

An Optimal High-Order Tensor Method for Convex Optimization

Bo JIANG ^{*} Haoyue WANG [†] Shuzhong ZHANG [‡]

April 20, 2020

Abstract

This paper is concerned with finding an *optimal* algorithm for minimizing a composite convex objective function. The basic setting is that the objective is the sum of two convex functions: the first function is smooth with up to the d -th order derivative information available, and the second function is possibly non-smooth, but its proximal tensor mappings can be computed approximately in an efficient manner. The problem is to find – in that setting – the best possible (optimal) iteration complexity for convex optimization. Along that line, for the smooth case (without the second non-smooth part in the objective) Nesterov proposed ([25], 1983) an optimal algorithm for the first-order methods ($d = 1$) with iteration complexity $O(1/k^2)$, while high-order tensor algorithms (using up to general d th order tensor information) with iteration complexity $O(1/k^{d+1})$ were recently established in [3, 27]. In this paper, we propose a new high-order tensor algorithm for the general composite case, with the iteration complexity of $O(1/k^{(3d+1)/2})$, which matches the lower bound for the d -th order methods as established in [27, 31], and hence is optimal. Our approach is based on the *Accelerated Hybrid Proximal Extragradient* (A-HPE) framework proposed by Monteiro and Svaiter in [24], where a bisection procedure is installed for each A-HPE iteration. At each bisection step a proximal tensor subproblem is approximately solved, and the total number of bisection steps per A-HPE iteration is shown to be bounded by a logarithmic factor in the precision required.

Keywords: convex optimization; tensor method; acceleration; iteration complexity.

Mathematics Subject Classification: 90C60, 90C25, 47H05

^{*}Research Institute for Interdisciplinary Sciences, School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China. Email: isybojiang@gmail.com. Research of this author was supported in part by NSFC Grants 11771269 and 11831002, and Program for Innovative Research Team of Shanghai University of Finance and Economics.

[†]Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, USA, (email: haoyuew@mit.edu)

[‡]Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, MN 55455, USA (email: zhangs@umn.edu); joint appointment with Institute of Data and Decision Analytics, The Chinese University of Hong Kong, Shenzhen, China (email: zhangs@umn.edu). Research of this author was supported in part by the National Science Foundation (Grant CMMI-1462408).

d

1 Introduction

In this paper, we consider the following composite unconstrained convex optimization:

$$\min_{x \in \mathbb{R}^n} F(x) := f(x) + h(x), \quad (1.1)$$

where f is differentiable and convex, and h is convex but possibly non-smooth. In this context, we assume that convex tensor (polynomial) proximal mappings regarding h can be approximately computed efficiently. Given that structure, a fundamental quest is to find an *optimal* algorithm that solves the above problem, using the available derivative information of the smooth part f .

In case $F(x) = f(x)$, and only the gradient information of f is available, Nesterov [25] proposed a gradient-type algorithm, which achieves the overall iteration complexity of $O(1/k^2)$, matching the lower bound on the iteration complexity of this class of solution methods, hence is known to be an *optimal* algorithm among all the first-order methods. Since Nesterov’s seminal work [25], especially in the recent years when the large scale machine learning applications have come under the spotlight, there has been a surge of research effort to extend Nesterov’s approach to more general settings; see e.g. [4, 12, 19, 14, 30], and/or to incorporate certain adaptive strategies to enhance the practical performances of the acceleration; see e.g. [20, 29, 13]. At the same time, there has also been a considerable research effort to fully understand the underpinning mechanism of the first-order acceleration phenomenon; see e.g. [7, 32, 33, 34].

When the Hessian information is available, Nesterov [26] proposed an acceleration scheme for cubic regularized Newton’s method, and he showed that the iteration complexity bound improves from $O(1/k^2)$ to $O(1/k^3)$. A few years later, Monteiro and Svaiter [24] proposed a totally different acceleration scheme, which they termed as *Accelerated Hybrid Proximal Extragradient Method* (A-HPE) framework, and they proved that if the second-order information is incorporated into the A-HPE framework then the corresponding accelerated Newton proximal extragradient method has a superior iteration complexity bound of $O(1/k^{7/2})$ over $O(1/k^3)$. In 2018, Arjevani, Shamir and Shiff [31] showed that $O(1/k^{7/2})$ is actually a lower bound for the oracle complexity of the second-order methods for convex smooth optimization. This shows that the accelerated Newton proximal extragradient method is an optimal second-order method.

As evidenced by the special cases $d = 1$ and $d = 2$, there is a clear tradeoff between the level of derivation information required and the overall iteration complexity improved. Therefore, a natural and important question arises:

What is the *exact* tradeoff relationship between d and the worst-case iteration complexity?

Such question has been in fact raised and addressed in some way in recent works [5, 10, 11, 22] in the context of nonconvex optimization. For convex optimization, the accelerated cubic regularized Newton method was generalized to the general high-order case [3, 27] with the iteration complexity

being $O(1/k^{d+1})$, where d is the order of derivative information used in the algorithm. Jiang, Lin and Zhang [18] extended Nesterov’s approach to accommodate the composite optimization (1.1) and relaxed the requirement on the knowledge of problem parameters such as the Lipschitz constants and the requirement on the exact solutions of the subproblems while maintaining the same iteration bound as in [3, 27]. Along the line of bounding the worst case iteration complexity using up to the d -th order derivative information, there have also been significant progresses as well. Arjevani, Shamir and Shiff [31] showed that the worst case iteration complexity of any algorithm in that setting cannot be better than $O(1/k^{(3d+1)/2})$. A simplified analysis of the bound can be found in Nesterov [27]. So, there was a gap between the achieved iteration bound $O(1/k^{d+1})$ and the best possible bound of $O(1/k^{(3d+1)/2})$. Clearly at least one of the two bounds is improvable. In this paper, we aim to settle the above theoretical quest by providing a new implementable algorithm whose iteration complexity is precisely $O(1/k^{(3d+1)/2})$. As a result, the tradeoff relationship discussed above is pinned down to be exactly $O(1/k^{(3d+1)/2})$.

Our algorithm is based on the A-HPE framework of Monteiro and Svaiter [24], which is presented as Algorithm 1 in this paper. In fact, our algorithm specifies a way to generate an approximate solution through the use of high order derivative information by Taylor expansion. In each iteration, such approximate solution is computed by means of a bisection process. At each bisection step, a regulated convex tensor (polynomial) optimization subproblem is approximately solved. Moreover, we show that, to implement one A-HPE iteration, the number of bisection steps – each calling to solve a convex tensor subproblems – is upper bounded by a logarithmic factor in the inverse of the required precision. Our bisection procedure is similar to the one proposed in [24] for the case $d = 2$; however, a key modification is applied which enables the removal of the so-called “bracketing stage” used in [24]. After submitting the first version of the paper, we became aware of two other independent works [15, 8] establishing similar iteration bounds as ours, with the main difference being that the focus of [15, 8] is on the smooth case: $F(x) = f(x)$, while our method accommodates a composite objective function. The common theoretical development by the three groups was subsequently jointly announced in the form of abstract at Conference on Learning Theory (COLT) [17]. It is also worth mentioning that other than the afore-mentioned three papers there are some other related works on high-order optimization methods [6, 1, 2] based on large-step A-HPE framework.

The rest of the paper is organized as follows. In Section 2, we introduce some preliminaries including the assumptions and the high-order oracle model used throughout this paper. Then we present our optimal tensor method and its iteration complexity analysis in Section 3. The line search subroutine being used in the main procedure of our optimal tensor method is presented and analyzed in Section 4. Finally, some technical proofs and lemmas are provided in the appendix.

2 Preliminaries

2.1 Notations

We denote $\nabla^d f(x)$ to be the d -th order derivative tensor at point x of function f with the (i_1, \dots, i_d) component given as:

$$\nabla^d f(x)_{i_1, \dots, i_d} = \frac{\partial^d f}{\partial x_{i_1} \dots \partial x_{i_d}}(x), \quad \forall 1 \leq i_1, \dots, i_d \leq n.$$

Given a d -th order tensor \mathcal{T} and vectors $z^1, \dots, z^d \in \mathbb{R}^n$, we denote

$$\mathcal{T}[z^1, \dots, z^d] := \sum_{i_1, \dots, i_d=1}^n \mathcal{T}_{i_1, \dots, i_d} z_{i_1}^1 \dots z_{i_d}^d.$$

The operator norm associated with \mathcal{T} is defined as:

$$\|\mathcal{T}\| := \max_{\|z^i\|=1, i=1, \dots, d} \mathcal{T}[z^1, \dots, z^d].$$

For given z^{k+1}, \dots, z^d , $\mathcal{T}[z^{k+1}, \dots, z^d]$ is a k -th order tensor with the associated (i_1, \dots, i_k) component defined as:

$$\mathcal{T}[z^{k+1}, \dots, z^d]_{i_1, \dots, i_k} := \sum_{i_{k+1}, \dots, i_d=1}^n \mathcal{T}_{i_1, \dots, i_k, i_{k+1}, \dots, i_d} z_{i_{k+1}}^{k+1} \dots z_{i_d}^d$$

for $1 \leq i_1, \dots, i_k \leq n$. Denote

$$(z_*^1, \dots, z_*^k) := \operatorname{argmax}_{\|y^i\|=1, i=1, \dots, k} \left(\mathcal{T}[z^{k+1}, \dots, z^d] \right) [y^1, \dots, y^k].$$

One has

$$\|\mathcal{T}[z^{k+1}, \dots, z^d]\| = \mathcal{T}[z_*^1, \dots, z_*^k, z^{k+1}, \dots, z^d] \leq \|\mathcal{T}\| \|z^{k+1}\| \dots \|z^d\|. \quad (2.1)$$

As a matter of convention, for quantities x and y , we use the notation $y = \Theta(x)$ to indicate the relation that there are positive constants a and b such that $ax \leq y \leq bx$. If a is absent, then we shall indicate the relation as $y = O(x)$.

2.2 High-Order Oracle Model and Regularized Tensor Approximation

In this paper, we consider the following high-order oracle model and the algorithm we are going to propose is such oracle model.

d -th Order Oracle Model

- f is d times Lipschitz-continuous and differentiable with Lipschitz constant L_d for d -th order derivative tensor; i.e.

$$\|\nabla^d f(x) - \nabla^d f(y)\| \leq L_d \|x - y\| \quad \forall x, y \in \mathbb{R}^n, \quad (2.2)$$

where the left side is the d -th order tensor operator norm.

- Given any x , the oracle returns $f(x), \nabla f(x), \nabla^2 f(x), \dots, \nabla^d f(x)$.
- At iteration k , x_k is generated from a deterministic function h and the oracle's responses at any linear combination of x_1, x_2, \dots, x_{k-1} and $\nabla^i f(x_1), \nabla^i f(x_2), \dots, \nabla^i f(x_{k-1})$, where $1 \leq i \leq d$.

Recall that the exact proximal minimization at point x with stepsize $\lambda > 0$ is defined as

$$\min_{y \in \mathbb{R}^n} f(y) + h(y) + \frac{1}{2\lambda} \|y - x\|^2. \quad (2.3)$$

To utilize all the derivative information, we consider the regularized tensor approximation of $f(y)$ at point x :

$$f_x(y) := f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2} \nabla^2 f(x) [y - x]^2 + \dots + \frac{1}{d!} \nabla^d f(x) [y - x]^d + \frac{M}{(d+1)!} \|y - x\|^{d+1}, \quad (2.4)$$

where $M > 0$ is the parameter of the high-order regularization term $\|y - x\|^{d+1}$. Then, by (2.2) and the Taylor expansion, we can bound the gap between $f_x(\cdot)$ and $f(\cdot)$ for any x (see Nesterov [27]):

Lemma 2.1 *For every $x, y \in \mathbb{R}^n$,*

$$\|\nabla f(y) - \nabla f_x(y)\| \leq \frac{L_d + M}{d!} \|y - x\|^d.$$

Therefore, it is natural to consider the tensor approximation of (2.3):

$$\min_{y \in \mathbb{R}^n} f_x(y) + h(y) + \frac{1}{2\lambda} \|y - x\|^2. \quad (2.5)$$

In fact, (2.5) is the subproblem to be solved in the Optimal Tensor Method that will be introduced later. Note that similar subproblems have appeared in [24] and [27]. Specifically, the one used in [24] corresponds to $d = 2$ in (2.5) without the term involving $\|y - x\|^{d+1}$ (i.e., $M = 0$), while [27] uses the subproblem that only minimizes $f_x(y)$ (i.e., without the nonsmooth term $h(y)$ and the quadratic regularization term $\frac{1}{2\lambda} \|y - x\|^2$). In contrast, our above subproblem installs both the high-order and quadratic regularization terms.

Note that the unique solution y of (2.5) is characterized by the following optimality condition:

$$u \in (\nabla f_x + \partial h)(y), \quad \lambda u + y - x = 0. \quad (2.6)$$

For a scalar $\epsilon \geq 0$, the ϵ -subdifferential of a proper closed convex function h is defined as:

$$\partial_\epsilon h(x) := \{u \mid h(y) \geq h(x) + \langle y - x, u \rangle - \epsilon, \forall y \in \mathbb{R}^n\}.$$

With the above notion in mind, let us consider the following approximate solution for (2.6) (hence (2.5)).

Definition 2.1 *Given $(\lambda, x) \in \mathbb{R}_{++} \times \mathbb{R}^n$ and $\hat{\sigma} \geq 0$, the triplet $(y, u, \epsilon) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_+$ is called a $\hat{\sigma}$ -approximate solution of (2.5) at (λ, x) if*

$$u \in (\nabla f_x + \partial_\epsilon h)(y) \quad \text{and} \quad \|\lambda u + y - x\|^2 + 2\lambda\epsilon \leq \hat{\sigma}^2 \|y - x\|^2. \quad (2.7)$$

Obviously, if (y, u) is the solution pair of (2.6), then $(y, u, 0)$ is a $\hat{\sigma}$ -approximate solution of (2.5) at (λ, x) for any $\hat{\sigma} \geq 0$. In the rest of our analysis, we assume the availability of a subroutine which, for given (λ, x) and $\hat{\sigma} > 0$, returns a $\hat{\sigma}$ -approximate solution (y, u, ϵ) . Let us call this subroutine **ATS** (Approximate Tensor Subroutine). Different from [27], where a similar subproblem as (2.5) without a possible nonsmooth function $h(\cdot)$ and regularization term $\frac{1}{2\lambda}\|y - x\|^2$ is exactly solved, we only assume an approximate solution in the form of (2.7) is available and no further assumption on $h(\cdot)$ is required. Note that the possibly nonsmooth function $h(\cdot)$ can be viewed as a fixed parameter in **ATS**. Once $h(\cdot)$ is given, **ATS** could be called in each step of the bisection search, which itself is a subroutine in the main procedure of our algorithm.

3 The Optimal Tensor Method

3.1 The tensor algorithm and its iteration complexity

Our bid to the optimal tensor algorithm is based on the so-called *Accelerated Hybrid Proximal Extragradient* (A-HPE) framework proposed by Monteiro and Svaiter [24] for problem (1.1), whose main steps can be schematically sketched below:

Algorithm 1 A-HPE framework

STEP 1. Let $x_0, y_0 \in \mathbb{R}^n$, $0 < \sigma < 1$ be given, and set $A_0 = 0$ and $k = 0$.

STEP 2. If $0 \in \partial F(y_k)$, then **STOP**.

STEP 3. Otherwise, find $\lambda_{k+1} > 0$ and a triplet $(\tilde{y}_{k+1}, v_{k+1}, \epsilon_{k+1})$ such that

$$v_{k+1} \in \partial_{\epsilon_{k+1}} F(\tilde{y}_{k+1}), \quad (3.1)$$

$$\|\lambda_{k+1} v_{k+1} + \tilde{y}_{k+1} - \tilde{x}_k\|^2 + 2\lambda_{k+1} \epsilon_{k+1} \leq \sigma^2 \|\tilde{y}_{k+1} - \tilde{x}_k\|^2 \quad (3.2)$$

where

$$\begin{aligned} \tilde{x}_k &= \frac{A_k}{A_k + a_{k+1}} y_k + \frac{a_{k+1}}{A_k + a_{k+1}} x_k, \\ a_{k+1} &= \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}}{2}. \end{aligned}$$

STEP 4. Choose y_{k+1} such that $F(y_{k+1}) \leq F(\tilde{y}_{k+1})$ and let

$$\begin{aligned} A_{k+1} &= A_k + a_{k+1}, \\ x_{k+1} &= x_k - a_{k+1} v_{k+1}. \end{aligned}$$

STEP 5. Set $k \leftarrow k + 1$, and go to STEP 2.

Note that in STEP 2, the stopping condition is $0 \in \partial F(y_k)$. However, in practice, the condition is replaced by an approximate version of it (see Algorithm 2 below). In the following, we quote some technical results derived in [24] for A-HPE. Since our proposed algorithm is within that framework, the results in Lemma 3.1 hold true for our method as well, and they will be used in the subsequent analysis.

Lemma 3.1 *Suppose the sequence $\{x_k, y_k, \tilde{x}_k, \tilde{y}_k\}$ is generated from Algorithm 1. Let x_* be the projection of x_0 onto the set of optimal value points X_* , F_* be the optimal value, and D be the distance from x_0 to X_* . Then for any integer $k \geq 1$, it holds that (Theorem 3.6 in [24]),*

$$\frac{1}{2} \|x_* - x_k\|^2 + A_k (F(y_k) - F_*) + \frac{1 - \sigma^2}{2} \sum_{j=1}^k \frac{A_j}{\lambda_j} \|\tilde{y}_j - \tilde{x}_{j-1}\|^2 \leq \frac{1}{2} D^2. \quad (3.3)$$

Therefore,

$$\sum_{j=1}^k \frac{A_j}{\lambda_j} \|\tilde{y}_j - \tilde{x}_{j-1}\|^2 \leq \frac{D^2}{1 - \sigma^2}. \quad (3.4)$$

Furthermore, A_k and λ_k has the following relation (Lemma 3.7 in [24])

$$A_k \geq \frac{1}{4} \left(\sum_{j=1}^k \sqrt{\lambda_j} \right)^2. \quad (3.5)$$

If y_k is chosen as \tilde{y}_k for all k , the distance between y_k and x_* can be bounded as follows (Theorem 3.10 in [24]),

$$\|y_k - x_*\| \leq \left(\frac{2}{\sqrt{1 - \sigma^2}} + 1 \right) D. \quad (3.6)$$

Now we are ready to propose our optimal tensor method in Algorithm 2.

Algorithm 2 The optimal tensor method

STEP 1. Let $x_0 = y_0 \in \mathbb{R}^n$, $v_0 \in \partial f(x_0)$, $\epsilon_0 = 0$, $k = 0$ and set $0 < \bar{\epsilon}, \bar{\rho} < 1$, $M \geq L_d$. Let $\hat{\sigma} \geq 0$, $0 < \sigma_l < \sigma_u < 1$ such that $\sigma := \hat{\sigma} + \sigma_u < 1$ and $\sigma_l(1 + \hat{\sigma})^{d-1} < \sigma_u(1 - \hat{\sigma})^{d-1}$.

STEP 2. If $\|v_k\| \leq \bar{\rho}$ and $\epsilon_k \leq \bar{\epsilon}$, then **STOP**. Else, go to STEP 3.

STEP 3. Find λ_{k+1} and a $\hat{\sigma}$ -approximate solution

$(y_{k+1}, u_{k+1}, \epsilon_{k+1}) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_+$ of (2.5) at $(\lambda_{k+1}, \tilde{x}_k)$ such that either

$$\frac{d! \sigma_l}{L_d + M} \leq \lambda_{k+1} \|y_{k+1} - \tilde{x}_k\|^{d-1} \leq \frac{d! \sigma_u}{L_d + M} \quad (3.7)$$

or $\|\nabla f(y_{k+1}) + u_{k+1} - \nabla f_{\tilde{x}_k}(y_{k+1})\| \leq \bar{\rho}$ and $\epsilon_{k+1} \leq \bar{\epsilon}$ hold, where

$$\tilde{x}_k = \frac{A_k}{A_k + a_{k+1}} y_k + \frac{a_{k+1}}{A_k + a_{k+1}} x_k \quad (3.8)$$

and

$$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1}A_k}}{2}. \quad (3.9)$$

(Note that λ_{k+1} appears in both (3.7) and (3.9), and seeking a proper λ_{k+1} requires a bisection procedure, to be called Algorithm 3 in Section 4.)

STEP 4. Let

$$\begin{aligned} v_{k+1} &= \nabla f(y_{k+1}) + u_{k+1} - \nabla f_{\tilde{x}_k}(y_{k+1}), \\ A_{k+1} &= A_k + a_{k+1}, \\ x_{k+1} &= x_k - a_{k+1}v_{k+1}. \end{aligned} \quad (3.10)$$

Set $k \leftarrow k + 1$ and go to STEP 2.

At this point, neither Algorithm 1 nor Algorithm 2 has been shown to be implementable. In fact, STEP 3 in both algorithms presented above remain unspecified. Since λ_{k+1} appears in both (3.7) and (3.9), it is even unclear why such solutions as required by STEP 3 exist at all. Actually, the double roles played by λ_{k+1} in (3.7) and (3.9) are crucial for the overall $O(1/k^{(3d+1)/2})$ convergence rate. As a tradeoff, such λ_{k+1} is not easy to find. In Section 4, we shall discuss a *practical* method to find a proper λ_{k+1} (and thus establish a practical implementation of STEP 3 in Algorithm 2) via the Approximate Tensor Subroutine (**ATS**) in combination with a line-search subroutine.

First, let us remark that Algorithm 2 is indeed a specialization of A-HPE. For simplicity, we let $y_{k+1} =$

\tilde{y}_{k+1} in STEP 4 of Algorithm 1. Because $(y_{k+1}, u_{k+1}, \epsilon_{k+1})$ is a $\hat{\sigma}$ -approximate solution at $(\lambda_{k+1}, \tilde{x}_k)$, one has that $u_{k+1} \in (\nabla f_{\tilde{x}_k} + \partial_{\epsilon_{k+1}} h)(y_{k+1})$, and so we have

$$\begin{aligned} v_{k+1} &\in \nabla f(y_{k+1}) - \nabla f_{\tilde{x}_k}(y_{k+1}) + (\nabla f_{\tilde{x}_k} + \partial_{\epsilon_{k+1}} h)(y_{k+1}) \\ &= \nabla f(y_{k+1}) + \partial_{\epsilon_{k+1}} h(y_{k+1}) \subseteq \partial_{\epsilon_{k+1}}(f + h)(y_{k+1}) \end{aligned}$$

which satisfies (3.1). To establish (3.2), we need the following proposition.

Proposition 3.2 *Let (y, u, ϵ) be a $\hat{\sigma}$ -approximate solution of (2.5) at (λ, \tilde{x}) such that (3.7) holds. Define $v := \nabla f(y) + u - \nabla f_{\tilde{x}}(y)$. Then,*

$$\|\lambda v + y - \tilde{x}\|^2 + 2\lambda\epsilon \leq \left(\hat{\sigma} + \lambda \frac{L_d + M}{d!} \|y - \tilde{x}\|^{d-1} \right)^2 \|y - \tilde{x}\|^2. \quad (3.11)$$

Consequently,

$$\|\lambda v + y - \tilde{x}\|^2 + 2\lambda\epsilon \leq \sigma^2 \|y - \tilde{x}\|^2 \quad \text{with} \quad \sigma = \sigma_u + \hat{\sigma}, \quad (3.12)$$

where σ_u is a input paramter in Algorithm 2 and also appears in (3.7).

Proof. First of all, according to Lemma 2.1, it follows that

$$\lambda \|u - v\| = \lambda \|\nabla f(y) - \nabla f_{\tilde{x}}(y)\| \leq \lambda \frac{L_d + M}{d!} \|y - \tilde{x}\|^d.$$

Combining the above inequality with (2.7), one has that

$$\begin{aligned} &\|\lambda v + y - \tilde{x}\|^2 + 2\lambda\epsilon \\ &\leq (\|\lambda u + y - \tilde{x}\| + \lambda \|u - v\|)^2 + 2\lambda\epsilon \\ &= (\|\lambda u + y - \tilde{x}\|^2 + 2\lambda\epsilon) + 2\lambda \|u - v\| \|\lambda u + y - \tilde{x}\| + \lambda^2 \|u - v\|^2 \\ &\leq \hat{\sigma}^2 \|y - \tilde{x}\|^2 + 2 \left(\lambda \frac{L_d + M}{d!} \|y - \tilde{x}\|^d \right) \hat{\sigma} \|y - \tilde{x}\| + \left(\lambda \frac{L_d + M}{d!} \|y - \tilde{x}\|^d \right)^2 \\ &= \left(\hat{\sigma} + \lambda \frac{L_d + M}{d!} \|y - \tilde{x}\|^{d-1} \right)^2 \|y - \tilde{x}\|^2, \end{aligned}$$

proving the first inequality. Then, by the right hand side of (3.7), $\lambda \frac{L_d + M}{d!} \|y - \tilde{x}\|^{d-1} \leq \sigma_u$, and so the second inequality follows. \square

We summarize the above discussion in the theorem below.

Theorem 3.3 *Algorithm 2 is a manifestation of the A-HPE framework, and thus the results of Lemma 3.1 hold for the sequence generated by Algorithm 2.*

Before addressing the implementation of STEP 3 in Algorithm 2, let us first present the overall iteration complexity of Algorithm 2, assuming STEP 3 could be implemented.

Theorem 3.4 *Let D be the distance of x_0 to X_* . Then, for any integer $k \geq 1$, the iterate y_k generated by Algorithm 2 satisfies:*

$$F(y_k) - F_* \leq \left(\frac{d+1}{2}\right)^{\frac{3d+1}{2}} \frac{2^d}{(1 - (\hat{\sigma} + \sigma_u)^2)^{\frac{d-1}{2}} d! \sigma_l} D^{d+1} (L_d + M) k^{-\frac{3d+1}{2}}.$$

The above theorem establishes the $O(1/k^{\frac{3d+1}{2}})$ iteration complexity for Algorithm 2. Since Algorithm 2 falls into the category of the High-Order Oracle Model, whose iteration complexity has a lower bound of $O(1/k^{\frac{3d+1}{2}})$; see Arjevanim, Shamir and Shiff [31] and Nesterov [27]. The worst-case iteration complexity of Algorithm 2 matches this lower bound and it is therefore an *optimal* method.

3.2 Proof of Theorem 3.4

We first provide a recursive bound on A_k as an intermediate step.

Proposition 3.5 *Let D be the distance of x_0 to X_* . Suppose $\{A_k\}_{k=1}^\infty$ is generated from Algorithm 2, then*

$$A_k \geq \frac{1}{4} C^{-\frac{2p}{q}} \left(\sum_{j=1}^k A_j^{\frac{1}{q}} \right)^{2p} \quad (3.13)$$

where $q = \frac{3d+1}{d-1}$, $p = \frac{3d+1}{2d+2}$ and $C = \frac{D^2}{(1 - (\hat{\sigma} + \sigma_u)^2)} \left(\frac{d! \sigma_l}{L_d + M} \right)^{-\frac{2}{d-1}}$.

Proof. Suppose $\{x_k, y_k, \tilde{x}_k\}$ is the sequence generated by Algorithm 2. Then, according to (3.4) and Proposition 3.2, it holds that

$$\sum_{j=1}^k \frac{A_j}{\lambda_j} \|y_j - \tilde{x}_{j-1}\|^2 \leq \frac{D^2}{1 - (\hat{\sigma} + \sigma_u)^2},$$

which together with the left hand side of (3.7) implies

$$\begin{aligned} \sum_{j=1}^k \frac{A_j}{\lambda_j^{\frac{d+1}{d-1}}} &= \sum_{j=1}^k \frac{A_j \|y_j - \tilde{x}_{j-1}\|^2}{\lambda_j} \cdot \frac{1}{\lambda_j^{\frac{2}{d-1}} \|y_j - \tilde{x}_{j-1}\|^2} \\ &\leq \frac{D^2}{(1 - (\hat{\sigma} + \sigma_u)^2)} \left(\frac{d! \sigma_l}{L_d + M} \right)^{-\frac{2}{d-1}} = C. \end{aligned} \quad (3.14)$$

By the definition of p and q , we have $\frac{1}{p} + \frac{1}{q} = 1$. Using Hölder's inequality, together with (3.14), we have

$$\left(\sum_{j=1}^k \sqrt{\lambda_j} \right)^{\frac{1}{p}} C^{\frac{1}{q}} \geq \left(\sum_{j=1}^k \sqrt{\lambda_j} \right)^{\frac{1}{p}} \left(\sum_{j=1}^k \frac{A_j}{\lambda_j^{\frac{d+1}{d-1}}} \right)^{\frac{1}{q}} \geq \sum_{j=1}^k \lambda_j^{\frac{1}{2p}} \frac{A_j^{\frac{1}{q}}}{\lambda_j^{\frac{q(d+1)}{q(d-1)}}} = \sum_{j=1}^k A_j^{\frac{1}{q}}.$$

Finally, by (3.5) we obtain

$$A_k \geq \frac{1}{4} \left(\sum_{j=1}^k \sqrt{\lambda_j} \right)^2 \geq \frac{1}{4} C^{-\frac{2p}{q}} \left[\sum_{j=1}^k A_j^{\frac{1}{q}} \right]^{2p}.$$

□

Proof of Theorem 3.4. Let p , q and C be defined as in Proposition 3.5. Construct $\{B_k\}$ such that $B_1 = A_1$ and $B_i = T^{\frac{1-(2p/q)^{i-1}}{1-2p/q}} (A_1)^{(2p/q)^{i-1}}$ for $i \geq 2$, where $T := \frac{1}{4} \left(\frac{1}{C}\right)^{\frac{2p}{q}} \left(\frac{2}{d+1}\right)^{2p}$. Next, we shall apply induction to show that for any $k \geq 1$,

$$A_k \geq B_i k^{r_i}, \quad \forall i \geq 1, \quad (3.15)$$

where $r_i = \frac{3d+1}{2} [1 - (2p/q)^{i-1}]$. When $i = 1$, this is obvious because $A_k \geq A_1 = B_1 k^{r_1}$. Now suppose that for any $k \geq 1$, $A_k \geq B_i k^{r_i}$ for some i . Then, by the induction hypothesis and (3.13) it holds that

$$\begin{aligned} A_k &\geq \frac{1}{4} C^{-\frac{2p}{q}} \left(\sum_{j=1}^k A_j^{\frac{1}{q}} \right)^{2p} \geq \frac{1}{4} C^{-\frac{2p}{q}} \left(\sum_{j=1}^k (B_i j^{r_i})^{\frac{1}{q}} \right)^{2p} \\ &= \frac{1}{4} \left(\frac{B_i}{C} \right)^{\frac{2p}{q}} \left(\sum_{j=1}^k j^{\frac{r_i}{q}} \right)^{2p} \geq \frac{1}{4} \left(\frac{B_i}{C} \right)^{\frac{2p}{q}} \left(\int_0^k x^{\frac{r_i}{q}} dx \right)^{2p} \\ &= \frac{1}{4} \left(\frac{B_i}{C} \right)^{\frac{2p}{q}} \left(\frac{1}{1+r_i/q} k^{\frac{r_i}{q}+1} \right)^{2p} = \frac{1}{4} \left(\frac{B_i}{C} \right)^{\frac{2p}{q}} \left(\frac{q}{q+r_i} \right)^{2p} k^{2p(\frac{r_i}{q}+1)} \\ &\geq \frac{1}{4} \left(\frac{B_i}{C} \right)^{\frac{2p}{q}} \left(\frac{2}{d+1} \right)^{2p} k^{2p(\frac{r_i}{q}+1)}, \end{aligned} \quad (3.16)$$

where the last inequality follows from

$$\frac{q}{q+r_i} = \frac{\frac{3d+1}{d-1}}{\frac{3d+1}{d-1} + \frac{3d+1}{2} [1 - (2p/q)^{i-1}]} = \frac{1}{1 + \frac{d-1}{2} [1 - (2p/q)^{i-1}]} \geq \frac{1}{1 + \frac{d-1}{2}} = \frac{2}{d+1}.$$

Let us further simplify the expression. First of all, from the definition of T and B_i , one observes that

$$\begin{aligned} \frac{1}{4} \left(\frac{B_i}{C} \right)^{\frac{2p}{q}} \left(\frac{2}{d+1} \right)^{2p} &= B_i^{\frac{2p}{q}} T = \left[T^{\frac{1-(2p/q)^{i-1}}{1-2p/q}} A_1^{(2p/q)^{i-1}} \right]^{\frac{2p}{q}} T \\ &= T^{\frac{2p/q - (2p/q)^i}{1-2p/q} + 1} A_1^{(2p/q)^i} \\ &= T^{\frac{1-(2p/q)^i}{1-2p/q}} A_1^{(2p/q)^i} = B_{i+1}. \end{aligned} \quad (3.17)$$

Then, the construction of q and r_i implies that

$$\begin{aligned}
2p \left(\frac{r_i}{q} + 1 \right) &= \frac{3d+1}{d+1} \left(1 + \frac{\frac{3d+1}{2}(1 - (2p/q)^{i-1})}{\frac{3d+1}{d-1}} \right) \\
&= \frac{3d+1}{d+1} \left(1 + \frac{d-1}{2} (1 - (2p/q)^{i-1}) \right) \\
&= \frac{3d+1}{d+1} \left(\frac{d+1}{2} - \frac{d-1}{2} (2p/q)^{i-1} \right) \\
&= \frac{3d+1}{2} (1 - (2p/q)^i) = r_{i+1},
\end{aligned} \tag{3.18}$$

where the second last equality holds true due to the fact that $2p/q = (d-1)/(d+1)$. Now the desired inequality (3.15) follows by combining (3.16), (3.17) and (3.18). Observe that $2p/q = (d-1)/(d+1) < 1$, and so $\lim_{i \rightarrow \infty} B_i = T^{\frac{1}{1-2p/q}} = T^{\frac{d+1}{2}}$ and $\lim_{i \rightarrow \infty} r_i = \frac{3d+1}{2}$. Finally, by letting $i \rightarrow \infty$ in (3.15) and using the definition of C in (3.14), we have

$$\begin{aligned}
A_k \geq T^{\frac{d+1}{2}} k^{\frac{3d+1}{2}} &= \left[\frac{1}{4} \left(\frac{1}{C} \right)^{\frac{d-1}{d+1}} \left(\frac{2}{d+1} \right)^{\frac{3d+1}{d+1}} \right]^{\frac{d+1}{2}} k^{\frac{3d+1}{2}} \\
&= \left(\frac{1}{2} \right)^{d+1} \frac{d! \sigma_l}{L_d + M} \left(\frac{1 - \sigma^2}{D^2} \right)^{\frac{d-1}{2}} \left(\frac{2}{d+1} \right)^{\frac{3d+1}{2}} k^{\frac{3d+1}{2}}.
\end{aligned}$$

Combining it with (3.3), we have

$$F(y_k) - F_* \leq \frac{1}{2A_k} D^2 \leq \left(\frac{d+1}{2} \right)^{\frac{3d+1}{2}} \frac{2^d}{(1 - (\hat{\sigma} + \sigma_u)^2)^{\frac{d-1}{2}} d! \sigma_l} D^{d+1} (L_d + M) k^{-\frac{3d+1}{2}}.$$

□

3.3 Comparison with Nesterov's Accelerated Tensor Method

In Nesterov's accelerated tensor method [27], an auxiliary function

$$\psi_k(x) = l_k(x) + M \|x - x_0\|^{d+1} \tag{3.19}$$

with l_k being some linear function, is constructed to satisfy

$$\begin{aligned}
\mathcal{R}_k^1 &: \beta_k := \min_x \psi_k(x) - A_k F(y_k) \geq 0, \\
\mathcal{R}_k^2 &: \psi_k(x) \leq A_k F(x) + M \|x - x_0\|^{d+1}, \quad \forall x \in \mathbb{R}^n
\end{aligned}$$

where $A_k = \Theta(k^{d+1})$. In fact, the function $\psi_k(x)$ serves as a bridge to guarantee the following relation:

$$A_k F(y_k) \leq \min_x \psi_k(x) \leq \psi_k(x_*) \leq A_k F_* + M \|x_* - x_0\|^{d+1}. \tag{3.20}$$

As a result, $F(y_k) - F_* \leq \frac{M}{A_k} \|x_* - x_0\|^{d+1}$ yielding the iteration complexity of $O(1/k^{d+1})$.

In the implementation of high-order A-HPE framework, it is crucial to ensure that condition (3.7) is satisfied. In the remainder of the paper, we shall focus on how to satisfy (3.7) in STEP 3 of Algorithm 2. Our bid is to use bisection on a parameter λ (to be introduced later), while calling an Approximate Tensor Subroutine (**ATS**). Observe that \tilde{x} , which is the point to define $f_{\tilde{x}}(y)$ in (2.4) to approximate the smooth function $f(y)$, is indeed heavily dependent on λ . In other words, we need to search for the point where the Taylor expansion (2.4) is to be computed. This is a key difference between the A-HPE framework and Nesterov's approach [27]. Once condition (3.7) is satisfied, then inequality (3.3) would follow, which leads to the following tighter estimation than (3.20):

$$A_k F(y_k) + \beta_k \leq A_k F_* + \frac{1}{2} \|x_* - x_0\|^2,$$

as $\beta_k = \frac{1-\sigma^2}{2} \sum_{j=1}^k \frac{A_j}{\lambda_j} \|\tilde{y}_j - \tilde{x}_{j-1}\|^2 \geq 0$ is totally missing in (3.20). The above inequality also gives an upper bound on β_k . Together with the lower bound (3.6) this gives a better lower bound on A_k , namely $A_k \geq O(k^{\frac{3d+1}{2}})$, which leads to the optimal iteration complexity presented in Theorem 3.4.

4 A Line Search Subroutine and Its Iteration Complexity

After establishing the overall iteration complexity for Algorithm 2, it remains to find a way to implement STEP 3 of the algorithm. In this section we discuss how this can be done, from a special case to the general one. The idea is better illustrated by considering the special case. Finally, for the general composite objective function, assuming the tensor proximal mapping regarding $h(x)$ is possible, our approach is based on a line-search procedure for the point on which the Taylor expansion is computed.

4.1 The Non-Composite Case

Let us first consider a special case for Algorithm 2 where $F(x) = f(x)$ in the objective function and y_{k+1} is the exact solution of the following convex tensor proximal point problem:

$$\min_y f_{\tilde{x}_k}(y) + \frac{1}{2\lambda_{k+1}} \|y - \tilde{x}_k\|^2.$$

We shall discuss how to find λ_{k+1} to satisfy the alternative condition in STEP 3 of Algorithm 2.

Note that for fixed x_k and y_k , \tilde{x}_k and y_{k+1} are uniquely determined by λ_{k+1} . Therefore the functions $\tilde{x}_k(\lambda)$ and $y_{k+1}(\lambda)$ are continuous with respect to λ (where we denote λ_{k+1} to be λ). Next, we show that:

- (i) $\lambda \|y_{k+1}(\lambda) - \tilde{x}_k(\lambda)\|^{d-1} \rightarrow 0$, as $\lambda \rightarrow 0$;
- (ii) Either there exists an increasing sub-sequence $\lambda_j \uparrow \infty$, such that $\lambda_j \|y_{k+1}(\lambda_j) - \tilde{x}_k(\lambda_j)\|^{d-1} \rightarrow \infty$ as $j \rightarrow \infty$, or there exists $\hat{\lambda}$ such that $\|\nabla f(y_{k+1}(\lambda))\| \leq \bar{\rho}$ for any $\lambda \geq \hat{\lambda}$.

Observe that

$$\begin{aligned}
& f_{\tilde{x}_k(\lambda)}(y_{k+1}(\lambda)) + \frac{1}{2\lambda} \|y_{k+1}(\lambda) - \tilde{x}_k(\lambda)\|^2 \\
&= \min_y f_{\tilde{x}_k(\lambda)}(y) + \frac{1}{2\lambda} \|y - \tilde{x}_k(\lambda)\|^2 \\
&\leq f_{\tilde{x}_k(\lambda)}(\tilde{x}_k(\lambda)) \\
&= f(\tilde{x}_k(\lambda)) < \infty, \forall \lambda > 0
\end{aligned}$$

where $f(\tilde{x}_k(\lambda))$ is bounded, since $\tilde{x}_k(\lambda)$ is a convex combination of x_k and y_k . Letting $\lambda \rightarrow 0$ in the above inequality leads to $\|y_{k+1}(\lambda) - \tilde{x}_k(\lambda)\|^2 \rightarrow 0$, which implies $\lambda \|y_{k+1}(\lambda) - \tilde{x}_k(\lambda)\|^{d-1} \rightarrow 0$ as $\lambda \rightarrow 0$, proving **(i)**.

To prove **(ii)**, it suffices to show that if the “either” part does not hold, then the “or” part must hold. In this case, there must exist $C_1 > 0$ such that when $\lambda \rightarrow \infty$, $\lambda \|y_{k+1}(\lambda) - \tilde{x}_k(\lambda)\|^{d-1} \leq C_1$, and thus $\|y_{k+1}(\lambda) - \tilde{x}_k(\lambda)\| \rightarrow 0$. Moreover, for any $\lambda > 0$ the optimality condition is

$$\nabla f_{\tilde{x}_k(\lambda)}(y_{k+1}(\lambda)) + \frac{1}{\lambda} (y_{k+1}(\lambda) - \tilde{x}_k(\lambda)) = 0.$$

Letting $\lambda \rightarrow \infty$ in the above identity yields that $\nabla f_{\tilde{x}_k(\lambda)}(y_{k+1}(\lambda)) \rightarrow 0$. Recall that in this case we have $\|y_{k+1}(\lambda) - \tilde{x}_k(\lambda)\| \rightarrow 0$, thus $\nabla f(y_{k+1}(\lambda)) \rightarrow 0$ proving the “or” part.

To summarize, either we have $\lambda \|y_{k+1}(\lambda) - \tilde{x}_k(\lambda)\|^{d-1} \rightarrow 0$ as $\lambda \rightarrow 0$ and $\lambda_j \|y_{k+1}(\lambda_j) - \tilde{x}_k(\lambda_j)\|^{d-1} \rightarrow \infty$ as $j \rightarrow \infty$, which guarantees the existence of λ to satisfy (3.7) due to the continuity of $\lambda \|y_{k+1}(\lambda) - \tilde{x}_k(\lambda)\|^{d-1}$ on λ . Or we have a λ_{k+1} such that $\|\nabla f(y_{k+1}(\lambda))\| \leq \bar{\rho}$. In this case, since $h(x)$ is not present, $u_{k+1} = \nabla f_{\tilde{x}_k}(y_{k+1})$ and $\|\nabla f(y_{k+1}) + u_{k+1} - \nabla f_{\tilde{x}_k}(y_{k+1})\| = \|\nabla f(y_{k+1})\| \leq \bar{\rho}$. Therefore, we have shown that the alternative condition in STEP 3 is actually satisfied.

4.2 A Bisection Subroutine

To present the algorithm that computes λ satisfying the conditions in STEP 3, we first construct $\beta_{k+1} = \frac{a_{k+1}}{A_k + a_{k+1}}$. From (3.9), we can see that $\lambda_{k+1} = \frac{a_{k+1}^2}{A_k + a_{k+1}}$. Therefore, we are able to represent λ_{k+1} and \tilde{x}_k by means of β_{k+1} :

$$\begin{cases} \lambda_{k+1} &= A_k \frac{\beta_{k+1}^2}{1 - \beta_{k+1}}, \\ \tilde{x}_k &= \beta_{k+1} x_k + (1 - \beta_{k+1}) y_k. \end{cases}$$

In the k -th iteration, we denote

$$\lambda(\beta) = A_k \frac{\beta^2}{1 - \beta}, \quad \beta \in (0, 1). \quad (4.1)$$

Its inverse on the domain $\lambda > 0$ is

$$\beta(\lambda) = \frac{\sqrt{\lambda^2 + 4\lambda A_k} - \lambda}{2A_k},$$

which is monotonically increasing.

We shall perform bisection on β instead of λ in STEP 3 of Algorithm 2 to search for λ_{k+1} . In that way, the initial interval for the bisection is $[0, 1]$. (Monteiro and Svaiter [24] presented a bisection process for their A-HPE algorithm too. However, we can skip what they called the bracketing stage in [24]).

Algorithm 3 Bisection on β based on the subroutine **ATS**

INPUT: $M \geq L_d, \hat{\sigma} \geq 0, 0 < \sigma_l < \sigma_u < 1$ such that $\sigma := \hat{\sigma} + \sigma_u < 1$ and $\sigma_l(1 + \hat{\sigma})^{d-1} < \sigma_u(1 - \hat{\sigma})^{d-1}$, tolerance $\bar{\rho} > 0$ and $\bar{\epsilon} > 0$.

STEP 1. Let $\alpha_+ = \frac{d! \sigma_u}{L_d + M}$ and $\alpha_- = \frac{d! \sigma_l}{L_d + M}$.

STEP 2. (Bisection Setup) Set $\beta_- = 0, \beta_+ = 1, \lambda_+ = \lambda(\beta_+) = +\infty, \lambda_- = \lambda(\beta_-)$.

2.a. Let $\beta = \frac{\beta_- + \beta_+}{2}$ and let

$$\lambda_\beta = \lambda(\beta), \quad x_\beta = (1 - \beta)y_k + \beta x_k, \quad (4.2)$$

and use **ATS** to compute $(y_\beta, u_\beta, \epsilon_\beta)$ as a $\hat{\sigma}$ -approximate solution at (λ_β, x_β) , and $v_\beta = \nabla f(y_\beta) - \nabla f_{x_\beta}(y_\beta) + u_\beta$.

2.b.

if $\|v_\beta\| \leq \bar{\rho}$ and $\epsilon_\beta \leq \bar{\epsilon}$ **then**

output $(\lambda_\beta, x_\beta, y_\beta, u_\beta, \epsilon_\beta)$ and **STOP**.

else if $\lambda_\beta \|y_\beta - x_\beta\|^{d-1} \in [\alpha_-, \alpha_+]$ **then**

set $(\beta_{k+1}, \tilde{x}_k, y_{k+1}, v_{k+1}) = (\beta, x_\beta, y_\beta, v_\beta)$ and **STOP**.

else if $\lambda_\beta \|y_\beta - x_\beta\|^{d-1} > \alpha_+$ **then**

set $\beta_+ \leftarrow \beta$, and go to STEP 2.a.

else if $\lambda_\beta \|y_\beta - x_\beta\|^{d-1} < \alpha_-$ **then**

set $\beta_- \leftarrow \beta$, and go to STEP 2.a.

end if

We remark that the conditions on $\bar{\rho}$ and $\bar{\epsilon}$ are only used in the final stage of the algorithm to decide the point that is close to optimum. In the implementation, it is reasonable to set a lower precision at the beginning stage of the algorithm. Now an upper bound for the overall number of iterations required by Algorithm 3 is presented in the following theorem, whose proof will be postponed to the subsequent section.

Theorem 4.1 *Algorithm 3 needs to perform no more than*

$$\Theta \left(\max\{\log_2(\bar{\epsilon}^{-1}), \log_2(\bar{\rho}^{-1})\} \right) \quad (4.3)$$

bisection steps before reaching $\lambda_{k+1} > 0$ and a $\hat{\sigma}$ -approximate solution $(y_{k+1}, u_{k+1}, \epsilon_{k+1})$ at $(\lambda_{k+1}, \tilde{x}_k(\lambda_{k+1}))$ satisfying

$$\alpha_- \leq \lambda_{k+1} \|\tilde{x}_k(\lambda_{k+1}) - y_{k+1}\|^{d-1} \leq \alpha_+,$$

or to return v_{k+1} and ϵ_{k+1} such that $\|v_{k+1}\| \leq \bar{\rho}$ and $\epsilon_{k+1} \leq \bar{\epsilon}$.

4.3 The Iteration Complexity Analysis

In this subsection, we establish the iteration bound of Algorithm 3 and give a proof for Theorem 4.1. First, we review some facts for maximal monotone operator. For a point-to-set operator $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$,

its graph is defined as:

$$\text{Gr}(T) = \{(z, v) \in \mathbb{R}^n \times \mathbb{R}^n \mid v \in T(z)\},$$

and the operator T is called *monotone* if

$$\langle v - \tilde{v}, z - \tilde{z} \rangle \geq 0 \quad \forall (z, v), (\tilde{z}, \tilde{v}) \in \text{Gr}(T),$$

and T is *maximal monotone* if it is monotone and maximal in the family of monotone operators with respect to the partial order of inclusion. Given a maximal monotone operator $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ and a scalar ϵ , the associated ϵ -enlargement $T^\epsilon : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is defined as:

$$T^\epsilon(z) = \{v \in \mathbb{R}^n \mid \langle z - \tilde{z}, v - \tilde{v} \rangle \geq -\epsilon, \forall \tilde{z} \in \mathbb{R}^n, \tilde{v} \in T(\tilde{z})\}, \quad \forall z \in \mathbb{R}^n.$$

For a convex function f , its subdifferential ∂f is monotone if f is a proper function. If f is a proper lower semicontinuous convex function, then ∂f is maximal monotone [28].

Recall that the optimality condition of subproblem (2.5) is characterized by (2.6), which is:

$$0 \in \lambda(\nabla f_x + \partial h)(y) + y - x = (\lambda(\nabla f_x + \partial h) + I)(y) - x.$$

Furthermore, x is optimal to (1.1) if and only if $y = x$. Therefore, it is natural to consider the residual

$$\varphi(\lambda; x) := \lambda \left\| (I + \lambda(\nabla f_x + \partial h))^{-1}(x) - x \right\|$$

for any $\lambda > 0, x \in \mathbb{R}^n$. The above residual was adopted in [24] for the quadratic subproblem. In this paper, to accommodate the high-order information, we consider the following modified residual:

$$\psi(\lambda; x) := \lambda \left\| (I + \lambda(\nabla f_x + \partial h))^{-1}(x) - x \right\|^{d-1}.$$

We have an immediate property regarding $\psi(\cdot)$.

Proposition 4.2 *Let $x \in \mathbb{R}^n, \lambda > 0$ and $\hat{\sigma} \geq 0$. If (y, u, ϵ) is a $\hat{\sigma}$ -approximate solution of (2.5) at (λ, x) , then*

$$\lambda(1 - \hat{\sigma})^{d-1} \|y - x\|^{d-1} \leq \psi(\lambda; x) \leq \lambda(1 + \hat{\sigma})^{d-1} \|y - x\|^{d-1}. \quad (4.4)$$

Proof. From proposition 7.3 in [24], it holds that

$$(\lambda(1 - \hat{\sigma}) \|y - x\|)^{d-1} \leq \varphi^{d-1}(\lambda; x) \leq (\lambda(1 + \hat{\sigma}) \|y - x\|)^{d-1}. \quad (4.5)$$

Notice $\varphi^{d-1}(\lambda; x) = \lambda^{d-2} \psi(\lambda; x)$, and so (4.4) readily follows by combining the above inequalities and identity. \square

Lemma 4.3 *Let scalars $\bar{\rho} > 0, \bar{\epsilon} > 0, \hat{\sigma} \geq 0$ and $\alpha > 0$ be given and satisfy $\hat{\sigma} + \frac{L_d + M}{d!} \alpha := \sigma < 1$. Suppose*

$$\lambda \geq \max \left\{ \alpha^{1/d} \left[\frac{1}{\bar{\rho}} \left(1 + \hat{\sigma} + \frac{L_d + M}{d!} \alpha \right) \right]^{1 - \frac{1}{d}}, \left(\frac{\sigma^2 \alpha^{\frac{2}{d-1}}}{2\bar{\epsilon}} \right)^{\frac{d-1}{d+1}} \right\}, \quad (4.6)$$

*and (y, u, ϵ) is a $\hat{\sigma}$ -approximate solution of (2.5) at (λ, x) for some vector $x \in \mathbb{R}^n$. Then, one of the following holds: either **(a)** $\lambda \|y - x\|^{d-1} > \alpha$; or **(b)** the vector $v := \nabla f(y) - \nabla f_x(y) + u$ satisfies*

$$v \in (\nabla f + (\partial h)^\epsilon)(y), \quad \|v\| \leq \bar{\rho}, \quad \epsilon \leq \bar{\epsilon}. \quad (4.7)$$

Proof. Suppose that λ satisfies (4.6) but not **(a)**, namely

$$\lambda\|y - x\|^{d-1} \leq \alpha. \quad (4.8)$$

In that case, recall that ∂h_ϵ is the ϵ -subdifferential of h and $(\partial h)^\epsilon$ is the ϵ -enlargement of operator ∂h . According to Proposition 3 in [9], one has $\partial h_\epsilon(x) \subseteq (\partial h)^\epsilon(x)$ for any $\epsilon \geq 0$ and $x \in \mathbb{R}^n$. Therefore, the inclusion in (4.7) directly follows from Proposition 3.2. Moreover, inequality (3.11) leads to

$$\lambda\|v\| - \|y - x\| \leq \|\lambda v + y - x\| \leq \left(\hat{\sigma} + \lambda \frac{L_d + M}{d!} \|y - x\|^{d-1} \right) \|y - x\|.$$

Together with (4.6) and (4.8), the above inequality yields

$$\begin{aligned} \|v\| &\leq \frac{1}{\lambda} \left(1 + \hat{\sigma} + \frac{L_d + M}{d!} \lambda \|y - x\|^{d-1} \right) \|y - x\| \\ &\leq \frac{1}{\lambda} \left(1 + \hat{\sigma} + \frac{L_d + M}{d!} \alpha \right) \left(\frac{\alpha}{\lambda} \right)^{\frac{1}{d-1}} \\ &\leq \bar{\rho}. \end{aligned}$$

On the other hand, inequality (3.11) also implies that

$$2\lambda\epsilon \leq \left(\hat{\sigma} + \frac{L_d + M}{d!} \lambda \|y - x\|^{d-1} \right)^2 \|y - x\|^2 \leq \left(\hat{\sigma} + \frac{L_d + M}{d!} \alpha \right)^2 \|y - x\|^2 \leq \sigma^2 \|y - x\|^2.$$

Combined with (4.6) and (4.8) this leads to

$$\epsilon \leq \frac{\sigma^2 \|y - x\|^2}{2\lambda} \leq \frac{\sigma^2}{2\lambda} \left(\frac{\alpha}{\lambda} \right)^{\frac{2}{d-1}} \leq \bar{\epsilon}.$$

Hence, **(b)** must hold in this case. □

In the rest of this section, we simply let $\alpha = \alpha_-$ in Lemma 4.3 and denote

$$\bar{\lambda} = \max \left\{ \alpha_-^{1/d} \left[\frac{1}{\bar{\rho}} \left(1 + \hat{\sigma} + \frac{L_d + M}{d!} \alpha_- \right) \right]^{1-\frac{1}{d}}, \left(\frac{\sigma^2 \alpha_-^{\frac{2}{d-1}}}{2\bar{\epsilon}} \right)^{\frac{d-1}{d+1}} \right\}. \quad (4.9)$$

Lemma 4.3 implies that if λ is sufficiently large, then either Algorithm 3 stops because (4.7) is satisfied, or $\lambda\|y - x\|^{d-1} \geq \alpha_-$, which achieves half of the bisection goal. Now we are ready to prove Theorem 4.1.

Proof of Theorem 4.1. Suppose that Algorithm 3 has performed j bisection steps before triggering the stopping criteria. We aim to show $j \leq \Theta(\max\{\log_2(\bar{\epsilon}^{-1}), \log_2(\bar{\rho}^{-1})\})$. At that iteration let us denote $x_+ = x_{\beta_+}$, $x_- = x_{\beta_-}$, $y_+ = y_{\beta_+}$ and $y_- = y_{\beta_-}$, and we also have $\beta_+ - \beta_- = \frac{1}{2^j}$. Denote $\bar{\beta} = \beta(\bar{\lambda})$, where $\bar{\lambda}$ is as defined in (4.9). If $\bar{\beta} \leq \frac{1}{2}$ then $\frac{1}{1-\bar{\beta}} \leq 2$; if $\bar{\beta} > \frac{1}{2}$, then (4.1) gives

$$\frac{1}{1-\bar{\beta}} = \frac{\bar{\lambda}}{A_k \bar{\beta}^2} < \frac{4\bar{\lambda}}{A_k} \leq \max \left\{ \Theta \left((\bar{\rho}^{-1})^{\frac{d-1}{d}} \right), \Theta \left((\bar{\epsilon}^{-1})^{\frac{d-1}{d+1}} \right) \right\}.$$

Therefore, in the rest of the proof we may assume $j \geq \log_2(2/(1-\bar{\beta}))$, for otherwise $j < \log_2(2/(1-\bar{\beta})) \leq \Theta(\max\{\log_2(\bar{\rho}^{-1}), \log_2(\bar{\epsilon}^{-1})\})$ already holds.

Note that the bisection search starts with $\beta_+ = 1$, corresponding to $\lambda_+ = +\infty$ according to (4.1) when β_+ is not updated during the procedure. However, the following lemma tells us that after running Algorithm 3 for a number of iterations, λ_+ will be reduced and upper bounded by some constant depending on $\bar{\epsilon}$ and $\bar{\rho}$.

Lemma 4.4 *Suppose that Algorithm 3 has performed j bisection steps with $j \geq \log_2(2/(1-\bar{\beta}))$, where $\bar{\beta} = \beta(\bar{\lambda})$ and $\bar{\lambda}$ are as defined in (4.9). Then we have*

$$\lambda_+ \leq \max \{9A_k/4, 8\bar{\lambda}\} = \max \left\{ \Theta(\bar{\epsilon}^{-1}), \Theta \left((\bar{\rho}^{-1})^{\frac{d+1}{d}} \right) \right\}. \quad (4.10)$$

We shall continue our discussion without disruption here and leave the proof of Lemma 4.4 to the appendix. Since Algorithm 3 did not stop before iteration j , the bound on β_+ must have been previously updated, and so

$$\lambda_+ \|y_{\beta_+} - x_{\beta_+}\|^{d-1} > \alpha_+, \quad \lambda_- \|y_{\beta_-} - x_{\beta_-}\|^{d-1} < \alpha_-,$$

where λ_+ is upper bounded due to Lemma 4.4.

By Proposition 4.2, we have that

$$\begin{aligned} \psi_+ &:= \psi(\lambda_+; x_+) \geq \lambda_+ (1 - \hat{\sigma})^{d-1} \|y_+ - x_+\|^{d-1} > (1 - \hat{\sigma})^{d-1} \alpha_+, \\ \psi_- &:= \psi(\lambda_-; x_-) \leq \lambda_- (1 + \hat{\sigma})^{d-1} \|y_- - x_-\|^{d-1} < (1 + \hat{\sigma})^{d-1} \alpha_-. \end{aligned}$$

Consequently,

$$\psi_+ - \psi_- > (1 - \hat{\sigma})^{d-1} \alpha_+ - (1 + \hat{\sigma})^{d-1} \alpha_-. \quad (4.11)$$

The parameters α_+ and α_- are pre-specified. Therefore, it suffices to show that $\psi_+ - \psi_-$ is upper bounded by $\beta_+ - \beta_-$ multiplied by some constant factor and hence the number of bisection search j can be bounded as well. To this end, denote

$$\bar{y}_+ = (I + \lambda_+(\nabla f_{x_+} + \partial h))^{-1}(x_+) \quad \text{and} \quad \bar{y}_- = (I + \lambda_-(\nabla f_{x_-} + \partial h))^{-1}(x_-). \quad (4.12)$$

Then, there exist

$$\begin{aligned} \bar{u}_+ &\in (\nabla f_{x_+} + \partial h)(\bar{y}_+), \quad \text{s.t.} \quad \lambda_+ \bar{u}_+ = x_+ - \bar{y}_+, \\ &\psi_+ = \lambda_+ \|\bar{y}_+ - x_+\|^{d-1} = \lambda_+^d \|\bar{u}_+\|^{d-1} \end{aligned} \quad (4.13)$$

and

$$\begin{aligned} \bar{u}_- \in (\nabla f_{x_-} + \partial h)(\bar{y}_-), \quad \text{s.t.} \quad \lambda_- \bar{u}_- = x_- - \bar{y}_-, \\ \psi_- = \lambda_- \|\bar{y}_- - x_-\|^{d-1} = \lambda_-^d \|\bar{u}_-\|^{d-1}. \end{aligned} \quad (4.14)$$

To proceed, we have the following bound on $\lambda_-^2 \|\bar{u}_+ - \bar{u}_-\|$ whose proof can be found in the appendix.

Lemma 4.5 *It holds that*

$$\lambda_-^2 \|\bar{u}_+ - \bar{u}_-\| \leq 2\lambda_-^2 \|\nabla f_{x_+}(\bar{y}_+) - \nabla f_{x_-}(\bar{y}_+)\| + |\lambda_+ - \lambda_-| \|\bar{y}_+ - x_+\| + \lambda_- \|x_+ - x_-\|. \quad (4.15)$$

Note that

$$\left| a^{d-1} - b^{d-1} \right| = \left| (a-b)(a^{d-2} + a^{d-3}b + \dots + b^{d-2}) \right| \leq (d-1)|a-b| \max\{a, b\}^{d-2}, \quad (4.16)$$

for any $a, b > 0$. Now combining (4.13), (4.14), (4.15) and (4.16) we have

$$\begin{aligned} & |\psi_+ - \psi_-| \\ &= \left| \lambda_+^d \|\bar{u}_+\|^{d-1} - \lambda_-^d \|\bar{u}_-\|^{d-1} \right| \\ &\leq \left| \lambda_+^d - \lambda_-^d \right| \|\bar{u}_+\|^{d-1} + \left| \|\bar{u}_+\|^{d-1} - \|\bar{u}_-\|^{d-1} \right| \lambda_-^d \\ &\leq |\lambda_+ - \lambda_-| d \lambda_+^{d-1} \|\bar{u}_+\|^{d-1} + \|\bar{u}_+ - \bar{u}_-\| (d-1) \max\{\|\bar{u}_+\|, \|\bar{u}_-\|\}^{d-2} \lambda_-^d \\ &= |\lambda_+ - \lambda_-| d \lambda_+^{d-1} \|\bar{u}_+\|^{d-1} + (d-1) \|\bar{u}_+ - \bar{u}_-\| \max\{\|\lambda_- \bar{u}_+\|, \|\lambda_- \bar{u}_-\|\}^{d-2} \lambda_-^2 \\ &\leq |\lambda_+ - \lambda_-| d \lambda_+^{d-1} \|\bar{u}_+\|^{d-1} + (d-1) \|\bar{u}_+ - \bar{u}_-\| \max\{\|\lambda_+ \bar{u}_+\|, \|\lambda_- \bar{u}_-\|\}^{d-2} \lambda_-^2 \\ &= d |\lambda_+ - \lambda_-| \|\bar{y}_+ - x_+\|^{d-1} + (d-1) \|\bar{u}_+ - \bar{u}_-\| \max\{\|x_+ - \bar{y}_+\|, \|x_- - \bar{y}_-\|\}^{d-2} \lambda_-^2 \\ &\leq d |\lambda_+ - \lambda_-| \|\bar{y}_+ - x_+\|^{d-1} + (d-1) \max\{\|x_+ - \bar{y}_+\|, \|x_- - \bar{y}_-\|\}^{d-2} \\ &\quad \times (2\lambda_-^2 \|\nabla f_{x_+}(\bar{y}_+) - \nabla f_{x_-}(\bar{y}_+)\| + |\lambda_+ - \lambda_-| \|\bar{y}_+ - x_+\| + \lambda_- \|x_+ - x_-\|). \end{aligned} \quad (4.17)$$

Next, by applying (1.9), (4.9), Lemma A.2, Lemma A.5 and Lemma A.6, we have

$$\begin{aligned} \lambda_- \leq \bar{\lambda} &= \max \left\{ \Theta \left(\bar{\epsilon}^{-\frac{d-1}{d+1}} \right), \Theta \left(\bar{\rho}^{-\frac{d-1}{d}} \right) \right\} \leq \max \left\{ \Theta(\bar{\epsilon}^{-1}), \Theta \left(\bar{\rho}^{-\frac{d+1}{d}} \right) \right\}, \\ \lambda_+ - \lambda_- &\leq \max \left\{ \Theta(\bar{\epsilon}^{-2}), \Theta \left(\bar{\rho}^{-\frac{2(d+1)}{d}} \right) \right\} (\beta_+ - \beta_-), \\ \|x_+ - \bar{y}_+\| &\leq \max \left\{ \Theta(\bar{\epsilon}^{-1}), \Theta \left(\bar{\rho}^{-\frac{d+1}{d}} \right) \right\}, \\ \|x_- - \bar{y}_-\| &\leq \max \left\{ \Theta \left(\bar{\epsilon}^{-\frac{d-1}{d+1}} \right), \Theta \left(\bar{\rho}^{-\frac{d-1}{d}} \right) \right\} \leq \max \left\{ \Theta(\bar{\epsilon}^{-1}), \Theta \left(\bar{\rho}^{-\frac{d+1}{d}} \right) \right\}, \\ \|\nabla f_{x_+}(\bar{y}_+) - \nabla f_{x_-}(\bar{y}_+)\| &\leq \max \left\{ \Theta \left(\bar{\epsilon}^{-d+1} \right), \Theta \left(\bar{\rho}^{-\frac{(d-1)(d+1)}{d}} \right) \right\} (\beta_+ - \beta_-). \end{aligned}$$

Combining the bounds above with (4.17) yields

$$\begin{aligned}
& |\psi_+ - \psi_-| \\
& \leq d \max \left\{ \Theta \left(\bar{\epsilon}^{-d-1} \right), \Theta \left(\bar{\rho}^{-\frac{(d+1)^2}{d}} \right) \right\} (\beta_+ - \beta_-) + (d-1) \max \left\{ \Theta \left(\bar{\epsilon}^{-d+2} \right), \Theta \left(\bar{\rho}^{-\frac{(d+1)(d-2)}{d}} \right) \right\} \\
& \quad \times \left(2 \max \left\{ \Theta \left(\bar{\epsilon}^{-d-1} \right), \Theta \left(\bar{\rho}^{-\frac{(d+1)^2}{d}} \right) \right\} + \max \left\{ \Theta \left(\bar{\epsilon}^{-3} \right), \Theta \left(\bar{\rho}^{-\frac{3(d+1)}{d}} \right) \right\} \right. \\
& \quad \left. + \max \left\{ \Theta \left(\bar{\epsilon}^{-1} \right), \Theta \left(\bar{\rho}^{-\frac{(d+1)}{d}} \right) \right\} \right) (\beta_+ - \beta_-) \\
& \leq \max \left\{ \Theta \left(\bar{\epsilon}^{-2d+1} \right), \Theta \left(\bar{\rho}^{-\frac{(2d-1)(d+1)}{d}} \right) \right\} (\beta_+ - \beta_-),
\end{aligned}$$

where the last inequality is due to $d \geq 2$. Because $\beta_+ - \beta_- = \frac{1}{2^j}$, from (4.11) we have

$$(1 - \hat{\sigma})^{d-1} \alpha_+ - (1 + \hat{\sigma})^{d-1} \alpha_- \leq \max \left\{ \Theta \left(\bar{\epsilon}^{-2d+1} \right), \Theta \left(\bar{\rho}^{-\frac{(2d-1)(d+1)}{d}} \right) \right\} \frac{1}{2^j}.$$

The left hand side of the above inequality is a positive constant. Therefore,

$$j \leq \Theta \left(\max \{ \log_2(\bar{\epsilon}^{-1}), \log_2(\bar{\rho}^{-1}) \} \right)$$

as required. \square

Remark 4.6 *In fact, we can quantify the constants in the proof of Theorem 4.1 more explicitly, and obtain the exact form of the bound $\Theta \left(\max \{ \log_2(\bar{\epsilon}^{-1}), \log_2(\bar{\rho}^{-1}) \} \right)$. Recall that*

$$\bar{\lambda} = \max \left\{ \alpha_-^{1/d} \left[\frac{1}{\bar{\rho}} \left(1 + \hat{\sigma} + \frac{L_d + M}{d!} \alpha_- \right) \right]^{1-1/d}, \left(\frac{\sigma^2 \alpha_-^{2/(d-1)}}{2\bar{\epsilon}} \right)^{\frac{d-1}{d+1}} \right\} \quad \text{and} \quad D_1 = \left(2 + \frac{2}{\sqrt{1 - \sigma^2}} D \right).$$

Introduce the following constants

$$\begin{aligned}
G_1 &= \frac{4(\bar{\lambda} + 4\bar{C})^2}{\hat{C}}, \\
G_2 &= D + \frac{L_d D}{d!} \max\left(\frac{9}{4}\bar{C}, 8\bar{\lambda}\right), \\
G_3 &= (1 + \hat{\sigma}) \left[D_1 + \frac{L_d D_1^{d+1}}{d!} \bar{\lambda} \right], \\
G_4 &= \sum_{l=2}^d \left[(l-1) B_l D_1 (D_1 + G_2)^{l-2} + B_{l+1} D_1 (D_1 + G_2)^{l-1} \right] + B_2 D_1,
\end{aligned}$$

where

$$\hat{C} = \frac{d! \sigma_l}{(L_d + M) D_1^{d-1}}, \quad \bar{C} = \max \left\{ \frac{\sigma^2 D}{2(1 - \sigma^2) \bar{\epsilon}}, \frac{D^{(3d-1)/(2d)} (1 + \sigma)^{1/d}}{(1 - \sigma) \alpha_-^{\frac{d-1}{d(d-2)}}} \left(\frac{1}{\bar{\rho}} \right)^{\frac{d+1}{d}} \right\}$$

and B_1, \dots, B_d is a sequence defined by

$$B_d = \|\nabla^d f(x_*)\|, \quad B_{l-1} = \|\nabla^{l-1} f(x_*)\| + 2D_1 B_l, \quad l = 2, \dots, d.$$

Then the complexity bound in Theorem 4.4 can be explicitly expressed by

$$\log \left(\frac{dG_1 \bar{\lambda}^{d-1} + (d-1) \max(G_2, G_3)^{d-2} \bar{\lambda}^2 (2\bar{\lambda}^2 G_4 + G_1 G_2 + \bar{\lambda} D_1)}{(1-\hat{\sigma})^{d-1} \alpha_+ - (1+\hat{\sigma})^{d-1} \alpha_-} \right). \quad (4.18)$$

It is clear that the dependence of the resulting bound depends logarithmically on the parameters L_d, D , and input parameters $\alpha_+, \alpha_-, \sigma_u$, and polynomially on d . The derivation of (4.18) is skipped for the sake of succinctness.

Now, for a given $\epsilon > 0$, we denote

$$\bar{D}_\epsilon := \sup\{\|x - x_*\| : \exists y \in \partial F(x) \text{ s.t. } \|y\| < \epsilon\}.$$

Combining the bounds provided in Theorem 3.4, Theorem 4.1 and (4.18), we obtain the overall iteration bound for Algorithm 2 in terms of the **ATS** calls as follows:

Theorem 4.7 *Given $\epsilon > 0$. Assume that Algorithm 2 is implemented with $M \leq 2L_d$. Set $\bar{\epsilon} = \epsilon/2$, $\bar{\rho} \leq \min\left\{\frac{\epsilon}{2\bar{D}_\epsilon}, \epsilon\right\}$, and define*

$$K_\epsilon := \left\lceil \frac{d+1}{2} \left(\frac{2^d}{(1-(\hat{\sigma} + \sigma_u)^2)^{(d-1)/2} d! \sigma_l} \right)^{\frac{2}{3d+1}} \left(\frac{(L_d + M) D^{d+1}}{\epsilon} \right)^{\frac{2}{3d+1}} T_\epsilon \right\rceil$$

where

$$T_\epsilon = \log \left(\frac{dG_1 \bar{\lambda}^{d-1} + (d-1) \max(G_2, G_3)^{d-2} \bar{\lambda}^2 (2\bar{\lambda}^2 G_4 + G_1 G_2 + \bar{\lambda} D_1)}{(1-\hat{\sigma})^{d-1} \alpha_+ - (1+\hat{\sigma})^{d-1} \alpha_-} \right).$$

Then, a point $z \in \mathbb{R}^n$ satisfying

$$F(z) - F_* \leq \epsilon$$

can be found by Algorithm 2 with no more than K_ϵ calls of **ATS**.

Proof. We consider two cases separately. In the first case, Algorithm 2 terminates because we find a $k \leq K_\epsilon$ such that $\|v_k\| \leq \bar{\rho}$ and $\|\epsilon_k\| \leq \bar{\epsilon}$. As $v_k = \nabla f(y_k) - \nabla f_{x_k}(y_k) + u_k$ and $u_k \in \nabla f_{x_k}(y_k) + \partial_{\epsilon_k} h(y_k)$, we have $v_k \in \nabla f(y_k) + \partial_{\epsilon_k} h(y_k)$. Let x_* be the projection of x_0 onto X_* . By the convexity of f and h ,

$$\begin{aligned} f(x_*) &\geq f(y_k) + \langle \nabla f(y_k), x_* - y_k \rangle \\ h(x_*) &\geq h(y_k) + \langle v_k - \nabla f(y_k), x_* - y_k \rangle - \epsilon_k. \end{aligned}$$

Summing up the two inequalities above yields

$$F_* \geq F(y_k) + \langle v_k, x_* - y_k \rangle - \epsilon_k \geq F(y_k) - \bar{\rho} \|y_k - x_*\| - \bar{\epsilon}.$$

By the construction of $\bar{\rho}$, we have that $\|v_k\| \leq \bar{\rho} \leq \epsilon$. Together with the definition of \bar{D}_ϵ , this implies that $\|y_k - x_*\| \leq \bar{D}_\epsilon$. Again, by evoking the construction of $\bar{\rho}$ and $\bar{\epsilon}$, it holds

$$F_* \geq F(y_k) - \frac{\epsilon}{2} - \frac{\epsilon}{2} = F(y_k) - \epsilon.$$

In the other case, condition (3.7) holds for every $k \leq K_\epsilon$. Then, according to Theorem 3.4, for all $k \leq K_\epsilon$ we have

$$F(y_k) - F_* \leq \left(\frac{d+1}{2}\right)^{\frac{3d+1}{2}} \frac{2^d}{(1 - (\hat{\sigma} + \sigma_u)^2)^{\frac{d-1}{2}} d! \sigma_l} D^{d+1} (L_d + M) k^{-\frac{3d+1}{2}}.$$

By the definition of K_ϵ and letting $k = K_\epsilon$ in the above inequality, one has

$$F(y_{K_\epsilon}) - F_* \leq \epsilon.$$

□

Note that the implementation of our framework is based on the assumption that the **ATS** can be efficiently computed. By the construction, the objective in (2.5) has a $\frac{1}{\lambda}$ -strongly convex smooth part, which can be solved by many start-of-the-art optimization algorithms. However, there is few efficient algorithms customized for problem (2.5). Without further knowledge of the problem structure, the proposed approach is not necessarily more efficient as compared to, e.g., a direct application of a general-purpose convex optimization algorithm on (1.1). At the end of the paper, we shall briefly discuss a method for the subroutine in the special case $d = 3$ introduced by Nesterov [27], which opens the door for this line of research. Of course, how to efficiently solve **ATS** in general remains a further research topic.

5 Concluding Remark

To conclude this paper, we shall discuss how to compute **ATS** efficiently with $d = 3$. Note that in STEP 2.a of Algorithm 3, an Approximate Tensor Subroutine (**ATS**) is required, which can be implemented in polynomial time in the case of convex optimization. In some applications, **ATS** may be implemented efficiently if some additional structures on the tensor (Taylor) expansion and/or the h function exist. In this subsection, we show how **ATS** (i.e., solve problem (2.5)) may be computed efficiently in the absence of the non-smooth part, i.e. $F(x) = f(x)$, when $d = 3$. Note that since $h = 0$, the ϵ_β in the bisection subroutine may be simply set to 0.

In this case, the objective function in (2.5) becomes: $f_x(y) + \frac{1}{\lambda}\|y - x\|^2 = f(x) + \Omega(y - x)$ where

$$\Omega(z) = z^\top \nabla f(x) + \frac{1}{2} z^\top \left(\nabla^2 f(x) + \frac{1}{\lambda} I \right) z + \frac{1}{3!} \nabla^3 f(x)[z]^3 + \frac{M}{4!} \|z\|^4.$$

Therefore, the subproblem (2.5) is equivalent to $\min_{z \in \mathbb{R}^n} \Omega(z)$. Let $M = 3\kappa^2 L_3$ with $\kappa > 1$. Then, a similar argument as in Lemma 4 of [27] implies that function $\Omega(z)$ satisfies the strong relative smoothness condition

$$\nabla^2 \rho(z) \preceq \nabla^2 \Omega(z) \preceq \frac{\kappa + 1}{\kappa - 1} \nabla^2 \rho(z) \tag{5.1}$$

with respect to function

$$\rho(z) = z^\top \left(\frac{\kappa - 1}{2\kappa} \nabla^2 f(x) + \frac{\kappa - 1}{2\lambda(\kappa + 1)} I \right) z + \frac{M - 3\kappa L_3}{6} \|z\|^4.$$

Such condition allows to minimize $\Omega(z)$ efficiently by a gradient method described in [21, 27], where we need to solve the following problem in every iteration:

$$\min_{z \in \mathbb{R}^n} \left(a^\top z + \frac{1}{2} z^\top A z + \frac{\gamma}{4} \|z\|^4 \right), \quad A \succeq 0, \quad \gamma > 0,$$

which was considered at the end of Section 5 in [27]. According to a min-max argument in [27], the above problem is shown to be equivalent to

$$\min_{\tau > 0} \left(\gamma \tau^2 + \frac{1}{2} a^\top (\gamma \tau I + A)^{-1} a \right),$$

which is actually a univariate optimization problem with a strongly convex and analytic objective function, hence is easily solvable in practice. Note that a matrix inverse operation is required in the univariate optimization above. Since in all iterations of the gradient method described in [21, 27], the matrix A is exactly $\nabla^2 f(x)$ throughout and only the vector a varies, the matrix inverse operation needs to be performed only once. As the gradient method in [21, 27] is linearly convergent, the total computational cost of an **ATS** call is in the order of $O(n^3 + n^2 \log(\bar{\rho}^{-1}))$.

Acknowledgments.

We would like to thank Tianyi Lin at UC Berkeley for the helpful discussions at the early stages of the project.

References

- [1] M.M. Alves, R.D.C. Monteiro and B.F. Svaiter (2014) Primal-dual regularized SQP and SQCQP type methods for convex programming and their complexity analysis.
- [2] M.M. Alves, R.D.C. Monteiro and B.F. Svaiter (2016) Iteration-complexity of a Rockafellar’s proximal method of multipliers for convex programming based on second-order approximations. *arXiv:1602.06794*
- [3] M. Baes (2009) Estimate sequence methods: extensions and approximations. *Institute for Operations Research, ETH, Zurich, Switzerland.*
- [4] A. Beck and M. Teboulle (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.
- [5] E.G. Birgin, J.L. Gardenghi, J.M. Martinez, S.A. Santos, and Ph.L. Toint (2017) Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1-2):359–368.

- [6] B.Bullins (2018) Fast minimization of structured convex quartics. *ArXiv Preprint: 1812.10349*.
- [7] S. Bubeck, Y.T. Lee, and M. Singh (2015) A geometric alternative to Nesterov’s accelerated gradient descent. *ArXiv Preprint: 1506.08187*.
- [8] S. Bubeck, Q. Jiang, Y.T. Lee, Y. Li and A. Sidford (2018) Near-optimal method for highly smooth convex optimization. *ArXiv Preprint: 1812.08026*.
- [9] R.S. Burachik, A.N. Iusem and B.F. Svaiter (1997) Enlargement of monotone operators with applications to variational inequalities, *Set-Valued Anal.*, 5, 159-180.
- [10] C. Cartis, N.I.M. Gould, and Ph.L. Toint (2017) Improved second-order evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *arXiv:1708.04044*.
- [11] C. Cartis, N.I.M. Gould, and Ph.L. Toint (2018) Universal regularization methods: varying the power, the smoothness and the accuracy. *arXiv:1811.07057v1*.
- [12] A. Cotter, O. Shamir, N. Srebro, and K. Sridharan (2011) Better mini-batch algorithms via accelerated gradient methods. In *NIPS*, pages 1647–1655.
- [13] L. Calatroni and A. Chambolle (2017) Backtracking strategies for accelerated descent methods with smooth composite objectives. *arXiv:1709.09004*.
- [14] Y. Drori and M. Teboulle (2014) Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482.
- [15] A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych and C. A. Uribe (2018) The global rate of convergence for optimal tensor methods in smooth convex optimization. *arXiv:1809.00382*.
- [16] G.N. Grapiglia and Yu. Nesterov (2018) Accelerated regularized newton methods for minimizing composite convex functions. *Technical Report CORE Discussion paper*.
- [17] A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, C.A. Uribe, B. Jiang, H. Wang, S. Zhang, S. Bubeck, Q. Jiang, Y.T. Lee, Y. Li and A. Sidford (2019) Near optimal methods for minimizing convex functions with Lipschitz p -th derivatives. *In: Conference on Learning Theory (COLT)*, pp. 1392-1393.
- [18] B. Jiang, T. Lin, and S. Zhang (2018) A Unified Adaptive Tensor Approximation Scheme to Accelerate Composite Convex Optimization. *arXiv:1811.02427*.
- [19] G. Lan (2012) An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397.
- [20] Q. Lin and L. Xiao (2014) An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. *Computational Optimization and Applications*, 60(3):633–674.
- [21] H. Lu and R.M. Freund and Yu. Nesterov (2018) Relatively smooth convex optimization by first-order methods and applications. *SIAM Journal on Optimization*, 28 (1), pp. 333-354.

- [22] J.M. Martínez (2017) On high-order model regularization for constrained optimization. *SIAM Journal on Optimization*, 27(4):2447–2458.
- [23] R.D.C. Monteiro and B.F. Svaiter (2012) Iteration-Complexity of a Newton Proximal Extragradient Method for Monotone Variational Inequalities and Inclusion Problems. *SIAM Journal on Optimization*, 22, 914-935.
- [24] R.D.C. Monteiro and B.F. Svaiter (2013) An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23 (2), pp. 1092-1125.
- [25] Yu. Nesterov (1983) A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Doklady AN SSSR, translated as Soviet Math. Docl.*, 269:543–547.
- [26] Yu. Nesterov (2008) Accelerating the cubic regularization of Newton’s method on convex problems. *Mathematical Programming*, 112(1):159–181.
- [27] Yu. Nesterov (2018) Implementable tensor methods in unconstrained convex optimization, *CORE Discussion Paper 2018/05, Catholic University of Louvain, Center for Operations Research and Econometrics (CORE)*.
- [28] R.T. Rockafellar (1970) On the maximal monotonicity of subdifferential mappings. *Pacific J. Math.*, 33:209-216.
- [29] K. Scheinberg, D. Goldfarb and X. Bai (2014) Fast first-order methods for composite convex optimization with backtracking. *Foundations of Computational Mathematics*, 14:389–417.
- [30] S. Shalev-Shwartz and T. Zhang (2014) Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *ICML*, pages 64–72.
- [31] Y. Arjevani, O. Shamir and R. Shiff (2018) Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming*, published online.
- [32] W. Su, S. Boyd and E.J. Candes (2016) A differential equation for modeling Nesterov’s accelerated gradient method: theory and insights. *Journal of Machine Learning Research*, 17(153):1–43.
- [33] A. Wibisono, A.C. Wilson and M.I. Jordan (2016) A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, pages 7351–7358.
- [34] A.C. Wilson, B. Recht and M.I. Jordan (2016) A Lyapunov analysis of momentum methods in optimization. *ArXiv Preprint: 1611.02635*.

A Proofs of the lemmas in Section 4

We first establish an uniform lower bound as well as an upper bound for the sequence $\{A_k\}$.

Lemma A.1 *Let D be the distance of x_0 to X_* . Suppose $\{A_k\}_{k=1}^\ell$ is generated from Algorithm 2, and the algorithm has not stopped at iteration ℓ . Then for any integer $1 \leq k \leq \ell$, it holds that*

$$A_k \geq \frac{d! \sigma_l}{(L_d + M) \left(\frac{2}{\sqrt{1-\sigma^2}} + 2 \right)^{d-1} D^{d-1}}, \quad (1.1)$$

and

$$A_k \leq \max \left\{ \frac{\sigma^2 D}{2\bar{\epsilon}(1-\sigma^2)}, \frac{D^{\frac{3d-1}{2d}} (1+\sigma)^{\frac{1}{d}}}{(1-\sigma)(\alpha_-)^{\frac{(d-1)}{d(d-2)}}} \left(\frac{1}{\bar{\rho}} \right)^{\frac{d+1}{d}} \right\}. \quad (1.2)$$

Proof. We first establish the lower bound. Since $\{A_k\}$ is monotonically increasing, it suffices to lower bound A_1 . Recall that $A_0 = 0$ and $A_1 = A_0 + a_1 = \lambda_1$, and the choice of large-step (3.7) in Algorithm 2 leads to

$$\frac{d! \sigma_l}{L_d + M} \leq \lambda_1 \|y_1 - \tilde{x}_0\|^{d-1}.$$

Moreover, Lemma 3.1 implies that

$$\|x_k - x_*\| \leq D, \quad \text{and} \quad \|y_k - x_*\| \leq \left(\frac{2}{\sqrt{1-\sigma^2}} + 1 \right) D, \quad (1.3)$$

where x_* is the projection of x_0 onto the optimal solution set X_* . Combining the above two inequalities with the fact that $\tilde{x}_0 = x_0$, it follows that

$$\|y_1 - \tilde{x}_0\| \leq \|y_1 - x_*\| + \|x_* - \tilde{x}_0\| \leq \left(\frac{2}{\sqrt{1-\sigma^2}} + 2 \right) D.$$

Therefore,

$$A_1 = \lambda_1 \geq \frac{d! \sigma_l}{(L_d + M) \left(\frac{2}{\sqrt{1-\sigma^2}} + 2 \right)^{d-1} D^{d-1}},$$

which is a uniform lower bound of the sequence $\{A_k\}$.

Next, we provide the upper bound. By invoking (3.12) to $(y_k, v_k, \epsilon_k, \lambda_k, \tilde{x}_{k-1})$, it holds that

$$\lambda_k \|v_k\| \leq (1+\sigma) \|y_k - \tilde{x}_{k-1}\|, \quad (1.4)$$

$$2\lambda_k \epsilon_k \leq \sigma^2 \|y_k - \tilde{x}_{k-1}\|^2. \quad (1.5)$$

Then, combining (1.4) with (3.4) leads to

$$A_k \lambda_k \|v_k\|^2 \leq (1+\sigma)^2 \frac{A_k}{\lambda_k} \|y_k - \tilde{x}_{k-1}\|^2 \leq (1+\sigma)^2 \frac{D^2}{1-\sigma^2}. \quad (1.6)$$

Moreover, it follows from (3.14) that

$$\frac{A_k}{\lambda_k^{\frac{d+1}{d-1}}} \leq \sum_{j=1}^k \frac{A_j}{\lambda_j^{\frac{d+1}{d-1}}} \leq \frac{D^2}{(1-\sigma^2)(\alpha_-)^{\frac{2}{d-2}}}$$

where $\alpha_- = \frac{d! \sigma_l}{L_d + M}$. Combining the above two inequalities yields

$$\left(\frac{A_k (1 - \sigma^2) (\alpha_-)^{\frac{2}{d-2}}}{D^2} \right)^{\frac{d-1}{d+1}} \|v_k\|^2 A_k \leq \lambda_k \|v_k\|^2 A_k \leq \frac{1 + \sigma}{1 - \sigma} D,$$

or equivalently,

$$A_k \leq \frac{D^{\frac{3d-1}{2d}} (1 + \sigma)^{\frac{1}{d}}}{(1 - \sigma) (\alpha_-)^{\frac{(d-1)}{d(d-2)}}} \left(\frac{1}{\|v_k\|} \right)^{\frac{d+1}{d}}. \quad (1.7)$$

On the other hand, (1.5) together with (3.4) implies that

$$2A_k \epsilon_k \leq \frac{A_k}{\lambda_k} \sigma^2 \|y_k - \tilde{x}_{k-1}\|^2 \leq \frac{\sigma^2 D}{1 - \sigma^2}.$$

Consequently,

$$A_k \leq \frac{\sigma^2 D}{2(1 - \sigma^2)} \frac{1}{\epsilon_k}. \quad (1.8)$$

Now, since the algorithm has not been terminated, we have either $\|v_k\| \geq \bar{\rho}$ or $\epsilon_k \geq \bar{\epsilon}$, which combined with (1.7) and (1.8) yields that

$$A_k \leq \max \left\{ \frac{\sigma^2 D}{2\bar{\epsilon}(1 - \sigma^2)}, \frac{D^{\frac{3d-1}{2d}} (1 + \sigma)^{\frac{1}{d}}}{(1 - \sigma)} \left(\frac{1}{\alpha_1} \right)^{\frac{(d-1)}{d(d-2)}} \left(\frac{1}{\bar{\rho}} \right)^{\frac{d+1}{d}} \right\},$$

and the conclusion follows. \square

Below we prove Lemma 4.4 and Lemma 4.5 respectively.

Proof of Lemma 4.4. We first demonstrate that

$$\beta_- \leq \bar{\beta} = \beta(\bar{\lambda}). \quad (1.9)$$

Otherwise, we have $\beta_- \geq \bar{\beta}$ and $\lambda_- > \bar{\lambda}$ as the function $\lambda(\beta)$ is strictly increasing in β . This together with Lemma 4.3 and the fact that $\lambda_- \|y_- - x_-\| \leq \alpha_-$, implies that the vector $v_{\beta_-} = \nabla f(y_-) - \nabla f_{x_-}(y_-) + u_{\beta_-}$ satisfies

$$v_{\beta_-} \in (\nabla f + (\partial h)^\epsilon)(y_-), \quad \|v_{\beta_-}\| \leq \bar{\rho}, \quad \epsilon \leq \bar{\epsilon},$$

and thus the algorithm would have been terminated, yielding a contradiction. So we have $\beta_- \leq \bar{\beta}$. Since $j \geq \log_2(2/(1 - \bar{\beta}))$, we have

$$\beta_+ = \beta_- + \frac{1}{2^j} \leq \bar{\beta} + \frac{1 - \bar{\beta}}{2} = \frac{1 + \bar{\beta}}{2} < 1. \quad (1.10)$$

Together with the monotonicity of the function $\lambda(\beta)$ this implies that

$$\lambda_+ \leq \lambda \left(\frac{1 + \bar{\beta}}{2} \right) = A_k \frac{((1 + \bar{\beta})/2)^2}{1 - (1 + \bar{\beta})/2} = A_k \frac{(1 + \bar{\beta})^2}{2(1 - \bar{\beta})} = \bar{\lambda} \frac{(1 + \bar{\beta})^2}{\bar{\beta}^2}.$$

Therefore, $\lambda_+ \leq A_k \frac{(1+\bar{\beta})^2}{2(1-\bar{\beta})} \leq \left(\frac{3}{2}\right)^2 A_k$ when $\bar{\beta} \leq \frac{1}{2}$ and $\lambda_+ \leq \bar{\lambda} \frac{(1+\bar{\beta})^2}{\bar{\beta}^2} \leq \left(\frac{2}{1/2}\right)^2 \bar{\lambda}$ when $\bar{\beta} \geq \frac{1}{2}$. Combining the bounds in both cases, we have

$$\begin{aligned} \lambda_+ &\leq \max \{9A_k/4, 8\bar{\lambda}\} \leq \max \left\{ \Theta(\bar{\epsilon}^{-1}), \Theta\left((\bar{\rho}^{-1})^{\frac{d+1}{d}}\right), \Theta\left((\bar{\epsilon}^{-1})^{\frac{d-1}{d+1}}\right), \Theta\left((\bar{\rho}^{-1})^{\frac{d-1}{d}}\right) \right\} \\ &= \max \left\{ \Theta(\bar{\epsilon}^{-1}), \Theta\left((\bar{\rho}^{-1})^{\frac{d+1}{d}}\right) \right\}, \end{aligned}$$

where the second inequality is due to the upper bounds of $\bar{\lambda}$ and A_k in (4.9) and (1.2) respectively. \square

Proof of Lemma 4.5. Let $\bar{v} := \bar{u}_+ - \nabla f_{x_+}(\bar{y}_+) + \nabla f_{x_-}(\bar{y}_+)$. Then $\bar{v} \in (\nabla f_{x_-} + \partial h)(\bar{y}_+)$. By (4.13) and (4.14), it holds that

$$\begin{aligned} &\bar{y}_+ - \bar{y}_- + \lambda_+ \bar{u}_+ - \lambda_- \bar{u}_- = x_+ - x_- \\ \iff &\bar{y}_+ - \bar{y}_- + \lambda_- (\bar{u}_+ - \bar{u}_-) = (\lambda_- - \lambda_+) \bar{u}_+ + x_+ - x_- \\ \iff &\bar{y}_+ - \bar{y}_- + \lambda_- (\bar{v} - \bar{u}_+) = \lambda_- (\bar{v} - \bar{u}_+) + (\lambda_- - \lambda_+) \bar{u}_+ + x_+ - x_-. \end{aligned}$$

Recall that $\bar{v} \in (\nabla f_{x_-} + \partial h)(\bar{y}_+)$ and $\bar{u}_- \in (\nabla f_{x_-} + \partial h)(\bar{y}_-)$, and from the convexity of $f_{x_-} + h$ it holds that

$$\langle \bar{y}_+ - \bar{y}_-, \bar{v} - \bar{u}_- \rangle \geq 0.$$

Therefore,

$$\begin{aligned} \lambda_- \|\bar{v} - \bar{u}_-\| &\leq \|\bar{y}_+ - \bar{y}_- + \lambda_- (\bar{v} - \bar{u}_-)\| \\ &= \left\| \lambda_- (\bar{v} - \bar{u}_+) + (\lambda_- - \lambda_+) \bar{u}_+ + x_+ - x_- \right\| \\ &\leq \lambda_- \|\bar{v} - \bar{u}_+\| + |\lambda_- - \lambda_+| \|\bar{u}_+\| + \|x_+ - x_-\|. \end{aligned}$$

Using the previous identity and the triangle inequality of the norms implies that

$$\begin{aligned} \lambda_-^2 \|\bar{u}_+ - \bar{u}_-\| &\leq \lambda_-^2 (\|\bar{u}_+ - \bar{v}\| + \|\bar{v} - \bar{u}_-\|) \\ &\leq 2\lambda_-^2 \|\bar{v} - \bar{u}_+\| + |\lambda_- - \lambda_+| \lambda_- \|\bar{u}_+\| + \lambda_- \|x_+ - x_-\| \\ &\leq 2\lambda_-^2 \|\nabla f_{x_+}(\bar{y}_+) - f_{x_-}(\bar{y}_+)\| + |\lambda_- - \lambda_+| \|\bar{y}_+ - x_+\| + \lambda_- \|x_+ - x_-\|. \end{aligned}$$

\square

Next we shall present the lemmas with proofs that were used in Section 4.

Lemma A.2 *Suppose λ_+ , λ_- , β_+ and β_- are generated from Algorithm 3. When the number of iteration j in Algorithm 3 satisfying $j \geq \log_2(2/(1-\bar{\beta}))$ with $\bar{\beta} = \beta(\bar{\lambda})$ and $\bar{\lambda}$ defined in (4.9), we have*

$$\lambda_+ - \lambda_- \leq \left(\max \left\{ \Theta(\bar{\epsilon}^{-1}), \Theta\left(\bar{\rho}^{-\frac{d+1}{d}}\right) \right\} \right)^2 (\beta_+ - \beta_-). \quad (1.11)$$

Proof. Since $j \geq \log_2(2/(1-\bar{\beta}))$, inequality (1.10) holds. By the mean-value theorem and the definition of $\lambda(\beta)$, there exists $\eta \in (\beta_-, \beta_+)$ such that

$$\lambda_+ - \lambda_- = A_k \left(\frac{1}{(1-\eta)^2} - 1 \right) (\beta_+ - \beta_-) \leq A_k \left(\frac{4}{(1-\bar{\beta})^2} - 1 \right) (\beta_+ - \beta_-),$$

where the inequality is due to (1.10). Recall that

$$\bar{\beta} = \frac{\sqrt{\bar{\lambda}^2 + 4\bar{\lambda}A_k} - \bar{\lambda}}{2A_k} = \frac{2\bar{\lambda}}{\sqrt{\bar{\lambda}^2 + 4\bar{\lambda}A_k} + \bar{\lambda}}.$$

The relation of β and λ in (4.1) gives

$$\frac{A_k}{(1 - \bar{\beta})^2} = \frac{\bar{\lambda}^2}{A_k \bar{\beta}^4} = \frac{(\sqrt{\bar{\lambda}^2 + 4\bar{\lambda}A_k} + \bar{\lambda})^4}{16 A_k \bar{\lambda}^2} \leq \frac{(\bar{\lambda} + 4A_k)^2}{A_k}.$$

Therefore, by invoking (4.9), (1.1) and (1.2), we have

$$\begin{aligned} & \lambda_+ - \lambda_- \\ & \leq A_k \left(\frac{4}{(1 - \bar{\beta})^2} - 1 \right) (\beta_+ - \beta_-) \\ & \leq \frac{4A_k}{(1 - \bar{\beta})^2} (\beta_+ - \beta_-) \\ & \leq \frac{4(\bar{\lambda} + 4A_k)^2}{A_k} (\beta_+ - \beta_-) \\ & \leq \left(\max \left\{ \Theta \left((\bar{\epsilon}^{-1})^{\frac{d-1}{d+1}} \right), \Theta \left((\bar{\rho}^{-1})^{\frac{d-1}{d}} \right) \right\} + \max \left\{ \Theta \left(\bar{\epsilon}^{-1} \right), \Theta \left((\bar{\rho}^{-1})^{\frac{d+1}{d}} \right) \right\} \right)^2 (\beta_+ - \beta_-) \\ & \leq \left(\max \left\{ \Theta \left(\bar{\epsilon}^{-1} \right), \Theta \left((\bar{\rho}^{-1})^{\frac{d+1}{d}} \right) \right\} \right)^2 (\beta_+ - \beta_-). \end{aligned}$$

□

The following lemma is exactly Proposition 4.5 in [23].

Lemma A.3 *Let $A : \mathbb{R}^s \rightrightarrows \mathbb{R}^s$ be a maximal monotone operator. Then for any $x, \tilde{x} \in \mathbb{R}^s$, we have*

$$\|(I + \lambda A)^{-1}(x) - (I + \lambda A)^{-1}(\tilde{x})\| \leq \|x - \tilde{x}\|. \quad (1.12)$$

Moreover, if $x_* \in A^{-1}(0)$ then

$$\max\{\|(I + \lambda A)^{-1}(x) - x\|, \|(I + \lambda A)^{-1}(x) - x_*\|\} \leq \|x - x_*\|. \quad (1.13)$$

Now we can bound the residual in terms of the distance between current iterate and an optimal solution.

Lemma A.4 *Let $T := \nabla f + \partial h$ and $T_x := \nabla f_x + \partial h$. Assume that $x_* \in T^{-1}(0) = (\nabla f + \partial h)^{-1}(0)$ and let $\bar{x}, x \in \mathbb{R}^n$ be given. Then,*

$$\|x - (I + \lambda T_{\bar{x}})^{-1}(x)\| \leq \|x - x_*\| + \frac{\lambda(L_d + M)}{d!} \|\bar{x} - x_*\|^d. \quad (1.14)$$

As a consequence, for every $x \in \mathbb{R}^n$, $x_* \in T^{-1}(0)$, and $\lambda > 0$, it holds that

$$\lambda \|x - (I + \lambda T_{\bar{x}})^{-1}(x)\|^{d-1} \leq \lambda \left(\|x - x_*\| + \frac{\lambda(L_d + M)}{d!} \|x - x_*\|^d \right)^{d-1}. \quad (1.15)$$

Proof. Let r be a constant mapping such that $r(x) = \nabla f(x_*) - \nabla f_{\bar{x}}(x_*)$ for any $x \in \mathbb{R}^n$. Then, construct $A := T_{\bar{x}} + r$, where A is also a maximal monotone operator. By Lemma A.3,

$$\|(I + \lambda A)^{-1}(x) - x\| \leq \|x - x_*\|. \quad (1.16)$$

Let $y = x + \lambda(\nabla f(x_*) - \nabla f_{\bar{x}}(x_*))$ and $z = (I + \lambda r + \lambda T_{\bar{x}})^{-1}(y)$. We have

$$x + \lambda(\nabla f(x_*) - \nabla f_{\bar{x}}(x_*)) = y = (I + \lambda r + \lambda T_{\bar{x}})(z) = z + \lambda(\nabla f(x_*) - \nabla f_{\bar{x}}(x_*)) + \lambda T_{\bar{x}}(z).$$

Canceling $\lambda(\nabla f(x_*) - \nabla f_{\bar{x}}(x_*))$ on both sides leads to

$$(I + \lambda T_{\bar{x}})^{-1}(x) = z = (I + \lambda r + \lambda T_{\bar{x}})^{-1}(y) = (I + \lambda A)^{-1}(y).$$

Combining the above inequality with (1.16) and Lemma 2.1 we have

$$\begin{aligned} \|x - (I + \lambda T_{\bar{x}})^{-1}(x)\| &= \|x - (I + \lambda A)^{-1}(y)\| \\ &\leq \|x - (I + \lambda A)^{-1}(x)\| + \|(I + \lambda A)^{-1}(x) - (I + \lambda A)^{-1}(y)\| \\ &\leq \|x - x_*\| + \lambda \|\nabla f(x_*) - \nabla f_{\bar{x}}(x_*)\| \\ &\leq \|x - x_*\| + \frac{\lambda(L_d + M)}{d!} \|\bar{x} - x_*\|^d, \end{aligned}$$

which proves (1.14), and (1.15) follows from (1.14) straightforwardly. \square

Lemma A.5 *Suppose $x_+ = x_{\beta_+}$ and $x_- = x_{\beta_-}$ are generated from Algorithm 3, and \bar{y}_+, \bar{y}_- are defined in (4.12). When the number of iterations j in Algorithm 3 satisfies $j \geq \log_2(2/(1 - \bar{\beta}))$ with $\bar{\beta} = \beta(\bar{\lambda})$ and $\bar{\lambda}$ defined in (4.9), we have*

$$\|x_+ - x_-\| \leq \left(2 + \frac{2}{\sqrt{1 - \sigma^2}}\right) D(\beta_+ - \beta_-), \quad \|x_+ - \bar{y}_+\| \leq \max \left\{ \Theta(\bar{\epsilon}^{-1}), \Theta\left(\bar{\rho}^{-\frac{d+1}{d}}\right) \right\}$$

and

$$\|x_- - \bar{y}_-\| \leq \max \left\{ \Theta\left(\bar{\epsilon}^{-\frac{d-1}{d+1}}\right), \Theta\left(\bar{\rho}^{-\frac{d-1}{d}}\right) \right\}.$$

Proof. Let x_* be the projection of x_0 onto the optimal solution set X_* . According to Lemma 3.1, it holds that

$$\|x_k - x_*\| \leq D \quad \text{and} \quad \|y_k - x_*\| \leq \left(\frac{2}{\sqrt{1 - \sigma^2}} + 1\right) D.$$

By (4.2), we have $x_+ = (1 - \beta_+)y_k + \beta_+x_k$ and $x_- = (1 - \beta_-)y_k + \beta_-x_k$. Therefore,

$$\|x_+ - x_*\| \leq D, \quad \text{and} \quad \|x_- - x_*\| \leq \left(\frac{2}{\sqrt{1 - \sigma^2}} + 1\right) D, \quad (1.17)$$

and

$$\begin{aligned} \|x_+ - x_-\| &= \|(\beta_+ - \beta_-)(x_k - y_k)\| \leq (\|x_k - x_*\| + \|y_k - x_*\|)(\beta_+ - \beta_-) \\ &\leq \left(2 + \frac{2}{\sqrt{1 - \sigma^2}}\right) D(\beta_+ - \beta_-). \end{aligned}$$

Recall that in the proof of Theorem 4.1, we showed that when $j \geq \log_2(2/(1-\bar{\beta}))$, $\beta_- \leq \bar{\beta}$ and inequality (4.10) holds. Applying Lemma A.4 with $x = \bar{x} = x_+$, inequality (4.10) and (4.5), we have

$$\|x_+ - y_+\| \leq (1 + \hat{\sigma})\|x_+ - x_*\| + (1 + \hat{\sigma})\frac{\lambda_+ L_d}{d!}\|x_+ - x_*\|^{d+1} \leq \max\left\{\Theta(\bar{\epsilon}^{-1}), \Theta\left(\bar{\rho}^{-\frac{d+1}{d}}\right)\right\}.$$

Since $\lambda(\beta)$ is monotonically increasing in β , $\beta_- \leq \bar{\beta}$ amounts to

$$\lambda_- \leq \bar{\lambda} = \max\left\{\Theta\left(\bar{\epsilon}^{-\frac{d-1}{d+1}}\right), \Theta\left(\bar{\rho}^{-\frac{d-1}{d}}\right)\right\}.$$

Finally, applying Lemma A.4 again with $x = \bar{x} = x_-$ and using (4.5) yields

$$\|x_- - y_-\| \leq (1 + \hat{\sigma})\|x_- - x_*\| + (1 + \hat{\sigma})\frac{\lambda_- L_d}{d!}\|x_- - x_*\|^{d+1} \leq \max\left\{\Theta\left(\bar{\epsilon}^{-\frac{d-1}{d+1}}\right), \Theta\left(\bar{\rho}^{-\frac{d-1}{d}}\right)\right\}.$$

□

Lemma A.6 *Suppose that $x_+ = x_{\beta_+}$ and $x_- = x_{\beta_-}$ are generated by Algorithm 3, and \bar{y}_+ , \bar{y}_- are defined in (4.12). If the number of iterations j in Algorithm 3 satisfies $j \geq \log_2(2/(1-\bar{\beta}))$ with $\bar{\beta} = \beta(\bar{\lambda})$ and $\bar{\lambda}$ defined in (4.9), then we have*

$$\|\nabla f_{x_+}(\bar{y}_+) - \nabla f_{x_-}(\bar{y}_+)\| \leq \max\left\{\Theta\left(\bar{\epsilon}^{-d+1}\right), \Theta\left(\bar{\rho}^{-\frac{(d-1)(d+1)}{d}}\right)\right\}(\beta_+ - \beta_-).$$

Proof. According to the definition of function $f_x(\cdot)$ in (2.4), it holds that

$$\begin{aligned} & \|\nabla f_{x_-}(\bar{y}_+) - \nabla f_{x_+}(\bar{y}_+)\| \\ = & \left\| \sum_{\ell=1}^d \frac{1}{(\ell-1)!} \left(\nabla^\ell f(x_+)[\bar{y}_+ - x_+]^{\ell-1} - \nabla^\ell f(x_-)[\bar{y}_+ - x_-]^{\ell-1} \right) \right. \\ & \left. + \frac{M}{d!} \left(\|\bar{y}_+ - x_-\|^{d-1}(y_+ - x_-) - \|\bar{y}_+ - x_+\|^{d-1}(\bar{y}_+ - x_+) \right) \right\| \\ \leq & \sum_{\ell=1}^d \frac{1}{(\ell-1)!} \left\| \nabla^\ell f(x_+)[\bar{y}_+ - x_+]^{\ell-1} - \nabla^\ell f(x_-)[\bar{y}_+ - x_-]^{\ell-1} \right\| \end{aligned} \quad (1.18)$$

$$+ \frac{M}{d!} \left\| \|\bar{y}_+ - x_-\|^{d-1}(y_+ - x_-) - \|\bar{y}_+ - x_+\|^{d-1}(\bar{y}_+ - x_+) \right\|. \quad (1.19)$$

Note that (1.19) can be further bounded as follows:

$$\begin{aligned}
& \frac{M}{d!} \left\| \|\bar{y}_+ - x_-\|^{d-1}(\bar{y}_+ - x_-) - \|\bar{y}_+ - x_+\|^{d-1}(\bar{y}_+ - x_+) \right\| \\
= & \frac{M}{d!} \left\| (\|\bar{y}_+ - x_-\|^{d-1} - \|\bar{y}_+ - x_+\|^{d-1})(\bar{y}_+ - x_-) + \|\bar{y}_+ - x_+\|^{d-1}(x_+ - x_-) \right\| \\
\leq & \frac{M}{d!} \left((d-1) \left| \|\bar{y}_+ - x_-\| - \|\bar{y}_+ - x_+\| \right| \max \{ \|\bar{y}_+ - x_-\|, \|\bar{y}_+ - x_+\| \}^{d-2} \|\bar{y}_+ - x_-\| \right. \\
& \left. + \|\bar{y}_+ - x_+\|^{d-1} \|x_+ - x_-\| \right) \\
\leq & \frac{M}{d!} \left((d-1) \max \{ \|\bar{y}_+ - x_-\|, \|\bar{y}_+ - x_+\| \}^{d-2} \|\bar{y}_+ - x_-\| + \|\bar{y}_+ - x_+\|^{d-1} \right) \|x_+ - x_-\| \\
\leq & \frac{M}{d!} \left(d(\|\bar{y}_+ - x_+\| + \|x_+ - x_-\|)^{d-1} \right) \left(2 + \frac{2}{\sqrt{1-\sigma^2}} \right) D(\beta_+ - \beta_-) \\
\leq & \max \left\{ \Theta[(\bar{\epsilon}^{-1})^{d-1}], \Theta[(\bar{\rho}^{-1})^{\frac{(d+1)(d-1)}{d}}] \right\} (\beta_+ - \beta_-) \tag{1.20}
\end{aligned}$$

where the first inequality is due to (4.16), and the second last inequality is from Lemma A.5, and that $\|\bar{y}_+ - x_-\| \leq \|\bar{y}_+ - x_+\| + \|x_+ - x_-\|$.

It remains to bound (1.18). We first show by induction that for $1 \leq \ell \leq d$ and any convex combination of x_-, x_+ and x_* denoted by z ,

$$\|\nabla^\ell f(z)\| \leq \Theta(1), \tag{1.21}$$

$$\|\nabla^\ell f(x_+) - \nabla^\ell f(x_-)\| \leq \Theta(1)(\beta_+ - \beta_-). \tag{1.22}$$

Our induction works backwardly starting from the base case: $\ell = d$. Recall that z is a convex combination of x_-, x_+ and x_* . By (1.17) we have

$$\|z - x_*\| \leq \max \{ \|x_+ - x_*\|, \|x_- - x_*\| \} \leq \Theta(1).$$

Therefore,

$$\begin{aligned}
\|\nabla^d f(z)\| & \leq \|\nabla^d f(x_*)\| + \|\nabla^d f(x_*) - \nabla^d f(z)\| \\
& \leq \|\nabla^d f(x_*)\| + L_d \|z - x_*\| \\
& \leq \Theta(1).
\end{aligned}$$

Moreover, by invoking (2.2) and Lemma A.5, we have

$$\begin{aligned}
\|\nabla^d f(x_+) - \nabla^d f(x_-)\| & \leq L_d \|x_+ - x_-\| \\
& \leq L_d \left(2 + \frac{2}{\sqrt{1-\sigma^2}} \right) D(\beta_+ - \beta_-) \leq \Theta(1)(\beta_+ - \beta_-).
\end{aligned}$$

Now suppose that the conclusion holds for some $\ell + 1$. Consider

$$z = t_1 x_- + t_2 x_+ + (1 - t_1 - t_2) x_*, \quad \forall 0 \leq t_1, t_2 \leq 1.$$

Denote $D_1 := \left(2 + \frac{2}{\sqrt{1-\sigma^2}}\right)D$. By letting $x_t = \frac{t_1}{t_1+t_2}x_- + \frac{t_2}{t_1+t_2}x_+$ and (1.17), we have $\|x_t - x_*\| \leq \|x_- - x_*\| + \|x_+ - x_*\| \leq \left(2 + \frac{2}{\sqrt{1-\sigma^2}}\right)D = D_1$. Consequently,

$$\begin{aligned} \|\nabla^\ell f(z) - \nabla^\ell f(x_*)\| &= \|\nabla^\ell f(x_* + (t_1 + t_2)(x_t - x_*)) - \nabla^\ell f(x_*)\| \\ &= \left\| \int_0^{t_1+t_2} \nabla^{\ell+1} f(x_* + u(x_t - x_*)) [x_t - x_*] du \right\| \\ &\leq \int_0^{t_1+t_2} \|\nabla^{\ell+1} f(x_* + u(x_t - x_*))\| \|x_t - x_*\| du \\ &\leq (t_1 + t_2)D_1\Theta(1), \end{aligned}$$

where the second last inequality is due to (2.1) and the last inequality follows from the induction hypothesis on (1.21). Then, it follows that

$$\|\nabla^\ell f(z)\| \leq \|\nabla^\ell f(x_*)\| + (t_1 + t_2)D_1\Theta(1) \leq \Theta(1).$$

Now by induction on (1.21), applying Lemma A.5 and using (2.1) we have

$$\begin{aligned} \|\nabla^\ell f(x_+) - \nabla^\ell f(x_-)\| &= \left\| \int_0^1 \nabla^{\ell+1} f(x_- + t(x_+ - x_-)) [x_+ - x_-] dt \right\| \\ &\leq \Theta(1)D_1(\beta_+ - \beta_-) \\ &\leq \Theta(1)(\beta_+ - \beta_-). \end{aligned}$$

Therefore, by induction it follows that (1.21) and (1.22) hold for any $1 \leq \ell \leq d$.

Now we come back to bound (1.18). For $2 \leq \ell \leq d$,

$$\begin{aligned} &\left\| \nabla^\ell f(x_+) [\bar{y}_+ - x_+]^{\ell-1} - \nabla^\ell f(x_-) [\bar{y}_+ - x_-]^{\ell-1} \right\| \\ &\leq \left\| \nabla^\ell f(x_+) [\bar{y}_+ - x_+]^{\ell-1} - \nabla^\ell f(x_+) [\bar{y}_+ - x_-]^{\ell-1} \right\| \\ &\quad + \left\| \nabla^\ell f(x_+) [\bar{y}_+ - x_-]^{\ell-1} - \nabla^\ell f(x_-) [\bar{y}_+ - x_-]^{\ell-1} \right\|. \end{aligned} \tag{1.23}$$

Applying Lemma A.5 and (1.21), the first term on the right hand side of (1.23) can be further upper bounded as follows:

$$\begin{aligned} &\left\| \nabla^\ell f(x_+) [\bar{y}_+ - x_+]^{\ell-1} - \nabla^\ell f(x_+) [\bar{y}_+ - x_-]^{\ell-1} \right\| \\ &= \left\| \sum_{j=1}^{\ell-1} \nabla^\ell f(x_+) \left[[\bar{y}_+ - x_+]^{j-1} [x_- - x_+] [\bar{y}_+ - x_-]^{\ell-j-1} \right] \right\| \\ &\leq \sum_{j=1}^{\ell-1} \|\nabla^\ell f(x_+)\| \|\bar{y}_+ - x_+\|^{j-1} \|\bar{y}_+ - x_-\|^{\ell-j-1} \|x_+ - x_-\| \\ &\leq \sum_{j=1}^{\ell-1} \Theta(1) \|\bar{y}_+ - x_+\|^{j-1} \left(\|x_+ - \bar{y}_+\| + \|x_- - x_+\| \right)^{\ell-j-1} \|x_+ - x_-\| \\ &\leq (\ell-1)\Theta(1) \left(\|x_+ - x_-\| + \|x_+ - \bar{y}_+\| \right)^{\ell-2} D_1(\beta_+ - \beta_-) \\ &\leq \max \left\{ \Theta[(\bar{\epsilon}^{-1})^{\ell-2}], \Theta[(\bar{\rho}^{-1})^{\frac{(d+1)(\ell-2)}{d}}] \right\} (\beta_+ - \beta_-). \end{aligned} \tag{1.24}$$

Moreover, applying Lemma A.5, (1.21) and (2.1) to the second term on the right hand side of (1.23) gives that

$$\begin{aligned}
& \left\| \nabla^\ell f(x_+) [\bar{y}_+ - x_-]^{\ell-1} - \nabla^\ell f(x_-) [\bar{y}_+ - x_-]^{\ell-1} \right\| \\
& \leq \left\| \nabla^\ell f(x_+) - \nabla^\ell f(x_-) \right\| \|\bar{y}_+ - x_-\|^{\ell-1} \\
& \leq \Theta(1)(\beta_+ - \beta_-) \left(\|x_+ - x_-\| + \|x_+ - \bar{y}_+\| \right)^{\ell-1} \\
& \leq \max \left\{ \Theta[(\bar{\epsilon}^{-1})^{\ell-1}], \Theta[(\bar{\rho}^{-1})^{\frac{(d+1)(\ell-1)}{d}}] \right\} (\beta_+ - \beta_-). \tag{1.25}
\end{aligned}$$

Putting (1.23), (1.24) and (1.25) together yields

$$\begin{aligned}
& \left\| \nabla^\ell f(x_+) [y_+ - x_+]^{\ell-1} - \nabla^\ell f(x_-) [y_+ - x_-]^{\ell-1} \right\| \\
& \leq \max \left\{ \Theta[(\bar{\epsilon}^{-1})^{\ell-1}], \Theta[(\bar{\rho}^{-1})^{\frac{(d+1)(\ell-1)}{d}}] \right\} (\beta_+ - \beta_-)
\end{aligned}$$

for $\ell = 2, \dots, d$. When $\ell = 1$, (1.22) guarantees that

$$\left\| \nabla f(x_+) - \nabla f(x_-) \right\| \leq \Theta(1)(\beta_+ - \beta_-).$$

Therefore, the quantity in (1.18) can be bounded as

$$\begin{aligned}
& \sum_{\ell=1}^d \frac{1}{(\ell-1)!} \left\| \nabla^\ell f(x_+) [\bar{y}_+ - x_+]^{\ell-1} - \nabla^\ell f(x_-) [\bar{y}_+ - x_-]^{\ell-1} \right\| \\
& \leq \max \left\{ \Theta[(\bar{\epsilon}^{-1})^{d-1}], \Theta[(\bar{\rho}^{-1})^{\frac{(d+1)(d-1)}{d}}] \right\} (\beta_+ - \beta_-). \tag{1.26}
\end{aligned}$$

Finally, replacing (1.19) and (1.18) with (1.20) and (1.26) respectively leads to the desired conclusion. \square