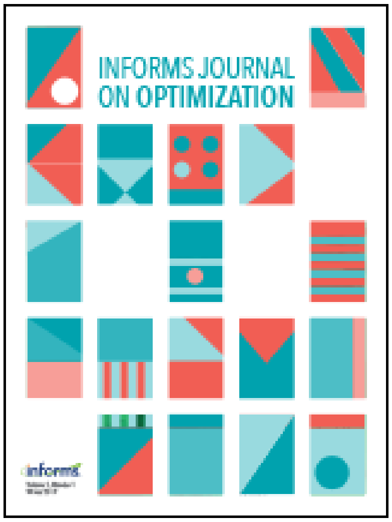


INFORMS Journal on Optimization

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Inexact Nonconvex Newton-Type Methods

Zhewei Yao, Peng Xu,Fred Roosta, Michael W. Mahoney

informs.
http://pubsonline.informs.org/journal/ijoo

INFORMS JOURNAL ON OPTIMIZATION
Vol. 3, No. 2, Spring 2021, pp. 154–182
ISSN 2572-1484 (print), ISSN 2575-1492 (online)

Inexact Nonconvex Newton-Type Methods

Zhewei Yao,[a] **Peng Xu,**[b] **Fred Roosta,**[c,d] **Michael W. Mahoney**[d,e]

[a] Department of Mathematics, University of California at Berkeley, Berkeley, California 94720; [b] Institute for Computational and Mathematical Engineering, Stanford University, Stanford, California 94305; [c] School of Mathematics and Physics, University of Queensland, Brisbane, QLD 4072, Australia; [d] International Computer Science Institute, Berkeley, California 94704; [e] Department of Statistics, University of California at Berkeley, Berkeley, California 94720

Contact: zheweiy@berkeley.edu, https://orcid.org/0000-0001-7678-4321 (ZY); pengxu@stanford.edu (PX); fred.roosta@uq.edu.au, https://orcid.org/0000-0002-6920-7072 (FR); mmahoney@stat.berkeley.edu (MWM)

Published Online in Articles in Advance:
March 4, 2021

https://doi.org/10.1287/ijoo.2019.0043

Copyright: © 2021 INFORMS

Abstract. For solving large-scale nonconvex problems, we propose inexact variants of trust region and adaptive cubic regularization methods, which, to increase efficiency, incorporate various approximations. In particular, in addition to inexact subproblem solves, both the gradient and Hessian are suitably estimated. Using certain conditions on such approximations, we show that our proposed inexact methods achieve similar optimal worst-case iteration complexities as the exact counterparts. In the context of finite-sum problems, we then explore randomized subsampling methods as ways to construct the gradient and Hessian approximations and examine the empirical performance of our algorithms on some model problems. We empirically demonstrate that our proposed algorithms are practically implementable in that failure to precisely fine-tune the associated hyperparameters is unlikely to result in unwanted behaviors, for example, divergence or stagnation.

Funding: M. W. Mahoney and F. Roosta gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory under agreement number FA8750-17-2-0122. F. Roosta is also thankful for the support of the Australian Research Council through a Discovery Early Career Researcher Award [Grant DE180100923] and the Australian Centre of Excellence for Mathematical and Statistical Frontiers [Grant CE140100049].

Keywords: nonconvex optimization • inexact gradient • inexact Hessian • trust region • cubic regularization

1. Introduction

We consider the following generic optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}), \tag{1}$$

where $F:\mathbb{R}^d \to \mathbb{R}$ is a smooth but possibly nonconvex function. Over the last few decades, many optimization algorithms have been developed to solve (1). The bulk of these efforts in the machine learning (ML) community has been on developing first-order methods, that is, those that solely rely on gradient information; see the recent textbooks Beck (2017), Lan (2020), and Lin et al. (2020) for excellent and in-depth treatments of such class of methods. Such algorithms, however, can generally be, at best, ensured to converge to *first-order stationary points*, that is, $\mathbf{x}$ for which $\|\nabla F(\mathbf{x})\| = 0$, which include *saddle points*. It is argued that converging to saddle points can be undesirable for obtaining good generalization errors with many nonconvex machine learning models, such as deep neural networks (LeCun et al. 2012, Saxe et al. 2013, Dauphin et al. 2014, Choromanska et al. 2015). In fact, it is also shown that, in certain settings, existence of "bad" local minima, that is, suboptimal local minima with high training error, can significantly hurt the performance of the trained model at test time (Fukumizu and Amari 2000, Swirszcz et al. 2016). Important cases have also been demonstrated in which stochastic gradient descent, which is, nowadays, arguably the optimization method of choice in ML, indeed stagnates at high training error (He et al. 2016a). As a result, scalable algorithms that avoid saddle points and guarantee convergence to a local minimum are desired.

Second-order methods, on the other hand, that effectively employ the curvature information in the form of a Hessian, have the potential for convergence to second-order stationary points, that is, $\mathbf{x}$ for which $\|\nabla F(\mathbf{x})\| = 0$ and $\nabla^2 F(\mathbf{x}) \succeq 0$. However, the main challenge preventing the ubiquitous use of these methods is the computational costs involving the application of the underlying matrices, for example, Hessian. In an effort to address these computational challenges, for large-scale convex settings, stochastic variants of Newton's methods are shown not only to enjoy desirable theoretical properties, for example, fast convergence rates and robustness to problem ill conditioning (Xu et al. 2016, Bollapragada et al. 2018, Roosta and Mahoney 2019), but also to exhibit superior empirical performance (Berahas et al. 2017, Kylasa et al. 2019).

For nonconvex optimization, however, the development of similar efficient methods lags significantly behind. Indeed, designing efficient and Hessian-free variants of classic nonconvex Newton-type methods, such as trust-region (TR) (Conn et al. 2000), cubic regularization (CR) (Nesterov and Polyak 2006), and its adaptive variant (ARC) (Cartis et al. 2011a, b), can be an appropriate place to start bridging this gap. This is, in particular, encouraging because Hessian-free methods only involve Hessian-vector products, which, in many cases, including neural networks (Griewank 1993, Pearlmutter 1994), are computed as efficiently as evaluating gradients. In this light, coupling *stochastic approximation* with *Hessian-free* techniques indeed holds promise for many of the challenging ML problems of today, for example, Martens (2010), Xu et al. (2020), and Regier et al. (2017).

In many applications, however, even accessing the exact gradient information can be very expensive. For example, for finite-sum problems in high dimensions in which

$$F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{2}$$

computing the exact gradient requires a pass over the entire data, which can be costly when $n \gg 1$. Inexact access to *both* the gradient and Hessian information can usually help reduce the underlying computational costs (Tripuraneni et al. 2018, Roosta and Mahoney 2019). Here, we aim to advance the developments in the aforementioned directions.

The rest of this paper is organized as follows. We briefly highlight the main contributions of the present paper in Section 1.1. A brief survey of the related work is gathered in Section 1.2. Notation and assumptions used throughout the paper are introduced in Section 1.3. We present the detailed theoretical analysis of our proposed methods in Section 2. In particular, analyses of TR and ARC are, respectively, given in Sections 2.1 and 2.2. Treatment of the finite-sum problem (2) is presented in Section 2.3. Section 3 contains some numerical examples. Conclusions and further thoughts are gathered in Section 4.

1.1. Contributions

Here, we further these ideas by analyzing *inexact* variants of TR and ARC algorithms, which, to increase efficiency, incorporate *approximations* of *gradient* and *Hessian information* as well as solutions of the underlying *subproblems*. Our algorithms are motivated by the works of Cartis et al. (2012) and Xu et al. (2019), which analyze the variants of TR and ARC in which Hessian is approximated but the accurate gradient information is required. We show that, under mild conditions on approximations of the gradient, Hessian, as well as subproblem solves, our proposed inexact TR and ARC algorithms can retain the same optimal worst-case convergence guarantees as the exact counterparts (Cartis et al. 2011c, 2012). More specifically, to achieve (ϵ_g, ϵ_H)-optimality (cf. Definition 1), we show the following:

i. Inexact TR (Algorithm 1) under Condition 1 on the gradient and Hessian approximation and Condition 2 on approximate subproblem solves requires the optimal iteration complexity of $\mathcal{O}(\max\{\epsilon_g^{-2}\epsilon_H^{-1}, \epsilon_H^{-3}\})$. In particular, we obtain convergence for a practical case in which the accuracy tolerances in gradient and Hessian estimations, that is, (δ_g, δ_H) in Condition 1, are adaptively chosen; see Section 2.1 for more details.

ii. Inexact ARC (Algorithm 2) under Condition 3 on the gradient and Hessian approximation and Condition 4 on approximate subproblem solves requires less than $\mathcal{O}(\max\{\epsilon_g^{-2}, \epsilon_H^{-3}\})$, which is suboptimal. These conditions are described in Section 2.2.1. However, under respectively stronger Conditions 5 and 6, the optimal iteration complexity of $\mathcal{O}(\max\{\epsilon_g^{-3/2}, \epsilon_H^{-3}\})$ is recovered. Unfortunately, we were unable to provide convergence guarantees with adaptive tolerances, and as a result, (δ_g, δ_H) in Condition 3 are set fixed a priori to a sufficiently small value. The details are given in Section 2.2.2.

To prove our results, we follow a similar line of reasoning as that in Xu et al. (2019). However, incorporating gradient approximation introduces several new layers of technical difficulty. These difficulties arise as a result of the discrepancy between the true objective function and its quadratic and cubic approximations within inexact TR and ARC, respectively, which now involve an additional "bias" term. Properly controlling such an added error term necessitates a much finer grained analysis. For example, one has to consider various scenarios arising from large and small gradient norms. Among all of these, the case in which the true gradient is small enough to be of similar magnitude as the approximation noise level requires a special treatment and analysis.

We finally empirically demonstrate the advantages of our methods on several model problems in Section 3. In addition to showing favorable performance, for example, in terms of efficiency, we also highlight some additional features of our algorithms, such as robustness to hyperparameter tuning. Such properties amount to a *practically implementable* algorithm for which failure to fine-tune the hyperparameters is unlikely to result in divergence or stagnation.

A snapshot of comparison among our proposed methods and other similar algorithms is given in Table 1.

1.2. Related Work

Because of the resurgence of deep learning, recently, there has been a rise of interest in efficient nonconvex optimization algorithms. For nonconvex problems in which saddle points have been shown to give understandable generalization performance, several variants of stochastic gradient descent (SGD) have recently been devised that promise to efficiently escape saddle points and, instead, converge to second-order stationary points (Ge et al. 2015, Levy 2016, Jin et al. 2017).

As for second-order methods, there have been a few empirical studies of the application of inexact curvature information for, mostly, deep-learning applications; for example, see the work of Martens (2010) and follow-ups, for example, Wiesler et al. (2013), Vinyals and Povey (2012), He et al. (2016b), and Kiros (2013). However, the theoretical understanding of these inexact methods remains largely understudied. Among a few related theoretical prior works, most notable are the ones that study derivative-free and probabilistic models in general and Hessian approximation in particular for trust-region methods (Conn et al. 2009, Bandeira et al. 2014, Chen et al. 2015, Larson and Billups 2016, Gratton et al. 2017, Shashaani et al. 2018, Blanchet et al. 2019).

For cubic regularization, the seminal works of Cartis et al. (2011a, b) are the first to study Hessian approximation, and the resulting algorithm is an adaptive variant of the cubic regularization, referred to as ARC. In Cartis et al. (2012), similar Hessian inexactness is also extended to trust-region methods. However, to guarantee optimal complexity, they require not only exact gradient information, but also progressively accurate Hessian information, which can be difficulty to satisfy. More general treatment of line search and cubic regularization methods based on probabilistic models are given in Cartis and Scheinberg (2018). For minimization of a finite sum (2), a subsampled variant of ARC is proposed in Kohler and Lucchi (2017), which directly relies on the analysis of Cartis et al. (2011a, b). A more refined analysis is given in Chen et al. (2018), which incorporates subsampling strategies to develop both standard and accelerated ARC variants for convex objectives. More recently, Tripuraneni et al. (2018) propose a stochastic variant of cubic regularization, henceforth referred to as SCR, in which, in order to guarantee optimal performance, only the stochastic gradient and Hessian are required. However, for the practical implementation of their algorithm, one must either assume to know rather unknowable problem-related constants, for example, the Lipschitz continuity of the gradient and Hessian, or perform an exhaustive grid search over the space of hyperparameters.

In the context of both TR and ARC, under milder Hessian approximation conditions than prior works, Xu et al. (2019) analyze optimal complexity of variants in which the Hessian matrix is approximated but the exact gradient is used. Our approach here builds upon the ideas in Xu et al. (2019).

1.3. Notation and Assumptions

1.3.1. Notation. Throughout the paper, vectors and matrices are denoted by bold lowercase and blackboard bold uppercase letters, for example, $\mathbf{v}$ and $\mathbf{V}$, respectively. We use regular lowercase and uppercase letters to denote scalar constants, for example, c or K. The transpose of a real vector $\mathbf{v}$ is denoted by $\mathbf{v}^T$. The inner product between two vectors $\mathbf{v}, \mathbf{w}$ is denoted by $\langle \mathbf{v}, \mathbf{w} \rangle$. For a vector $\mathbf{v}$ and a matrix $\mathbf{V}$, $\|\mathbf{v}\|$ and $\|\mathbf{V}\|$ denote the vector ℓ_2 norm and the matrix spectral norm, respectively. The subscript, for example, $\mathbf{x}_t$, denotes the

Table 1. Comparison of Optimal Worst-Case Iteration Complexities for Convergence to a $(\epsilon, \sqrt{\epsilon})-$ Optimality (cf. Definition 1) Among Different Second-Order Methods for Nonconvex Optimization

Method class	Iteration complexity	Inexact Hessian	Inexact gradient	Practically implementable
TR (Cartis et al. 2012)	$\mathcal{O}(\epsilon^{-2.5})$	✓	✗	✓
TR (Xu et al. 2019)	$\mathcal{O}(\epsilon^{-2.5})$	✓	✗	✓
TR (Algorithm 1)	$\mathcal{O}(\epsilon^{-2.5})$	✓	✓	✓
CR (Cartis et al. 2012)	$\mathcal{O}(\epsilon^{-1.5})$	✓	✗	✓
CR (Xu et al. 2019)	$\mathcal{O}(\epsilon^{-1.5})$	✓	✗	✓
CR (Tripuraneni et al. 2018)	$\mathcal{O}(\epsilon^{-1.5})$	✓	✓	✗
CR (Algorithm 2)	$\mathcal{O}(\epsilon^{-1.5})$	✓	✓	✓

Notes. TR and CR refer, respectively, to the class of trust region and cubic regularization methods. "Practically implementable" refers to an algorithm that not only does not require exhaustive search over hyperparameter space for tuning, but also failure to precisely fine-tune is not likely to result in unwanted behaviors, for example, divergence or stagnation.

iteration counter. At iteration t, the approximations of the gradient and Hessian are written, respectively, as $\mathbf{g}_t$ and $\mathbf{H}_t$. The smallest eigenvalue of matrix $\mathbf{V}$ is denoted by $\lambda_{\min}(\mathbf{V})$.

1.3.2. Assumptions. Unlike convex problems in which tracking the first-order condition, that is, the norm of the gradient, is sufficient to evaluate (approximate) optimality, in nonconvex settings, the situation is much more involved; for example, see examples of Murty and Kabadi (1987) and Hillar and Lim (2013). In this light, one typically sets out to design algorithms that can guarantee convergence to approximate second-order optimality.

Definition 1. ((ϵ_g, ϵ_H)-Optimality). Given $0 < \epsilon_g, \epsilon_H < 1$, $\mathbf{x}$ is an (ϵ_g, ϵ_H)-optimal solution of (1) if

$$\|\nabla F(\mathbf{x})\| \le \epsilon_g, \quad \text{and} \quad \lambda_{\min}\big(\nabla^2 F(\mathbf{x})\big) \ge -\epsilon_H. \tag{3}$$

For our analysis throughout the paper, we make the following standard assumptions on the smoothness of objective function F.

Assumption 1 (Hessian Regularity). *$F(\mathbf{x})$ is twice continuously differentiable. Furthermore, there are some constants $0 < L_F, K_F < \infty$ such that, for any $\mathbf{x} = \mathbf{x}_t + \tau \mathbf{s}_t$, $\tau \in [0,1]$, we have*

$$\|\nabla^2 F(\mathbf{x}) - \nabla^2 F(\mathbf{x}_t)\| \le L_F \|\mathbf{x} - \mathbf{x}_t\|, \tag{4a}$$

$$\|\nabla^2 F(\mathbf{x}_t)\| \le K_F, \tag{4b}$$

where $\mathbf{x}_t$ and $\mathbf{s}_t$ are, respectively, the iterate and update direction at the t^{th} iteration.

For our inexact algorithms, we require that the approximate gradient, $\mathbf{g}_t$, and the inexact Hessian, $\mathbf{H}_t$, at each iteration t, satisfy the following conditions.

Assumption 2 (Gradient and Hessian Approximation Error). *For some $0 < \delta_g, \delta_H < 1$, the approximations of the gradient and Hessian t^{th} iteration satisfy*

$$\|\mathbf{g}_t - \nabla F(\mathbf{x}_t)\| \le \delta_g,$$
$$\|\mathbf{H}_t - \nabla^2 F(\mathbf{x}_t)\| \le \delta_H.$$

Note that, by the triangle inequality, Assumptions 1 and 2 imply that $\|\mathbf{H}_t\| \le K_H$, where $K_H \le K_F + \delta_H$.

2. Algorithms and Theoretical Analysis

In this section, we present our main algorithms as well as their respective analyses, that is, inexact variants of TR (Algorithm 1) and ARC (Algorithm 2) in which the gradient, Hessian, and solution to the subproblems are all approximated.

As can be seen from Algorithms 1 and 2, compared with the standard classical counterparts, the main differences in iterations lie in using the approximations of the gradient, Hessian, and solution to the corresponding subproblems (6) and (13). Another notable difference is when the gradient estimate is small, that is, $\|\mathbf{g}_t\| \le \epsilon_g$, in which case our algorithm completely ignores the gradient; see step 5 of Algorithms 1 and 2. This turns out to be crucial in obtaining the optimal iteration complexity for both algorithms. Intuitively, when the gradient is too small, its approximation involves a great degree of noisy information. As a result, solving the subproblems using such noisy gradient information can result in directions of ascent. In practice, however, such unfortunate steps are usually simply corrected by the subsequent steps, and hence, one can always safely employ the approximate gradient without any such safeguard. In this light, in our experiments, we never needed to enforce this step and opted to retain the gradient term even when it was small.

It can also be seen that Algorithms 1 and 2 are highly similar in their corresponding steps. In particular, after initialization, one computes a local model m_t of F around $\mathbf{x}_t$ and obtains a step that guarantees model reduction $m_t(\mathbf{s}_t) < m_t(\mathbf{0}) = 0$. Subsequently, one checks that the actual reduction in F is in accordance with what is predicted using the local model. More specifically, at every iteration of Algorithms 1 and 2, by computing

$$\rho_t := \frac{F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t)}{m_t(\mathbf{s}_t)}, \tag{5}$$

one checks whether $F(\mathbf{x}_t) - F(\mathbf{x}_T + \mathbf{s}_T)$ is large enough relative to the reduction in the local model $m_t(\mathbf{s}_t) - m_t(\mathbf{0})$. If ρ_t is larger than a preset threshold, the update $\mathbf{s}_t$ is accepted, and we set $\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{s}_t$. In this case, local models are "loosened" to allow for larger trial steps in the next iteration. However, a small value of ρ_t hints at a large discrepancy between the predicted and the actual reduction in F, which implies that the local models mismatch the actual function. In this case, the step is rejected, and the local models are "tightened" by adjusting the trust region or cubic regularization parameters.

At a high level, these similar algorithmic steps give rise to similar analytical steps as well. For example, the notion of Cauchy and Eigen points plays a crucial role in the analysis of both algorithms. Generally, when the gradient is large, both algorithms adopt the Cauchy point, otherwise the Eigen point is used as a trial step. Furthermore, it is shown that, as long as Δ_t is small enough and σ_t is large enough, the trial steps generated, respectively, by Algorithms 1 and 2 are accepted. This, in turn, implies that, after a fixed number of rejections, both algorithms are guaranteed to eventually accept their trial steps and, hence, make progress toward optimality. Because the overall number of rejected trial steps is upper-bounded, we are guaranteed to obtain convergence for both algorithms. Although the high-level analyses have many common grounds, the analysis of Algorithms 1 and 2 have distinctive technical features as well. In particular, obtaining optimal complexity of Algorithm 2 requires more restrictive conditions on the solution of the subproblem than simple Cauchy or Eigen points, and it also necessitates a more refined analysis and careful control over the size of the accepted steps at each successful iteration.

Algorithm 1 (Inexact TR).

```
Input:
    - Starting point: $\mathbf{x}_0$
    - Initial trust-region radius: $\Delta_0 > 0$
    - Other parameters: $0 \le \epsilon_g, 0 \le \epsilon_H, 0 < \eta \le 1, \gamma > 1$.
$t = 0$
while $\|\mathbf{g}_t\| \ge \epsilon_g,\ \lambda_{\min}(\mathbf{H}_t) \le -\epsilon_H$ , do
    if $\|\mathbf{g}_t\| \le \epsilon_g$, then
        $\mathbf{g}_t = 0$
    end if
    Find $\mathbf{s}_t$ as in (6)
    Set $\rho_t$ as in (5) with $m_t$ as in (6b)
    if $\rho_t \ge \eta$, then
        $\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{s}_t$ and $\Delta_{t+1} = \gamma\Delta_t$
    else
        $\mathbf{x}_{t+1} = \mathbf{x}_t$ and $\Delta_{t+1} = \Delta_t/\gamma$
    end if
    $t = t + 1$
end while
Output: $\mathbf{x}_t$
```

Algorithm 2 (Inexact ARC).

```
Input:
    - Starting point: $\mathbf{x}_0$
    - Initial regularization parameter: $\sigma_0 > 0$
    • Other parameters: $0 \le \epsilon_g, 0 \le \epsilon_H, 0 < \eta \le 1, \gamma > 1$.
$t = 0$
while $\|\mathbf{g}_t\| \ge \epsilon_g,\ \lambda_{\min}(\mathbf{H}_t) \le -\epsilon_H$ , do
    if $\|\mathbf{g}_t\| \le \epsilon_g$, then
        $\mathbf{g}_t = 0$
    end if
    Find $\mathbf{s}_t$ as in (13)
    Set $\rho_t$ as in (5) with $m_t$ as in (13b)
    if $\rho_t \ge \eta$, then
        $\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{s}_t$ and $\sigma_{t+1} = \sigma_t/\gamma$
    else
        $\mathbf{x}_{t+1} = \mathbf{x}_t$ and $\sigma_{t+1} = \gamma\sigma_t$
    end if
    $t = t + 1$
end while
Output: $\mathbf{x}_t$
```

2.1. Inexact Trust Region

The inexact TR algorithm is depicted in Algorithm 1. Every iteration of Algorithm 1 involves an approximate solution to a subproblem of the form

$$\mathbf{s}_t \approx \operatorname*{argmin}_{\|\mathbf{s}\| \le \Delta_t} m_t(\mathbf{s}), \tag{6a}$$

where

$$m_t(\mathbf{s}) \triangleq \begin{cases} \langle \mathbf{g}_t, \mathbf{s} \rangle + \dfrac{1}{2}\langle \mathbf{s}, \mathbf{H}_t \mathbf{s} \rangle, & \|\mathbf{g}_t\| \ge \epsilon_g \\ \langle \mathbf{s}, \mathbf{H}_t \mathbf{s} \rangle, & \text{Otherwise} \end{cases}. \tag{6b}$$

Classically, analysis of the TR method involves obtaining a minimum descent along two important directions, namely negative gradient and (approximate) negative curvature. Updating the current point using these directions gives, respectively, what are known as Cauchy and Eigen points (Conn et al. 2000). In other words, the Cauchy and Eigen points, respectively, correspond to the optimal solution of (6) along the negative gradient and the negative curvature direction (if it exists).

Definition 2 (Cauchy Point for Algorithm 1)**.** When $\|\mathbf{g}_t\| \ge \epsilon_g$, the Cauchy point for Algorithm 1 is obtained from (6) as

$$\mathbf{s}_t^C = -\frac{\alpha^C}{\|\mathbf{g}_t\|}\mathbf{g}_t, \quad \alpha^C = \operatorname*{argmin}_{0 \le \alpha \le \Delta_t} m_t\left(-\frac{\alpha}{\|\mathbf{g}_t\|}\mathbf{g}_t\right). \tag{7a}$$

Definition 3 (Eigen Point for Algorithm 1)**.** When $\lambda_{\min}(\mathbf{H}_t) \le -\epsilon_H$, the Eigen point for Algorithm 1 is obtained from (6) as

$$\mathbf{s}_t^E = \alpha^E \mathbf{u}_t, \quad \alpha^E = \operatorname*{argmin}_{|\alpha| \le \Delta_t} m_t(\alpha \mathbf{u}_t), \tag{7b}$$

where $\mathbf{u}_t$ is an approximation to the corresponding negative curvature direction; that is, for some $0 < \nu < 1$,

$$\langle \mathbf{u}_t, \mathbf{H}_t \mathbf{u}_t \rangle \le \nu \lambda_{\min}(\mathbf{H}_t) \text{ and } \|\mathbf{u}_t\| = 1.$$

The properties of Cauchy and Eigen points are studied in Cartis et al. (2011a, b) and Xu et al. (2019) and are also stated in Lemmas 1 and 2.

We are now ready to give the convergence guarantee of Algorithm 1. For this, we first present sufficient conditions (Condition 1) on the degree of inexactness of the gradient and Hessian. In other words, we now give conditions on δ_g, δ_H in Assumption 2 that ensure convergence.

Condition 1 (Gradient and Hessian Approximation for Algorithm 1)**.** *Given the termination criteria, ϵ_g, ϵ_H, in Algorithm 1, we require the inexact gradient and Hessian to satisfy*

$$\delta_g \le \left(\frac{1-\eta}{4}\right)\max\{\epsilon_g, \|\mathbf{g}_t\|\} \quad \textit{and} \quad \delta_H \le \min\left\{\max\left\{\frac{(1-\eta)\nu\epsilon_H}{2}, \Delta_t\right\}, 1\right\}. \tag{8}$$

Note that Condition 1 is *adaptive*, which can have desirable consequences in practice. For example, when Δ_t is large (which is typically the case during the early stages of the algorithm), one can afford a cruder approximation of the Hessian by choosing larger δ_H. Similarly, the condition on δ_g, for large $\|\mathbf{g}_t\|$, amounts to a relative error condition. Although this latter condition on δ_g is perhaps not easily enforceable a priori (unless one has a lower-bound estimate of $\|\mathbf{g}_t\|$), it nonetheless qualitatively indicates that, when the true gradient is large, one can very well employ loose approximations; see also Remark 1. As the algorithm progresses toward convergence, Condition 1 implies that, ultimately, we must seek to have $\delta_g \in \mathcal{O}(\epsilon_g), \delta_H \in \mathcal{O}(\epsilon_H)$. These bounds are indeed the minimum requirements for the gradient and Hessian approximations to achieve (ϵ_g, ϵ_H)-optimality; see the termination step for Algorithm 1.

In Algorithm 1, subproblem (6) needs only be solved approximately. Indeed, in large-scale settings, obtaining the exact solution of subproblem (6) is computationally prohibitive. For this, as has been classically done, we require that an approximate solution of the subproblem satisfies what are known as Cauchy and Eigen conditions (Conn et al. 2000, Cartis et al. 2010, Xu et al. 2019). In other words, we require that an approximate solution to (6) is at least as good as the Cauchy and Eigen points in Definitions 2 and 3, respectively. Condition 2 makes this explicit.

Condition 2. (Approximate Solution of (6) for Algorithm 1). *If $\|\mathbf{g}_t\| \geq \epsilon_g$, then we take the Cauchy point, that is, $\mathbf{s}_t = \mathbf{s}_t^C$; otherwise, we take the Eigen point, that is, $\mathbf{s}_t = \mathbf{s}_t^E$. Here, $\mathbf{s}_t^C$ and $\mathbf{s}_t^E$ are Cauchy and Eigen points as in Definitions 2 and 3, respectively.*

Under Assumptions 1 and 2, as well as assuming Conditions 1 and 2 hold, we are now ready to give the optimal iteration complexity of Algorithm 1. We first give the following two standard lemmas regarding Cauchy and Eigen points (Conn et al. 2000), which establish Condition 2.

Lemma 1. (Cauchy Points; Conn et al. 2000, Corollary 6.3.2). *Suppose that $\mathbf{s}_t^C = \arg\min_{\|\alpha \mathbf{g}_k\| \leq \Delta_t} m_t(-\alpha \mathbf{g}_t)$. We have*

$$-m_t(\mathbf{s}_t^C) \geq \frac{1}{2}\|\mathbf{g}_t\| \min\left\{\frac{\|\mathbf{g}_t\|}{1+\|\mathbf{H}_t\|}, \Delta_t\right\}. \tag{9}$$

Lemma 2. (Eigen Points; Conn et al. 2000, Theorem 6.6.1). *When $\lambda_{\min}(\mathbf{H}_t)$ is negative, suppose $\mathbf{u}_t$ satisfies*

$$\langle \mathbf{g}_t, \mathbf{u}_t \rangle \leq 0, \quad \text{and} \quad \langle \mathbf{u}_t, \mathbf{H}_t \mathbf{u}_t \rangle \leq -\nu |\lambda_{\min}(\mathbf{H}_t)| \|\mathbf{u}_t\|^2, \tag{10}$$

and let $\mathbf{s}_t^E = \arg\min_{\|\mathbf{s}_t\| \leq \Delta_t} m_t(\alpha \mathbf{u}_t)$. We have

$$-m_t(\mathbf{s}_t^E) \geq \frac{\nu}{2}|\lambda_{\min}(\mathbf{H}_t)|\Delta_t^2. \tag{11}$$

These two lemmas show the descent that can be obtained by Cauchy and Eigen points. The following lemma bounds the difference between the actual decrement, that is, $F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t)$, and the one predicted by $m(\mathbf{s}_t)$. The detailed proof is included in the appendix.

Lemma 3. *Under Assumptions 1 and 2, we have*

$$F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t) - m_t(\mathbf{s}_t) \leq \begin{cases} \delta_g \Delta_t + \frac{1}{2}\delta_H {\Delta_t}^2 + \frac{1}{2} L_F {\Delta_t}^3, & \|\mathbf{g}_t\| \geq \epsilon_g, \\ \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t) \rangle + \frac{1}{2}\delta_H {\Delta_t}^2 + \frac{1}{2} L_F {\Delta_t}^3, & \|\mathbf{g}_t\| < \epsilon_g. \end{cases} \tag{12}$$

By combining Lemmas 1 and 3, Lemma 4 guarantees that, in case $\|\mathbf{g}_t\| \geq \epsilon_g$, the iteration is successful and the update is accepted.

Lemma 4. *Suppose Assumptions 1 and 2 as well as Conditions 1 and 2 hold. Furthermore, suppose, at iteration t, we have $\|\mathbf{g}_t\| \geq \epsilon_g$ and*

$$\Delta_t \leq \min\left\{\frac{\|\mathbf{g}_t\|}{1+K_H}, \sqrt{\frac{(1-\eta)\|\mathbf{g}_t\|}{12 L_F}}, \frac{(1-\eta)\|\mathbf{g}_t\|}{3}\right\}.$$

Then the iteration t is successful; that is, $\Delta_{t+1} = \gamma \Delta_t$.

Proof. First, because $\|\mathbf{g}_t\| \geq \epsilon_g$ and $\Delta_t \leq \|\mathbf{g}_t\|/(1+K_H)$, by Condition 2, we have $\mathbf{s}_t = \mathbf{s}_t^C$ and

$$-m_t(\mathbf{s}_t) \geq \frac{1}{2}\|\mathbf{g}_t\| \min\left\{\frac{\|\mathbf{g}_t\|}{1+\|\mathbf{H}_t\|}, \Delta_t\right\} = \frac{1}{2}\|\mathbf{g}_t\|\Delta_t.$$

Now according to Lemma 3, we have

$$\begin{aligned} 1-\rho_t &= \frac{F(\mathbf{x}_t+\mathbf{s}_t) - F(\mathbf{x}_t) - m_t(\mathbf{s}_t)}{-m_t(\mathbf{s}_t)} \leq \frac{\delta_g \Delta_t + \frac{1}{2}\delta_H \Delta_t^2 + \frac{1}{2} L_F {\Delta_t}^3}{\frac{1}{2}\|\mathbf{g}_t\|\Delta_t} \\ &= 2\frac{\delta_g}{\|\mathbf{g}_t\|} + \frac{\delta_H}{\|\mathbf{g}_t\|}\Delta_t + \frac{L_F}{\|\mathbf{g}_t\|}{\Delta_t}^2 \leq \frac{1-\eta}{2} + \frac{\delta_H}{\|\mathbf{g}_t\|}\Delta_t + \frac{L_F}{\|\mathbf{g}_t\|}{\Delta_t}^2. \end{aligned}$$

Let

$$r(t) = \frac{L_F}{\|\mathbf{g}_t\|}t^2 + \frac{\delta_H}{\|\mathbf{g}_t\|}t - \frac{1-\eta}{2}.$$

It is not hard to see that $-\delta_H + \sqrt{\delta_H^2 + 2L_F(1-\eta)\|\mathbf{g}_t\|}/(2L_F)$ is the positive root of $r(t)$. Then, by the fact that $-y + \sqrt{y^2 + 2L_F(1-\eta\|\mathbf{g}\|_t)}/(2L_F)$ is monotonically decreasing for $y \geq 0$ and Condition 1 ($\delta_H < 1$), it follows that

$$\frac{-\delta_H + \sqrt{\delta_H^2 + 2L_F(1-\eta)\|\mathbf{g}_t\|}}{2L_F} \geq \frac{-1 + \sqrt{1 + 2L_F(1-\eta)\|\mathbf{g}_t\|}}{2L_F}.$$

Now, we consider two cases. If $2L_F(1-\eta)\|\mathbf{g}_t\| \leq 1$, it is not hard to show that

$$-1 + \sqrt{1 + 2L_F(1-\eta)\|\mathbf{g}_t\|} \geq \frac{2L_F(1-\eta)\|\mathbf{g}_t\|}{3}.$$

Otherwise, if $2L_F(1-\eta)\|\mathbf{g}_t\| > 1$, then it can be shown that

$$-1 + \sqrt{1 + 2L_F(1-\eta)\|\mathbf{g}_t\|} \geq \sqrt{\frac{L_F(1-\eta)\|\mathbf{g}_t\|}{3}}.$$

By assumption $\Delta_t \leq \min\{\sqrt{(1-\eta)\|\mathbf{g}_t\|/(12L_F)}, (1-\eta)\|\mathbf{g}_t\|/3\}$, so

$$\Delta_t \leq -1 + \sqrt{1 + 2L_F(1-\eta)\|\mathbf{g}_t\|}/(2L_F),$$

and $r(\Delta_t) \leq 0$. Therefore, it follows that

$$1 - \rho_t \leq \frac{1-\eta}{2} + \frac{\delta_H}{\|\mathbf{g}_t\|}\Delta_t + \frac{L_F}{\|\mathbf{g}_t\|}\Delta_t{}^2 \leq (1-\eta) + r(\Delta_t) \leq 1 - \eta,$$

which implies that the iteration t is successful.

Remark 1. It can be easily seen that, if $\delta_g \leq 3\Delta_t/4$, the lemma still holds. Indeed,

$$\delta_g \leq \frac{3}{4}\Delta_t \leq \frac{3}{4}\min\left\{\frac{\|\mathbf{g}_t\|}{1+K_H}, \sqrt{\frac{(1-\eta)\|\mathbf{g}_t\|}{12L_F}}, \frac{(1-\eta)\|\mathbf{g}_t\|}{3}\right\} \leq \frac{(1-\eta)\|\mathbf{g}_t\|}{4}.$$

Although $\delta_g \leq 3\Delta_t/4$ can be looser than what Condition 1 requires, it nonetheless can be used in practice as a rough bound for gradient approximations.

Now, we consider the case when $\|\mathbf{g}_t\| \leq \epsilon_g$. As alluded to earlier in this section, in this case, we have to rely on the negative curvature of the Hessian because dealing with the first-order term in (12) is particularly challenging when $\|\mathbf{g}_t\| < \epsilon_g$. Hence, by solely considering the negative eigenvectors of the Hessian, we drop the first-order term $\langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)\rangle$ in the quadratic model. Lemma 5 gives the corresponding details.

Lemma 5. *Suppose Assumptions 1 and 2 as well as Conditions 1 and 2 hold. Further, suppose, at iteration t, we have* $\|\mathbf{g}_t\| < \epsilon_g, \lambda_{\min}(H_t) < -\epsilon_H$ *and*

$$\Delta_t \leq \left(\frac{1-\eta}{2}\right)\left(\frac{\nu|\lambda_{\min}(\mathbf{H}_t)|}{L_F + 1}\right).$$

Then, the t^{th} *is successful; that is,* $\Delta_{t+1} = \gamma\Delta_t$.

Proof. Here, by Condition 2, we have $\mathbf{s}_t = \mathbf{s}_t^E$, which, by (11), implies $-m_t(\mathbf{s}_t) \geq \nu|\lambda_{\min}(\mathbf{H}_t)|\Delta_t{}^2/2$. Hence, recalling (12), we have

$$F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t) - m_t(\mathbf{s}_t) \leq \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)\rangle + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \frac{1}{2}L_F\|\mathbf{s}_t\|^3.$$

Because either $\mathbf{s}_t$ or $-\mathbf{s}_t$ could be a searching direction, at least one of

$$\langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)\rangle \le 0 \qquad \text{or} \qquad \langle -\mathbf{s}_t, \nabla F(\mathbf{x}_t)\rangle \le 0,$$

is true. Without loss of generality, assume $\langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)\rangle \le 0$. Hence,

$$F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t) - m_t(\mathbf{s}_t) \le \frac{1}{2}\delta_H \|\mathbf{s}_t\|^2 + \frac{1}{2}L_F\|\mathbf{s}_t\|^3.$$

Next, suppose $\Delta_t \le (1-\eta)\nu\epsilon_H/2$, which, from (8), implies that $\delta_H \le (1-\eta)\nu\epsilon_H/2$. We have

$$\begin{aligned} 1-\rho_t = \frac{F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t) - m_t(\mathbf{s}_t)}{-m_t(\mathbf{s}_t)} &\le \frac{\delta_H {\Delta_t}^2/2 + L_F{\Delta_t}^3/2}{\nu|\lambda_{\min}(\mathbf{H}_t)|{\Delta_t}^2/2} = \frac{\delta_H + L_F\Delta_t}{\nu|\lambda_{\min}(\mathbf{H}_t)|} \\ &\le \frac{(1-\eta)\nu\epsilon_H/2 + L_F(1-\eta)\nu|\lambda_{\min}(H_t)|/(2(L_F+1))}{\nu|\lambda_{\min}(\mathbf{H}_t)|} < 1-\eta. \end{aligned}$$

Now, consider $\Delta_t \ge (1-\eta)\nu\epsilon_H/2$, which, from (8), implies that $\delta_H \le \Delta_t$. Similarly, we have

$$1-\rho_t = \frac{F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t) - m_t(\mathbf{s}_t)}{-m_t(\mathbf{s}_t)} \le \frac{\delta_H {\Delta_t}^2/2 + L_F{\Delta_t}^3/2}{\nu|\lambda_{\min}(\mathbf{H}_t)|{\Delta_t}^2/2} = \frac{(L_F+1)\Delta_t}{\nu|\lambda_{\min}(\mathbf{H}_t)|} < (1-\eta)/2 < 1-\eta.$$

Hence, in both cases, we have $\rho_t \ge \eta$, and the iteration is successful. ■

Based on Lemmas 4 and 5, the following lemma helps to get the lower bound of Δ_t, whose proof can be found in Xu et al. (2019).

Lemma 6. *Under Assumptions 1 and 2 and Conditions 1 and 2, for Algorithm 1 and for all t, we have*

$$\Delta_t \ge \frac{1}{\gamma}\min\left\{\frac{\epsilon_g}{1+K_H}, \sqrt{\frac{(1-\eta)\epsilon_g}{12L_H}}, \frac{(1-\eta)\epsilon_g}{3}, \frac{(1-\eta)\nu\epsilon_H}{2(L_F+1)}\right\}.$$

As a consequence, we now can give the upper bound on the number of successful iterations.

Lemma 7 (Successful Iterations)**.** *Let $\mathcal{T}_{succ}$ denote the set of all the successful iterations before Algorithm 1 stops. Under Assumptions 1 and 2 and Conditions 1 and 2, the number of successful iterations is upper-bounded by*

$$|\mathcal{T}_{succ}| \le \frac{F(\mathbf{x}_0) - F(\mathbf{x}^*)}{C\epsilon_H \min\{{\epsilon_g}^2, {\epsilon_H}^2\}},$$

where C is a constant depending on L_F, K_H, η, ν.

Proof. Suppose Algorithm 1 doesn't terminate at iteration t. Then, either $\|\mathbf{g}_t\| \ge \epsilon_g$ or $\lambda_{\min}(\mathbf{H}_t) \le -\epsilon_H$. If $\|\mathbf{g}_t\| \ge \epsilon_g$, according to (9), we have

$$-m_t(\mathbf{s}_t) \ge \frac{1}{2}\|\mathbf{g}_t\| \min\left\{\frac{\|\mathbf{g}_t\|}{1+\|\mathbf{H}_t\|}, \Delta_t\right\} \ge \frac{1}{2}\epsilon_g \min\left\{\frac{\epsilon_g}{1+K_H}, C_0\epsilon_g, C_1\epsilon_H\right\} \ge C_2\epsilon_g \min\{\epsilon_g, \epsilon_H\}.$$

Similarly, in the second case $\lambda_{\min}(\mathbf{H}_t) \le -\epsilon_H$, from (11),

$$-m_t(\mathbf{s}_t) \ge \frac{1}{2}\nu\|\lambda_{\min}(\mathbf{H}_t)\|{\Delta_t}^2 \ge C_3\epsilon_H \min\{{\epsilon_g}^2, {\epsilon_H}^2\}.$$

Because $F(\mathbf{x}_t)$ is monotonically decreasing as t increases, we have

$$\begin{aligned} F(\mathbf{x}_0) - F(\mathbf{x}^*) &\ge \sum_{t=0}^{\infty} F(\mathbf{x}_t) - F(\mathbf{x}_{t+1}) \ge \sum_{t\in\mathcal{T}_{\text{succ}}} F(\mathbf{x}_t) - F(\mathbf{x}_{t+1}) \\ &\ge \eta \sum_{t\in\mathcal{T}_{\text{succ}}} \min\{C_2\epsilon_g \min\{\epsilon_g, \epsilon_H\}, C_3\epsilon_H \min\{{\epsilon_g}^2, {\epsilon_H}^2\}\} \\ &\ge \|\mathcal{T}_{\text{succ}}\|C\epsilon_H \min\{{\epsilon_g}^2, {\epsilon_H}^2\}. \end{aligned}$$

Because one of these cases must happen for a successful iteration, it follows that

$$\|\mathcal{T}_{\text{succ}}\| \le \frac{F(\mathbf{x}_0) - F(\mathbf{x}^*)}{C\epsilon_H \min\{\epsilon_g{}^2, \epsilon_H{}^2\}}.$$

■

Using the preceding lemma, the proof of following theorem can be found in Xu et al. (2019).

Theorem 1 (Optimal Complexity of Algorithm 1)**.** *Let Assumption 1 hold and suppose that $\mathbf{g}_t$ and $\mathbf{H}_t$ satisfy Assumption 2 with δ_g and δ_H under Condition 1. If the approximate solution to the subproblem (6) satisfies Condition 2, then Algorithm 1 terminates after at most*

$$T \in \mathcal{O}\left(\max\{\epsilon_g{}^{-2}\epsilon_H{}^{-1}, \epsilon_H{}^{-3}\}\right),$$

iterations

The worst iteration complexity of Theorem 1 matches the bound obtained in Conn et al. (2000), Cartis et al. (2012), and Xu et al. (2019), which is known to be optimal in the worst-case sense (Cartis et al. 2012). Further, it follows immediately that the terminating points of Algorithm 1 satisfy $\|\mathbf{g}_T\| \le \epsilon_g + \delta_g$ and $\lambda_{\min}(\mathbf{H}_T) \ge -\epsilon_H - \delta_h$; that is, $\mathbf{x}_T$ is a $(\epsilon_g + \delta_g, \epsilon_H + \delta_h)$-optimal solution of (1).

2.2. Inexact ARC

The inexact ARC algorithm is given in Algorithm 2. Every iteration of Algorithm 2 involves an approximate solution to the following subproblem:

$$\mathbf{s}_t \approx \underset{\mathbf{s}\in\mathbb{R}^d}{\operatorname{argmin}}\, m_t(\mathbf{s}), \tag{13a}$$

where

$$m_t(\mathbf{s}) \triangleq \begin{cases} \langle \mathbf{g}_t, \mathbf{s}\rangle + \dfrac{1}{2}\langle \mathbf{s}, \mathbf{H}_t\mathbf{s}\rangle + \dfrac{\sigma_t}{3}\|\mathbf{s}\|^3, & \|\mathbf{g}_t\| \ge \epsilon_g \\ \dfrac{1}{2}\langle \mathbf{s}, \mathbf{H}_t\mathbf{s}\rangle + \dfrac{\sigma_t}{3}\|\mathbf{s}\|^3, & \text{Otherwise} \end{cases}. \tag{13b}$$

Similar to Section 2.1, our analysis for inexact ARC also involves Cauchy and Eigen points obtained from (13) as follows.

Definition 4 (Cauchy Point for Algorithm 2)**.** When $\|\mathbf{g}_t\| \ge \epsilon_g$, the Cauchy point for Algorithm 2 is obtained from (13) as

$$\mathbf{s}_t^C = -\alpha^C\mathbf{g}_t, \quad \alpha^C = \underset{\alpha\ge 0}{\operatorname{argmin}}\, m_t(-\alpha\mathbf{g}_t). \tag{14a}$$

Definition 5 (Eigen Point for Algorithm 2)**.** When $\lambda_{\min}(\mathbf{H}_t) \le -\epsilon_H$, the Eigen point for Algorithm 2 is obtained from (13) as

$$\mathbf{s}_t^E = \alpha^E\mathbf{u}_t, \quad \alpha^E = \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}}\, m_t(\alpha\mathbf{u}_t), \tag{14b}$$

where $\mathbf{u}_t$ is an approximation to the corresponding negative curvature direction; that is, for some $0 < \nu < 1$,

$$\langle \mathbf{u}_t, \mathbf{H}_t\mathbf{u}_t\rangle \le \nu\lambda_{\min}(\mathbf{H}_t) \ \text{ and } \ \|\mathbf{u}_t\| = 1.$$

Note that, because both $\mathbf{s}_t^C$ and $\mathbf{s}_t^E$ are line minimizers of $m_t(\mathbf{s})$ along the directions $-\mathbf{g}_t$ and $\mathbf{u}_t$, respectively, they satisfy

$$\langle -\mathbf{g}_t, \nabla m_t(\mathbf{s}_t^C)\rangle = \langle \mathbf{s}_t^C, \nabla m_t(\mathbf{s}_t^C)\rangle = 0,$$
$$\langle \mathbf{u}_t, \nabla m_t(\mathbf{s}_t^E)\rangle = \langle \mathbf{s}_t^E, \nabla m_t(\mathbf{s}_t^E)\rangle = 0.$$

Further properties of Cauchy and Eigen points for the cubic problem can be found in Lemmas 9 and 10.

As we show, the worst-case iteration complexity of inexact ARC depends on how accurately we approximate the gradient and Hessian as well as the problem solves. In Section 2.2.1, we show that under *nearly* minimum requirement of the gradient and Hessian approximation (Condition 3), the inexact ARC can achieve *suboptimal* complexity $\mathcal{O}(\max\{\epsilon_g^{-2}, \epsilon_H^{-3}\})$. In Section 2.2.2, we then show that, under the more restricted approximation condition (Condition 5), the *optimal* worst-case complexity $\mathcal{O}(\max\{\epsilon_g^{-1.5}, \epsilon_H^{-3}\})$ can be recovered.

2.2.1. Suboptimal Complexity for Algorithm 2. In this section, we provide sufficient conditions on approximating the gradient and Hessian as well as the subproblem solves for inexact ARC to achieve the suboptimal complexity $\mathcal{O}(\max\{\epsilon_g^{-2}, \epsilon_H^{-3}\})$.

First, similar to Section 2.1, we require that the estimates of the gradient and Hessian satisfy the following condition.

Condition 3 (Gradient and Hessian Approximation for Algorithm 2)**.** *Given the termination criteria, ϵ_g, ϵ_H, in Algorithm 2, we require the inexact gradient and Hessian to satisfy*

$$\delta_g \le \left(\frac{1-\eta}{12}\right)\max\{\epsilon_g, \|\mathbf{g}_t\|\}, \quad \text{and} \quad \delta_H \le \left(\frac{1-\eta}{6}\right)\min\left\{\nu \max\{-\lambda_{\min}(\mathbf{H}_t), \epsilon_H\}, \sqrt{2L_F\epsilon_g}\right\}. \tag{15}$$

It is easy to see that $\delta_g \in \mathcal{O}(\epsilon_g)$, $\delta_H \in \mathcal{O}(\min\{\sqrt{\epsilon_g}, \epsilon_H\})$. Similar constraints on δ_H have appeared in several previous works, for example, Tripuraneni et al. (2018) and Xu et al. (2019). These are nearly minimum requirements for the approximation to determine whether the iteration satisfies (ϵ_g, ϵ_H)-optimality (Definition 1). In the case when $\epsilon_H = \mathcal{O}(\sqrt{\epsilon_g})$, Condition 3 is indeed the minimum requirement. We note that the conditions on δ_g and δ_H are adaptive in that, for large $\|\mathbf{g}_t\|$ and $-\lambda_{\min}(\mathbf{H}_t)$, they amount to relative error conditions on the approximate gradient and Hessian, respectively. In fact the condition on δ_g is very similar to that in Condition 1. Of course, in such cases, these conditions cannot be a priori enforced in a straightforward manner. Nonetheless, they qualitatively indicate that, in regions with large gradient and negative curvature, one can rely on loose approximations of the gradient and Hessian, respectively. As the algorithm progresses toward convergence, Condition 3 implies that, ultimately, we must seek to have $\delta_g \in \mathcal{O}(\epsilon_g)$ and $\delta_H \in \mathcal{O}(\min\{\sqrt{\epsilon_g}, \epsilon_H\})$.

As for solving the subproblem, we require the following.

Condition 4. (Approximate Solution of (13) for Algorithm 2). *We use the same trial steps as in Condition 2 but with $\mathbf{s}_t^C$ and $\mathbf{s}_t^E$ as in Definitions 4 and 5, respectively.*

Condition 4 implies that, when the gradient is large enough, we take the Cauchy step. Otherwise, we update along the Eigen point direction.

Under Assumptions 1 and 2 as well as Conditions 3 and 4, we now present the proof of suboptimal complexity of Algorithm 2. First let's denote $\mathcal{T}_{\text{succ}}$ as the set of all the successful iterations and $\mathcal{T}_{\text{fail}}$ as the set of all the failure iterations. Now we upper bound the iteration complexity $T := \|\mathcal{T}_{\text{succ}}\| + \|\mathcal{T}_{\text{fail}}\|$. First, we present the following lemma that gives an upper bound of $\|\mathcal{T}_{\text{fail}}\|$.

Lemma 8. *In Algorithm 2, suppose we have $\sigma_t \le C$, where C is some constant, for all the iterations t before it stops. Then, we have $\|\mathcal{T}_{fail}\| \le \|\mathcal{T}_{succ}\| + \mathcal{O}(1)$.*

Proof. Because $\sigma_t \le C$, then $\sigma_T = \sigma_0\gamma^{\|\mathcal{T}_{\text{succ}}\| - \|\mathcal{T}_{\text{fail}}\|} \le C$. Then, we immediately obtain

$$\|\mathcal{T}_{\text{fail}}\| \le \log(C/\sigma_0)/\log\gamma + \|\mathcal{T}_{\text{fail}}\| = \|\mathcal{T}_{\text{succ}}\| + \mathcal{O}(1).$$

■

Now, for the rest of the analysis, we first show that there is a uniform upper bound for all σ_t and, subsequently, we obtain a bound on the number of all the successful iterations.

We now present Lemma 9, which is very similar to Xu et al. (2019, lemma 6), but is slightly more refined. The main difference lies in the quantity K_t defined in Lemma 9. In Xu et al. (2019, lemma 6), a simple global upper bound of this quantity is used. However, here, we retain its local nature, which is found to be crucial in proving Lemma 12. This is a subtle distinction that arises as a result of using gradient approximations here compared with exact gradients in Xu et al. (2019).

Lemma 9 (Cauchy Point)**.** *When $\|\mathbf{g}_t\| \ge \epsilon_g$, let*

$$\mathbf{s}_t^C = \arg\min_{\alpha \ge 0} m_t(-\alpha \mathbf{g}_t).$$

Then, we have

$$\|\mathbf{s}_t^C\| = \frac{1}{2\sigma_t}\left(\sqrt{K_t^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_t\right), \tag{16a}$$

$$-m_t\left(\mathbf{s}_t^C\right) \geq \max\left\{\frac{1}{12}\|\mathbf{s}_t^C\|^2\left(\sqrt{K_t^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_t\right), \frac{\|\mathbf{g}_t\|}{2\sqrt{3}}\min\left\{\frac{\|\mathbf{g}_t\|}{|K_t|}, \frac{\|\mathbf{g}_t\|}{\sqrt{\sigma_t\|\mathbf{g}_t\|}}\right\}\right\}, \tag{16b}$$

where $K_t = \langle \mathbf{H}_t\mathbf{g}_t, \mathbf{g}_t\rangle / \|\mathbf{g}_t\|^2$.

Proof. The proof is organized as follows. We first use the definition of Cauchy point to get an expression in terms of $\mathbf{s}_t^C$. Subsequently, we use the fact that $m_t(\mathbf{s}_t^C) \leq m_t(\alpha\mathbf{g}_t)$, $\forall \alpha \geq 0$ to bound $m_t(\mathbf{s}_t^C)$ by leveraging the quadratic form of $m_t(\alpha\mathbf{g}_t)$ in terms of α. First, we have

$$\left\langle \mathbf{g}_t, \mathbf{s}_t^C \right\rangle + \left\langle \mathbf{s}_t^C, \mathbf{H}_t\mathbf{s}_t^C \right\rangle + \sigma_t\left\|\mathbf{s}_t^C\right\|^3 = 0.$$

Because $\mathbf{s}_t^C = -\alpha\mathbf{g}_t$ for some $\alpha > 0$,

$$-\alpha\|\mathbf{g}_t\|^2 + \alpha^2\langle \mathbf{g}_t, \mathbf{H}_t\mathbf{g}_t\rangle + \sigma_t\alpha^3\|\mathbf{g}_t\|^3 = 0.$$

We can find an explicit formula for such α by finding the roots of the quadratic function

$$r(\alpha) = -\|\mathbf{g}_t\|^2 + \alpha\langle \mathbf{g}_t, \mathbf{H}_t\mathbf{g}_t\rangle + \sigma_t\alpha^2\|\mathbf{g}_t\|^3.$$

We have

$$\alpha = \frac{-\langle \mathbf{g}_t, \mathbf{H}_t\mathbf{g}_t\rangle + \sqrt{\langle \mathbf{g}_t, \mathbf{H}_t\mathbf{g}_t\rangle^2 + 4\sigma_t\|\mathbf{g}_t\|^5}}{2\sigma_t\|\mathbf{g}_t\|^3},$$

and

$$2\alpha\sigma_t\|\mathbf{g}_t\| = \sqrt{K_t^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_t.$$

Hence, it follows that

$$\|\mathbf{s}_t^C\| = \alpha\|\mathbf{g}_t\| = \frac{1}{2\sigma_t}\left(\sqrt{K_t^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_t\right).$$

Now, from Cartis et al. (2012, lemma 2.1), we get

$$-m_t\left(\mathbf{s}_t^C\right) \geq \frac{1}{6}\sigma_t\|\mathbf{s}_t^C\|^3 = \frac{1}{6}\sigma_t\|\mathbf{s}_t^C\|^2\alpha\|\mathbf{g}_t\| = \frac{1}{12}\|\mathbf{s}_t^C\|^2\left(\sqrt{K_t^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_t\right).$$

Alternatively, we have

$$\begin{aligned} m_t\left(\mathbf{s}_t^C\right) \leq m_t\left(-\alpha\mathbf{g}_t\right) &= -\alpha\|\mathbf{g}_t\|^2 + \frac{1}{2}\alpha^2\langle \mathbf{g}_t, \mathbf{H}_t\mathbf{g}_t\rangle + \frac{\alpha^3}{3}\sigma_t\|\mathbf{g}_t\|^3 \\ &= \frac{\alpha\|\mathbf{g}_t\|^2}{6}\left(-6 + 3\alpha K_t + 2\alpha^2\sigma_t\|\mathbf{g}_t\|\right). \end{aligned}$$

Consider the quadratic part,

$$r(\alpha) = -6 + 3\alpha K_t + 2\alpha^2\sigma_t\|\mathbf{g}_t\|.$$

We have $r(\alpha) \leq 0$ for $\alpha \in [0, \bar{\alpha}]$, where

$$\bar{\alpha} = \frac{-3K_t + \sqrt{9K_t^2 + 48\sigma_t\|\mathbf{g}_t\|}}{4\sigma_t\|\mathbf{g}_t\|}.$$

We can express $\bar{\alpha}$ as

$$\bar{\alpha} = \frac{12}{3K_t + \sqrt{9K_t^2 + 48\sigma_t\|\mathbf{g}_t\|}}.$$

Note that

$$\sqrt{9K_t^2 + 48\sigma_t\|\mathbf{g}_t\|} \le 3|K_t| + 4\sqrt{3\sigma_t\|\mathbf{g}_t\|} \le 8\sqrt{3}\max\left\{|K_t|, \sqrt{\sigma_t\|\mathbf{g}_t\|}\right\}.$$

Also,

$$3K_t \le 2\sqrt{3}\max\left\{|K_t|, \sqrt{\sigma_t\|\mathbf{g}_t\|}\right\} \le 4\sqrt{3}\max\left\{|K_t|, \sqrt{\sigma_t\|\mathbf{g}_t\|}\right\}.$$

Hence, defining

$$\alpha_0 = \frac{1}{\sqrt{3}\max\left\{|K_t|, \sqrt{\sigma_t\|\mathbf{g}_t\|}\right\}},$$

it is clear that $0 \le \alpha_0 \le \bar{\alpha}$. With α_0, we have

$$r(\alpha_0) \le 2/3 + 3/\sqrt{3} - 6 \le -3.$$

So, finally, we get

$$\begin{aligned} m_t\left(\mathbf{s}_t^C\right) \le \frac{-3\|\mathbf{g}_t\|^2}{6\sqrt{3}} \frac{1}{\max\left\{|K_t|, \sqrt{\sigma_t\|\mathbf{g}_t\|}\right\}} &= \frac{-\|\mathbf{g}_t\|^2}{2\sqrt{3}}\min\left\{\frac{1}{|K_t|}, \frac{1}{\sqrt{\sigma_t\|\mathbf{g}_t\|}}\right\} \\ &= \frac{-\|\mathbf{g}_t\|}{2\sqrt{3}}\min\left\{\frac{\|\mathbf{g}_t\|}{|K_t|}, \frac{\|\mathbf{g}_t\|}{\sqrt{\sigma_t\|\mathbf{g}_t\|}}\right\}. \end{aligned}$$

■

When $\mathbf{H}_t$ has a negative eigenvalue, the Eigen point has the following properties.

Lemma 10 (Eigen Point)**.** *Suppose $\lambda_{\min}(\mathbf{H}_t) < 0$, and for some $\nu \in (0, 1]$, let*

$$\mathbf{s}_t^E = \arg\min_{\alpha \in R} m_t(\alpha \mathbf{u}_t),$$

where $\mathbf{u}_t$ is the approximate most negative eigenvector defined as

$$\langle \mathbf{u}_t, \mathbf{H}_t\mathbf{u}_t \rangle \le \nu\lambda_{\min}(\mathbf{H}_t)\|\mathbf{u}_t\|^2 \le 0.$$

We have

$$\|\mathbf{s}_t^E\| \ge \frac{\nu\|\lambda_{\min}(\mathbf{H}_t)\|}{\sigma_t}, \tag{17a}$$

$$-m_t\left(\mathbf{s}_t^E\right) \ge \frac{\nu|\lambda_{\min}(\mathbf{H}_t)|}{6}\|\mathbf{s}_t^E\|^2. \tag{17b}$$

Proof. Again, we know that

$$\left\langle \mathbf{g}_t, \mathbf{s}_t^E \right\rangle + \left\langle \mathbf{s}_t^E, \mathbf{H}_t\mathbf{s}_t^E \right\rangle + \sigma_t\left\|\mathbf{s}_t^E\right\|^3 = 0.$$

Meanwhile, because $-\mathbf{s}_t$ would keep the last two terms as the same value, without loss of generality, we could assume $\langle \mathbf{g}_t, \mathbf{s}_t^E \rangle \le 0$, which means

$$\left\langle \mathbf{s}_t^E, \mathbf{H}_t\mathbf{s}_t^E \right\rangle + \sigma_t\left\|\mathbf{s}_t^E\right\|^3 \ge 0.$$

Now, from Cartis et al. (2012, lemma 2.1),

$$-m_t(\mathbf{s}_t) \geq \frac{1}{6}\sigma_t\|\mathbf{s}_t\|^3 \geq -\frac{1}{6}\langle \mathbf{s}_t^E, \mathbf{H}_t\mathbf{s}_t^E\rangle \geq \frac{1}{6}\nu|\lambda_{\min}(H_t)|\|\mathbf{s}_t^E\|^2.$$

■

The following lemma gives a bound on the difference between the decrease of the objective function and the value of the quadratic model $m(\mathbf{s}_t)$. This lemma can be easily obtained by the smoothness assumption of the objective function; the detailed proof is included in the appendix.

Lemma 11. *Under Assumptions 1 and 2, we have*

$$F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t) - m_t(\mathbf{s}_t) \leq \begin{cases} \delta_g\|\mathbf{s}_t\| + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{L_F}{2} - \frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3, & \|\mathbf{g}_t\| \geq \epsilon_g, \\ \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)\rangle + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{L_F}{2} - \frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3, & \|\mathbf{g}_t\| < \epsilon_g. \end{cases} \tag{18}$$

Based on these lemmas, the following lemma shows that iteration t is successful when $\|\mathbf{g}_t\| \geq \epsilon_g$.

Lemma 12. *Suppose* Assumptions 1 *and* 2 *and* Condition 3 *hold. Further, suppose at iteration t, we have $\|\mathbf{g}_t\| \geq \epsilon_g$ and $\sigma_t \geq 2L_F$. Then, the iteration t is successful; that is, $\sigma_{t+1} = \sigma_t/\gamma$.*

Proof. Using Lemma 11, we get

$$\begin{aligned} F\left(\mathbf{x}_t + \mathbf{s}_t^C\right) - F(\mathbf{x}_t) - m_t\left(\mathbf{s}_t^C\right) &\leq \delta_g\|\mathbf{s}_t^C\| + \frac{1}{2}\delta_H\|\mathbf{s}_t^C\|^2 + \left(\frac{L_F}{2} - \frac{\sigma_t}{3}\right)\|\mathbf{s}_t^C\|^3 \\ &\leq \delta_g\|\mathbf{s}_t^C\| + \frac{1}{2}\delta_H\|\mathbf{s}_t^C\|^2, \end{aligned}$$

because $\sigma_t \geq 2L_F$. We divide it into two cases.

First, if $K_t = \langle \mathbf{H}_t\mathbf{g}_t, \mathbf{g}_t\rangle/\|\mathbf{g}_t\|^2 \leq 0$, then, from (16a), it follows that

$$\|\mathbf{s}_t^C\| \geq \frac{1}{2\sigma_t}\sqrt{4\sigma_t\|\mathbf{g}_t\|} = \sqrt{\|\mathbf{g}_t\|/\sigma_t}.$$

Using Cartis et al. (2012, lemma 2.1), we get

$$\begin{aligned} 1 - \rho_t = \frac{F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t) - m_t(\mathbf{s}_t)}{-m_t(\mathbf{s}_t)} &\leq \frac{\delta_g\|\mathbf{s}_t^C\| + \frac{1}{2}\delta_H\|\mathbf{s}_t^C\|^2}{\frac{\sigma_t\|\mathbf{s}_t^C\|^3}{6}} = \frac{\delta_g + \frac{1}{2}\delta_H\|\mathbf{s}_t^C\|}{\frac{\sigma_t\|\mathbf{s}_t^C\|^2}{6}} \\ &\leq \frac{6\delta_g}{\|\mathbf{g}_t\|} + \frac{3\delta_H}{\sqrt{2\epsilon_g L_F}} \leq \frac{1-\eta}{2} + \frac{1-\eta}{2} = 1 - \eta, \end{aligned}$$

where the last inequality follows from the condition on δ_g and δ_H.

For the second case in which $K_t > 0$, from (16a) in Lemma 9, it follows that

$$\|\mathbf{s}_t^C\| = \frac{\sqrt{K_t^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_t}{2\sigma_t} = \frac{2\|\mathbf{g}_t\|}{\sqrt{K_t^2 + 4\sigma_t\|\mathbf{g}_t\|} + K_t}.$$

Now, we consider two cases: (a) $K_t^2 \geq \sigma_t\|\mathbf{g}_t\|$ and (b) $K_t^2 \leq \sigma_t\|\mathbf{g}_t\|$.

a. When $K_H^2 \geq K_t^2 \geq \sigma_t\|\mathbf{g}_t\|$, from the preceding equality, we have

$$\|\mathbf{s}_t^C\| \leq \frac{\|\mathbf{g}_t\|}{K_t}.$$

Meanwhile, because $K_t^2 \geq \sigma_t\|\mathbf{g}_t\|$, from Lemma 9, we have

$$-m_t\left(\mathbf{s}_t^C\right) \geq \frac{\|\mathbf{g}_t\|}{2\sqrt{3}}\min\left\{\frac{\|\mathbf{g}_t\|}{|K_t|}, \frac{\|\mathbf{g}_t\|}{\sqrt{\sigma_t\|\mathbf{g}_t\|}}\right\} = \frac{\|\mathbf{g}_t\|^2}{2\sqrt{3}K_t}.$$

Combining the inequalities together, it follows that

$$1-\rho_t=\frac{F(\mathbf{x}_t+\mathbf{s}_t)-F(\mathbf{x}_t)-m_t(\mathbf{s}_t)}{-m_t(\mathbf{s}_t)}\le\frac{\delta_g\|\mathbf{s}_t^C\|+\frac{1}{2}\delta_H\|\mathbf{s}_t^C\|^2}{\frac{\|\mathbf{g}_t\|^2}{2\sqrt{3}K_t}}\le\frac{\delta_g\frac{\|\mathbf{g}_t\|}{K_t}+\frac{1}{2}\delta_H\left(\frac{\|\mathbf{g}_t\|}{K_t}\right)^2}{\frac{\|\mathbf{g}_t\|^2}{2\sqrt{3}K_t}}$$

$$=\frac{2\sqrt{3}\delta_g}{\|\mathbf{g}_t\|}+\frac{\sqrt{3}\delta_H}{K_t}\le\frac{2\sqrt{3}\delta_g}{\|\mathbf{g}_t\|}+\frac{\sqrt{3}\delta_H}{\sqrt{2L_F\epsilon_g}}\le\frac{1-\eta}{2}+\frac{1-\eta}{2}=1-\eta.$$

b. When $K_t^2\le\sigma_t\|\mathbf{g}_t\|$, we have

$$\|\mathbf{s}_t^C\|\le\frac{\|\mathbf{g}_t\|}{\sqrt{\|\mathbf{g}_t\sigma_t\|}},$$

and

$$-m_t(\mathbf{s}_t^C)\ge\frac{\|\mathbf{g}_t\|}{2\sqrt{3}}\min\left\{\frac{\|\mathbf{g}_t\|}{|K_t|},\frac{\|\mathbf{g}_t\|}{\sqrt{\sigma_t\|\mathbf{g}_t\|}}\right\}\ge\frac{\|\mathbf{g}_t\|^{3/2}}{2\sqrt{3}\sqrt{\sigma_t}}.$$

Then,

$$1-\rho_t=\frac{F(\mathbf{x}_t+\mathbf{s}_t)-F(\mathbf{x}_t)-m_t(\mathbf{s}_t)}{-m_t(\mathbf{s}_t)}\le\frac{\delta_g\|\mathbf{s}_t^C\|+\frac{1}{2}\delta_H\|\mathbf{s}_t^C\|^2}{\frac{\|\mathbf{g}_t\|^{3/2}}{2\sqrt{3}\sqrt{\sigma_t}}}=\frac{2\sqrt{3}\delta_g}{\|\mathbf{g}_t\|}+\frac{\sqrt{3}\delta_H}{\sqrt{\sigma_t\epsilon_g}}$$

$$\le\frac{2\sqrt{3}\delta_g}{\|\mathbf{g}_t\|}+\frac{\sqrt{3}\delta_H}{\sqrt{2L_F\epsilon_g}}\le\frac{1-\eta}{2}+\frac{1-\eta}{2}=1-\eta.$$

From these, it follows that the t^{th} iteration is successful; that is, $\sigma_{t+1}=\sigma_t/\gamma$ when $\|\mathbf{g}_t\|\ge\epsilon_g$. ■

The following lemma, whose proof is similar to Xu et al. (2019, lemma 9), helps bound $F(\mathbf{x}_t+\mathbf{s}_t)-F(\mathbf{x}_t)-m_t(\mathbf{s}_t)$ when the Hessian has negative eigenvalues; the detailed proof is included in the appendix.

Lemma 13. (Xu et al. 2019, Lemma 9). *Suppose Assumptions 1 and 2 and Condition 3 hold and $\sigma_t\ge 2L_F$. Then, if $\lambda_{\min}(\mathbf{H}_t)<-\epsilon_H$, we have*

$$\frac{\delta_H}{2}\|\mathbf{s}_t\|^2+\left(\frac{L_F}{2}-\frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3\le\frac{\delta_H}{2}\|\mathbf{s}_t^E\|^2.$$

Then, the following lemma shows Eigen points also yield a descent similarly as in Lemma 5.

Lemma 14. *Suppose Assumptions 1 and 2 and Conditions 3 and 4 hold. Further, suppose, at iteration t, we have $\lambda_{\min}(\mathbf{H}_t)<-\epsilon_H$, $\|\mathbf{g}_t\|\le\epsilon_g$ and $\sigma_t\ge 2L_F$. Then, iteration t is successful; that is, $\sigma_{t+1}=\sigma_t/\gamma$.*

Proof. If $\lambda_{\min}(\mathbf{H}_t)<-\epsilon_H$ and $\|\mathbf{g}_t\|\le\epsilon_g$, recall that our subproblem is

$$m_t(\mathbf{s})=\frac{1}{2}\langle\mathbf{s},\mathbf{H}_t\mathbf{s}\rangle+\frac{\sigma_t}{3}\|\mathbf{s}\|^3,$$

and we pick the Eigen point direction, that is, $\mathbf{s}_t=\mathbf{s}_t^E$. Now, it is clear that, if $\mathbf{s}_t$ is an approximate solution of the preceding problem, then so is $-\mathbf{s}_t$. Similar to Lemma 5, without loss of generality, assume $\langle\mathbf{s}_t,\nabla F(\mathbf{x}_t)\rangle\le 0$. Then, according to (18),

$$F(\mathbf{x}_t+\mathbf{s}_t)-F(\mathbf{x}_t)-m_t(\mathbf{s}_t)\le\frac{\delta_H}{2}\|\mathbf{s}_t\|^2+\frac{(L_F-\sigma_t/3)}{2}\|\mathbf{s}_t\|^3.$$

Therefore, according to (17b) and Lemma 13,

$$\begin{aligned}1-\rho_t &= \frac{F(\mathbf{x}_t+\mathbf{s}_t)-F(\mathbf{x}_t)-m_t(\mathbf{s}_t)}{-m_t(\mathbf{s}_t)} \le \frac{\delta_H\|\mathbf{s}_t\|^2+(L_F-\sigma_t/3)\|\mathbf{s}_t\|^3}{-2m_t(\mathbf{s}_t)}\\ &\le \frac{\delta_H\|\mathbf{s}_t\|^2/2}{\nu|\lambda_{\min}(\mathbf{H}_t)|\|\mathbf{s}_t\|^2/6} = \frac{3\delta_H}{\nu\|\lambda_{\min}(\mathbf{H}_t)\|} \le 1-\eta,\end{aligned}$$

which means the iteration t is successful. ■

With the help of these lemmas, we can now show an upper bound for σ_t as in Lemma 15.

Lemma 15. *Under Assumptions 1 and 2 and Conditions 3 and 4, we have $\sigma_t \le \max\{2\gamma L_F, \sigma_0\}$ for all t.*

Proof. We consider two cases. First, suppose $\sigma_0 \le 2\gamma L_F$. We prove the claim in this case by contradiction. Suppose that the t^{th} iteration is the first unsuccessful iteration such that $\sigma_{t+1} = \gamma\sigma_t \ge 2\gamma L_F$, which implies that $\sigma_t \ge 2L_F$. However, according to Lemmas 12 and 14, respectively, if $\|\mathbf{g}_t\| \ge \epsilon_g$ or $\lambda_{\min}(\mathbf{H}_t) \le -\epsilon_H$, then the iteration is successful, and we must have $\sigma_{t+1} = \sigma_t/\gamma \le \sigma_t$, which is a contradiction. Second, consider the case in which $\sigma_0 > 2\gamma L_F$. According to Lemmas 12 and 14, any iteration t with $\sigma_t \ge 2L_F$ is successful, which implies that $\sigma_t \le \sigma_0, ; \forall t$. ■

Now, we upper bound the number of all successful iterations $\|\mathcal{T}_{\text{succ}}\|$, which is shown in Lemma 16. The proof is similar to Xu et al. (2019, lemma 13).

Lemma 16 (Successful Iterations)**.** *Under Assumptions 1 and 2 and Conditions 3 and 4, the number of successful iterations is upper bounded by*

$$\|\mathcal{T}_{succ}\| \le \left(\frac{F(x_0)-F(x^*)}{C}\right)\max\{{\epsilon_g}^{-2}, {\epsilon_H}^{-3}\}.$$

Based on these lemmas, Theorem 2 follows.

Theorem 2 (Complexity of Algorithm 2)**.** *Let Assumption 1 hold and consider any $0 < \epsilon_g, \epsilon_H < 1$. Further, suppose that $\mathbf{g}_t$ and $\mathbf{H}_t$ satisfy Assumption 2 with δ_g and δ_H under Condition 3. If the approximate solution to the subproblem (13) satisfies Condition 4, then Algorithm 2 terminates after at most*

$$T \in \mathcal{O}\left(\max\{{\epsilon_g}^{-2}, {\epsilon_H}^{-3}\}\right),$$

iterations.

Remark 3. To obtain similar suboptimal iteration complexity, the sufficient condition on approximating a Hessian in Xu et al. (2019) requires that $\delta_H \in \mathcal{O}(\min\{\epsilon_g, \epsilon_H\})$, which is stronger than Condition 3.

2.2.2. Optimal Complexity for Algorithm 2.

In this section, we show that, by better approximation of the gradient and the Hessian as well as subproblem (13), Algorithm 2 indeed enjoys optimal iteration complexity.

First, we require the following condition on approximating the gradient and Hessian.

Condition 5 (Gradient and Hessian Approximation for Algorithm 2)**.** *Given the termination criteria, ϵ_g, ϵ_H, in Algorithm 2, we require the inexact gradient and Hessian to satisfy*

$$\delta_g \le \frac{(1-\eta)}{192L_F}\left(\sqrt{K_H^2+8L_F\max\{\min\{\|\mathbf{g}_t\|, \|\mathbf{g}_{t+1}\|\}, \epsilon_g\}}-K_H\right)^2, \tag{19a}$$

$$\delta_H \le \frac{(1-\eta)}{6}\min\left\{\frac{1}{4}\left(\sqrt{K_H^2+8L_F\|\mathbf{g}_t\|}-K_H\right), \nu\epsilon_H\right\}, \tag{19b}$$

$$\delta_g \le \delta_H \le \zeta\epsilon_g, \tag{19c}$$

where $0 < \zeta < 1-\sqrt{2}/2$.

Condition 5 implies $\delta_g = \mathcal{O}(\epsilon_g^2)$ and $\delta_H = \mathcal{O}(\min\{\epsilon_g, \epsilon_H\})$, which is strictly stronger than Condition 3 in Section 2.2.1. Admittedly, although Condition 5 allows one to obtain optimal iteration complexity of Algorithm 2, it also implies more computations; for example, for the finite-sum problems of Section 2.3, this translates to larger sampling complexities. We suspect that, instead of being an inherent property of Algorithm 2, this is merely a by-product of our analysis. In this light, we conjecture that the same requirement as (15) should also be sufficient for Algorithm 2; investigating this conjecture is left for future work.

Now, we provide a sufficient condition on approximating the solution of the subproblem (13). Here, we require that subproblem (13) is solved more accurately than in Condition 4. To obtain optimal complexity, similar conditions have been considered in several previous works (Cartis et al. 2010, Xu et al. 2019). Specifically, we require that the solution is not only as good as the Cauchy and Eigen points, but also that it satisfies an extra requirement, (20), which accelerates the convergence to first-order critical points.

Condition 6. (Approximate Solution of (13) for Algorithm 2). *If $\|\mathbf{g}_t\| \geq \epsilon_g$, find $\mathbf{s}_t$ such that $m_t(\mathbf{s}_t) \leq m_t(\mathbf{s}_t^C)$ and*

$$\|\nabla m(\mathbf{s}_t)\| \leq \theta_t \|\mathbf{g}_t\|, \qquad \theta_t \leq \min\{\zeta, 1/5, \|\mathbf{s}_t\|/5\}. \tag{20}$$

Otherwise, we take the Eigen point, that is, $\mathbf{s}_t = \mathbf{s}_t^E$. Here, $\mathbf{s}_t^C$ and $\mathbf{s}_t^E$ are Cauchy and Eigen points as in Definitions 4 and 5, respectively.

It is not hard to see that, compared with Condition 4, when the gradient is large enough, Condition 6 involves a more accurate solution of (13) than a simple Cauchy point. For a given $p \ll d$, let $\mathbf{U}_t \in \mathbb{R}^{d\times p}$ be any orthonormal basis for some p-dimensional subspace $\mathcal{S}$ such that $\text{Span}\{\mathbf{s}_t^C\} \subseteq \mathcal{S} \subset \mathbb{R}^d$. Such a subspace can be easily constructed from $\mathbf{s}_t^C$ and $\mathbf{H}_t$ using standard methods, such as the Lanczos process (Ascher and Greif 2011, section 7.5). Now, a practical way to ensure Condition 6 for the case in which $\|\mathbf{g}_t\| \geq \epsilon_g$ is by approximating the unconstrained high-dimensional subproblem (13) with the following lower-dimensional problem:

$$\min_{\mathbf{v}\in\mathbb{R}^p} \langle \mathbf{U}_t\mathbf{v}, \mathbf{g}_t\rangle + \frac{1}{2}\langle \mathbf{U}_t\mathbf{v}, \mathbf{H}_t\mathbf{U}_t\mathbf{v}\rangle + \frac{\sigma_t}{3}\|\mathbf{U}_t\mathbf{v}\|^3,$$

followed by setting $\mathbf{s}_t = \mathbf{U}_t\mathbf{v}$. Obviously, when $p \ll d$, solving such a lower-dimensional problem, which involves a smaller matrix and vectors, can be significantly easier than the original high-dimensional one. One can consider a sequence of such reduced subproblems using progressively larger subspaces and stop when (20) holds. Because, ultimately, $\|\nabla m(\mathbf{s}_t)\| = 0$ for when $\mathcal{S} = \mathbb{R}^d$, we are guaranteed to also satisfy (20) for large enough $\mathcal{S} \subset \mathbb{R}^d$.

As a result of the stricter condition imposed by Condition 6, we need to refine some lemmas in Section 2.2.1. First, we need use the following result, which gives conditions for a successful iteration when $\|\mathbf{g}_t\| \geq \epsilon_g$.

Lemma 17. *Suppose Assumptions 1 and 2 and Condition 5 hold. If $\sigma_t \geq 2L_F$ and $\|\mathbf{g}_t\| > \epsilon_g$, then*

$$\delta_g\|\mathbf{s}_t\| + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{1}{2}L_F - \frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3 \leq \delta_g\|\mathbf{s}_t^C\| + \frac{1}{2}\delta_H\|\mathbf{s}_t^C\|^2. \tag{21}$$

Proof. We consider the following two cases:

i. If $\|\mathbf{s}_t\| \leq \|\mathbf{s}_t^C\|$, then, from the assumption of σ_t, it immediately follows that

$$\delta_g\|\mathbf{s}_t\| + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{1}{2}L_F - \frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3 \leq \delta_g\|\mathbf{s}_t\| + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 \leq \delta_g\|\mathbf{s}_t^C\| + \frac{1}{2}\delta_H\|\mathbf{s}_t^C\|^2.$$

ii. If $\|\mathbf{s}_t\| \geq \|\mathbf{s}_t^C\|$, first, because $L_F \leq \sigma_t/2$,

$$\delta_g\|\mathbf{s}_t\| + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{1}{2}L_F - \frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3 \leq \delta_g\|\mathbf{s}_t\| + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 - \frac{\sigma_t}{12}\|\mathbf{s}_t\|^3.$$

Now, let's define function $r(x) = \delta_g + \delta_H x/2 - \sigma_t x^2/12$. The derivative of $r(x)$ is given by $r'(x) = \delta_H/2 - \sigma_t x/6$. For any $x \geq \|\mathbf{s}_t^C\|$, according to (16a), we have

$$r'(x) \leq \frac{1}{2}\delta_H - \frac{1}{6}\sigma_t\|\mathbf{s}_t^C\| \leq \frac{1}{2}\delta_H - \frac{\sqrt{K_H^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_H}{12} \leq 0.$$

Therefore,

$$r(\|\mathbf{s}_t\|) \le r\left(\|\mathbf{s}_t^C\|\right) = \delta_g + \frac{1}{2}\delta_H\|\mathbf{s}_t^C\| - \frac{1}{12}\sigma_t\|\mathbf{s}_t^C\|^2 \le \delta_g + \left(\frac{1}{2}\delta_H - \frac{\sqrt{K_H^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_H}{24}\right)\|\mathbf{s}_t^C\|$$
$$\le \delta_g - \frac{\sqrt{K_H^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_H}{48}\|\mathbf{s}_t^C\| \le \frac{\sqrt{K_H^2 + 8L_F\|\mathbf{g}_t\|} - K_H}{192L_F} - \frac{\sqrt{K_H^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_H}{96\sigma_t} \le 0.$$

The last inequality follows from the fact that $p(x) := (\sqrt{a^2 + x} - a)^2/x$ is an increasing function over $\mathbb{R}_+$. Then, we have

$$\delta_g\|\mathbf{s}_t\| + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{1}{2}L_F - \frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3 = \|\mathbf{s}_t\|r(\|\mathbf{s}_t\|) \le 0,$$

which completes the proof. ■

With the help of the preceding lemma, we show that iteration t is successful when $\|\mathbf{g}_t\| \ge \epsilon_g$.

Lemma 18. *Suppose Assumptions 1 and 2 and Conditions 5 and 6 hold. Further, suppose at iteration t, we have $\|\mathbf{g}_t\| > \epsilon_g$ and $\sigma_t \ge 2L_F$. Then, iteration t is successful; that is, $\sigma_{t+1} = \sigma_t/\gamma$.*

Proof. First, because $\|\mathbf{g}_t\| \ge \epsilon_g$, by Lemmas 11 and 17, we have

$$F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t) - m_t(\mathbf{s}_t) \le \delta_g\|\mathbf{s}_t^C\| + \frac{1}{2}\epsilon_H\|\mathbf{s}_t^C\|^2.$$

Now from Condition 6 and (16a), we get

$$-m_t(\mathbf{s}_t) \ge -m_t\left(\mathbf{s}_t^C\right) \ge \frac{1}{12}\|\mathbf{s}_t^C\|^2\left(\sqrt{K_H^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_H\right).$$

Consider the approximation quality ρ_t,

$$1 - \rho_t = \frac{F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t) - m_t(\mathbf{s}_t)}{-m_t(\mathbf{s}_t)} \le \frac{\delta_g\|\mathbf{s}_t^C\| + \frac{1}{2}\delta_H\|\mathbf{s}_t^C\|^2}{\frac{1}{12}\|\mathbf{s}_t^C\|^2\left(\sqrt{K_H^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_H\right)}$$
$$= \frac{12\delta_g}{\|\mathbf{s}_t^C\|\left(\sqrt{K_H^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_H\right)} + \frac{6\delta_H}{\sqrt{K_H^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_H}$$
$$\le \frac{24\sigma_t\delta_g}{\left(\sqrt{K_H^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_H\right)^2} + \frac{6\delta_H}{\sqrt{K_H^2 + 4\sigma_t\|\mathbf{g}_t\|} - K_H}$$
$$\le \frac{48L_F\delta_g}{\left(\sqrt{K_H^2 + 8L_F\|\mathbf{g}_t\|} - K_H\right)^2} + \frac{6\delta_H}{\sqrt{K_H^2 + 8L_F\|\mathbf{g}_t\|} - K_H},$$

where the second inequality follows from (11) and the last inequality follows from $\sigma_t \ge 2L_F$ as well as the fact that function $r(x) := x/(\sqrt{a^2 + x} - a)^2$ is monotonically decreasing over $\mathbb{R}_+$. Now, because $\delta_H \le (1-\eta)(\sqrt{K_H^2 + 4L_F\|\mathbf{g}_t\|} - K_H)/24$, we get $6\delta_H/(\sqrt{K_H^2 + 8L_F\|\mathbf{g}_t\|} - K_H) \le (1-\eta)/4$. Similarly, because $\delta_g \le (1-\eta)(\sqrt{K_H^2 + 8L_F\|\mathbf{g}_t\|} - K_H)^2/(192L_F)$, we get $48L_F\delta_g/(\sqrt{K_H^2 + 8L_F\|\mathbf{g}_t\|} - K_H)^2 \le (1-\eta)/4$. Therefore, $1 - \rho_t \le 1 - \eta$, which means the iteration is successful. ■

Now, as in Lemma 15, we have the following:

Lemma 19. *Under Assumptions 1 and 2 and Conditions 5 and 6, we have $\sigma_t \le 2\gamma L_F$ for all t.*

We are now in position to prove the optimal complexity of Algorithm 2 under Condition 6. Recall that Lemma 15 still holds. Hence, we only need to prove a tighter bound for $|\mathcal{T}_{\text{succ}}|$. In particular, we separate $\mathcal{T}_{\text{succ}}$ into the following three subsets:

$$\mathcal{T}_{\text{succ}}^1 \triangleq \left\{t \in \mathcal{T}_{\text{succ}} \mid \|\mathbf{g}_{t+1}\| \ge \epsilon_g\right\}, \tag{22}$$
$$\mathcal{T}_{\text{succ}}^2 \triangleq \left\{t \in \mathcal{T}_{\text{succ}} \mid \|\mathbf{g}_{t+1}\| \le \epsilon_g \text{ and } \lambda_{\min}(\mathbf{H}_{t+1}) \le -\epsilon_H\right\}, \tag{23}$$
$$\mathcal{T}_{\text{succ}}^3 \triangleq \left\{t \in \mathcal{T}_{\text{succ}} \mid \|\mathbf{g}_{t+1}\| \le \epsilon_g \text{ and } \lambda_{\min}(\mathbf{H}_{t+1}) \ge -\epsilon_H\right\}. \tag{24}$$

Clearly, $\mathscr{T}_{\text{succ}} = \mathscr{T}^1_{\text{succ}} \cup \mathscr{T}^2_{\text{succ}} \cup \mathscr{T}^3_{\text{succ}}$, and trivially, $\|\mathscr{T}^3_{\text{succ}}\| = 1$.

First, let us bound $\mathscr{T}^2_{\text{succ}}$.Intuitively, we can see that we need each update to yield sufficient descent in order to bound $\mathscr{T}^1_{\text{succ}}$. Equivalently, we need each $\mathbf{s}_t$ to be bounded below to get sufficient decrease; see the following lemma.

Lemma 20. *Under Assumptions 1 and 2 and Conditions 5 and 6, we have the following upper bound:*

$$\|\mathscr{T}^2_{succ}\| \le C\epsilon_H{}^{-3}.$$

Proof. Because $F(\mathbf{x}_t)$ is monotonically decreasing, then

$$\begin{aligned}
F(\mathbf{x}_0) - F_{\min} &\ge \sum_{t=0}^{T-1} F(\mathbf{x}_t) - F(\mathbf{x}_{t+1}) = F(\mathbf{x}_0) - F(\mathbf{x}_1) + \sum_{t=0}^{T-1} F(\mathbf{x}_t) - F(\mathbf{x}_{t+1}) \\
&\ge F(\mathbf{x}_0) - F(\mathbf{x}_1) + \sum_{t \in \mathscr{T}^2_{\text{succ}}} F(\mathbf{x}_t) - F(\mathbf{x}_{t+1}) \\
&\ge F(\mathbf{x}_0) - F(\mathbf{x}_1) + \sum_{t \in \mathscr{T}^2_{\text{succ}}} \eta m_{t+1}(\mathbf{s}_{t+1}) \\
&\ge F(\mathbf{x}_0) - F(\mathbf{x}_1) + \eta \sum_{t \in \mathscr{T}^2_{\text{succ}}} \frac{\nu^3 \epsilon_H^3}{24\gamma^2 L_F^2},
\end{aligned}$$

where the last inequality follows from (17b). Hence,

$$\|\mathscr{T}^2_{\text{succ}}\| \le \frac{(F(\mathbf{x}_1) - F_{\min}) 24\gamma^2 L_F^2}{\eta \nu^3} \epsilon_H^{-3} = \mathcal{O}(\epsilon_H^{-3}). \quad \blacksquare$$

Intuitively, we can see that we need each update to yield suffcient descent in order to bound $\mathscr{T}^1_{\text{succ}}$. Equivalently, we need each s_t to be bounded below to get suffcient decrease; see the following lemma.

Lemma 21. *Suppose Assumptions 1 and 2 and Conditions 5 and 6 hold. If iteration t is successful and $\|\mathbf{g}_t\| \ge \epsilon_g$, then*

$$\|\mathbf{s}_t\| \ge \kappa_g \left[\left(1 - \zeta - \frac{\zeta}{1 - 2\zeta}\right) \|\mathbf{g}_{t+1}\| - \frac{5}{2}\delta_g \right],$$

where

$$\kappa_g := \min\left\{ \left(\frac{L_F}{2} + 2\gamma L_F + \epsilon_0 + \zeta K_F\right)^{-1}, \left(\frac{L_F}{2} + 2\gamma L_F + \frac{\zeta}{1 - 2\zeta} K_F + \zeta K_F\right)^{-1} \right\}.$$

Proof. Using Condition 6, we get

$$\|\mathbf{g}_{t+1}\| \le \|\mathbf{g}_{t+1} - \nabla m_t(\mathbf{s}_t)\| + \|\nabla m_t(\mathbf{s}_t)\| \le \|\mathbf{g}_{t+1} - \nabla m_t(\mathbf{s}_t)\| + \theta_t \|\mathbf{g}_t\|. \tag{25}$$

Noting that $\nabla m_t(\mathbf{s}_t) = \mathbf{g}_t + \mathbf{H}_t \mathbf{s}_t + \sigma_t \|\mathbf{s}_t\| \mathbf{s}_t$, by Assumptions 1 *and* 2, we have

$$\begin{aligned}
\|\mathbf{g}_{t+1} - \nabla m_t(\mathbf{s}_t)\| &\le \|\mathbf{g}_{t+1} - \mathbf{g}_t - \mathbf{H}_t \mathbf{s}_t\| + \sigma_t \|\mathbf{s}_t\|^2 \\
&\le \left\| \int_0^1 \left(\nabla^2 F(\mathbf{x}_t + \tau \mathbf{s}_t) - \nabla^2 F(\mathbf{x}_t)\right) \mathbf{s}_t d\tau + \left(\nabla^2 F(\mathbf{x}_t) - \mathbf{H}_t\right) \mathbf{s}_t \right\| \\
&\quad + \|\mathbf{g}_t - \nabla F(\mathbf{x}_t)\| + \|\mathbf{g}_{t+1} - \nabla F(\mathbf{x}_t + \tau \mathbf{s}_t)\| + \sigma_t \|\mathbf{s}_t\|^2 \\
&\le \left(\frac{L_F}{2} + 2\gamma L_F\right) \|\mathbf{s}_t\|^2 + \delta_H \|\mathbf{s}_t\| + 2\delta_g.
\end{aligned} \tag{26}$$

Also, according to Assumption 2, we get

$$\begin{aligned}\|\mathbf{g}_t\| \le \|\mathbf{g}_t - \nabla F(\mathbf{x}_t)\| + \|\nabla F(\mathbf{x}_t))\| &\le \delta_g + K_H\|\mathbf{s}_t\| + \|\nabla F(\mathbf{x}_t + \mathbf{s}_t)\| \\ &\le 2\delta_g + K_H\|\mathbf{s}_t\| + \|\mathbf{g}_{t+1}\|.\end{aligned} \tag{27}$$

By combining (25)–(27) and using definition of θ_t in (20), we get

$$\begin{aligned}\|\mathbf{g}_{t+1}\| &\le \left(\frac{L_F}{2} + 2\gamma L_F\right)\|\mathbf{s}_t\|^2 + (\delta_H + \theta_t K_F)\|\mathbf{s}_t\| + 2(1+\theta_t)\delta_g + \theta_t\|\mathbf{g}_{t+1}\| \\ &\le \left(\frac{L_F}{2} + 2\gamma L_F\right)\|\mathbf{s}_t\|^2 + (\delta_H + \theta_t K_F)\|\mathbf{s}_t\| + \frac{5}{2}\delta_g + \zeta\|\mathbf{g}_{t+1}\|,\end{aligned}$$

which implies

$$(1-\zeta)\|\mathbf{g}_{t+1}\| - \frac{5}{2}\delta_g \le \left(\frac{L_F}{2} + 2\gamma L_F\right)\|\mathbf{s}_t\|^2 + (\delta_H + \theta_t K_F)\|\mathbf{s}_t\|. \tag{28}$$

Now, consider two cases:

i. If $\|\mathbf{s}_t\| \ge 1$, then

$$(\delta_H + \theta_t K_F)\|\mathbf{s}_t\| \le (\epsilon_H + \zeta K_F)\|\mathbf{s}_t\|^2.$$

It follows that

$$(1-\zeta)\|\mathbf{g}_{t+1}\| - 5/2\delta_g \le \left(\frac{L_F}{2} + 2\gamma L_F + \epsilon_H + \zeta K_F\right)\|\mathbf{s}_t\|^2,$$

that is,

$$\|\mathbf{s}_t^2\| \ge \frac{(1-\zeta)\|\mathbf{g}_{t+1}\| - \frac{5}{2}\delta_g}{L_F/2 + 2\gamma L_F + \epsilon_H + \zeta K_F}.$$

ii. If $\|\mathbf{s}_t\| \le 1$, then

$$\begin{aligned}\delta_H &\le \zeta\epsilon_g \le \zeta\|\mathbf{g}_t\| \\ &\le \zeta\big(\|\mathbf{g}_{t+1}\| + \|\nabla F(\mathbf{x}_t + \mathbf{s}_t) - \mathbf{g}_{t+1}\| + \|\nabla F(\mathbf{x}_t) - \nabla F(\mathbf{x}_t + \mathbf{s}_t)\| + \|\mathbf{g}_t - \nabla F(\mathbf{x}_t)\|\big) \\ &\le \zeta\big(2\delta_g + K_F\|\mathbf{s}_t\| + \|\mathbf{g}_{t+1}\|\big) \\ &\le \zeta\big(2\delta_H + K_F\|\mathbf{s}_t\| + \|\mathbf{g}_{t+1}\|\big),\end{aligned}$$

where the third inequality is from the triangular inequality and the last inequality follows from $\delta_g \le \delta_H$ in (19c) in Condition 6. Therefore, we have

$$\delta_H\|\mathbf{s}_t\| \le \frac{\zeta}{1-2\zeta}\big(K_F\|\mathbf{s}_t\| + \|\mathbf{g}_{t+1}\|\big)\|\mathbf{s}_t\| \le \frac{\zeta}{1-2\zeta}\big(K_F\|\mathbf{s}_t\|^2 + \|\mathbf{g}_{t+1}\|\big).$$

Then, using $\theta_t \le \zeta$ in (20),

$$(\delta_H + \theta_t K_F)\|\mathbf{s}_t\| \le \left(\frac{\zeta}{1-2\zeta} + \zeta\right)K_F\|\mathbf{s}_t\|^2 + \frac{\zeta}{1-2\zeta}\|\mathbf{g}_{t+1}\|.$$

Substituting this into (28), we have

$$\left(1 - \zeta - \frac{\zeta}{1-2\zeta}\right)\|\mathbf{g}_{t+1}\| - \frac{5}{2}\delta_g \le \left(\frac{L_F}{2} + 2\gamma L_F + \frac{\zeta}{1-2\zeta}K_F + \zeta K_F\right)\|\mathbf{s}_t\|^2,$$

that is,

$$\|\mathbf{s}_t\|^2 \ge \left(\left(1 - \zeta - \frac{\zeta}{1-2\zeta}\right)\|\mathbf{g}_{t+1}\| - \frac{5}{2}\delta_g\right)\left(\frac{L_F}{2} + 2\gamma L_F + \frac{\zeta}{1-2\zeta}K_F + \zeta K_F\right)^{-1}.$$

The two cases complete the proof. ■

Now, based on Lemma 21, it is easy to bound $\|\mathcal{T}^1_{\text{succ}}\|$.

Lemma 22. *Given the same setting as Lemma 21, the success iteration $\mathcal{T}^1_{succ}$ is bounded by*

$$\|\mathcal{T}^1_{\text{succ}}\| \le C\max\{\epsilon_g{}^{-1.5}, \epsilon_H{}^{-3}\}.$$

Proof. First, according to (19a) in Condition 5, we have

$$\delta_g \le \frac{1-\eta}{192L_F}\left(\sqrt{K_H^2 + 8L_F\max\{\min\{\|\mathbf{g}_t\|, \|\mathbf{g}_{t+1}\|\}, \epsilon_g\}} - K_H\right)^2 \le \frac{(1-\eta)8L_F\max\{\min\{\|\mathbf{g}_t\|, \|\mathbf{g}_{t+1}\|\}, \epsilon_g\}}{192L_F}$$
$$\le \frac{\max\{\min\{\|\mathbf{g}_t\|, \|\mathbf{g}_{t+1}\|\}, \epsilon_g\}}{24}.$$

If $\|\mathbf{g}_{t+1}\| \ge \epsilon_g$ and $\|\mathbf{g}_t\| \ge \epsilon_g$, according to Lemma 21 and substituting $\zeta = 1/4$, we have

$$\|\mathbf{s}_t\|^2 \ge \kappa_g\left[\left(1 - 1/4 - \frac{1/4}{1-2/4}\right)\epsilon_g - 5/2\frac{1}{24}\epsilon_g\right] = \frac{1}{8}\kappa_g\epsilon_g.$$

Now, consider any $t \in \mathcal{T}^1_{\text{succ}}$. Because $\|\mathbf{g}_t\| \ge \epsilon_g$, then we have

$$-m_t(\mathbf{s}_t) \ge \frac{\sigma_t}{6}\|\mathbf{s}_t\|^3 \ge \frac{\sigma_{\min}}{6}\left(\frac{\kappa_g\epsilon_g}{8}\right)^{3/2} \ge c_g\epsilon_g{}^{3/2},$$

where $c_g \triangleq \kappa_g^{3/2}\sigma_{\min}/200$. Otherwise, we must have $\lambda_{\min}(\mathbf{H}_t) \le -\epsilon_H$, and by (17b), we have

$$-m_t(\mathbf{s}_t) \ge \frac{\nu^3\epsilon_H^3}{24\gamma^2L_F^2} = c_H\epsilon_H^3,$$

where $c_H \triangleq \frac{\nu^3}{24\gamma^2L_F^2}$. Therefore,

$$-m_t(\mathbf{s}_t) \ge \min\left\{c_g\epsilon_g^{3/2}, c_H\epsilon_H^3\right\}.$$

Because $F(\mathbf{x}_t)$ is monotonically decreasing and $F(\mathbf{x})$ is lower bounded by $F_{\min}$, it follows that

$$F(\mathbf{x}_0) - F_{\min} \ge \sum_{t=0}^{T-1} F(\mathbf{x}_t) - F(\mathbf{x}_{t+1}) \ge \sum_{t\in\mathcal{T}^1_{\text{succ}}} F(\mathbf{x}_t) - F(\mathbf{x}_{t+1}) \ge \sum_{t\in\mathcal{T}^1_{\text{succ}}} -\eta m_t(\mathbf{s}_t)$$
$$\ge \sum_{t\in\mathcal{T}^1_{\text{succ}}} \min\left\{c_g\epsilon_g^{3/2}, c_H\epsilon_H^3\right\} = \|\mathcal{T}^1_{\text{succ}}\|\min\left\{c_g\epsilon_g^{3/2}, c_H\epsilon_H^3\right\}.$$

Therefore,

$$\|\mathcal{T}^1_{\text{succ}}\| \le \max\left\{\frac{F(\mathbf{x}_0) - F_{\min}}{c_g}\epsilon_g^{-3/2}, \frac{F(\mathbf{x}_0) - F_{\min}}{c_H}\epsilon_H^{-3}\right\},$$

which completes the proof. ■

Because $\mathcal{T}_{\text{succ}} = \mathcal{T}^1_{\text{succ}} \cup \mathcal{T}^2_{\text{succ}} \cup \mathcal{T}^3_{\text{succ}}$, we can get a bound of the total number of successful iterations.

Lemma 23. *Given Assumptions 1 and 2 as well as Conditions 5 and 6, the number of successful iterations $\|\mathcal{T}_{succ}\|$ is bounded by*

$$\|\mathcal{T}_{succ}\| \le C\max\{\epsilon_g{}^{-1.5}, \epsilon_H{}^{-3}\}.$$

Proof. It immediately follows from Lemmas 20 and 22. ■

The optimal iteration complexity of Algorithm 2 is stated in Theorem 4.

Theorem 4 (Optimal Complexity of Algorithm 2). *Let Assumption 1 hold and consider any $0 < \epsilon_g, \epsilon_H < 1$. Further, suppose that $\mathbf{g}_t$ and $\mathbf{H}_t$ satisfy Assumption 2 with δ_g and δ_H under Condition 5. If the approximate solution to subproblem (13) satisfies Condition 6, then Algorithm 2 terminates after at most*

$$T \in \mathcal{O}(\max\{\epsilon_g{}^{-1.5}, \epsilon_H{}^{-3}\}),$$

iterations.

Remark 5. If we assume L_F is known (set $\sigma_t \equiv L_F$) and $\mathbf{s}_t$ is close enough to the best solution $\mathbf{s}_t^*$ of $m_t(\mathbf{s})$, by using Taylor expansion, it is not hard to show that

$$F(\mathbf{x}_t + \mathbf{s}_t) - F(\mathbf{x}_t) \ge -c_1 m_t(\mathbf{s}_t) \ge -c_2 m_t\big(\mathbf{s}_t^*\big).$$

Given that $\|\mathbf{g}_t\|$ or $-\lambda_{\min}(\mathbf{H}_t)$ is large, $-m(\mathbf{s}_t^*)$ would then be large. Therefore, there could be enough descent along $\mathbf{s}_t$. Roughly speaking, we could drop Lemmas 9–15 and get the same iteration complexity results, that is, $T \in \mathcal{O}(\max\{\epsilon_g{}^{-1.5}, \epsilon_H{}^{-3}\})$. For example, we do not need Lemma 9 to show the Cauchy point is one of the directions for $-m_t(\mathbf{s}_t)$. Also, in this case, either Lemma 17 or 18 becomes redundant.

2.3. Finite-Sum Problems

As a special class of (1), we now consider a nonconvex finite-sum minimization of (2), where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is smooth and nonconvex. In big-data regimes in which $n \gg 1$, one can consider subsampling schemes to speed up various aspects of many Newton-type methods; for example, see Roosta and Mahoney (2019), Xu et al. (2016), and Bollapragada et al. (2018) for such techniques in the context of convex optimization. More specifically, we consider the subsampled gradient and Hessian as

$$\mathbf{g} \triangleq \frac{1}{\|\mathcal{S}_g\|} \sum_{i \in \mathcal{S}_g} \nabla f_i(\mathbf{x}), \quad \text{and} \quad \mathbf{H} \triangleq \frac{1}{\|\mathcal{S}_H\|} \sum_{i \in \mathcal{S}_H} \nabla^2 f_i(\mathbf{x}), \tag{29}$$

where $\mathcal{S}_g, \mathcal{S}_H \subset \{1, \cdots, n\}$ are the subsample batches for the estimates of the gradient and Hessian, respectively. In this setting, a relevant question is that of how large sample sizes $\mathcal{S}_g$ and $\mathcal{S}_H$ should be to guarantee, at least with high probability, that $\mathbf{g}$ and $\mathbf{H}$ in (29) satisfy Assumption 2. As long as $\|\mathcal{S}_g\| \ll n$ and $\|\mathcal{S}_H\| \ll n$, such subsampling strategies can result in significant reduction in overall computational costs.

If sampling is done uniformly at random, we have the following sampling complexity bounds, whose proofs can be found in Roosta and Mahoney (2019) and Xu et al. (2019). For more sophisticated sampling/sketching schemes, see Pilanci and Wainwright (2015) and Xu et al. (2016, 2019).

Lemma 24. (Sampling Complexity; Roosta and Mahoney 2019, Xu et al. 2019). *For any $0 < \delta_g, \delta_H, \delta < 1$, let $\mathbf{g}$ and $\mathbf{H}$ be as in (29) with*

$$\|\mathcal{S}_g\| \ge \frac{16K_g^2}{\delta_g^2} \log\frac{1}{\delta} \quad \textit{and} \quad \|\mathcal{S}_H\| \ge \frac{16K_H^2}{\delta_H^2} \log\frac{2d}{\delta},$$

where $0 < K_g, K_H < \infty$ are such that $\|\nabla f_i(\mathbf{x})\| \le K_g$ and $\|\nabla^2 f_i(\mathbf{x})\| \le K_H$. Then, with probability at least $1 - \delta$, Assumption 2 holds with the corresponding δ_g and δ_H. Combining Lemma 24 with the sufficient conditions presented earlier, that is, Conditions 1 and 2 for Algorithm 1 and Conditions 3 and 4 or Conditions 5 and 6 for Algorithm 2, we can immediately obtain similar but probabilistic iteration complexities as in Sections 2.1 and 2.2. For completeness, we bring such a result for Algorithm 1 and omit those related to Algorithm 2.

Because Conditions 1, 3, and 5 are only guaranteed probabilistically, in order to guarantee success, a small failure probability across all iterations is required. In particular, in order to get an accumulative success probability of $1 - \delta$ for the entire T iterations, the per-iteration failure probability is set as $(1 - \sqrt{[}T](1 - \delta)) \in \mathcal{O}(\delta/T)$. Fortunately, this failure probability appears only in the "log factor" in Lemma 24, and so it is not the dominating cost. For example, for $T \in \mathcal{O}(\max\{\epsilon_g^{-2}\epsilon_H^{-1}, \epsilon_H^{-3}\})$, as in Theorem 1, we can set the per-iteration failure probability to $\delta \min\{\epsilon_g^2 \epsilon_H, \epsilon_H^3\}$.

Corollary 1. (Optimal Complexity of Algorithm 1 for Finite-Sum Problem (2)). *Consider any $0 < \epsilon_g, \epsilon_H, \delta < 1$. Let δ_g and δ_H be as in Condition 1 and set $\delta_0 = \delta \min\{\epsilon_g^2 \epsilon_H, \epsilon_H^3\}$. Furthermore, for such δ_g, δ_H, and δ_0, let the sample size $\|\mathcal{S}_g\|$ and $\|\mathcal{S}_H\|$ be as in Lemma 24 and form the subsampled gradient and Hessian as in* **H** *as in (29). For problem (2), under Assumptions 1 and 2 and Conditions 1 and 2, Algorithm 1 terminates in at most $T \in \mathcal{O}(\max\{\epsilon_g^{-2}\epsilon_H^{-1}, \epsilon_H^{-3}\})$ iterations, upon which, with probability $1 - \delta$, we have that $\|\nabla F(\mathbf{x})\| \le \epsilon_g + \delta_g$, and $\lambda_{\min}(\nabla^2 F(\mathbf{x})) \ge -(\delta_H + \epsilon_H)$.*

3. Experiments

In this section, we provide empirical results evaluating the performance of Algorithms 1 and 2. We aim to demonstrate that an approximate gradient, Hessian, and subproblem solves indeed help improve the computational efficiency. For our experiments, we consider the following methods:

- Full ARC: Standard ARC algorithms with exact gradient and Hessian.

- SubH TR/ARC (Xu et al. 2019): TR and ARC with exact gradient and subsampled Hessian.
- SCR (GD) (Tripuraneni et al. 2018): CR with subsampled gradient and Hessian. The subproblems are solved by gradient descent (GD) (Carmon and Duchi 2016).
- SCR (Lanczos): CR that is similar to SCR (GD) (Tripuraneni et al. 2018), but the subproblems are solved by the generalized Lanczos method (Cartis et al. 2011a).
- SGD: Stochastic gradient descent with momentum (Sutskever et al. 2013). The momentum parameter is set to the typical value of 0.9. The gradient size is set to be 1,000.
- Adagrad: An adaptive first-order method developed in Duchi et al. (2011). The gradient size is set to be 1,000.
- ADAM: A modification of Adagrad that has become the method of choice within the machine learning community (Kingma and Ba 2014). The two momentum terms in ADAM are set to be 0.9 and 0.999, which are typically chosen in practice. The gradient size is set to be 1,000.
- Inexact TR/ARC (this work): TR and ARC with subsampled gradient and Hessian as described in Algorithms 1 and 2. The subproblems of Algorithms 1 and 2 are solved, respectively, by CG-Steihaug (Steihaug 1983) and by the generalized Lanczos method (Cartis et al. 2011a). For both algorithms, the gradient sample size is adaptively chosen as follows: if $\|g_t\| \geq 1.2\|g_{t-1}\|$ or $\|g_t\| \leq \|g_{t-1}\|/1.2$, we, respectively, decrease or increase the sample size for gradient estimation by a factor of 1.2. Otherwise, the sample size stays the same as the previous iteration.

For our experiments, except SCR (GD), we use the CG-Steihaug (Nocedal and Wright 2006) and generalized Lanczos methods (Cartis et al. 2011a) to solve the subproblems of TR and ARC, respectively. Also following Xu et al. (2020), we set the maximum iterations for the subproblem solvers to 250. Further specific hyperparameters as well as samples sizes used for second-order algorithms in our experiments are gathered in Table 2.

Similar to Xu et al. (2020), the performance of all the algorithms in our experiments is measured by tallying total *number of propagations*, that is, the number of oracle calls of function, gradient, and Hessian-vector products. More specifically, for each i in (2), after computing $f_i(\mathbf{x})$, computing $\nabla f_i(\mathbf{x})$ is equivalent to one additional function evaluation. In our implementations, we merely require Hessian-vector products $\nabla^2 f_i(\mathbf{x})\mathbf{v}$ instead of forming the explicit Hessian, which amounts to two additional function evaluations as compared with gradient evaluation. We would like note that we opted to choose propagations as the complexity because "wall clock" time can be highly affected by particular implementation details as well as system specifications. In contrast, counting the number of propagations (or oracle calls) is implementation and system independent and is, hence, more appropriate and fair. For experiments of Section 3.1, we use a GTX Titan X GPU with 12 Gb RAM memory. The code is based on Python with framework PyTorch 1.2.0. In Section 3.2, the experiments are performed on a Macbook Pro, 2017c (2.9 GHz Intel Core i7-7820HQ, 16 Gb RAM) with Matlab. Our code is publicly available at https://github.com/yaozhewei/Inexact_Newton_Method.

3.1. Multilayer Perceptron

We first evaluate the performance of Algorithm 1 in terms of running time as measured by the training loss versus total number of propagations. We do this using a simple multilayer perceptron (MLP) model on the MNIST data set, which is available from LIBSVM library (Chang and Lin 2011).

Here, we consider an MLP involving one hidden layer and one output layer to determine the assigned class of the input image. All intermediate neurons involve the SoftPlus activation function (Glorot et al. 2011), which amounts to a smooth optimization problem. We consider three instances of such an MLP with hidden layer sizes of 16, 128, and 1,024. Table 3 gathers the dimensions of the resulting optimization problems.

Table 2. The Hyperparameters and the Samples Sizes Used for Newton-Type Methods Used in the Experiments

Method	Full ARC	SubH TR	SubH ARC	SCR	Algorithm 1	Algorithm 2
Hyperparameter	$\sigma_0 = 10$	$\Delta_0 = 10$	$\sigma_0 = 10$	$\sigma = 10$ (Fig. 2)	$\Delta_0 = 10$	$\sigma_0 = 10$
$\|\mathcal{S}_g\|$ (Section 3.1)	n	n	n	N/A	5,000	5,000
$\|\mathcal{S}_H\|$ (Section 3.1)	n	1,000	1,000	N/A	1,000	1,000
$\|\mathcal{S}_g\|$ (Section 3.2)	n	n	n	$0.1n$	$0.1n$	$0.1n$
$\|\mathcal{S}_H\|$ (Section 3.2)	n	$0.01n$	$0.01n$	$0.01n$	$0.01n$	$0.01n$

Notes. n is as in Tables 3 and 4. Recall that $\|\mathcal{S}_g\|$ is adjusted adaptively for Algorithms 1 and 2, and hence, the values given here refer to the gradient sample size at initialization. For Hessian estimation, however, we use a fixed sample size for both Algorithms 1 and 2.

Table 3. The Dimension of the Parameter Space for Various Hidden Layer Sizes in the MLP Experiment

Hidden layer size	n	d
16	60,000	12,704
128	60,000	101,632
1,024	60,000	813,056

Similar to the observations in Xu et al. (2020), despite the best of our efforts, we were unable to obtain the expected performance of any variant of ARC and CR on this model problem using a variety of implementations. As a result, we did not include them in this experiment.

For all first-order methods, several fixed step sizes in the range $\alpha = [0.0001, 0.001, 0.01, 0.1]$ are tested. As is clearly observed here and also is reported in similar literature (Berahas et al. 2017, Kylasa et al. 2019, Xu et al. 2020), the performance of first-order methods strongly depends on the particular choice of their main hyperparameter, that is, the step size. For example, in Figure 1, (g)–(i), a finely tuned ADAM can have superior performance. However, if the step size is not chosen appropriately, the performance of ADAM could be unpleasantly erratic. As can also be seen, even the best performing step size for ADAM ceases to be appropriate at later stages of the algorithm. As a result, to obtain a solution with higher accuracy, one needs to pick a new step size at later stages of the algorithm as, otherwise, ADAM ultimately diverges or exhibits violent zigzagging behavior.

3.2. Nonlinear Least Squares

Because we did not manage to obtain a reasonable performance using any variants of ARC and CR, we opted to exclude them from the previous experiments in Section 3.1. Nonetheless, on a simpler nonlinear least squares problem, we were able to compare and contrast various properties of these methods, which we include in this section.

3.2.1. Computational Efficiency (Figure 2). We now consider the running time of Algorithm 2 in the context of simple yet illustrative, nonlinear least squares arising from the task of binary classification with squared loss.[1] Specifically, given training data $\{\mathbf{a}_i, b_i\}_{i=1}^n$, where $\mathbf{a}_i \in \mathbb{R}^d, b_i \in \{0, 1\}$, consider the following empirical risk minimization problem:

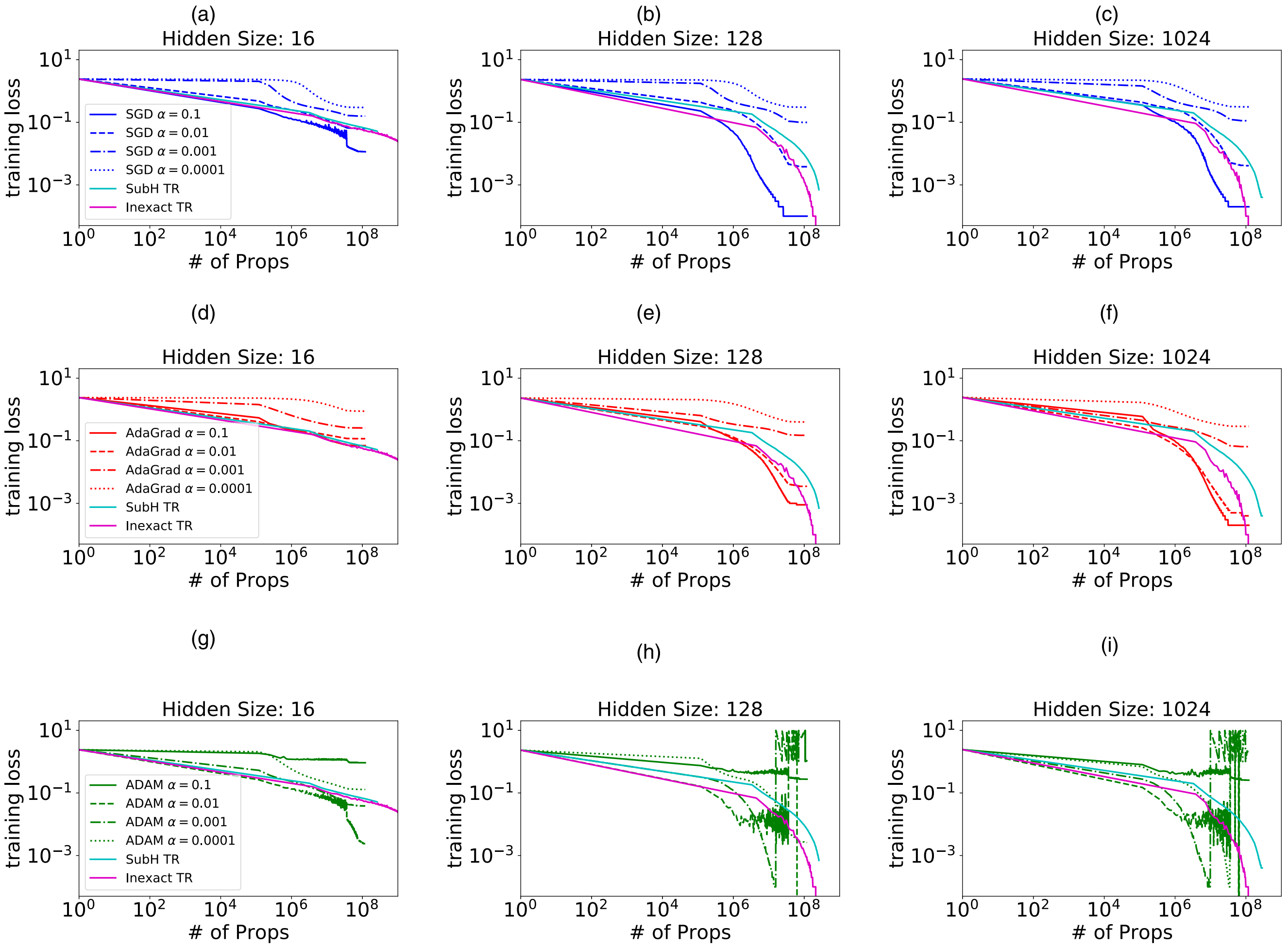
$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\big(b_i - \phi(\langle \mathbf{a}_i, \mathbf{x}\rangle)\big)^2,$$

where $\phi(z)$ is the sigmoid function; that is, $\phi(z) = 1/(1 + e^{-z})$. Data sets are taken from LIBSVM library (Chang and Lin 2011); see Table 4. We use the same setup in Xu et al. (2020).

Figure 2 depicts the results. For all variants of SCR, we hand-tuned the algorithm by performing an exhaustive grid search over the involved hyperparameters, and we show the best results. For all variants of ARC, we chose the same initial parameters, σ_0. We can observe that all methods achieve similar training errors, and Algorithm 2 does so with a much fewer number of propagation calls as compared with other methods. Furthermore, all variants of ARC perform similarly or better than all variants of CR. This is empirical evidence that the “optimal” worst-case analysis of CR, although theoretically interesting, might not translate to many practical applications of interest.

3.2.2. Robustness to Hyperparameters (Figure 3). Next, we highlight the practical challenges arising with algorithms that heavily rely on the knowledge of hard-to-estimate parameters. In particular, we aim here to demonstrate that an algorithm whose performance is greatly affected by specific settings of parameters that cannot be easily estimated lacks the versatility needed in many practical applications. To do so, we perform one such demonstration by focusing on the sensitivity/robustness of Algorithm 2 and SCR to the cubic regularization parameter σ. The results are gathered in Figure 3. One can see that the performance of SCR is highly dependent on the choice of its main hyperparameter, that is, σ. Indeed, if σ is not chosen appropriately, SCR either converges very slowly or does not converge at all. Determining an appropriate value of σ requires an expensive (in human or CPU time) hyperparameter search. This is in sharp contrast with Algorithm 2, which shows great robustness to the choice of σ_0 and works more-or-less “out of the box.”

Figure 1. (Color online) Results of Variants of TR (SubH TR and Inexact TR) and First Order Methods (SGD, AdaGrad, and ADAM) on MLP with Different Hidden Sizes (16, 128, and 1024)

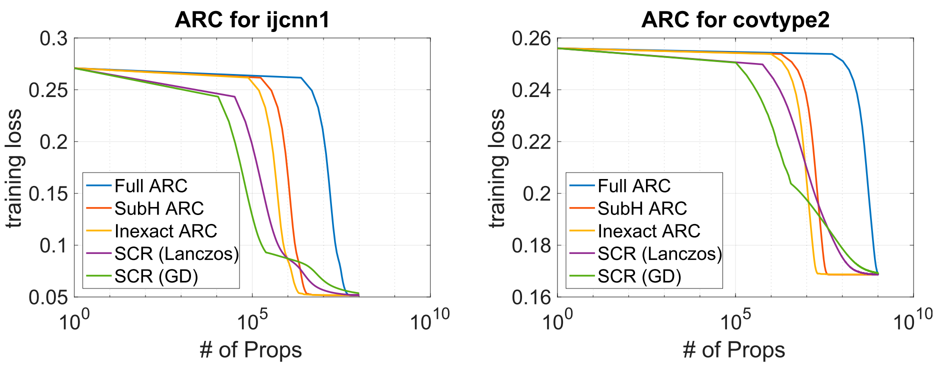


Note. Both x-axis and y-axis are drawn using the logarithmic scaling.

3.3. Summary of Numerical Experiments

From these numerical examples, that is, multilayer perceptron in Section 3.1 and nonlinear least squares in Section 3.2, we can make the following general observations regarding the overall performance of Algorithms 1 and 2.

Figure 2. (Color online) Performance of Variants of ARC and CR Methods on ijcnn1 and covertype for Binary Linear Classification

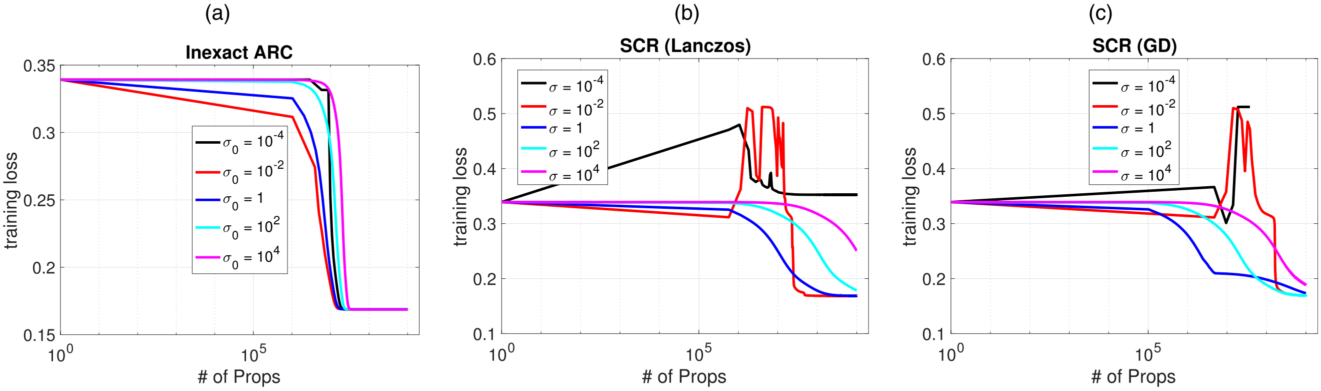


Note. The x-axis is drawn on the logarithmic scale.

Table 4. Data Sets Used for Experiments with Nonlinear Least Squares.

Data	n	d
covertype	464,810	54
ijcnn1	49,990	22

Figure 3. (Color online) Robustness of Algorithm 2 and Sensitivity of SCR with Respect to the Cubic Regularization Parameter on the covertype Data Set



Notes. For Algorithm 2, this parameter, initially set to _0, adaptively changes across iterations although for SCR, it is kept fixed at a certain σ for all iterations. (a) Robustness of Algorithm 2 to the choice of _0, where _0 varies over several orders of magnitude. (b, c) Sensitivity of SCR with two different subproblem solvers (Lanczos and GD) and several choices of the fixed cubic regularization σ. For SCR (GD), the step size of GD for solving the subproblem is hand-tuned to obtain the best performance (which can be extremely expensive).

i. Within the context of both inexact TR and ARC, we can clearly see the added efficiency obtained from subsampling both the gradient and the Hessian. This is illustrated by competitive performance compared with several first-order methods as well as superior performance relative to more expensive variants, that is, exact algorithms and those in which only the Hessian is approximated as in Xu et al. (2019).

ii. In terms of tuning the respective underlying hyperparameters, our inexact ARC variant is significantly more robust compared with SCR (Tripuraneni et al. 2018). Similarly, in contrast to first-order algorithms whose performance is greatly affected by the choice of the main hyperparameter, that is, step size, the performance of the proposed Newton-type methods exhibits significant resilience to particular choices of hyperparameters.

4. Conclusions and Further Thoughts

In this paper, we consider inexact variants of trust region and adaptive cubic regularization in which, to increase efficiency, the gradient and Hessian as well as the solution to the underlying subproblems are all suitably approximated. We show that, under certain conditions on these approximations, to coverage to second-order criticality, the inexact variants achieve the same optimal iteration complexity as the exact counterparts. The advantages and perhaps shortcomings of our algorithms are also numerically demonstrated.

We note that, unlike Conditions 2, 4, and 6, ensuring Conditions 1, 3, and 5 is not generally as straightforward and remains the main practical challenge in our work and, to our knowledge, all of related literature. Although deterministic approaches, such as finite-difference schemes, can theoretically guarantee these conditions, obtaining an appropriate discretization scheme relies on the knowledge of problem-dependent constants that are typically hard to estimate. Similarly, for the case of finite-sum minimization in Section 2.3, which is an important driving application for our results here, randomized subsampling techniques can give sufficient sample sizes to guarantee such conditions. However, this also requires estimates of the constants L_F, K_H, and K_g. Fortunately, for several problems in machine learning, obtaining such estimates is in fact straightforward, for example, linear predictor models in Xu et al. (2019, table 1) and Roosta and Mahoney (2019, table 2) as well as deep learning in Fazlyab et al. (2019). Furthermore, in our experience as well as that of many others, the performance of such subsampled algorithms is most resilient to under-sampling. This is in sharp contrast, however, to the quality of the subproblem solutions, which significantly affect the overall performance of the algorithms.

As a by-product of our analysis, the bound on gradient approximations for obtaining the optimal iteration complexity of inexact ARC remains very pessimistic, and tightening such a bound is left for future work. An important missing piece from our work here is incorporating function approximations as a way to further reduce the computational costs, which we are currently pursuing. Finally, our results here only consider iteration complexities of the proposed algorithms. A much finer grained analysis is required to obtain overall running time, which is an important avenue for future work.

Acknowledgments

Part of this work was done while FR and MM were visiting the Simons Institute for the Theory of Computing.

Appendix. Additional Proofs

In this section, we give the proofs of some lemmas mentioned in the main text.

A.1. Proof of Lemma 3

When $\|\mathbf{g}_t\| \geq \epsilon_g$, using Taylor expansion of $F(\mathbf{x}_t)$ at point $\mathbf{x}_t$,

$$\begin{aligned}
F(\mathbf{x}_t+\mathbf{s}_t)-F(\mathbf{x}_t)-m_t(\mathbf{s}_t) &= \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)-\mathbf{g}_t\rangle + \frac{1}{2}\langle \mathbf{s}_t, (\nabla^2 F(\mathbf{x}_t+\tau\mathbf{s}_t)-\mathbf{H}_t)\mathbf{s}_t\rangle \\
&\leq \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)-\mathbf{g}_t\rangle + |\frac{1}{2}\langle \mathbf{s}_t, (\mathbf{H}_t-\nabla^2 F(\mathbf{x}_t+\tau\mathbf{s}_t))\mathbf{s}_t\rangle| \\
&\leq \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)-\mathbf{g}_t\rangle + |\frac{1}{2}\langle \mathbf{s}_t, (\mathbf{H}_t-\nabla^2 F(\mathbf{x}_t))\mathbf{s}_t\rangle| \\
&\qquad + |\frac{1}{2}\langle \mathbf{s}_t, (\nabla^2 F(\mathbf{x}_t+\tau\mathbf{s}_t)-\nabla^2 F(\mathbf{x}_t))\mathbf{s}_t\rangle| \\
&\leq \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)-\mathbf{g}_t\rangle + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \frac{1}{2}L_F\|\mathbf{s}_t\|^3 \\
&\leq \delta_g\Delta_t + \frac{1}{2}\delta_H{\Delta_t}^2 + \frac{1}{2}L_F{\Delta_t}^3,
\end{aligned}$$

where $\tau \in [0,1]$. Similarly, when $\|\mathbf{g}_t\| < \epsilon_g$,

$$F(\mathbf{x}_t+\mathbf{s}_t)-F(\mathbf{x}_t)-m_t(\mathbf{s}_t) \leq \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)\rangle + \frac{1}{2}\delta_H{\Delta_t}^2 + \frac{1}{2}L_F{\Delta_t}^3.$$

A.2. Proof of Lemma 11

When $\|\mathbf{g}_t\| \geq \epsilon_g$, using Taylor expansion of $F(\mathbf{x})$ at point $\mathbf{x}_t$,

$$\begin{aligned}
F(\mathbf{x}_t+\mathbf{s}_t)-F(\mathbf{x}_t)-m_t(\mathbf{s}_t) &= \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)-\mathbf{g}_t\rangle + \frac{1}{2}\langle \mathbf{s}_t, (\nabla^2 F(\mathbf{x}_t+\tau\mathbf{s}_t)-\mathbf{H}_t)\mathbf{s}_t\rangle - \frac{\sigma_t}{3}\|\mathbf{s}_t\|^3 \\
&\leq \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)-\mathbf{g}_t\rangle + |\frac{1}{2}\langle \mathbf{s}_t, \Big(\mathbf{H}_t-\nabla^2 F(\mathbf{x}_t+\tau\mathbf{s}_t)\mathbf{s}_t\rangle| - \frac{\sigma_t}{3}\|\mathbf{s}_t\|^3 \\
&\leq \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)-\mathbf{g}_t\rangle + |\frac{1}{2}\langle \mathbf{s}_t, \big(\mathbf{H}_t-\nabla^2 F(\mathbf{x}_t)\big)\mathbf{s}_t\rangle| \\
&+ |\frac{1}{2}\langle \mathbf{s}_t, \big(\nabla^2 F(\mathbf{x}_t+\tau\mathbf{s}_t)-\nabla^2 F(\mathbf{x}_t)\big)\mathbf{s}_t\rangle| - \frac{\sigma_t}{3}\|\mathbf{s}_t\|^3 \\
&\leq \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)-\mathbf{g}_t\rangle + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{L_F}{2}-\frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3, \\
&\leq \delta_g\|\mathbf{s}_t\| + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{L_F}{2}-\frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3,
\end{aligned}$$

where $\tau \in [0,1]$. Similarly, when $\|\mathbf{g}_t\| < \epsilon_g$,

$$F(\mathbf{x}_t+\mathbf{s}_t)-F(\mathbf{x}_t)-m_t(\mathbf{s}_t) \leq \langle \mathbf{s}_t, \nabla F(\mathbf{x}_t)\rangle + \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{L_F}{2}-\frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3.$$

A.3. Proof of Lemma 13

If $\|\mathbf{s}_t\| \le \mathbf{s}_e^E$, then based on the condition of $\sigma_t \ge 2L_F$, we have

$$\frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{1}{2}L_F - \frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3 \le \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 \le \frac{\delta_H}{2}\|\mathbf{s}_t^E\|^2.$$

When $\|\mathbf{s}_t\| > \|\mathbf{s}_e^E\|$, because $L_f \le \sigma_t/2$,

$$\begin{aligned}\frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 + \left(\frac{1}{2}L_F - \frac{\sigma_t}{3}\right)\|\mathbf{s}_t\|^3 &\le \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 - \frac{\sigma_t}{12}\|\mathbf{s}_t\|^3\\ &\le \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 - \frac{\sigma_t}{12}\|\mathbf{s}_t^E\|\|\mathbf{s}_t\|^2\\ &\le \frac{1}{2}\delta_H\|\mathbf{s}_t\|^2 - \frac{\nu|\lambda_{\min}(\mathbf{H}_t)|}{12}\|\mathbf{s}_t\|^2\\ &\le \big((1-\eta)\nu|\lambda_{\min}(\mathbf{H}_t)| - \nu|\lambda_{\min}(\mathbf{H}_t)|\big)\|\mathbf{s}_t\|^2/12\\ &\le 0 \le \frac{\delta_H}{2}\|\mathbf{s}_t^E\|^2,\end{aligned}$$

where the third and fourth inequalities follow from (17a) and (15), respectively.

Endnote

[1] Because logistic loss, which is the "standard" loss used in this task, leads to a convex problem, we use square loss to obtain a nonconvex objective.

References

Ascher U, Greif C (2011) *A First Course on Numerical Methods*. Computational Science and Engineering (Society for Industrial and Applied Mathematics).

Bandeira AS, Scheinberg K, Vicente LN (2014) Convergence of trust-region methods based on probabilistic models. *SIAM J. Optim.* 24(3):1238–1264.

Beck A (2017) *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization (Society for Industrial and Applied Mathematics, Philadelphia).

Berahas AS, Bollapragada R, Nocedal J (2017) An investigation of Newton-sketch and subsampled Newton methods. *Optimization Methods and Software*:1–20.

Blanchet J, Cartis C, Menickelly M, Scheinberg K (2019) Convergence rate analysis of a stochastic trust-region method via supermartingales. *INFORMS J. Optim.* 1(2):92–119.

Bollapragada R, Byrd RH, Nocedal J (2018) Exact and inexact subsampled Newton methods for optimization. *IMA J. Numerical Anal.* 39(2):545–578.

Carmon Y, Duchi JC (2016) Gradient descent efficiently finds the cubic-regularized non-convex Newton step. arXiv preprint arXiv:1612.00547.

Cartis C, Scheinberg K (2018) Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Math. Programming* 169(2):337–375.

Cartis C, Gould NI, Toint PL (2010) On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. *SIAM J. Optim.* 20(6):2833–2852.

Cartis C, Gould NI, Toint PL (2011a) Adaptive cubic regularisation methods for unconstrained optimization. Part I: Motivation, convergence and numerical results. *Math. Programming* 127(2):245–295.

Cartis C, Gould NI, Toint PL (2011b) Adaptive cubic regularisation methods for unconstrained optimization. Part II: Worst-case function-and derivative-evaluation complexity. *Math. Programming* 130(2):295–319.

Cartis C, Gould NI, Toint PL (2011c) Optimal Newton-type methods for nonconvex smooth optimization problems. ERGO technical report 11-009, School of Mathematics, University of Edinburgh, Scotland.

Cartis C, Gould NI, Toint PL (2012) Complexity bounds for second-order optimality in unconstrained optimization. *J. Complexity* 28(1):93–108.

Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Systems Tech.* 2(27):1–27.

Chen R, Menickelly M, Scheinberg K (2015) Stochastic optimization using a trust-region method and random models. *Math. Program.* 169(2):447–487.

Chen X, Jiang B, Lin T, Zhang S (2018) Adaptively Accelerating Cubic Regularized Newton's Methods for Convex Optimization via Random Sampling. arXiv preprint arXiv:1802.05426.

Choromanska A, Henaff M, Mathieu M, Arous GB, LeCun Y (2015) The loss surfaces of multilayer networks. *Artificial Intelligence Statist.*, 192–204.

Conn AR, Gould NI, Toint PL (2000) *Trust Region Methods* (SIAM, Philadelphia).

Conn AR, Scheinberg K, Vicente LN (2009) Global convergence of general derivative-free trust-region algorithms to first-and second-order critical points. *SIAM J. Optim.* 20(1):387–415.

Dauphin YN, Pascanu R, Gulcehre C, Cho K, Ganguli S, Bengio Y (2014) Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Adv. Neural Inform. Processing Systems* 27:2933–2941.

Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. *J. Maching Learn. Res.* 12(July):2121–2159.

Fazlyab M, Robey A, Hassani H, Morari M, Pappas G (2019) Efficient and accurate estimation of Lipschitz constants for deep neural networks. *Adv. Neural Inform. Processing Systems* 33:11427–11438.

Fukumizu K, Amari S (2000) Local minima and plateaus in hierarchical structures of multilayer perceptrons. *Neural Networks* 13(3):317–327.

Ge R, Huang F, Jin C, Yuan Y (2015) Escaping from saddle points-online stochastic gradient for tensor decomposition. *COLT*, 797–842.

Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. *Proc. 14th Internat. Conf. Artificial Intelligence Statist.*, 315–323.</Conf>

Gratton S, Royer CW, Vicente LN, Zhang Z (2017) Complexity and global rates of trust-region methods based on probabilistic models. *IMA J. Numerical Anal.* 38(3):1579–1597.

Griewank A (1993) Some bounds on the complexity of gradients, Jacobians, and Hessians. *Complexity in Numerical Optimization* (World Scientific), 128–162.

He K, Zhang X, Ren S, Sun J (2016a) Deep residual learning for image recognition. *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 770–778.

He X, Mudigere D, Smelyanskiy M, Takác M (2016b) Large scale distributed Hessian-free optimization for deep neural network. arXiv preprint arXiv:1606.00511.

Hillar CJ, Lim LH (2013) Most tensor problems are NP-hard. *J. ACM* 60(6):1–39.

Jin C, Ge R, Netrapalli P, Kakade SM, Jordan MI (2017) How to escape saddle points efficiently. *Proc. 34th Internat. Conf. Machine Learn.*,vol. 70, 1724–1732.

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kiros R (2013) Training neural networks with stochastic Hessian-free optimization. arXiv:1301.3641.

Kohler JM, Lucchi A (2017) Sub-sampled cubic regularization for non-convex optimization. *Proc. 34th Internat. Conf. Machine Learn.*, vol. 70, 1895–1904.

Kylasa S, Roosta F, Mahoney MW, Grama A (2019) GPU accelerated sub-sampled Newton's method for convex classification problems. *Proc. 2019 SIAM Internat. Conf. Data Mining*, 702–710.

Lan G (2020) *First-order and Stochastic Optimization Methods for Machine Learning*. Springer Series in the Data Sciences (Springer International Publishing, Springer, Cham).

Larson J, Billups SC (2016) Stochastic derivative-free optimization using a trust region framework. *Comput. Optim. Appl.* 64(3):619–645.

LeCun YA, Bottou L, Orr GB, Müller KR (2012) Efficient backprop. *Neural Networks: Tricks of the Trade* (Springer, Berlin, Heidelberg), 9–48.

Levy KY (2016) The power of normalization: Faster evasion of saddle points. arXiv:1611.04831.

Lin Z, Li H, Fang C (2020) *Accelerated Optimization for Machine Learning: First-Order Algorithms* (Springer, Singapore).

Martens J (2010) Deep learning via Hessian-free optimization. *Internat. Conf. Machine Learn.* (Vol. 27, pp. 735–742).

Murty KG, Kabadi SN (1987) Some NP-complete problems in quadratic and nonlinear programming. *Math. Programming* 39(2):117–129.

Nesterov Y, Polyak BT (2006) Cubic regularization of Newton method and its global performance. *Math. Programming* 108(1):177–205.

Nocedal J, Wright S (2006) *Numerical Optimization* (Springer Science & Business Media, New York).

Pearlmutter BA (1994) Fast exact multiplication by the Hessian. *Neural Comput.* 6(1):147–160.

Pilanci M, Wainwright MJ (2015) Newton sketch: A linear-time optimization algorithm with linear-quadratic convergence. *SIAM J. Optimization* 27(1):205–245.

Regier J, Jordan MI, McAuliffe J (2017) Fast black-box variational inference through stochastic trust-region optimization. *Advances in Neural Information Processing Systems* (pp. 2402–2411).

Roosta F, Mahoney MW (2019) Sub-sampled Newton methods. *Math. Programming* 174(1–2):293–326.

Saxe AM, McClelland JL, Ganguli S (2013) Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv:1312.6120.

Shashaani S, Hashemi FS, Pasupathy R (2018) ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free stochastic optimization. *SIAM J. Optim.* 28(4):3145–3176.

Steihaug T (1983) The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numerical Anal.* 20(3):626–637.

Sutskever I, Martens J, Dahl G, Hinton G (2013) On the importance of initialization and momentum in deep learning. *Internat. Conf. Machine Learn.*, 1139–1147.

Swirszcz G, Czarnecki WM, Pascanu R (2016) Local minima in training of deep networks. arXiv:1611.06310.

Tripuraneni N, Stern M, Jin C, Regier J, Jordan MI (2018) Stochastic cubic regularization for fast nonconvex optimization. *Adv. Neural Inform. Processing Systems* 32:2899–2908.

Vinyals O, Povey D (2012) Krylov subspace descent for deep learning. *AISTATS*, 1261–1268.

Wiesler S, Li J, Xue J (2013) Investigations on Hessian-free optimization for cross-entropy training of deep neural networks. *INTERSPEECH*, 3317–3321.

Xu P, Roosta F, Mahoney MW (2019) Newton-type methods for non-convex optimization under inexact Hessian information. *Math. Programming* 184:35–70.

Xu P, Roosta F, Mahoney MW (2020) Second-order optimization for non-convex machine learning: An empirical study. *Proc. 2020 SIAM Internat. Conf. Data Mining* (SIAM).

Xu P, Yang J, Roosta F, Ré C, Mahoney MW (2016) Sub-sampled Newton methods with non-uniform sampling. *Adv. Neural Inform. Processing Systems* 29:3000–3008.