Journal of Machine Learning Research 24 (2023) 1-37 Submitted 5/22; Revised 3/23; Published 4/23

Restarted Nonconvex Accelerated Gradient Descent: No More Polylogarithmic Factor in the $\mathcal{O}(\epsilon^{-7/4})$ Complexity

Huan Li LIHUANSS@NANKAI.EDU.CN
Institute of Robotics and Automatic Information Systems
College of Artificial Intelligence
Nankai University
Tianjin 300071, China

Zhouchen Lin ZLIN@PKU.EDU.CN
National Key Lab of General AI, School of Intelligence Science and Technology, Peking University
Institute for Artificial Intelligence, Peking University, Beijing 100871, China
Peng Cheng Laboratory, Shenzhen 518055, China

Editor: Ohad Shamir

Abstract

This paper studies accelerated gradient methods for nonconvex optimization with Lipschitz continuous gradient and Hessian. We propose two simple accelerated gradient methods, restarted accelerated gradient descent (AGD) and restarted heavy ball (HB) method, and establish that our methods achieve an ϵ-approximate first-order stationary point within $\mathcal{O}(\epsilon^{-7/4})$ number of gradient evaluations by elementary proofs. Theoretically, our complexity does not hide any polylogarithmic factors, and thus it improves over the best known one by the $\mathcal{O}(\log \frac{1}{\epsilon})$ factor. Our algorithms are simple in the sense that they only consist of Nesterov's classical AGD or Polyak's HB iterations, as well as a restart mechanism. They do not invoke negative curvature exploitation or minimization of regularized surrogate functions as the subroutines. In contrast with existing analysis, our elementary proofs use less advanced techniques and do not invoke the analysis of strongly convex AGD or HB.

Keywords: accelerated gradient descent, heavy ball method, restart, nonconvex optimization, first-order stationary point

1. Introduction

Nonconvex optimization has become the foundation of training machine learning models and emerging machine learning tasks can be modeled as nonconvex problems. Typical examples include matrix completion (Hardt, 2014), one bit matrix completion (Davenport et al., 2014), robust PCA (Netrapalli et al., 2014), phase retrieval (Candès et al., 2015), and deep learning (LeCun et al., 2015). In this paper, we consider the following general nonconvex problem:

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

where $f(\mathbf{x})$ has Lipschitz continuous gradient and Hessian and it is bounded from below. Our goal is to find an ϵ-approximate first-order stationary point, defined as

$$\|\nabla f(\mathbf{x})\| \leq \epsilon.$$

1. Parts of this work appeared in ICML 2022 (Li and Lin, 2022). Corresponding author: Zhouchen Lin.

Gradient descent, a fundamental algorithm in machine learning, is commonly used due to its simplicity and practical efficiency. Theoretically, gradient descent is the optimal method among the first-order algorithms for nonconvex optimization under the assumption that the gradient is Lipschitz (Carmon et al., 2020), which means that we cannot find a first-order method with theoretically faster convergence rate under these conditions. When we assume additional structure, such as the Hessian Lipschitz geometry, improvement is possible. On the other hand, for convex optimization, gradient descent is known to be suboptimal and several accelerated gradient methods with theoretically faster convergence rate were proposed. Typical examples include Polyak's heavy ball (HB) method (Polyak, 1964) and Nesterov's accelerated gradient descent (AGD) (Nesterov, 1983, 1988, 2005). Motivated by the theoretical optimality and practical efficiency of convex AGD and HB, AGD and HB have been extended to nonconvex optimization (Carmon et al., 2018, 2017; Agarwal et al., 2017; Jin et al., 2018). But there are still some issues, such as the suboptimal convergence rate and complex algorithms and proofs. In this paper, we study the restarted AGD and HB method, variants of the original AGD and HB by employing a restart mechanism. Our aim is to establish a slightly faster convergence rate than the state-of-the-art accelerated methods by elementary analysis for the two simple methods.

1.1 Literature Review

In this section, we briefly review the convergence rates of gradient descent, accelerated gradient descent, and the heavy ball method for convex optimization, as well as the state-of-the-art accelerated methods for nonconvex optimization.

1.1.1 Accelerated Gradient Methods for Convex Optimization

For convex problems, gradient descent is known to converge to an ϵ-optimal solution within $\mathcal{O}(\frac{L}{\epsilon})$ and $\mathcal{O}(\frac{L}{\mu}\log\frac{1}{\epsilon})$ iterations for L-smooth convex problems and μ-strongly convex problems, respectively (Nesterov, 2004). Polyak's heavy ball method (Polyak, 1964) was the first accelerated first-order method, which finds an ϵ-optimal solution in $\mathcal{O}(\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon})$ steps when the objective function is twice continuously differentiable, L-smooth, μ-strongly convex, and the initializer is close enough to the minimum. Recently, Wang et al. (2022) extended HB to the case without the locality condition. However, the $\mathcal{O}(\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon})$ complexity only holds after $\mathcal{O}(\frac{L}{\mu})$ iterations. When strong convexity is absent, currently, only the $\mathcal{O}(\frac{L}{\epsilon})$ complexity is proved for smooth convex problems (Ghadimi et al., 2015), which is the same as gradient descent. In a series of celebrated works (Nesterov, 1983, 1988, 2005), Nesterov proposed several accelerated gradient descent methods. The same $\mathcal{O}(\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon})$ complexity is established for strongly convex problems without the twice continuous differentiability and locality assumptions. Moreover, when the objective is L-smooth and convex, Nesterov's accelerated methods find an ϵ-optimal solution in $\mathcal{O}(\sqrt{\frac{L}{\epsilon}})$ iterations, which is faster than gradient descent and the heavy ball method in theory. Nesterov's accelerated methods are proven to be optimal among the first-order methods for convex optimization (Nesterov, 2004). For more topics on accelerated methods for convex optimization, interested readers can refer to the survey paper (Li et al., 2020), for example.

1.1.2 Accelerated Gradient Methods to Achieve Nonconvex First-order Stationary Point

For nonconvex problems, gradient descent finds an ϵ-approximate first-order stationary point of problem (1) in $\mathcal{O}(\epsilon^{-2})$ iterations (Nesterov, 2004). Enormous amount of effort has been spent on speeding up gradient descent in the last decade. Zavriev and Kostyuk (1993); Ochs et al. (2014); Ochs (2018); Liang et al. (2016) studied the convergence of the HB method, while Ghadimi and Lan (2016); Li and Lin (2015); Li et al. (2017) studied AGD. The practical efficiency is verified empirically and there is no theoretical speedup under the assumption of Lipschitz gradient. With the additional Lipschitz Hessian assumption, Carmon et al. (2017) proposed a "convex until proven guilty" mechanism with nested-loop, which converges to an ϵ-approximate first-order stationary point within $\mathcal{O}(\epsilon^{-7/4}\log\frac{1}{\epsilon})$ gradient and function evaluations. Their method alternates between negative curvature exploitation and inexact minimization of a regularized surrogate function, where in the latter subroutine, Carmon et al. (2017) add a proximal term to reduce the nonconvex subproblem to a convex one and use the convex AGD to minimize it until the function is "guilty" of being nonconvex. When the third-order derivative of the objective is Lipschitz, the $\mathcal{O}(\epsilon^{-5/3}\log\frac{1}{\epsilon})$ complexity can be obtained (Carmon et al., 2017).

1.1.3 Accelerated Gradient Methods to Achieve Nonconvex Second-order Stationary Point

When studying nonconvex accelerated methods, most works concentrate on the second-order stationary point (see definition in (5)). Carmon et al. (2018) combined the Lanczos method and regularized accelerated gradient descent, where the former is used to compute the eigenvector corresponding to the smallest negative eigenvalue to search descent directions of negative curvature. Agarwal et al. (2017) implemented the cubic regularized Newton method (Nesterov and Polyak, 2006) carefully and computed the descent direction using accelerated method for fast approximate matrix inversion, while Carmon and Duchi (2020, 2018) employed the Krylov subspace method to solve the cubic regularized Newton subproblems. The above methods find an ϵ-approximate second-order stationary point with probability at least $1-\delta$ in $\mathcal{O}(\epsilon^{-7/4}\log\frac{d}{\epsilon\delta})$ gradient and Hessian-vector product evaluations[1], where d is the dimension of $\mathbf{x}$ in problem (1). To avoid the Hessian-vector products, Xu et al. (2018) and Allen-Zhu and Li (2018) proposed the NEON and NEON2 first-order procedures to extract directions of negative curvature from the Hessian, respectively, which can be used to turn a first-order stationary point finding algorithm into a second-order stationary point finding one. Other typical algorithms include the Newton-conjugate gradient (Royer et al., 2020) and the second-order line-search method (Royer and Wright, 2018), which are beyond the class of accelerated methods.

The above methods are nested-loop algorithms, where the outer loop needs to call a serious of subroutines such as negative curvature exploitation, minimization of regularized surrogate functions using convex AGD (Carmon et al., 2018, 2017), or computation of cubic regularized Newton directions (Agarwal et al., 2017; Carmon and Duchi, 2020, 2018). Jin et al. (2018) proposed the first single-loop accelerated method, which also finds an ϵ-approximate second-order stationary point in $\mathcal{O}(\epsilon^{-7/4}\log\frac{d}{\epsilon\delta})$ gradient and function computations with

1. Carmon et al. (2018) use $\nabla^2 f(\mathbf{x})\mathbf{v} = \lim_{h\to 0}\frac{\nabla f(\mathbf{x}+h\mathbf{v})-\nabla f(\mathbf{x})}{h}$ to approximate the Hessian-vector product.

probability at least $1-\delta$. The algorithm in (Jin et al., 2018) runs the classical AGD until the function becomes "too nonconvex" locally, then it calls negative curvature exploitation. To the best of our knowledge, it is the simplest method among the nonconvex accelerated algorithms with fast rate guarantees.

Although achieving second-order stationary point guarantees the method to escape strict saddle points, some researchers show that gradient descent and its accelerated variants that converge to first-order stationary point always converge to local minimum. Lee et al. (2016) proved that gradient descent converges to a local minimizer almost surely with random initialization. Sun et al. (2019) gave the similar result for the heavy ball method. O'Neill and Wright (2019) examined the behavior of HB and AGD near strict saddle points and proved that both methods diverge from these points more rapidly than gradient descent for specific quadratic functions.

1.1.4 Lower Bound for Second-order Smooth Nonconvex Problems

Carmon et al. (2021) studied the lower bounds for finding stationary point using first-order methods. For nonconvex functions with Lipschitz continuous gradient and Hessian, they established that deterministic first-order methods cannot find ϵ-approximate first-order stationary points in less than $\mathcal{O}(\epsilon^{-12/7})$ gradient evaluations. There exists a gap of $\mathcal{O}(\epsilon^{-1/28}\log\frac{1}{\epsilon})$ between this lower bound and the best known upper bound (Carmon et al., 2017). It remains an open problem of how to close this gap. It is also unclear which of the upper bound and lower bound is tight (Carmon et al., 2021, Section 7).

1.2 Contribution

All the above accelerated algorithms (Carmon et al., 2017, 2018; Agarwal et al., 2017; Carmon and Duchi, 2020; Jin et al., 2018) share the state-of-the-art $\mathcal{O}(\epsilon^{-7/4}\log\frac{1}{\epsilon})$ complexity, which has a $\mathcal{O}(\log\frac{1}{\epsilon})$ factor. As far as we know, even when we apply the methods designed to find second-order stationary point to the easier problem of finding first-order stationary one, we still cannot remove the $\mathcal{O}(\log\frac{1}{\epsilon})$ factor. On the other hand, almost all the existing accelerated methods need to call additional subroutines and thus they are complex with nested loops. Even the single-loop method proposed in (Jin et al., 2018) requires negative curvature exploitation.

In this paper, we propose two simple accelerated methods, restarted AGD and restarted HB, which have the following three advantages:

1. Our algorithms find an ϵ-approximate first-order stationary point within $\mathcal{O}(\epsilon^{-7/4})$ number of gradient evaluations under the conditions that both the gradient and Hessian are Lipschitz continuous. We do not hide any polylogarithmic factors in our complexity, and thus it improves over the best known one by the $\mathcal{O}(\log\frac{1}{\epsilon})$ factor.

2. Our algorithms are simple in the sense that they only consist of Nesterov's classical AGD or Polyak's HB iterations, as well as a restart mechanism. They do not invoke negative curvature exploitation or minimization of regularized surrogate functions or computation of cubic regularized Newton directions as the subroutines.

3. Technically, our elementary proofs use less advanced techniques compared with existing works. Especially, it is irrelevant to the analysis of strongly convex AGD or HB, which is crucial to cancel the $\mathcal{O}(\log \frac{1}{\epsilon})$ factor.

1.3 Difference from Our Conference Paper

This paper extends our conference paper (Li and Lin, 2022) and we have rewritten the contents. The major difference is that we have added a new algorithm, the restarted heavy ball method, in this version. In contrast, our conference version only focused on the restarted AGD. Although we use the same proof framework for the two algorithms, the details are different, and it is not a parallel extension of our conference version. Specifically, see the proofs of Lemmas 11 and 19, as well as the comparison in Remark 20. Empirically, the HB-style momentum is more commonly used in the deep learning literature (Sutskever et al., 2013), and it is verified to outperform the AGD-style momentum in our experiments.

1.4 Notations and Assumptions

We use lowercase bold letters to represent vectors, uppercase bold letters for matrices, and non-bold (both lowercase and uppercase) letters for scalars. Denote $\mathbf{x}_j$ and $\nabla_j f(\mathbf{x})$ as the jth element of $\mathbf{x}$ and $\nabla f(\mathbf{x})$, respectively. For the vectors produced in the iterative algorithms, for example, $\mathbf{x}$, denote $\mathbf{x}^k$ to be the value at the kth iteration. We denote $\|\cdot\|$ to be the ℓ_2 Euclidean norm for vectors, $\|\cdot\|_2$ as the spectral norm and $\|\cdot\|_F$ as the Frobenius norm for matrices. We make the following standard assumptions in this paper.

Assumption 1 *1. $f(\mathbf{x})$ is L-gradient Lipschitz: $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,*

2. $f(\mathbf{x})$ is ρ-Hessian Lipschitz: $\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\|_2 \le \rho\|\mathbf{x}-\mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

which yield the following two well-known inequalities:

$$|f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle| \le \tfrac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2, \tag{2}$$

$$\big|f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle - (\mathbf{y}-\mathbf{x})^T\nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x})\big| \le \tfrac{\rho}{6}\|\mathbf{y}-\mathbf{x}\|^3. \tag{3}$$

We also assume that the objective function is lower bounded, that is, $\min_{\mathbf{x}} f(\mathbf{x}) > -\infty$.

2. Restarted Accelerated Gradient Descent

Nesterov's classical AGD consists of the following iterations:

$$\mathbf{y}^k = \mathbf{x}^k + (1-\theta)(\mathbf{x}^k - \mathbf{x}^{k-1}), \quad \mathbf{x}^{k+1} = \mathbf{y}^k - \eta\nabla f(\mathbf{y}^k),$$

where $\theta = \frac{2\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$ for strongly convex problems and it varies as $\frac{3}{k+2}$ at the kth iteration for convex problems. The term $(\mathbf{x}^k - \mathbf{x}^{k-1})$ is often regarded as momentum. When applying the above iteration to nonconvex problems, the major challenge in faster convergence analysis is that the objective function (even the Hamiltonian potential function used in (Jin et al., 2018)) does not decrease monotonically, especially when we set $\eta = \mathcal{O}(\frac{1}{L})$ and θ small (for example, of the order ϵ). To address this issue, Jin et al. (2018) invoke negative curvature

Algorithm 1 Restarted AGD for Nonconvex Optimization (RAGD-NC)

Initialize $\mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{x}_{int}$, $k = 0$.
while $k < K$ **do**
  $\mathbf{y}^k = \mathbf{x}^k + (1-\theta)(\mathbf{x}^k - \mathbf{x}^{k-1})$
  $\mathbf{x}^{k+1} = \mathbf{y}^k - \eta \nabla f(\mathbf{y}^k)$
  $k = k + 1$
  if $k \sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 > B^2$ **then**
    $\mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{x}^k$, $k = 0$
  end if
end while
$K_0 = \operatorname{argmin}_{\lfloor \frac{K}{2} \rfloor \le k \le K-1} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|$
Output $\hat{\mathbf{y}} = \frac{1}{K_0+1} \sum_{k=0}^{K_0} \mathbf{y}^k$

exploitation when the local objective function is very nonconvex. An open problem is asked in Section 5 of (Jin et al., 2018) whether negative curvature exploitation is indispensable to guarantee the fast rate. In contrast with (Jin et al., 2018), we use the restart mechanism to ensure the decrease of the objective function, and thus avoid negative curvature exploitation.

We present our method in Algorithm 1. It runs Nesterov's classical AGD iterations until the "if condition" triggers. Then we reset $\mathbf{x}^0$ and $\mathbf{x}^{-1}$ equal to $\mathbf{x}^k$ and continue to the next round of AGD. The method terminates and outputs a specific average when the "if condition" does not trigger in K iterations. To simplify the description, we define one round of AGD between two successive restarts to be one "epoch". The restart trick, first proposed in (O'Donoghue and Candès, 2015), is motivated by (Fang et al., 2019), where a ball-mechanism is proposed as the stopping criteria to analyze SGD.

Our main result is described in Theorem 1, which establishes the $\mathcal{O}(\epsilon^{-7/4})$ complexity to achieve an ϵ-approximate first-order stationary point. We defer the proofs until Section 4.1.

Theorem 1 *Suppose that Assumption 1 holds. Let* $\eta = \frac{1}{4L}$, $B = \sqrt{\frac{\epsilon}{\rho}}$, $\theta = 4(\epsilon\rho\eta^2)^{1/4} \in (0, 1]$, *and* $K = \frac{1}{\theta}$. *Then Algorithm 1 terminates in at most* $\frac{\triangle_f L^{1/2}\rho^{1/4}}{\epsilon^{7/4}}$ *gradient computations and the output satisfies* $\|\nabla f(\hat{\mathbf{y}})\| \le 82\epsilon$, *where* $\triangle_f = f(\mathbf{x}_{int}) - \min_{\mathbf{x}} f(\mathbf{x})$.

Among the existing methods, the "convex until proven guilty" method proposed in (Carmon et al., 2017) achieves an ϵ-approximate first-order stationary point in $\mathcal{O}(\frac{\triangle_f L^{1/2}\rho^{1/4}}{\epsilon^{7/4}} \log \frac{L\triangle_f}{\epsilon})$ gradient and function evaluations, which is slower than our method by the $\mathcal{O}(\log \frac{1}{\epsilon})$ factor. The complexity established in other work focusing on second-order stationary point, such as (Carmon et al., 2018; Agarwal et al., 2017; Carmon and Duchi, 2020; Jin et al., 2018), also has the additional $\mathcal{O}(\log \frac{1}{\epsilon})$ factor even when only pursuing first-order stationary point. Take (Jin et al., 2018) as the example. Their Lemma 7 concentrates on the first-order stationary point. They built the proofs of their Lemmas 9 and 17 upon the analysis of strongly convex AGD, which generally requires $\mathcal{O}(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon})$ iterations such that the gradient norm will be less than ϵ. Thus, the $\mathcal{O}(\log \frac{1}{\epsilon})$ factor appears.

Remark 2 *1. The specific average on lines 10 and 11 of Algorithm 1 is the crucial technique to remove the $\mathcal{O}(\log \frac{1}{\epsilon})$ factor. See the proof of Lemma 13. This phenomenon that some averaged iterate converges faster than the final iterate theoretically has also been observed in other algorithms. For example, for Lipschitz and strongly convex functions, but not necessarily differentiable, Shamir and Zhang (2013) proved the $\mathcal{O}(\frac{\log T}{T})$ error of the final iterate of SGD while the $\mathcal{O}(\frac{1}{T})$ one for the suffix averaged iterate. Both rates are tight matching the corresponding lower bounds (Harvey et al., 2019). For linearly constrained convex problems, Davis and Yin (2017) proved the $\mathcal{O}(\frac{1}{\sqrt{T}})$ rate for the final iterate of ADMM while the $\mathcal{O}(\frac{1}{T})$ one for the averaged iterate. The two rates are also tight (Davis and Yin, 2017).*

We can extend this technical trick to the method proposed in (Jin et al., 2018) and greatly simplify their proofs with the slightly faster $\mathcal{O}(\epsilon^{-7/4})$ convergence rate. See the supplementary material of our conference version (Li and Lin, 2022). On the other hand, we can also prove that the gradient at the last iterate in our method is small with norm being less than ϵ by employing the proof techniques in (Jin et al., 2018), at the expense of introducing the additional $\mathcal{O}(\log \frac{1}{\epsilon})$ factor and complicating the proofs.

2. Restart plays the role of decreasing the objective function at each epoch of AGD. See Corollary 12. Intuitively, when the iterates are far from the local starting point $\mathbf{x}^0$ or the momentum $\mathbf{x}^k - \mathbf{x}^{k-1}$ is large such that it may potentially increase the objective function, restart cancels the effect of momentum by setting it to 0.

3. As discussed in Section 4.2, since our proofs do not invoke the analysis of strongly convex AGD or HB, the acceleration mechanism for nonconvex optimization seems irrelevant to the analysis of convex AGD. Our proofs show that momentum and its parameter θ play an important role in the analysis of nonconvex acceleration mechanism.

2.1 Adaptive Implementation and Infrequent Restart

In Algorithm 1, we set B small in theory such that the method may restart frequently, making it almost reduce to the classical gradient descent, especially for high dimensional problems. To take advantage of the practical efficiency of AGD, we should reduce the frequency of restart. A straightforward idea is to set a large B initially and reduce it gradually. We present an adaptive implementation of Algorithm 1 in Algorithm 2, which relaxes the restart condition of $k\sum_{t=0}^{k-1}\|\mathbf{x}^{t+1}-\mathbf{x}^t\|^2 > B^2$ to $k\sum_{t=0}^{k-1}\|\mathbf{x}^{t+1}-\mathbf{x}^t\|^2 > \max\{B^2, B_0^2\}$, where B_0 can be initialized much larger than B and is decreased geometrically after each epoch. The decrease condition on line 8 of Algorithm 2 comes form Corollary 12. Intuitively, when $B_0 \le B$, we always have $f(\mathbf{x}^k) - f(\mathbf{x}^0) \le -\frac{7\epsilon^{3/2}}{8\sqrt{\rho}}$ from Corollary 12. That is, line 11 never executes when B_0 decreases to be smaller than B after $\mathcal{O}(\log_{c_0}\frac{1}{\epsilon})$ epochs and Algorithm 2 is equivalent to Algorithm 1 in this case. When the decrease condition on line 8 does not hold, which indicates that the algorithm may diverge, we discard the whole iterates in this epoch and go back to the last iterate of the previous epoch, which is stored in $\mathbf{x}_{cur}^0$. We terminate Algorithm 2 when $B_0 \le B$ and k equals to K. On the other hand, we output the one of $\mathbf{x}^K$ and $\hat{\mathbf{y}}$ with smaller gradient norm. In practice, the last iterate always converges faster than the averaged iterate. We describe the $\mathcal{O}(\epsilon^{-7/4})$ complexity of Algorithm 2 in Theorem 3 and defer the proofs until Section 4.3.

Algorithm 2 Adaptively Restarted AGD for Nonconvex Optimization (Ada-RAGD-NC)

1: Initialize $\mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{x}_{cur}^0 = \mathbf{x}_{int}$, $k = 0$, B_0.
2: **while** $k < K$ or $B_0 > B$ **do**
3: $\quad \mathbf{y}^k = \mathbf{x}^k + (1-\theta)(\mathbf{x}^k - \mathbf{x}^{k-1})$
4: $\quad \mathbf{x}^{k+1} = \mathbf{y}^k - \eta \nabla f(\mathbf{y}^k)$
5: $\quad k = k + 1$
6: $\quad$ **if** $k \sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 > \max\{B^2, B_0^2\}$ or $k > K$ **then**
7: $\quad\quad B_0 = B_0 / c_0$
8: $\quad\quad$ **if** $f(\mathbf{x}^k) - f(\mathbf{x}^0) \le -\gamma \frac{\epsilon^{3/2}}{\sqrt{\rho}}$ **then**
9: $\quad\quad\quad \mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{x}^k$, $\mathbf{x}_{cur}^0 = \mathbf{x}^k$, $k = 0$
10: $\quad\quad$ **else**
11: $\quad\quad\quad \mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{x}_{cur}^0$, $k = 0$, $B_0 = B_0 / c_1$
12: $\quad\quad$ **end if**
13: $\quad$ **end if**
14: **end while**
15: $K_0 = \operatorname{argmin}_{\lfloor \frac{K}{2} \rfloor \le k \le K-1} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|$
16: $\hat{\mathbf{y}} = \frac{1}{K_0+1} \sum_{k=0}^{K_0} \mathbf{y}^k$
17: Output $\mathbf{x}_{out} = \operatorname{argmin}_{\mathbf{x}^K, \hat{\mathbf{y}}} \{\|\nabla f(\mathbf{x}^K)\|, \|\nabla f(\hat{\mathbf{y}})\|\}$

Theorem 3 *Suppose that Assumption 1 holds. Let* $\eta = \frac{1}{4L}$, $B = \sqrt{\frac{\epsilon}{\rho}}$, $\theta = 4(\epsilon\rho\eta^2)^{1/4} \in (0,1)$, $K = \lfloor \frac{1}{\theta} \rfloor$, $\gamma \le \frac{7}{8}$, $c_0 > 1$, *and* $c_1 > 1$. *Then Algorithm 2 terminates in at most* $\mathcal{O}\left(\frac{\triangle_f L^{1/2} \rho^{1/4}}{\epsilon^{7/4}} + \frac{L^{1/2}}{\epsilon^{1/4}\rho^{1/4}} \log \frac{\rho B_0}{\epsilon}\right)$ *gradient computations and* $\mathcal{O}\left(\frac{\triangle_f \sqrt{\rho}}{\epsilon^{3/2}} + \log \frac{\rho B_0}{\epsilon}\right)$ *function evaluations, and the output satisfies* $\|\nabla f(\mathbf{x}_{out})\| \le \mathcal{O}(\epsilon)$.

Remark 4 *Algorithm 2 also applies to the case when the Lipschitz constants* L *and* ρ *are unknown. We can initialize a small guess of* ρ', *tune an appropriate* η, *and replace line 11 of Algorithm 2 by the following steps:*

$$\mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{x}_{cur}^0, \quad k = 0, \quad B_0 = \frac{B_0}{c_1}, \quad \eta = \max\left(\frac{\eta}{c_2}, \eta_{min}\right), \quad \rho' = \min(\rho' c_2^2, \rho'_{max}), \tag{4}$$

where $c_1 \ge c_2 > 1$, *and the output also satisfies* $\|\nabla f(\mathbf{x}_{out})\| \le \mathcal{O}(\epsilon)$ *within* $\mathcal{O}(\epsilon^{-7/4})$ *gradient computations and* $\mathcal{O}(\epsilon^{-3/2})$ *function evaluations. See Theorem 15 in Section 4.3 for the details.*

2.2 Extension to the Second-order Stationary Point

Our restarted AGD can also find ϵ-approximate second-order stationary point, namely a point $\mathbf{x}$ that satisfies

$$\|\nabla f(\mathbf{x})\| \le \epsilon \quad \text{and} \quad \lambda_{min}(\nabla^2 f(\mathbf{x})) \ge -\sqrt{\epsilon\rho}, \tag{5}$$

where λ_{min} means the smallest eigenvalue. We follow (Jin et al., 2017, 2018) to add perturbations to the iterates. Specifically, we only need to replace line 7 of Algorithm 1 by

the following step:

$$\mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{x}^k + \xi 1_{\|\nabla f(\mathbf{y}^{k-1})\| \le \frac{B}{\eta}}, \quad \xi \sim \text{Unif}(\mathbb{B}_0(r)), \quad k = 0, \tag{6}$$

where $\text{Unif}(\mathbb{B}_0(r))$ means the uniform distribution in the ball $\mathbb{B}_0(r)$ with radius r and center 0, and $1_{\|\nabla f(\mathbf{y}^{k-1})\| \le \frac{B}{\eta}} = \begin{cases} 1, & \text{if } \|\nabla f(\mathbf{y}^{k-1})\| \le \frac{B}{\eta}, \\ 0, & \text{otherwise.} \end{cases}$

The convergence and complexity is presented in Theorem 5. We see that the perturbed RAGD-NC needs at most $\mathcal{O}(\epsilon^{-7/4} \log \frac{d}{\zeta\epsilon})$ gradient evaluations to find an ϵ-approximate second-order stationary point with probability at least $1 - \zeta$, where d is the dimension of $\mathbf{x}$ in problem (1). Our algorithm has the same complexity with the one given in (Jin et al., 2018). Comparing with Theorem 1, we see that this complexity is higher by the $\mathcal{O}(\log \frac{d}{\zeta\epsilon})$ factor. Currently, it is unclear how to cancel it, and we conjecture that the polylogarithmic factor may not be removed when pursuing second-order stationary point (Simchowitz et al., 2017).

Theorem 5 *Suppose that Assumption 1 holds. Let* $\chi = \mathcal{O}(\log \frac{d}{\zeta\epsilon}) \ge 1$, $\eta = \frac{1}{4L}$, $B = \frac{1}{288\chi^2}\sqrt{\frac{\epsilon}{\rho}}$, $\theta = \frac{1}{2}\left(\frac{\epsilon\rho}{L^2}\right)^{1/4} < 1$, $K = \frac{2\chi}{\theta}$, $r = \min\{\frac{B}{2}, \frac{\theta B}{20K}, \sqrt{\frac{\theta B^2}{2K}}\} = \mathcal{O}(\epsilon)$. *Then the perturbed RAGD-NC (Algorithm 1 with (6)) terminates in at most* $\mathcal{O}\left(\frac{\triangle_f L^{1/2}\rho^{1/4}\chi^6}{\epsilon^{7/4}}\right)$ *gradient computations and the output satisfies* $\|\nabla f(\hat{\mathbf{y}})\| \le \epsilon$, *where* $\triangle_f = f(\mathbf{x}_{int}) - \min_{\mathbf{x}} f(\mathbf{x})$. *It also satisfies* $\lambda_{min}(\nabla^2 f(\hat{\mathbf{y}})) \ge -1.011\sqrt{\epsilon\rho}$ *with probability at least* $1 - \zeta$.

The proof of this theorem is essentially identical to those in (Jin et al., 2018). We omit the proofs and they can be found in the supplementary material of our conference version (Li and Lin, 2022).

3. Restarted Heavy Ball Method

Polyak's classical heavy ball method (Polyak, 1964) iterates with the following step

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \eta\nabla f(\mathbf{x}^k) + (1-\theta)(\mathbf{x}^k - \mathbf{x}^{k-1}),$$

where $\eta = \frac{4}{(\sqrt{L}+\sqrt{\mu})^2}$ and $1 - \theta = \frac{(\sqrt{L}-\sqrt{\mu})^2}{(\sqrt{L}+\sqrt{\mu})^2}$ for strongly convex problems. In the deep learning literature, people often use the following equivalent iterations empirically with the running average (Sutskever et al., 2013),

$$\mathbf{m}^k = \beta\mathbf{m}^{k-1} + \nabla f(\mathbf{x}^k), \quad \mathbf{x}^{k+1} = \mathbf{x}^k - \eta\mathbf{m}^k,$$

where $\mathbf{m}^{-1} = 0$ and $\beta = 1 - \theta$ for the deterministic problems. When applying the heavy ball iteration to nonconvex optimization, people often set $\eta = \mathcal{O}(\frac{\theta}{L})$ to ensure the convergence (Ochs et al., 2014; Sun et al., 2019), which prevents us from proving faster convergence in theory and slows down the algorithm in practice when θ is small. To address this issue, similar to RAGD-NC, we combine the restart mechanism with the heavy ball method such that $\eta = \mathcal{O}(\frac{1}{L})$ while maintaining θ small. Our method is presented in Algorithm 3. It runs Polyak's classical HB iteration until the "if condition" triggers. Then we restart from the

Algorithm 3 Restarted HB for Nonconvex Optimization (RHB-NC)

Initialize $\mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{x}_{int}$, $k = 0$.
while $k < K$ **do**
  $\mathbf{x}^{k+1} = \mathbf{x}^k - \eta \nabla f(\mathbf{x}^k) + (1-\theta)(\mathbf{x}^k - \mathbf{x}^{k-1})$
  $k = k + 1$
  if $k \sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 > B^2$ **then**
    $\mathbf{z}^k = \frac{\mathbf{x}^k + (1-2\theta)(1-\theta)\mathbf{x}^{k-1}}{1+(1-2\theta)(1-\theta)}$
    $\mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{z}^k$, $k = 0$
  end if
end while
$K_0 = \operatorname{argmin}_{\lfloor \frac{K}{2} \rfloor \le k \le K-1} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|$
Output $\hat{\mathbf{x}} = \frac{1}{K_0+1} \sum_{k=0}^{K_0} \mathbf{x}^k$

auxiliary vector $\mathbf{z}^k$, a convex combination of $\mathbf{x}^k$ and $\mathbf{x}^{k-1}$, and do the next round of HB iterations. Algorithm 3 shares almost the same framework as Algorithm 1, and the only difference comes from the iterate $\mathbf{z}^k$, which is designed to fit the proof. See Remark 20 for the detailed reason.

The main result is given in Theorem 6, which also establishes the $\mathcal{O}(\epsilon^{-7/4})$ complexity to find an ϵ-approximate first-order stationary point, and we defer the proofs until Section 4.4. Comparing with Theorem 1, we see that the two algorithms need the same assumptions, share the same convergence rate, and have almost the same parameter settings, which indicate that no one is superior to the other in theory for nonconvex optimization. As a comparison, the heavy ball method requires more assumptions for strongly convex problems and has the slower convergence rate in theory for convex problems than AGD.

Theorem 6 *Suppose that Assumption 1 holds. Let* $\eta = \frac{1}{4L}$, $B = \sqrt{\frac{\epsilon}{4\rho}}$, $\theta = 10\left(\epsilon\rho\eta^2\right)^{1/4} \in (0, \frac{1}{10}]$, *and* $K = \frac{1}{\theta}$. *Then Algorithm 3 terminates in at most* $\frac{\triangle_f L^{1/2}\rho^{1/4}}{\epsilon^{7/4}}$ *gradient computations and the output satisfies* $\|\nabla f(\hat{\mathbf{x}})\| \le 242\epsilon$, *where* $\triangle_f = f(\mathbf{x}_{int}) - \min_{\mathbf{x}} f(\mathbf{x})$.

In practice, Algorithm 3 has the same disadvantages as Algorithm 1 when B is small. Similar to Algorithm 2, we also propose an adaptive implementation of Algorithm 3, and present it in Algorithm 4. Theorem 7 gives the $\mathcal{O}(\epsilon^{-7/4})$ complexity.

Theorem 7 *Suppose that Assumption 1 holds. Let* $\eta = \frac{1}{4L}$, $B = \sqrt{\frac{\epsilon}{4\rho}}$, $\theta = 10\left(\epsilon\rho\eta^2\right)^{1/4} \in (0, \frac{1}{10}]$, $K = \lfloor \frac{1}{\theta} \rfloor$, $\gamma \le 1$, $c_0 > 1$, *and* $c_1 > 1$. *Then Algorithm 4 terminates in at most* $\mathcal{O}\left(\frac{\triangle_f L^{1/2}\rho^{1/4}}{\epsilon^{7/4}} + \frac{L^{1/2}}{\epsilon^{1/4}\rho^{1/4}} \log \frac{\rho B_0}{\epsilon}\right)$ *gradient computations and* $\mathcal{O}\left(\frac{\triangle_f \sqrt{\rho}}{\epsilon^{3/2}} + \log \frac{\rho B_0}{\epsilon}\right)$ *function evaluations, and the output satisfies* $\|\nabla f(\mathbf{x}_{out})\| \le \mathcal{O}(\epsilon)$.

4. Proof of the Theorems

We prove Theorems 1, 3, and 6 in this section. The proof of Theorem 7 is almost the same to that of Theorem 3 and we omit the details.

Algorithm 4 Adaptively Restarted HB for Nonconvex Optimization (Ada-RHB-NC)

1: Initialize $\mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{x}_{cur}^0 = \mathbf{x}_{int}$, $k = 0$, B_0.
2: **while** $k < K$ or $B_0 > B$ **do**
3: $\mathbf{x}^{k+1} = \mathbf{x}^k - \eta\nabla f(\mathbf{x}^k) + (1-\theta)(\mathbf{x}^k - \mathbf{x}^{k-1})$
4: $k = k + 1$
5: **if** $k\sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 > \max\{B^2, B_0^2\}$ or $k > K$ **then**
6: $\mathbf{z}^k = \frac{\mathbf{x}^k + (1-2\theta)(1-\theta)\mathbf{x}^{k-1}}{1+(1-2\theta)(1-\theta)}$
7: $B_0 = B_0/c_0$
8: **if** $f(\mathbf{z}^k) - f(\mathbf{x}^0) \le -\gamma\frac{\epsilon^{3/2}}{\sqrt{\rho}}$ **then**
9: $\mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{z}^k$, $\mathbf{x}_{cur}^0 = \mathbf{z}^k$, $k = 0$
10: **else**
11: $\mathbf{x}^{-1} = \mathbf{x}^0 = \mathbf{x}_{cur}^0$, $k = 0$, $B_0 = B_0/c_1$
12: **end if**
13: **end if**
14: **end while**
15: $K_0 = \mathrm{argmin}_{\lfloor \frac{K}{2} \rfloor \le k \le K-1} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|$
16: $\hat{\mathbf{x}} = \frac{1}{K_0+1}\sum_{k=0}^{K_0} \mathbf{x}^k$
17: Output $\mathbf{x}_{out} = \mathrm{argmin}_{\mathbf{x}^K, \hat{\mathbf{x}}}\{\|\nabla f(\mathbf{x}^K)\|, \|\nabla f(\hat{\mathbf{x}})\|\}$

4.1 Proof of Theorem 1

We prove the convergence rate of Algorithm 1 in this section. Denote $\mathcal{K}$ to be the iteration number when the "if condition" on line 6 of Algorithm 1 triggers, that is,

$$\mathcal{K} = \min_k \left\{ k \middle| k\sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 > B^2 \right\}. \tag{7}$$

For each epoch consisting of one round of AGD from iterations $k = 0$ to $k = \mathcal{K}$, we have

$$1 \le \mathcal{K} \le K, \qquad \mathcal{K}\sum_{t=0}^{\mathcal{K}-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 > B^2, \qquad \text{and} \tag{8a}$$

$$\|\mathbf{x}^k - \mathbf{x}^0\|^2 = \left\|\sum_{t=0}^{k-1} \mathbf{x}^{t+1} - \mathbf{x}^t\right\|^2 \le k\sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \le B^2, \forall k < \mathcal{K}, \tag{8b}$$

where the last inequality comes from the definition of $\mathcal{K}$. From the update of $\mathbf{y}$ on line 3 of Algorithm 1, we also have

$$\|\mathbf{y}^k - \mathbf{x}^0\| \le \|\mathbf{x}^k - \mathbf{x}^0\| + \|\mathbf{x}^k - \mathbf{x}^{k-1}\| \le 2B, \forall k < \mathcal{K}. \tag{9}$$

On the other hand, for the last epoch where the "if condition" does not trigger and the while loop breaks when k increases to K, we have

$$\|\mathbf{x}^k - \mathbf{x}^0\|^2 \le k\sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \le B^2, \forall k \le K, \tag{10a}$$

$$\|\mathbf{y}^k - \mathbf{x}^0\| \le 2B, \forall k \le K. \tag{10b}$$

We will show that the function value decreases at least $\mathcal{O}(\epsilon^{1.5})$ in each epoch except the last one in Sections 4.1.1 and 4.1.2. Thus, Algorithm 1 terminates in at most $\mathcal{O}(\epsilon^{-1.5})$ epochs. Since each epoch needs at most $\mathcal{O}(\epsilon^{-0.25})$ iterations, Algorithm 1 requires at most $\mathcal{O}(\epsilon^{-1.75})$ total gradient evaluations. In the last epoch, we will show in Section 4.1.3 that the gradient norm at the output iterate is less than $\mathcal{O}(\epsilon)$.

4.1.1 Large Gradient of $\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\|$

We first consider the case when $\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\|$ is large.

Lemma 8 *Suppose that Assumption 1 holds. Let $\eta \le \frac{1}{4L}$ and $0 \le \theta \le 1$. In each epoch of Algorithm 1 where the "if condition" triggers, when $\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\| > \frac{B}{\eta}$, we have*

$$f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \le -\frac{B^2}{4\eta}.$$

Proof As the gradient is L-Lipschitz, we have

$$\begin{aligned} f(\mathbf{x}^{k+1}) \le & f(\mathbf{y}^k) + \left\langle \nabla f(\mathbf{y}^k), \mathbf{x}^{k+1} - \mathbf{y}^k \right\rangle + \frac{L}{2}\|\mathbf{x}^{k+1} - \mathbf{y}^k\|^2 \\ = & f(\mathbf{y}^k) - \eta\|\nabla f(\mathbf{y}^k)\|^2 + \frac{L\eta^2}{2}\|\nabla f(\mathbf{y}^k)\|^2 \\ \le & f(\mathbf{y}^k) - \frac{7\eta}{8}\|\nabla f(\mathbf{y}^k)\|^2, \end{aligned} \tag{11}$$

where we use the AGD iteration on line 4 of Algorithm 1 and $\eta \le \frac{1}{4L}$. From the L-gradient Lipschitz, we also have

$$f(\mathbf{x}^k) \ge f(\mathbf{y}^k) + \left\langle \nabla f(\mathbf{y}^k), \mathbf{x}^k - \mathbf{y}^k \right\rangle - \frac{L}{2}\|\mathbf{x}^k - \mathbf{y}^k\|^2.$$

So we have

$$\begin{aligned} & f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k) \\ \le & -\left\langle \nabla f(\mathbf{y}^k), \mathbf{x}^k - \mathbf{y}^k \right\rangle + \frac{L}{2}\|\mathbf{x}^k - \mathbf{y}^k\|^2 - \frac{7\eta}{8}\|\nabla f(\mathbf{y}^k)\|^2 \\ = & \frac{1}{\eta}\left\langle \mathbf{x}^{k+1} - \mathbf{y}^k, \mathbf{x}^k - \mathbf{y}^k \right\rangle + \frac{L}{2}\|\mathbf{x}^k - \mathbf{y}^k\|^2 - \frac{7\eta}{8}\|\nabla f(\mathbf{y}^k)\|^2 \\ = & \frac{1}{2\eta}\left(\|\mathbf{x}^{k+1} - \mathbf{y}^k\|^2 + \|\mathbf{x}^k - \mathbf{y}^k\|^2 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2\right) + \frac{L}{2}\|\mathbf{x}^k - \mathbf{y}^k\|^2 - \frac{7\eta}{8}\|\nabla f(\mathbf{y}^k)\|^2 \\ \overset{a}{\le} & \frac{5}{8\eta}\|\mathbf{x}^k - \mathbf{y}^k\|^2 - \frac{1}{2\eta}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 - \frac{3\eta}{8}\|\nabla f(\mathbf{y}^k)\|^2 \\ \overset{b}{\le} & \frac{5}{8\eta}\|\mathbf{x}^k - \mathbf{x}^{k-1}\|^2 - \frac{1}{2\eta}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 - \frac{3\eta}{8}\|\nabla f(\mathbf{y}^k)\|^2, \end{aligned}$$

where we use $L \le \frac{1}{4\eta}$ in $\overset{a}{\le}$ and $\|\mathbf{x}^k - \mathbf{y}^k\| = (1-\theta)\|\mathbf{x}^k - \mathbf{x}^{k-1}\| \le \|\mathbf{x}^k - \mathbf{x}^{k-1}\|$ in $\overset{b}{\le}$. Summing over $k = 0, \cdots, \mathcal{K}-1$ and using $\mathbf{x}^0 = \mathbf{x}^{-1}$, we have

$$f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \le \frac{1}{8\eta}\sum_{k=0}^{\mathcal{K}-2}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 - \frac{3\eta}{8}\sum_{k=0}^{\mathcal{K}-1}\|\nabla f(\mathbf{y}^k)\|^2$$

$$\overset{c}{\leq}\frac{B^2}{8\eta}-\frac{3\eta}{8}\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\|^2\overset{d}{\leq}\frac{B^2}{8\eta}-\frac{3B^2}{8\eta}=-\frac{B^2}{4\eta},$$

where we use (8b) in $\overset{c}{\leq}$ and $\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\|>\frac{B}{\eta}$ in $\overset{d}{\leq}$. ■

4.1.2 Small Gradient of $\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\|$

If $\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\|\leq\frac{B}{\eta}$, then from the AGD iteration on line 4 and (9) we have

$$\|\mathbf{x}^{\mathcal{K}}-\mathbf{x}^0\|\leq\|\mathbf{y}^{\mathcal{K}-1}-\mathbf{x}^0\|+\eta\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\|\leq 3B.$$

For each epoch, denote $\mathbf{H}=\nabla^2 f(\mathbf{x}^0)$ to be the Hessian matrix at the starting iterate and $\mathbf{H}=\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ to be its eigenvalue decomposition with $\mathbf{U},\mathbf{\Lambda}\in\mathbb{R}^{d\times d}$. Let λ_j be the jth eigenvalue. Define $\widetilde{\mathbf{x}}=\mathbf{U}^T\mathbf{x}$, $\widetilde{\mathbf{y}}=\mathbf{U}^T\mathbf{y}$, and $\widetilde{\nabla}f(\mathbf{y})=\mathbf{U}^T\nabla f(\mathbf{y})$. As the Hessian is ρ-Lipschitz, we have

$$\begin{aligned}f(\mathbf{x}^{\mathcal{K}})-f(\mathbf{x}^0)\leq&\left\langle\nabla f(\mathbf{x}^0),\mathbf{x}^{\mathcal{K}}-\mathbf{x}^0\right\rangle+\frac{1}{2}(\mathbf{x}^{\mathcal{K}}-\mathbf{x}^0)^T\mathbf{H}(\mathbf{x}^{\mathcal{K}}-\mathbf{x}^0)+\frac{\rho}{6}\|\mathbf{x}^{\mathcal{K}}-\mathbf{x}^0\|^3\\=&\left\langle\widetilde{\nabla}f(\mathbf{x}^0),\widetilde{\mathbf{x}}^{\mathcal{K}}-\widetilde{\mathbf{x}}^0\right\rangle+\frac{1}{2}(\widetilde{\mathbf{x}}^{\mathcal{K}}-\widetilde{\mathbf{x}}^0)^T\mathbf{\Lambda}(\widetilde{\mathbf{x}}^{\mathcal{K}}-\widetilde{\mathbf{x}}^0)+\frac{\rho}{6}\|\mathbf{x}^{\mathcal{K}}-\mathbf{x}^0\|^3\\\leq&g(\widetilde{\mathbf{x}}^{\mathcal{K}})-g(\widetilde{\mathbf{x}}^0)+4.5\rho B^3,\end{aligned}\tag{12}$$

where we denote

$$\begin{aligned}g(\mathbf{x})=&\left\langle\widetilde{\nabla}f(\mathbf{x}^0),\mathbf{x}-\widetilde{\mathbf{x}}^0\right\rangle+\frac{1}{2}(\mathbf{x}-\widetilde{\mathbf{x}}^0)^T\mathbf{\Lambda}(\mathbf{x}-\widetilde{\mathbf{x}}^0)=\sum_{j=1}^{d}g_j(\mathbf{x}_j),\\g_j(x)=&\left\langle\widetilde{\nabla}_j f(\mathbf{x}^0),x-\widetilde{\mathbf{x}}_j^0\right\rangle+\frac{1}{2}\lambda_j(x-\widetilde{\mathbf{x}}_j^0)^2.\end{aligned}\tag{13}$$

Denoting

$$\widetilde{\delta}_j^k=\widetilde{\nabla}_j f(\mathbf{y}^k)-\nabla g_j(\widetilde{\mathbf{y}}_j^k),\qquad\widetilde{\delta}^k=\widetilde{\nabla}f(\mathbf{y}^k)-\nabla g(\widetilde{\mathbf{y}}^k),$$

then the AGD iterations in Algorithm 1 can be rewritten as

$$\widetilde{\mathbf{y}}_j^k=\widetilde{\mathbf{x}}_j^k+(1-\theta)(\widetilde{\mathbf{x}}_j^k-\widetilde{\mathbf{x}}_j^{k-1}),\tag{14a}$$

$$\widetilde{\mathbf{x}}_j^{k+1}=\widetilde{\mathbf{y}}_j^k-\eta\widetilde{\nabla}_j f(\mathbf{y}^k)=\widetilde{\mathbf{y}}_j^k-\eta\nabla g_j(\widetilde{\mathbf{y}}_j^k)-\eta\widetilde{\delta}_j^k,\tag{14b}$$

and $\|\widetilde{\delta}^k\|$ can be bounded as

$$\begin{aligned}\|\widetilde{\delta}^k\|=&\|\widetilde{\nabla}f(\mathbf{y}^k)-\widetilde{\nabla}f(\mathbf{x}^0)-\mathbf{\Lambda}(\widetilde{\mathbf{y}}^k-\widetilde{\mathbf{x}}^0)\|\\=&\|\nabla f(\mathbf{y}^k)-\nabla f(\mathbf{x}^0)-\mathbf{H}(\mathbf{y}^k-\mathbf{x}^0)\|\\=&\left\|\left(\int_0^1\nabla^2 f(\mathbf{x}^0+t(\mathbf{y}^k-\mathbf{x}^0))-\mathbf{H}\right)(\mathbf{y}^k-\mathbf{x}^0)dt\right\|\\\leq&\frac{\rho}{2}\|\mathbf{y}^k-\mathbf{x}^0\|^2\leq 2\rho B^2\end{aligned}\tag{15}$$

for any $k < \mathcal{K}$, where we use the ρ-Lipschitz Hessian assumption and (9) in the last two inequalities, respectively.

Thanks to (12), to prove the decrease from $f(\mathbf{x}^0)$ to $f(\mathbf{x}^{\mathcal{K}})$, we only need to study the decrease of $g(\mathbf{x})$. Iterations (14a) and (14b) can be regarded as applying AGD to the quadratic approximation $g(\mathbf{x})$ coordinately with the approximation error $\widetilde{\delta}^k$, where the later can be controlled within $\mathcal{O}(\rho B^2)$. The quadratic approximation $g(\mathbf{x})$ equals to the sum of d scalar functions $g_j(\mathbf{x}_j)$. We decompose $g(\mathbf{x})$ into $\sum_{j\in\mathcal{S}_1} g_j(\mathbf{x}_j)$ and $\sum_{j\in\mathcal{S}_2} g_j(\mathbf{x}_j)$, where

$$\mathcal{S}_1 = \left\{ j : \lambda_j \geq -\frac{\theta}{\eta} \right\} \quad \text{and} \quad \mathcal{S}_2 = \left\{ j : \lambda_j < -\frac{\theta}{\eta} \right\}.$$

We see that $g_j(x)$ is approximate convex when $j \in \mathcal{S}_1$, and strongly concave when $j \in \mathcal{S}_2$. We will prove the approximate decrease of $g_j(\mathbf{x}_j)$ in the above two cases. We first consider $\sum_{j\in\mathcal{S}_1} g_j(\mathbf{x}_j)$ in the following lemma.

Lemma 9 *Suppose that Assumption 1 holds. Let $\eta \leq \frac{1}{4L}$ and $0 < \theta \leq 1$. In each epoch of Algorithm 1 where the "if condition" triggers, when $\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\| \leq \frac{B}{\eta}$, we have*

$$\sum_{j\in\mathcal{S}_1} g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}}) - \sum_{j\in\mathcal{S}_1} g_j(\widetilde{\mathbf{x}}_j^0) \leq -\sum_{j\in\mathcal{S}_1} \frac{3\theta}{8\eta} \sum_{k=0}^{\mathcal{K}-1} |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{8\eta\rho^2 B^4 \mathcal{K}}{\theta}. \tag{16}$$

Proof Since $g_j(x)$ is quadratic, we have

$$\begin{aligned}
g_j(\widetilde{\mathbf{x}}_j^{k+1}) =& g_j(\widetilde{\mathbf{x}}_j^k) + \left\langle \nabla g_j(\widetilde{\mathbf{x}}_j^k), \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k \right\rangle + \frac{\lambda_j}{2} |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 \\
\overset{a}{=}& g_j(\widetilde{\mathbf{x}}_j^k) - \frac{1}{\eta} \left\langle \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{y}}_j^k + \eta\widetilde{\delta}_j^k, \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k \right\rangle \\
&+ \left\langle \nabla g_j(\widetilde{\mathbf{x}}_j^k) - \nabla g_j(\widetilde{\mathbf{y}}_j^k), \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k \right\rangle + \frac{\lambda_j}{2} |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 \\
=& g_j(\widetilde{\mathbf{x}}_j^k) - \frac{1}{\eta} \left\langle \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{y}}_j^k, \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k \right\rangle - \left\langle \widetilde{\delta}_j^k, \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k \right\rangle \\
&+ \lambda_j \left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{y}}_j^k, \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k \right\rangle + \frac{\lambda_j}{2} |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 \\
=& g_j(\widetilde{\mathbf{x}}_j^k) + \frac{1}{2\eta} \left(|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{y}}_j^k|^2 - |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{y}}_j^k|^2 - |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 \right) \\
&- \left\langle \widetilde{\delta}_j^k, \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k \right\rangle + \frac{\lambda_j}{2} \left(|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{y}}_j^k|^2 - |\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{y}}_j^k|^2 \right) \\
\leq& g_j(\widetilde{\mathbf{x}}_j^k) + \frac{1}{2\eta} \left(|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{y}}_j^k|^2 - |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{y}}_j^k|^2 - |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 \right) \\
&+ \frac{1}{2\alpha} |\widetilde{\delta}_j^k|^2 + \frac{\alpha}{2} |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{\lambda_j}{2} \left(|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{y}}_j^k|^2 - |\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{y}}_j^k|^2 \right),
\end{aligned}$$

for some positive constant α to be specified later, where we use (14b) in $\overset{a}{=}$. Using $L \geq \lambda_j \geq -\frac{\theta}{\eta}$ when $j \in \mathcal{S}_1 = \{j : \lambda_j \geq -\frac{\theta}{\eta}\}$ and $\left(-\frac{1}{2\eta} + \frac{\lambda_j}{2}\right) |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{y}}_j^k|^2 \leq \left(-2L + \frac{L}{2}\right) |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{y}}_j^k|^2 \leq$

0, we have for each $j \in \mathcal{S}_1$,

$$\begin{aligned} g_j(\widetilde{\mathbf{x}}_j^{k+1}) \leq & g_j(\widetilde{\mathbf{x}}_j^k) + \frac{1}{2\eta}\left(|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{y}}_j^k|^2 - |\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2\right) + \frac{1}{2\alpha}|\widetilde{\delta}_j^k|^2 + \frac{\alpha}{2}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{\theta}{2\eta}|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{y}}_j^k|^2 \\ \overset{b}{=} & g_j(\widetilde{\mathbf{x}}_j^k) + \frac{(1+\theta)(1-\theta)^2}{2\eta}|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 - \left(\frac{1}{2\eta} - \frac{\alpha}{2}\right)|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{1}{2\alpha}|\widetilde{\delta}_j^k|^2, \end{aligned}$$

where we use (14a) in $\overset{b}{=}$. Defining the potential function

$$\ell_j^{k+1} = g_j(\widetilde{\mathbf{x}}_j^{k+1}) + \frac{(1+\theta)(1-\theta)^2}{2\eta}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2,$$

we have

$$\begin{aligned} \ell_j^{k+1} \leq & \ell_j^k - \left(\frac{1}{2\eta} - \frac{\alpha}{2} - \frac{(1+\theta)(1-\theta)^2}{2\eta}\right)|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{1}{2\alpha}|\widetilde{\delta}_j^k|^2 \\ \overset{c}{\leq} & \ell_j^k - \frac{3\theta}{8\eta}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{2\eta}{\theta}|\widetilde{\delta}_j^k|^2, \end{aligned}$$

where we let $\alpha = \frac{\theta}{4\eta}$ in $\overset{c}{\leq}$ such that $\frac{1}{2\eta} - \frac{\theta}{8\eta} - \frac{(1+\theta)(1-\theta)^2}{2\eta} = \frac{3\theta}{8\eta} + \frac{\theta^2}{2\eta} - \frac{\theta^3}{2\eta} \geq \frac{3\theta}{8\eta}$. Summing over $k = 0, 1, \cdots, \mathcal{K}-1$ and $j \in \mathcal{S}_1$, using $\mathbf{x}^0 - \mathbf{x}^{-1} = 0$, we have

$$\begin{aligned} \sum_{j\in\mathcal{S}_1} g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}}) \leq \sum_{j\in\mathcal{S}_1} \ell_j^{\mathcal{K}} \leq & \sum_{j\in\mathcal{S}_1} g_j(\widetilde{\mathbf{x}}_j^0) - \sum_{j\in\mathcal{S}_1}\frac{3\theta}{8\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{2\eta}{\theta}\sum_{k=0}^{\mathcal{K}-1}\|\widetilde{\delta}^k\|^2 \\ \overset{d}{\leq} & \sum_{j\in\mathcal{S}_1} g_j(\widetilde{\mathbf{x}}_j^0) - \sum_{j\in\mathcal{S}_1}\frac{3\theta}{8\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{8\eta\rho^2 B^4\mathcal{K}}{\theta}, \end{aligned}$$

where we use (15) in $\overset{d}{\leq}$. ■

Next, we consider $\sum_{j\in\mathcal{S}_2} g_j(\mathbf{x}_j)$.

Lemma 10 *Suppose that Assumption 1 holds. Let $\eta \leq \frac{1}{4L}$ and $0 < \theta \leq 1$. In each epoch of Algorithm 1 where the "if condition" triggers, when $\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\| \leq \frac{B}{\eta}$, we have*

$$\sum_{j\in\mathcal{S}_2} g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}}) - \sum_{j\in\mathcal{S}_2} g_j(\widetilde{\mathbf{x}}_j^0) \leq -\sum_{j\in\mathcal{S}_2}\frac{\theta}{2\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{2\eta\rho^2 B^4\mathcal{K}}{\theta}. \tag{17}$$

Proof Denoting $\mathbf{v}_j = \widetilde{\mathbf{x}}_j^0 - \frac{1}{\lambda_j}\widetilde{\nabla}_j f(\mathbf{x}^0)$, $g_j(x)$ can be rewritten as

$$\begin{aligned} g_j(x) = & \frac{\lambda_j}{2}\left(x - \widetilde{\mathbf{x}}_j^0 + \frac{1}{\lambda_j}\widetilde{\nabla}_j f(\mathbf{x}^0)\right)^2 - \frac{1}{2\lambda_j}|\widetilde{\nabla}_j f(\mathbf{x}^0)|^2 \\ = & \frac{\lambda_j}{2}|x - \mathbf{v}_j|^2 - \frac{1}{2\lambda_j}|\widetilde{\nabla}_j f(\mathbf{x}^0)|^2. \end{aligned}$$

For each $j \in \mathcal{S}_2 = \{j : \lambda_j < -\frac{\theta}{\eta}\}$, we have

$$\begin{aligned} g_j(\widetilde{\mathbf{x}}_j^{k+1}) - g_j(\widetilde{\mathbf{x}}_j^k) =& \frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^{k+1} - \mathbf{v}_j|^2 - \frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^k - \mathbf{v}_j|^2 \\ =& \frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \lambda_j \left\langle \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k, \widetilde{\mathbf{x}}_j^k - \mathbf{v}_j \right\rangle \\ \leq& -\frac{\theta}{2\eta}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \lambda_j \left\langle \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k, \widetilde{\mathbf{x}}_j^k - \mathbf{v}_j \right\rangle. \end{aligned} \tag{18}$$

So we only need to bound the second term. From (14b) and (14a), we have

$$\begin{aligned} \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k =& \widetilde{\mathbf{y}}_j^k - \widetilde{\mathbf{x}}_j^k - \eta \nabla g_j(\widetilde{\mathbf{y}}_j^k) - \eta\widetilde{\delta}_j^k \\ =& (1-\theta)(\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}) - \eta \nabla g_j(\widetilde{\mathbf{y}}_j^k) - \eta\widetilde{\delta}_j^k \\ =& (1-\theta)(\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}) - \eta\lambda_j(\widetilde{\mathbf{y}}_j^k - \mathbf{v}_j) - \eta\widetilde{\delta}_j^k \\ =& (1-\theta)(\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}) - \eta\lambda_j(\widetilde{\mathbf{x}}_j^k - \mathbf{v}_j + (1-\theta)(\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1})) - \eta\widetilde{\delta}_j^k. \end{aligned}$$

So for each $j \in \mathcal{S}_2$, we have

$$\begin{aligned} &\lambda_j \left\langle \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k, \widetilde{\mathbf{x}}_j^k - \mathbf{v}_j \right\rangle \\ =& (1-\theta)\lambda_j \left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \widetilde{\mathbf{x}}_j^k - \mathbf{v}_j \right\rangle - \eta\lambda_j^2|\widetilde{\mathbf{x}}_j^k - \mathbf{v}_j|^2 - \eta\lambda_j^2(1-\theta)\left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \widetilde{\mathbf{x}}_j^k - \mathbf{v}_j \right\rangle - \eta\lambda_j \left\langle \widetilde{\delta}_j^k, \widetilde{\mathbf{x}}_j^k - \mathbf{v}_j \right\rangle \\ \leq& (1-\theta)\lambda_j \left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \widetilde{\mathbf{x}}_j^k - \mathbf{v}_j \right\rangle - \eta\lambda_j^2|\widetilde{\mathbf{x}}_j^k - \mathbf{v}_j|^2 \\ &+ \frac{\eta\lambda_j^2(1-\theta)}{2}\left(|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 + |\widetilde{\mathbf{x}}_j^k - \mathbf{v}_j|^2\right) + \frac{\eta}{2(1+\theta)}|\widetilde{\delta}_j^k|^2 + \frac{\eta\lambda_j^2(1+\theta)}{2}|\widetilde{\mathbf{x}}_j^k - \mathbf{v}_j|^2 \\ =& (1-\theta)\lambda_j \left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \widetilde{\mathbf{x}}_j^k - \mathbf{v}_j \right\rangle + \frac{\eta\lambda_j^2(1-\theta)}{2}|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 + \frac{\eta}{2(1+\theta)}|\widetilde{\delta}_j^k|^2 \\ =& (1-\theta)\lambda_j \left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \widetilde{\mathbf{x}}_j^{k-1} - \mathbf{v}_j \right\rangle + (1-\theta)\lambda_j|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 + \frac{\eta\lambda_j^2(1-\theta)}{2}|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 + \frac{\eta}{2(1+\theta)}|\widetilde{\delta}_j^k|^2 \\ \overset{a}{\leq}& (1-\theta)\lambda_j \left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \widetilde{\mathbf{x}}_j^{k-1} - \mathbf{v}_j \right\rangle + \frac{\eta}{2}|\widetilde{\delta}_j^k|^2, \end{aligned}$$

where we use $\left(1 + \frac{\eta\lambda_j}{2}\right)(1-\theta) \geq \left(1 - \frac{\eta L}{2}\right)(1-\theta) \geq 0$ and $\lambda_j < 0$ when $j \in \mathcal{S}_2$ in $\overset{a}{\geq}$. So we have

$$\begin{aligned} \lambda_j \left\langle \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k, \widetilde{\mathbf{x}}_j^k - \mathbf{v}_j \right\rangle \leq& (1-\theta)^k \lambda_j \left\langle \widetilde{\mathbf{x}}_j^1 - \widetilde{\mathbf{x}}_j^0, \widetilde{\mathbf{x}}_j^0 - \mathbf{v}_j \right\rangle + \frac{\eta}{2}\sum_{t=1}^k (1-\theta)^{k-t}|\widetilde{\delta}_j^t|^2 \\ \overset{b}{=}& -(1-\theta)^k\eta\lambda_j^2|\widetilde{\mathbf{x}}_j^0 - \mathbf{v}_j|^2 + \frac{\eta}{2}\sum_{t=1}^k (1-\theta)^{k-t}|\widetilde{\delta}_j^t|^2 \\ \leq& \frac{\eta}{2}\sum_{t=1}^k (1-\theta)^{k-t}|\widetilde{\delta}_j^t|^2, \end{aligned}$$

where we use

$$\begin{aligned}\widetilde{\mathbf{x}}_j^1 - \widetilde{\mathbf{x}}_j^0 =& \widetilde{\mathbf{x}}_j^1 - \widetilde{\mathbf{y}}_j^0 = -\eta\widetilde{\nabla}_j f(\mathbf{y}^0) = -\eta\widetilde{\nabla}_j f(\mathbf{x}^0)\\ =& -\eta\nabla g_j(\widetilde{\mathbf{x}}_j^0) = -\eta\lambda_j(\widetilde{\mathbf{x}}_j^0 - \mathbf{v}_j)\end{aligned}$$

in $\overset{b}{=}$. Plugging into (18), we have

$$g_j(\widetilde{\mathbf{x}}_j^{k+1}) - g_j(\widetilde{\mathbf{x}}_j^k) \le -\frac{\theta}{2\eta}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{\eta}{2}\sum_{t=1}^{k}(1-\theta)^{k-t}|\widetilde{\delta}_j^t|^2.$$

Summing over $k = 0, 1, \cdots, \mathcal{K}-1$ and $j \in \mathcal{S}_2$, we have

$$\begin{aligned}\sum_{j\in\mathcal{S}_2} g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}}) - \sum_{j\in\mathcal{S}_2} g_j(\widetilde{\mathbf{x}}_j^0) \le& -\sum_{j\in\mathcal{S}_2}\frac{\theta}{2\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{\eta}{2}\sum_{k=0}^{\mathcal{K}-1}\sum_{t=1}^{k}(1-\theta)^{k-t}\|\widetilde{\delta}^t\|^2\\ \overset{c}{\le}& -\sum_{j\in\mathcal{S}_2}\frac{\theta}{2\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + 2\eta\rho^2B^4\sum_{k=0}^{\mathcal{K}-1}\sum_{t=1}^{k}(1-\theta)^{k-t}\\ \le& -\sum_{j\in\mathcal{S}_2}\frac{\theta}{2\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{2\eta\rho^2B^4\mathcal{K}}{\theta},\end{aligned}$$

where we use (15) in $\overset{c}{\le}$. ■

Putting Lemmas 9 and 10 together, we have the following lemma.

Lemma 11 *Suppose that Assumption 1 holds. Let $\eta \le \frac{1}{4L}$ and $0 < \theta \le 1$. In each epoch of Algorithm 1 where the "if condition" triggers, when $\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\| \le \frac{B}{\eta}$, we have*

$$f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \le -\frac{3\theta B^2}{8\eta K} + \frac{10\eta\rho^2B^4K}{\theta} + 4.5\rho B^3. \tag{19}$$

Proof Summing over (16) and (17), we have

$$\begin{aligned}g(\widetilde{\mathbf{x}}^{\mathcal{K}}) - g(\widetilde{\mathbf{x}}^0) =& \sum_{j\in\mathcal{S}_1\cup\mathcal{S}_2} g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}}) - g_j(\widetilde{\mathbf{x}}_j^0)\\ \le& -\frac{3\theta}{8\eta}\sum_{k=0}^{\mathcal{K}-1}\|\widetilde{\mathbf{x}}^{k+1} - \widetilde{\mathbf{x}}^k\|^2 + \frac{10\eta\rho^2B^4\mathcal{K}}{\theta}\\ =& -\frac{3\theta}{8\eta}\sum_{k=0}^{\mathcal{K}-1}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 + \frac{10\eta\rho^2B^4\mathcal{K}}{\theta}\\ \overset{a}{\le}& -\frac{3\theta B^2}{8\eta\mathcal{K}} + \frac{10\eta\rho^2B^4\mathcal{K}}{\theta},\end{aligned}$$

where we use (8a) in $\overset{a}{\le}$. Plugging into (12) and using $\mathcal{K} \le K$, we have

$$\begin{aligned}f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \le& -\frac{3\theta B^2}{8\eta\mathcal{K}} + \frac{10\eta\rho^2B^4\mathcal{K}}{\theta} + 4.5\rho B^3\\ \le& -\frac{3\theta B^2}{8\eta K} + \frac{10\eta\rho^2B^4K}{\theta} + 4.5\rho B^3.\end{aligned}$$

■

From Lemmas 8 and 11, we can establish the decrease of $f(\mathbf{x})$ in each epoch.

Corollary 12 *Suppose that Assumption 1 holds. Use the parameter settings in Theorem 1. In each epoch of Algorithm 1 where the "if condition" triggers, we have*

$$f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \leq -\frac{7\epsilon^{3/2}}{8\sqrt{\rho}}. \tag{20}$$

Proof Combing Lemmas 8 and 11 and using the parameter settings, we have

$$f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \leq -\min\left\{\frac{3\theta B^2}{8\eta K} - \frac{10\eta\rho^2 B^4 K}{\theta} - 4.5\rho B^3, \frac{B^2}{4\eta}\right\} = -\min\left\{\frac{7\epsilon^{3/2}}{8\sqrt{\rho}}, \frac{\epsilon}{4\eta\rho}\right\}.$$

From $\theta = 4(\epsilon\rho\eta^2)^{1/4} \leq 1$, we have $\frac{7\epsilon^{3/2}}{8\sqrt{\rho}} \leq \frac{\epsilon}{4\eta\rho}$. ■

4.1.3 Small Gradient in the Last Epoch

We first give the following lemma for the last epoch.

Lemma 13 *Suppose that Assumption 1 holds. Use the parameter settings in Theorem 1. In the last epoch of Algorithm 1 where the "if condition" does not trigger, we have* $\|\nabla f(\hat{\mathbf{y}})\| \leq 82\epsilon$.

Proof Denote $\widetilde{\mathbf{y}} = \mathbf{U}^T\hat{\mathbf{y}} = \frac{1}{K_0+1}\sum_{k=0}^{K_0}\mathbf{U}^T\mathbf{y}^k = \frac{1}{K_0+1}\sum_{k=0}^{K_0}\widetilde{\mathbf{y}}^k$. Since g is quadratic, we have

$$\begin{aligned}
\|\nabla g(\widetilde{\mathbf{y}})\| =& \left\|\frac{1}{K_0+1}\sum_{k=0}^{K_0}\nabla g(\widetilde{\mathbf{y}}^k)\right\| \\
\overset{a}{=}& \frac{1}{\eta(K_0+1)}\left\|\sum_{k=0}^{K_0}\left(\widetilde{\mathbf{x}}^{k+1} - \widetilde{\mathbf{y}}^k + \eta\widetilde{\delta}^k\right)\right\| \\
=& \frac{1}{\eta(K_0+1)}\left\|\sum_{k=0}^{K_0}\left(\widetilde{\mathbf{x}}^{k+1} - \widetilde{\mathbf{x}}^k - (1-\theta)(\widetilde{\mathbf{x}}^k - \widetilde{\mathbf{x}}^{k-1}) + \eta\widetilde{\delta}^k\right)\right\| \\
\overset{b}{=}& \frac{1}{\eta(K_0+1)}\left\|\widetilde{\mathbf{x}}^{K_0+1} - \widetilde{\mathbf{x}}^0 - (1-\theta)(\widetilde{\mathbf{x}}^{K_0} - \widetilde{\mathbf{x}}^0) + \eta\sum_{k=0}^{K_0}\widetilde{\delta}^k\right\| \\
=& \frac{1}{\eta(K_0+1)}\left\|\widetilde{\mathbf{x}}^{K_0+1} - \widetilde{\mathbf{x}}^{K_0} + \theta(\widetilde{\mathbf{x}}^{K_0} - \widetilde{\mathbf{x}}^0) + \eta\sum_{k=0}^{K_0}\widetilde{\delta}^k\right\| \\
\leq& \frac{1}{\eta(K_0+1)}\left(\|\widetilde{\mathbf{x}}^{K_0+1} - \widetilde{\mathbf{x}}^{K_0}\| + \theta\|\widetilde{\mathbf{x}}^{K_0} - \widetilde{\mathbf{x}}^0\| + \eta\sum_{k=0}^{K_0}\|\widetilde{\delta}^k\|\right) \\
\overset{c}{\leq}& \frac{2}{\eta K}\|\widetilde{\mathbf{x}}^{K_0+1} - \widetilde{\mathbf{x}}^{K_0}\| + \frac{2\theta B}{\eta K} + 2\rho B^2,
\end{aligned} \tag{21}$$

where we use (14b) in $\overset{a}{=}$, $\mathbf{x}^{-1}=\mathbf{x}^0$ in $\overset{b}{=}$, $K_0+1\geq\frac{K}{2}$, (10a), (15), and (10b) in $\overset{c}{\leq}$. From $K_0=\operatorname{argmin}_{\lfloor\frac{K}{2}\rfloor\leq k\leq K-1}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|$, we have

$$\begin{aligned}\|\mathbf{x}^{K_0+1}-\mathbf{x}^{K_0}\|^2\leq&\frac{1}{K-\lfloor K/2\rfloor}\sum_{k=\lfloor K/2\rfloor}^{K-1}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2\\ \leq&\frac{1}{K-\lfloor K/2\rfloor}\sum_{k=0}^{K-1}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2\\ \overset{d}{\leq}&\frac{1}{K-\lfloor K/2\rfloor}\frac{B^2}{K}\leq\frac{2B^2}{K^2},\end{aligned}\tag{22}$$

where we use (10a) in $\overset{d}{\leq}$. On the other hand, we also have

$$\begin{aligned}\|\nabla f(\hat{\mathbf{y}})\|=\|\widetilde{\nabla}f(\hat{\mathbf{y}})\|\leq&\|\nabla g(\widetilde{\mathbf{y}})\|+\|\widetilde{\nabla}f(\hat{\mathbf{y}})-\nabla g(\widetilde{\mathbf{y}})\|\\ =&\|\nabla g(\widetilde{\mathbf{y}})\|+\|\widetilde{\nabla}f(\hat{\mathbf{y}})-\widetilde{\nabla}f(\mathbf{x}^0)-\mathbf{\Lambda}(\widetilde{\mathbf{y}}-\widetilde{\mathbf{x}}^0)\|\\ =&\|\nabla g(\widetilde{\mathbf{y}})\|+\|\nabla f(\hat{\mathbf{y}})-\nabla f(\mathbf{x}^0)-\mathbf{H}(\hat{\mathbf{y}}-\mathbf{x}^0)\|\\ \leq&\|\nabla g(\widetilde{\mathbf{y}})\|+\frac{\rho}{2}\|\hat{\mathbf{y}}-\mathbf{x}^0\|^2\overset{e}{\leq}\|\nabla g(\widetilde{\mathbf{y}})\|+2\rho B^2,\end{aligned}$$

where we use $\|\hat{\mathbf{y}}-\mathbf{x}^0\|\leq\frac{1}{K_0+1}\sum_{k=0}^{K_0}\|\mathbf{y}^k-\mathbf{x}^0\|\leq 2B$ from (10b) in $\overset{e}{\leq}$. So we have

$$\|\nabla f(\hat{\mathbf{y}})\|\leq\frac{2\sqrt{2}B}{\eta K^2}+\frac{2\theta B}{\eta K}+4\rho B^2\leq 82\epsilon.$$

■

Based Corollary 12 and Lemma 13, we can prove Theorem 1.

Proof For each epoch where the "if condition" triggers, we have (20). Note that at the beginning of each epoch, we set $\mathbf{x}^0$ to be the last iterate $\mathbf{x}^{\mathcal{K}}$ in the previous epoch. Summing (20) over all epochs, say N total epochs, and using $\min_{\mathbf{x}}f(\mathbf{x})\leq f(\mathbf{x}^{\mathcal{K}})$, we have

$$\min_{\mathbf{x}}f(\mathbf{x})-f(\mathbf{x}_{int})\leq-N\frac{7\epsilon^{3/2}}{8\sqrt{\rho}}.$$

So the algorithm will terminate (that is, the "if condition" does not trigger and the while loop breaks) in at most $\frac{8\triangle_f\sqrt{\rho}}{7\epsilon^{3/2}}$ epochs. Since each epoch needs at most $K=\frac{1}{2}\left(\frac{L^2}{\epsilon\rho}\right)^{1/4}$ gradient evaluations, the total number of gradient evaluations must be less than $\frac{\triangle_f L^{1/2}\rho^{1/4}}{\epsilon^{7/4}}$. On the other hand, in the last epoch, we have $\|\nabla f(\hat{\mathbf{y}})\|\leq 82\epsilon$ from Lemma 13. ■

Remark 14 *The purpose of using the specific average as the output and $k\sum_{t=0}^{k-1}\|\mathbf{x}^{t+1}-\mathbf{x}^t\|^2>B^2$ in the "if condition" in Algorithm 1, rather than $\|\mathbf{x}^k-\mathbf{x}^0\|\geq B$, is to establish (22).*

4.2 Discussion on the Acceleration Mechanism

When we replace the AGD iterations in Algorithm 1 by the gradient descent steps $\mathbf{x}^{k+1} = \mathbf{x}^k - \eta\nabla f(\mathbf{x}^k)$ with step-size $\eta = \frac{1}{4L}$, similar to (11), the descent property in each epoch becomes

$$f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \le -\frac{7}{8\eta}\sum_{k=0}^{\mathcal{K}-1}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2 \le -\frac{7B^2}{8\eta\mathcal{K}},$$

and the gradient norm at the averaged output $\hat{\mathbf{x}} = \frac{1}{K}\sum_{k=0}^{K-1}\mathbf{x}^k$ can be bounded as

$$\|\nabla g(\hat{\mathbf{x}})\| \le \frac{1}{\eta K}\|\mathbf{x}^K - \mathbf{x}^0\| + 2\rho B^2 \le \frac{B}{\eta K} + 2\rho B^2.$$

By setting $B = \sqrt{\frac{\epsilon}{\rho}}$ and $K = \frac{L}{\sqrt{\epsilon\rho}}$, we have the $\mathcal{O}(\epsilon^{-2})$ total complexity.

Comparing the above two inequalities with (19) and (21), respectively, we see that the momentum parameter θ is crucial to speedup the convergence of AGD because it allows smaller K than that of GD, that is, $\frac{1}{\epsilon^{1/4}}$ v.s. $\frac{1}{\epsilon^{1/2}}$ for AGD and GD, respectively. Accordingly, smaller K results in less total gradient evaluations since both methods need $\mathcal{O}(\epsilon^{-1.5})$ epochs. The above comparisons show the importance of momentum and its parameter θ in the nonconvex acceleration mechanism. On the other hand, since our proofs do not invoke the analysis of strongly convex AGD, we conjecture that the nonconvex acceleration mechanism seems irrelevant to the analysis of convex AGD.

4.3 Proof of Theorem 3

In this section, we prove a stronger theorem, where we replace lines 11 and 8 of Algorithm 2 by (4) and $f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \le -\gamma\frac{\epsilon^{3/2}}{\sqrt{\rho'}}$, respectively. Denote η_{int}, ρ'_{int}, and $B_{0,int}$ to be the initializations of η, ρ', and B_0, respectively.

Theorem 15 *Suppose that Assumption 1 holds. Let $B = \sqrt{\frac{\epsilon}{\rho'}}$, $\theta = 4(\epsilon\rho'\eta^2)^{1/4} \in (0,1)$, and $K = \lfloor\frac{1}{\theta}\rfloor$ in each epoch, where η and ρ' may change dynamically during epochs. Let $\gamma \le \frac{7}{8}$, $c_0 > 1$, $c_1 \ge c_2 > 1$, $\eta_{min} \le \frac{1}{4L}$, and $\rho'_{max} \ge \rho$, where $\frac{\rho'_{max}}{\rho'_{int}} = \left(\frac{\eta_{int}}{\eta_{min}}\right)^2$. Then Algorithm 2 with (4) terminates in at most $\mathcal{O}(\epsilon^{-7/4})$ gradient computations and $\mathcal{O}(\epsilon^{-3/2})$ function evaluations, and the output satisfies $\|\nabla f(\mathbf{x}_{out})\| \le \mathcal{O}(\epsilon)$.*

Using the same proofs of Corollary 12 and Lemma 13, we have the following two straight-forward corollaries.

Corollary 16 *Suppose that Assumption 1 holds. Let $B = \sqrt{\frac{\epsilon}{\rho'}}$, $\theta = 4\left(\epsilon\rho'\eta^2\right)^{1/4} \in (0,1]$, and $K = \lfloor\frac{1}{\theta}\rfloor$ in one epoch, where $\eta \le \frac{1}{4L}$ and $\rho' \ge \rho$. Assume that $\mathcal{K}\sum_{t=0}^{\mathcal{K}-1}\|\mathbf{x}^{t+1}-\mathbf{x}^t\|^2 > B^2$ for some $\mathcal{K} \le K$ and $k\sum_{t=0}^{k-1}\|\mathbf{x}^{t+1}-\mathbf{x}^t\|^2 \le B^2$ for all $k < \mathcal{K}$, then for the iterations*

$$\mathbf{y}^k = \mathbf{x}^k + (1-\theta)(\mathbf{x}^k - \mathbf{x}^{k-1}), \quad \mathbf{x}^{k+1} = \mathbf{y}^k - \eta\nabla f(\mathbf{y}^k) \tag{23}$$

starting from $\mathbf{x}^0 = \mathbf{x}^{-1}$, we have

$$f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \le -\frac{7\epsilon^{3/2}}{8\sqrt{\rho'}}. \tag{24}$$

Corollary 17 *Suppose that Assumption 1 holds. Let* $B = \sqrt{\frac{\epsilon}{\rho'}}$, $\theta = 4\left(\epsilon\rho'\eta^2\right)^{1/4} \in (0,1)$, *and* $K = \lfloor \frac{1}{\theta} \rfloor$ *in one epoch. Assume that* $k\sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \le B^2$ *for all* $k \le K$, *then for the iterations (23), we have*

$$\|\nabla f(\hat{\mathbf{y}})\| \le \frac{2\sqrt{2}B}{\eta K^2} + \frac{2\theta B}{\eta K} + 4\rho B^2 \le \frac{78\epsilon}{(1-\theta)^2} + \frac{4\rho\epsilon}{\rho'},$$

where $\hat{\mathbf{y}}$ *is defined on lines 15 and 16 of Algorithm 2.*

Now, we can prove Theorem 15.

Proof Recall that we define one round of AGD to be one epoch and the parameters η, ρ', B_0, and B do not change during each epoch. From the update of η and ρ', we know θ and K never change. That is, $\theta = 4\left(\epsilon\rho'_{int}\eta_{int}^2\right)^{1/4} = 4\left(\epsilon\rho'_{max}\eta_{min}^2\right)^{1/4}$ and $K = \lfloor \frac{1}{\theta} \rfloor$ all the time.

We first consider the last epoch if the algorithm terminates. From line 2 of Algorithm 2, we know the while loop breaks when $k \ge K$ and $B_0 \le B$. In the last epoch where the "if condition" on line 6 does not trigger, we have $k\sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \le \max\{B^2, B_0^2\}$ and $k \le K$. So the last epoch consists of K iterations and $k\sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \le B^2$ for all $k \le K$. From Corollary 17, we have $\|\nabla f(\hat{\mathbf{y}})\| \le \frac{78\epsilon}{(1-\theta)^2} + \frac{4\rho\epsilon}{\rho'_{int}} = \mathcal{O}(\epsilon)$.

Next, we prove the algorithm will terminate in at most $\mathcal{O}(\epsilon^{-3/2})$ epochs. In each epoch where the "if condition" on line 6 triggers, we execute either line 9 or line 11 (in fact, step (4)), depending on the condition on line 8. Denote one epoch to be valid when the "if condition" on line 8 holds. Otherwise, denote this epoch to be invalid, where invalid means that we discard the whole iterates in this epoch and reset $\mathbf{x}^0$ and $\mathbf{x}^{-1}$ to be the last iterate in the previous valid epoch, which is stored in $\mathbf{x}^0_{cur}$.

Consider the total number of invalid epochs. We can prove that invalid epoch never appears and line 11 never executes when $B_0 \le B$, $\eta \le \frac{1}{4L}$, and $\rho' \ge \rho$. In fact, for each epoch except the last one, when $B_0 \le B$, we always have $k < K$. Otherwise, the while loop on line 2 breaks. Thus, when the "if condition " on line 6 triggers, we must have $\mathcal{K}\sum_{t=0}^{\mathcal{K}-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 > B^2$ for some $\mathcal{K} \le K$ and $k\sum_{t=0}^{k-1} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \le B^2$ for all $k < \mathcal{K}$. From Corollary 16, we have $f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \le -\frac{7\epsilon^{3/2}}{8\sqrt{\rho'}} \le -\gamma\frac{\epsilon^{3/2}}{\sqrt{\rho'}}$, which triggers the condition on line 8. Thus, line 11 never executes. So we only need to count the number of epochs such that $B_0 \le B$, $\eta \le \frac{1}{4L}$, and $\rho' \ge \rho$. Letting

$$\frac{B_{0,int}}{(c_0c_1)^{N_{iv}}} \le \sqrt{\frac{\epsilon}{\rho'_{int}c_2^{2N_{iv}}}}, \quad \frac{\eta_{int}}{c_2^{N_{iv}}} \le \frac{1}{4L}, \quad \rho'_{int}c_2^{2N_{iv}} \ge \rho, \tag{25}$$

we have $N_{iv} = \mathcal{O}\left(\log_{c_0c_1/c_2}\frac{B_{0,int}\rho'_{int}}{\epsilon} + \log_{c_2}(L\eta_{int}) + \log_{c_2}\frac{\rho}{\rho'_{int}}\right)$. So we only need $\mathcal{O}(\log\frac{C}{\epsilon})$ invalid epochs for some constant C to get $B_0 \le B$, $\eta \le \frac{1}{4L}$, and $\rho' \ge \rho$.

Consider the valid epochs. Since each valid epoch decreases the objective value at least $\gamma\frac{\epsilon^{3/2}}{\sqrt{\rho'}} \ge \gamma\frac{\epsilon^{3/2}}{\sqrt{\rho'_{max}}}$, we have at most $\frac{\triangle_f\sqrt{\rho'_{max}}}{\gamma\epsilon^{3/2}}$ valid epochs.

Putting the two cases together, we need at most $\mathcal{O}\left(\frac{\triangle_f\sqrt{\rho'_{max}}}{\epsilon^{3/2}} + \log\frac{C}{\epsilon}\right)$ epochs, and accordingly, $\mathcal{O}\left(\frac{\triangle_f\sqrt{\rho'_{max}}}{\epsilon^{3/2}} + \log\frac{C}{\epsilon}\right)$ function evaluations. On the other hand, each epoch,

no matter valid or not, needs at most $K+1=\mathcal{O}\left(\frac{1}{(\epsilon\rho'\eta^2)^{1/4}}\right)=\mathcal{O}\left(\frac{1}{(\epsilon\rho'_{max}\eta^2_{min})^{1/4}}\right)$ gradient computations (see line 6 of Algorithm 2). So the total number of gradient computations is $\mathcal{O}\left(\left(\frac{\triangle_f\sqrt{\rho'_{max}}}{\epsilon^{3/2}}+\log\frac{C}{\epsilon}\right)\left(\frac{1}{(\epsilon\rho'_{max}\eta^2_{min})^{1/4}}\right)\right)=\mathcal{O}\left(\frac{\triangle_f(\rho'_{max})^{1/4}}{\epsilon^{7/4}\eta_{min}^{1/2}}+\frac{1}{(\epsilon\rho'_{max}\eta^2_{min})^{1/4}}\log\frac{C}{\epsilon}\right)$. ■

At last, we consider Theorem 3. In the original Algorithm 2 where we do not dynamically change ρ' and η, we only need to replace (25) by $\frac{B_{0,int}}{c_0^{N_{iv}}}\leq\sqrt{\frac{\epsilon}{\rho}}$ such that $N_{iv}=\mathcal{O}\left(\log_{c_0}\frac{\rho B_{0,int}}{\epsilon}\right)$, even if line 11 in Algorithm 2 is never triggered. Since ρ' and η are fixed at ρ and $\frac{1}{4L}$, respectively, we have the complexity of $\mathcal{O}\left(\frac{\triangle_f\sqrt{\rho}}{\epsilon^{3/2}}+\log\frac{\rho B_{0,int}}{\epsilon}\right)$ function evaluations and $\mathcal{O}\left(\frac{\triangle_f L^{1/2}\rho^{1/4}}{\epsilon^{7/4}}+\frac{L^{1/2}}{\epsilon^{1/4}\rho^{1/4}}\log\frac{\rho B_{0,int}}{\epsilon}\right)$ gradient computations.

4.4 Proof of Theorem 6

We follow the proof sketch in Section 4.1 and use the notations therein. Specifically, (8a), (8b), and (10a) also hold for the heavy ball method.

4.4.1 Large Gradient of $\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\|$

Similar to Lemma 8, we first consider the case when $\|\nabla f(\mathbf{y}^{\mathcal{K}-1})\|$ is large and give the following lemma for Algorithm 3.

Lemma 18 *Suppose that Assumption 1 holds. Let $\eta\leq\frac{1}{4L}$ and $0\leq\theta\leq\frac{1}{10}$. In each epoch of Algorithm 3 where the "if condition" triggers, when $\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\|>\frac{4B}{\eta}$, we have*

$$f(\mathbf{z}^{\mathcal{K}})-f(\mathbf{x}^0)\leq-\frac{3B^2}{16\eta}.$$

Proof As the gradient is L-Lipschitz, we have

$$\begin{aligned}
&f(\mathbf{x}^{k+1})\\
\leq&f(\mathbf{x}^k)+\left\langle\nabla f(\mathbf{x}^k),\mathbf{x}^{k+1}-\mathbf{x}^k\right\rangle+\frac{L}{2}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2\\
\overset{a}{=}&f(\mathbf{x}^k)+\left\langle\nabla f(\mathbf{x}^k),(1-\theta)(\mathbf{x}^k-\mathbf{x}^{k-1})-\eta\nabla f(\mathbf{x}^k)\right\rangle+\frac{L}{2}\|(1-\theta)(\mathbf{x}^k-\mathbf{x}^{k-1})-\eta\nabla f(\mathbf{x}^k)\|^2\\
\leq&f(\mathbf{x}^k)-\eta\|\nabla f(\mathbf{x}^k)\|^2+\frac{\eta}{2}\|\nabla f(\mathbf{x}^k)\|^2+\frac{1}{2\eta}\|\mathbf{x}^k-\mathbf{x}^{k-1}\|^2+L\|\mathbf{x}^k-\mathbf{x}^{k-1}\|^2+L\eta^2\|\nabla f(\mathbf{x}^k)\|^2\\
\overset{b}{\leq}&f(\mathbf{x}^k)-\frac{\eta}{4}\|\nabla f(\mathbf{x}^k)\|^2+\frac{3}{4\eta}\|\mathbf{x}^k-\mathbf{x}^{k-1}\|^2,
\end{aligned}$$

where we use the heavy ball iteration on line 3 of Algorithm 3 in $\overset{a}{=}$ and $\eta\leq\frac{1}{4L}$ in $\overset{b}{\leq}$. Summing over $k=0,\cdots,\mathcal{K}-1$ and using $\mathbf{x}^0=\mathbf{x}^{-1}$, we have

$$f(\mathbf{x}^{\mathcal{K}})-f(\mathbf{x}^0)\leq\frac{3}{4\eta}\sum_{k=0}^{\mathcal{K}-2}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2-\frac{\eta}{4}\sum_{k=0}^{\mathcal{K}-1}\|\nabla f(\mathbf{x}^k)\|^2\overset{c}{\leq}\frac{3B^2}{4\eta}-\frac{\eta}{4}\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\|^2,\quad(26)$$

where we use (8b) in $\overset{c}{\leq}$. On the other hand, we also have

$$f(\mathbf{x}^{\mathcal{K}-1}) - f(\mathbf{x}^0) \leq \frac{3}{4\eta}\sum_{k=0}^{\mathcal{K}-3}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2 - \frac{\eta}{4}\sum_{k=0}^{\mathcal{K}-2}\|\nabla f(\mathbf{x}^k)\|^2 \leq \frac{3B^2}{4\eta}. \tag{27}$$

Define $h(\mathbf{x}) = f(\mathbf{x}) + \frac{L}{2}\|\mathbf{x}-\mathbf{x}^0\|^2$. We know $h(\mathbf{x})$ is convex since $\nabla^2 f(\mathbf{x}) \succeq -L\mathbf{I}$. Thus we have $h(\mathbf{z}^{\mathcal{K}}) \leq \alpha h(\mathbf{x}^{\mathcal{K}}) + (1-\alpha)h(\mathbf{x}^{\mathcal{K}-1})$ with $\alpha = \frac{1}{1+(1-2\theta)(1-\theta)} \in [\frac{1}{2}, \frac{1}{1.72}]$ and $\mathbf{z}^{\mathcal{K}} = \alpha\mathbf{x}^{\mathcal{K}} + (1-\alpha)\mathbf{x}^{\mathcal{K}-1}$, which further yields

$$\begin{aligned}
f(\mathbf{z}^{\mathcal{K}}) \leq& \alpha f(\mathbf{x}^{\mathcal{K}}) + (1-\alpha)f(\mathbf{x}^{\mathcal{K}-1}) + \frac{L\alpha}{2}\|\mathbf{x}^{\mathcal{K}}-\mathbf{x}^0\|^2 + \frac{L(1-\alpha)}{2}\|\mathbf{x}^{\mathcal{K}-1}-\mathbf{x}^0\|^2 \\
&- \frac{L}{2}\|\alpha(\mathbf{x}^{\mathcal{K}}-\mathbf{x}^0) + (1-\alpha)(\mathbf{x}^{\mathcal{K}-1}-\mathbf{x}^0)\|^2 \\
\overset{d}{=}& \alpha f(\mathbf{x}^{\mathcal{K}}) + (1-\alpha)f(\mathbf{x}^{\mathcal{K}-1}) + \frac{L\alpha(1-\alpha)}{2}\|\mathbf{x}^{\mathcal{K}}-\mathbf{x}^{\mathcal{K}-1}\|^2 \\
\leq& \alpha f(\mathbf{x}^{\mathcal{K}}) + (1-\alpha)f(\mathbf{x}^{\mathcal{K}-1}) + \frac{1}{32\eta}\|\mathbf{x}^{\mathcal{K}}-\mathbf{x}^{\mathcal{K}-1}\|^2 \\
\leq& \alpha f(\mathbf{x}^{\mathcal{K}}) + (1-\alpha)f(\mathbf{x}^{\mathcal{K}-1}) + \frac{1}{16\eta}\|\mathbf{x}^{\mathcal{K}-1}-\mathbf{x}^{\mathcal{K}-2}\|^2 + \frac{\eta}{16}\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\|^2,
\end{aligned} \tag{28}$$

where we use $|\alpha x + (1-\alpha)y|^2 = \alpha x^2 + (1-\alpha)y^2 - \alpha(1-\alpha)|x-y|^2$ in $\overset{d}{=}$. Plugging (26) and (27) into (28) and using $\|\mathbf{x}^{\mathcal{K}-1}-\mathbf{x}^{\mathcal{K}-2}\|^2 \leq B^2$, we have

$$\begin{aligned}
f(\mathbf{z}^{\mathcal{K}}) - f(\mathbf{x}^0) \leq& \frac{3B^2}{4\eta} + \frac{B^2}{16\eta} - \frac{\eta\alpha}{4}\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\|^2 + \frac{\eta}{16}\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\|^2 \\
\leq& \frac{13B^2}{16\eta} - \frac{\eta}{16}\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\|^2 \overset{e}{\leq} -\frac{3B^2}{16\eta},
\end{aligned}$$

where we use $\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\| > \frac{4B}{\eta}$ in $\overset{e}{\leq}$. ■

4.4.2 Small Gradient of $\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\|$

If $\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\| \leq \frac{4B}{\eta}$, then from the heavy ball iteration on line 3 of Algorithm 3 and (8b) we have

$$\|\mathbf{x}^{\mathcal{K}}-\mathbf{x}^0\| \leq \|\mathbf{x}^{\mathcal{K}-1}-\mathbf{x}^0\| + \eta\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\| + (1-\theta)\|\mathbf{x}^{\mathcal{K}-1}-\mathbf{x}^{\mathcal{K}-2}\| \leq 6B.$$

Similar to (12), using the definition of $g(\mathbf{x})$ in (13), we have

$$f(\mathbf{x}^{\mathcal{K}}) - f(\mathbf{x}^0) \leq g(\widetilde{\mathbf{x}}^{\mathcal{K}}) - g(\widetilde{\mathbf{x}}^0) + 36\rho B^3. \tag{29}$$

Denoting

$$\widetilde{\delta}_j^k = \widetilde{\nabla}_j f(\mathbf{x}^k) - \nabla g_j(\widetilde{\mathbf{x}}_j^k), \qquad \widetilde{\delta}^k = \widetilde{\nabla} f(\mathbf{x}^k) - \nabla g(\widetilde{\mathbf{x}}^k),$$

then the heavy ball iteration in Algorithm 3 can be rewritten as

$$\begin{aligned}\widetilde{\mathbf{x}}_j^{k+1} =&\widetilde{\mathbf{x}}_j^k - \eta\widetilde{\nabla}_j f(\mathbf{x}^k) + (1-\theta)(\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1})\\ =&\widetilde{\mathbf{x}}_j^k - \eta\nabla g_j(\widetilde{\mathbf{x}}_j^k) - \eta\widetilde{\delta}_j^k + (1-\theta)(\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}).\end{aligned} \tag{30}$$

Similar to (15), $\|\widetilde{\delta}^k\|$ can also be bounded as

$$\|\widetilde{\delta}^k\| \le \frac{\rho}{2}\|\mathbf{x}^k - \mathbf{x}^0\|^2 \le \frac{\rho B^2}{2} \tag{31}$$

for any $k < \mathcal{K}$.

Lemma 19 *Suppose that Assumption 1 holds. Let $\eta \le \frac{1}{4L}$ and $0 \le \theta \le \frac{1}{10}$. In each epoch of Algorithm 3 where the "if condition" triggers, when $\|\nabla f(\mathbf{x}^{\mathcal{K}-1})\| \le \frac{4B}{\eta}$, we have*

$$f(\mathbf{z}^{\mathcal{K}}) - f(\mathbf{x}^0) \le -\frac{9\theta B^2}{40\eta\mathcal{K}} + \frac{\eta\rho^2 B^4\mathcal{K}}{4\theta} + 36\rho B^3.$$

Proof Since $g_j(x)$ is quadratic, we have

$$\begin{aligned}g_j(\widetilde{\mathbf{x}}_j^{k+1}) - g_j(\widetilde{\mathbf{x}}_j^k) =&\frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^0|^2 - \frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^0|^2 + \left\langle \widetilde{\nabla}_j f(\mathbf{x}^0), \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k\right\rangle\\ =&\frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \left\langle \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k, \lambda_j(\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^0) + \widetilde{\nabla}_j f(\mathbf{x}^0)\right\rangle\\ =&\frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \left\langle \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k, \nabla g_j(\widetilde{\mathbf{x}}_j^k)\right\rangle.\end{aligned} \tag{32}$$

From (30), we have

$$\begin{aligned}&\left\langle \widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k, \nabla g_j(\widetilde{\mathbf{x}}_j^k)\right\rangle\\ =&(1-\theta)\left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \nabla g_j(\widetilde{\mathbf{x}}_j^k)\right\rangle - \eta|\nabla g_j(\widetilde{\mathbf{x}}_j^k)|^2 - \eta\left\langle \widetilde{\delta}_j^k, \nabla g_j(\widetilde{\mathbf{x}}_j^k)\right\rangle\\ \le&(1-\theta)\left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \nabla g_j(\widetilde{\mathbf{x}}_j^k)\right\rangle - \eta|\nabla g_j(\widetilde{\mathbf{x}}_j^k)|^2 + \frac{\eta}{4\theta}|\widetilde{\delta}_j^k|^2 + \theta\eta|\nabla g_j(\widetilde{\mathbf{x}}_j^k)|^2\\ =&(1-\theta)\left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \nabla g_j(\widetilde{\mathbf{x}}_j^k)\right\rangle - (1-\theta)\eta|\nabla g_j(\widetilde{\mathbf{x}}_j^k)|^2 + \frac{\eta}{4\theta}|\widetilde{\delta}_j^k|^2.\end{aligned}$$

Plugging into (32), we have

$$\begin{aligned}g_j(\widetilde{\mathbf{x}}_j^{k+1}) - g_j(\widetilde{\mathbf{x}}_j^k) \le&\frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + (1-\theta)\left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \nabla g_j(\widetilde{\mathbf{x}}_j^k)\right\rangle\\ &- (1-\theta)\eta|\nabla g_j(\widetilde{\mathbf{x}}_j^k)|^2 + \frac{\eta}{4\theta}|\widetilde{\delta}_j^k|^2.\end{aligned} \tag{33}$$

Rearranging and squaring both sides of (30) and using $(a+b)^2 \le (1+\frac{\theta}{1-\theta})a^2 + (1+\frac{1-\theta}{\theta})b^2 = \frac{a^2}{1-\theta} + \frac{b^2}{\theta}$, we have

$$\begin{aligned}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 \le&\frac{1}{1-\theta}\left|(1-\theta)(\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}) - \eta\nabla g_j(\widetilde{\mathbf{x}}_j^k)\right|^2 + \frac{\eta^2|\widetilde{\delta}_j^k|^2}{\theta}\\ =&(1-\theta)|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 + \frac{\eta^2}{1-\theta}|\nabla g_j(\widetilde{\mathbf{x}}_j^k)|^2 - 2\eta\left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \nabla g_j(\widetilde{\mathbf{x}}_j^k)\right\rangle + \frac{\eta^2|\widetilde{\delta}_j^k|^2}{\theta}.\end{aligned} \tag{34}$$

Multiplying both sides of (34) by $\frac{(1-\theta)^2}{\eta}$, adding it to (33), and rearranging the terms, we have

$$\begin{aligned}
&g_j(\widetilde{\mathbf{x}}_j^{k+1}) - g_j(\widetilde{\mathbf{x}}_j^k) \\
\leq& -\left(\frac{(1-\theta)^2}{\eta} - \frac{\lambda_j}{2}\right)|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{(1-\theta)^3}{\eta}|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 \\
&- (1-2\theta)(1-\theta)\left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \nabla g_j(\widetilde{\mathbf{x}}_j^k)\right\rangle + \left(\frac{\eta}{4\theta} + \frac{\eta(1-\theta)^2}{\theta}\right)|\widetilde{\delta}_j^k|^2 \\
=& -\left(\frac{(1-\theta)^2}{\eta} - \frac{\lambda_j}{2}\right)|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{(1-\theta)^3}{\eta}|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 \\
&- (1-2\theta)(1-\theta)\left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \lambda_j(\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^0) + \widetilde{\nabla}_j f(\mathbf{x}^0)\right\rangle + \left(\frac{\eta}{4\theta} + \frac{\eta(1-\theta)^2}{\theta}\right)|\widetilde{\delta}_j^k|^2 \\
\leq& -\left(\frac{(1-\theta)^2}{\eta} - \frac{\lambda_j}{2}\right)|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{(1-\theta)^3}{\eta}|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 + \frac{5\eta}{4\theta}|\widetilde{\delta}_j^k|^2 \\
&- (1-2\theta)(1-\theta)\left(\frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 + \frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^0|^2 - \frac{\lambda_j}{2}|\widetilde{\mathbf{x}}_j^{k-1} - \widetilde{\mathbf{x}}_j^0|^2 + \left\langle \widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}, \widetilde{\nabla}_j f(\mathbf{x}^0)\right\rangle\right) \\
=& -\left(\frac{(1-\theta)^2}{\eta} - \frac{\lambda_j}{2}\right)|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \left(\frac{(1-\theta)^3}{\eta} - \frac{(1-2\theta)(1-\theta)\lambda_j}{2}\right)|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 \\
&+ \frac{5\eta}{4\theta}|\widetilde{\delta}_j^k|^2 - (1-2\theta)(1-\theta)\left(g_j(\widetilde{\mathbf{x}}_j^k) - g_j(\widetilde{\mathbf{x}}_j^{k-1})\right).
\end{aligned}$$

Note that

$$\begin{aligned}
&\frac{(1-\theta)^2}{\eta} - \frac{\lambda_j}{2} - \frac{(1-\theta)^3}{\eta} + \frac{(1-2\theta)(1-\theta)\lambda_j}{2} \\
=&\frac{(1-\theta)^2\theta}{\eta} - \frac{\lambda_j(3\theta - 2\theta^2)}{2} \overset{a}{\geq} \frac{(1-\theta)^2\theta}{\eta} - \frac{3\theta - 2\theta^2}{8\eta} \overset{b}{\geq} \frac{9\theta}{20\eta},
\end{aligned}$$

where we use $\lambda_j \leq L = \frac{1}{4\eta}$ in $\overset{a}{\geq}$ and $\theta \leq \frac{1}{10}$ in $\overset{b}{\geq}$. So we have

$$\begin{aligned}
&g_j(\widetilde{\mathbf{x}}_j^{k+1}) - g_j(\widetilde{\mathbf{x}}_j^k) + (1-2\theta)(1-\theta)\left(g_j(\widetilde{\mathbf{x}}_j^k) - g_j(\widetilde{\mathbf{x}}_j^{k-1})\right) \\
\leq& -\left(\frac{(1-\theta)^3}{\eta} - \frac{(1-2\theta)(1-\theta)\lambda_j}{2} + \frac{9\theta}{20\eta}\right)|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 \\
&+ \left(\frac{(1-\theta)^3}{\eta} - \frac{(1-2\theta)(1-\theta)\lambda_j}{2}\right)|\widetilde{\mathbf{x}}_j^k - \widetilde{\mathbf{x}}_j^{k-1}|^2 + \frac{5\eta}{4\theta}|\widetilde{\delta}_j^k|^2.
\end{aligned}$$

Summing over $k = 0, 1, \cdots, \mathcal{K}-1$ and using $\mathbf{x}^0 = \mathbf{x}^{-1}$, we have

$$\begin{aligned}
&g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}}) - g_j(\widetilde{\mathbf{x}}_j^0) + (1-2\theta)(1-\theta)\left(g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}-1}) - g_j(\widetilde{\mathbf{x}}_j^0)\right) \\
\leq& -\left(\frac{(1-\theta)^3}{\eta} - \frac{(1-2\theta)(1-\theta)\lambda_j}{2}\right)|\widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^{\mathcal{K}-1}|^2 - \frac{9\theta}{20\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{5\eta}{4\theta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\delta}_j^k|^2.
\end{aligned} \tag{35}$$

Denoting $\alpha = \frac{1}{1+(1-2\theta)(1-\theta)} \in [\frac{1}{2}, \frac{1}{1.72}]$ and multiplying both sides of (35) by α, we have

$$\begin{aligned}
&\alpha\left(g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}}) - g_j(\widetilde{\mathbf{x}}_j^0)\right) + (1-\alpha)\left(g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}-1}) - g_j(\widetilde{\mathbf{x}}_j^0)\right)\\
\leq& -\alpha\left(\frac{(1-\theta)^3}{\eta} - \frac{(1-2\theta)(1-\theta)\lambda_j}{2}\right)|\widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^{\mathcal{K}-1}|^2 - \frac{9\alpha\theta}{20\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{5\alpha\eta}{4\theta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\delta}_j^k|^2 \quad (36)\\
\leq& -\frac{1}{2}\left(\frac{(1-\theta)^3}{\eta} - \frac{(1-2\theta)(1-\theta)\lambda_j}{2}\right)|\widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^{\mathcal{K}-1}|^2 - \frac{9\theta}{40\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{\eta}{\theta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\delta}_j^k|^2,
\end{aligned}$$

where we use (37) in the last inequality. On the other hand, from $\mathbf{z}^{\mathcal{K}} = \frac{\mathbf{x}^{\mathcal{K}} + (1-2\theta)(1-\theta)\mathbf{x}^{\mathcal{K}-1}}{1+(1-2\theta)(1-\theta)} = \alpha\mathbf{x}^{\mathcal{K}} + (1-\alpha)\mathbf{x}^{\mathcal{K}-1}$, we have

$$\begin{aligned}
&g_j(\widetilde{\mathbf{z}}_j^{\mathcal{K}}) - g_j(\widetilde{\mathbf{x}}_j^0)\\
=&\frac{\lambda_j}{2}\left|\alpha(\widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^0) + (1-\alpha)(\widetilde{\mathbf{x}}_j^{\mathcal{K}-1} - \widetilde{\mathbf{x}}_j^0)\right|^2 + \alpha\left\langle\widetilde{\nabla}_j f(\mathbf{x}^0), \widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^0\right\rangle\\
&+ (1-\alpha)\left\langle\widetilde{\nabla}_j f(\mathbf{x}^0), \widetilde{\mathbf{x}}_j^{\mathcal{K}-1} - \widetilde{\mathbf{x}}_j^0\right\rangle\\
\overset{c}{=}&\frac{\lambda_j\alpha}{2}|\widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^0|^2 + \frac{\lambda_j(1-\alpha)}{2}|\widetilde{\mathbf{x}}_j^{\mathcal{K}-1} - \widetilde{\mathbf{x}}_j^0|^2 - \frac{\lambda_j\alpha(1-\alpha)}{2}|\widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^{\mathcal{K}-1}|^2\\
&+ \alpha\left\langle\widetilde{\nabla}_j f(\mathbf{x}^0), \widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^0\right\rangle + (1-\alpha)\left\langle\widetilde{\nabla}_j f(\mathbf{x}^0), \widetilde{\mathbf{x}}_j^{\mathcal{K}-1} - \widetilde{\mathbf{x}}_j^0\right\rangle\\
=&\alpha\left(g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}}) - g_j(\widetilde{\mathbf{x}}_j^0)\right) + (1-\alpha)\left(g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}-1}) - g_j(\widetilde{\mathbf{x}}_j^0)\right) - \frac{\lambda_j\alpha(1-\alpha)}{2}|\widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^{\mathcal{K}-1}|^2\\
\overset{d}{\leq}&\alpha\left(g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}}) - g_j(\widetilde{\mathbf{x}}_j^0)\right) + (1-\alpha)\left(g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}-1}) - g_j(\widetilde{\mathbf{x}}_j^0)\right) + \frac{1}{32\eta}|\widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^{\mathcal{K}-1}|^2\\
\overset{e}{\leq}&-\frac{1}{2}\left(\frac{(1-\theta)^3}{\eta} - \frac{(1-2\theta)(1-\theta)\lambda_j}{2} - \frac{1}{16\eta}\right)|\widetilde{\mathbf{x}}_j^{\mathcal{K}} - \widetilde{\mathbf{x}}_j^{\mathcal{K}-1}|^2 - \frac{9\theta}{40\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{\eta}{\theta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\delta}_j^k|^2\\
\overset{f}{\leq}&-\frac{9\theta}{40\eta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\mathbf{x}}_j^{k+1} - \widetilde{\mathbf{x}}_j^k|^2 + \frac{\eta}{\theta}\sum_{k=0}^{\mathcal{K}-1}|\widetilde{\delta}_j^k|^2,
\end{aligned}$$

where we use $|\alpha x + (1-\alpha)y|^2 = \alpha x^2 + (1-\alpha)y^2 - \alpha(1-\alpha)|x-y|^2$ in $\overset{c}{\leq}$, $\lambda_j \geq -L = -\frac{1}{4\eta}$ in $\overset{d}{\leq}$, (36) in $\overset{e}{\leq}$, and

$$\frac{(1-\theta)^3}{\eta} - \frac{(1-2\theta)(1-\theta)\lambda_j}{2} - \frac{1}{16\eta} \geq \frac{(1-\theta)^3}{\eta} - \frac{1}{8\eta} - \frac{1}{16\eta} \geq 0 \quad (37)$$

with $\theta \in [0, \frac{1}{10}]$ in $\overset{f}{\leq}$. Summing over j, using (31) and (8a), we have

$$\begin{aligned}
g(\widetilde{\mathbf{z}}^{\mathcal{K}}) - g(\widetilde{\mathbf{x}}^0) = \sum_j g_j(\widetilde{\mathbf{z}}_j^{\mathcal{K}}) - g_j(\widetilde{\mathbf{x}}_j^0) \leq& -\frac{9\theta}{40\eta}\sum_{k=0}^{\mathcal{K}-1}\|\widetilde{\mathbf{x}}^{k+1} - \widetilde{\mathbf{x}}^k\|^2 + \frac{\eta\rho^2 B^4\mathcal{K}}{4\theta}\\
\leq& -\frac{9\theta B^2}{40\eta\mathcal{K}} + \frac{\eta\rho^2 B^4\mathcal{K}}{4\theta}.
\end{aligned}$$

Plugging into (29), we have the conclusion. ■

Remark 20 *Comparing with the proofs in Lemmas 9 and 10, we do not divide the eigenvalues into two groups in the proof of Lemma 19. However, the bad thing is that* $g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}}) - g_j(\widetilde{\mathbf{x}}_j^0)$ *and* $g_j(\widetilde{\mathbf{x}}_j^{\mathcal{K}-1}) - g_j(\widetilde{\mathbf{x}}_j^0)$ *appear simultaneously on the left hand side of (35). This is the reason why we introduce the vector* $\mathbf{z}^k$ *in Algorithm 3.*

Combing Lemmas 18 and 19, similar to Corollary 12, we have

$$f(\mathbf{z}^{\mathcal{K}}) - f(\mathbf{x}^0) \leq -\frac{\epsilon^{3/2}}{\sqrt{\rho}}.$$

Similar to Lemma 13, we also have $\|\nabla f(\hat{\mathbf{x}})\| \leq \frac{2\sqrt{2}B}{\eta K^2} + \frac{2\theta B}{\eta K} + \rho B^2 \leq 242\epsilon$ in the last epoch. Using the same proofs of Theorem 1 at the end of Section 4.1.3, we can prove Theorem 6.

5. Experiments

We test the practical performance on the matrix completion problem (Negahban and Wainwright, 2012; Hardt, 2014) and one bit matrix completion problem (Davenport et al., 2014), and end this section by discussing the gap between theory and practice.

5.1 Matrix completion

In matrix completion (Negahban and Wainwright, 2012; Hardt, 2014), we aim to recover the true low rank matrix from a set of randomly observed entries, which can be formulated as follows:

$$\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \frac{1}{2N}\sum_{(i,j)\in\mathcal{O}} (\mathbf{X}_{i,j} - \mathbf{X}^*_{i,j})^2, \quad s.t. \quad \mathrm{rank}(\mathbf{X}) \leq r,$$

where $\mathcal{O}$ is the set of randomly observed entries with size N and $\mathbf{X}^*$ is the true low rank matrix to recover. We reformulate the above problem in the following matrix factorization form:

$$\min_{\mathbf{U}\in\mathbb{R}^{m\times r},\mathbf{V}\in\mathbb{R}^{n\times r}} \frac{1}{2N}\sum_{(i,j)\in\mathcal{O}} ((\mathbf{U}\mathbf{V}^T)_{i,j} - \mathbf{X}^*_{i,j})^2 + \frac{1}{2N}\|\mathbf{U}^T\mathbf{U} - \mathbf{V}^T\mathbf{V}\|_F^2,$$

where r is the rank of $\mathbf{X}^*$ and the regularization is used to balance $\mathbf{U}$ and $\mathbf{V}$.

We verify the performance on the Movielens-10M, Movielens-20M and Netflix data sets, where the corresponding observed matrices are of size 69878×10677, 138493×26744, and 480189×17770, respectively. We set $r = 10$. Denote $\mathbf{X}_{\mathcal{O}}$ to be the observed data and $\mathbf{A}\Sigma\mathbf{B}^T$ to be its SVD. We initialize $\mathbf{U} = \mathbf{A}_{:,1:r}\sqrt{\Sigma_{1:r,1:r}}$ and $\mathbf{V} = \mathbf{B}_{:,1:r}\sqrt{\Sigma_{1:r,1:r}}$ for all the compared methods. It is efficient to compute the maximal r singular values and the corresponding singular vectors of sparse matrices, for example, by Lanczos.

We compare Ada-RAGD-NC (Algorithm 2) and Ada-RHB-NC (Algorithm 4) with Jin's AGD (Jin et al., 2018), the "convex until proven guilty" method (Carmon et al., 2017),

heuristic restarted AGD (O'Donoghue and Candès, 2015), nonlinear conjugate gradient (CG) (Polak and Ribiere, 1969), and gradient descent (GD). We do not compare with RAGD-NC (Algorithm 1) and RHB-NC (Algorithm 3) because the two methods restart at almost every iteration due to the small hyperparameter B. Their performance is almost the same as GD and their plots almost coincide with that of GD. Heuristic RAGD consists of the following iterations

$$\mathbf{x}^{k+1} = \mathbf{y}^k - \eta \nabla f(\mathbf{y}^k), \quad \mathbf{y}^{k+1} = \mathbf{x}^{k+1} + \frac{m^{k+1}-1}{m^{k+1}+2}(\mathbf{x}^{k+1} - \mathbf{x}^k),$$

where $m^0 = 1$ and

$$m^{k+1} = \begin{cases} m^k + 1, & \text{if } f(\mathbf{x}^{k+1}) \le f(\mathbf{x}^k), \\ 1, & \text{otherwise.} \end{cases}$$

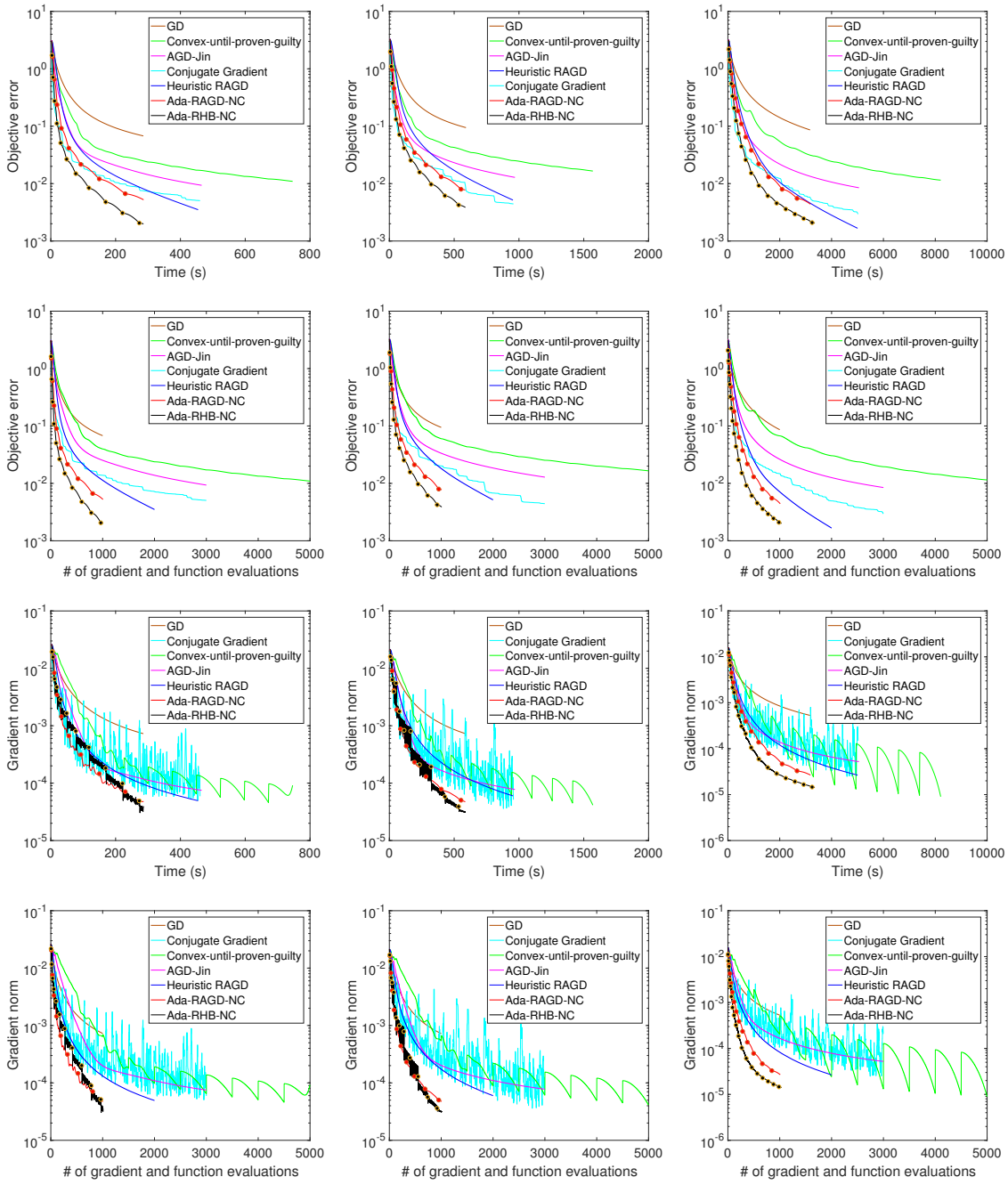
The nonlinear conjugate gradient has the following steps

$$\delta^k = -\nabla f(\mathbf{x}^k) + \max\left\{ \frac{\langle \nabla f(\mathbf{x}^k), \nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^{k-1}) \rangle}{\|\nabla f(\mathbf{x}^{k-1})\|^2}, 0 \right\} \delta^{k-1}, \quad \mathbf{x}^{k+1} = \mathbf{x}^k + \eta^k \delta^k,$$

where $\delta^{-1} = 0$ and we follow (Carmon et al., 2017) to set η^k by the following backtracking line search: set $\eta^k = 2\eta^{k-1}$ and check whether $f(\mathbf{x}^k + \eta^k \delta^k) \le f(\mathbf{x}^k) + \frac{\eta^k \langle \delta^k, \nabla f(\mathbf{x}^k) \rangle}{2}$ holds. If it does not hold, set $\eta^k = \eta^k/2$ and repeat.

We tune the best stepsize η for each compared method (except CG) on each dataset. For CG, we set the same stepsize as GD since CG adaptively tune η during the iterations. For Ada-RAGD-NC and Ada-RHB-NC, we set $\epsilon = 10^{-4}$, $B = \sqrt{\frac{\epsilon}{\rho}}$, $\theta = 0.005(\epsilon\rho\eta^2)^{1/4}$, $K = \lfloor 1/\theta \rfloor$, $B_0 = 100$, $\gamma = 10^{-5}$, $c_0 = 1 + 0.001t$ at the tth epoch, and $c_1 = 10$. We use (4) to adaptively tune η and ρ since ρ is unknown, where we set $c_2 = 2$ and initialize $\rho = 1$. When preparing the experiments, we observe that proper B_0, θ, and γ are crucial in the fast convergence of Ada-RAGD-NC and Ada-RHB-NC. We suggest to set θ in $(0, 0.01]$ and γ to be small such that line 11 in Algorithms 2 and 4 is less frequently triggered. B_0 can be set larger when the methods restart frequently. For CG, we stop the line search when it repeats more than 10 times. For Jin's AGD, we tune $\theta = 0.04(\epsilon\rho\eta^2)^{1/4}$ and follow (Jin et al., 2018) to set $\gamma = \frac{\theta^2}{\eta}$ and $s = \frac{\gamma}{4\rho}$ in their method. Since the Hessian Lipschitz constant ρ is unknown, we set it as 1 for Jin's AGD for simplicity. For the "convex until proven guilty" method, we follow the theory in (Carmon et al., 2017) to set the parameters except that we terminate the inner loop after 100 iterations to improve its practical performance. GD and heuristic RAGD have no hyperparameter to tune except the stepsize. Since the optimal function value $f(\mathbf{x}^*)$ is unknown, we run each method for 2000 iterations and use the minimum objective value to approximate the optimal one. We only plot the figures using the first 1000 iterations.

Figure 1 plots the objective error $f(\mathbf{x}^k) - f(\mathbf{x}^*)$ and gradient norm $\|\nabla f(\mathbf{x}^k)\|$. We use running time as the horizontal axis in the first and third row, and the number of function and gradient evaluations as the horizontal axis in the second and forth row. Note that the "convex until proven guilty" method needs at least two gradient evaluations at each iteration while the other methods only need one. For the function evaluations, GD needs



(a) MovieLens-10M (b) MovieLens-20M (c) Netflix

Figure 1: Comparisons on the matrix completion problem. The first and second row: objective error. The third and forth row: gradient norm. The first and third row: use time as the horizontal axis. The second and forth row: use the number of function and gradient evaluations as the horizontal axis. Circles in Ada-RAGD-NC and Ada-RHB-NC indicate where restart occurs.

none, Ada-RAGD-NC and Ada-RHB-NC need one only when restart occurs, heuristic RAGD needs one at each iteration, Jin's AGD needs at least two, CG needs at least one, and the "convex until proven guilty" method needs at least three at each iteration. Thus, GD and our Ada-RAGD-NC and Ada-RHB-NC need less total running time when we run all the methods for 1000 iterations. We see that all the accelerated methods perform better than GD, which verifies the efficiency of acceleration in nonconvex optimization. We also observe that our Ada-RAGD-NC and Ada-RHB-NC decrease the objective error and gradient norm to low level quickly. We observe that the gradient norms of CG oscillate during iterations. It may be because CG uses line search to tune the stepsize dynamically, which may be too large and aggressive. On the other hand, due to the specification of the matrix completion problem, we observe that Jin's AGD and the "convex until proven guilty" method seldom run negative curvature exploitation.

5.2 One bit matrix completion

In one bit matrix completion (Davenport et al., 2014), the signs of a random subset of entries are observed, rather than observing the actual entries. Given a probability density function, for example, the logistic function $f(x) = \frac{e^x}{1+e^x}$, we observe the sign of entry $\mathbf{X}_{i,j}$ as $\mathbf{Y}_{i,j} = 1$ with probability $f(\mathbf{X}_{i,j})$, and observe the sign as -1 with probability $1 - f(\mathbf{X}_{i,j})$. The training model is to minimize the following negative log-likelihood:

$$\begin{aligned} \min_{\mathbf{X}\in\mathbb{R}^{m\times n}} & -\frac{1}{N}\sum_{(i,j)\in\mathcal{O}} \left\{\mathbf{1}_{\mathbf{Y}_{i,j}=1}\log(f(\mathbf{X}_{i,j})) + \mathbf{1}_{\mathbf{Y}_{i,j}=-1}\log(1-f(\mathbf{X}_{i,j}))\right\}, \\ s.t. & \quad \operatorname{rank}(\mathbf{X}) \le r, \end{aligned}$$

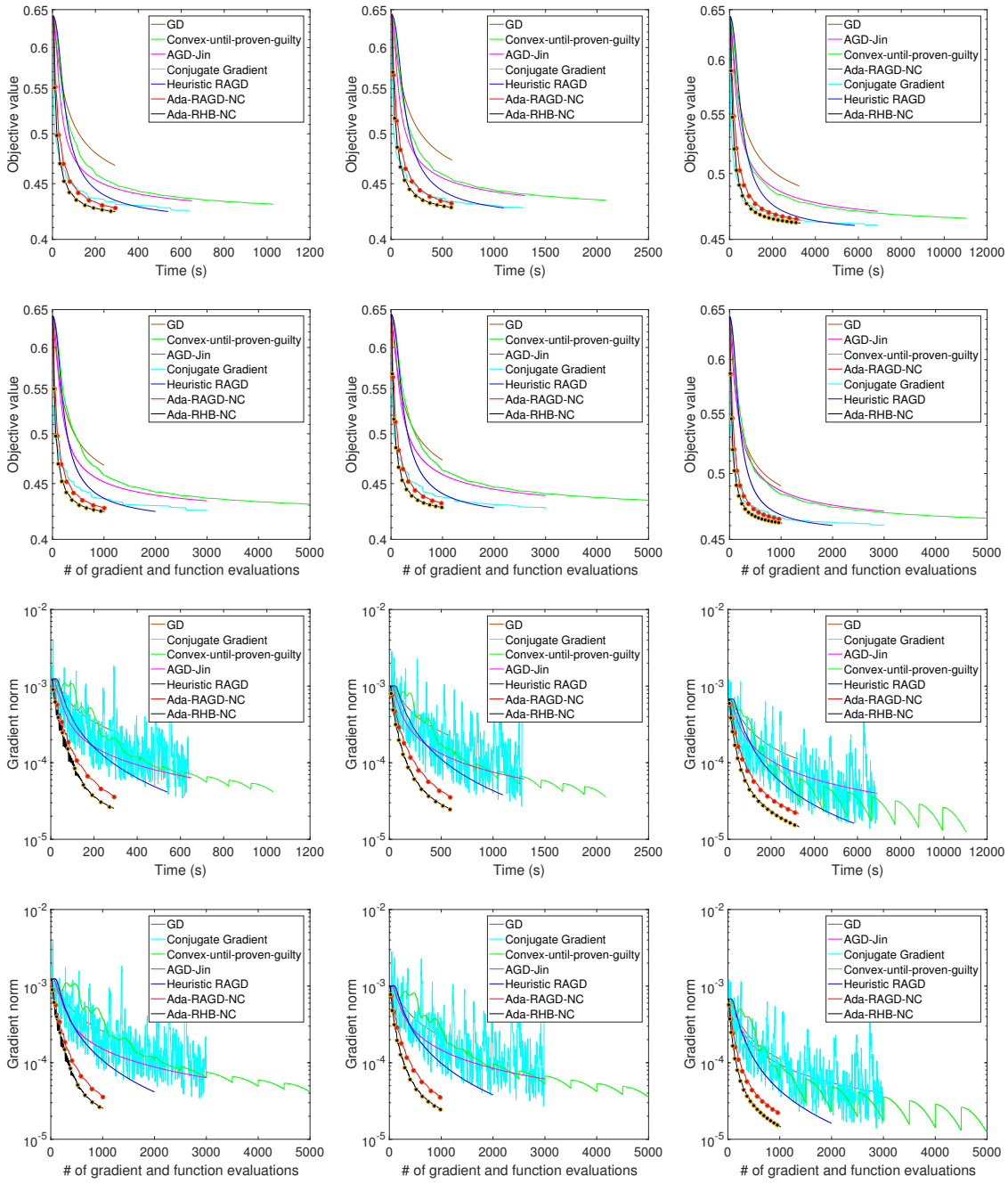
where $\mathbf{1}_{\mathbf{Y}_{i,j}=1} = \begin{cases} 1, & \text{if } \mathbf{Y}_{i,j} = 1, \\ 0, & \text{otherwise.} \end{cases}$ We solve the following reformulated matrix factorization model:

$$\min_{\mathbf{U},\mathbf{V}} -\frac{1}{N}\sum_{(i,j)\in\mathcal{O}}\left\{\mathbf{1}_{\mathbf{Y}_{i,j}=1}\log(f((\mathbf{U}\mathbf{V}^T)_{i,j}))+\mathbf{1}_{\mathbf{Y}_{i,j}=-1}\log(1-f((\mathbf{U}\mathbf{V}^T)_{i,j}))\right\}+\frac{1}{2N}\|\mathbf{U}^T\mathbf{U}-\mathbf{V}^T\mathbf{V}\|_F^2,$$

where $\mathbf{U} \in \mathbb{R}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}^{n\times r}$. We compare Ada-RAGD-NC (Algorithm 2) and Ada-RHB-NC (Algorithm 4) with the methods compared in Section 5.1. The best stepsize is tuned for each method on each data set. We use the same initialization and set the same parameters as those in Section 5.1, and also run each method for 1000 iterations. Figure 2 plots the results. We see that acceleration also takes effect in nonconvex optimization and our Ada-RAGD-NC and Ada-RHB-NC also decrease the objective value and gradient norm to low level quickly.

5.3 Gap Between Theory and Practice

In the previous two sections, we only run Ada-RAGD-NC and Ada-RHB-NC for 1000 iterations such that the objective error and gradient norm are reduced to low level quickly, which is sufficient for practical machine learning applications. In this section, we verity what



(a) MovieLens-10M (b) MovieLens-20M (c) Netflix

Figure 2: Comparisons on the 1 bit matrix completion problem. The first and second row: function value. The third and forth row: gradient norm. The first and third row: use time as the horizontal axis. The second and forth row: use the number of function and gradient evaluations as the horizontal axis. Circles in Ada-RAGD-NC and Ada-RHB-NC indicate where restart occurs.

happens when we run the two methods for a longer time and discuss the gap between theory and practice for nonadaptive RAGD-NC and RHB-NC (Algorithms 1 and 3).

We only report the observations on the Movielens-10M data set, and the results are similar on the other two. For both the matrix completion and one bit matrix completion problems, we run Ada-RAGD-NC and Ada-RHB-NC for 10^5 iterations and use the minimum function value to approximate the optimal one. We set the same parameters as those in Section 5.1. Figure 3 plots the results. We have the following observations and conclusions.

1. We see that line 11 (in fact, step (4)) in Algorithms 2 and 4 is invoked only once for both two methods (marked by the blue-green circle), at which time the objective error and gradient norm increase substantially. Then we decrease B_0 and η and increase the estimated ρ properly. After the adaptive update, line 11 is never invoked.

2. When B_0 decreases to be smaller than B (marked by the yellow circle), we see that both methods restart frequently. Specifically, we observe that for the matrix completion problem, Ada-RAGD-NC restarts every 16 iterations after $B_0 \leq B$ while Ada-RHB-NC restarts every 10 iterations. For the one bit matrix completion problem, Ada-RAGD-NC restarts every 4 iterations after $B_0 \leq B$ while Ada-RHB-NC restarts every 3 iterations. It seems to take an extremely long time to break the while loop (that is, no restart occurs in K iterations), especially for high dimensional problems ($\|\mathbf{x}^{t+1} - \mathbf{x}^t\|$ is not likely to be small even if $|\mathbf{x}_i^{t+1} - \mathbf{x}_i^t|$ is small for each $i = 1, 2 \cdots, d$). Thus, we suggest to stop the algorithm in practice when the gradient norm is smaller than a threshold or the number of iterations exceeds the maximum one.

3. Note that Ada-RAGD-NC and Ada-RHB-NC reduce to their nonadaptive counterparts (Algorithms 1 and 3) when $B_0 \leq B$, and the plots after the yellow circles may illustrate the practical performance of the nonadaptive methods. Thus, nonadaptive RAGD-NC and RHB-NC (Algorithms 1 and 3) are only for the theoretical purpose and we do not suggest to use them in practice due to their frequent restart, unless we do not follow the theory to set the parameters, especially the parameter B.

6. Conclusion

This paper proposes two simple accelerated gradient methods, restarted AGD and restarted HB, for general nonconvex problems with Lipschitz continuous gradient and Hessian. Our simple methods find an ϵ-approximate first-order stationary point within $\mathcal{O}(\epsilon^{-7/4})$ gradient evaluations, which improves over the best known complexity by the $\mathcal{O}(\log \frac{1}{\epsilon})$ factor. Our proofs only use elementary analysis. We hope our analysis may lead to a better understanding of the acceleration mechanism for nonconvex optimization.

Acknowledgments

The authors were supported by National Key R&D Program of China (2022ZD0160302), NSF China (grant no.s 62006116, 62276004), Jiangsu Province Basic Research Program (grant no. BK20200439), and the major key project of PCL, China (No. PCL2021A12).

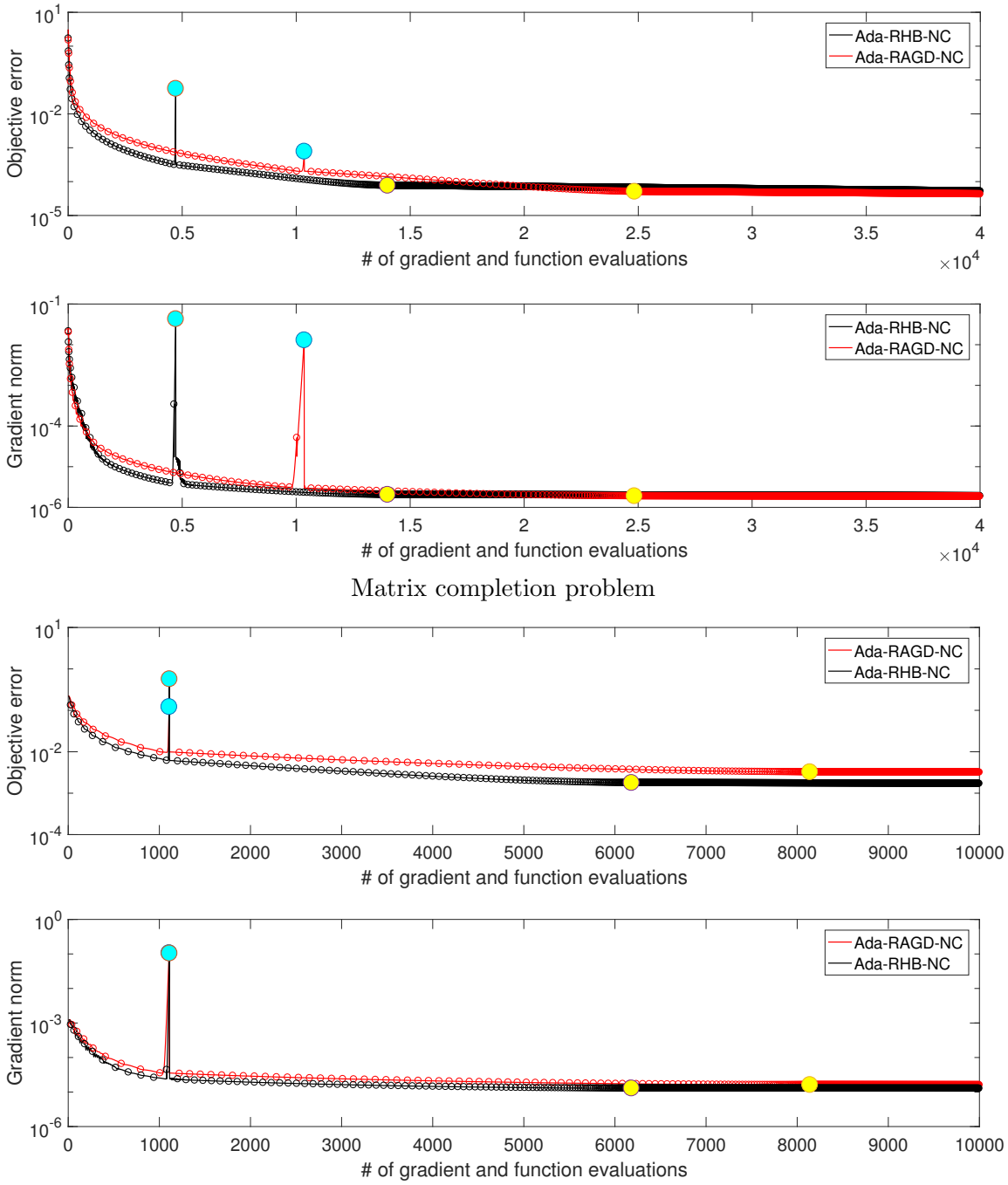


Figure 3: Comparisons of objective error and gradient norm on the Movielens-10M dataset. Top two: matrix completion. Bottom two: 1 bit matrix completion. Small hollow circles indicate where restart occurs. Blue-green circles indicate where line 11 in Algorithms 2 and 4 is invoked. Yellow circles indicate where B_0 decreases to be smaller than B for the first time.

References

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima for nonconvex optimization in linear time. In *ACM Symposium on the Theory of Computing (STOC)*, pages 1195–1199, 2017.

Zeyuan Allen-Zhu and Yuanzhi Li. Neon2: Finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3716–3726, 2018.

Emmanuel J. Candès, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.

Yair Carmon and John Duchi. Analysis of krylov subspace solutions of regularized nonconvex quadratic problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10728–10738, 2018.

Yair Carmon and John Duchi. First-order methods for nonconvex quadratic minimization. *SIAM Review*, 62(2):395–436, 2020.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Convex until proven guilty: Dimension-free acceleration of gradient descent on non-convex functions. In *International Conference on Machine Learning (ICML)*, pages 654–663, 2017.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

Yair Carmon, John Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184:71–120, 2020.

Yair Carmon, John Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: First-order methods. *Mathematical Programming*, 185:315–355, 2021.

Mark A. Davenport, Yaniv Plan, Ewout van den Berg, and Mary Wootters. 1-bit matrix completion. *Information and Inference*, 3(3):189–223, 2014.

Damek Davis and Wotao Yin. Convergence rate analysis of several splitting schemes. *Part of the Scientific Computation book series (SCIENTCOMP)*, 2017.

Cong Fang, Zhouchen Lin, and Tong Zhang. Sharp analysis for nonconvex SGD escaping from saddle points. In *Conference On Learning Theory (COLT)*, pages 1192–1234, 2019.

Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. Global convergence of the heavy-ball method for convex optimization. In *European Control Conference (ECC)*, pages 310–315, 2015.

Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156:59–99, 2016.

Moritz Hardt. Understanding alternating minimization for matrix completion. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 651–660, 2014.

Nicholas J. A. Harvey, Christopher Liaw, Yaniv Plan, and Sikander Randhawa. Tight analyses for non-smooth stochastic gradient descent. In *Conference On Learning Theory (COLT)*, pages 1579–1613, 2019.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning (ICML)*, pages 1724–1732, 2017.

Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference On Learning Theory (COLT)*, pages 1042–1085, 2018.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference On Learning Theory (COLT)*, pages 1246–1257, 2016.

Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in Neural Information Processing Systems (NIPS)*, pages 379–387, 2015.

Huan Li and Zhouchen Lin. Restarted nonconvex accelerated gradient descent: No more polylogarithmic factor in the $o(\epsilon^{-7/4})$ complexity. In *International Conference on Machine Learning (ICML)*, pages 12901–12916, 2022.

Huan Li, Cong Fang, and Zhouchen Lin. Accelerated first-order optimization algorithms for machine learning. *Proceedings of the IEEE*, 108(11):2067–2082, 2020.

Qunwei Li, Yi Zhou, Yingbin Liang, and Pramod K Varshney. Convergence analysis of proximal gradient with momentum for nonconvex optimization. In *International Conference on Machine Learning (ICML)*, pages 2111–2119, 2017.

Jingwei Liang, Jalal M. Fadili, and Gabriel Peyré. A multi-step inertial forward–backward splitting method for non-convex optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4035–4043, 2016.

Sahand Negahban and Martin J. Wainwright. Resticted strong convexity and weighted matrix completion: optimal bounds with noise. *Journal of Machine Learning Research*, 13(53):1665–1697, 2012.

Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Yurii Nesterov. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika I Mateaticheskie Metody*, 24(3):509–517, 1988.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science+Business Media, 2004.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

Yurii Nesterov and Boris T. Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108:177–205, 2006.

Praneeth Netrapalli, U N Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1107–1115, 2014.

Peter Ochs. Local convergence of the heavy-ball method and ipiano for non-convex optimization. *Journal of Optimization Theory and Applications*, 177:153–180, 2018.

Peter Ochs, Yunjin Chen, Thomas Brox, and Thomas Pock. ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

Brendan O'Donoghue and Emmanuel Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, 2015.

Michael O'Neill and Stephen J. Wright. Behavior of accelerated gradient methods near critical points of nonconvex functions. *Mathematical Programming*, 176:403–427, 2019.

E. Polak and G. Ribiere. Note sur la convergence de méthodes de directions conjuguées. *Revue française d'informatique et de recherche opérationnelle. Série rouge*, 3(16):35–43, 1969.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):791–803, 1964.

Clement W. Royer and Stephen J. Wright. Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM Journal on Optimization*, 28(2): 1448–1477, 2018.

Clement W. Royer, Michael O'Neill, and Stephen J. Wright. A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization. *Mathematical Programming*, 180:451–488, 2020.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning (ICML)*, pages 71–79, 2013.

Max Simchowitz, Ahmed El Alaoui, and Benjamin Recht. On the gap between strict-saddles and true convexity: An $\Omega(\log d)$ lower bound for eigenvector approximation. *Arxiv preprint: 1704.04548*, 2017.

Tao Sun, Dongsheng Li, Zhe Quan, Hao Jiang, Shengguo Li, and Yong Dou. Heavy-ball algorithms always escape saddle points. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3520–3526, 2019.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1139–1147, 2013.

Jun-Kun Wang, Chi-Heng Lin, Andre Wibisono, and Bin Hu. Provable acceleration of heaby ball beyond quadratics for a class of Polyak-Lojasiewicz functions when the non-convexity is averaged-out. In *International Conference on Machine Learning (ICML)*, pages 22839–22864, 2022.

Yi Xu, Rong Jin, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5535–5545, 2018.

S.K. Zavriev and F.V. Kostyuk. Heavy-ball method in nonconvex optimization problems. *Computational Mathematics and Modeling*, 4(4):336–341, 1993.