

A Newton-CG based barrier-augmented Lagrangian method for general nonconvex conic optimization

Chuan He*

Heng Huang†

Zhaosong Lu‡

April 2, 2023 (Revised: March 4, 2024; August 8, 2024; August 25, 2024)

Abstract

In this paper we consider finding an approximate second-order stationary point (SOSP) of general nonconvex conic optimization that minimizes a twice differentiable function subject to nonlinear equality constraints and also a convex conic constraint. In particular, we propose a Newton-conjugate gradient (Newton-CG) based barrier-augmented Lagrangian method for finding an approximate SOSP of this problem. Under some mild assumptions, we show that our method enjoys a total inner iteration complexity of $\tilde{\mathcal{O}}(\epsilon^{-11/2})$ and an operation complexity of $\tilde{\mathcal{O}}(\epsilon^{-11/2} \min\{n, \epsilon^{-5/4}\})$ for finding an $(\epsilon, \sqrt{\epsilon})$ -SOSP of general nonconvex conic optimization with high probability. Moreover, under a constraint qualification, these complexity bounds are improved to $\tilde{\mathcal{O}}(\epsilon^{-7/2})$ and $\tilde{\mathcal{O}}(\epsilon^{-7/2} \min\{n, \epsilon^{-3/4}\})$, respectively. To the best of our knowledge, this is the first study on the complexity of finding an approximate SOSP of general nonconvex conic optimization. Preliminary numerical results are presented to demonstrate superiority of the proposed method over first-order methods in terms of solution quality.

Keywords: Nonconvex conic optimization, second-order stationary point, augmented Lagrangian method, barrier method, Newton-conjugate gradient method, iteration complexity, operation complexity

Mathematics Subject Classification: 49M05, 49M15, 68Q25, 90C26, 90C30, 90C60

1 Introduction

In this paper we consider the following general nonconvex conic optimization problem:

$$\min_x \{f(x) : c(x) = 0, x \in \mathcal{K}\}, \quad (1)$$

where $\mathcal{K} \subseteq \mathbb{R}^n$ is a closed and pointed convex cone with a nonempty interior, and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are continuous in \mathcal{K} and twice continuously differentiable in the interior of \mathcal{K} . Assume that problem (1) has at least one optimal solution. Our goal is to propose an implementable method with complexity guarantees for finding an approximate second-order stationary point (SOSP) of (1) that will be introduced in Section 3.

In recent years, there has been considerable research on designing algorithms with complexity guarantees for finding an approximate SOSP of nonconvex optimization problems. In particular, numerous algorithms were developed for nonconvex unconstrained optimization, such as cubic regularized Newton methods [1, 18, 21, 57], trust-region methods [34, 35, 53], quadratic regularization method [14], accelerated gradient method [19, 20], second-order line-search method [61], Newton-conjugate gradient (Newton-CG) method [60], and gradient-based methods with random perturbations [2, 46, 71]. In addition, several methods with complexity guarantees have also been proposed for nonconvex optimization with relatively simple constraints. For example, interior-point method [10], log-barrier method [58], and projected gradient descent method [69] were proposed for nonconvex

*Department of Mathematics, Linköping University, Sweden (email: chuan.he@liu.se).

†Department of Computer Science, University of Maryland, USA (email: heng@umd.edu). The work of this author was partially supported by NSF Award IIS-2211492.

‡Department of Industrial and Systems Engineering, University of Minnesota, USA (email: zhaosong@umn.edu). The work of this author was partially supported by NSF Award IIS-2211491.

optimization with sign constraints. Besides, the interior-point method [10] was generalized in [42] for nonconvex optimization with sign constraints and additional linear equality constraints. Also, a projected gradient descent method with random perturbations was proposed in [51] for nonconvex optimization with linear inequality constraints. Iteration complexity of these methods has been established for finding an approximate SOSP. Besides, operation complexity in terms of the total number of fundamental operations has been studied for the methods [1, 2, 18, 19, 20, 34, 46, 60, 61, 71].

Several methods, including trust-region methods [17, 31], sequential quadratic programming method [15], two-phase method [11, 27, 30], penalty method [40], and augmented Lagrangian (AL) methods [4, 12, 44, 63, 70], were proposed for finding an approximate SOSP of equality constrained optimization:

$$\min_x \{f(x) : c(x) = 0\}, \quad (2)$$

which is special case of (1) with $\mathcal{K} = \mathbb{R}^n$. Moreover, total inner iteration complexity and operation complexity, which are respectively measured by the total number of iterations of the Newton-CG method in [60] and the total number of gradient evaluations and matrix-vector products performed in the method, were established in [44, 70] for finding an $(\epsilon, \sqrt{\epsilon})$ -SOSP x of (2) which together with some $\lambda \in \mathbb{R}^m$ satisfies

$$\begin{aligned} \|c(x)\| &\leq \epsilon, \quad \|\nabla f(x) + \nabla c(x)\lambda\| \leq \epsilon, \\ d^T(\nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 c_i(x))d &\geq -\sqrt{\epsilon} \|d\|^2, \quad \forall d \in \{d : \nabla c(x)^T d = 0\}, \end{aligned}$$

where ∇c denotes the transpose of the Jacobian of c . Specifically, under some suitable assumptions, including a generalized linear independence constraint qualification (GLICQ), the AL method [70] enjoys a total inner iteration complexity of $\tilde{\mathcal{O}}(\epsilon^{-11/2})$ and an operation complexity $\tilde{\mathcal{O}}(\epsilon^{-11/2} \min\{n, \epsilon^{-3/4}\})$,¹ while the AL method [44] achieves a total inner iteration complexity of $\tilde{\mathcal{O}}(\epsilon^{-7/2})$ and an operation complexity of $\tilde{\mathcal{O}}(\epsilon^{-7/2} \min\{n, \epsilon^{-3/4}\})$ for finding an $(\epsilon, \sqrt{\epsilon})$ -SOSP of problem (2) with high probability. On the other hand, when the GLICQ does not hold, the AL method [44] has a total inner iteration complexity of $\tilde{\mathcal{O}}(\epsilon^{-11/2})$ and an operation complexity of $\tilde{\mathcal{O}}(\epsilon^{-11/2} \min\{n, \epsilon^{-5/4}\})$. Besides, it shall be mentioned that Newton-CG based AL methods were developed for efficiently solving a variety of convex optimization problems (e.g., see [72, 73]), though their complexities remain unknown.

In addition, a Newton-CG based barrier method was recently proposed in [43] for finding an approximate SOSP of a class of nonconvex conic optimization of the form

$$\min_x \{f(x) : Ax - b = 0, x \in \mathcal{K}\} \quad (3)$$

for some $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, which is a special case of (1). Iteration and operation complexity of this method were established in [43] for finding an $(\epsilon, \sqrt{\epsilon})$ -SOSP x of (3) which together with some $\lambda \in \mathbb{R}^m$ satisfies

$$\begin{aligned} Ax = b, \quad x \in \text{int } \mathcal{K}, \quad \nabla f(x) + A^T \lambda \in \mathcal{K}^*, \quad \|\nabla f(x) + A^T \lambda\|_x^* &\leq \epsilon, \\ d^T \nabla^2 B(x)^{-1/2} \nabla^2 f(x) \nabla^2 B(x)^{-1/2} d &\geq -\sqrt{\epsilon} \|d\|^2, \quad \forall d \in \{d : A \nabla^2 B(x)^{-1/2} d = 0\}, \end{aligned}$$

where $\text{int } \mathcal{K}$ and \mathcal{K}^* are respectively the interior and dual cone of \mathcal{K} , B is a logarithmically homogeneous self-concordant barrier function for \mathcal{K} , and $\|\cdot\|_x^*$ is a local norm induced by B at x (see Section 2 for details). Under some suitable assumptions, this method achieves an iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ and an operation complexity² of $\tilde{\mathcal{O}}(\epsilon^{-3/2} \min\{n, \epsilon^{-1/4}\})$ for finding an $(\epsilon, \sqrt{\epsilon})$ -SOSP with high probability. Besides, a Hessian barrier algorithm was proposed in [38] for finding an approximate SOSP of problem (3). Given that this algorithm requires solving a cubic regularized subproblem exactly per iteration, it is generally expensive to implement.

It shall also be mentioned that finding an approximate *first-order stationary point* of (1) with $\mathcal{K} = \mathbb{R}_+^n$ was extensively studied in the literature (e.g., [5, 6, 7, 32, 33, 37, 39, 49, 55, 65, 66, 67]). Notably, a hybrid approach

¹In fact, a total inner iteration complexity of $\tilde{\mathcal{O}}(\epsilon^{-7})$ and an operation complexity $\tilde{\mathcal{O}}(\epsilon^{-7} \min\{n, \epsilon^{-1}\})$ were established in [70] for finding an (ϵ, ϵ) -SOSP of problem (1) with high probability; see [70, Theorem 4(ii), Corollary 3(ii), Theorem 5]. Nevertheless, they can be easily modified to obtain the aforementioned complexity for finding an $(\epsilon, \sqrt{\epsilon})$ -SOSP of (1) with high probability.

²The operation complexity of the barrier method [43] is measured by the amount of fundamental operations consisting of matrix-vector products, matrix multiplications, Cholesky factorizations, and backward or forward substitutions to a triangular linear system.

by combining barrier and AL methods was commonly used in [5, 6, 7, 33, 37, 39, 49, 55]). However, finding an approximate SOSP of (1) by such a hybrid approach has not been considered, even for (1) with $\mathcal{K} = \mathbb{R}_+^n$. Inspired by these and [43, 44], in this paper we propose a Newton-CG based barrier-AL method for finding an approximate SOSP of problem (1) with high probability. Our main contributions are as follows.

- We study first- and second-order optimality conditions for problem (1) and introduce an approximate counterpart of them.
- We propose an implementable Newton-CG based barrier-AL method for finding an approximate SOSP of (1), whose main operations consist of Cholesky factorizations and other fundamental operations including matrix-vector products and backward or forward substitutions to a triangular linear system.³
- We show that under some mild assumptions, our proposed method has a total inner iteration complexity of $\tilde{\mathcal{O}}(\epsilon^{-11/2})$ and an operation complexity of $\tilde{\mathcal{O}}(\epsilon^{-11/2} \min\{n, \epsilon^{-5/4}\})$ for finding an $(\epsilon, \sqrt{\epsilon})$ -SOSP of (1) with high probability. Furthermore, under a constraint qualification, we show that our method achieves an improved total inner iteration complexity of $\tilde{\mathcal{O}}(\epsilon^{-7/2})$ and an improved operation complexity of $\tilde{\mathcal{O}}(\epsilon^{-7/2} \min\{n, \epsilon^{-3/4}\})$.⁴ To the best of our knowledge, there was no complexity result for finding an approximate SOSP of problem (1) in the literature before.

The rest of this paper is organized as follows. In Section 2, we introduce some notation. In Section 3, we study optimality conditions of problem (1) and introduce an inexact counterpart of them. In Section 4, we propose a preconditioned Newton-CG method for solving a barrier problem and study its complexity. We then propose a Newton-CG based barrier-AL method for (1) and study its complexity in Section 5. We present in Section 6 some preliminary numerical results for the proposed method. In Section 7, we present the proofs of the main results. Finally, we make some concluding remarks in Section 8.

2 Notation and preliminaries

Throughout this paper, we let \mathbb{R}^n denote the n -dimensional Euclidean space. The symbol $\|\cdot\|$ stands for the Euclidean norm of a vector or the spectral norm of a matrix. The identity matrix is denoted by I . We denote by $\lambda_{\min}(H)$ the minimum eigenvalue of a real symmetric matrix H . For any two real symmetric matrices M_1 and M_2 , $M_1 \preceq M_2$ means that $M_2 - M_1$ is positive semidefinite. For any positive semidefinite matrix M , $M^{1/2}$ denotes a positive semidefinite matrix such that $M = M^{1/2}M^{1/2}$. For the closed convex cone \mathcal{K} , its interior and dual cone are respectively denoted by $\text{int } \mathcal{K}$ and \mathcal{K}^* . For any $x \in \mathcal{K}$, the normal cone and tangent cone of \mathcal{K} at x are denoted by $\mathcal{N}_{\mathcal{K}}(x)$ and $\mathcal{T}_{\mathcal{K}}(x)$, respectively. The Euclidean ball centered at the origin with radius $R \geq 0$ is denoted by $\mathcal{B}_R := \{x : \|x\| \leq R\}$, and we use $\Pi_{\mathcal{B}_R}(v)$ to denote the Euclidean projection of a vector v onto \mathcal{B}_R . For a given finite set \mathcal{A} , we let $|\mathcal{A}|$ denote its cardinality. For any $s \in \mathbb{R}$, we let $\text{sgn}(s)$ be 1 if $s \geq 0$ and let it be -1 otherwise. In addition, $\tilde{\mathcal{O}}(\cdot)$ represents $\mathcal{O}(\cdot)$ with logarithmic terms omitted.

Logarithmically homogeneous self-concordant (LHSC) barrier function is a key ingredient in the development of interior-point methods for convex programming (see the monograph [56]). It will also play a crucial role in the design and analysis of Newton-CG based barrier-AL method for solving problem (1). *Throughout this paper, we assume that the cone \mathcal{K} is equipped with a ϑ -logarithmically homogeneous self-concordant (ϑ -LHSC) barrier function B for some $\vartheta \geq 1$.* That is, $B : \text{int } \mathcal{K} \rightarrow \mathbb{R}$ satisfies the following conditions:

- (i) B is convex and three times continuously differentiable in $\text{int } \mathcal{K}$, and moreover, $|\psi'''(0)| \leq 2(\psi''(0))^{3/2}$ holds for all $x \in \text{int } \mathcal{K}$ and $u \in \mathbb{R}^n$, where $\psi(t) = B(x + tu)$;
- (ii) B is a barrier function for \mathcal{K} , that is, $B(x)$ goes to infinity as x approaches the boundary of \mathcal{K} ;
- (iii) B is logarithmically homogeneous, that is, $B(tx) = B(x) - \vartheta \ln t$ holds for all $x \in \text{int } \mathcal{K}$ and $t > 0$.

³The arithmetic complexity of Cholesky factorizations for a positive definite matrix is $\mathcal{O}(n^3)$ in general, while the arithmetic complexity of matrix-vector products and backward or forward substitutions is at most $\mathcal{O}(n^2)$, where n is the number of rows of the matrix.

⁴It shall be mentioned that the total numbers of Cholesky factorizations are only $\tilde{\mathcal{O}}(\epsilon^{-7/2})$ and $\tilde{\mathcal{O}}(\epsilon^{-11/2})$ respectively for the case where constraint qualification holds or not. See Subsections 5.3 and 5.4 for details.

For any $x \in \text{int } \mathcal{K}$, the function B induces the following local norms:

$$\begin{aligned}\|v\|_x &:= (v^T \nabla^2 B(x)v)^{1/2}, \quad \forall v \in \mathbb{R}^n, \\ \|v\|_x^* &:= (v^T \nabla^2 B(x)^{-1}v)^{1/2}, \quad \forall v \in \mathbb{R}^n, \\ \|M\|_x^* &:= \max_{\|v\|_x \leq 1} \|Mv\|_x^*, \quad \forall M \in \mathbb{R}^{n \times n}.\end{aligned}\tag{4}$$

In addition, $\nabla^2 B(x)^{-1}$ is well-defined only in $\text{int } \mathcal{K}$ but undefined on the boundary of \mathcal{K} . To capture the behavior of $\nabla^2 B(x)^{-1}$ as x approaches the boundary of \mathcal{K} , the concept of *the limiting inverse of the Hessian of B* was recently introduced in [43], which can be viewed as a generalization of $[\nabla^2 B]^{-1}$. Specifically, the limiting inverse of the Hessian of B is defined as follows:

$$\nabla^{-2} B(x) := \left\{ M : M = \lim_{k \rightarrow \infty} \nabla^2 B(x^k)^{-1} \text{ for some } \{x^k\} \subset \text{int } \mathcal{K} \text{ with } x^k \rightarrow x \text{ as } k \rightarrow \infty \right\}, \quad \forall x \in \mathcal{K}.$$

As established in [43, Theorem 1], the inverse of $\nabla^2 B(x)$ is bounded in any nonempty bounded subset of $\text{int } \mathcal{K}$. Consequently, $\nabla^{-2} B(x) \neq \emptyset$ for all $x \in \mathcal{K}$. Moreover, the following property holds for $\nabla^{-2} B$, whose proof can be found in [43, Theorem 2].

Lemma 2.1. *For any $x \in \mathcal{K}$, it holds that $\{x + M^{1/2}d : \|d\| < 1\} \subseteq \mathcal{K}$ for all $M \in \nabla^{-2} B(x)$.*

3 Optimality conditions

Classical first- and second-order optimality conditions for nonlinear optimization can be specialized to problem (1) (e.g., see [62, Theorems 3.38 and 3.46]). However, the inexact counterparts of them are not suitable for the design and analysis of a barrier-AL method for solving (1). In this section we study some alternative first- and second-order optimality conditions for (1) and also introduce an inexact counterpart of them.

Suppose that x^* is a local minimizer of problem (1). To derive optimality conditions, one typically needs to impose a constraint qualification (CQ) for x^* . The Robinson's CQ, $\{\nabla c(x^*)^T d : d \in \mathcal{T}_{\mathcal{K}}(x^*)\} = \mathbb{R}^m$, is a natural and general one (e.g., see [62, Section 3.3.2]). However, verification of Robinson's CQ may not be easy for a general cone \mathcal{K} . Thus, we instead consider a more easily verifiable CQ that $M^{1/2} \nabla c(x^*)$ has full column rank for some $M \in \nabla^{-2} B(x^*)$, which turns out to be stronger than Robinson's CQ. Indeed, suppose that such a CQ holds at x^* for some $M \in \nabla^{-2} B(x^*)$. It then follows from Lemma 2.1 that $\{M^{1/2} \tilde{d} : \|\tilde{d}\| < 1\} \subseteq \mathcal{T}_{\mathcal{K}}(x^*)$ and hence $\{M^{1/2} \tilde{d} : \tilde{d} \in \mathbb{R}^n\} \subseteq \mathcal{T}_{\mathcal{K}}(x^*)$. By this and the full column rank of $M^{1/2} \nabla c(x^*)$, one has

$$\{\nabla c(x^*)^T d : d \in \mathcal{T}_{\mathcal{K}}(x^*)\} \supseteq \{\nabla c(x^*)^T M^{1/2} \tilde{d} : \tilde{d} \in \mathbb{R}^n\} = \mathbb{R}^m,$$

and hence Robinson's CQ holds at x^* .

We are now ready to establish some first- and second-order optimality conditions for problem (1) under the aforementioned CQ, whose proof is relegated to Section 7.1.

Theorem 3.1 (first- and second-order optimality conditions). *Let x^* be a local minimizer of problem (1). Suppose that f is twice continuously differentiable at x^* and $M^{1/2} \nabla c(x^*)$ has full column rank for some $M \in \nabla^{-2} B(x^*)$. Then there exists a Lagrangian multiplier $\lambda^* \in \mathbb{R}^m$ such that*

$$\nabla f(x^*) + \nabla c(x^*) \lambda^* \in \mathcal{K}^*, \tag{5}$$

$$M^{1/2}(\nabla f(x^*) + \nabla c(x^*) \lambda^*) = 0, \tag{6}$$

and additionally,

$$d^T M^{1/2} \left(\nabla^2 f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla^2 c_i(x^*) \right) M^{1/2} d \geq 0, \quad \forall d \in \{d : \nabla c(x^*)^T M^{1/2} d = 0\}. \tag{7}$$

Remark 3.1. *The relations (5) and (6) are the first-order optimality conditions of problem (1), which are actually equivalent to the classical optimality condition $\nabla f(x^*) + \nabla c(x^*) \lambda^* \in -\mathcal{N}_{\mathcal{K}}(x^*)$ (see [43, Proposition 1]).*

Notice that it is generally impossible to find a point exactly satisfying the above first- and second-order optimality conditions. We are instead interested in finding a point satisfying their approximate counterparts. To this end, we next introduce the definition of an approximate first-order stationary point (FOSP) and second-order stationary point (SOSP) of problem (1).

Definition 3.1 (ϵ_1 -first-order stationary point). *For any $\epsilon_1 > 0$, a point x is called an ϵ_1 -first-order stationary point (ϵ_1 -FOSP) of problem (1) if it, together with some $\lambda \in \mathbb{R}^m$, satisfies*

$$\|c(x)\| \leq \epsilon_1, \quad x \in \text{int } \mathcal{K}, \quad (8)$$

$$\nabla f(x) + \nabla c(x)\lambda \in \mathcal{K}^*, \quad (9)$$

$$\|\nabla f(x) + \nabla c(x)\lambda\|_x^* \leq \epsilon_1. \quad (10)$$

Definition 3.2 ((ϵ_1, ϵ_2) -second-order stationary point). *For any $\epsilon_1, \epsilon_2 > 0$, a point x is called an (ϵ_1, ϵ_2) -second-order stationary point ((ϵ_1, ϵ_2) -SOSP) of problem (1) if it, together with some $\lambda \in \mathbb{R}^m$, satisfies (8)-(10) and additionally*

$$d^T \nabla^2 B(x)^{-1/2} \left(\nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 c_i(x) \right) \nabla^2 B(x)^{-1/2} d \geq -\epsilon_2 \|d\|^2, \quad \forall d \in \mathcal{C}(x), \quad (11)$$

where $\mathcal{C}(\cdot)$ is defined as

$$\mathcal{C}(x) := \{d : \nabla c(x)^T \nabla^2 B(x)^{-1/2} d = 0\}. \quad (12)$$

Remark 3.2. Notice that if the pair (x, λ) satisfies (10) and (11), then it nearly satisfies (6) and (7) with (x^*, λ^*) replaced by (x, λ) . Thus, (10) and (11) can be viewed as inexact counterparts of (6) and (7). Moreover, the above definitions of ϵ_1 -FOSP and (ϵ_1, ϵ_2) -SOSP are reduced to the ones introduced in [43] for the case where c is affine.

4 A preconditioned Newton-CG method for barrier problems

In this section we propose a preconditioned Newton-CG method in Algorithm 1, which is a modification of the Newton-CG based barrier method [43, Algorithm 2], for finding an approximate SOSP of the barrier problem

$$\min_x \{\phi_\mu(x) := F(x) + \mu B(x)\}, \quad (13)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice continuously differentiable in $\text{int } \mathcal{K}$ and $\mu > 0$ is a given barrier parameter. Specifically, the proposed method finds an (ϵ_g, ϵ_H) -SOSP x of problem (13) that satisfies

$$\|\nabla \phi_\mu(x)\|_x^* \leq \epsilon_g, \quad \lambda_{\min}(\nabla^2 B(x)^{-1/2} \nabla^2 \phi_\mu(x) \nabla^2 B(x)^{-1/2}) \geq -\epsilon_H \quad (14)$$

for any prescribed tolerances $\epsilon_g, \epsilon_H \in (0, 1)$. It will be used to solve the subproblems arising in the barrier-AL method later.

Our preconditioned Newton-CG method (Algorithm 1) consists of two main components. The first main component is a modified CG method, referred to as *capped CG method*, which was proposed in [60, Algorithm 1] for solving a possibly indefinite linear system

$$(H + 2\varepsilon I)\hat{d} = -g, \quad (15)$$

where $0 \neq g \in \mathbb{R}^n$, $\varepsilon > 0$, and $H \in \mathbb{R}^{n \times n}$ is a symmetric matrix. The capped CG method terminates within a finite number of iterations and returns either an approximate solution \hat{d} to (15) satisfying $\|(H + 2\varepsilon I)\hat{d} + g\| \leq \hat{\zeta}\|g\|$ and $\hat{d}^T H \hat{d} \geq -\varepsilon\|\hat{d}\|^2$ for some $\hat{\zeta} \in (0, 1)$ or a sufficiently negative curvature direction \hat{d} of H with $\hat{d}^T H \hat{d} < -\varepsilon\|\hat{d}\|^2$. The second main component is a minimum eigenvalue oracle. Given a symmetric matrix $H \in \mathbb{R}^{n \times n}$ and $\varepsilon > 0$, this oracle either produces a sufficiently negative curvature direction v of H with $\|v\| = 1$ and $v^T H v \leq -\varepsilon/2$ or certifies that $\lambda_{\min}(H) \geq -\varepsilon$ holds with high probability. For ease of reference, we present these two main components in Algorithms 3 and 4 in Appendices A and B, respectively.

We are now ready to describe our preconditioned Newton-CG method (Algorithm 1) for solving (13). At iteration t , if the first relation in (14) is not satisfied at the iterate x^t , the capped CG method (Algorithm 3) is invoked to find a descent direction for ϕ_μ by solving the following damped preconditioned Newton system

$$(M_t^T \nabla^2 \phi_\mu(x^t) M_t + 2\epsilon_H I) \hat{d} = -M_t^T \nabla \phi_\mu(x^t),$$

where M_t is a matrix such that

$$\nabla^2 B(x^t)^{-1} = M_t M_t^T. \quad (16)$$

A line search along this descent direction is then performed to result in a reduction on ϕ_μ . Otherwise, the minimum eigenvalue oracle (Algorithm 4) is invoked. This oracle either produces a sufficiently negative curvature direction of $M_t^T \nabla^2 \phi_\mu(x^t) M_t$ along which a line search is performed to result in a reduction on ϕ_μ , or certifies that the iterate x^t also satisfies the second relation in (14) with high probability and terminates the preconditioned Newton-CG method. The detailed description of our preconditioned Newton-CG method is presented in Algorithm 1.

Algorithm 1 A preconditioned Newton-CG method for problem (13)

Input: tolerances $\epsilon_g, \epsilon_H \in (0, 1)$, backtracking ratio $\theta \in (0, 1)$, starting point $u^0 \in \text{int } \mathcal{K}$, CG-accuracy parameter $\zeta \in (0, 1)$, maximum step length $\beta \in [\epsilon_H, 1)$, line-search parameter $\eta \in (0, 1)$, probability parameter $\delta \in (0, 1)$;

Set $x^0 = u^0$;

for $t = 0, 1, 2, \dots$ **do**

if $\|\nabla \phi_\mu(x^t)\|_{x^t}^* > \epsilon_g$ **then**

 Call Algorithm 3 (see Appendix A) with $H = M_t^T \nabla^2 \phi_\mu(x^t) M_t$, $\varepsilon = \epsilon_H$, $g = M_t^T \nabla \phi_\mu(x^t)$, accuracy parameter ζ , and bound $U = 0$ to obtain outputs \hat{d}^t , d_type, where M_t is given in (16);

if d_type=NC **then**

$$d^t \leftarrow -\text{sgn}((\hat{d}^t)^T M_t^T \nabla \phi_\mu(x^t)) \min \left\{ \frac{|(\hat{d}^t)^T M_t^T \nabla^2 \phi_\mu(x^t) M_t \hat{d}^t|}{\|\hat{d}^t\|^3}, \frac{\beta}{\|\hat{d}^t\|} \right\} \hat{d}^t; \quad (17)$$

else {d_type=SOL}

$$d^t \leftarrow \min \left\{ 1, \frac{\beta}{\|\hat{d}^t\|} \right\} \hat{d}^t; \quad (18)$$

end if

 Go to **Line Search**;

else

 Call Algorithm 4 (see Appendix B) with $H = M_t^T \nabla^2 \phi_\mu(x^t) M_t$, $\varepsilon = \epsilon_H$, and probability parameter δ ;

if Algorithm 4 certifies that $\lambda_{\min}(M_t^T \nabla^2 \phi_\mu(x^t) M_t) \geq -\epsilon_H$ **then**

 Output x^t and terminate;

else {Sufficiently negative curvature direction v returned by Algorithm 4}

 Set d_type=NC and

$$d^t \leftarrow -\text{sgn}(v^T M_t^T \nabla \phi_\mu(x^t)) \min\{|v^T M_t^T \nabla^2 \phi_\mu(x^t) M_t v|, \beta\} v; \quad (19)$$

 Go to **Line Search**;

end if

end if

Line Search:

if d_type=SOL **then**

 Find $\alpha_t = \theta^{j_t}$, where j_t is the smallest nonnegative integer j such that

$$\phi_\mu(x^t + \theta^j M_t d^t) < \phi_\mu(x^t) - \eta \epsilon_H \theta^{2j} \|d^t\|^2; \quad (20)$$

else {d_type=NC}

 Find $\alpha_t = \theta^{j_t}$, where j_t is the smallest nonnegative integer j such that

$$\phi_\mu(x^t + \theta^j M_t d^t) < \phi_\mu(x^t) - \eta \theta^{2j} \|d^t\|^3 / 2; \quad (21)$$

end if

$x^{t+1} = x^t + \alpha_t M_t d^t$;

end for

4.1 Iteration and operation complexity of Algorithm 1

In this subsection we study iteration and operation complexity of Algorithm 1. To proceed, we make the following assumptions on problem (13).

Assumption 4.1. (a) *There exists a finite ϕ_{low} such that*

$$\phi_\mu(x) \geq \phi_{\text{low}}, \quad \forall x \in \text{int } \mathcal{K}, \quad (22)$$

$$\mathcal{S} = \{x \in \text{int } \mathcal{K} : \phi_\mu(x) \leq \phi_\mu(u^0)\} \text{ is bounded}, \quad (23)$$

where $u^0 \in \text{int } \mathcal{K}$ is the initial point of Algorithm 1 and ϕ_μ is given in (13).

(b) *There exists $L_H^F > 0$ such that*

$$\|\nabla^2 F(y) - \nabla^2 F(x)\|_x^* \leq L_H^F \|y - x\|_x, \quad \forall x, y \in \Omega \text{ with } \|y - x\|_x \leq \beta,$$

where $\Omega \subset \text{int } \mathcal{K}$ is an open bounded convex neighborhood of \mathcal{S} and $\beta \in (0, 1)$ is an input of Algorithm 1.

(c) *The quantities U_g^F and U_H^F are finite, where*

$$U_g^F := \sup_{x \in \mathcal{S}} \|\nabla F(x)\|_x^*, \quad U_H^F := \sup_{x \in \mathcal{S}} \|\nabla^2 F(x)\|_x^*. \quad (24)$$

Before establishing operation complexity of Algorithm 1, let us make some observations on its fundamental operations. Firstly, at iteration t , the main effort of Algorithm 1 is on the execution of Algorithm 3 or 4 with $H = M_t^T \nabla^2 \phi_\mu(x^t) M_t$. Secondly, the main computational cost of Algorithms 3 and 4 per iteration is on the product of H and a vector v . Consequently, it suffices to focus on computing Hv . Indeed, notice from (13) and (16) that

$$Hv = M_t^T \nabla^2 \phi_\mu(x^t) M_t v = M_t^T \nabla^2 F(x^t) M_t v + \mu v.$$

Thus, computing Hv consists of one Hessian-vector product of F and two matrix-vector products involving M_t and M_t^T , respectively. We next discuss how to efficiently compute the product of M_t or M_t^T and a vector.

- When \mathcal{K} is the nonnegative orthant, its associated barrier function is $B(x) = -\sum_{i=1}^n \ln x_i$. Notice that $\nabla^2 B(x)$ is a diagonal matrix and so is M_t . As a result, the operation cost for computing the product of M_t or M_t^T and a vector is $\mathcal{O}(n)$, which is typically cheaper than the Hessian-vector product of F .
- When \mathcal{K} is a general cone, directly computing M_t may be too expensive. In view of $\nabla^2 B(x^t) = M_t^{-T} M_t^{-1}$ (see (16)), one can instead choose M_t^{-T} as the Cholesky factor of $\nabla^2 B(x^t)$, which is computed only once in each iteration of Algorithm 1. Once M_t^{-T} is available, the product of M_t or M_t^T and a vector can be computed by performing backward or forward substitution to a linear system with coefficient matrix M_t^{-1} or M_t^{-T} .

Based on the above discussion, we conclude that: (i) when \mathcal{K} is the nonnegative orthant, the fundamental operations of Algorithm 1 consist only of the Hessian-vector products of F ; (ii) when \mathcal{K} is a general cone, the fundamental operations of Algorithm 1 consist of the Hessian-vector products of F , Cholesky factorizations of $\nabla^2 B$, and backward or forward substitutions to a triangular linear system.

The following theorem states the iteration and operation complexity of Algorithm 1, whose proof is deferred to Section 7.2.

Theorem 4.1 (Complexity of Algorithm 1). *Suppose that Assumption 4.1 holds. Let*

$$T_1 = \left\lceil \frac{\phi_{\text{hi}} - \phi_{\text{low}}}{\min\{c_{\text{sol}}, c_{\text{nc}}\}} \max\{\epsilon_g^{-2} \epsilon_H, \epsilon_H^{-3}\} \right\rceil + \left\lceil \frac{\phi_{\text{hi}} - \phi_{\text{low}}}{c_{\text{nc}}} \epsilon_H^{-3} \right\rceil + 1, \quad T_2 = \left\lceil \frac{\phi_{\text{hi}} - \phi_{\text{low}}}{c_{\text{nc}}} \epsilon_H^{-3} \right\rceil + 1, \quad (25)$$

where $\phi_{\text{hi}} = \phi_\mu(u^0)$, ϕ_{low} is given in (22), and

$$c_{\text{sol}} = \eta \min \left\{ \left[\frac{4(1-\beta)}{4+\zeta+\sqrt{(4+\zeta)^2+8[(1-\beta)L_H^F+\mu(2-\beta)/(1-\beta)]}} \right]^2, \left[\frac{\min\{6(1-\eta), 2\}\theta}{L_H^F+\mu(2-\beta)/(1-\beta)^2} \right]^2 \right\}, \quad (26)$$

$$c_{\text{nc}} = \frac{\eta}{16} \min \left\{ 1, \left[\frac{\min\{3(1-\eta), 1\}\theta}{L_H^F+\mu(2-\beta)/(1-\beta)^2} \right]^2 \right\}. \quad (27)$$

Then the following statements hold.

- (i) The total number of calls of Algorithm 4 in Algorithm 1 is at most T_2 .
- (ii) The total number of calls of Algorithm 3 in Algorithm 1 is at most T_1 .
- (iii) (iteration complexity) Algorithm 1 terminates in at most $T_1 + T_2$ iterations with

$$T_1 + T_2 = \mathcal{O}((\phi_{\text{hi}} - \phi_{\text{low}})(L_H^F)^2 \max\{\epsilon_g^{-2}\epsilon_H, \epsilon_H^{-3}\}). \quad (28)$$

Moreover, its output x^t satisfies the first relation in (14) deterministically and the second relation in (14) with probability at least $1 - \delta$ for some $0 \leq t \leq T_1 + T_2$.

- (iv) (operation complexity) The total numbers of Cholesky factorizations and other fundamental operations consisting of the Hessian-vector products of F and backward or forward substitutions to a triangular linear system required by Algorithm 1 are at most $T_1 + T_2$ and

$$\tilde{\mathcal{O}}((\phi_{\text{hi}} - \phi_{\text{low}})(L_H^F)^2 \max\{\epsilon_g^{-2}\epsilon_H, \epsilon_H^{-3}\} \min\{n, (U_H^F/\epsilon_H)^{1/2}\}),$$

respectively, where U_H^F is given in (24).

5 A Newton-CG based barrier-AL method for problem (1)

In this section we propose a Newton-CG based barrier-AL method for finding a stochastic $(\epsilon, \sqrt{\epsilon})$ -SOSP of problem (1) for any prescribed tolerance $\epsilon \in (0, 1)$.

Recall that B is the ϑ -LHSC barrier function associated with \mathcal{K} for some $\vartheta \geq 1$. We now make the following additional assumptions on problem (1).

Assumption 5.1. (a) An $\epsilon/2$ -approximately strictly feasible point z_ϵ of problem (1), namely satisfying $z_\epsilon \in \text{int } \mathcal{K}$ and $\|c(z_\epsilon)\| \leq \epsilon/2$, is known.

(b) There exist constants $f_{\text{hi}}, f_{\text{low}} \in \mathbb{R}$ and $\gamma, \delta_f, \delta_c > 0$, independent of ϵ , such that

$$f(z_\epsilon) + \mu B(z_\epsilon) \leq f_{\text{hi}}, \quad \forall \mu \in (0, \mu_0], \quad (29)$$

$$f(x) + \mu B(x) + \gamma \|c(x)\|^2/2 \geq f_{\text{low}}, \quad \forall \mu \in (0, \mu_0], x \in \text{int } \mathcal{K}, \quad (30)$$

$$\mathcal{S}(\delta_f, \delta_c) := \bigcup_{\mu \in (0, \mu_0]} \{x \in \text{int } \mathcal{K} : f(x) + \mu B(x) \leq f_{\text{hi}} + \delta_f, \|c(x)\| \leq 1 + \delta_c\} \text{ is bounded,} \quad (31)$$

where $\mu_0 = 1/(2\vartheta^{1/2} + 2)$ and z_ϵ is given in (a).

(c) There exist $L_H^f, L_H^c > 0$ and $\beta \in (0, 1)$ such that

$$\begin{aligned} \|\nabla^2 f(y) - \nabla^2 f(x)\|_x^* &\leq L_H^f \|y - x\|_x, \quad \forall x, y \in \Omega(\delta_f, \delta_c) \text{ with } \|y - x\|_x \leq \beta, \\ \|\nabla^2 c_i(y) - \nabla^2 c_i(x)\|_x^* &\leq L_H^c \|y - x\|_x, \quad \forall x, y \in \Omega(\delta_f, \delta_c) \text{ with } \|y - x\|_x \leq \beta, \quad 1 \leq i \leq m, \end{aligned} \quad (32)$$

where $\Omega(\delta_f, \delta_c) \subset \text{int } \mathcal{K}$ is an open bounded convex neighborhood of $\mathcal{S}(\delta_f, \delta_c)$.

(d) The quantities U_g^f, U_g^c, U_H^f and U_H^c are finite, where

$$U_g^f = \sup_{x \in \Omega(\delta_f, \delta_c)} \|\nabla f(x)\|_x^*, \quad U_g^c = \sup_{x \in \Omega(\delta_f, \delta_c)} \max_{1 \leq i \leq m} \|\nabla c_i(x)\|_x^*, \quad (33)$$

$$U_H^f = \sup_{x \in \Omega(\delta_f, \delta_c)} \|\nabla^2 f(x)\|_x^*, \quad U_H^c = \sup_{x \in \Omega(\delta_f, \delta_c)} \max_{1 \leq i \leq m} \|\nabla^2 c_i(x)\|_x^*. \quad (34)$$

We next make some remarks about Assumption 5.1.

Remark 5.1. (i) A similar assumption as Assumption 5.1(a) was considered in the study of AL methods for nonconvex equality constrained optimization (e.g., see [28, 41, 44, 52, 70]). By imposing Assumption 5.1(a), we restrict our study on problem (1) for which an $\epsilon/2$ -approximately strictly feasible point z_ϵ can be found by an inexpensive procedure. Assumption 5.1(a) often holds in practice. For example, when the constraints of (1) consist of sphere and nonnegative orthant constraints, a strictly feasible point is readily

available. Also, when c is an affine mapping and \mathcal{K} is the nonnegative orthant, a strictly feasible point of (1) can be found using interior point methods. In addition, when the generalized LICQ condition $\lambda_{\min}(\nabla c(x)^T \nabla^2 B(x)^{-1} \nabla c(x)) \geq \sigma^2$ (see Assumption 5.2 below) holds on a level set of $\|c(x)\|^2 + \mu B(x)$ for $\mu = \sigma\epsilon/(2\vartheta^{1/2})$ and some constant $\sigma > 0$ and $F(x) = \|c(x)\|^2$ satisfies Assumption 4.1, the point z_ϵ can be found by applying our preconditioned Newton-CG method (Algorithm 1) to the barrier problem $\min_x \|c(x)\|^2 + \mu B(x)$ to find z_ϵ satisfying $\|\nabla(\|c(z_\epsilon)\|^2 + \mu B(z_\epsilon))\|_{z_\epsilon}^* \leq \sigma\epsilon/2$. It can be verified that such z_ϵ satisfies Assumption 5.1(a). As observed from Theorem 4.1, the resulting iteration and operation complexity for finding such z_ϵ are respectively $\mathcal{O}(\epsilon^{-3/2})$ and $\tilde{\mathcal{O}}(\epsilon^{-3/2} \min\{n, \epsilon^{-1/4}\})$, which are negligible compared with those of our barrier-AL method (see Theorems 5.4 and 5.5 below).

(ii) Assumption 5.1(b) is mild. In particular, the assumption in (29) holds if $f(x) + \mu_0[B(x)]_+$ is bounded above for all $x \in \text{int } \mathcal{K}$ with $\|c(x)\| \leq 1$. Besides, the function $f(x) + \mu B(x) + \gamma \|c(x)\|^2/2$ is a barrier-quadratic penalty function of problem (1) and is typically bounded below on $\text{int } \mathcal{K}$. In addition, letting z^0 be an arbitrary point in $\text{int } \mathcal{K}$, it can be shown that $\mathcal{S}(\delta_f, \delta_c) \subseteq \mathcal{S}_1 \cup \mathcal{S}_2$, where

$$\mathcal{S}_1 = \left\{ x \in \text{int } \mathcal{K} : f(x) \leq f_{\text{hi}} + \delta_f + \mu_0 + \mu_0[B(z^0)]_+, B(x) \geq -1 - [B(z^0)]_+, \|c(x)\| \leq 1 + \delta_c \right\},$$

$$\mathcal{S}_2 = \left\{ x \in \text{int } \mathcal{K} : \frac{f(x)}{-B(x)} \leq \frac{[f_{\text{hi}} + \delta_f]_+}{1 + [B(z^0)]_+} + \mu_0, B(x) \leq -1 - [B(z^0)]_+, \|c(x)\| \leq 1 + \delta_c \right\},$$

and $t_+ = \max\{0, t\}$ for all $t \in \mathbb{R}$. Thus, the assumption in (31) holds if \mathcal{S}_1 and \mathcal{S}_2 are bounded. The latter holds, for example, for the problem with $f(x) = \ell(x) + \sum_{i=1}^n x_i^p$, $B(x) = -\sum_{i=1}^n \ln x_i$ and $\mathcal{K} = \mathbb{R}_+^n$ studied in [42], where $\ell : \mathbb{R}^n \rightarrow \mathbb{R}_+$ is a loss function and $p > 0$.

(iii) Assumptions 5.1(c) means that $\nabla^2 f$ and $\nabla^2 c_i$, $1 \leq i \leq m$, are locally Lipschitz continuous in $\Omega(\delta_f, \delta_c)$ with respect to the local norms. As pointed out in [43, Section 5], such local Lipschitz continuity is weaker than the global Lipschitz continuity of $\nabla^2 f$ and $\nabla^2 c_i$, $1 \leq i \leq m$, in $\Omega(\delta_f, \delta_c)$. Besides, Assumption 5.1(d) holds if f and c are twice continuously differentiable in an open set containing \mathcal{K} .

5.1 A Newton-CG based barrier-AL method

We now describe our Newton-CG based barrier-AL method (Algorithm 2) for finding a stochastic $(\epsilon, \sqrt{\epsilon})$ -SOSP of problem (1) for a prescribed tolerance $\epsilon \in (0, 1)$. Instead of solving (1) directly, our method solves a sequence of perturbed equality constrained barrier problems

$$\min_x \{f(x) + \mu_k B(x) : \tilde{c}(x) = 0\}, \quad (35)$$

where μ_k is given in Algorithm 2, z_ϵ is given in Assumption 5.1(a), and

$$\tilde{c}(x) := c(x) - c(z_\epsilon). \quad (36)$$

It follows a similar AL framework as the one proposed in [44]. In particular, at the k th iteration, it first applies the preconditioned Newton-CG method (Algorithm 1) to find an approximate stochastic SOSP x^{k+1} of the subproblem:

$$\min_x \left\{ \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k) := f(x) + \mu_k B(x) + (\lambda^k)^T \tilde{c}(x) + \frac{\rho_k}{2} \|\tilde{c}(x)\|^2 \right\}, \quad (37)$$

which is an AL subproblem associated with (35). Then the standard multiplier estimate $\tilde{\lambda}^{k+1}$ is updated by the classical scheme (see step 3 of Algorithm 2), and the truncated Lagrangian multiplier λ^{k+1} is updated by projecting $\tilde{\lambda}^{k+1}$ onto a Euclidean ball (see step 5 of Algorithm 2).⁵ Finally, the penalty parameter ρ_{k+1} is adaptively updated according to the improvement on constraint violation (see step 6 of Algorithm 2). This update scheme is very practical and widely used in AL type methods (e.g., see [3, 8, 28]).

Remark 5.2. (i) Notice that the starting point x_{init}^0 of Algorithm 2 can be different from z_ϵ and it may be rather infeasible, though z_ϵ is a nearly feasible point of (1). Besides, z_ϵ is used to monitor convergence of Algorithm 2. Specifically, if the algorithm runs into a “poorly infeasible point” x^k , namely satisfying

⁵The λ^{k+1} is also called a safeguarded Lagrangian multiplier, which has been used in the literature for designing some AL methods (e.g., see [3, 13, 44, 47]). It has been shown to enjoy many practical and theoretical advantages (e.g., see [13]).

Algorithm 2 A Newton-CG based barrier-AL method for problem (1)

Let γ and μ be given in Assumption 5.1.

Input: $\epsilon \in (0, 1)$, $\Lambda > 0$, $x^0 \in \text{int } \mathcal{K}$, $\lambda^0 \in \mathcal{B}_\Lambda$, $\rho_0 > 2\gamma$, $\alpha \in (0, 1)$, $r > 1$, $\delta \in (0, 1)$, z_ϵ given in Assumption 5.1(a), and $\mu_k = \max\{\epsilon, r^{k \log \epsilon / \log 2}\} / (2\vartheta^{1/2} + 2)$ for all $k \geq 0$.

- 1: Set $k = 0$.
- 2: Call Algorithm 1 with $\epsilon_g = \mu_k$, $\epsilon_H = \sqrt{\mu_k}$ and $u^0 = x_{\text{init}}^k$ to find an approximate solution $x^{k+1} \in \text{int } \mathcal{K}$ to $\min_x \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k)$ such that

$$\mathcal{L}_{\mu_k}(x^{k+1}, \lambda^k; \rho_k) \leq f(z_\epsilon) + \mu_k B(z_\epsilon), \quad \|\nabla_x \mathcal{L}_{\mu_k}(x^{k+1}, \lambda^k; \rho_k)\|_{x^{k+1}}^* \leq \mu_k, \quad (38)$$

$$\lambda_{\min}(M_{k+1}^T \nabla_{xx}^2 \mathcal{L}_{\mu_k}(x^{k+1}, \lambda^k; \rho_k) M_{k+1}) \geq -\sqrt{\mu_k} \text{ with probability at least } 1 - \delta, \quad (39)$$

where M_{k+1} is defined as in (16) and

$$x_{\text{init}}^k = \begin{cases} z_\epsilon & \text{if } \mathcal{L}_{\mu_k}(x^k, \lambda^k; \rho_k) > f(z_\epsilon) + \mu_k B(z_\epsilon), \\ x^k & \text{otherwise,} \end{cases} \quad \text{for } k \geq 0. \quad (40)$$

- 3: Set $\tilde{\lambda}^{k+1} = \lambda^k + \rho_k \tilde{c}(x^{k+1})$.
- 4: If $\mu_k \leq \epsilon / (2\vartheta^{1/2} + 2)$ and $\|c(x^{k+1})\| \leq \epsilon$, then output $(x^{k+1}, \tilde{\lambda}^{k+1})$ and terminate.
- 5: Set $\lambda^{k+1} = \Pi_{\mathcal{B}_\Lambda}(\tilde{\lambda}^{k+1})$.
- 6: If $k = 0$ or $\|\tilde{c}(x^{k+1})\| > \alpha \|\tilde{c}(x^k)\|$, set $\rho_{k+1} = r \rho_k$. Otherwise, set $\rho_{k+1} = \rho_k$.
- 7: Set $k \leftarrow k + 1$, and go to step 2.

$\mathcal{L}_{\mu_k}(x^k, \lambda^k; \rho_k) > f(z_\epsilon) + \mu_k B(z_\epsilon)$, it will be superseded by z_ϵ (see (40)), which prevents the iterates $\{x^k\}$ from converging to an infeasible point. Yet, x^k may be rather infeasible when k is not large. Thus, Algorithm 2 substantially differs from a funneling or two-phase type algorithm, in which a nearly feasible point is found in Phase 1, and then approximate stationarity is sought while near feasibility is maintained throughout Phase 2 (e.g., see [11, 16, 22, 23, 24, 25, 26, 36]).

- (ii) The choice of ρ_0 in Algorithm 2 is mainly for the simplicity of complexity analysis. Yet, it may be overly large and lead to highly ill-conditioned AL subproblems in practice. To make Algorithm 2 practically more efficient, one can possibly modify it by choosing a relatively small initial penalty parameter, then solving the subsequent AL subproblems by a first-order method until an ϵ_1 -first-order stationary point \hat{x} of (35) along with a Lagrangian multiplier $\hat{\lambda}$ is found, and finally performing the steps described in Algorithm 2 but with $x^0 = \hat{x}$ and $\lambda^0 = \Pi_{\mathcal{B}_\Lambda}(\hat{\lambda})$.
- (iii) Algorithm 2 can be easily extended to find an $(\epsilon, \sqrt{\epsilon})$ -SOSP of a more general conic optimization problem of the form $\min_{x, y} \{\tilde{f}(x, y) : \tilde{c}(x, y) = 0, y \in \mathcal{K}\}$. Indeed, one can follow almost the same framework as Algorithm 2, except that the associated subproblems are solved by a preconditioned Newton-CG method, which is a slight modification of Algorithm 1 by choosing the preconditioning matrix \tilde{M}_k as the one satisfying

$$\begin{bmatrix} I & 0 \\ 0 & \nabla^2 B(y^k) \end{bmatrix}^{-1} = \tilde{M}_k \tilde{M}_k^T.$$

- (iv) It is worth mentioning that Algorithm 2 shares some similarities with classical primal interior point methods for convex conic optimization. Specifically, Algorithm 2 applies a damped Newton's method to solve a sequence of barrier-AL subproblems, while the primal interior point method applies a projected Newton's method to solve a sequence of constrained barrier subproblems.

Before analyzing the complexity of Algorithm 2, we first argue that it is well-defined if ρ_0 is suitably chosen. Specifically, we will show that when ρ_0 is sufficiently large, one can apply Algorithm 1 to the subproblem $\min_x \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k)$ with x_{init}^k as the initial point to find an x^{k+1} satisfying (38) and (39). To this end, we start by noting from (29), (36), (37) and (40) that

$$\mathcal{L}_{\mu_k}(x_{\text{init}}^k, \lambda^k; \rho_k) \stackrel{(40)}{\leq} \max\{\mathcal{L}_{\mu_k}(z_\epsilon, \lambda^k; \rho_k), f(z_\epsilon) + \mu_k B(z_\epsilon)\} \stackrel{(36)(37)}{=} f(z_\epsilon) + \mu_k B(z_\epsilon) \stackrel{(29)}{\leq} f_{\text{hi}}. \quad (41)$$

Based on this observation, we show in the next lemma that when ρ_0 is sufficiently large, $\mathcal{L}_{\mu_k}(\cdot, \lambda^k; \rho_k)$ is bounded below and its certain level set is bounded, whose proof is deferred to Section 7.2.

Lemma 5.1 (Properties of $\mathcal{L}_{\mu_k}(\cdot, \lambda^k; \rho_k)$ and $\mathcal{L}(\cdot, \lambda^k; \rho_k)$). Suppose that Assumption 5.1 holds. Let (λ^k, ρ_k) be generated at the k th iteration of Algorithm 2 for some $k \geq 0$, and

$$\mathcal{L}(x, \lambda^k; \rho_k) := f(x) + (\lambda^k)^T \tilde{c}(x) + \frac{\rho_k}{2} \|\tilde{c}(x)\|^2. \quad (42)$$

Let $\mathcal{S}(\delta_f, \delta_c)$ and x_{init}^k be respectively defined in (31) and (40), μ_k be given in Algorithm 2, and let $\delta_f, \delta_c, f_{\text{hi}}, f_{\text{low}}, L_H^f, L_H^c, U_H^f, U_g^c, U_H^c$ and $\Omega(\delta_f, \delta_c)$ be given in Assumption 5.1. Suppose that ρ_0 is sufficiently large such that $\delta_{f,1} \leq \delta_f$ and $\delta_{c,1} \leq \delta_c$, where

$$\delta_{f,1} := \Lambda^2/(2\rho_0) \quad \text{and} \quad \delta_{c,1} := \sqrt{\frac{2(f_{\text{hi}} - f_{\text{low}} + \gamma)}{\rho_0 - 2\gamma} + \frac{\Lambda^2}{(\rho_0 - 2\gamma)^2}} + \frac{\Lambda}{\rho_0 - 2\gamma}. \quad (43)$$

Then the following statements hold.

- (i) $\{x \in \text{int } \mathcal{K} : \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k) \leq \mathcal{L}_{\mu_k}(x_{\text{init}}^k, \lambda^k; \rho_k)\} \subseteq \mathcal{S}(\delta_f, \delta_c)$.
- (ii) $\inf_{x \in \text{int } \mathcal{K}} \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k) \geq f_{\text{low}} - \gamma - \Lambda \delta_c$.
- (iii) $\|\nabla_{xx}^2 \mathcal{L}(y, \lambda^k; \rho_k) - \nabla_{xx}^2 \mathcal{L}(x, \lambda^k; \rho_k)\|_x^* \leq L_{k,H} \|y - x\|_x$ for all $x, y \in \Omega(\delta_f, \delta_c)$ with $\|y - x\|_x \leq \beta$, where

$$L_{k,H} := L_H^f + \|\lambda^k\|_1 L_H^c + \rho_k m \left[(1 + U^c) L_H^c + \frac{U_g^c U_H^c}{1 - \beta} + \frac{(2 - \beta) U_g^c U_H^c}{(1 - \beta)^3} \right], \quad U^c := \sup_{z \in \Omega(\delta_f, \delta_c)} \|c(z)\|. \quad (44)$$

- (iv) The quantities $U_{k,g}$ and $U_{k,H}$ are finite, where

$$U_{k,g} := \sup_{x \in \mathcal{S}(\delta_f, \delta_c)} \|\nabla_x \mathcal{L}(x, \lambda^k; \rho_k)\|_x^*, \quad U_{k,H} := \sup_{x \in \mathcal{S}(\delta_f, \delta_c)} \|\nabla_{xx}^2 \mathcal{L}(x, \lambda^k; \rho_k)\|_x^*.$$

Moreover, $U_{k,g} \leq U_g^f + \|\lambda^k\|_1 U_g^c + \rho_k \sqrt{m}(2 + \delta_c) U_g^c$ and $U_{k,H} \leq U_H^f + \|\lambda^k\|_1 U_H^c + \rho_k (m(U_g^c)^2 + \sqrt{m}(2 + \delta_c) U_H^c)$.

In view of (31) and Lemma 5.1(i) and (ii), one can see that the level set $\{x \in \text{int } \mathcal{K} : \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k) \leq \mathcal{L}_{\mu_k}(x_{\text{init}}^k, \lambda^k; \rho_k)\}$ is bounded and $\mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k)$ is bounded below for all $x \in \text{int } \mathcal{K}$. By these and Lemma 5.1(iii) and (iv), one can further see that Assumption 4.1 holds for $F(\cdot) = \mathcal{L}(\cdot, \lambda^k; \rho_k)$ and $u^0 = x_{\text{init}}^k$. Based on this and the discussion in Section 4, we can conclude that Algorithm 1, starting with $u^0 = x_{\text{init}}^k$, is applicable to the subproblem $\min_x \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k)$. Moreover, it follows from Theorem 4.1 that this algorithm with $\epsilon_g = \mu_k$ and $\epsilon_H = \sqrt{\mu_k}$ can produce a point x^{k+1} satisfying (39) and also the second relation in (38). In addition, since this algorithm is descent and its starting point is x_{init}^k , its output x^{k+1} must satisfy $\mathcal{L}_{\mu_k}(x^{k+1}, \lambda^k; \rho_k) \leq \mathcal{L}_{\mu_k}(x_{\text{init}}^k, \lambda^k; \rho_k)$, which along with (41) implies that $\mathcal{L}_{\mu_k}(x^{k+1}, \lambda^k; \rho_k) \leq f(z_\epsilon) + \mu_k B(z_\epsilon)$ and thus x^{k+1} also satisfies the first relation in (38).

The above discussion leads to the following conclusion concerning the well-definedness of Algorithm 2.

Theorem 5.1 (Well-definedness of Algorithm 2). Under the same settings as in Lemma 5.1, the preconditioned Newton-CG method (Algorithm 1), when applied to the subproblem $\min_x \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k)$ with $u^0 = x_{\text{init}}^k$, can find a point x^{k+1} satisfying (38) and (39).

The following theorem characterizes the output of Algorithm 2, whose proof is deferred to Section 7.3.

Theorem 5.2 (Output of Algorithm 2). Suppose that Assumption 5.1 holds and that ρ_0 is sufficiently large such that $\delta_{f,1} \leq \delta_f$ and $\delta_{c,1} \leq \delta_c$, where $\delta_{f,1}$ and $\delta_{c,1}$ are defined in (43). If Algorithm 2 terminates at some iteration k , then its output x^{k+1} is a deterministic ϵ -FOSP of problem (1), and moreover, it is an $(\epsilon, \sqrt{\epsilon})$ -SOSP of (1) with probability at least $1 - \delta$.

Remark 5.3. As seen from Theorem 5.2, the output of Algorithm 2 is a stochastic $(\epsilon, \sqrt{\epsilon})$ -SOSP of problem (1). On the other hand, this algorithm can be easily modified to find other approximate solutions of (1) as well. For example, if only an ϵ -FOSP of (1) is to be sought, one can remove the condition (39) from Algorithm 2. In addition, if one aims to find a deterministic $(\epsilon, \sqrt{\epsilon})$ -SOSP of (1), one can replace the condition (39) and Algorithm 1 by $\lambda_{\min}(M_{k+1}^T \nabla_{xx}^2 \mathcal{L}_{\mu_k}(x^{k+1}, \lambda^k; \rho_k) M_{k+1}) \geq -\sqrt{\mu_k}$ and a deterministic counterpart, respectively.

5.2 Outer iteration complexity of Algorithm 2

In this subsection we establish *outer iteration complexity* of Algorithm 2, which measures the number of its outer iterations. Notice that μ_k can be rewritten as

$$\mu_k = \max\{\epsilon, \omega^k\}/(2\vartheta^{1/2} + 2) \text{ with } \omega := r^{\log \epsilon / \log 2}, \quad \forall k \geq 0, \quad (45)$$

where r is an input of Algorithm 2. By $\epsilon \in (0, 1)$ and $r > 1$, one has $\omega \in (0, 1)$. For notational convenience, we introduce the following quantity that will be frequently used later:

$$K_\epsilon := \lceil \min\{k \geq 0 : \omega^k \leq \epsilon\} \rceil. \quad (46)$$

In view of this and (45), we obtain that $\mu_k = \epsilon/(2\vartheta^{1/2} + 2)$ for all $k \geq K_\epsilon$. This along with the termination criterion of Algorithm 2 implies that it runs for at least K_ϵ iterations and terminates once $\|c(x^{k+1})\| \leq \epsilon$ for some $k \geq K_\epsilon$. Consequently, to establish outer iteration complexity of Algorithm 2, it suffices to bound such k . The resulting outer iteration complexity is presented below, whose proof is deferred to Section 7.3.

Theorem 5.3 (Outer iteration complexity of Algorithm 2). *Suppose that Assumption 5.1 holds and that ρ_0 is sufficiently large such that $\delta_{f,1} \leq \delta_f$ and $\delta_{c,1} \leq \delta_c$, where $\delta_{f,1}$ and $\delta_{c,1}$ are defined in (43). Let*

$$\rho_{\epsilon,1} := \max \{8(f_{\text{hi}} - f_{\text{low}} + \gamma)\epsilon^{-2} + 4\Lambda\epsilon^{-1} + 2\gamma, 2\rho_0\}, \quad (47)$$

$$\overline{K}_\epsilon := \inf\{k \geq K_\epsilon : \|c(x^{k+1})\| \leq \epsilon\}, \quad (48)$$

where K_ϵ is defined in (46), and γ , f_{hi} and f_{low} are given in Assumption 5.1. Then \overline{K}_ϵ is finite, and Algorithm 2 terminates at iteration \overline{K}_ϵ with

$$\overline{K}_\epsilon \leq \left(\frac{\log(\rho_{\epsilon,1}\rho_0^{-1})}{\log r} + 1 \right) \left(\left| \frac{\log(\epsilon(2\delta_{c,1})^{-1})}{\log \alpha} \right| + 2 \right) + 1. \quad (49)$$

Moreover, $\rho_k \leq r\rho_{\epsilon,1}$ holds for $0 \leq k \leq \overline{K}_\epsilon$.

Remark 5.4 (Upper bounds for \overline{K}_ϵ and $\{\rho_k\}$). *As seen from Theorem 5.3, the number of outer iterations of Algorithm 2 for finding a stochastic $(\epsilon, \sqrt{\epsilon})$ -SOSP of problem (1) is at most of $\mathcal{O}(|\log \epsilon|^2)$. In addition, the penalty parameters $\{\rho_k\}$ generated by this algorithm are at most of $\mathcal{O}(\epsilon^{-2})$.*

5.3 Total inner iteration and operation complexity of Algorithm 2

In this subsection we present the *total inner iteration and operation complexity* of Algorithm 2, which measures the total number of iterations and fundamental operations performed by Algorithm 1 in Algorithm 2. Its proof is deferred to Section 7.3.

Theorem 5.4 (Total inner iteration and operation complexity of Algorithm 2). *Suppose that Assumption 5.1 holds and that ρ_0 is sufficiently large such that $\delta_{f,1} \leq \delta_f$ and $\delta_{c,1} \leq \delta_c$, where $\delta_{f,1}$ and $\delta_{c,1}$ are defined in (43). Then the following statements hold.*

- (i) *The total number of inner iterations of Algorithm 2, namely, the total number of iterations of Algorithm 1 performed in Algorithm 2, is at most $\tilde{\mathcal{O}}(\epsilon^{-11/2})$. If c is further assumed to be affine, it is at most $\tilde{\mathcal{O}}(\epsilon^{-3/2})$.*
- (ii) *The total numbers of Cholesky factorizations and other fundamental operations consisting of the Hessian-vector products of f and c and backward or forward substitutions to a triangular linear system required by Algorithm 1 in Algorithm 2 are at most $\tilde{\mathcal{O}}(\epsilon^{-11/2})$ and $\tilde{\mathcal{O}}(\epsilon^{-11/2} \min\{n, \epsilon^{-5/4}\})$, respectively. If c is further assumed to be affine, they are at most $\tilde{\mathcal{O}}(\epsilon^{-3/2})$ and $\tilde{\mathcal{O}}(\epsilon^{-3/2} \min\{n, \epsilon^{-5/4}\})$, respectively.*

Remark 5.5. (i) *It is worth mentioning that the above complexity results are established without assuming any constraint qualification. Moreover, when \mathcal{K} is the nonnegative orthant, these results match the best-known ones achieved by a Newton-CG based AL method [44] for nonconvex equality constrained optimization without imposing a constraint qualification.*

- (ii) *For the unconstrained case, where $\mathcal{K} = \mathbb{R}^n$ and $c \equiv 0$, we can eliminate the barrier function for handling \mathcal{K} and the AL function for handling c from Algorithm 2. Consequently, Algorithm 2 reduces to the Newton-CG method presented in [44, Algorithm 1] and [60, Algorithm 3]. The corresponding operation complexity bound can be improved to $\tilde{\mathcal{O}}(\epsilon^{-3/2} \min\{n, \epsilon^{-1/4}\})$, which matches the results provided in [44, Theorem 3.1] and [60, Theorem 4 and Corollary 2] for unconstrained optimization.*

5.4 Enhanced complexity of Algorithm 2 under constraint qualification

In this subsection we study complexity of Algorithm 2 under one additional assumption that a generalized linear independence constraint qualification (GLICQ) holds for problem (1), which is introduced below. In particular, under GLICQ we will obtain an enhanced total inner iteration complexity of $\tilde{\mathcal{O}}(\epsilon^{-7/2})$ and an enhanced operation complexity of $\tilde{\mathcal{O}}(\epsilon^{-7/2} \min\{n, \epsilon^{-3/4}\})$ for Algorithm 2 when the equality constraints in problem (1) are nonlinear, which are significantly better than the ones in Theorem 5.4. We now introduce the GLICQ assumption for (1).

Assumption 5.2 (GLICQ). *There exists some $\sigma > 0$ such that*

$$\lambda_{\min}(\nabla c(x)^T \nabla^2 B(x)^{-1} \nabla c(x)) \geq \sigma^2, \quad \forall x \in \mathcal{S}(\delta_f, \delta_c), \quad (50)$$

where $\mathcal{S}(\delta_f, \delta_c)$ is defined in (31).

The following theorem shows that under Assumption 5.2, the total inner iteration and operation complexity results presented in Theorem 5.4 can be significantly improved, whose proof is deferred to Section 7.3.

Theorem 5.5 (Enhanced total inner iteration and operation complexity of Algorithm 2). *Suppose that Assumptions 5.1 and 5.2 hold and that ρ_0 is sufficiently large such that $\delta_{f,1} \leq \delta_f$ and $\delta_{c,1} \leq \delta_c$, where $\delta_{f,1}$ and $\delta_{c,1}$ are defined in (43). Then the following statements hold.*

- (i) *The total number of inner iterations of Algorithm 2, namely, the total number of iterations of Algorithm 1 performed in Algorithm 2, is at most $\tilde{\mathcal{O}}(\epsilon^{-7/2})$. If c is further assumed to be affine, it is at most $\tilde{\mathcal{O}}(\epsilon^{-3/2})$.*
- (ii) *The total numbers of Cholesky factorizations and other fundamental operations consisting of the Hessian-vector products of f and c and backward or forward substitutions to a triangular linear system required by Algorithm 1 in Algorithm 2 are at most $\tilde{\mathcal{O}}(\epsilon^{-7/2})$ and $\tilde{\mathcal{O}}(\epsilon^{-7/2} \min\{n, \epsilon^{-3/4}\})$, respectively. If c is further assumed to be affine, they are at most $\tilde{\mathcal{O}}(\epsilon^{-3/2})$ and $\tilde{\mathcal{O}}(\epsilon^{-3/2} \min\{n, \epsilon^{-3/4}\})$, respectively.*

Remark 5.6. *As seen from Theorem 5.5, under GLICQ and some other suitable assumptions, Algorithm 2 achieves significantly better complexity bounds than the ones in Theorem 5.4 when the equality constraints in (1) are nonlinear. Moreover, when \mathcal{K} is the nonnegative orthant, the complexity results in Theorem 5.5 match the best-known ones achieved by a Newton-CG based AL method [44] for nonconvex equality constrained optimization under the constraint qualification that is obtained from the above GLICQ by replacing $\nabla^2 B(x)$ by the identity matrix.*

6 Numerical results

In this section we conduct some preliminary numerical experiments to test the performance of our Newton-CG based barrier-AL method (Algorithm 2) for solving a low-rank matrix recovery problem, a simplex-constrained nonnegative matrix factorization problem, and a sphere-constrained nonnegative matrix factorization problem. In our experiments, all the algorithms are coded in Matlab and all the computations are performed on a desktop with a 3.79 GHz AMD 3900XT 12-Core processor and 32 GB of RAM.

6.1 Low-rank matrix recovery

In this subsection we consider a low-rank matrix recovery problem (e.g., see [9, 29, 59])

$$\min_{U \in \mathbb{R}^{n \times l}} \left\{ \frac{1}{2} \|\mathcal{A}(UU^T) - y\|^2 : \|U\|_F^2 \leq b \right\}, \quad (51)$$

where $\mathcal{A} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^m$ is a linear operator and $\|\cdot\|_F$ is the Frobenius norm.

For each triple (n, l, m) , we randomly generate 10 instances of problem (51) in a similar manner as described in [9]. In particular, we first randomly generate a linear operator \mathcal{A} by setting $\mathcal{A}(\cdot) = A(\text{vec}(\cdot))$, where A is an $m \times n^2$ matrix with all entries chosen from the normal distribution with mean zero and standard deviation $1/\sqrt{m}$, and $\text{vec}(\cdot)$ is the vectorization of the associated matrix.⁶ Then we randomly generate the ground-truth

⁶The vectorization of a matrix is the column vector obtained by stacking the columns of the matrix on top of one another.

n	l	m	Relative error		Objective value	
			Algorithm 2	SpaRSA	Algorithm 2	SpaRSA
20	1	40	6.3×10^{-4}	6.3×10^{-4}	9.9×10^{-4}	9.8×10^{-4}
20	2	80	3.3×10^{-4}	0.60	2.0×10^{-3}	7.8×10^3
40	2	160	1.7×10^{-4}	0.66	4.2×10^{-3}	7.1×10^4
40	4	320	1.2×10^{-4}	0.81	8.0×10^{-3}	5.5×10^5
60	3	360	9.2×10^{-5}	0.78	9.3×10^{-3}	8.4×10^5
60	6	720	6.3×10^{-5}	0.85	1.9×10^{-2}	5.1×10^6
80	4	640	5.8×10^{-5}	0.83	1.6×10^{-2}	4.4×10^6
80	8	1280	3.9×10^{-5}	0.90	3.3×10^{-2}	2.5×10^7
100	5	1000	4.2×10^{-5}	0.89	2.6×10^{-2}	1.6×10^7
100	10	2000	2.8×10^{-5}	0.92	5.2×10^{-2}	8.1×10^7

Table 1: Numerical results for problem (51)

low-rank matrix $X^* = \tilde{U}\tilde{U}^T$ with all entries of \tilde{U} chosen from the standard normal distribution. We finally set $b = \|\tilde{U}\|_F^2$ and $y = \mathcal{A}(X^*) + e$, where e_i , $1 \leq i \leq m$, is generated according to the normal distribution with mean zero and standard deviation 0.01.

Observe that problem (51) is equivalent to

$$\min_{U,s} \left\{ \frac{1}{2} \|\mathcal{A}(UU^T) - y\|^2 : \|U\|_F^2 + s = b, s \geq 0 \right\}. \quad (52)$$

In this experiment, we apply Algorithm 2 to find a $(10^{-4}, 10^{-2})$ -SOSP of (52) and hence of (51). To ensure that the output of Algorithm 2 is a deterministic approximate second-order stationary point, we use a minimum eigenvalue oracle that returns a deterministic output in Algorithm 2 instead, which calls the Matlab subroutine `[v,λ] = eigs(H,1,'smallestreal')` to find the minimum eigenvalue λ and its associated unit eigenvector v of a real symmetric matrix H . Besides, we apply [68, Algorithm SpaRSA], which is a nonmonotone proximal gradient method, to find a 10^{-4} -FOSP of (51) by generating a sequence $\{U^t\}$ according to

$$U^t = \arg \min_U \{ \|U - U^{t-1} + \nabla f(U^{t-1})/\alpha_{t-1}\|_F : \|U\|_F^2 \leq b \},$$

where f is the objective function of (51) and α_{t-1} is chosen by a backtracking line search scheme such that $f(U^t) \leq \max_{[t-M-1]_+ \leq i \leq t-1} f(U^i) - \sigma \alpha_{t-1} \|U^t - U^{t-1}\|_F^2/2$ for some $\sigma \in (0, 1)$ and a positive integer M (see [68] for details). We terminate SpaRSA once the condition

$$\|\alpha_{t-1}(U^t - U^{t-1}) + \nabla f(U^{t-1}) - \nabla f(U^t)\|_F \leq 10^{-4}$$

is met. It can be verified that such U^t is a 10^{-4} -FOSP of (51). We choose the initial point U^0 with all entries equal to $\sqrt{b/(2nl)}$ for both methods, $s^0 = b/2$ for Algorithm 2, and set

- $(\Lambda, \rho_0, \lambda^0, \alpha, r) = (10^3, 10^2, 0, 0.25, 1.5)$ for Algorithm 2, and $(\theta, \zeta, \eta, \beta) = (0.5, 0.5, 0.01, 0.9)$ for Algorithm 1;
- $(\sigma, M, \alpha_{\min}, \alpha_{\max}, \eta) = (0.01, 5, 10^{-30}, 10^{30}, 2)$ for SpaRSA [68].

Notice that the approximate solution obtained by SpaRSA must be a feasible point of (51), while the one found by Algorithm 2 may not be a feasible point of (51). For a fair comparison, we project the latter one into the feasible region of (51) to obtain a feasible approximate solution. Then we compare the quality of these feasible approximate solutions in terms of objective value and relative error defined as $\|UU^T - X^*\|_F/\|X^*\|_F$ for a given U . The computational results of Algorithm 2 and SpaRSA for the instances randomly generated above are presented in Table 1. In detail, the values of n , l and m are listed in the first three columns, respectively. For each triple (n, l, m) , the average relative error and the average objective value of the feasible approximate solutions found by each method over 10 random instances are given in the rest columns. One can observe that the approximate SOSP found by Algorithm 2 has significantly lower relative error and objective value than the approximate FOSP obtained by SpaRSA, except for the instances with $(n, l, m) = (20, 1, 40)$.

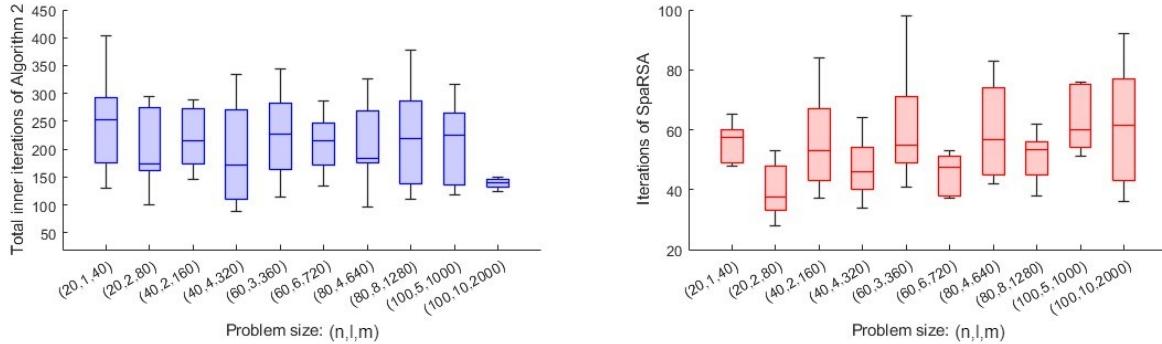


Figure 1: Left: The total number of inner iterations of Algorithm 2 for finding a $(10^{-4}, 10^{-2})$ -SOSP of (52) for each problem size over 10 random instances. Right: The number of iterations of SpaRSA for finding a 10^{-4} -FOSP of (51) for each problem size over 10 random instances.

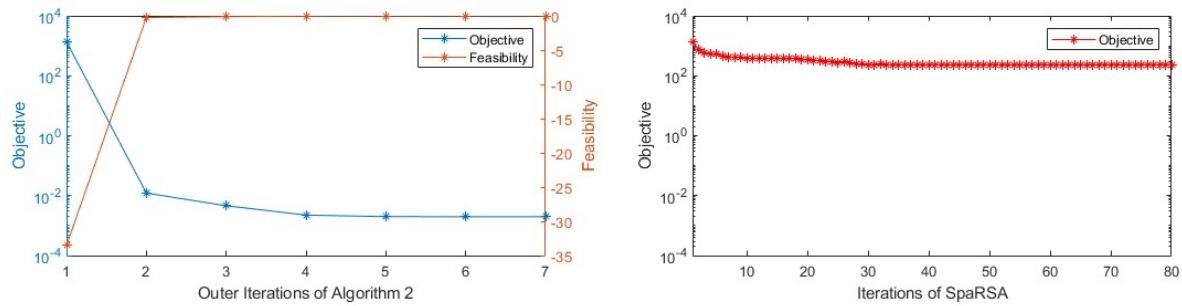


Figure 2: Numerical results of Algorithm 2 and SpaRSA on a single random instance of problem (51) with $(n, l, m) = (20, 2, 80)$. These two figures illustrate the convergence behavior of both methods in terms of objective value $\frac{1}{2} \|\mathcal{A}(U^t(U^t)^T) - y\|^2$ and feasibility $[\|U^t\|_F^2 - b]_+$.

For each triple (n, l, m) , we use box charts in Figure 1 to present the total number of inner iterations of Algorithm 2 for finding a $(10^{-4}, 10^{-2})$ -SOSP of problem (52) and the number of iterations of SpaRSA for finding a 10^{-4} -FOSP of problem (51) over 10 random instances. We observe that the total number of inner iterations of Algorithm 2 remains at a similar level when the problem size becomes larger. In addition, Figure 2 illustrates the convergence behavior of both methods for solving a single random instance of (51) with $(n, l, m) = (20, 2, 80)$. We observe that SpaRSA converges in fewer (inner) iterations than Algorithm 2. However, it converges to a suboptimal solution with a significantly larger objective value compared to the solution found by Algorithm 2.

6.2 A simplex-constrained nonnegative matrix factorization

In this subsection we consider a simplex-constrained nonnegative matrix factorization (e.g., see [45, 50, 54, 64]) in the form of

$$\min_{U \in \mathbb{R}^{n \times l}, V \in \mathbb{R}^{l \times m}} \left\{ \frac{1}{2} \|X - UV\|_F^2 + \gamma(\|U\|_F^2 + \|V\|_F^2) : V^T e_l = e_m, U \geq 0, V \geq 0 \right\} \quad (53)$$

for some $\gamma > 0$, where $\|\cdot\|_F$ is the Frobenius norm and e_d is the d -dimensional all-ones vector for any $d \geq 1$.

n	l	m	Relative error		Objective value	
			Algorithm 2	SpaRSA	Algorithm 2	SpaRSA
20	2	10	4.8×10^{-3}	0.15	0.30	3.1
20	2	20	3.6×10^{-3}	0.16	0.35	6.4
20	2	30	3.2×10^{-3}	0.16	0.39	9.7
30	3	15	5.8×10^{-3}	0.16	0.62	7.6
30	3	30	4.3×10^{-3}	0.17	0.70	16.1
30	3	45	3.6×10^{-3}	0.17	0.76	23.8
40	4	20	6.3×10^{-3}	0.15	1.0	11.1
40	4	40	4.6×10^{-3}	0.15	1.2	21.7
40	4	60	4.0×10^{-3}	0.15	1.2	31.8
50	5	25	6.8×10^{-3}	0.14	1.6	15.1
50	5	50	5.0×10^{-3}	0.14	1.8	29.8
50	5	75	4.3×10^{-3}	0.14	1.9	43.9

Table 2: Numerical results for problem (53)

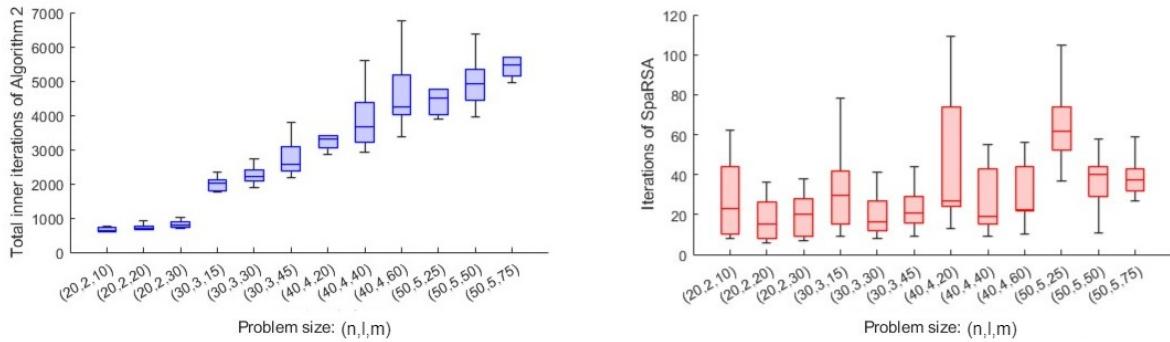


Figure 3: Left: The total inner iterations of Algorithm 2 before finding a $(10^{-4}, 10^{-2})$ -SOSP of (53) for each problem size over 10 random instances. Right: The number of iterations of SpaRSA before finding a 10^{-4} -FOSP of (53) for each problem size over 10 random instances.

For each triple (n, l, m) , we randomly generate 10 instances of problem (53). In particular, we first randomly generate U^* with all entries chosen from the uniform distribution over $[0, 2]$. We next randomly generate \tilde{V} with all entries chosen from the standard uniform distribution and set $V^* = \tilde{V}D$, where D is a diagonal matrix such that $(V^*)^T e_l = e_m$. In addition, we set $\gamma = 0.005$ and $X = U^*V^* + E$, where the entries of E follow the normal distribution with mean zero and standard deviation 0.01.

Our aim is to apply Algorithm 2 and SpaRSA [68] to solve (53) and compare the solution quality of these methods in terms of objective value and relative error defined as $\|UV - U^*V^*\|_F/\|U^*V^*\|_F$. In particular, we first apply Algorithm 2 to find a $(10^{-4}, 10^{-2})$ -SOSP of (53), in which a minimum eigenvalue oracle that returns a deterministic output, namely the Matlab subroutine $[v, \lambda] = \text{eigs}(H, 1, \text{'smallestreal'})$ is used. Given that the obtained approximate SOSP may not be a feasible point of (53), we post-multiply it by a suitable diagonal matrix to obtain a feasible approximate solution of (53). In addition, we apply SpaRSA [68] to find a 10^{-4} -FOSP of (53) by generating a sequence $\{(U^t, V^t)\}$ according to

$$(U^t, V^t) = \arg \min_{U, V} \left\{ \|(U, V) - (U^{t-1}, V^{t-1}) + \nabla f(U^{t-1}, V^{t-1})/\alpha_{t-1}\|_F : V^T e_l = e_m, U \geq 0, V \geq 0 \right\},$$

where f is the objective function of (53) and α_{t-1} is chosen by a backtracking line search scheme such that $f(U^t, V^t) \leq \max_{[t-M-1]_+ \leq i \leq t-1} f(U^i, V^i) - \sigma \alpha_{t-1} \|(U^t, V^t) - (U^{t-1}, V^{t-1})\|_F^2/2$ for some $\sigma \in (0, 1)$ and a

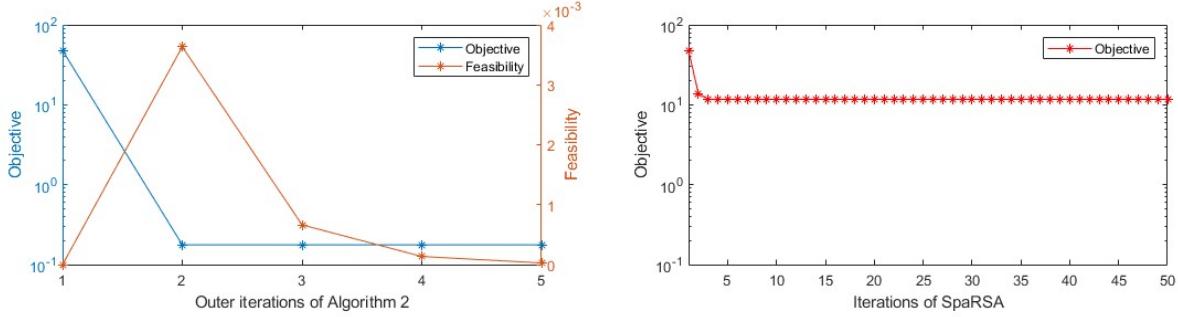


Figure 4: Numerical results of Algorithm 2 and SpaRSA on a single random instance of problem (53) with $(n, l, m) = (20, 2, 20)$. These two figures illustrate the convergence behavior of both methods in terms of objective value $\frac{1}{2}\|X - U^t V^t\|_F^2 + \gamma(\|U^t\|_F^2 + \|V^t\|_F^2)$ and feasibility $\|(V^t)^T e_l - e_m\|$.

positive integer M (see [68] for details). We terminate SpaRSA once the condition

$$\|\alpha_{t-1}((U^t, V^t) - (U^{t-1}, V^{t-1})) + \nabla f(U^{t-1}, V^{t-1}) - \nabla f(U^t, V^t)\|_F \leq 10^{-4}$$

is met. It can be verified that such (U^t, V^t) is a 10^{-4} -FOSP of (53). In addition, we choose the initial point U^0 and V^0 with all entries equal 1 and $1/k$ respectively for all the methods. We set the parameters for Algorithm 2 as $(\Lambda, \rho_0, \alpha, r) = (10^3, 10^2, 0.25, 1.5)$ and $\lambda^0 = (0, \dots, 0)^T$, and choose the same parameters for Algorithm 1 and SpaRSA as the ones described in Subsection 6.1.

The computational results of Algorithm 2 and SpaRSA [68] for the instances randomly generated above are presented in Table 2. In detail, the values of n , k and m are listed in the first three columns, respectively. For each triple (n, k, m) , the average relative error and the average objective value of the feasible approximate solutions found by each method over 10 random instances are given in the rest columns. One can observe that the approximate SOSP found by Algorithm 2 has significantly lower relative error and objective value than the approximate FOSP obtained by SpaRSA.

For each triple (n, l, m) , we use box charts in Figure 3 to present the total number of inner iterations of Algorithm 2 for finding a $(10^{-4}, 10^{-2})$ -SOSP of problem (53) and the number of iterations of SpaRSA for finding a 10^{-4} -FOSP of problem (53) over 10 random instances. We observe that the total number of inner iterations of Algorithm 2 increases when the problem size becomes larger. In addition, Figure 4 illustrates the convergence behavior of both methods for solving a single random instance of (53) with $(n, l, m) = (20, 2, 20)$. We observe that SpaRSA converges in fewer (inner) iterations than Algorithm 2, but it converges to a poorer approximate solution. Specifically, it quickly gets stuck at a suboptimal point with a significantly larger objective value compared to the solution found by Algorithm 2.

6.3 A sphere-constrained nonnegative matrix factorization

In this subsection we consider a sphere-constrained nonnegative matrix factorization in the form of

$$\min_{U \in \mathbb{R}^{n \times l}, V \in \mathbb{R}^{l \times m}} \left\{ \frac{1}{2} \|X - UV\|_F^2 + \gamma(\|U\|_F^2 + \|V\|_F^2) : \|V\|_F^2 = m, U \geq 0, V \geq 0 \right\}, \quad (54)$$

where $\|\cdot\|_F$ is the Frobenius norm.

For each triple (n, l, m) , we randomly generate 10 instances of problem (54). In particular, we first randomly generate U^* with all entries chosen from the uniform distribution over $[0, 2]$. We next randomly generate \tilde{V} with all entries chosen from the standard uniform distribution and set $V^* = \sqrt{m}\tilde{V}/\|\tilde{V}\|_F$. In addition, we set $\gamma = 0.005$ and $X = U^*V^* + E$, where the entries of E follow the normal distribution with mean zero and standard deviation 0.01.

n	l	m	Relative error		Objective value	
			Algorithm 2	SpaRSA	Algorithm 2	SpaRSA
20	2	5	5.5×10^{-3}	0.10	0.29	2.3
20	2	10	3.9×10^{-3}	0.12	0.33	5.5
20	2	15	3.4×10^{-3}	0.13	0.36	8.6
20	2	20	3.1×10^{-3}	0.12	0.39	10.9
20	2	25	2.9×10^{-3}	0.12	0.42	13.5
20	2	30	2.8×10^{-3}	0.12	0.45	16.5
40	4	10	4.5×10^{-3}	0.12	1.1	18.1
40	4	20	3.1×10^{-3}	0.12	1.2	36.8
40	4	30	2.8×10^{-3}	0.12	1.3	54.6
40	4	40	2.6×10^{-3}	0.13	1.3	75.0
40	4	50	2.2×10^{-3}	0.13	1.4	94.5
40	4	60	2.3×10^{-3}	0.13	1.5	113.0

Table 3: Numerical results for problem (54)

Our aim is to apply Algorithm 2 and SpaRSA [68] to solve (54) and compare the solution quality of these methods in terms of objective value and relative error defined as $\|UV - U^*V^*\|_F/\|U^*V^*\|_F$. In particular, we first apply Algorithm 2 to find a $(10^{-4}, 10^{-2})$ -SOSP of (54), in which a minimum eigenvalue oracle that returns a deterministic output, namely the Matlab subroutine $[\mathbf{v}, \lambda] = \text{eigs}(\mathbf{H}, 1, \text{'smallestreal'})$ is used. Given that the obtained approximate SOSP may not be a feasible point of (54), we post-multiply it by a suitable diagonal matrix to obtain a feasible approximate solution of (54). In addition, we apply SpaRSA [68] to find a 10^{-4} -FOSP of (53) by generating a sequence $\{(U^t, V^t)\}$ according to

$$(U^t, V^t) = \arg \min_{U, V} \left\{ \|(U, V) - (U^{t-1}, V^{t-1}) + \nabla f(U^{t-1}, V^{t-1})/\alpha_{t-1}\|_F : \|V\|_F^2 = m, U \geq 0, V \geq 0 \right\},$$

where f is the objective function of (54) and α_{t-1} is chosen by a backtracking line search scheme such that $f(U^t, V^t) \leq \max_{[t-M-1]_+ \leq i \leq t-1} f(U^i, V^i) - \sigma \alpha_{t-1} \|(U^t, V^t) - (U^{t-1}, V^{t-1})\|_F^2/2$ for some $\sigma \in (0, 1)$ and a positive integer M (see [68] for details). We terminate SpaRSA once the condition

$$\|\alpha_{t-1}((U^t, V^t) - (U^{t-1}, V^{t-1})) + \nabla f(U^{t-1}, V^{t-1}) - \nabla f(U^t, V^t)\|_F \leq 10^{-4}$$

is met. It can be verified that such (U^t, V^t) is a 10^{-4} -FOSP of (54). Furthermore, we choose the initial point U^0 and V^0 with all entries equal to 1 and $1/l$ respectively for all the methods. We set the parameters for Algorithm 2 as $(\Lambda, \rho_0, \alpha, r) = (10^3, 10^2, 0.25, 1.5)$ and $\lambda^0 = 0$, and choose the same parameters for Algorithm 1 and SpaRSA as the ones described in Subsection 6.1.

The computational results of Algorithm 2 and SpaRSA [68] for the instances randomly generated above are presented in Table 3. In detail, the values of n , l and m are listed in the first three columns, respectively. For each triple (n, l, m) , the average relative error and the average objective value of the feasible approximate solutions over 10 random instances are given in the rest columns. One can observe that the approximate SOSP found by Algorithm 2 has significantly lower relative error and objective value than the approximate FOSP obtained by SpaRSA.

For each triple (n, l, m) , we use box charts in Figure 5 to present the total number of inner iterations of Algorithm 2 for finding a $(10^{-4}, 10^{-2})$ -SOSP of problem (54) and the number of iterations of SpaRSA for finding a 10^{-4} -FOSP of problem (54) over 10 random instances. We observe that the total number of inner iterations of Algorithm 2 increases when the problem size becomes larger. In addition, Figure 6 illustrates the convergence behavior of both methods for solving a single random instance of (54) with $(n, l, m) = (20, 2, 5)$. We observe that SpaRSA converges in fewer (inner) iterations than Algorithm 2. However, it converges to a suboptimal solution with a much larger objective value compared to the solution found by Algorithm 2.

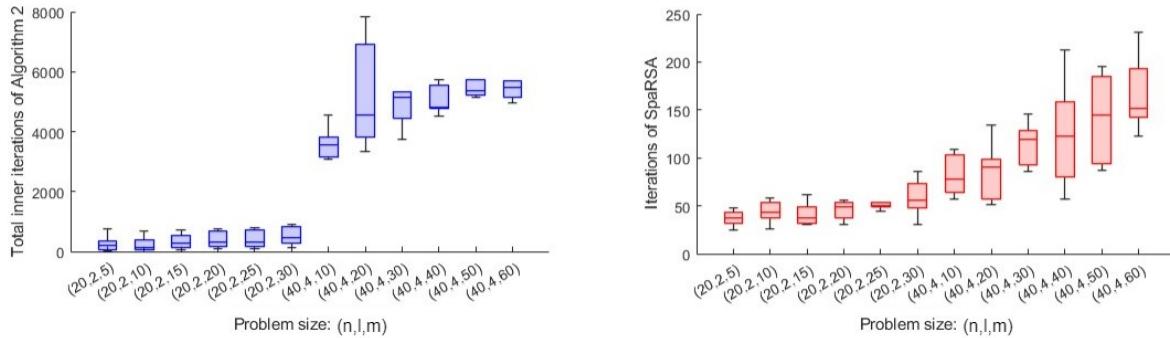


Figure 5: Left: The total inner iterations of Algorithm 2 before finding a $(10^{-4}, 10^{-2})$ -SOSP of (54) for each problem size over 10 random instances. Right: The number of iterations of SpaRSA before finding a 10^{-4} -FOSP of (54) for each problem size over 10 random instances.

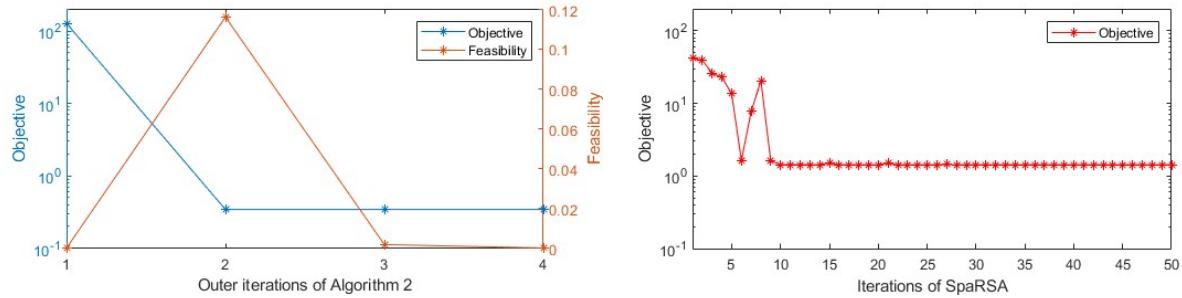


Figure 6: Numerical results of Algorithm 2 and SpaRSA on a single random instance of problem (54) with $(n, l, m) = (20, 2, 5)$. These two figures illustrate the convergence behavior of both methods in terms of objective value $\frac{1}{2}\|X - U^t V^t\|_F^2 + \gamma(\|U^t\|_F^2 + \|V^t\|_F^2)$ and feasibility $||V^t\|_F^2 - m|$.

7 Proof of the main results

In this section we provide a proof of our main results presented in Sections 3, 4, and 5, which are, particularly, Theorems 3.1 and 4.1, Lemma 5.1, and Theorems 5.2, 5.3, 5.4, and 5.5.

Let us start with the following lemma concerning some properties of the ϑ -LHSC barrier function.

Lemma 7.1. *Let $x \in \text{int } \mathcal{K}$ and $\beta \in (0, 1)$ be given. Then the following statements hold for the ϑ -LHSC barrier function B .*

- (i) $(\|\nabla B(x)\|_x^*)^2 = -x^T \nabla B(x) = \|x\|_x^2 = \vartheta$.
- (ii) $-\nabla B(x) \in \text{int } \mathcal{K}^*$.
- (iii) $\{y : \|y - x\|_x < 1\} \subset \text{int } \mathcal{K}$.
- (iv) *For any y satisfying $\|y - x\|_x \leq \beta$, it holds that*

$$(1 - \beta)\|v\|_x \leq \|v\|_y \leq (1 - \beta)^{-1}\|v\|_x, \quad \forall v \in \mathbb{R}^n, \quad (55)$$

$$(1 - \beta)\|v\|_x^* \leq \|v\|_y^* \leq (1 - \beta)^{-1}\|v\|_x^*, \quad \forall v \in \mathbb{R}^n. \quad (56)$$

(v) $\{s : \|s + \nabla B(x)\|_x^* \leq 1\} \subseteq \mathcal{K}^*$.

(vi) $\|\nabla^2 B(y) - \nabla^2 B(x)\|_x^* \leq \frac{2-\beta}{(1-\beta)^2} \|y - x\|_x$ holds for all y with $\|y - x\|_x \leq \beta$.

Proof. The proof of statements (i)-(v) can be found in [43, Lemma 1].

We next prove statement (vi). Let y be such that $\|y - x\|_x \leq \beta$. It follows from [56, Theorem 2.1.1] that

$$(1 - \|y - x\|_x)^2 I \preceq \nabla^2 B(x)^{-1/2} \nabla^2 B(y) \nabla^2 B(x)^{-1/2} \preceq (1 - \|y - x\|_x)^{-2} I. \quad (57)$$

By (4), (57), and $\|y - x\|_x \leq \beta$, one has

$$\begin{aligned} \|\nabla^2 B(y) - \nabla^2 B(x)\|_x^* &\stackrel{(4)}{=} \max_{\|u\| \leq 1} \|\nabla^2 B(x)^{-1/2} (\nabla^2 B(y) - \nabla^2 B(x)) \nabla^2 B(x)^{-1/2} u\| \\ &= \|\nabla^2 B(x)^{-1/2} \nabla^2 B(y) \nabla^2 B(x)^{-1/2} - I\| \\ &\stackrel{(57)}{\leq} \max\{1 - (1 - \|y - x\|_x)^2, (1 - \|y - x\|_x)^{-2} - 1\} = (1 - \|y - x\|_x)^{-2} - 1 \\ &= \frac{2 - \|y - x\|_x}{(1 - \|y - x\|_x)^2} \|y - x\|_x \leq \frac{2 - \beta}{(1 - \beta)^2} \|y - x\|_x, \end{aligned}$$

where the last inequality is due to $\|y - x\|_x \leq \beta$. Hence, statement (vi) holds as desired. \square

7.1 Proof of the main results in Section 3

In this subsection we provide a proof of Theorems 3.1.

Proof of Theorem 3.1. By $M \in \nabla^{-2} B(x^*)$, the full column rank of $M^{1/2} \nabla c(x^*)$, and also the discussion in Section 3, one knows that Robinson's constraint qualification holds at x^* . Since x^* is a local minimizer of (1), it then follows from [62, Theorem 3.38] that there exists some $\lambda^* \in \mathbb{R}^m$ such that

$$\nabla f(x^*) + \nabla c(x^*) \lambda^* \in -\mathcal{N}_K(x^*). \quad (58)$$

Further, by [43, Proposition 1], one knows that (58) holds if and only if (5) and (6) hold. Consequently, (5) and (6) hold as desired.

We next prove (7). It follows from Lemma 2.1 that $\{x^* + M^{1/2}d : \|d\| < 1\} \subseteq \mathcal{K}$. Using this and the fact that x^* is a local minimizer of (1), we see that $d^* = 0$ is a local minimizer of the problem

$$\min_d \left\{ f(x^* + M^{1/2}d) : c(x^* + M^{1/2}d) = 0 \right\}. \quad (59)$$

In addition, since $M^{1/2} \nabla c(x^*)$ has full column rank, it is clear that LICQ holds at $d^* = 0$ for (59). By the first- and second-order optimality conditions of (59) at $d^* = 0$, there exists some $\tilde{\lambda}^* \in \mathbb{R}^m$ such that

$$M^{1/2}(\nabla f(x^*) + \nabla c(x^*) \tilde{\lambda}^*) = 0, \quad (60)$$

$$d^T M^{1/2} \left(\nabla^2 f(x^*) + \sum_{i=1}^m \tilde{\lambda}_i^* \nabla^2 c_i(x^*) \right) M^{1/2} d \geq 0, \quad \forall d \in \{d : \nabla c(x^*)^T M^{1/2} d = 0\}. \quad (61)$$

In view of (6), (60), and the fact that $M^{1/2} \nabla c(x^*)$ has full column rank, one can see that $\tilde{\lambda}^* = \lambda^*$. Using this and (61), we conclude that (7) holds. \square

7.2 Proof of the main results in Section 4

In this subsection we first establish several technical lemmas and then use them to prove Theorem 4.1.

To proceed, by (16) and the definitions of local norms, one can verify that

$$\|d\| = \|M_t d\|_{x^t}, \quad \|d\|_{x^t}^* = \|M_t d\|, \quad \|H\|_{x^t}^* = \|M_t H M_t\|, \quad \forall d \in \mathbb{R}^n, H \in \mathbb{R}^{n \times n}. \quad (62)$$

In addition, as a consequence of Assumption 4.1(b) and Lemma 7.1(vi), one can observe that ϕ_μ is locally Lipschitz continuous in Ω with respect to the local norms, i.e.,

$$\|\nabla^2 \phi_\mu(y) - \nabla^2 \phi_\mu(x)\|_x^* \leq L_H^\phi \|y - x\|_x, \quad \forall x, y \in \Omega \text{ with } \|y - x\|_x \leq \beta, \quad (63)$$

where

$$L_H^\phi := L_H^F + \mu(2 - \beta)/(1 - \beta)^2. \quad (64)$$

The following lemma directly follows from (63). Its proof can be found in [43, Lemma 3].

Lemma 7.2. *Under Assumption 4.1(b), the following inequalities hold:*

$$\|\nabla \phi_\mu(y) - \nabla \phi_\mu(x) - \nabla^2 \phi_\mu(x)(y - x)\|_x^* \leq \frac{1}{2} L_H^\phi \|y - x\|_x^2, \quad \forall x, y \in \Omega \text{ with } \|y - x\|_x \leq \beta, \quad (65)$$

$$\phi_\mu(y) \leq \phi_\mu(x) + \nabla \phi_\mu(x)^T (y - x) + \frac{1}{2} (y - x)^T \nabla^2 \phi_\mu(x) (y - x) + \frac{1}{6} L_H^\phi \|y - x\|_x^3, \quad \forall x, y \in \Omega \text{ with } \|y - x\|_x \leq \beta, \quad (66)$$

where Ω and L_H^ϕ are given in Assumption 4.1(b) and (64), respectively.

The following lemma shows that all iterates generated by Algorithm 1 lie in \mathcal{S} .

Lemma 7.3. *Suppose that Assumption 4.1 holds. Let $\{x^t\}_{t \in \mathbb{T}}$ be all the iterates generated by Algorithm 1, where \mathbb{T} is a subset of consecutive nonnegative integers starting from 0. Then $\{x^t\}_{t \in \mathbb{T}} \subset \mathcal{S}$, where \mathcal{S} is defined in (23).*

Proof. We first prove $\{x^t\}_{t \in \mathbb{T}} \subset \text{int } \mathcal{K}$ by induction. Observe from Algorithm 1 that $x^0 = u^0 \in \text{int } \mathcal{K}$. Suppose that $x^t \in \text{int } \mathcal{K}$ is generated at iteration t of Algorithm 1 and x^{t+1} is generated at iteration $t+1$. We next prove $x^{t+1} \in \text{int } \mathcal{K}$. Indeed, observe from Algorithm 1 that $x^{t+1} = x^t + \alpha_t M_t d^t$ with $\alpha_t \in (0, 1]$ and d^t given in one of (17)-(19). It follows from (17)-(19) that $\|d^t\| \leq \beta$. By these and the first relation in (62), one has

$$\|x^{t+1} - x^t\|_{x^t} = \alpha_t \|M_t d^t\|_{x^t} \leq \|M_t d^t\|_{x^t} \stackrel{(62)}{=} \|d^t\| \leq \beta, \quad (67)$$

which, along with $x^t \in \text{int } \mathcal{K}$, $\beta < 1$ and Lemma 7.1(iii), implies that $x^{t+1} \in \text{int } \mathcal{K}$. Hence, the induction is completed, and we have $\{x^t\}_{t \in \mathbb{T}} \subset \text{int } \mathcal{K}$.

In addition, observe from Algorithm 1 that $\{\phi_\mu(x^t)\}_{t \in \mathbb{T}}$ is descent. By this, $x^0 = u^0$, $\{x^t\}_{t \in \mathbb{T}} \subset \text{int } \mathcal{K}$, and (23), one can see that $\{x^t\}_{t \in \mathbb{T}} \subset \mathcal{S}$. \square

The following lemma provides some properties of the output of Algorithm 3, whose proof is similar to the ones of [60, Lemma 3] and [58, Lemma 7] and thus omitted here.

Lemma 7.4. *Suppose that Assumption 4.1 holds and the direction d^t results from the output \hat{d}^t of Algorithm 3 with a type specified in d_type at some iteration t of Algorithm 1. Let M_t be given in (16) and $\gamma_t := \max\{\|\hat{d}^t\|/\beta, 1\}$. Then the following statements hold.*

(i) *If $d_type= SOL$, then d^t satisfies*

$$\epsilon_H \|d^t\|^2 \leq (d^t)^T (M_t^T \nabla^2 \phi_\mu(x^t) M_t + 2\epsilon_H I) d^t, \quad (68)$$

$$\|d^t\| \leq 1.1 \epsilon_H^{-1} \|M_t^T \nabla \phi_\mu(x^t)\|, \quad (69)$$

$$(d^t)^T M_t^T \nabla \phi_\mu(x^t) = -\gamma_t (d^t)^T (M_t^T \nabla^2 \phi_\mu(x^t) M_t + 2\epsilon_H I) d^t. \quad (70)$$

If $\|\hat{d}^t\| \leq \beta$, then d^t also satisfies

$$\|(M_t^T \nabla^2 \phi_\mu(x^t) M_t + 2\epsilon_H I) d^t + M_t^T \nabla \phi_\mu(x^t)\| \leq \epsilon_H \zeta \|d^t\|/2. \quad (71)$$

(ii) *If $d_type= NC$, then d^t satisfies $(d^t)^T M_t^T \nabla \phi_\mu(x^t) \leq 0$ and*

$$\frac{(d^t)^T M_t^T \nabla^2 \phi_\mu(x^t) M_t d^t}{\|d^t\|^2} \leq -\|d^t\| \leq -\epsilon_H. \quad (72)$$

The following lemma shows that when the search direction d^t in Algorithm 1 is of type ‘SOL’, the line search step results in a sufficient reduction on ϕ_μ .

Lemma 7.5. *Suppose that Assumption 4.1 holds and the direction d^t results from the output \hat{d}^t of Algorithm 3 with d_type=SOL at some iteration t of Algorithm 1. Then the following statements hold.*

(i) *The step length α_t is well-defined, and moreover,*

$$\alpha_t \geq \min \left\{ 1, \sqrt{\frac{\min\{6(1-\eta), 2\}}{1.1[L_H^F + \mu(2-\beta)/(1-\beta)^2](U_g^F + \mu\sqrt{\vartheta})}} \theta \epsilon_H \right\}. \quad (73)$$

(ii) *The next iterate $x^{t+1} = x^t + \alpha_t M_t d^t$ satisfies*

$$\phi_\mu(x^t) - \phi_\mu(x^{t+1}) \geq c_{\text{sol}} \min\{(\|\nabla \phi_\mu(x^{t+1})\|_{x^{t+1}}^*)^2 \epsilon_H^{-1}, \epsilon_H^3\}, \quad (74)$$

where M_t and c_{sol} are given in (16) and (26), respectively.

Proof. Notice from Lemma 7.3 that $x^t \in \mathcal{S}$, that is, $x^t \in \text{int } \mathcal{K}$ and $\phi_\mu(x^t) \leq \phi_\mu(u^0)$. It then follows from (16), (24) and Lemma 7.1(i) that

$$\|M_t^T \nabla \phi_\mu(x^t)\| = \|\nabla \phi_\mu(x^t)\|_{x^t}^* \leq \|\nabla F(x^t)\|_{x^t}^* + \mu \|\nabla B(x^t)\|_{x^t}^* \leq U_g^F + \mu \sqrt{\vartheta}. \quad (75)$$

Since d^t results from the output of Algorithm 3 with d_type=SOL, one can see that $\|M_t^T \nabla \phi_\mu(x^t)\| > \epsilon_g$ and the relations (68)-(70) hold. Also, one can observe from Algorithm 3 that its output \hat{d}^t satisfies

$$\|(M_t^T \nabla^2 \phi_\mu(x^t) M_t + 2\epsilon_H I) \hat{d}^t + M_t^T \nabla \phi_\mu(x^t)\| \leq \hat{\zeta} \|M_t^T \nabla \phi_\mu(x^t)\|$$

for some $\hat{\zeta} \in (0, 1/6)$, which together with $\|M_t^T \nabla \phi_\mu(x^t)\| > \epsilon_g$ implies that $\hat{d}^t \neq 0$. It then follows from this and (18) that $d^t \neq 0$.

We first prove statement (i). If (20) holds for $j = 0$, then $\alpha_t = 1$, which clearly implies that (73) holds. We now suppose that (20) fails for $j = 0$. Claim that for all $j \geq 0$ that violate (20), it holds that

$$\theta^{2j} \geq \min\{6(1-\eta), 2\} \epsilon_H (L_H^\phi)^{-1} \|d^t\|^{-1}, \quad (76)$$

where L_H^ϕ is defined in (64). Indeed, we suppose that (20) is violated by some $j \geq 0$. We next show that (76) holds for such j by considering two separate cases below.

Case 1) $\phi_\mu(x^t + \theta^j M_t d^t) > \phi_\mu(x^t)$. Let $\varphi(\alpha) = \phi_\mu(x^t + \alpha M_t d^t)$. Then $\varphi(\theta^j) > \varphi(0)$. In addition, by (68), (70), $\gamma_t = \max\{\|\hat{d}^t\|/\beta, 1\} \geq 1$, and $d^t \neq 0$, one has

$$\varphi'(0) = (d^t)^T M_t^T \nabla \phi_\mu(x^t) \stackrel{(70)}{=} -\gamma_t (d^t)^T (M_t^T \nabla^2 \phi_\mu(x^t) M_t + 2\epsilon_H I) d^t \stackrel{(68)}{\leq} -\gamma_t \epsilon_H \|d^t\|^2 < 0.$$

In view of these, we can observe that there exists a local minimizer $\alpha^* \in (0, \theta^j)$ of φ such that $\varphi(\alpha^*) < \varphi(0)$ and

$$\varphi'(\alpha^*) = \nabla \phi_\mu(x^t + \alpha^* M_t d^t)^T M_t d^t = 0. \quad (77)$$

By $\phi_\mu(x^t) \leq \phi_\mu(u^0)$ and $\varphi(\alpha^*) < \varphi(0)$, one has $\phi_\mu(x^t + \alpha^* M_t d^t) < \phi_\mu(u^0)$. In addition, using (67) and $0 < \alpha^* < \theta^j \leq 1$, we have $\|\alpha^* M_t d^t\|_{x^t} \leq \|M_t d^t\|_{x^t} \leq \beta$. Hence, (65) holds for $x = x^t$ and $y = x^t + \alpha^* M_t d^t$. By this, (62), (68), (70), (77), $0 < \alpha^* < 1$ and $\gamma_t \geq 1$, one has

$$\begin{aligned} \frac{(\alpha^*)^2 L_H^\phi}{2} \|d^t\|^3 &\stackrel{(62)}{=} \frac{(\alpha^*)^2 L_H^\phi}{2} \|d^t\| \|M_t d^t\|_{x^t}^2 \stackrel{(65)}{\geq} \|d^t\| \|\nabla \phi_\mu(x^t + \alpha^* M_t d^t) - \nabla \phi_\mu(x^t) - \alpha^* \nabla^2 \phi_\mu(x^t) M_t d^t\|_{x^t}^* \\ &\geq (d^t)^T (M_t^T \nabla \phi_\mu(x^t + \alpha^* M_t d^t) - M_t^T \nabla \phi_\mu(x^t) - \alpha^* M_t^T \nabla^2 \phi_\mu(x^t) M_t d^t) \\ &\stackrel{(77)}{=} -(d^t)^T M_t^T \nabla \phi_\mu(x^t) - \alpha^* (d^t)^T M_t^T \nabla^2 \phi_\mu(x^t) M_t d^t \\ &\stackrel{(70)}{=} (\gamma_t - \alpha^*) (d^t)^T (M_t^T \nabla^2 \phi_\mu(x^t) M_t + 2\epsilon_H I) d^t + 2\alpha^* \epsilon_H \|d^t\|^2 \\ &\stackrel{(68)}{\geq} (\gamma_t - \alpha^*) \epsilon_H \|d^t\|^2 + 2\alpha^* \epsilon_H \|d^t\|^2 = (\gamma_t + \alpha^*) \epsilon_H \|d^t\|^2 \geq \epsilon_H \|d^t\|^2, \end{aligned}$$

which along with $d^t \neq 0$ implies that $(\alpha^*)^2 \geq 2\epsilon_H(L_H^\phi)^{-1}\|d^t\|^{-1}$. Using this and $\theta^j > \alpha^*$, we conclude that (76) holds in this case.

Case 2) $\phi_\mu(x^t + \theta^j M_t d^t) \leq \phi_\mu(x^t)$. By this and $\phi_\mu(x^t) \leq \phi_\mu(u^0)$, one has $\phi_\mu(x^t + \theta^j M_t d^t) \leq \phi_\mu(u^0)$. Also, using (67) and $\theta \in (0, 1)$, we have $\|\theta^j M_t d^t\|_{x^t} \leq \|M_t d^t\|_{x^t} \leq \beta$. Hence, (66) holds for $x = x^t$ and $y = x^t + \theta^j M_t d^t$. Using this, (62), (68), (70) and the fact that j violates (20), we obtain that

$$\begin{aligned}
-\eta\epsilon_H\theta^{2j}\|d^t\|^2 &\leq \phi_\mu(x^t + \theta^j M_t d^t) - \phi_\mu(x^t) \\
&\stackrel{(66)}{\leq} \theta^j \nabla \phi_\mu(x^t)^T M_t d^t + \frac{\theta^{2j}}{2} (d^t)^T M_t^T \nabla^2 \phi_\mu(x^t) M_t d^t + \frac{L_H^\phi}{6} \theta^{3j} \|M_t d^t\|_{x^t}^3 \\
&\stackrel{(62)(70)}{=} -\theta^j \gamma_t (d^t)^T (M_t^T \nabla^2 \phi_\mu(x^t) M_t + 2\epsilon_H I) d^t + \frac{\theta^{2j}}{2} (d^t)^T M_t^T \nabla^2 \phi_\mu(x^t) M_t d^t + \frac{L_H^\phi}{6} \theta^{3j} \|d^t\|^3 \\
&= -\theta^j \left(\gamma_t - \frac{\theta^j}{2} \right) (d^t)^T (M_t^T \nabla^2 \phi_\mu(x^t) M_t + 2\epsilon_H I) d^t - \theta^{2j} \epsilon_H \|d^t\|^2 + \frac{L_H^\phi}{6} \theta^{3j} \|d^t\|^3 \\
&\stackrel{(68)}{\leq} -\theta^j \left(\gamma_t - \frac{\theta^j}{2} \right) \epsilon_H \|d^t\|^2 - \theta^{2j} \epsilon_H \|d^t\|^2 + \frac{L_H^\phi}{6} \theta^{3j} \|d^t\|^3 \leq -\theta^j \epsilon_H \gamma_t \|d^t\|^2 + \frac{L_H^\phi}{6} \theta^{3j} \|d^t\|^3, \quad (78)
\end{aligned}$$

where the first inequality is due to the violation of (20) by such j . Recall that $d^t \neq 0$. Dividing both sides of (78) by $L_H^\phi \theta^j \|d^t\|^3 / 6$ and using $\eta, \theta \in (0, 1)$ and $\gamma_t \geq 1$, we have

$$\theta^{2j} \geq 6(\gamma_t - \eta\theta^j)\epsilon_H(L_H^\phi)^{-1}\|d^t\|^{-1} \geq 6(1 - \eta)\epsilon_H(L_H^\phi)^{-1}\|d^t\|^{-1}.$$

Hence, (76) also holds in this case.

Combining the above two cases, we conclude that (76) holds for any $j \geq 0$ violating (20). By this and $\theta \in (0, 1)$, one can see that all $j \geq 0$ that violate (20) must be bounded above. It then follows that the step length α_t associated with (20) is well-defined. We next prove (73). Observe from the definition of j_t in Algorithm 1 that $j = j_t - 1$ violates (20) and hence (76) holds for $j = j_t - 1$. Then, by (64), (76) with $j = j_t - 1$, and $\alpha_t = \theta^{j_t}$, one has

$$\begin{aligned}
\alpha_t &= \theta^{j_t} \geq \sqrt{\min\{6(1 - \eta), 2\}\epsilon_H(L_H^\phi)^{-1}\theta\|d^t\|^{-1/2}} \\
&= \sqrt{\min\{6(1 - \eta), 2\}\epsilon_H[L_H^F + \mu(2 - \beta)/(1 - \beta)^2]^{-1}\theta\|d^t\|^{-1/2}}, \quad (79)
\end{aligned}$$

which along with (69) and (75) implies that (73) holds.

We next prove statement (ii), particularly, (74) by considering three separate cases below.

Case 1) $\alpha_t = 1$ and $\|\hat{d}^t\| \geq \beta$. It then follows from (18) that $d^t = \beta\hat{d}^t/\|\hat{d}^t\|$. Notice from Algorithm 1 that $\beta \geq \epsilon_H$. Using this and $d^t = \beta\hat{d}^t/\|\hat{d}^t\|$, we see that $\|d^t\| = \beta \geq \epsilon_H$, which together with (20) and $\alpha_t = 1$ implies that (74) holds.

Case 2) $\alpha_t = 1$ and $\|\hat{d}^t\| < \beta$. Notice from $\alpha_t = 1$ that $j = 0$ is accepted by (20). Then one can see that $\phi_\mu(x^t + M_t d^t) \leq \phi_\mu(x^t) \leq \phi_\mu(u^0)$. Also, observe from (67) that $\|M_t d^t\|_{x^t} \leq \beta$. Hence, (65) holds for $x = x^t$ and $y = x^t + M_t d^t$. By these, (56) and (71), one has

$$\begin{aligned}
(1 - \beta)\|\nabla \phi_\mu(x^{t+1})\|_{x^{t+1}}^* &\stackrel{(56)}{\leq} \|\nabla \phi_\mu(x^{t+1})\|_{x^t}^* = \|\nabla \phi_\mu(x^t + M_t d^t)\|_{x^t}^* \\
&\leq \|\nabla \phi_\mu(x^t + M_t d^t) - \nabla \phi_\mu(x^t) - \nabla^2 \phi_\mu(x^t) M_t d^t\|_{x^t}^* + \|\nabla \phi_\mu(x^t) + \nabla^2 \phi_\mu(x^t) M_t d^t\|_{x^t}^* \\
&= \|\nabla \phi_\mu(x^t + M_t d^t) - \nabla \phi_\mu(x^t) - \nabla^2 \phi_\mu(x^t) M_t d^t\|_{x^t}^* + \|M_t^T (\nabla \phi_\mu(x^t) + \nabla^2 \phi_\mu(x^t) M_t d^t)\|_{x^t}^* \\
&\leq \|\nabla \phi_\mu(x^t + M_t d^t) - \nabla \phi_\mu(x^t) - \nabla^2 \phi_\mu(x^t) M_t d^t\|_{x^t}^* \\
&\quad + \|(M_t^T \nabla^2 \phi_\mu(x^t) M_t + 2\epsilon_H I) d^t + M_t^T \nabla \phi_\mu(x^t)\| + 2\epsilon_H \|d^t\| \\
&\stackrel{(65)(71)}{\leq} L_H^\phi \|M_t d^t\|_{x^t}^2 / 2 + (4 + \zeta)\epsilon_H \|d^t\| / 2 \stackrel{(62)}{=} L_H^\phi \|d^t\|^2 / 2 + (4 + \zeta)\epsilon_H \|d^t\| / 2,
\end{aligned}$$

where the second inequality is due to the triangle inequality, and the second equality follows from (4) and the second relation in (62). Solving the above inequality for $\|d^t\|$ and using (64) and the fact that $\|d^t\| > 0$, we

obtain that

$$\begin{aligned}
\|d^t\| &\geq \frac{-(4+\zeta)\epsilon_H + \sqrt{(4+\zeta)^2\epsilon_H^2 + 8(1-\beta)L_H^\phi \|\nabla\phi_\mu(x^{t+1})\|_{x^{t+1}}^*}}{2L_H^\phi} \\
&\geq \frac{-(4+\zeta)\epsilon_H + \sqrt{(4+\zeta)^2\epsilon_H^2 + 8(1-\beta)L_H^\phi \epsilon_H^2}}{2L_H^\phi} \min\{\|\nabla\phi_\mu(x^{t+1})\|_{x^{t+1}}^* \epsilon_H^{-2}, 1\} \\
&= \frac{\frac{4(1-\beta)}{4+\zeta + \sqrt{(4+\zeta)^2 + 8(1-\beta)L_H^\phi}} \min\{\|\nabla\phi_\mu(x^{t+1})\|_{x^{t+1}}^* \epsilon_H^{-1}, \epsilon_H\}}{\frac{4(1-\beta)}{4+\zeta + \sqrt{(4+\zeta)^2 + 8[(1-\beta)L_H^F + \mu(2-\beta)/(1-\beta)]}}} \min\{\|\nabla\phi_\mu(x^{t+1})\|_{x^{t+1}}^* \epsilon_H^{-1}, \epsilon_H\}, \\
&\stackrel{(64)}{=} \frac{\frac{4(1-\beta)}{4+\zeta + \sqrt{(4+\zeta)^2 + 8[(1-\beta)L_H^F + \mu(2-\beta)/(1-\beta)]}} \min\{\|\nabla\phi_\mu(x^{t+1})\|_{x^{t+1}}^* \epsilon_H^{-1}, \epsilon_H\}},
\end{aligned}$$

where the second inequality follows from the inequality $-a + \sqrt{a^2 + bs} \geq (-a + \sqrt{a^2 + b}) \min\{s, 1\}$ for all $a, b, s \geq 0$, which can be easily verified by performing a rationalization to the terms $-a + \sqrt{a^2 + bs}$ and $-a + \sqrt{a^2 + b}$, respectively. In view of this, $\alpha_t = 1$, (20) and (26), one can see that (74) holds.

Case 3) $\alpha_t < 1$. By this, one has that $j = 0$ violates (20) and hence (76) holds for $j = 0$. Letting $j = 0$ in (76), we obtain that $\|d^t\| \geq \min\{6(1-\eta), 2\}\epsilon_H/L_H^\phi$, which along with (20), (64) and (79) implies that

$$\phi_\mu(x^t) - \phi_\mu(x^{t+1}) \stackrel{(20)}{\geq} \eta\epsilon_H\theta^{2j_t}\|d^t\|^2 \geq \eta \left[\frac{\min\{6(1-\eta), 2\}\theta}{L_H^\phi} \right]^2 \epsilon_H^3 \stackrel{(64)}{=} \eta \left[\frac{\min\{6(1-\eta), 2\}\theta}{L_H^F + \mu(2-\beta)/(1-\beta)^2} \right]^2 \epsilon_H^3.$$

By this and (26), one can immediately see that (74) also holds in this case. \square

The next lemma shows that when the search direction d^t in Algorithm 1 is of type ‘NC’, the line search step results in a sufficient reduction on ϕ_μ as well.

Lemma 7.6. *Suppose that Assumption 4.1 holds and the direction d^t results from either the output \hat{d}^t of Algorithm 3 with $d_type=NC$ or the output v of Algorithm 4 at some iteration t of Algorithm 1. Then the following statements hold.*

(i) *The step length α_t is well-defined, and moreover,*

$$\alpha_t \geq \min \left\{ 1, \frac{\min\{1, 3(1-\eta)\}\theta}{L_H^F + \mu(2-\beta)/(1-\beta)^2} \right\}. \quad (80)$$

(ii) *The next iterate $x^{t+1} = x^t + \alpha_t M_t d^t$ satisfies $\phi_\mu(x^t) - \phi_\mu(x^{t+1}) \geq c_{nc}\epsilon_H^3$, where M_t and c_{nc} are given in (16) and (27), respectively.*

Proof. It follows from Lemma 7.3 that $x^t \in \mathcal{S}$, that is, $x^t \in \text{int } \mathcal{K}$ and $\phi_\mu(x^t) \leq \phi_\mu(u^0)$. By the assumption on d^t , one can see from Algorithm 1 that d^t is a negative curvature direction given in (17) or (19) and thus $d^t \neq 0$. Also, the vector v satisfies $\|v\| = 1$ whenever it is returned from Algorithm 4. By these, Lemma 7.4(ii), (17) and (19), one has

$$\nabla\phi_\mu(x^t)^T M_t d^t \leq 0, \quad (d^t)^T M_t^T \nabla^2 \phi_\mu(x^t) M_t d^t \leq -\|d^t\|^3 < 0. \quad (81)$$

We first prove statement (i). If (21) holds for $j = 0$, then $\alpha_t = 1$, which clearly implies that (80) holds. We now suppose that (21) fails for $j = 0$. Claim that for all $j \geq 0$ that violate (21), it holds that

$$\theta^j \geq \min\{1, 3(1-\eta)\}/L_H^\phi, \quad (82)$$

where L_H^ϕ is defined in (64). Indeed, suppose that (21) is violated by some $j \geq 0$. We now prove that (82) holds for such j by considering two separate cases below.

Case 1) $\phi_\mu(x^t + \theta^j M_t d^t) > \phi_\mu(x^t)$. Let $\varphi(\alpha) = \phi_\mu(x^t + \alpha M_t d^t)$. Then $\varphi(\theta^j) > \varphi(0)$. Also, by (81), one has

$$\varphi'(0) = \nabla\phi_\mu(x^t)^T M_t d^t \leq 0, \quad \varphi''(0) = (d^t)^T M_t^T \nabla^2 \phi_\mu(x^t) M_t d^t < 0.$$

From these, we can observe that there exists a local minimizer $\alpha^* \in (0, \theta^j)$ of φ such that $\varphi(\alpha^*) < \varphi(0)$. By the second-order necessary optimality condition of φ at α^* , one has

$$\varphi''(\alpha^*) = (d^t)^T M_t^T \nabla^2 \phi_\mu(x^t + \alpha^* M_t d^t) M_t d^t \geq 0. \quad (83)$$

In addition, by $\phi_\mu(x^t) \leq \phi_\mu(u^0)$ and $\varphi(\alpha^*) < \varphi(0)$, one has $\phi_\mu(x^t + \alpha^* M_t d^t) < \phi_\mu(u^0)$. Using (67) and $0 < \alpha^* < \theta^j \leq 1$, we see that $\|\alpha^* M_t d^t\|_{x^t} \leq \|M_t d^t\|_{x^t} \leq \beta$. Hence, (63) holds for $x = x^t$ and $y = x^t + \alpha^* M_t d^t$. Using this, (4), (62), (63), (81) and (83), we obtain that

$$\begin{aligned} L_H^\phi \alpha^* \|d^t\|^3 &\stackrel{(62)}{=} L_H^\phi \alpha^* \|d^t\|^2 \|M_t d^t\|_{x^t} \stackrel{(63)}{\geq} \|d^t\|^2 \|\nabla^2 \phi_\mu(x^t + \alpha^* M_t d^t) - \nabla^2 \phi_\mu(x^t)\|_{x^t}^* \\ &\stackrel{(4)}{=} \|d^t\|^2 \|M_t^T (\nabla^2 \phi_\mu(x^t + \alpha^* M_t d^t) - \nabla^2 \phi_\mu(x^t)) M_t\| \geq (d^t)^T M_t^T (\nabla^2 \phi_\mu(x^t + \alpha^* M_t d^t) - \nabla^2 \phi_\mu(x^t)) M_t d^t \\ &\stackrel{(83)}{\geq} -(d^t)^T M_t^T \nabla^2 \phi_\mu(x^t) M_t d^t \stackrel{(81)}{\geq} \|d^t\|^3. \end{aligned}$$

It then follows from this and $d^t \neq 0$ that $\alpha^* \geq 1/L_H^\phi$, which along with $\theta^j > \alpha^*$ implies that (82) holds in this case.

Case 2) $\phi_\mu(x^t + \theta^j M_t d^t) \leq \phi_\mu(x^t)$. By this and $\phi_\mu(x^t) \leq \phi_\mu(u^0)$, one has $\phi_\mu(x^t + \theta^j M_t d^t) \leq \phi_\mu(u^0)$. In addition, it follows from (67) and $\theta \in (0, 1)$ that $\|\theta^j M_t d^t\|_{x^t} \leq \|M_t d^t\|_{x^t} \leq \beta$. Hence, (66) holds for $x = x^t$ and $y = x^t + \theta^j M_t d^t$. By this, (62), (81) and the fact that j violates (21), one has

$$\begin{aligned} -\frac{\eta}{2} \theta^{2j} \|d^t\|^3 &\leq \phi_\mu(x^t + \theta^j M_t d^t) - \phi_\mu(x^t) \stackrel{(66)}{\leq} \theta^j \nabla \phi_\mu(x^t)^T M_t d^t + \frac{\theta^{2j}}{2} (d^t)^T M_t^T \nabla^2 \phi_\mu(x^t) M_t d^t + \frac{L_H^\phi}{6} \theta^{3j} \|M_t d^t\|_{x^t}^3 \\ &\stackrel{(62)(81)}{\leq} -\frac{\theta^{2j}}{2} \|d^t\|^3 + \frac{L_H^\phi}{6} \theta^{3j} \|d^t\|^3, \end{aligned}$$

where the first inequality is due to the violation of (21) by such j . Using this and $d^t \neq 0$, we see that $\theta^j \geq 3(1 - \eta)/L_H^\phi$. Hence, (82) also holds in this case.

Combining the above two cases, we conclude that (82) holds for all $j \geq 0$ violating (21). By this and $\theta \in (0, 1)$, one can see that all $j \geq 0$ that violate (21) must be bounded above. It then follows that the step length α_t associated with (21) is well-defined. We next derive a lower bound for α_t . Notice that $j = j_t - 1$ violates (21) and hence (82) holds for $j = j_t - 1$. Then by (82) with $j = j_t - 1$ and $\alpha_t = \theta^{j_t}$, one can observe that $\alpha_t = \theta^{j_t} \geq \min\{1, 3(1 - \eta)\} \theta / L_H^\phi$, which along with (64) yields (80) as desired.

We next prove statement (ii) by considering two separate cases below.

Case 1) d^t results from the output \hat{d}^t of Algorithm 3 with d_type=NC. By this and (72), one has $\|d^t\| \geq \epsilon_H$, which along with statement (i) and (21) implies that statement (ii) holds.

Case 2) d^t results from the output v of Algorithm 4. Notice from Algorithm 4 that $\|v\| = 1$ and $v^T M_t^T \nabla^2 \phi_\mu(x^t) M_t v \leq -\epsilon_H/2$. It then follows from (19) and $\beta \geq \epsilon_H$ that $\|d^t\| \geq \epsilon_H/2$. Using this, (21) and statement (i), we see that statement (ii) also holds in this case. \square

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. Notice from Lemma 7.3 that all the iterates generated by Algorithm 1 lie in \mathcal{S} . By this, (4), (62) and (24), one has

$$\|M_t^T \nabla^2 \phi_\mu(x^t) M_t\| \stackrel{(4)(62)}{=} \|\nabla^2 \phi_\mu(x^t)\|_{x^t}^* \leq \|\nabla^2 F(x^t)\|_{x^t}^* + \mu \|\nabla^2 B(x^t)\|_{x^t}^* \leq U_H^F + \mu, \quad (84)$$

where the last inequality follows from (24) and the fact that $\|\nabla^2 B(x^t)\|_{x^t}^* = 1$ due to (4) and (16).

(i) Suppose for contradiction that the total number of calls of Algorithm 4 in Algorithm 1 is more than T_2 . Observe from Algorithm 1 and Lemma 7.6(ii) that each of these calls, except the last one, returns a sufficiently negative curvature direction, and each of them results in a reduction on ϕ_μ at least by $c_{\text{nc}} \epsilon_H^3$. Using this and the fact that $x^0 = u^0$, we obtain that

$$T_2 c_{\text{nc}} \epsilon_H^3 \leq \sum_{t \in \mathbb{T}} [\phi_\mu(x^t) - \phi_\mu(x^{t+1})] \leq \phi_\mu(x^0) - \phi_{\text{low}} = \phi_{\text{hi}} - \phi_{\text{low}},$$

where \mathbb{T} is given in Lemma 7.3. This contradicts with the definition of T_2 given in (25).

(ii) Suppose for contradiction that the total number of calls of Algorithm 3 in Algorithm 1 is more than T_1 . Note that if Algorithm 3 is called at some iteration t and generates x^{t+1} satisfying $\|\nabla \phi_\mu(x^{t+1})\|_{x^{t+1}}^* \leq \epsilon_g$, then Algorithm 4 must be called at the next iteration $t+1$. Using this and statement (i), we see that the total number of such iterations t is at most T_2 . Hence, the total number of iterations t of Algorithm 1 at which Algorithm 3 is

called and generates x^{t+1} satisfying $\|\nabla\phi_\mu(x^{t+1})\|_{x^{t+1}}^* > \epsilon_g$ is at least $T_1 - T_2 + 1$. Moreover, for each of such iterations t , it follows from Lemmas 7.5(ii) and 7.6(ii) that $\phi_\mu(x^t) - \phi_\mu(x^{t+1}) \geq \min\{c_{\text{sol}}, c_{\text{nc}}\} \min\{\epsilon_g^2 \epsilon_H^{-1}, \epsilon_H^3\}$. Thus, one has

$$(T_1 - T_2 + 1) \min\{c_{\text{sol}}, c_{\text{nc}}\} \min\{\epsilon_g^2 \epsilon_H^{-1}, \epsilon_H^3\} \leq \sum_{t \in \mathbb{T}} [\phi_\mu(x^t) - \phi_\mu(x^{t+1})] \leq \phi_{\text{hi}} - \phi_{\text{low}},$$

where \mathbb{T} is given in Lemma 7.3. This contradicts the definitions of T_1 and T_2 given in (25).

(iii) Notice that either Algorithm 3 or Algorithm 4 is called at each iteration of Algorithm 1. By this and statements (i) and (ii), one has that the total number of iterations of Algorithm 1 is at most $T_1 + T_2$. In addition, the relation (28) follows from (25), (26) and (27). It is also not hard to see that the output x^t of Algorithm 1 satisfies $\|\nabla\phi_\mu(x^t)\|_{x^t}^* \leq \epsilon_g$ deterministically and $\lambda_{\min}(M_t^T \nabla^2 \phi_\mu(x^t) M_t) \geq -\epsilon_H$ with probability at least $1 - \delta$ for some $0 \leq t \leq T_1 + T_2$, where the probability is due to Algorithm 4. Hence, statement (iii) holds.

(iv) Recall that each iteration of Algorithm 1 requires Cholesky factorization of $\nabla^2 B$ at one point only. This together with statement (iii) implies that the total number of Cholesky factorizations required by Algorithm 1 is at most $T_1 + T_2$. By (84) and Theorem A.1 with $(H, \varepsilon) = (M_t^T \nabla^2 \phi_\mu(x^t) M_t, \epsilon_H)$, one can see that the number of products of H and a vector v required by each call of Algorithm 3 is at most $\tilde{\mathcal{O}}(\min\{n, [(U_H^F + \mu)/\epsilon_H]^{1/2}\})$. In addition, by (84), Theorem B.1 with $(H, \varepsilon) = (M_t^T \nabla^2 \phi_\mu(x^t) M_t, \epsilon_H)$, and the fact that each iteration of the Lanczos method requires only one product of H and a vector v , one can observe that the number of products of H and a vector v required by each call of Algorithm 4 is also at most $\tilde{\mathcal{O}}(\min\{n, [(U_H^F + \mu)/\epsilon_H]^{1/2}\})$. Recall from Section 4.1 that the product of H and a vector v requires at most three fundamental operations, which are one Hessian-vector product of F , one backward and forward substitutions to a triangular linear system. Hence, each call of Algorithm 3 or 4 requires at most $\tilde{\mathcal{O}}(\min\{n, [(U_H^F + \mu)/\epsilon_H]^{1/2}\})$ fundamental operations. By these observations and statement (iii), we conclude that statement (iv) holds. \square

7.3 Proof of the main results in Section 5

In this subsection we provide a proof of Lemma 5.1 and Theorems 5.2, 5.3, 5.4 and 5.5.

Before proceeding, we recall that $\|c(z_\epsilon)\| \leq \epsilon/2 < 1$. Using this, (30) and (36), we obtain that

$$f(x) + \mu B(x) + \gamma \|\tilde{c}(x)\|^2 \geq f(x) + \mu B(x) + \frac{\gamma}{2} \|c(x)\|^2 - \gamma \|c(z_\epsilon)\|^2 \geq f_{\text{low}} - \gamma, \quad \forall x \in \text{int } \mathcal{K}, \mu \in (0, \mu_0], \quad (85)$$

where the first inequality follows from (36) and $\|a - b\|^2 \geq \|a\|^2/2 - \|b\|^2$ for all $a, b \in \mathbb{R}^n$. In addition, by (29), the first relation in (38), and the fact that $0 < \mu_k \leq \mu_0$ (see Algorithm 2), one has

$$\mathcal{L}_{\mu_k}(x^{k+1}, \lambda^k; \rho_k) \leq f_{\text{hi}} \text{ whenever } x^{k+1} \text{ is generated.} \quad (86)$$

We next present an auxiliary lemma that will be frequently used later. Its proof is identical to the one of [44, Lemma 5.4] with f replaced by $f + \mu B$, and thus omitted here.

Lemma 7.7. *Suppose that Assumption 5.1 holds. Let $\gamma, \mu_0, f_{\text{hi}}$ and f_{low} be given in Assumption 5.1. Assume that $\rho > 2\gamma$, $\mu \in (0, \mu_0]$, $\lambda \in \mathbb{R}^m$ and $x \in \text{int } \mathcal{K}$ satisfy $\mathcal{L}_\mu(x, \lambda; \rho) \leq f_{\text{hi}}$, where \mathcal{L}_μ is defined in (37). Then the following statements hold.*

- (i) $f(x) + \mu B(x) \leq f_{\text{hi}} + \|\lambda\|^2/(2\rho)$.
- (ii) $\|\tilde{c}(x)\| \leq \sqrt{2(f_{\text{hi}} - f_{\text{low}} + \gamma)/(\rho - 2\gamma)} + \|\lambda\|^2/(\rho - 2\gamma)^2 + \|\lambda\|/(\rho - 2\gamma)$.
- (iii) *If $\rho \geq \|\lambda\|^2/(2\tilde{\delta}_f)$ for some $\tilde{\delta}_f > 0$, then $f(x) + \mu B(x) \leq f_{\text{hi}} + \tilde{\delta}_f$.*
- (iv) *If $\rho \geq 2(f_{\text{hi}} - f_{\text{low}} + \gamma)\tilde{\delta}_c^{-2} + 2\|\lambda\|\tilde{\delta}_c^{-1} + 2\gamma$ for some $\tilde{\delta}_c > 0$, then $\|\tilde{c}(x)\| \leq \tilde{\delta}_c$.*

The following lemma establishes the local Lipschitz continuity of c_i and ∇c_i with respect to the local norm.

Lemma 7.8. *Under Assumption 5.1, the following inequalities hold:*

$$|c_i(y) - c_i(x)| \leq \frac{U_g^c}{1 - \beta} \|y - x\|_x, \quad \forall x, y \in \Omega(\delta_f, \delta_c) \text{ with } \|y - x\|_x \leq \beta, \quad 1 \leq i \leq m, \quad (87)$$

$$\|\nabla c_i(y) - \nabla c_i(x)\|_x^* \leq \frac{U_H^c}{(1-\beta)^2} \|y - x\|_x, \quad \forall x, y \in \Omega(\delta_f, \delta_c) \text{ with } \|y - x\|_x \leq \beta, \quad 1 \leq i \leq m, \quad (88)$$

where $\Omega(\delta_f, \delta_c)$ is given in Assumption 5.1, and U_g^c, U_H^c are defined in (33) and (34), respectively.

Proof. Fix any $x, y \in \Omega(\delta_f, \delta_c)$ with $\|y - x\|_x \leq \beta$ and any $1 \leq i \leq m$. Let $z_t = x + t(y - x)$ for all $t \in [0, 1]$. It then follows that $\|z_t - x\|_x \leq \beta$ and $z_t \in \Omega(\delta_f, \delta_c)$. By these, (33), (34), (55) and (56), one has

$$\|\nabla c_i(z_t)\|_{z_t}^* \leq U_g^c, \quad \|\nabla^2 c_i(z_t)\|_{z_t}^* \leq U_H^c, \quad \|v\|_{z_t} \leq (1-\beta)^{-1} \|v\|_x, \quad \|v\|_x^* \leq (1-\beta)^{-1} \|v\|_{z_t}^*, \quad \forall v \in \mathbb{R}^n, t \in [0, 1].$$

By virtue of these and (4), we obtain

$$\begin{aligned} |c_i(y) - c_i(x)| &= \left| \int_0^1 \nabla c_i(z_t)^T (y - x) dt \right| \leq \int_0^1 \|\nabla c_i(z_t)\|_{z_t}^* \|y - x\|_{z_t} dt \leq \frac{U_g^c}{1-\beta} \|y - x\|_x, \\ \|\nabla c_i(y) - \nabla c_i(x)\|_x^* &= \left\| \int_0^1 \nabla^2 c_i(z_t)(y - x) dt \right\|_x^* \leq \int_0^1 \|\nabla^2 c_i(z_t)(y - x)\|_x^* dt \leq \frac{1}{1-\beta} \int_0^1 \|\nabla^2 c_i(z_t)(y - x)\|_{z_t}^* dt \\ &\leq \frac{1}{1-\beta} \int_0^1 \|\nabla^2 c_i(z_t)\|_{z_t}^* \|y - x\|_{z_t} dt \leq \frac{U_H^c}{(1-\beta)^2} \|y - x\|_x. \end{aligned}$$

Hence, (87) and (88) hold. \square

We are now ready to prove Lemma 5.1.

Proof of Lemma 5.1. The proofs of statements (i) and (ii) follow similar arguments to those used in proving [44, Lemma 4.1]

(iii) Fix $x, y \in \Omega(\delta_f, \delta_c)$ with $\|y - x\|_x \leq \beta$ and $1 \leq i \leq m$. By this, (32), (34), (36), (44), (87) and $\|c(z_\epsilon)\| \leq 1$, one has

$$\begin{aligned} \|\tilde{c}_i(y)\nabla^2 c_i(y) - \tilde{c}_i(x)\nabla^2 c_i(x)\|_x^* &= \|\tilde{c}_i(y)(\nabla^2 c_i(y) - \nabla^2 c_i(x)) + (\tilde{c}_i(y) - \tilde{c}_i(x))\nabla^2 c_i(x)\|_x^* \\ &\leq |c_i(y) - c_i(z_\epsilon)| \|\nabla^2 c_i(y) - \nabla^2 c_i(x)\|_x^* + |c_i(y) - c_i(x)| \|\nabla^2 c_i(x)\|_x^* \leq (1+U^c)L_H^c \|y - x\|_x + \frac{U_g^c U_H^c}{1-\beta} \|y - x\|_x \\ &= \left[(1+U^c)L_H^c + \frac{U_g^c U_H^c}{1-\beta} \right] \|y - x\|_x. \end{aligned} \quad (89)$$

In addition, by (33), (56) and (88), one has

$$\begin{aligned} \|\nabla c_i(y)\nabla c_i(y)^T - \nabla c_i(x)\nabla c_i(x)^T\|_x^* &= \|\nabla c_i(y)(\nabla c_i(y) - \nabla c_i(x))^T + (\nabla c_i(y) - \nabla c_i(x))\nabla c_i(x)^T\|_x^* \\ &\leq \|\nabla c_i(y)\|_x^* \|\nabla c_i(y) - \nabla c_i(x)\|_x^* + \|\nabla c_i(x)\|_x^* \|\nabla c_i(y) - \nabla c_i(x)\|_x^* \\ &\leq \left(\frac{1}{1-\beta} \|\nabla c_i(y)\|_y^* + \|\nabla c_i(x)\|_x^* \right) \|\nabla c_i(y) - \nabla c_i(x)\|_x^* \leq \frac{(2-\beta)U_g^c U_H^c}{(1-\beta)^3} \|y - x\|_x. \end{aligned} \quad (90)$$

In view of (42) and the fact that $\nabla \tilde{c} = \nabla c$ and $\nabla^2 \tilde{c}_i = \nabla^2 c_i$, $1 \leq i \leq m$, we see that

$$\nabla_{xx}^2 \mathcal{L}(x, \lambda^k; \rho_k) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i^k \nabla^2 c_i(x) + \rho_k \sum_{i=1}^m (\nabla c_i(x)\nabla c_i(x)^T + \tilde{c}_i(x)\nabla^2 c_i(x)), \quad (91)$$

which implies that

$$\begin{aligned} \|\nabla_{xx}^2 \mathcal{L}(y, \lambda^k; \rho_k) - \nabla_{xx}^2 \mathcal{L}(x, \lambda^k; \rho_k)\|_x^* &\leq \|\nabla^2 f(y) - \nabla^2 f(x)\|_x^* + \sum_{i=1}^m |\lambda_i^k| \|\nabla^2 c_i(y) - \nabla^2 c_i(x)\|_x^* \\ &\quad + \rho_k \sum_{i=1}^m (\|\nabla c_i(y)\nabla c_i(y)^T - \nabla c_i(x)\nabla c_i(x)^T\|_x^* + \|\tilde{c}_i(y)\nabla^2 c_i(y) - \tilde{c}_i(x)\nabla^2 c_i(x)\|_x^*). \end{aligned}$$

Statement (iii) then follows from this, (89) and (90).

(iv) Notice from (42) and $\nabla \tilde{c} = \nabla c$ that $\nabla_x \mathcal{L}(x, \lambda^k; \rho_k) = \nabla f(x) + \nabla c(x)\lambda^k + \rho_k \nabla c(x)\tilde{c}(x)$. Also, one can see from (31), (36) and $\|c(z_\epsilon)\| \leq 1$ that $\|\tilde{c}(x)\| \leq 2 + \delta_c$ for any $x \in \mathcal{S}(\delta_f, \delta_c)$. In addition, by (4), one can observe that $\|\nabla c_i(x)\nabla c_i(x)^T\|_x^* = (\|\nabla c_i(x)\|_x^*)^2$ for all i . In view of these and (91), we have

$$\begin{aligned}\|\nabla_x^2 \mathcal{L}(x, \lambda^k; \rho_k)\|_x^* &\leq \|\nabla f(x)\|_x^* + \sum_{i=1}^m |\lambda_i^k| \|\nabla c_i(x)\|_x^* + \rho_k \sum_{i=1}^m |\tilde{c}_i(x)| \|\nabla c_i(x)\|_x^*, \\ \|\nabla_{xx}^2 \mathcal{L}(x, \lambda^k; \rho_k)\|_x^* &\leq \|\nabla^2 f(x)\|_x^* + \sum_{i=1}^m |\lambda_i^k| \|\nabla^2 c_i(x)\|_x^* + \rho_k \sum_{i=1}^m ((\|\nabla c_i(x)\|_x^*)^2 + |\tilde{c}_i(x)| \|\nabla^2 c_i(x)\|_x^*),\end{aligned}$$

which, together with (33), (34) and the fact that $\|\tilde{c}(x)\| \leq 2 + \delta_c$ for any $x \in \mathcal{S}(\delta_f, \delta_c)$, implies that

$$\begin{aligned}U_{k,g} &= \sup_{x \in \mathcal{S}(\delta_f, \delta_c)} \|\nabla_x^2 \mathcal{L}(x, \lambda^k; \rho_k)\|_x^* \leq U_g^f + \|\lambda^k\|_1 U_g^c + \rho_k \sqrt{m}(2 + \delta_c) U_g^c, \\ U_{k,H} &= \sup_{x \in \mathcal{S}(\delta_f, \delta_c)} \|\nabla_{xx}^2 \mathcal{L}(x, \lambda^k; \rho_k)\|_x^* \leq U_H^f + \|\lambda^k\|_1 U_H^c + \rho_k (m(U_g^c)^2 + \sqrt{m}(2 + \delta_c) U_H^c).\end{aligned}$$

Hence, statement (iv) holds. \square

We now prove Theorem 5.2.

Proof of Theorem 5.2. Suppose that Algorithm 2 terminates at some iteration k , that is, $\mu_k \leq \epsilon/(2\vartheta^{1/2} + 2)$ and $\|c(x^{k+1})\| \leq \epsilon$ hold. By $\mu_k \leq \epsilon/(2\vartheta^{1/2} + 2)$, $\tilde{\lambda}^{k+1} = \lambda^k + \rho_k \tilde{c}(x^{k+1})$, $\nabla \tilde{c} = \nabla c$, and the second relation in (38), one has

$$\begin{aligned}\|\nabla f(x^{k+1}) + \nabla c(x^{k+1})\tilde{\lambda}^{k+1} + \mu_k \nabla B(x^{k+1})\|_{x^{k+1}}^* &= \|\nabla f(x^{k+1}) + \nabla \tilde{c}(x^{k+1})(\lambda^k + \rho_k \tilde{c}(x^{k+1})) + \mu_k \nabla B(x^{k+1})\|_{x^{k+1}}^* \\ &= \|\nabla_x \mathcal{L}_{\mu_k}(x^{k+1}, \lambda^k; \rho_k)\|_{x^{k+1}}^* \leq \mu_k.\end{aligned}\tag{92}$$

This along with $\mu_k > 0$ yields that

$$\|(\nabla f(x^{k+1}) + \nabla c(x^{k+1})\tilde{\lambda}^{k+1})/\mu_k + \nabla B(x^{k+1})\|_{x^{k+1}}^* \leq 1.$$

By this, $x^{k+1} \in \text{int } \mathcal{K}$ and Lemma 7.1(v), one has $(\nabla f(x^{k+1}) + \nabla c(x^{k+1})\tilde{\lambda}^{k+1})/\mu_k \in \mathcal{K}^*$, which implies that (9) holds for $(x^{k+1}, \tilde{\lambda}^{k+1})$. We next prove that (10) holds for $(x^{k+1}, \tilde{\lambda}^{k+1})$ with $\epsilon_1 = \epsilon$. Indeed, by (92), $\mu_k \leq \epsilon/(2\vartheta^{1/2} + 2)$, $x^{k+1} \in \text{int } \mathcal{K}$ and Lemma 7.1(i), one has

$$\begin{aligned}\|\nabla f(x^{k+1}) + \nabla c(x^{k+1})\tilde{\lambda}^{k+1}\|_{x^{k+1}}^* &\leq \|\nabla f(x^{k+1}) + \nabla c(x^{k+1})\tilde{\lambda}^{k+1} + \mu_k \nabla B(x^{k+1})\|_{x^{k+1}}^* + \mu_k \|\nabla B(x^{k+1})\|_{x^{k+1}}^* \\ &\leq \mu_k + \mu_k \vartheta^{1/2} = \epsilon/2 < \epsilon,\end{aligned}$$

and hence (10) holds for $(x^{k+1}, \tilde{\lambda}^{k+1})$ with $\epsilon_1 = \epsilon$. In view of these, $\|c(x^{k+1})\| \leq \epsilon$ and $x^{k+1} \in \text{int } \mathcal{K}$, we conclude that x^{k+1} is a deterministic ϵ -FOSP of problem (1).

In addition, recall from (39) that

$$\lambda_{\min}(M_{k+1}^T \nabla_{xx}^2 \mathcal{L}_{\mu}(x^{k+1}, \lambda^k; \rho_k) M_{k+1}) \geq -\sqrt{\mu_k}$$

holds with probability at least $1 - \delta$, which implies that $\hat{d}^T M_{k+1}^T \nabla_{xx}^2 \mathcal{L}_{\mu}(x^{k+1}, \lambda^k; \rho_k) M_{k+1} \hat{d} \geq -\sqrt{\mu_k} \|\hat{d}\|^2$ holds for all $\hat{d} \in \mathbb{R}^n$ with probability at least $1 - \delta$. Substituting $\hat{d} = M_{k+1}^{-1} \nabla^2 B(x^{k+1})^{-1/2} d$ in this inequality and using (16), $\tilde{\lambda}^{k+1} = \lambda^k + \rho_k \tilde{c}(x^{k+1})$, $\nabla \tilde{c} = \nabla c$ and $\nabla^2 \tilde{c}_i = \nabla^2 c_i$, $1 \leq i \leq m$, we obtain that with probability at least $1 - \delta$, it holds that

$$\begin{aligned}d^T \nabla^2 B(x^{k+1})^{-1/2} \left(\nabla^2 f(x^{k+1}) + \sum_{i=1}^m \tilde{\lambda}_i^{k+1} \nabla^2 c_i(x^{k+1}) + \rho_k \nabla c(x^{k+1}) \nabla c(x^{k+1})^T + \mu_k \nabla^2 B(x^{k+1}) \right) \nabla^2 B(x^{k+1})^{-1/2} d \\ \geq -\sqrt{\mu_k} \|M_{k+1}^{-1} \nabla^2 B(x^{k+1})^{-1/2} d\|^2 \stackrel{(16)}{=} -\sqrt{\mu_k} \|d\|^2, \quad \forall d \in \mathbb{R}^n,\end{aligned}$$

which together with $\mu_k \leq \epsilon/(2\vartheta^{1/2} + 2) \in (0, 1)$ and $\vartheta \geq 1$ implies that

$$d^T \nabla^2 B(x^{k+1})^{-1/2} \left(\nabla^2 f(x^{k+1}) + \sum_{i=1}^m \tilde{\lambda}_i^{k+1} \nabla^2 c_i(x^{k+1}) \right) \nabla^2 B(x^{k+1})^{-1/2} d \geq -(\sqrt{\mu_k} + \mu_k) \|d\|^2$$

$$\geq -2\sqrt{\mu_k}\|d\|^2 = -2\sqrt{\epsilon/(2\vartheta^{1/2} + 2)}\|d\|^2 \geq -\sqrt{\epsilon}\|d\|^2, \quad \forall d \in \mathcal{C}(x^{k+1}),$$

where $\mathcal{C}(\cdot)$ is defined in (12). Hence, with probability at least $1 - \delta$, the relation (11) holds for $(x^{k+1}, \tilde{\lambda}^{k+1})$ with $\epsilon_2 = \sqrt{\epsilon}$. In view of this and the fact that x^{k+1} is a deterministic ϵ -FOSP of (1), we conclude that the output x^{k+1} is an $(\epsilon, \sqrt{\epsilon})$ -SOSP of (1) with probability at least $1 - \delta$. \square

We next provide a proof for Theorem 5.3.

Proof of Theorem 5.3. Notice from (47) that $\rho_{\epsilon,1} \geq 2\rho_0$, which along with (45) and (46) implies that

$$K_\epsilon \stackrel{(46)}{=} \lceil \log \epsilon / \log \omega \rceil \stackrel{(45)}{=} \lceil \log 2 / \log r \rceil \leq \log(\rho_{\epsilon,1}\rho_0^{-1}) / \log r + 1. \quad (93)$$

Since $\{\rho_k\}$ is either unchanged or increased by a ratio r as k increases, it follows from (93) that

$$\max_{0 \leq k \leq K_\epsilon} \rho_k \leq r^{K_\epsilon} \rho_0 \stackrel{(93)}{\leq} r^{\frac{\log(\rho_{\epsilon,1}\rho_0^{-1})}{\log r} + 1} \rho_0 = r\rho_{\epsilon,1}. \quad (94)$$

In addition, observe from Algorithm 2 that $\rho_k > 2\gamma$ and $\|\lambda^k\| \leq \Lambda$. By these, (86), and Lemma 7.7(ii) with $(x, \lambda, \rho) = (x^{k+1}, \lambda^k, \rho_k)$, we obtain that

$$\|\tilde{c}(x^{k+1})\| \leq \sqrt{\frac{2(f_{\text{hi}} - f_{\text{low}} + \gamma)}{\rho_k - 2\gamma} + \frac{\|\lambda^k\|^2}{(\rho_k - 2\gamma)^2}} + \frac{\|\lambda^k\|}{\rho_k - 2\gamma} \leq \sqrt{\frac{2(f_{\text{hi}} - f_{\text{low}} + \gamma)}{\rho_k - 2\gamma} + \frac{\Lambda^2}{(\rho_k - 2\gamma)^2}} + \frac{\Lambda}{\rho_k - 2\gamma}. \quad (95)$$

On the other hand, notice from $\|c(z_\epsilon)\| \leq \epsilon/2$ and (36) that

$$\|c(x^{k+1})\| \leq \|\tilde{c}(x^{k+1})\| + \|c(z_\epsilon)\| \leq \|\tilde{c}(x^{k+1})\| + \epsilon/2. \quad (96)$$

We now prove that \bar{K}_ϵ is finite. Suppose for contradiction that \bar{K}_ϵ is infinite. By this and (48), one has that $\|c(x^{k+1})\| > \epsilon$ for all $k \geq K_\epsilon$. This along with (96) implies that $\|\tilde{c}(x^{k+1})\| > \epsilon/2$ for all $k \geq K_\epsilon$. It then follows that $\|\tilde{c}(x^{k+1})\| > \alpha\|\tilde{c}(x^k)\|$ must hold for infinitely many k 's, which, together with the update scheme on $\{\rho_k\}$, further implies $\rho_{k+1} = r\rho_k$ holds for infinitely many k 's. Using this and the monotonicity of $\{\rho_k\}$, we see that $\rho_k \rightarrow \infty$ as $k \rightarrow \infty$. This along with (95) yields that $\|\tilde{c}(x^{k+1})\| \rightarrow 0$ as $k \rightarrow \infty$, which leads to a contradiction with the fact that $\|\tilde{c}(x^{k+1})\| > \epsilon/2$ for all $k \geq K_\epsilon$. Hence, \bar{K}_ϵ is finite. In addition, notice from $\mu_k = \max\{\epsilon, \omega^k\}/(2\vartheta^{1/2} + 2)$ and (46) that $\mu_k = \epsilon/(2\vartheta^{1/2} + 2)$ for all $k \geq K_\epsilon$. Combining this with the termination criterion of Algorithm 2 and the definition of \bar{K}_ϵ , we conclude that Algorithm 2 with $\mu_k = \max\{\epsilon, \omega^k\}/(2\vartheta^{1/2} + 2)$ must terminate at iteration \bar{K}_ϵ .

We next prove (49) and that $\rho_k \leq r\rho_{\epsilon,1}$ holds for $0 \leq k \leq \bar{K}_\epsilon$ by considering two separate cases below.

Case 1) $\|c(x^{K_\epsilon+1})\| \leq \epsilon$. It then follows from (48) that $\bar{K}_\epsilon = K_\epsilon$, and thus (49) holds due to (93). In addition, by $\bar{K}_\epsilon = K_\epsilon$ and (94), one has that $\rho_k \leq r\rho_{\epsilon,1}$ holds for $0 \leq k \leq \bar{K}_\epsilon$ as well.

Case 2) $\|c(x^{K_\epsilon+1})\| > \epsilon$. It then follows from (48) that $\bar{K}_\epsilon > K_\epsilon$, and moreover, $\|c(x^{k+1})\| > \epsilon$ for all $K_\epsilon \leq k \leq \bar{K}_\epsilon - 1$. This along with (96) implies that

$$\|\tilde{c}(x^{k+1})\| > \epsilon/2, \quad \forall K_\epsilon \leq k \leq \bar{K}_\epsilon - 1. \quad (97)$$

By this, $\|\lambda^k\| \leq \Lambda$, (47), (86), and Lemma 7.7(iv) with $(x, \lambda, \rho, \tilde{\delta}_c) = (x^{k+1}, \lambda^k, \rho_k, \epsilon/2)$, one has

$$\begin{aligned} \rho_k &< 8(f_{\text{hi}} - f_{\text{low}} + \gamma)\epsilon^{-2} + 4\|\lambda^k\|\epsilon^{-1} + 2\gamma \\ &\leq 8(f_{\text{hi}} - f_{\text{low}} + \gamma)\epsilon^{-2} + 4\Lambda\epsilon^{-1} + 2\gamma \stackrel{(47)}{\leq} \rho_{\epsilon,1}, \quad \forall K_\epsilon \leq k \leq \bar{K}_\epsilon - 1. \end{aligned} \quad (98)$$

In view of this, (94), and the fact $\rho_{\bar{K}_\epsilon} \leq r\rho_{\bar{K}_\epsilon-1}$, we obtain that $\rho_k \leq r\rho_{\epsilon,1}$ holds for $0 \leq k \leq \bar{K}_\epsilon$. It remains to prove (49). To this end, let

$$\mathbb{K} = \{k : \rho_{k+1} = r\rho_k, K_\epsilon \leq k \leq \bar{K}_\epsilon - 2\}.$$

By (98) and the update scheme of ρ_k , one has $r^{|\mathbb{K}|}\rho_{K_\epsilon} = \max_{K_\epsilon \leq k \leq \bar{K}_\epsilon - 1} \rho_k \leq \rho_{\epsilon,1}$, which along with $\rho_{K_\epsilon} \geq \rho_0$ implies that

$$|\mathbb{K}| \leq \log(\rho_{\epsilon,1}\rho_0^{-1}) / \log r \leq \log(\rho_{\epsilon,1}\rho_0^{-1}) / \log r. \quad (99)$$

Let $\{k_1, k_2, \dots, k_{|\mathbb{K}|}\}$ denote all the elements of \mathbb{K} arranged in ascending order, and let $k_0 = K_\epsilon$ and $k_{|\mathbb{K}|+1} = \overline{K}_\epsilon - 1$. We next derive an upper bound for $k_{j+1} - k_j$ for $j = 0, 1, \dots, |\mathbb{K}|$. Using the definition of \mathbb{K} , we see that $\rho_k = \rho_{k'}$ for $k_j < k, k' \leq k_{j+1}$. By this and the update scheme of ρ_k , one can see that

$$\|\tilde{c}(x^{k+1})\| \leq \alpha \|\tilde{c}(x^k)\|, \quad \forall k_j < k < k_{j+1}. \quad (100)$$

In addition, by (43), (95) and $\rho_k \geq \rho_0$, one has $\|\tilde{c}(x^{k+1})\| \leq \delta_{c,1}$ for $0 \leq k \leq \overline{K}_\epsilon$. Using this and (97), we obtain that

$$\epsilon/2 < \|\tilde{c}(x^{k+1})\| \leq \delta_{c,1}, \quad \forall K_\epsilon \leq k \leq \overline{K}_\epsilon - 1. \quad (101)$$

Now, we notice that either $k_{j+1} - k_j = 1$ or $k_{j+1} - k_j > 1$. In the latter case, one can apply (100) with $k = k_{j+1} - 1, \dots, k_j + 1$ along with (101) to deduce that

$$\epsilon/2 < \|\tilde{c}(x^{k_{j+1}})\| \leq \alpha \|\tilde{c}(x^{k_{j+1}-1})\| \leq \dots \leq \alpha^{k_{j+1}-k_j-1} \|\tilde{c}(x^{k_j})\| \leq \alpha^{k_{j+1}-k_j-1} \delta_{c,1}, \quad \forall j = 0, 1, \dots, |\mathbb{K}|.$$

Combining the two cases, we deduce that

$$k_{j+1} - k_j \leq \lfloor \log(\epsilon(2\delta_{c,1})^{-1}) / \log \alpha \rfloor + 1, \quad \forall j = 0, 1, \dots, |\mathbb{K}|. \quad (102)$$

Summing up these inequalities, and using (93), (99), $k_0 = K_\epsilon$ and $k_{|\mathbb{K}|+1} = \overline{K}_\epsilon - 1$, we have

$$\begin{aligned} \overline{K}_\epsilon &= 1 + k_{|\mathbb{K}|+1} = 1 + k_0 + \sum_{j=0}^{|\mathbb{K}|} (k_{j+1} - k_j) \stackrel{(102)}{\leq} 1 + K_\epsilon + (|\mathbb{K}| + 1) \left(\left\lfloor \frac{\log(\epsilon(2\delta_{c,1})^{-1})}{\log \alpha} \right\rfloor + 1 \right) \\ &\leq 2 + \frac{\log(\rho_{\epsilon,1}\rho_0^{-1})}{\log r} + \left(\frac{\log(\rho_{\epsilon,1}\rho_0^{-1})}{\log r} + 1 \right) \left(\left\lfloor \frac{\log(\epsilon(2\delta_{c,1})^{-1})}{\log \alpha} \right\rfloor + 1 \right) = 1 + \left(\frac{\log(\rho_{\epsilon,1}\rho_0^{-1})}{\log r} + 1 \right) \left(\left\lfloor \frac{\log(\epsilon(2\delta_{c,1})^{-1})}{\log \alpha} \right\rfloor + 2 \right), \end{aligned}$$

where the second inequality is due to (93) and (99). Hence, (49) holds as well in this case. \square

We next prove Theorem 5.4. Before proceeding, we recall from Lemma 5.1 and the discussions in Section 5.1 that the subproblem $\min_x \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k)$ satisfies Assumptions 4.1(b) and 4.1(c) with $(F(\cdot), \mathcal{S}, \Omega, L_H^F, U_g^F, U_H^F) = (\mathcal{L}(\cdot, \lambda^k; \rho_k), \mathcal{S}(\delta_f, \delta_c), \Omega(\delta_f, \delta_c), L_{k,H}, U_{k,g}, U_{k,H})$. Moreover, in view of the fact that $\|\lambda^k\| \leq \Lambda$, one can see from (44) and Lemma 5.1(iv) that there exist some constants L_1, L_2, U_1 and U_2 , depending only on $f, c, B, \beta, \Lambda, m, \delta_f$ and δ_c , such that

$$L_{k,H} \leq L_1 + \rho_k L_2, \quad U_{k,H} \leq U_1 + \rho_k U_2. \quad (103)$$

We are now ready to prove Theorem 5.4.

Proof of Theorem 5.4. Let T_k and N_k denote the number of iterations and fundamental operations performed by Algorithm 1 at outer iteration k of Algorithm 2, respectively. It then follows from Theorem 5.3 that the total number of iterations of Algorithm 1 performed in Algorithm 2 is $\sum_{k=0}^{\overline{K}_\epsilon} T_k$, and moreover, the total number of Cholesky factorizations and other fundamental operations performed by Algorithm 1 in Algorithm 2 are $\sum_{k=0}^{\overline{K}_\epsilon} T_k$ and $\sum_{k=0}^{\overline{K}_\epsilon} N_k$, respectively. In addition, notice from (47) and Theorem 5.3 that $\rho_{\epsilon,1} = \mathcal{O}(\epsilon^{-2})$ and $\rho_k \leq r\rho_{\epsilon,1}$, which yield $\rho_k = \mathcal{O}(\epsilon^{-2})$ for all $0 \leq k \leq \overline{K}_\epsilon$.

(i) Recall from Lemmas 5.1(i) and 5.1(iii) that $L_{k,H}$ is a Lipschitz constant of $\nabla_{xx}^2 \mathcal{L}(x, \lambda^k; \rho_k)$ with respect to the local norm on an open convex neighborhood of $\{x \in \text{int } \mathcal{K} : \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k) \leq \mathcal{L}_{\mu_k}(x_{\text{init}}, \lambda^k; \rho_k)\}$. In addition, recall from Lemma 5.1(ii) that $\inf_{x \in \text{int } \mathcal{K}} \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k) \geq f_{\text{low}} - \gamma - \Lambda \delta_c$. By these, (41), (103), and Theorem 4.1(iii) with $(\phi_{\text{hi}}, \phi_{\text{low}}, L_H^F, \epsilon_g, \epsilon_H) = (\mathcal{L}_{\mu_k}(x_{\text{init}}, \lambda^k; \rho_k), f_{\text{low}} - \gamma - \Lambda \delta_c, L_{k,H}, \mu_k, \sqrt{\mu_k})$, one has

$$T_k = \mathcal{O}((f_{\text{hi}} - f_{\text{low}} + \gamma + \Lambda \delta_c) L_{k,H}^2 \mu_k^{-3/2}) \stackrel{(103)}{=} \mathcal{O}(\rho_k^2 \mu_k^{-3/2}) = \mathcal{O}(\epsilon^{-11/2}), \quad (104)$$

where the last equality follows from $\mu_k \geq \epsilon/(2\vartheta^{1/2} + 2)$ and $\rho_k = \mathcal{O}(\epsilon^{-2})$. On the other hand, if c is assumed to be affine, namely, $c(x) = Ax - b$ for some $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, then $\nabla c(x) = A^T$ and $\nabla^2 c_i(x) = 0$ for $1 \leq i \leq m$. Using these and (91), we observe that $L_{k,H} = \mathcal{O}(1)$. By this and the similar arguments as for (104), one has $T_k = \mathcal{O}(\mu_k^{-3/2}) = \mathcal{O}(\epsilon^{-3/2})$. Combining these with (104) and $\overline{K}_\epsilon = \mathcal{O}(|\log \epsilon|^2)$ (see Remark 5.4), we conclude that statement (i) holds.

(ii) By Lemmas 5.1(i) and 5.1(iv), one has

$$U_{k,H} \geq \sup_{x \in \text{int } \mathcal{K}} \{ \|\nabla_{xx}^2 \mathcal{L}(x, \lambda^k; \rho_k)\|_x^* : \mathcal{L}_{\mu_k}(x, \lambda^k; \rho_k) \leq \mathcal{L}_{\mu_k}(x_{\text{init}}^k, \lambda^k; \rho_k) \}.$$

In view of this, $\mathcal{L}_{\mu_k}(x_{\text{init}}^k, \lambda^k; \rho_k) \leq f_{\text{hi}}$, (103), and Theorem 4.1(iv) with $(\phi_{\text{hi}}, \phi_{\text{low}}, L_H^F, U_H^F, \epsilon_g, \epsilon_H) = (\mathcal{L}_{\mu_k}(x_{\text{init}}^k, \lambda^k; \rho_k), f_{\text{low}} - \gamma - \Lambda \delta_c, L_{k,H}, U_{k,H}, \mu_k, \sqrt{\mu_k})$, we obtain that

$$\begin{aligned} N_k &= \tilde{\mathcal{O}}((f_{\text{hi}} - f_{\text{low}} + \gamma + \Lambda \delta_c) L_{k,H}^2 \mu_k^{-3/2} \min\{n, U_{k,H}^{1/2} \mu_k^{-1/4}\}) \\ (103) \quad &= \tilde{\mathcal{O}}(\rho_k^2 \mu_k^{-3/2} \min\{n, \rho_k^{1/2} \mu_k^{-1/4}\}) = \tilde{\mathcal{O}}(\epsilon^{-11/2} \min\{n, \epsilon^{-5/4}\}), \end{aligned} \quad (105)$$

where the last equality follows from $\mu_k \geq \epsilon/(2\vartheta + 2)$ and $\rho_k = \mathcal{O}(\epsilon^{-2})$. On the other hand, if c is assumed to be affine, it follows from the above discussion that $L_{k,H} = \mathcal{O}(1)$. By this, $U_{k,H} \leq U_1 + \rho_k U_2$, and the similar arguments as for (105), one has $N_k = \tilde{\mathcal{O}}(\mu_k^{-3/2} \min\{n, \rho_k^{1/2} \mu_k^{-1/4}\}) = \tilde{\mathcal{O}}(\epsilon^{-3/2} \min\{n, \epsilon^{-5/4}\})$. Combining these with (105) and $\bar{K}_\epsilon = \mathcal{O}(|\log \epsilon|^2)$ (see Remark 5.4), we conclude that statement (ii) holds. \square

We next establish two technical lemmas that will be used to prove Theorem 5.5.

Lemma 7.9. *Suppose that Assumptions 5.1 and 5.2 hold and that ρ_0 is sufficiently large such that $\delta_{f,1} \leq \delta_f$ and $\delta_{c,1} \leq \delta_c$, where $\delta_{f,1}$ and $\delta_{c,1}$ are defined in (43). Let $\{(x^k, \lambda^k, \rho_k)\}$ be generated by Algorithm 2. Suppose that*

$$\rho_k \geq \max\{\Lambda^2(2\delta_f)^{-1}, 2(f_{\text{hi}} - f_{\text{low}} + \gamma)\delta_c^{-2} + 2\Lambda\delta_c^{-1} + 2\gamma, 2(U_g^f + \sqrt{m}U_g^c\Lambda + \sqrt{\vartheta} + 1)(\sigma\epsilon)^{-1}\} \quad (106)$$

for some $k \geq 0$, where $\gamma, f_{\text{hi}}, f_{\text{low}}, \delta_f, \delta_c, U_g^f$ and U_g^c are given in Assumption 5.1, and σ is given in (50). Then it holds that $\|c(x^{k+1})\| \leq \epsilon$.

Proof. Using $\|\lambda^k\| \leq \Lambda$ (see step 5 of Algorithm 2) and (106), we have

$$\rho_k \geq \max\{\|\lambda^k\|^2(2\delta_f)^{-1}, 2(f_{\text{hi}} - f_{\text{low}} + \gamma)\delta_c^{-2} + 2\|\lambda^k\|\delta_c^{-1} + 2\gamma\}.$$

By this, (86), and Lemmas 7.7(iii) and 7.7(iv) with $(x, \lambda, \mu, \rho, \tilde{\delta}_f, \tilde{\delta}_c) = (x^{k+1}, \lambda^k, \mu_k, \rho_k, \delta_f, \delta_c)$, one has $f(x^{k+1}) + \mu_k B(x^{k+1}) \leq f_{\text{hi}} + \delta_f$ and $\|\tilde{c}(x^{k+1})\| \leq \delta_c$. Also, notice from $\|c(z_\epsilon)\| \leq 1$ and (36) that $\|c(x^{k+1})\| \leq 1 + \|\tilde{c}(x^{k+1})\|$. These along with (31), $x^{k+1} \in \text{int } \mathcal{K}$, and $\mu_k \in (0, \mu_0]$ yield that $x^{k+1} \in \mathcal{S}(\delta_f, \delta_c)$. It then follows from (33) that $\|\nabla f(x^{k+1})\|_{x^{k+1}}^* \leq U_g^f$ and $\|\nabla c_i(x^{k+1})\|_{x^{k+1}}^* \leq U_g^c$ for all $1 \leq i \leq m$. By these, $\mu_k \leq 1$, $\|\lambda^k\| \leq \Lambda$, (36) and the second relation in (38), one has

$$\begin{aligned} \rho_k \|\nabla^2 B(x^{k+1})^{-1/2} \nabla c(x^{k+1}) \tilde{c}(x^{k+1})\| &= \rho_k \|\nabla c(x^{k+1}) \tilde{c}(x^{k+1})\|_{x^{k+1}}^* \\ &\leq \|\nabla f(x^{k+1}) + \nabla c(x^{k+1}) \lambda^k\|_{x^{k+1}}^* + \mu_k \|\nabla B(x^{k+1})\|_{x^{k+1}}^* + \|\nabla_x \mathcal{L}_{\mu_k}(x^{k+1}, \lambda^k; \rho_k)\|_{x^{k+1}}^* \\ &\leq \|\nabla f(x^{k+1})\|_{x^{k+1}}^* + \sum_{i=1}^m |\lambda_i^k| \|\nabla c_i(x^{k+1})\|_{x^{k+1}}^* + \mu_k \sqrt{\vartheta} + \mu_k \leq U_g^f + \sqrt{m}U_g^c\Lambda + \sqrt{\vartheta} + 1, \end{aligned} \quad (107)$$

where the first inequality follows from the triangle inequality, and the second inequality follows from $\|\nabla B(x^{k+1})\|_{x^{k+1}}^* = \sqrt{\vartheta}$ and the second relation in (38). In addition, by $x^{k+1} \in \mathcal{S}(\delta_f, \delta_c)$ and (50), one has $\lambda_{\min}(\nabla c(x^{k+1})^T \nabla^2 B(x^{k+1})^{-1} \nabla c(x^{k+1})) \geq \sigma^2$, which along with (107) implies that

$$\begin{aligned} \|\tilde{c}(x^{k+1})\| &\leq \|[\nabla c(x^{k+1})^T \nabla^2 B(x^{k+1})^{-1} \nabla c(x^{k+1})]^{-1} \nabla c(x^{k+1})^T \nabla^2 B(x^{k+1})^{-1/2}\| \|\nabla^2 B(x^{k+1})^{-1/2} \nabla c(x^{k+1}) \tilde{c}(x^{k+1})\| \\ &= \|[\nabla c(x^{k+1})^T \nabla^2 B(x^{k+1})^{-1} \nabla c(x^{k+1})]^{-1}\|^{1/2} \|\nabla^2 B(x^{k+1})^{-1/2} \nabla c(x^{k+1}) \tilde{c}(x^{k+1})\| \\ &= \lambda_{\min}(\nabla c(x^{k+1})^T \nabla^2 B(x^{k+1})^{-1} \nabla c(x^{k+1}))^{-1/2} \|\nabla^2 B(x^{k+1})^{-1/2} \nabla c(x^{k+1}) \tilde{c}(x^{k+1})\| \\ &\leq (U_g^f + \sqrt{m}U_g^c\Lambda + \sqrt{\vartheta} + 1)/(\sigma\rho_k). \end{aligned} \quad (108)$$

Observe from (106) that $\rho_k \geq 2(U_g^f + \sqrt{m}U_g^c\Lambda + \sqrt{\vartheta} + 1)(\sigma\epsilon)^{-1}$, which along with (108) implies $\|\tilde{c}(x^{k+1})\| \leq \epsilon/2$. Combining this with $\|c(z_\epsilon)\| \leq \epsilon/2$ and (36), we obtain $\|c(x^{k+1})\| \leq \epsilon$ as desired. \square

The next lemma establishes a stronger upper bound for $\{\rho_k\}$ than the one given in Theorem 5.3.

Lemma 7.10. Suppose that Assumptions 5.1 and 5.2 hold and that ρ_0 is sufficiently large such that $\delta_{f,1} \leq \delta_f$ and $\delta_{c,1} \leq \delta_c$, where $\delta_{f,1}$ and $\delta_{c,1}$ are defined in (43). Let $\{\rho_k\}$ be generated by Algorithm 2 and

$$\rho_{\epsilon,2} := \max\{\Lambda^2(2\delta_f)^{-1}, 2(f_{\text{hi}} - f_{\text{low}} + \gamma)\delta_c^{-2} + 2\Lambda\delta_c^{-1} + 2\gamma, 2(U_g^f + \sqrt{m}U_g^c\Lambda + \sqrt{\vartheta} + 1)(\sigma\epsilon)^{-1}, 2\rho_0\}, \quad (109)$$

where γ , f_{hi} , f_{low} , δ_f , δ_c , U_g^f and U_g^c are given in Assumption 5.1, and σ is given in (50). Then $\rho_k \leq r\rho_{\epsilon,2}$ holds for $0 \leq k \leq \bar{K}_\epsilon$, where \bar{K}_ϵ is defined in (48).

Proof. Observe from (109) that $\rho_{\epsilon,2} \geq 2\rho_0$. Using this and similar arguments as for (93), we have $K_\epsilon \leq \log(\rho_{\epsilon,2}\rho_0^{-1})/\log r + 1$, where K_ϵ is defined in (46). By this, the update scheme for $\{\rho_k\}$, and similar arguments as for (94), one has

$$\max_{0 \leq k \leq K_\epsilon} \rho_k \leq r\rho_{\epsilon,2}. \quad (110)$$

If $\|c(x^{K_\epsilon+1})\| \leq \epsilon$, it follows from (48) that $\bar{K}_\epsilon = K_\epsilon$, which along with (110) implies that $\rho_k \leq r\rho_{\epsilon,2}$ holds for $0 \leq k \leq \bar{K}_\epsilon$. On the other hand, if $\|c(x^{K_\epsilon+1})\| > \epsilon$, it follows from (48) that $\|c(x^{k+1})\| > \epsilon$ for $K_\epsilon \leq k \leq \bar{K}_\epsilon - 1$, which together with Lemma 7.9 and (109) implies that

$$\rho_k < \max\left\{\frac{\Lambda^2}{2\delta_f}, \frac{2(f_{\text{hi}} - f_{\text{low}} + \gamma)}{\delta_c^2} + \frac{2\Lambda}{\delta_c} + 2\gamma, \frac{2(U_g^f + \sqrt{m}U_g^c\Lambda + \sqrt{\vartheta} + 1)}{\sigma\epsilon}\right\} \stackrel{(109)}{\leq} \rho_{\epsilon,2}, \quad \forall K_\epsilon \leq k \leq \bar{K}_\epsilon - 1.$$

Using this, (110), and $\rho_{\bar{K}_\epsilon} \leq r\rho_{\bar{K}_\epsilon-1}$, we also conclude that $\rho_k \leq r\rho_{\epsilon,2}$ holds for $0 \leq k \leq \bar{K}_\epsilon$. \square

We now provide a proof for Theorem 5.5.

Proof of Theorem 5.5. Notice from (109) and Lemma 7.10 that $\rho_{\epsilon,2} = \mathcal{O}(\epsilon^{-1})$ and $\rho_k \leq r\rho_{\epsilon,2}$, which imply $\rho_k = \mathcal{O}(\epsilon^{-1})$. The rest of the proof follows from the same arguments as for Theorem 5.4 with $\rho_k = \mathcal{O}(\epsilon^{-2})$ replaced by $\rho_k = \mathcal{O}(\epsilon^{-1})$. \square

8 Concluding remarks

In this paper we proposed a Newton-CG based barrier-AL method for finding an approximate SOSOP of general nonconvex conic optimization problem (1). We also established the worst-case iteration and operation complexity bounds of the proposed method for finding an approximate SOSOP of problem (1). In addition, we conducted preliminary numerical experiments to demonstrate the superior solution quality of our method over a well-known first-order method, SpaRSA.

There are several potential directions for future research. Firstly, conducting extensive numerical studies could provide new insights into improving the practical performance of our method. Secondly, it would be interesting to extend our method to solve a more general conic optimization problem, $\min_{x,y}\{\tilde{f}(x,y) : \tilde{c}(x,y) = 0, y \in \mathcal{K}\}$, which includes the problem $\min_x\{f(x) : c(x) = 0, d(x) \leq 0\}$ and, more generally, the problem $\min_x\{f(x) : c(x) = 0, d(x) \in \mathcal{K}\}$ as special cases. Notice the latter problem can be equivalently solved as $\min_{x,y}\{f(x) : c(x) = 0, d(x) - y = 0, y \in \mathcal{K}\}$, which is a specific instance of the problem considered in this paper. Consequently, it can be suitably solved by our proposed method. Lastly, extending our approach to finding an approximate SOSOP for nonconvex optimization problems with a general convex set constraint, beyond the convex conic constraint, remains an open question.

Data availability: The codes for generating the random data and implementing the algorithms in the numerical section are available from the first author upon request.

Competing interests: The third author is an editorial board member of this journal.

References

- [1] N. Agarwal, Z. Allen-Zhu, B. Bullins, E. Hazan, and T. Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1195–1199, 2017.
- [2] Z. Allen-Zhu and Y. Li. Neon2: Finding local minima via first-order oracles. *Advances in Neural Information Processing Systems*, 31:3716–3726, 2018.
- [3] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. On augmented Lagrangian methods with general lower-level constraints. *SIAM J. Optim.*, 18(4):1286–1309, 2008.
- [4] R. Andreani, G. Haeser, A. Ramos, and P. J. Silva. A second-order sequential optimality condition associated to the convergence of optimization algorithms. *IMA J. Numer. Anal.*, 37(4):1902–1929, 2017.
- [5] M. Argáez and R. Tapia. On the global convergence of a modified augmented Lagrangian linesearch interior-point Newton method for nonlinear programming. *J. Optim. Theory Appl.*, 114(1):1–25, 2002.
- [6] P. Armand and R. Omheni. A mixed logarithmic barrier-augmented Lagrangian method for nonlinear optimization. *J. Optim. Theory Appl.*, 173(2):523–547, 2017.
- [7] P. Armand and N. N. Tran. Rapid infeasibility detection in a mixed logarithmic barrier-augmented Lagrangian method for nonlinear optimization. *Optim. Methods Softw.*, 34(5):991–1013, 2019.
- [8] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.
- [9] S. Bhojanapalli, B. Neyshabur, and N. Srebro. Global optimality of local search for low rank matrix recovery. *Advances in Neural Information Processing Systems*, 29:3873–3881, 2016.
- [10] W. Bian, X. Chen, and Y. Ye. Complexity analysis of interior point algorithms for non-Lipschitz and nonconvex minimization. *Math. Program.*, 149(1):301–327, 2015.
- [11] E. G. Birgin, J. Gardenghi, J. M. Martínez, S. A. Santos, and P. L. Toint. Evaluation complexity for nonlinear constrained optimization using unscaled KKT conditions and high-order models. *SIAM J. Optim.*, 26(2):951–967, 2016.
- [12] E. G. Birgin, G. Haeser, and A. Ramos. Augmented Lagrangians with constrained subproblems and convergence to second-order stationary points. *Comput. Optim. and Appl.*, 69(1):51–75, 2018.
- [13] E. G. Birgin and J. M. Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. SIAM, 2014.
- [14] E. G. Birgin and J. M. Martínez. The use of quadratic regularization with a cubic descent condition for unconstrained optimization. *SIAM J. Optim.*, 27(2):1049–1074, 2017.
- [15] J. F. Bonnans and G. Launay. Sequential quadratic programming with penalization of the displacement. *SIAM J. Optim.*, 5(4):792–812, 1995.
- [16] L. F. Bueno and J. M. Martínez. On the complexity of an inexact restoration method for constrained optimization. *SIAM J. Optim.*, 30(1):80–101, 2020.
- [17] R. H. Byrd, R. B. Schnabel, and G. A. Shultz. A trust region algorithm for nonlinearly constrained optimization. *SIAM J. Numer. Anal.*, 24(5):1152–1170, 1987.
- [18] Y. Carmon and J. Duchi. Gradient descent finds the cubic-regularized nonconvex newton step. *SIAM J. Optim.*, 29(3):2146–2178, 2019.
- [19] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. “Convex until proven guilty”: Dimension-free acceleration of gradient descent on non-convex functions. In *International Conference on Machine Learning*, pages 654–663. PMLR, 2017.

[20] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Accelerated methods for nonconvex optimization. *SIAM J. Optim.*, 28(2):1751–1772, 2018.

[21] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Math. Program.*, 127(2):245–295, 2011.

[22] C. Cartis, N. I. Gould, and P. L. Toint. On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization. *SIAM J. Optim.*, 23(3):1553–1574, 2013.

[23] C. Cartis, N. I. Gould, and P. L. Toint. On the complexity of finding first-order critical points in constrained nonlinear optimization. *Math. Program.*, 144(1):93–106, 2014.

[24] C. Cartis, N. I. Gould, and P. L. Toint. On the complexity of finding first-order critical points in constrained nonlinear optimization. *Math. Program.*, 144(1):93–106, 2014.

[25] C. Cartis, N. I. Gould, and P. L. Toint. On the evaluation complexity of constrained nonlinear least-squares and general constrained nonlinear optimization using second-order methods. *SIAM J. Numer. Anal.*, 53(2):836–851, 2015.

[26] C. Cartis, N. I. Gould, and P. L. Toint. Evaluation complexity bounds for smooth constrained nonlinear optimization using scaled KKT conditions, high-order models and the criticality measure χ . In *Approximation and Optimization: Algorithms, Complexity and Applications*, pages 5–26. Springer, 2019.

[27] C. Cartis, N. I. Gould, and P. L. Toint. Optimality of orders one to three and beyond: characterization and evaluation complexity in constrained nonconvex optimization. *J. Complex.*, 53:68–94, 2019.

[28] X. Chen, L. Guo, Z. Lu, and J. J. Ye. An augmented Lagrangian method for non-Lipschitz nonconvex programming. *SIAM J. Numer. Anal.*, 55(1):168–193, 2017.

[29] Y. Chi, Y. M. Lu, and Y. Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Trans. Signal Process.*, 67(20):5239–5269, 2019.

[30] D. Cifuentes and A. Moitra. Polynomial time guarantees for the Burer-Monteiro method. *arXiv preprint arXiv:1912.01745*, 2019.

[31] T. F. Coleman, J. Liu, and W. Yuan. A new trust-region algorithm for equality constrained optimization. *Comput. Optim. Appl.*, 21(2):177–199, 2002.

[32] A. R. Conn, G. Gould, and P. L. Toint. *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*, volume 17. Springer Science & Business Media, 2013.

[33] A. R. Conn, N. I. Gould, and P. L. Toint. A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. *Math. Comput.*, 66(217):261–288, 1997.

[34] F. E. Curtis, D. P. Robinson, C. W. Royer, and S. J. Wright. Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization. *SIAM J. Optim.*, 31(1):518–544, 2021.

[35] F. E. Curtis, D. P. Robinson, and M. Samadi. A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. *Math. Program.*, 1(162):1–32, 2016.

[36] F. E. Curtis, D. P. Robinson, and M. Samadi. Complexity analysis of a trust funnel algorithm for equality constrained optimization. *SIAM J. Optim.*, 28(2):1533–1563, 2018.

[37] E. P. De Carvalho, A. dos Santos Júnior, and T. F. Ma. Reduced gradient method combined with augmented Lagrangian and barrier for the optimal power flow problem. *Appl. Math. Comput.*, 200(2):529–536, 2008.

[38] P. Dvurechensky and M. Staudigl. Hessian barrier algorithms for non-convex conic optimization. *arXiv preprint arXiv:2111.00100*, 2021.

[39] D. Goldfarb, R. Polyak, K. Scheinberg, and I. Yuzefovich. A modified barrier-augmented Lagrangian method for constrained minimization. *Comput. Optim. Appl.*, 14(1):55–74, 1999.

[40] F. Goyens, A. Eftekhari, and N. Boumal. Computing second-order points under equality constraints: revisiting Fletcher’s augmented Lagrangian. *arXiv preprint arXiv:2204.01448*, 2022.

[41] G. N. Grapiglia and Y.-x. Yuan. On the complexity of an augmented Lagrangian method for nonconvex optimization. *IMA J. Numer. Anal.*, 41(2):1546–1568, 2021.

[42] G. Haeser, H. Liu, and Y. Ye. Optimality condition and complexity analysis for linearly-constrained optimization without differentiability on the boundary. *Math. Program.*, 178(1):263–299, 2019.

[43] C. He and Z. Lu. A Newton-CG based barrier method for finding a second-order stationary point of nonconvex conic optimization with complexity guarantees. *SIAM J. Optim.*, 33(2):1191–1222, 2023.

[44] C. He, Z. Lu, and T. K. Pong. A Newton-CG based augmented Lagrangian method for finding a second-order stationary point of nonconvex equality constrained optimization with complexity guarantees. *SIAM J. Optim.*, 33(3):1734–1766, 2023.

[45] A. Huck, M. Guillaume, and J. Blanc-Talon. Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data. *IEEE Trans. Geosci. Remote Sens.*, 48(6):2590–2602, 2010.

[46] C. Jin, P. Netrapalli, and M. I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference On Learning Theory*, pages 1042–1085. PMLR, 2018.

[47] C. Kanzow and D. Steck. An example comparing the standard and safeguarded augmented Lagrangian methods. *Oper. Res. Lett.*, 45(6):598–603, 2017.

[48] J. Kuczyński and H. Woźniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM J. Matrix Anal. Appl.*, 13(4):1094–1122, 1992.

[49] R. Kuhlmann and C. Büskens. A primal–dual augmented Lagrangian penalty-interior-point filter line search algorithm. *Math. Method Oper. Res.*, 87(3):451–483, 2018.

[50] X. Liu, W. Xia, B. Wang, and L. Zhang. An approach based on constrained nonnegative matrix factorization to unmix hyperspectral data. *IEEE Trans. Geosci. Remote Sens.*, 49(2):757–772, 2010.

[51] S. Lu, M. Razavyayn, B. Yang, K. Huang, and M. Hong. Finding second-order stationary points efficiently in smooth nonconvex linearly constrained optimization problems. *Advances in Neural Information Processing Systems*, 33:2811–2822, 2020.

[52] Z. Lu and Y. Zhang. An augmented Lagrangian approach for sparse principal component analysis. *Math. Program.*, 135(1):149–193, 2012.

[53] J. M. Martínez and M. Raydan. Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. *J. Glob. Optim.*, 68(2):367–385, 2017.

[54] L. Miao and H. Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Trans. Geosci. Remote Sens.*, 45(3):765–777, 2007.

[55] J. M. Moguerza and F. J. Prieto. An augmented Lagrangian interior-point method using directions of negative curvature. *Math. Program.*, 95(3):573–616, 2003.

[56] Y. Nesterov and A. Nemirovskii. *Interior-point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.

[57] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Math. Program.*, 108(1):177–205, 2006.

[58] M. O’Neill and S. J. Wright. A log-barrier Newton-CG method for bound constrained optimization with complexity guarantees. *IMA J. Numer. Anal.*, 41(1):84–121, 2021.

- [59] D. Park, A. Kyriolidis, C. Carmanis, and S. Sanghavi. Non-square matrix sensing without spurious local minima via the burer-monteiro approach. In *Artificial Intelligence and Statistics*, pages 65–74. PMLR, 2017.
- [60] C. W. Royer, M. O’Neill, and S. J. Wright. A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization. *Math. Program.*, 180(1):451–488, 2020.
- [61] C. W. Royer and S. J. Wright. Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM J. Optim.*, 28(2):1448–1477, 2018.
- [62] A. Ruszczynski. *Nonlinear Optimization*. Princeton university press, 2011.
- [63] M. F. Sahin, A. Eftekhari, A. Alacaoglu, F. Latorre, and V. Cevher. An inexact augmented Lagrangian framework for nonconvex optimization with nonlinear constraints. *Advances in Neural Information Processing Systems*, 32:632–650, 2019.
- [64] O. V. Thanh, N. Gillis, and F. Lecron. Bounded simplex-structured matrix factorization. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9062–9066. IEEE, 2022.
- [65] R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Comput. Optim. Appl.*, 13(1):231–252, 1999.
- [66] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1):25–57, 2006.
- [67] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Math. Program.*, 107(3):391–408, 2006.
- [68] S. J. Wright, R. D. Nowak, and M. A. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.*, 57(7):2479–2493, 2009.
- [69] Y. Xie and S. J. Wright. Complexity of projected Newton methods for bound-constrained optimization. *arXiv preprint arXiv:2103.15989*, 2021.
- [70] Y. Xie and S. J. Wright. Complexity of proximal augmented Lagrangian for nonconvex optimization with nonlinear equality constraints. *J. Sci. Comput.*, 86(3):1–30, 2021.
- [71] Y. Xu, R. Jin, and T. Yang. NEON+: Accelerated gradient methods for extracting negative curvature for non-convex optimization. *arXiv preprint arXiv:1712.01033*, 2017.
- [72] L. Yang, D. Sun, and K. C. Toh. SDPNAL+: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Math. Program. Comput.*, 7(3):331–366, 2015.
- [73] X. Zhao, D. Sun, and K. C. Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM J. Optim.*, 20(4):1737–1765, 2010.

Appendix

A A capped conjugate gradient method

In this part we present the capped CG method proposed in [60, Algorithm 1] for solving a possibly indefinite linear system (15). As briefly discussed in Section 4, the capped CG method finds either an approximate solution to (15) or a sufficiently negative curvature direction of the associated matrix H . More details about this method can be found in [60, Section 3.1].

The following theorem presents the iteration complexity of Algorithm 3, whose proof can be found in [44, Theorem A.1], and thus omitted here.

Algorithm 3 A capped conjugate gradient method

Input: symmetric matrix $H \in \mathbb{R}^{n \times n}$, vector $g \neq 0$, damping parameter $\varepsilon \in (0, 1)$, desired relative accuracy $\zeta \in (0, 1)$.

Optional input: scalar $U \geq 0$ such that $\|H\| \leq U$ (set to 0 if not provided).

Outputs: \hat{d} , d_type.

Secondary outputs: final values of $U, \kappa, \hat{\zeta}, \tau$, and T .

Set

$$\bar{H} := H + 2\varepsilon I, \quad \kappa := \frac{U + 2\varepsilon}{\varepsilon}, \quad \hat{\zeta} := \frac{\zeta}{3\kappa}, \quad \tau := \frac{\sqrt{\kappa}}{\sqrt{\kappa} + 1}, \quad T := \frac{4\kappa^4}{(1 - \sqrt{\tau})^2},$$

$$y^0 \leftarrow 0, r^0 \leftarrow g, p^0 \leftarrow -g, j \leftarrow 0.$$

if $(p^0)^T \bar{H} p^0 < \varepsilon \|p^0\|^2$ **then**

 Set $\hat{d} \leftarrow p^0$ and terminate with d_type = NC;

else if $\|H p^0\| > U \|p^0\|$ **then**

 Set $U \leftarrow \|H p^0\| / \|p^0\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;

end if

while TRUE **do**

$\alpha_j \leftarrow (r^j)^T r^j / (p^j)^T \bar{H} p^j$; {Begin Standard CG Operations}

$y^{j+1} \leftarrow y^j + \alpha_j p^j$;

$r^{j+1} \leftarrow r^j + \alpha_j \bar{H} p^j$;

$\beta_{j+1} \leftarrow \|r^{j+1}\|^2 / \|r^j\|^2$;

$p^{j+1} \leftarrow -r^{j+1} + \beta_{j+1} p^j$; {End Standard CG Operations}

$j \leftarrow j + 1$;

if $\|H p^j\| > U \|p^j\|$ **then**

 Set $U \leftarrow \|H p^j\| / \|p^j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;

end if

if $\|H y^j\| > U \|y^j\|$ **then**

 Set $U \leftarrow \|H y^j\| / \|y^j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;

end if

if $\|H r^j\| > U \|r^j\|$ **then**

 Set $U \leftarrow \|H r^j\| / \|r^j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly;

end if

if $(y^j)^T \bar{H} y^j < \varepsilon \|y^j\|^2$ **then**

 Set $\hat{d} \leftarrow y^j$ and terminate with d_type = NC;

else if $\|r^j\| \leq \hat{\zeta} \|r^0\|$ **then**

 Set $\hat{d} \leftarrow y^j$ and terminate with d_type = SOL;

else if $(p^j)^T \bar{H} p^j < \varepsilon \|p^j\|^2$ **then**

 Set $\hat{d} \leftarrow p^j$ and terminate with d_type = NC;

else if $\|r^j\| > \sqrt{T} \tau^{j/2} \|r^0\|$ **then**

 Compute α_j, y^{j+1} as in the main loop above;

 Find $i \in \{0, \dots, j-1\}$ such that

$$(y^{j+1} - y^i)^T \bar{H} (y^{j+1} - y^i) < \varepsilon \|y^{j+1} - y^i\|^2;$$

 Set $\hat{d} \leftarrow y^{j+1} - y^i$ and terminate with d_type = NC;

end if

end while

Theorem A.1 (iteration complexity of Algorithm 3). *Consider applying Algorithm 3 with the optional input $U = 0$ to the linear system (15) with $g \neq 0$, $\varepsilon > 0$, and H being an $n \times n$ symmetric matrix. Then the number of iterations of Algorithm 3 is $\tilde{O}(\min\{n, \sqrt{\|H\|/\varepsilon}\})$.*

B A randomized Lanczos based minimum eigenvalue oracle

In this part we present the randomized Lanczos method proposed in [60, Section 3.2], which can be used as a minimum eigenvalue oracle for Algorithm 1. As mentioned in Section 4, this oracle either outputs a sufficiently negative curvature direction of H or certifies that H is nearly positive semidefinite with high probability. More details about it can be found in [60, Section 3.2].

The following theorem justifies that Algorithm 4 is a suitable minimum eigenvalue oracle for Algorithm 1. Its proof is identical to that of [60, Lemma 2] and thus omitted.

Theorem B.1 (iteration complexity of Algorithm 4). *Consider Algorithm 4 with tolerance $\varepsilon > 0$, probability parameter $\delta \in (0, 1)$, and symmetric matrix $H \in \mathbb{R}^{n \times n}$ as its input. Then it either finds a sufficiently negative*

Algorithm 4 A randomized Lanczos based minimum eigenvalue oracle

Input: symmetric matrix $H \in \mathbb{R}^{n \times n}$, tolerance $\varepsilon > 0$, and probability parameter $\delta \in (0, 1)$.

Output: a sufficiently negative curvature direction v satisfying $v^T Hv \leq -\varepsilon/2$ and $\|v\| = 1$; or a certificate that $\lambda_{\min}(H) \geq -\varepsilon$ with probability at least $1 - \delta$.

Apply the Lanczos method [48] to estimate $\lambda_{\min}(H)$ starting with a random vector uniformly generated on the unit sphere, and run it for at most

$$N(\varepsilon, \delta) := \min \left\{ n, 1 + \left\lceil \frac{\ln(2.75n/\delta^2)}{2} \sqrt{\frac{\|H\|}{\varepsilon}} \right\rceil \right\} \quad (111)$$

iterations.

- (i) If it finds a unit vector v such that $v^T Hv \leq -\varepsilon/2$ at some iteration, it terminates immediately and returns v .
- (ii) Otherwise, it certifies that $\lambda_{\min}(H) \geq -\varepsilon$ holds with probability at least $1 - \delta$.

curvature direction v satisfying $v^T Hv \leq -\varepsilon/2$ and $\|v\| = 1$ or certifies that $\lambda_{\min}(H) \geq -\varepsilon$ holds with probability at least $1 - \delta$ in at most $N(\varepsilon, \delta)$ iterations, where $N(\varepsilon, \delta)$ is defined in (111).

Notice that generally, computing $\|H\|$ in Algorithm 4 may not be cheap when n is large. Nevertheless, $\|H\|$ can be efficiently estimated via a randomization scheme with high confidence (e.g., see the discussion in [60, Appendix B3]).