2}{\sigma_{\min}}\right)\right)\left[1 + \left(\frac{C_1}{\varepsilon}\right)^{\frac{1}{2}}\right],$$

where $C_1 = \frac{2L_1 + 2\bar{\sigma}_1 + \tau_0}{2} D^2$.

4.2. Second-order adaptive accelerating method.

4.2.1. Exact Hessian approximation. The second-order approximation of f under exact Hessian is given by

$$\overline{m}(\mathbf{y};\mathbf{x}) = \tilde{f}_2(\mathbf{y};\mathbf{x}) = f(\mathbf{x}) + (\mathbf{y}-\mathbf{x})^\top \nabla f(\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathbf{x})^\top \nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x}).$$

It is still a convex function. Moreover, by (2.2) and (2.3), (i) and (ii) in Definition 2.3 are satisfied with

$$\kappa_2 = \beta_2 = \frac{L_2}{6}, \quad \rho_2 = \frac{L_2}{2}, \quad \bar{\kappa}_2 = 0.$$

Moreover, since $\nabla\overline{m}(\mathbf{y};\mathbf{x}) = \nabla^2 f(\mathbf{x}) \succeq 0$, $\overline{m}(\mathbf{y};\mathbf{x})$ is a convex function. Therefore, we have the following iteration bound.

THEOREM 4.2. *Letting* $\overline{m}(\mathbf{y};\mathbf{x}) = \tilde{f}_2(\mathbf{y};\mathbf{x})$ *in* UAA, *we obtain an adaptive accelerating cubic regularized Newton's method, and the total iteration number for obtaining an* ϵ*-optimal solution is*

$$2 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_1}{\sigma_{\min}}\right) + \left\lceil \frac{1}{\log(\gamma_3)} \log\left(\frac{4\left(\frac{L_2}{6} + \bar{\sigma}_2 + \kappa_\theta\right)^3}{\eta^2\tau_0}\right)\right\rceil + \left(1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_2}{\sigma_{\min}}\right)\right)\left[1 + \left(\frac{C_2}{\varepsilon}\right)^{\frac{1}{3}}\right],$$

where $C_2 = 6\left(\frac{L_2 + 2\bar{\sigma}_1 + \tau_0}{6} D^3 + 8\kappa_\theta D^3\right)$.

4.2.2. Inexact Hessian approximation. We study the scenario where the Hessian information is possibly unavailable; instead, we can construct an approximation of the Hessian $\nabla^2 f(\mathbf{x}_i)$ by first computing d forward gradient differences at $\mathbf{x}_i$ with a step size $h_i \in \mathbb{R}$,

$$A_i = \left[\frac{\nabla f(\mathbf{x}_i + h_i\mathbf{e}_1) - \nabla f(\mathbf{x}_i)}{h_i}, \ldots, \frac{\nabla f(\mathbf{x}_i + h_i\mathbf{e}_d) - \nabla f(\mathbf{x}_i)}{h_i}\right],$$

symmetrizing the resulting matrix $\widehat{H}(\mathbf{x}_i) = \frac{1}{2}\left(A_i + A_i^\top\right)$, and then further adding a sufficiently large constant multiple of an identity matrix to $\widehat{H}(\mathbf{x}_i)$: $H(\mathbf{x}_i) = \widehat{H}(\mathbf{x}_i) + \kappa_c h_i I$, where $\mathbf{e}_j$ is the jth vector of the canonical basis. It is well known from section 7.1 of [50] that for some constant $\kappa_e > 0$, we have

$$\left\|\widehat{H}(\mathbf{x}_i) - \nabla^2 f(\mathbf{x}_i)\right\| \le \kappa_e h_i.$$

Consequently, it holds that

$$\left\|H(\mathbf{x}_i) - \nabla^2 f(\mathbf{x}_i)\right\| \leq (\kappa_e + \kappa_c)\, h_i.$$

That is to say, the gap between exact and inexact Hessian can be bounded by a multiple of the step size h_i. This, together with Algorithm 4.1 in [17], motivates a procedure for searching a pair of $(h_i, \mathbf{x}_{i+1})$ such that

$$h_i \leq \min\{\kappa_{hs}, \kappa_{hs}\, \|\mathbf{x}_{i+1} - \mathbf{x}_i\|\} \quad \text{for some} \quad \kappa_{hs} > 0. \tag{4.1}$$

This procedure is adapted from Algorithm 2 in the first version of this paper [33] by replacing the early stop criterion $\|\nabla f(\mathbf{x}_{i+1})\| \leq \epsilon$ by $\|\nabla f(\mathbf{x}_{i+1}) + \xi\| \leq \epsilon$ with $\xi \in \partial r\,(\mathbf{x}_{i+1})$ as we consider composite optimization in this paper. Similarly to Lemma 4.1 in [33], we can show that one call of this procedure requires an $O(\log(1/\epsilon))$ number of iterations with n additional gradient computations in each iteration. Since this procedure is needed in both successful and unsuccessful iterations in the main loop, it will add a logarithmic factor of $1/\epsilon$ to the overall iteration complexity of the method. Letting $\kappa = (\kappa_e + \kappa_c)\, \kappa_{hs}$, we conclude that

$$\left\|H(\mathbf{x}_i) - \nabla^2 f(\mathbf{x}_i)\right\| \leq \kappa\, \|\mathbf{x}_{i+1} - \mathbf{x}_i\|\,. \tag{4.2}$$

Therefore, we set

$$\overline{m}(\mathbf{y}; \mathbf{x}) = f(\mathbf{x}) + (\mathbf{y} - \mathbf{x})^\top \nabla f(\mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^\top H(\mathbf{x})(\mathbf{y} - \mathbf{x})$$

as the second-order approximation of g under the inexact Hessian. It follows from (2.2), (2.3), and (4.2) that

$$\begin{aligned}
&|f(\mathbf{x}_{i+1}) - \overline{m}(\mathbf{x}_{i+1}; \mathbf{x}_i)| \\
&\leq \left|\frac{1}{2}(\mathbf{x}_{i+1} - \mathbf{x}_i)^\top \nabla^2 f(\mathbf{x}_i)(\mathbf{x}_{i+1} - \mathbf{x}_i) - \frac{1}{2}(\mathbf{x}_{i+1} - \mathbf{x}_i)^\top H(\mathbf{x}_i)(\mathbf{x}_{i+1} - \mathbf{x}_i)\right| \\
&\quad + |f(\mathbf{x}_{i+1}) - f_2(\mathbf{x}_{i+1}; \mathbf{x}_i)| \\
&\leq \frac{L_2}{6}\|\mathbf{x}_{i+1} - \mathbf{x}_i\|^3 + \frac{\kappa}{2}\|\mathbf{x}_{i+1} - \mathbf{x}_i\|^3 = \frac{L_2 + 3\kappa}{6}\|\mathbf{x}_{i+1} - \mathbf{x}_i\|^3,
\end{aligned}$$

and

$$\begin{aligned}
&|\nabla f(\mathbf{x}_{i+1}) - \nabla \overline{m}(\mathbf{x}_{i+1}; \mathbf{x}_i)| \\
&\leq |\nabla f(\mathbf{x}_{i+1}) - \nabla f_2(\mathbf{x}_{i+1}; \mathbf{x}_i)| + \left|\nabla^2 f(\mathbf{x}_i)(\mathbf{x}_{i+1} - \mathbf{x}_i) - H(\mathbf{x}_i)(\mathbf{x}_{i+1} - \mathbf{x}_i)\right| \\
&\leq \frac{L_2}{2}\|\mathbf{x}_{i+1} - \mathbf{x}_i\|^2 + \kappa\|\mathbf{x}_{i+1} - \mathbf{x}_i\|^2 = \frac{L_2 + 2\kappa}{2}\|\mathbf{x}_{i+1} - \mathbf{x}_i\|^2.
\end{aligned}$$

Moreover, since $\nabla f(\mathbf{x})$ is Lipschitz continuous with $L_1 > 0$, we have

$$\left\|\nabla^2 f(\mathbf{x})\right\| \leq L_1, \quad \mathbf{x} \in \mathsf{dom}(F),$$

and thus $\|\widehat{H}(\mathbf{x})\| \leq L_1$, which further implies $\|H(\mathbf{x})\| \leq L_1 + \kappa_c h_i \leq L_1 + \kappa_c \kappa_{hs} \leq L_1 + \kappa$. As a result,

$$\begin{aligned}
&|f(\mathbf{y}) - \overline{m}(\mathbf{y}; \mathbf{x})| \\
&\leq |f(\mathbf{y}) - f_2(\mathbf{y}; \mathbf{x})| + \left|\frac{1}{2}(\mathbf{y} - \mathbf{x})^\top \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x}) - \frac{1}{2}(\mathbf{y} - \mathbf{x})^\top H(\mathbf{x})(\mathbf{y} - \mathbf{x})\right| \\
&\leq \frac{L_2}{6}\|\mathbf{y} - \mathbf{x}\|^3 + \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|^2 \left(\|\nabla^2 f(\mathbf{x})\| + \|H(\mathbf{x})\|\right) \\
&\leq \frac{L_2}{6}\|\mathbf{y} - \mathbf{x}\|^3 + (L_1 + \kappa)\|\mathbf{y} - \mathbf{x}\|^2.
\end{aligned}$$

Finally, since f is convex and κ_c is sufficiently large such that $\kappa_c \geq \kappa_e$, we have

$$H(\mathbf{x}_i) \succeq \nabla^2 f(\mathbf{x}_i) - \kappa_e h_i I + \kappa_c h_i I \succeq 0,$$

and $\overline{m}(\mathbf{y};\mathbf{x}_i)$ is convex as well. Therefore, all three conditions in Definition 2.4 are satisfied with

$$\beta_2 = \frac{L_2 + 3\kappa}{6}, \qquad \rho_2 = \frac{L_2 + 2\kappa}{2}, \quad \kappa_2 = \frac{L_2}{6}, \quad \bar{\kappa}_2 = L_1 + \kappa.$$

Therefore, we have the following iteration bound.

THEOREM 4.3. *Letting* $\overline{m}(\mathbf{y};\mathbf{x}) = f(\mathbf{x}) + (\mathbf{y}-\mathbf{x})^\top \nabla f(\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathbf{x})^\top H(\mathbf{x})(\mathbf{y}-\mathbf{x})$ *in* UAA*, we obtain an adaptive accelerating cubic regularized approximate Newton's method, and the total iteration number for obtaining an* ϵ*-optimal solution is*

$$2 + \frac{2\log\left(\frac{\bar{\sigma}_1}{\sigma_{\min}}\right)}{\log(\gamma_1)} + \left\lceil \frac{1}{\log(\gamma_3)} \log\left(\frac{4\left(\frac{L_2+2\kappa}{2} + \bar{\sigma}_2 + \kappa_\theta\right)^3}{\eta^2 \tau_0} \right) \right\rceil$$
$$+ \left(1 + \frac{2}{\log(\gamma_1)} \log\left(\frac{\bar{\sigma}_2}{\sigma_{\min}}\right)\right) \left[1 + \left(\frac{\bar{C}_2}{\varepsilon}\right)^{\frac{1}{3}}\right],$$

where $\bar{C}_2 = 6\left(\frac{L_2+2\bar{\sigma}_1+\tau_0}{6} D^3 + L_1 D^2 + 8\kappa_\theta D^3\right)$.

4.3. High-order adaptive acceleration method. To utilize high-order information, we let

$$\overline{m}(\mathbf{y};\mathbf{x}) = \tilde{f}_p(\mathbf{y};\mathbf{x}).$$

Then by invoking (2.2) and (2.3), (i) and (ii) in Definition 2.4 are satisfied with

$$\kappa_p = \beta_p = \frac{L_p}{(p+1)!}, \quad \beta_p = \frac{L_p}{p!}, \quad \bar{\kappa}_p = 0.$$

Unfortunately, $\overline{m}(\mathbf{y};\mathbf{x})$ is not necessarily convex in this case. According to Theorem 1 in [49],

$$\overline{m}(\mathbf{y};\mathbf{x}) + \frac{\sigma \|\mathbf{y}-\mathbf{x}\|^{p+1}}{p+1}$$

is a convex function when $\sigma \geq \frac{L_p}{(p-1)!}$. However, the choice of σ is dependent on the problem parameter L_p. Moreover, checking the convexity of a polynomial function is NP hard in general [1], and it remains a challenging task even when the polynomial is well structured (for instance, a sum of squares [32]). Fortunately, as shown in the proof of Theorem A.7, only the convexity of

$$m(\mathbf{y};\mathbf{x}_0,\sigma) = \overline{m}(\mathbf{y};\mathbf{x}_0) + \frac{\sigma \|\mathbf{y}-\mathbf{x}_0\|^{p+1}}{p+1} + r(\mathbf{y})$$

at point $\bar{\mathbf{x}}_0$ suffices to get an upper bound of $\psi_0(\mathbf{z},\tau_0)$, where $\bar{\mathbf{x}}_0$ is the output solution of SAS. Note that

$$\nabla^2 \left(\frac{\sigma \|\mathbf{y}-\mathbf{x}\|^{p+1}}{p+1} \right) = \sigma(p-1)\|\mathbf{y}-\mathbf{x}\|^{p-3}(\mathbf{y}-\mathbf{x})(\mathbf{y}-\mathbf{x})^\top + \sigma\|\mathbf{y}-\mathbf{x}\|^{p-1} I$$
$$\succeq \sigma\|\mathbf{y}-\mathbf{x}\|^{p-1} I. \tag{4.3}$$

Therefore, $m(\mathbf{y};\mathbf{x}_0,\sigma)$ is convex at $\bar{\mathbf{x}}_0$ as long as $\sigma \geq -\lambda_{\min}\left(\nabla^2\overline{m}(\bar{\mathbf{x}}_0;\mathbf{x}_0)\right)/\|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|^{p-1}$. Recall that σ^{SAS} is the adaptive regularizing parameter associated with $(\bar{\mathbf{x}}_0;\mathbf{x})$. Then we can let the input adaptive parameter of AAS be

$$\sigma_0^{\mathsf{AAS}} = \max\{\sigma^{\mathsf{SAS}}, -\lambda_{\min}\left(\nabla^2\overline{m}(\bar{\mathbf{x}}_0;\mathbf{x}_0)\right)/\|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|^{p-1}\} \tag{4.4}$$

to guarantee the convexity of $m(\mathbf{y};\mathbf{x}_0,\sigma_0^{\mathsf{AAS}})$ at $\bar{\mathbf{x}}_0$. Moreover, we have that $F(\bar{\mathbf{x}}_0) < m(\bar{\mathbf{x}}_0;\mathbf{x}_0,\sigma^{\mathsf{SAS}}) \leq m(\bar{\mathbf{x}}_0;\mathbf{x}_0,\sigma_0^{\mathsf{AAS}})$, and so $\bar{\mathbf{x}}_0$ is still a successful iterate in SAS. In practice, we may further add a small positive number to σ_0 to get rid of ill-conditioning caused by the numerical error when computing $-\lambda_{\min}\left(\nabla^2\overline{m}(\bar{\mathbf{x}}_0;\mathbf{x})\right)/\|\bar{\mathbf{x}}_0 - \mathbf{x}\|^{p-1}$. Finally, we arrive at the following iteration bound.

THEOREM 4.4. *Letting $\overline{m}(\mathbf{y};\mathbf{x}) = \tilde{f}_p(\mathbf{y};\mathbf{x})$ in UAA, we obtain an adaptive accelerating pth-order method, and the total iteration number for obtaining an ϵ-optimal solution is*

$$2 + \frac{2}{\log(\gamma_1)}\log\left(\frac{\bar{\sigma}_1}{\sigma_{\min}}\right) + \left\lceil \frac{1}{\log(\gamma_3)}\log\left(\frac{2\left(\frac{L_p}{(p+1)!} + \bar{\sigma}_2 + \kappa_\theta\right)^{p+1} p^{p-1}}{\eta^p (p-1)!\tau_0}\right)\right\rceil$$
$$+ \left(1 + \frac{2}{\log(\gamma_1)}\log\left(\frac{\bar{\sigma}_2}{\sigma_{\min}}\right)\right)\left[1 + \left(\frac{C_p}{\varepsilon}\right)^{\frac{1}{p+1}}\right],$$

where

$$C_p = (p+1)!\left(\frac{\frac{2L_p}{p!} + 2\hat{\sigma}_1 + \hat{\sigma}_2}{2(p+1)}D^{p+1} + (\kappa_\theta + \hat{\sigma}_1)(2D)^{p+1}\right).$$

5. Numerical experiments. In this section, we present the results of some numerical experiments for solving the ℓ_1-/ℓ_2-regularized logistic regression problems. The reason for this choice is that the logistic loss function is known to be convex, and the corresponding ℓ_1-/ℓ_2-regularized problems are convex as well. Thus, the proposed methods are directly applicable. Moreover, these two problems are common in testing the performance of various second-order methods in the literature; see [28, 52] for ℓ_1-regularized problems and [5, 6] for ℓ_2-regularized problems. All experiments are conducted on a MacBook Pro with Mac OS High Sierra 10.13.6, an Intel i5 2.6GHz CPU, and 16GB memory.

5.1. ℓ_2-regularized logistic regression problem. We first test the performance of the algorithms by evaluating the following ℓ_2-regularized logistic regression problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\ln\left(1 + \exp\left(-b_i\cdot\mathbf{a}_i^\top\mathbf{x}\right)\right) + \frac{\lambda}{2}\|\mathbf{x}\|^2, \tag{5.1}$$

where $(\mathbf{a}_i, b_i)_{i=1}^n$ are the samples in the data set, and the regularization parameter is set as $\lambda = 10^{-5}$. To observe the acceleration, we make the starting point randomly generated from a Gaussian random variable with zero mean and a large variance (say 5000). In this way, initial solutions are likely to be far away from the global solution.

We implement a variant of Algorithm 3.1 with cubic regularization, referred to as *adaptively accelerated cubic regularized* (AARC) Newton's method. In this variant we set $\sigma_0 = \tau_0 = 1$, $\sigma_{\min} = 10^{-16}$, $\kappa_\theta = 0.1$, $\gamma_1 = \gamma_2 = \gamma_3 = 2$, and $\eta = 0.01$. We first run Algorithm 3.1 and then switch to the adaptive cubic regularization (ARC) phase

TABLE 2
Statistics of datasets for ℓ_2-regularized logistic regression.

Dataset	n	d
a9a	32,561	123
phishing	11,055	68
sonar	208	60
svmguide3	1243	22
w8a	49,749	300
SUSY	5,000,000	18

of Newton's method in [14, 15] when the iterates are close to the global optimum. In particular, the switch is activated after 10 successful iterations of AAS are performed and the progress made by each iteration is small, i.e., $\frac{|f(\mathbf{x}_{k+1})-f(\mathbf{x}_k)|}{|f(\mathbf{x}_k)|} \leq 0.1$. The final stopping criterion is set to be $\|\nabla f(\mathbf{x})\| \leq 10^{-9}$ after switching to the ARC phase. In the implementation, we apply the so-called Lanczos process to approximately solve the subproblem $\min_{\mathbf{y}\in\mathbb{R}^d} m(\mathbf{y}; \mathbf{x}_i, \sigma_i)$. In addition to (3.2), the approximate solution $\mathbf{s}$ also is made to satisfy

$$(\mathbf{y}-\mathbf{x}_i)^\top \nabla f(\mathbf{x}_i) + (\mathbf{y}-\mathbf{x}_i)^\top \nabla^2 f(\mathbf{x}_i)(\mathbf{y}-\mathbf{x}_i) + \sigma \|\mathbf{y}-\mathbf{x}_i\|^3 = 0 \tag{5.2}$$

for given $\mathbf{x}_i$ and σ_i. Note that (5.2) is a consequence of the first-order necessary condition, and, as shown in Lemma 3.2 of [14], the global minimizer of $m(\mathbf{y}; \mathbf{x}_i, \sigma_i)$ when restricted to a Krylov subspace

$$\mathcal{K} := \operatorname{span}\{\nabla f(\mathbf{x}_i), \nabla^2 f(\mathbf{x}_i)\nabla f(\mathbf{x}_i), \left(\nabla^2 f(\mathbf{x}_i)\right)^2 \nabla f(\mathbf{x}_i), \ldots\}$$

satisfies (5.2) independently of the subspace dimension. Minimizing $m(\mathbf{y}; \mathbf{x}_i, \sigma_i)$ in the Krylov subspace also is computationally favorable, as it can be done at the cost of $O(d)$ involving only factorizing a tridiagonal matrix. Thus, the associated approximate solution can be found through the so-called Lanczos process, where the dimension of $\mathcal{K}$ is gradually increased and an orthogonal basis of each subspace $\mathcal{K}$ is built up, which typically involves one matrix-vector product. Condition (3.2) can be used as the termination criterion for the Lanczos process to find a suitable trial step before the dimension of $\mathcal{K}$ approaches d.

We compare the new AARC method with four other methods including the ARC, the trust region method (TR), the limited memory Broyden–Fletcher–Goldfarb–Shanno method (L-BFGS), and the adaptive gradient descent method (AGD). We adopt the implementation of ARC and TR in the public package[1] with the default parameters, but we use the full rather than the subsampled batch of the component functions in ARC, and the upper bound on the radius of the trust region in TR is set to be 10^4. To implement the L-BFGS method, we use the Wolfe conditions to perform the line search and set the descent parameters in Armijo rule, the curvature condition, and the memory size as 0.01, 0.9, and 50, respectively. AGD is implemented based on AdaGrad in [23]. The experiments are conducted on six LIBSVM sets[2] for binary classification, and the summaries of those datasets are shown in Table 2.

[1] https://github.com/dalab/subsampled_cubic_regularization
[2] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

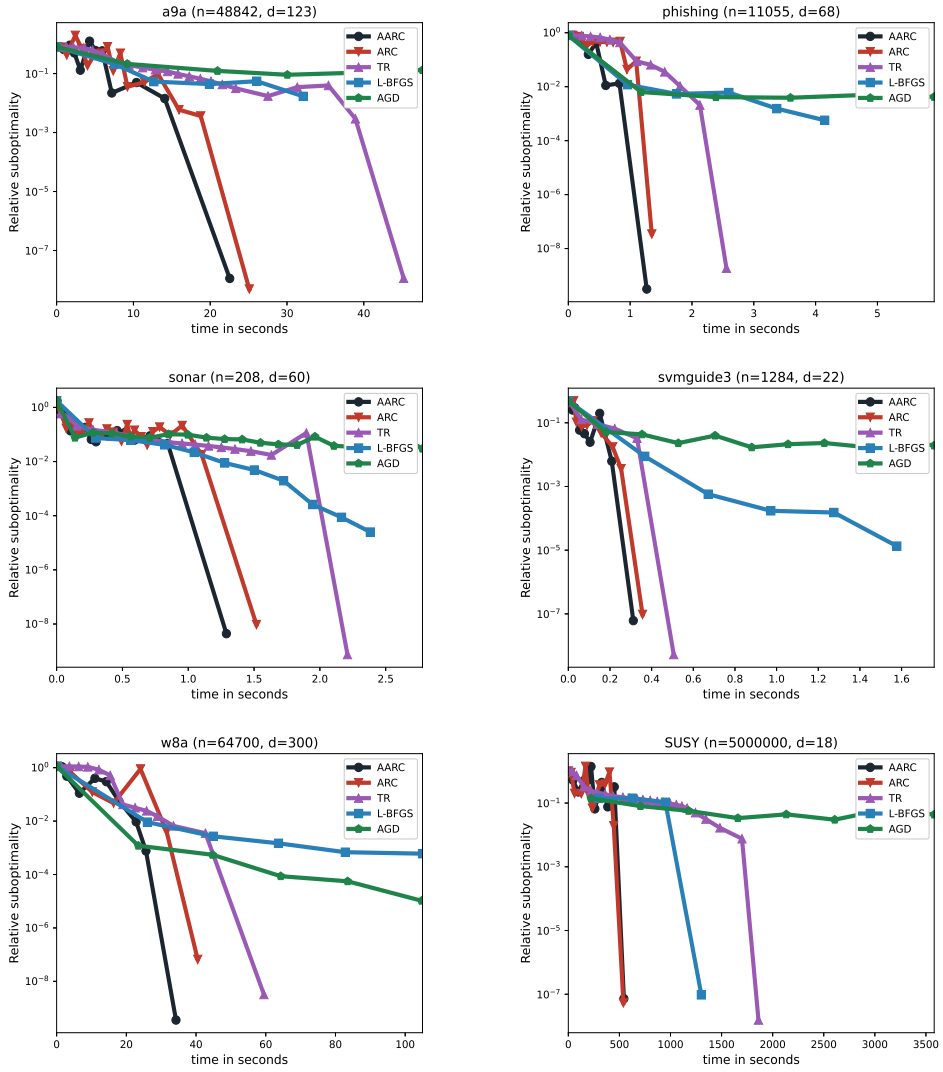

FIG. 1. *Performance of AARC and all benchmark methods on the task of ℓ_2-regularized logistic regression (loss versus time).*

The results in Figures 1 and 2 confirm that AARC indeed accelerates ARC—especially when the current iterate has not yet entered the local region of quadratic convergence. Furthermore, AARC outperforms other methods in both computational time and iteration counts in the given datasets. Compared to TR with a local constrained quadratic model, AARC achieves more progress and cheaper per-iteration cost at each iteration because of the advantages of the unconstrained local cubic approximation model (3.1) and the flexible stopping criterion (3.2); compared to L-BFGS, AARC suffers from relatively higher per-iteration cost, but its solution can achieve higher accuracy.

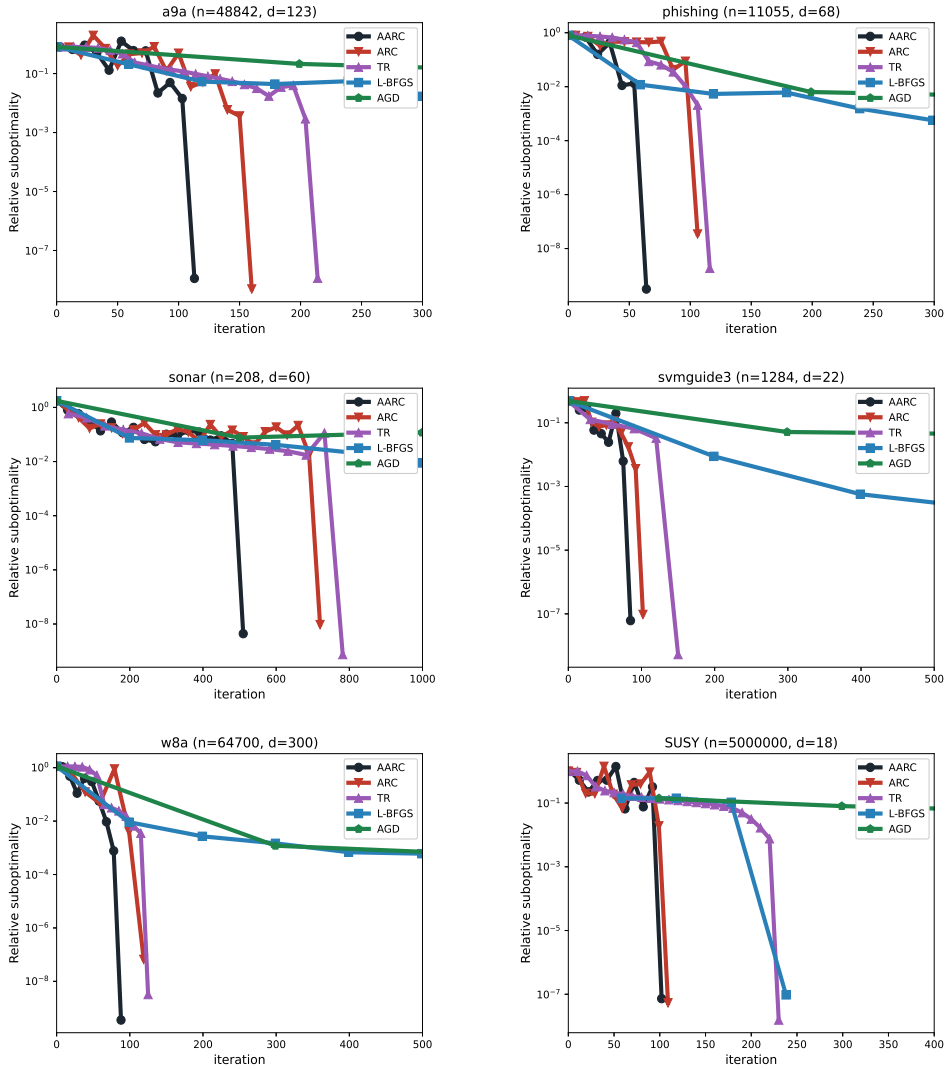

FIG. 2. *Performance of AARC and all benchmark methods on the task of ℓ_2-regularized logistic regression (loss versus iterations).*

5.2. ℓ_1-regularized logistic regression problem. Now we test the algorithms on the following ℓ_1-regularized logistic regression problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \log\left(1+\exp(-y_i \mathbf{w}_i^\top \mathbf{x})\right) + \lambda \|\mathbf{x}\|_1 , \tag{5.3}$$

where $\{(\mathbf{w}_i, y_i)\}_{i=1}^n$ is a collection of data samples, with $y_i \in \{-1, 1\}$ being the label. The regularization term $\|\mathbf{x}\|_1$ promotes sparse solutions, and $\lambda > 0$ balances sparsity with goodness-of-fit and generalization. In addition, λ was chosen by LIBLINEAR with fivefold cross validation. The experiments are conducted on three datasets, all which

TABLE 3
Statistics of datasets for ℓ_1-regularized logistic regression.

Name	Description	n	d	Scaled interval	λ
a9a	UCI adult	48842	123	$[0,1]$	4.5e-03
covetype	forest covetype	581012	54	$[0,1]$	2.6e-03
w8a	-	64700	300	$[0,1]$	7.0e-04

come from LIBSVM,[3] and the summaries of those datasets are shown in Table 3.

We first test how the inexactness of the Hessian matrix affects the performance of AARC on ℓ_1-regularized logistic regression problem (5.3). In particular, we implement inexact AARC with different values of κ_{hs} in (4.1) and set the step size to construct the approximated Hessian as $h_i = \min\{\kappa_{hs}, \kappa_{hs}\|\mathbf{x}_i - \mathbf{x}_{i-1}\|\}$, where $\|\mathbf{x}_i - \mathbf{x}_{i-1}\|$ is the difference of the last two consecutive iterates. We plot relative suboptimality versus iteration counts on all datasets in Figure 3, but we do not plot the figures regarding the run-time, as all the methods in Figure 3 solve similar subproblems, and the run-time is proportional to the iteration count. Figure 3 indicates that inexact AARC works well in general, and the corresponding iteration complexity decreases as the value of κ_{hs} decreases, which makes sense and implies that more accuracy of the Hessian leads to faster convergence of the proposed algorithm. Note that we do not choose a very small value of κ_{hs} because if we do, then the corresponding curves will be very close to those of exact AARC, making it hard to distinguish the two curves.

We compare the AARC with Nesterov's accelerated gradient method (Nesterov83) (adapted for composite optimization), the fast iterative shrinkage thresholding algorithm (FISTA) [4], and the accelerated regularized Newton methods proposed by Grapiglia and Nesterov (GN) [29] on ℓ_1-regularized logistic regression problem (5.3). We use the TFOCS[4] implementation with default parameter settings for Nesterov83 and FISTA. Note that Nesterov83 and FISTA have different coefficients on the momentum term, and their numerical performances behave differently as shown in Figure 4. For AARC, we use the same setting as that for the ℓ_2-regularized logistic regression problem, e.g., $\sigma_0 = 1$, $\sigma_{\min} = 10^{-16}$, $\kappa_\theta = 0.1$, $\gamma_1 = \gamma_2 = \gamma_3 = 2$, and $\eta = 0.01$. The difference is that we adopt FISTA [4] to solve the subproblem in AARC, as the subproblem itself is a convex composite optimization problem. The maximum number of iterations for solving those subproblems is 500, and the parameter setting for FISTA is default. Finally, we manage to implement the accelerated regularized Newton methods (GN) in [29] with two minor modifications: (i) the subproblem in GN is approximately solved with the stopping criterion (3.2), where we set $\kappa_\theta = 10^{-20}$ such that the subproblem is almost solved exactly; and (ii) the nonsmooth objective function in the auxiliary function in GN is replaced by its subgradient to avoid computing another proximal mapping by iterative algorithms for computational efficiency; otherwise, the per-iteration cost would be doubled.

We plot relative suboptimality versus iteration counts as well as relative suboptimality versus time on all datasets in Figure 4. It is clear in Figure 4 that our method consistently outperforms Nesterov83 and FISTA in terms of the number of iterations and the overall computational time, although the subproblem in AARC does not have a closed-form solution and is much more time-consuming to solve, which is in contrast with the subproblems of accelerated first-order methods. Compared to the accelerated second-order method GN, AARC has slightly smaller iteration counts. This is

[3]The collection is available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets.
[4]http://cvxr.com/tfocs/

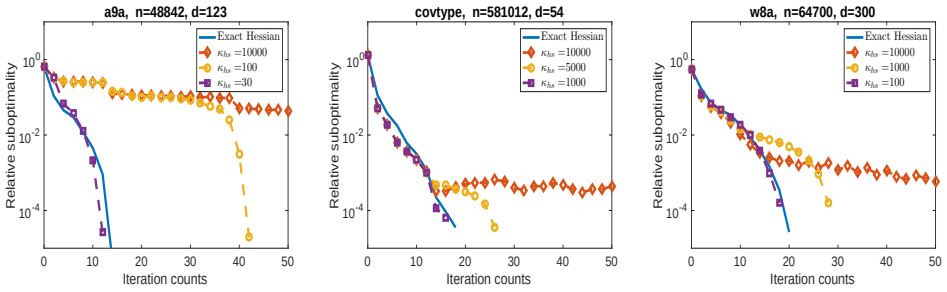

FIG. 3. *Iteration counts of AARC with Inexact Hessians on ℓ_1-regularized logistic regression.*

possibly due to the dynamic adjustment of the adaptive parameter τ_{j+1} of the auxiliary function $\psi_{j+1}(\mathbf{z}, \tau_{j+1})$ in our AAS subroutine, while a similar parameter in GN is updated by solving a certain univariate polynomial equation. Besides the slight difference in the iteration counts, GN is also more time-consuming per iteration as it needs to solve the subproblem more accurately.

Appendix A. Technical proofs in section 3. First, we bound the total number of iterations in SAS, denoted as T_1, and the total number of iterations in AAS, denoted as T_2.

LEMMA A.1. *Let $\bar{\sigma}_1 = \max\{\sigma_0,\ (p+1)\gamma_2\beta_p\}$, where β_p is defined as in (2.5). We have $T_1 \le 1 + \frac{2}{\log(\gamma_1)}\log(\frac{\bar{\sigma}_1}{\sigma_{\min}})$.*

The lemma above is motivated by Theorem 2.1 in [15], and the proof is omitted as it is mostly identical to the one in [15].

LEMMA A.2. *Let $\mathcal{S}$ be the set of successful iteration counts in the total iteration count of* AAS *and let*

$$\bar{\sigma}_2 = \max\{\bar{\sigma}_1,\ \gamma_2(\kappa_\theta + \rho_p + \eta), -\lambda_{\min}(\nabla^2\overline{m}(\bar{\mathbf{x}}_0; \mathbf{x}_0))/\|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|^{p-1}\},$$

where κ_θ and ρ_p are defined as in (2.6) and (3.2), respectively; $\mathbf{x}_0$ is the initial point of SAS*; and $\bar{\mathbf{x}}_0$ is the output of* SAS*. Then we have $\sigma_{\min} \le \sigma_i \le \bar{\sigma}_2$ for all i in* AAS*, and $T_2 \le (1 + \frac{2}{\log(\gamma_1)}\log(\frac{\bar{\sigma}_2}{\sigma_{\min}}))|\mathcal{S}|$.*

Proof. We observe that

$$\begin{aligned}
(\mathrm{A.1}) \qquad & (\mathbf{y}_j - \mathbf{x}_{i+1})^\top \left(\nabla\overline{m}(\mathbf{x}_{i+1}; \mathbf{y}_j) + \sigma_i \|\mathbf{x}_{i+1} - \mathbf{y}_j\|^{p-1}(\mathbf{x}_{i+1} - \mathbf{y}_j) + \xi_{i+1}\right) \\
& \ge -\|\mathbf{y}_j - \mathbf{x}_{i+1}\| \cdot \left\|\nabla\overline{m}(\mathbf{x}_{i+1}; \mathbf{y}_j) + \sigma_i \|\mathbf{x}_{i+1} - \mathbf{y}_j\|^{p-1}(\mathbf{x}_{i+1} - \mathbf{y}_j) + \xi_{i+1}\right\| \\
& \overset{(3.2)}{\ge} -\kappa_\theta \|\mathbf{x}_{i+1} - \mathbf{y}_j\|^{p+1}.
\end{aligned}$$

Consequently, we conclude that

$$\begin{aligned}
\theta(\mathbf{x}_{i+1}, \mathbf{y}_j, \xi_{i+1}) &= \frac{(\mathbf{y}_j - \mathbf{x}_{i+1})^\top (\nabla f(\mathbf{x}_{i+1}) + \xi_{i+1})}{\|\mathbf{y}_j - \mathbf{x}_{i+1}\|^{p+1}} \\
&= \frac{(\mathbf{y}_j - \mathbf{x}_{i+1})^\top \left(\nabla f(\mathbf{x}_{i+1}) - \nabla\overline{m}(\mathbf{x}_{i+1}; \mathbf{y}_j) - \sigma_i \|\mathbf{x}_{i+1} - \mathbf{y}_j\|^{p-1}(\mathbf{x}_{i+1} - \mathbf{y}_j)\right)}{\|\mathbf{y}_j - \mathbf{x}_{i+1}\|^{p+1}}
\end{aligned}$$

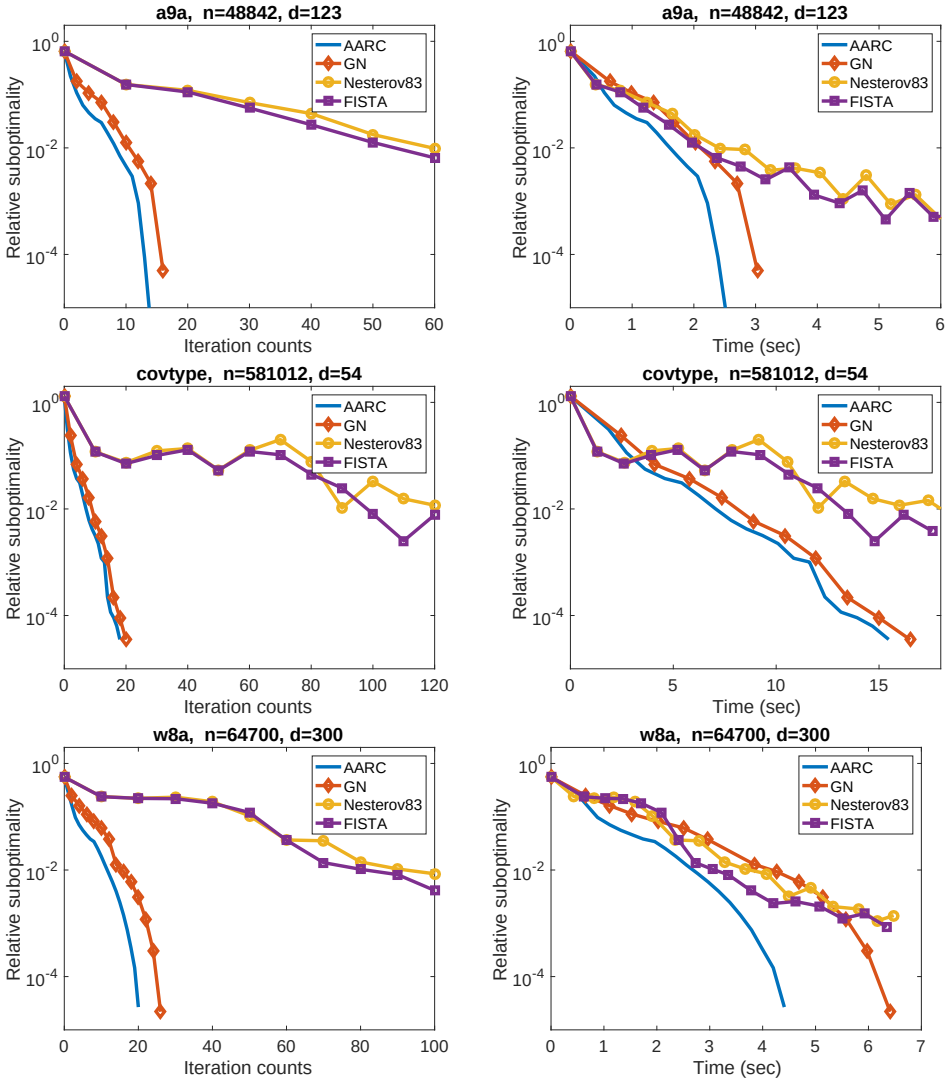

FIG. 4. *Iteration counts and computational times of the four methods on ℓ_1-regularized logistic regression.*

$$
\begin{aligned}
&+\frac{(\mathbf{y}_j-\mathbf{x}_{i+1})^\top\left(\nabla\overline{m}(\mathbf{x}_{i+1};\mathbf{y}_j)+\sigma_i\|\mathbf{x}_{i+1}-\mathbf{y}_j\|^{p-1}(\mathbf{x}_{i+1}-\mathbf{y}_j)+\xi_{i+1}\right)}{\|\mathbf{y}_j-\mathbf{x}_{i+1}\|^{p+1}}\\
&\overset{(A.1)}{\geq}\ \sigma_i-\kappa_\theta+\frac{(\mathbf{y}_j-\mathbf{x}_{i+1})^\top(\nabla f(\mathbf{x}_{i+1})-\nabla\overline{m}(\mathbf{x}_{i+1};\mathbf{y}_j))}{\|\mathbf{y}_j-\mathbf{x}_{i+1}\|^{p+1}}\\
&\geq \sigma_i-\kappa_\theta-\frac{\|\mathbf{y}_j-\mathbf{x}_{i+1}\|\,\|\nabla f(\mathbf{x}_{i+1})-\nabla\overline{m}(\mathbf{x}_{i+1};\mathbf{y}_j)\|}{\|\mathbf{y}_j-\mathbf{x}_{i+1}\|^{p+1}}\\
&\overset{(2.6)}{\geq}\ \sigma_i-\kappa_\theta-\frac{\rho_p\|\mathbf{y}_j-\mathbf{x}_{i+1}\|^{p+1}}{\|\mathbf{y}_j-\mathbf{x}_{i+1}\|^{p+1}}\\
&=\sigma_i-\kappa_\theta-\rho_p,
\end{aligned}
$$

and $\sigma_i \geq \kappa_\theta + \rho_p + \eta \implies \theta(\mathbf{x}_{i+1}, \mathbf{y}_j, \xi_{i+1}) \geq \eta$. This implies that

$$\sigma_{i+1} \leq \sigma_i \leq \gamma_2 \sigma_{i-1} \leq \gamma_2 (\kappa_\theta + \rho_p + \eta) \quad \forall\, i \in \mathcal{S}.$$

Therefore, σ_i can be upper bounded by $\bar{\sigma}_2$ and lower bounded by $\sigma_{\min}$ in AAS. In addition, $\gamma_1 \sigma_i \leq \sigma_{i+1}$ for any $i \notin \mathcal{S}$. Therefore, we have

$$\frac{\bar{\sigma}_2}{\sigma_{\min}} \geq \frac{\sigma_{T_2}}{\sigma_0} = \prod_{i \in \mathcal{S}} \frac{\sigma_{i+1}}{\sigma_i} \cdot \prod_{i \notin \mathcal{S}} \frac{\sigma_{i+1}}{\sigma_i} \geq \gamma_1^{T_2 - |\mathcal{S}|} \left(\frac{\sigma_{\min}}{\bar{\sigma}_2} \right)^{|\mathcal{S}|},$$

which further implies an upper bound for T_2, completing the proof. ∎

Next we proceed to bound the total number of times that we update the regularization parameter τ in the auxiliary model, which is denoted as T_3. This requires three key technical lemmas, which are presented below.

LEMMA A.3 (Lemma 2 in [46]). *For any $\mathbf{g} \in \mathbb{R}^d$, $\mathbf{s} \in \mathbb{R}^d$, and integer $q \geq 2$, we have*

$$\mathbf{g}^\top \mathbf{s} + \frac{\sigma \|\mathbf{s}\|^q}{q} \geq -\frac{q-1}{q} \left(\frac{\|\mathbf{g}\|^q}{\sigma} \right)^{\frac{1}{q-1}}. \tag{A.2}$$

LEMMA A.4. *For the minimizer of $\psi_j(\mathbf{z}, \tau_j)$, i.e., $\mathbf{z}_j = \operatorname{argmin}_{\mathbf{z} \in \mathbb{R}^d} \psi_j(\mathbf{z}, \tau_j)$, we have*

$$\psi_j(\mathbf{z}, \tau_j) - \psi_j(\mathbf{z}_j, \tau_j) \geq \frac{\tau_j}{2^p} \frac{\|\mathbf{z} - \mathbf{z}_j\|^{p+1}}{p+1}.$$

Proof. Recall that $\psi_j(\mathbf{z}, \tau_j)$ is the sum of a linear function and a $(p+1)$th powered regularization function: $\psi_j(\mathbf{z}, \tau_j) = l_j(\mathbf{z}) + \tau_j R(\mathbf{z}) = l_j(\mathbf{z}) + \frac{\tau_j}{2} \frac{\|\mathbf{z} - \mathbf{x}_0\|^{p+1}}{p+1}$. Thus, we have

$$\begin{aligned}
&\psi_j(\mathbf{z}, \tau_j) - \psi_j(\mathbf{z}_j, \tau_j) \\
&= (\mathbf{z} - \mathbf{z}_j)^\top \nabla l_j(\mathbf{z}_j) + \tau_j (R(\mathbf{z}) - R(\mathbf{z}_j)) \\
&\geq (\mathbf{z} - \mathbf{z}_j)^\top \nabla l_j(\mathbf{z}_j) + \tau_j (\mathbf{z} - \mathbf{z}_j)^\top \nabla R(\mathbf{z}_j) + \frac{\tau_j}{2^p} \frac{\|\mathbf{z} - \mathbf{z}_j\|^{p+1}}{p+1},
\end{aligned}$$

where the inequality is due to Lemma 4 in [46]. Since $\mathbf{z}_j$ is the minimizer of $\psi_j(\mathbf{z}, \tau_j)$ over $\mathbf{z} \in \mathbb{R}^d$, we have $\nabla l_j(\mathbf{z}_j) + \tau_j \nabla R(\mathbf{z}_j) = \nabla \psi_j(\mathbf{z}_j, \tau_j) = 0$. Combining the above two formulas yields the desired result. ∎

LEMMA A.5. *For any $j \geq 0$ in AAS, we have*

$$\left\| \nabla f(\bar{\mathbf{x}}_{j+1}) + \bar{\xi}_{j+1} \right\| \leq (\rho_p + \bar{\sigma}_2 + \kappa_\theta) \|\bar{\mathbf{x}}_{j+1} - \mathbf{y}_j\|^p.$$

Proof. We observe that

$$\begin{aligned}
&\left\| \nabla \overline{m}(\bar{\mathbf{x}}_{j+1}; \mathbf{y}_j) + \bar{\xi}_{j+1} \right\| \\
&\leq \left\| \nabla \overline{m}(\bar{\mathbf{x}}_{j+1}; \mathbf{y}_j) + \sigma_i \|\bar{\mathbf{x}}_{j+1} - \mathbf{y}_j\|^{p-1} (\bar{\mathbf{x}}_{j+1} - \mathbf{y}_j) + \bar{\xi}_{j+1} \right\| + \sigma_i \|\bar{\mathbf{x}}_{j+1} - \mathbf{y}_j\|^p \\
&\overset{(3.2)}{\leq} \kappa_\theta \|\bar{\mathbf{x}}_{j+1} - \mathbf{y}_j\|^p + \sigma_i \|\bar{\mathbf{x}}_{j+1} - \mathbf{y}_j\|^p \\
&\overset{\text{Lemma A.2}}{\leq} (\kappa_\theta + \bar{\sigma}_2) \|\bar{\mathbf{x}}_{j+1} - \mathbf{y}_j\|^p.
\end{aligned} \tag{A.3}$$

Therefore,

$$\begin{aligned}\left\|\nabla f(\bar{\mathbf{x}}_{j+1})+\bar{\xi}_{j+1}\right\| &\leq \|\nabla f(\bar{\mathbf{x}}_{j+1})-\nabla \overline{m}(\mathbf{x}_{j+1};\mathbf{y}_j)\|+\left\|\nabla \overline{m}(\bar{\mathbf{x}}_{j+1};\mathbf{y}_j)+\bar{\xi}_{j+1}\right\| \\ &\overset{(2.6)}{\leq} \rho_p\|\bar{\mathbf{x}}_{j+1}-\mathbf{y}_j\|^p+(\kappa_\theta+\bar{\sigma}_2)\|\bar{\mathbf{x}}_{j+1}-\mathbf{y}_j\|^p \\ &= (\rho_p+\bar{\sigma}_2+\kappa_\theta)\|\bar{\mathbf{x}}_{j+1}-\mathbf{y}_j\|^p.\end{aligned}$$

We remark that the above result is motivated from Lemma 5.2 in [15], which originally worked for cubic regularized methods with a smooth objective function. Next, we bound T_3, the total number of times we update τ in the auxiliary model.

LEMMA A.6. *For any successful iteration $j \geq 0$ in* AAS*, we have*

$$\psi_j(\mathbf{z}_j,\tau_j) \geq \frac{\Pi_{\ell=1}^{p+1}(j+\ell)}{(p+1)!}F(\bar{\mathbf{x}}_j)$$

provided that $\tau_j \geq \frac{2^p(\rho_p+\bar{\sigma}_2+\kappa_\theta)^{p+1}p^{p-1}}{\eta^p(p-1)!} > 0$. *As a consequence,*

$$T_3 \leq 1+\left\lceil \frac{1}{\log(\gamma_3)}\log\left(\frac{2^p(\rho_p+\bar{\sigma}_2+\kappa_\theta)^{p+1}p^{p-1}}{\eta^p(p-1)!\tau_0}\right)\right\rceil.$$

Proof. We prove this by induction. First, the base case of $j=0$ holds true due to the fact that $\psi_0(\mathbf{z}_0,\tau_0)=\min_{\mathbf{z}\in\mathbb{R}^d}F(\bar{\mathbf{x}}_0)+\frac{\tau_0\|\mathbf{z}-\bar{\mathbf{x}}_0\|^{p+1}}{2(p+1)}=F(\bar{\mathbf{x}}_0)$. Then, we assume the result holds for some $j=j_0$. It remains to prove the result for the case $j=j_0+1$. By the induction hypothesis and Lemma A.4, we have

$$\begin{aligned}\psi_{j_0}(\mathbf{z},\tau_{j_0}) &\geq \psi_{j_0}(\mathbf{z}_{j_0},\tau_{j_0})+\frac{\tau_{j_0}\|\mathbf{z}-\mathbf{z}_{j_0}\|^{p+1}}{2^p(p+1)} \\ &\geq \frac{\Pi_{\ell=1}^{p+1}(j_0+\ell)}{(p+1)!}F(\bar{\mathbf{x}}_{j_0})+\frac{\tau_{j_0}\|\mathbf{z}-\mathbf{z}_{j_0}\|^{p+1}}{2^p(p+1)}. \qquad \text{(A.4)}\end{aligned}$$

Furthermore, observe that

$$\begin{aligned}&\psi_{j_0+1}(\mathbf{z}_{j_0+1},\tau_{j_0+1}) \\ &= \min_{\mathbf{z}\in\mathbb{R}^d}\ \psi_{j_0+1}(\mathbf{z},\tau_{j_0+1}) \\ &= \min_{\mathbf{z}\in\mathbb{R}^d}\ \{l_{j_0+1}(\mathbf{z})+\tau_{j_0+1}R(\mathbf{z})\} \\ &= \min_{\mathbf{z}\in\mathbb{R}^d}\ \left\{l_{j_0}(\mathbf{z})+\Delta l_{j_0}(\mathbf{z};\bar{\mathbf{x}}_{j_0+1},\bar{\xi}_{j_0+1})+\tau_{j_0}R(\mathbf{z})+(\tau_{j_0+1}-\tau_{j_0})R(\mathbf{z})\right\} \\ &\geq \min_{\mathbf{z}\in\mathbb{R}^d}\ \left\{\psi_{j_0}(\mathbf{z},\tau_{j_0})+\Delta l_{j_0}(\mathbf{z};\bar{\mathbf{x}}_{j_0+1},\bar{\xi}_{j_0+1})\right\}, \qquad \text{(A.5)}\end{aligned}$$

where the last inequality is due to the facts that $\tau_{j_0+1}\geq\tau_{j_0}$ and $R(\mathbf{z})\geq 0$ for any $\mathbf{z}\in\mathbb{R}^d$, and

$$\Delta l_{j_0}(\mathbf{z};\bar{\mathbf{x}}_{j_0+1},\bar{\xi}_{j_0+1}) = \frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\left[F(\bar{\mathbf{x}}_{j_0+1})+(\mathbf{z}-\bar{\mathbf{x}}_{j_0+1})^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right)\right].$$

Therefore, we have

$$
\begin{aligned}
&\psi_{j_0}(\mathbf{z},\tau_{j_0})+\Delta l_{j_0}(\mathbf{z},\bar{\mathbf{x}}_{j_0+1})\\
\overset{(A.4)}{\geq}\quad &\frac{\Pi_{\ell=1}^{p+1}(j_0+\ell)}{(p+1)!}F(\bar{\mathbf{x}}_{j_0})+\frac{\tau_{j_0}\left\|\mathbf{z}-\mathbf{z}_{j_0}\right\|^{p+1}}{2^p(p+1)}\\
&+\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\left[F(\bar{\mathbf{x}}_{j_0+1})+\left(\mathbf{z}-\bar{\mathbf{x}}_{j_0+1}\right)^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right)\right]\\
\overset{\text{Assumption 2.1}}{\geq}\quad &\frac{\Pi_{\ell=1}^{p+1}(j_0+\ell)}{(p+1)!}\left[F(\bar{\mathbf{x}}_{j_0+1})+\left(\bar{\mathbf{x}}_{j_0}-\bar{\mathbf{x}}_{j_0+1}\right)^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right)\right]\\
&+\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\left[F(\bar{\mathbf{x}}_{j_0+1})+\left(\mathbf{z}-\bar{\mathbf{x}}_{j_0+1}\right)^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right)\right]\\
&+\frac{\tau_{j_0}\left\|\mathbf{z}-\mathbf{z}_{j_0}\right\|^{p+1}}{2^p(p+1)}\\
=\quad &\frac{\Pi_{\ell=1}^{p+1}(j_0+1+\ell)}{(p+1)!}F(\bar{\mathbf{x}}_{j_0+1})+\frac{\tau_{j_0}\left\|\mathbf{z}-\mathbf{z}_{j_0}\right\|^{p+1}}{2^p(p+1)}\\
&+\frac{\Pi_{\ell=1}^{p+1}(j_0+\ell)}{(p+1)!}\left(\bar{\mathbf{x}}_{j_0}-\bar{\mathbf{x}}_{j_0+1}\right)^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right)\\
&+\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\left(\mathbf{z}-\bar{\mathbf{x}}_{j_0+1}\right)^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right), \qquad \text{(A.6)}
\end{aligned}
$$

where the last equality is due to the fact that

$$
\begin{aligned}
\frac{\Pi_{\ell=1}^{p+1}(j_0+\ell)}{(p+1)!}+\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}&=\frac{(j_0+1)\Pi_{\ell=2}^{p+1}(j_0+\ell)+(p+1)\Pi_{\ell=2}^{p+1}(j_0+\ell)}{(p+1)!}\\
&=\frac{\Pi_{\ell=2}^{p+2}(j_0+\ell)}{(p+1)!}=\frac{\Pi_{\ell=1}^{p+1}(j_0+1+\ell)}{(p+1)!}.
\end{aligned}
$$

Moreover, $\mathbf{y}_{j_0}$ in the algorithm is constructed to satisfy $\mathbf{y}_{j_0}=\frac{j_0+1}{j_0+p+2}\bar{\mathbf{x}}_{j_0}+\frac{p+1}{j_0+p+2}\mathbf{z}_{j_0}$, and thus

$$
\begin{aligned}
\frac{\Pi_{\ell=1}^{p+1}(j_0+\ell)}{(p+1)!}\bar{\mathbf{x}}_{j_0}&=\frac{\Pi_{\ell=1}^{p+1}(j_0+1+\ell)}{(p+1)!}\left(\frac{j_0+1}{j_0+p+2}\bar{\mathbf{x}}_{j_0}\right)\\
&=\frac{\Pi_{\ell=1}^{p+1}(j_0+1+\ell)}{(p+1)!}\left(\mathbf{y}_{j_0}-\frac{p+1}{j_0+p+2}\mathbf{z}_{j_0}\right)\\
&=\frac{\Pi_{\ell=1}^{p+1}(j_0+1+\ell)}{(p+1)!}\mathbf{y}_{j_0}-\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\mathbf{z}_{j_0}. \qquad \text{(A.7)}
\end{aligned}
$$

Combining (A.5), (A.6), and (A.7) yields

$$
\begin{aligned}
&\psi_{j_0+1}(\mathbf{z}_{j_0+1},\tau_{j_0+1})\\
&\geq\min_{\mathbf{z}\in\mathbb{R}^d}\left\{\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\left(\mathbf{z}-\mathbf{z}_{j_0}\right)^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right)+\frac{\tau_{j_0}\left\|\mathbf{z}-\mathbf{z}_{j_0}\right\|^{p+1}}{2^p(p+1)}\right\}\\
&\quad+\frac{\Pi_{\ell=1}^{p+1}(j_0+1+\ell)}{(p+1)!}\left(\mathbf{y}_{j_0}-\bar{\mathbf{x}}_{j_0+1}\right)^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right)\\
&\quad+\frac{\Pi_{\ell=1}^{p+1}(j_0+1+\ell)}{(p+1)!}F(\bar{\mathbf{x}}_{j_0+1}).
\end{aligned}
$$

Furthermore, since j_0 is a successful iteration, we have

$$\begin{aligned}\left(\mathbf{y}_{j_0}-\bar{\mathbf{x}}_{j_0+1}\right)^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right) &\geq \eta\left\|\mathbf{y}_{j_0}-\bar{\mathbf{x}}_{j_0+1}\right\|^{p+1}\\ &\overset{\text{Lemma A.5}}{\geq} \eta\left(\frac{\left\|\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right\|}{\rho_p+\bar{\sigma}_2+\kappa_\theta}\right)^{1+\frac{1}{p}}.\end{aligned}$$

Thus, it suffices to establish

$$\begin{aligned}&\frac{\eta\Pi_{\ell=1}^{p+1}(j_0+1+\ell)}{(p+1)!}\left(\frac{\left\|\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right\|}{\rho_p+\bar{\sigma}_2+\kappa_\theta}\right)^{1+\frac{1}{p}}+\frac{\tau_{j_0}\left\|\mathbf{z}-\mathbf{z}_{j_0}\right\|^{p+1}}{2^p(p+1)}\\ &\qquad+\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\left(\mathbf{z}-\mathbf{z}_{j_0}\right)^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right)\geq 0\ \forall\,\mathbf{z}\in\mathbb{R}^d.\end{aligned}\tag{A.8}$$

Indeed, applying (A.2) with

$$\mathbf{g}=\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right),\quad \mathbf{s}=\mathbf{z}-\mathbf{z}_{j_0},\quad \sigma=\frac{\tau_{j_0}}{2^p},\quad q=p+1,$$

we obtain that

$$\begin{aligned}&\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\left(\mathbf{z}-\mathbf{z}_{j_0}\right)^\top\left(\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right)+\frac{\tau_{j_0}\left\|\mathbf{z}-\mathbf{z}_{j_0}\right\|^{p+1}}{2^p(p+1)}\\ &\geq-\frac{p}{p+1}\left(\frac{2^p}{\tau_{j_0}}\right)^{\frac{1}{p}}\left(\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\left\|\nabla f(\bar{\mathbf{x}}_{j_0+1})+\bar{\xi}_{j_0+1}\right\|\right)^{1+\frac{1}{p}}.\end{aligned}$$

Therefore, (A.8) is equivalent to

$$\begin{aligned}\tau_{j_0}&\geq\frac{2^p\left(\rho_p+\bar{\sigma}_2+\kappa_\theta\right)^{p+1}}{\eta^p}\left(\frac{p}{p+1}\right)^p\left(\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{p!}\right)^{p+1}\left(\frac{(p+1)!}{\Pi_{\ell=1}^{p+1}(j_0+1+\ell)}\right)^p\\ &=\frac{2^p\left(\rho_p+\bar{\sigma}_2+\kappa_\theta\right)^{p+1}}{\eta^p}\left(\Pi_{\ell=2}^{p+1}(j_0+\ell)\right)\left(\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{\Pi_{\ell=1}^{p+1}(j_0+1+\ell)}\right)^p\frac{p^p}{p!}\\ &=\frac{2^p\left(\rho_p+\bar{\sigma}_2+\kappa_\theta\right)^{p+1}}{\eta^p}\left(\frac{\Pi_{\ell=2}^{p+1}(j_0+\ell)}{(j_0+p+2)^p}\right)\frac{p^p}{p!}.\end{aligned}$$

Now, observe that the first part of the conclusion would follow if

$$\tau_{j_0}\geq\frac{2^p\left(\rho_p+\bar{\sigma}_2+\kappa_\theta\right)^{p+1}}{\eta^p}\frac{p^{p-1}}{(p-1)!}$$

were to hold, which is the condition of the lemma.

To prove the remaining part of the conclusion, we note that τ_j can only be updated in the successful iteration of AAS, and it increases by a factor of γ_3 when updated. Recall that T_3 is the total number of updating counts for τ_j. Then according to the first part of the conclusion, τ_j will not be updated when

$$\tau_0\gamma_3^{T_3}\geq\frac{2^p\left(\rho_p+\bar{\sigma}_2+\kappa_\theta\right)^{p+1}}{\eta^p}\frac{p^{p-1}}{(p-1)!},$$

which means that T_3 is the least integer that makes the above inequality hold, and thus the conclusion follows. ∎

Now, we analyze the initial iterate in AAS, which is also the reinitialized iterate returned by SAS.

THEOREM A.7. *Let* $\bar{\mathbf{x}}_0$ *be the initial iterate in* AAS *of Algorithm* 3.1*; then by letting* $\hat{\sigma}_1 := \max\{\bar{\sigma}_1, \frac{L_p}{(p-1)!}\}$ *we have that*

$$F(\bar{\mathbf{x}}_0) \le \psi_0(\mathbf{z}, \tau_0) \le F(\mathbf{z}) + \frac{(p+1)\kappa_p + \hat{\sigma}_1}{p+1} \|\mathbf{z} - \mathbf{x}_0\|^{p+1} + \bar{\kappa}_p \|\mathbf{z} - \mathbf{x}_0\|^p + \frac{\tau_0 \|\mathbf{z} - \bar{\mathbf{x}}_0\|^{p+1}}{2(p+1)} + (\kappa_\theta + \hat{\sigma}_1)(2D)^{p+1}.$$

Proof. Recall that $F(\bar{\mathbf{x}}_0) = \min_{\mathbf{z}\in\mathbb{R}^d} \{F(\bar{\mathbf{x}}_0) + \tau_0 R(\mathbf{z})\} = \psi_0(\mathbf{z}_0, \tau_0)$. It suffices to show the inequality on the right-hand side. Denote by $\mathbf{x}_0 \in \mathbb{R}^d$ the initial iterate of SAS, by σ^{SAS} the regularized parameter associated with $\bar{\mathbf{x}}_0$, and by $\bar{\mathbf{x}}_0^m \in \mathbb{R}^d$ the global minimizer of $m(\mathbf{x}; \mathbf{x}_0, \sigma^{\mathsf{SAS}})$ over $\mathbb{R}^d$. Since $\bar{\mathbf{x}}_0$ is also the output returned by SAS, it holds that $m(\bar{\mathbf{x}}_0^m; \mathbf{x}_0, \sigma^{\mathsf{SAS}}) \le m(\bar{\mathbf{x}}_0; \mathbf{x}_0, \sigma^{\mathsf{SAS}}) \le m(\mathbf{x}_0; \mathbf{x}_0, \sigma^{\mathsf{SAS}}) = F(\mathbf{x}_0)$. Moreover, the updating rule of σ_i in SAS implies that $\sigma^{\mathsf{SAS}} \ge \sigma_{\min}$, which further indicates $m(\mathbf{x}; \mathbf{x}_0, \sigma_{\min}) \le m(\mathbf{x}; \mathbf{x}_0, \sigma^{\mathsf{SAS}})$ for all $\mathbf{x}$, and thus $\mathcal{L}(\mathbf{x}_0, \sigma^{\mathsf{SAS}}) \subseteq \mathcal{L}(\mathbf{x}_0, \bar{\sigma}_{\min})$. Then according to (3.4),

$$\|\bar{\mathbf{x}}_0 - \mathbf{x}^*\| \le D \quad \text{and} \quad \|\bar{\mathbf{x}}_0^m - \mathbf{x}^*\| \le D. \tag{A.9}$$

If $\overline{m}(\mathbf{y}; \mathbf{x})$ is convex, then $m(\mathbf{y}; \mathbf{x}, \sigma)$ is convex as well. Moreover, as we mentioned earlier, $m(\mathbf{y}; \mathbf{x}, \sigma)$ is not necessarily convex for the high-order adaptive accelerating method. In this case, let $\sigma_0^{\mathsf{AAS}} = \max\{\sigma^{\mathsf{SAS}}, -\lambda_{\min}\left(\nabla^2 \overline{m}(\bar{\mathbf{x}}_0; \mathbf{x}_0)\right)/\|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|^{p-1}\}$. According to the discussion above (4.4), $m(\mathbf{y}; \mathbf{x}_0, \sigma_0^{\mathsf{AAS}})$ is convex at $\bar{\mathbf{x}}_0$ and we have

$$F(\bar{\mathbf{x}}_0) \le m(\bar{\mathbf{x}}_0; \mathbf{x}_0, \sigma_0^{\mathsf{AAS}}) = m(\bar{\mathbf{x}}_0; \mathbf{x}_0, \sigma_0^{\mathsf{AAS}}) - m(\bar{\mathbf{x}}_0^m; \mathbf{x}_0, \sigma_0^{\mathsf{AAS}}) + m(\bar{\mathbf{x}}_0^m; \mathbf{x}_0, \sigma_0^{\mathsf{AAS}}).$$

Moreover, combining equality (2.3) in [49] and (4.3) yields

$$\nabla^2 f(\mathbf{y}) \preceq \nabla^2 \overline{m}(\mathbf{y}; \mathbf{x}) + \frac{L_p \|\mathbf{y} - \mathbf{x}\|^{p-1}}{(p-1)!} I \preceq \nabla^2 \overline{m}(\mathbf{y}; \mathbf{x}) + \sigma \|\mathbf{y} - \mathbf{x}\|^{p-1} I$$

when $\sigma \ge \frac{L_p}{(p-1)!}$ and $m(\mathbf{y}; \mathbf{x}, \sigma)$ is a convex function for any $\mathbf{x}$. Therefore,

$$\sigma_0^{\mathsf{AAS}} \le \hat{\sigma}_1 = \max\left\{\bar{\sigma}_1, \frac{L_p}{(p-1)!}\right\}, \tag{A.10}$$

and there exists some $\bar{\xi}_0 \in \partial r(\bar{\mathbf{x}}_0)$ (e.g., $\bar{\xi}_0$ could be the one that validates (3.2)) such that

$$\begin{aligned}
& m(\bar{\mathbf{x}}_0; \mathbf{x}_0, \sigma_0^{\mathsf{AAS}}) - m(\bar{\mathbf{x}}_0^m; \mathbf{x}_0, \sigma_0^{\mathsf{AAS}}) \\
\le\ & -(\nabla \overline{m}(\bar{\mathbf{x}}_0; \mathbf{x}_0) + \sigma_0^{\mathsf{AAS}} \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|^{p-1} (\bar{\mathbf{x}}_0 - \mathbf{x}_0) + \bar{\xi}_0)^\top (\bar{\mathbf{x}}_0^m - \bar{\mathbf{x}}_0) \\
\le\ & \left\| \nabla \overline{m}(\bar{\mathbf{x}}_0; \mathbf{x}_0) + \sigma^{\mathsf{SAS}} \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|^{p-1} (\bar{\mathbf{x}}_0 - \mathbf{x}_0) + \bar{\xi}_0 \right\| \|\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}_0^m\| \\
& + \left\| (\sigma_0^{\mathsf{AAS}} - \sigma^{\mathsf{SAS}}) \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|^{p-1} (\bar{\mathbf{x}}_0 - \mathbf{x}_0) \right\| \cdot \|\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}_0^m\| \\
\overset{(3.2),(A.10)}{\le}\ & (\kappa_\theta + \hat{\sigma}_1) \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|^p \|\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}_0^m\|.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
&m\left(\bar{\mathbf{x}}_0^m;\mathbf{x}_0,\sigma_0^{\mathsf{AAS}}\right)=\overline{m}(\bar{\mathbf{x}}_0^m;\mathbf{x}_0)+\frac{\sigma_0^{\mathsf{AAS}}\left\|\bar{\mathbf{x}}_0^m-\mathbf{x}_0\right\|^{p+1}}{p+1}+r(\bar{\mathbf{x}}_0^m)\\
&\le \overline{m}(\mathbf{z};\mathbf{x}_0)+\frac{\sigma_0^{\mathsf{AAS}}\left\|\mathbf{z}-\mathbf{x}_0\right\|^{p+1}}{p+1}+r(\mathbf{z})\\
&\overset{(2.4)}{\le} f(\mathbf{z})+\kappa_p\left\|\mathbf{z}-\mathbf{x}_0\right\|^{p+1}+\bar{\kappa}_p\|\mathbf{z}-\mathbf{x}_0\|^p+\frac{\sigma_0^{\mathsf{AAS}}\left\|\mathbf{z}-\mathbf{x}_0\right\|^{p+1}}{p+1}+r(\mathbf{z})\\
&\overset{(A.10)}{\le} F(\mathbf{z})+\frac{(p+1)\kappa_p+\hat{\sigma}_1}{p+1}\left\|\mathbf{z}-\mathbf{x}_0\right\|^{p+1}+\bar{\kappa}_p\left\|\mathbf{z}-\mathbf{x}_0\right\|^p.
\end{aligned}$$

Combining the above two inequalities, we have

$$\begin{aligned}
&\psi_0(\mathbf{z},\tau_0)=F(\bar{\mathbf{x}}_0)+\frac{\tau_0\left\|\mathbf{z}-\bar{\mathbf{x}}_0\right\|^{p+1}}{2(p+1)}\\
&\le \left(m(\bar{\mathbf{x}}_0;\mathbf{x}_0,\sigma_0^{\mathsf{AAS}})-m(\bar{\mathbf{x}}_0^m;\mathbf{x}_0,\sigma_0^{\mathsf{AAS}})\right)+m(\bar{\mathbf{x}}_0^m;\mathbf{x}_0,\sigma_0^{\mathsf{AAS}})+\frac{\tau_0\left\|\mathbf{z}-\bar{\mathbf{x}}_0\right\|^{p+1}}{2(p+1)}\\
&\le F(\mathbf{z})+\frac{(p+1)\kappa_p+\hat{\sigma}_1}{p+1}\left\|\mathbf{z}-\mathbf{x}_0\right\|^{p+1}+\bar{\kappa}_p\left\|\mathbf{z}-\mathbf{x}_0\right\|^p+\frac{\tau_0\left\|\mathbf{z}-\bar{\mathbf{x}}_0\right\|^{p+1}}{2(p+1)}\\
&\quad+(\kappa_\theta+\hat{\sigma}_1)\left\|\bar{\mathbf{x}}_0-\mathbf{x}_0\right\|^p\left\|\bar{\mathbf{x}}_0-\bar{\mathbf{x}}_0^m\right\|\\
&\le F(\mathbf{z})+\frac{(p+1)\kappa_p+\hat{\sigma}_1}{p+1}\left\|\mathbf{z}-\mathbf{x}_0\right\|^{p+1}+\bar{\kappa}_p\left\|\mathbf{z}-\mathbf{x}_0\right\|^p+\frac{\tau_0\left\|\mathbf{z}-\bar{\mathbf{x}}_0\right\|^{p+1}}{2(p+1)}\\
&\quad+(\kappa_\theta+\hat{\sigma}_1)(2D)^{p+1},
\end{aligned}$$

where the last inequality is due to (A.9) and (3.4). ∎

Next, we proceed to analyzing all the iterates in AAS.

THEOREM A.8. *The sequence* $\{\bar{\mathbf{x}}_j,\ j\ge 0\}$ *generated by* AAS *in* UAA *satisfies*

$$\begin{aligned}
&\frac{\Pi_{\ell=1}^{p+1}(j+\ell)}{(p+1)!}F(\bar{\mathbf{x}}_j)\ \le\ \psi_j(\mathbf{z},\tau_j)\le\frac{\Pi_{\ell=1}^{p+1}(j+\ell)}{(p+1)!}F(\mathbf{z})+\frac{(p+1)\kappa_p+\hat{\sigma}_1}{p+1}\left\|\mathbf{z}-\mathbf{x}_0\right\|^{p+1}\\
&\quad+\bar{\kappa}_p\left\|\mathbf{z}-\mathbf{x}_0\right\|^p+\frac{\tau_j\left\|\mathbf{z}-\bar{\mathbf{x}}_0\right\|^{p+1}}{2(p+1)}+(\kappa_\theta+\hat{\sigma}_1)(2D)^{p+1}. \qquad \text{(A.11)}
\end{aligned}$$

Proof. By the way in which $\psi_{j+1}(\mathbf{z}_{j+1},\tau_{j+1})$ is updated in AAS, we have

$$\frac{\Pi_{\ell=1}^{p+1}(j+1+\ell)}{(p+1)!}F(\bar{\mathbf{x}}_{j+1})\ \le\ \psi_j(\mathbf{z}_{j+1},\tau_{j+1})\ \le\ \psi_j(\mathbf{z},\tau_{j+1})\quad\forall\, j\ge 0.$$

It thus suffices to show the inequality on the right-hand side by induction. The base case of $j=0$ has already been proved in Theorem A.7. We now assume the result

holds for some $j = j_0$. For the case $j = j_0 + 1$, indeed, we have

$$
\begin{aligned}
&\psi_{j_0+1}(\mathbf{z}_{j_0+1}, \tau_{j_0+1}) \\
&\le \psi_{j_0+1}(\mathbf{z}, \tau_{j_0+1}) \\
&= l_{j_0}(\mathbf{z}) + \Delta l_{j_0}(\mathbf{z}; \bar{\mathbf{x}}_{j_0+1}, \bar{\xi}_{j_0+1}) + \frac{\tau_{j_0+1} \|\mathbf{z} - \mathbf{x}_0\|^{p+1}}{2(p+1)} \\
&= \psi_{j_0}(\mathbf{z}, \tau_{j_0}) + \Delta l_{j_0}(\mathbf{z}; \bar{\mathbf{x}}_{j_0+1}, \bar{\xi}_{j_0+1}) + \frac{(\tau_{j_0+1} - \tau_{j_0}) \|\mathbf{z} - \mathbf{x}_0\|^{p+1}}{2(p+1)} \\
&\le \frac{\Pi_{\ell=1}^{p+1}(j_0 + \ell)}{(p+1)!} F(\mathbf{z}) + \frac{(p+1)\kappa_p + \hat{\sigma}_1}{p+1} \|\mathbf{z} - \mathbf{x}_0\|^{p+1} + \bar{\kappa}_p \|\mathbf{z} - \mathbf{x}_0\|^p + \frac{\tau_{j_0} \|\mathbf{z} - \bar{\mathbf{x}}_0\|^{p+1}}{2(p+1)} \\
&\quad + (\kappa_\theta + \hat{\sigma}_1)(2D)^{p+1} + \frac{\Pi_{\ell=2}^{p+1}(j_0 + \ell)}{p!} \left[F(\bar{\mathbf{x}}_{j_0+1}) + (\mathbf{z} - \bar{\mathbf{x}}_{j_0+1})^\top \left(\nabla f(\bar{\mathbf{x}}_{j_0+1}) + \bar{\xi}_{j_0+1} \right) \right] \\
&\le \frac{\Pi_{\ell=1}^{p+1}(j_0 + 1 + \ell)}{(p+1)!} F(\mathbf{z}) + \frac{(p+1)\kappa_p + \hat{\sigma}_1}{p+1} \|\mathbf{z} - \mathbf{x}_0\|^{p+1} + \bar{\kappa}_p \|\mathbf{z} - \mathbf{x}_0\|^p \\
&\quad + \frac{\tau_{j_0} \|\mathbf{z} - \bar{\mathbf{x}}_0\|^{p+1}}{2(p+1)} + (\kappa_\theta + \hat{\sigma}_1)(2D)^{p+1},
\end{aligned}
$$

where the last inequality is due to Assumption 2.1, and the second-to-last inequality due to the mathematical induction, and τ_j is monotonically increasing. This completes the proof. ∎

Proof of Theorem 3.3. Recall that in the proof of Theorem A.7, we have shown $\|\mathbf{x}^* - \bar{\mathbf{x}}_0\| \le D$. Then, taking $\mathbf{z} = \mathbf{x}^*$ in (A.11) yields that

$$
\begin{aligned}
&\frac{\Pi_{\ell=1}^{p+1}(j + \ell)}{(p+1)!} F(\bar{\mathbf{x}}_j) \\
&\le \frac{\Pi_{\ell=1}^{p+1}(j + \ell)}{(p+1)!} F(\mathbf{x}^*) + \frac{(p+1)\kappa_p + \hat{\sigma}_1}{p+1} \|\mathbf{x}^* - \mathbf{x}_0\|^{p+1} + \bar{\kappa}_p \|\mathbf{x}^* - \mathbf{x}_0\|^p \\
&\quad + \frac{\tau_j \|\mathbf{x}^* - \bar{\mathbf{x}}_0\|^{p+1}}{2(p+1)} + (\kappa_\theta + \hat{\sigma}_1)(2D)^{p+1} \\
&\le \frac{\Pi_{\ell=1}^{p+1}(j + \ell)}{(p+1)!} F(\mathbf{x}^*) + \frac{(p+1)\kappa_p + \hat{\sigma}_1}{p+1} D^{p+1} + \bar{\kappa}_p D^p + \frac{\tau_j D^{p+1}}{2(p+1)} + (\kappa_\theta + \hat{\sigma}_1)(2D)^{p+1}.
\end{aligned}
$$

According to Lemma A.6, τ_j will not be increased once it exceeds $\frac{2^p(\rho_p + \bar{\sigma}_2 + \kappa_\theta)^{p+1} p^{p-1}}{\eta^p (p-1)!}$. Therefore, we have

$$
\tau_j \le \max \left\{ \tau_0, \frac{2^p \gamma_3 \left(\rho_p + \bar{\sigma}_2 + \kappa_\theta\right)^{p+1} p^{p-1}}{\eta^p (p-1)!} \right\} = \hat{\sigma}_2.
$$

Combining the two equalities above, it holds that

$$
F(\bar{\mathbf{x}}_j) - F(\mathbf{x}^*) \le \frac{(p+1)! \left(\frac{2(p+1)\kappa_p + 2\hat{\sigma}_1 + \hat{\sigma}_2}{2(p+1)} D^{p+1} + \bar{\kappa}_p D^p + (\kappa_\theta + \hat{\sigma}_1)(2D)^{p+1} \right)}{\Pi_{\ell=1}^{p+1}(j + \ell)}.
$$

Combining this inequality with Lemmas A.1, A.2, and A.6 implies the conclusion. ∎

Acknowledgments. We would like to thank the associate editor and the two anonymous referees for their insightful comments, and we would like to also thank Professor Xi Chen of the Stern School of Business at New York University for fruitful discussions at various stages of this project.

REFERENCES

[1] A. A. Ahmadi, A. Olshevsky, P. A. Parrilo, and J. N. Tsitsiklis, *NP-hardness of deciding convexity of quartic polynomials and related problems*, Math. Program., 137 (2013), pp. 453–476.

[2] Y. Arjevani, O. Shamir, and R. Shiff, *Oracle complexity of second-order methods for smooth convex optimization*, Math. Program., 178 (2019), pp. 327–360.

[3] M. Baes, *Estimate Sequence Methods: Extensions and Approximations*, Institute for Operations Research, ETH, Zürich, Switzerland, 2009.

[4] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[5] A. S. Berahas, R. Bollapragada, and J. Nocedal, *An investigation of Newton-Sketch and subsampled Newton methods*, Optim. Methods Software, 35 (2020), pp. 661–680.

[6] R. Bollapragada, R. H. Byrd, and J. Nocedal, *Exact and inexact subsampled Newton methods for optimization*, IMA J. Numer. Anal., 39 (2019), pp. 545–578.

[7] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint, *Evaluation complexity for nonlinear constrained optimization using unscaled KKT conditions and high-order models*, SIAM J. Optim., 26 (2016), pp. 951–967, https://doi.org/10.1137/15M1031631.

[8] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint, *Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models*, Math. Program., 163 (2017), pp. 359–368.

[9] B. Bullins and R. Peng, *Higher-Order Accelerated Methods for Faster Nonsmooth Optimization*, preprint, https://arxiv.org/abs/1906.01621, 2019.

[10] S. Bubeck, Q. Jiang, Y. T. Lee, Y. Li, and A. Sidford, *Near-Optimal Method for Highly Smooth Convex Optimization*, preprint, https://arxiv.org/abs/1812.08026, 2018.

[11] R. H. Byrd, J. Nocedal, and F. Oztoprak, *An inexact successive quadratic approximation method for l_1 regularized optimization*, Math. Program., 157 (2016), pp. 375–396.

[12] L. Calatroni and A. Chambolle, *Backtracking Strategies for Accelerated Descent Methods with Smooth Composite Objectives*, preprint, https://arxiv.org/abs/1709.09004, 2017.

[13] Y. Carmon and J. Duchi, *Gradient Descent Efficiently Finds the Cubic-Regularized Nonconvex Newton Step*, preprint, https://arxiv.org/abs/1612.00547v2, 2016.

[14] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization, part* I*: Motivation, convergence and numerical results*, Math. Program., 127 (2011), pp. 245–295.

[15] C. Cartis, N. I. M. Gould, and P. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization, part* II*: Worst-case function-and derivative-evaluation complexity*, Math. Program., 130 (2011), pp. 295–319.

[16] C. Cartis, N. I. M. Gould, and P. L. Toint, *Evaluation complexity of adaptive cubic regularization methods for convex unconstrained optimization*, Optim. Methods Software, 27 (2012), pp. 197–219.

[17] C. Cartis, N. I. M. Gould, and P. L. Toint, *On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization*, SIAM J. Optim., 22 (2012), pp. 66–86, https://doi.org/10.1137/100812276.

[18] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Improved Second-Order Evaluation Complexity for Unconstrained Nonlinear Optimization Using High-Order Regularized Models*, preprint, https://arxiv.org/abs/1708.04044, 2017.

[19] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Second-order optimality and beyond: Characterization and evaluation complexity in convexly constrained nonlinear optimization*, Found. Comput. Math., 18 (2018), pp. 1073–1107.

[20] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Universal regularization methods: Varying the power, the smoothness and the accuracy*, SIAM J. Optim., 29 (2019), pp. 595–615, https://doi.org/10.1137/16M1106316.

[21] X. Chen, Ph. L. Toint, and H. Wang, *Complexity of partially separable convexly constrained optimization with non-Lipschitzian singularities*, SIAM J. Optim., 29 (2019), pp. 874–903, https://doi.org/10.1137/18M1166511.

[22] X. Chen and Ph. L. Toint, *High-order evaluation complexity for convexly-constrained optimization with non-Lipschitzian group sparsity terms*, Math. Program., to appear; published online Jan. 28, 2020, https://doi.org/10.1007/s10107-020-01470-9.
[23] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, J. Mach. Learn. Res., 12 (2011), pp. 2121–2159.
[24] A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, C. A. Uribe, B. Jiang, H. Wang, S. Zhang, S. Bubeck, Q. Jiang, Y. T. Lee, Y. Li, and A. Sidford, *Near optimal methods for minimizing convex functions with Lipschitz p-th derivatives*, in Proceedings of the Thirty-Second Conference Conference on Learning Theory (COLT), Phoenix, AZ, Proc. Mach. Learn. Res. 99, 2019, pp. 1392–1393.
[25] S. Ghadimi, G. Lan, and H. Zhang, *Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization*, Math. Program., 155 (2016), pp. 267–305.
[26] S. Ghadimi and G. Lan *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Math. Program., 156 (2016), pp. 59–99.
[27] S. Ghadimi, H. Liu, and T. Zhang, *Second-Order Methods with Cubic Regularization under Inexact Information*, preprint, https://arxiv.org/abs/1710.05782, 2017.
[28] H. Ghanbari and K. Scheinberg, *Proximal quasi-Newton methods for regularized convex optimization with linear and accelerated sublinear convergence rates*, Comput. Optim. Appl., 68 (2018), pp. 597–627.
[29] G. N. Grapiglia and Y. Nesterov, *Accelerated regularized Newton methods for minimizing composite convex functions*, SIAM J. Optim., 29 (2019), pp. 77–99, https://doi.org/10.1137/17M1142077.
[30] G. N. Grapiglia and Y. Nesterov, *Tensor Methods for Minimizing Functions with Hölder Continuous Higher-Order Derivatives*, preprint, https://arxiv.org/abs/1904.12559, 2019.
[31] G. N. Grapiglia and Y. Nesterov, *Tensor Methods for Finding Approximate Stationary Points of Convex Functions*, preprint, https://arxiv.org/abs/1907.07053, 2019.
[32] B. Jiang, Z. Li, and S. Zhang, *On cones of nonnegative quartic forms*, Found. Comput. Math., 17 (2017), pp. 161–197.
[33] B. Jiang, T. Lin, and S. Zhang, *A Unified Scheme to Accelerate Adaptive Cubic Regularization and Gradient Methods for Convex Optimization*, preprint, https://arxiv.org/abs/1710.04788, 2017.
[34] B. Jiang, T. Lin, S. Ma, and S. Zhang, *Structured nonconvex and nonsmooth optimization: Algorithms and iteration complexity analysis*, Comput. Optim. Appl., 72 (2019), pp. 115–157.
[35] B. Jiang, H. Wang, and S. Zhang, *An optimal high-order tensor method for convex optimization*, Math. Oper. Res., to appear.
[36] A. Karparthy, *A Peak at Trends in Machine Learning*, Medium, posted April 7, 2017, https://medium.com/{@}karpathy/a-peek-at-trends-in-machine-learning-ab8a1085a106.
[37] D. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, preprint, https://arxiv.org/abs/1412.6980, 2014.
[38] J. D. Lee, Y. Sun, and M. A. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM J. Optim., 24 (2014), pp. 1420–1443, https://doi.org/10.1137/130921428.
[39] Q. Lin and L. Xiao, *An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization*, Comput. Optim. Appl., 60 (2014), pp. 633–674.
[40] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 4th ed., Internat. Ser. Oper. Res. Management Sci. 228, Springer, Cham, 2016.
[41] J. M. Martínez, *On high-order model regularization for constrained optimization*, SIAM J. Optim., 27 (2017), pp. 2447–2458, https://doi.org/10.1137/17M1115472.
[42] R. D. C. Monteiro, C. Ortiz, and B. F. Svaiter, *An adaptive accelerated first-order method for convex optimization*, Comput. Optim. Appl., 64 (2016), pp. 31–73.
[43] R. D. C. Monteiro and B. F. Svaiter, *An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods*, SIAM J. Optim., 23 (2013), pp. 1092–1125, https://doi.org/10.1137/110833786.
[44] Y. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$*, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547 (in Russian).
[45] Y. Nesterov and B. T. Polyak, *Cubic regularization of Newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.
[46] Y. Nesterov, *Accelerating the cubic regularization of Newton's method on convex problems*, Math. Program., 112 (2008), pp. 159–181.
[47] Y. Nesterov, *Gradient methods for minimizing composite functions*, Math. Program., 140 (2013), pp. 125–161.

[48] Y. Nesterov, *Lectures on Convex Optimization*, 2nd ed., Springer Optim. Appl. 137, Springer, Cham, 2018.
[49] Y. Nesterov, *Implementable tensor methods in unconstrained convex optimization*, Math. Program., to appear; published online Nov. 21, 2019, https://doi.org/10.1007/s10107-019-01449-1.
[50] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 2006.
[51] K. Scheinberg, D. Goldfarb, and X. Bai, *Fast first-order methods for composite convex optimization with backtracking*, Found. Comput. Math., 14 (2014), pp. 389–417.
[52] K. Scheinberg and X. Tang, *Practical inexact proximal quasi-Newton method with global complexity analysis*, Math. Program., 160 (2016), pp. 495–529.
[53] T. Tieleman and G. Hinton, *Lecture* 6.5-*RMSProp: Divide the gradient by a running average of its recent magnitude*, COURSERA: Neural Networks for Machine Learning, 4 (2012), pp. 26–31.
[54] A. Wilson, L. Mackey, and A. Wibisono, *Accelerating Rescaled Gradient Descent: Fast Optimization of Smooth Functions*, preprint, https://arxiv.org/abs/1902.08825, 2019.