A LOG-BARRIER NEWTON-CG METHOD FOR BOUND CONSTRAINED OPTIMIZATION WITH COMPLEXITY GUARANTEES*

MICHAEL O'NEILL AND STEPHEN J. WRIGHT

Abstract. We describe an algorithm based on a logarithmic barrier function, Newton's method, and linear conjugate gradients that seeks an approximate minimizer of a smooth function over the nonnegative orthant. We develop a bound on the complexity of the approach, stated in terms of the required accuracy and the cost of a single gradient evaluation of the objective function and/or a matrix-vector multiplication involving the Hessian of the objective. The approach can be implemented without explicit calculation or storage of the Hessian.

1 Introduction We consider the following constrained optimization problem:

$$\min f(x) \quad \text{subject to } x \geq 0, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a nonconvex function, twice uniformly Lipschitz continuously differentiable in the interior of the nonnegative orthant. We assume that explicit storage of the Hessian $\nabla^2 f(x)$ for $x > 0$ is undesirable, but that Hessian-vector products of the form $\nabla^2 f(x)v$ can be computed at any $x > 0$ for arbitrary vectors v. Computational differentiation techniques [29] can be used to evaluate such products at a cost that is a small multiple of the cost of evaluation of the gradient ∇f.

The problem (1) is well studied, with numerous algorithms being proposed over the years, based on such strategies as active set, gradient projection, and Newton's method. Other possible approaches include interior-point and barrier methods, which generate iterates that remain strictly feasible. The primal log-barrier method minimizes the log-barrier function

$$\phi_\mu(x) = f(x) - \mu \sum_{i=1}^{n} \log(x_i), \tag{2}$$

for some decreasing sequence of positive scalars μ [27]. The function ϕ_μ can be minimized using Newton's method with a line search strategy that maintains strict positivity of the components of x as well as ensuring sufficient decrease at each iteration.

Our goal in this paper is to design and analyze a method with attractive worst-case complexity guarantees comparable to those that have been attained recently for unconstrained minimization of smooth nonconvex functions. The algorithm we describe in this paper combines the primal log-barrier formulation (2) with the Newton-Conjugate-Gradient ("Newton-CG") algorithm of [36]. We minimize the log-barrier function ϕ_μ for only a single value of μ, chosen judiciously to ensure that its approximate minimizer coincides with an approximate solution to (1) that satisfies our accuracy criteria. The Newton-CG method applied to ϕ_μ uses a safeguarded version of the linear CG method to minimize a slightly damped second-order Taylor series approximation of ϕ_μ at each iteration. In contrast to its application to unconstrained optimization, the linear system is preconditioned to control the norm of its coefficient

*Version of December 3, 2019. Research supported by NSF Awards IIS-1447449, 1628384, 1634597, and 1740707; AFOSR Award FA9550-13-1-0138; Subcontracts 3F-30222 and 8F-30039 from Argonne National Laboratory; and Award N660011824020 from the DARPA Lagrange Program.

matrix to ensure that the number of CG iterations is bounded by a quantity that depends on the accuracy of the desired solution. The safeguarded CG method monitors its iterates for evidence of indefiniteness in the Hessian, and outputs a direction of negative curvature for this matrix if indefiniteness is detected. If no indefiniteness is detected, this CG procedure finds an approximate Newton step. In either case, we do a backtracking line search along the chosen direction, and show that the decrease in ϕ_μ at each step is sufficient to place an overall bound on the number of iterations, allowing worst-case complexity results to be proved.

Although practical efficiency of the method is not our main concern in this paper, we note that our method is a "long-step" interior-point method, of the kind that has been useful in other settings.

The rest of this paper is organized as follows. Section 2 reviews related work, puts our paper in context, and outlines our main result. In Section 3 we derive a first- and second-order approximate optimality condition for (1). Section 4 describes our log-barrier Newton-CG algorithm, while Section 5 presents the worst-case complexity analysis for the first- and second-order approximate KKT conditions. Some conclusions appear in Section 6.

Assumptions, Background, Notation. We assume the following throughout, concerning smoothness and boundedness of f.

Assumption 1. *The function f is twice uniformly Lipschitz continuously differentiable on an open neighborhood of the path of the iterates and trial points. We denote by L_g the Lipschtiz constant for ∇f and L_H the Lipschitz constant for $\nabla^2 f$ on this set.*

Assumption 2. *The function f is bounded below by f_{low}.*

Assumption 3. *The iterates $\{x^k\}$ satisfy,*

$$\|\nabla f(x^k)\| \le U_g, \quad \|\nabla^2 f(x^k)\| \le U_H,$$

for some scalars $U_g > 0$ and $U_H > 0$.

(Here and throughout we use $\|\cdot\|$ to denote the Euclidean norm, or its induced norm on matrices.) We observe that U_H is a Lipschitz constant for the gradient of f.

For any x and y such that Assumption 1 is satisfied, we have

$$\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x)\| \le \frac{1}{2} L_H \|x-y\|^2, \tag{3}$$

$$f(y) \le f(x) + \nabla f(x)^\top (y-x) + \frac{1}{2}(y-x)^\top \nabla^2 f(x)(y-x) + \frac{1}{6} L_H \|x-y\|^3. \tag{4}$$

Order notation $\mathcal{O}$ is used in its usual sense, whereas $\tilde{\mathcal{O}}$ represents $\mathcal{O}$ with logarithmic factors omitted.

We define $e = (1,\dots,1)^\top$ to be the vector of ones and $e_i = (0,\dots,0,1,0,\dots,0)^\top$ to be the unit vector with 1 as the ith component and zeros elsewhere. The ith component of a vector v is denoted by v_i or $[v]_i$. Given a vector $x \in \mathbb{R}^n_+$ (where $\mathbb{R}^n_+$ is the nonnegative orthant), we denote by X the diagonal matrix formed by the components of x, by $\bar{x}$ the vector whose components are $\min(x_i, 1)$,[1] and by $\bar{X}$ the diagonal matrix formed from $\bar{x}$. That is,

$$X = \operatorname{diag}\,(x_1, x_2, \dots, x_n), \quad \bar{x} = \min(x, e), \quad \bar{X} = \operatorname{diag}\,(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n). \tag{5}$$

[1] We use a threshold of 1 for clarity of presentation. Any other positive value could be used instead, with minimal effect on the results.

Our algorithm seeks a point x satisfying the following approximate optimality conditions for (1):

$$x > 0, \tag{6a}$$

$$\nabla f(x) \geq -\epsilon_g e, \tag{6b}$$

$$\|\bar{X}\nabla f(x)\|_\infty \leq \epsilon_g, \tag{6c}$$

$$\bar{X}\nabla^2 f(x)\bar{X} \succeq -\epsilon_H I, \tag{6d}$$

for small positive tolerances ϵ_g and ϵ_H. The conditions (6c) and (6d) differ from the scaled gradient and Hessian conditions used elsewhere, through the substitution of the bounded matrix $\bar{X}$ for X. The theoretical basis for these conditions as well as their relation to those used in previous works is presented in Section 3.

2 Related Work There is considerable recent work on algorithms for *unconstrained* smooth nonconvex optimization that have optimal worst-case iteration complexity for finding points that satisfy approximate first- and second-order optimality conditions. When applied to twice Lipschitz continuously differentiable functions, classical Newton-trust-region schemes [22] require at most $\mathcal{O}\left(\max\left\{\epsilon_g^{-2}\epsilon_H^{-1}, \epsilon_H^{-3}\right\}\right)$ iterations [16] to find a point satisfying

$$\|\nabla f(x)\| \leq \epsilon_g \text{ and } \lambda_{\min}(\nabla^2 f(x)) \geq -\epsilon_H. \tag{7}$$

For this class of problems, the optimal iteration complexity for finding a second-order optimal point is $\mathcal{O}\left(\max\left\{\epsilon_g^{-3/2}, \epsilon_H^{-3}\right\}\right)$ [9, 14, 19]. This iteration complexity was first achieved by cubic regularization of Newton's method [34]. Numerous other algorithms have also been proposed that match this iteration bound; see for example [5, 13, 23, 25, 33].

Some works also account for the computational cost of each iteration, thus yielding a bound on the overall computational complexity. A number of works have focused on efficiently computing a solution to the cubically regularized subproblem, either through direct matrix factorization techniques [8, 13, 34] and/or Krylov subspace based methods [13, 26]. These approaches yield a worst case operational complexity of $\mathcal{O}(n\epsilon_g^{-3/2})$ when $\epsilon_H = \epsilon_g^{1/2}$. Two independently proposed algorithms, respectively based on adapting accelerated gradient to the nonconvex setting [11] and approximately solving the cubic regularization subproblem [1], require $\tilde{\mathcal{O}}(\epsilon_g^{-7/4})$ operations (with high probability, showing dependency only on ϵ_g) to find a point x that satisfies (7) when $\epsilon_H = \epsilon_g^{1/2}$. The difference of a factor of $\epsilon_g^{-1/4}$ with the iteration complexity bounds arises from the cost of computing a negative curvature direction of $\nabla^2 f(x_k)$ and/or the cost of solving a linear system. The probabilistic nature of the bound is due to the introduction of randomness in the curvature estimation process. A complexity bound of the same type was also established for a variant of accelerated gradient based only on gradient calculations, that periodically adds a random perturbation to the iterate when the gradient norm is small [32].

In another line of work, [37] developed a damped Newton algorithm which inexactly minimizes the Newton system by the method of conjugate gradients and requires at most $\tilde{\mathcal{O}}(\min\{n\epsilon_g^{-3/2}, \epsilon_g^{-7/4}\})$ operations to satisfy (7), to high probability. For purposes of computational complexity, this paper defines the unit of computation to be one Hessian-vector product *or* one gradient evaluation. We also adopt this definition here; it relies implicitly on the observation from computational / algorithmic differentiation [29] that these two operations differ in cost only by a modest factor, independent of the dimension n. In a followup to [37], the paper [36] built on techniques from [10] to create a modified CG method to solve the Newton system. This algorithm, which is a foundation of the method described in this paper, again finds a point satisfying (7) in $\tilde{\mathcal{O}}(\min\{n\epsilon_g^{-3/2}, \epsilon_g^{-7/4}\})$ operations, to high probability, and requires the same number of operations to find an approximate first-order critical point *deterministically*.

A number of algorithms have also been proposed for *constrained* optimization problems that require at most $\mathcal{O}(\max\{\epsilon_g^{-3/2}, \epsilon_H^{-3}\})$ iterations to find a point which satisfies some first-order (and sometimes second-order) optimality conditions. Although the optimality conditions vary between papers, the works [15, 17] achieve this iteration complexity bound for some first-order optimality condition by solving a constrained cubic regularization subproblem at each iteration. These approaches have been greatly simplified in recent times for problems involving "inexpensive" convex constraints [18, 20]. A different proposal finds a first-order point in $\mathcal{O}(\epsilon_g^{-3/2})$ iterations for linear equality and bound constraints through the use of an active set method [6]. When optimizing on a single face of the polytope, this method also uses a cubic regularization model. However, these papers do not account for the cost of solving the subproblem at each iteration, noting either that this subproblem may be NP-hard, or suggesting that a simple first-order, gradient-based method can solve it reliably. Many other methods have been proposed for constrained optimization which have good worst-case iteration complexity results, such as two-phase methods [4, 12, 24], an interior-point method [31], and augmented Lagrangian methods [7, 28, 38].

Turning to our bound-constrained problem (1), a second-order interior-point method was proposed in [3]. This method minimizes a preconditioned second-order trust-region model at each iteration and finds a point satisfying approximate second-order conditions in at most $\mathcal{O}(\epsilon_g^{-3/2})$ iterations when $\epsilon_H = \epsilon_g^{1/2}$. However, the first-order conditions are strictly weaker than those used in the current work as they consist only of feasibility of x along with a scaled gradient condition that is an "unbounded" version of (6c) in which $\bar{X}$ is replaced by X. Without additional assumptions on f, the absence of condition (6b) in the optimality conditions implies that sequences of (strictly feasible) points that satisfy the scaled gradient condition may not approach KKT points as ϵ_g approaches 0; see [30, Section 2] for a discussion of this issue. Our approximate optimality conditions (6) here do not suffer from these issues, as we show in Section 3. In a follow up to [3], an interior-point method for linear equality and bound constraints was described in [30]. This method, which also achieves an iteration complexity of $\mathcal{O}(\epsilon_g^{-3/2})$ (when $\epsilon_H = \epsilon_g^{1/2}$), applies a constrained second-order trust-region algorithm to the log-barrier function, with a (potentially) small trust-region radius. The authors of [30] were more interested in iteration complexity than computational complexity, but we note that each of their subproblems requires evaluation of the Hessian (which in the worst case requires evaluation of n Hessian-vector products, where the latter is one of our units of computational complexity), together with $\tilde{\mathcal{O}}(n^3)$ floating point operations associated with performing a bisection scheme to solve the subproblem. These considerations suggest an overall worst-case computational complexity of at least $\mathcal{O}(n\epsilon_g^{-3/2})$ for the algorithm of [30].

In this paper, we adapt the Newton-CG method of [36] for unconstrained optimization to the problem of minimizing the primal log-barrier function (2), for a small, fixed value of μ. We target the optimality conditions (6), which avoid enforcing tighter conditions on Hessian and gradient components that correspond to components of x

that are far from zero at optimality. This change allows us to solve a preconditioned Newton system of linear equations at each iteration in which the norm of the matrix can be bounded by a constant independent of iteration number. The Capped CG method developed in [36] is used to solve this system, returning a useful search direction in a reasonable number of iterations. When $\epsilon_H = \epsilon_g^{1/2}$, our algorithm finds a point satisfying (6) in $\tilde{\mathcal{O}}(n\epsilon_g^{-1/2} + \epsilon_g^{-3/2})$ iterations (Theorem 16). The computational complexity, in terms of gradient evaluations/Hessian vector products, is $\tilde{\mathcal{O}}(n\epsilon_g^{-3/4} + \epsilon_g^{-7/4})$ for large values of n, and $\tilde{\mathcal{O}}(n\epsilon_g^{-3/2})$ for smaller n; see Corollary 17 and the comments following this result. The appearance of n in our complexity expressions is an apparently unavoidable consequence of using log-barrier methodology, along with making the mildest possible assumptions on the problem (1) and the algorithm. For example, we do not assume a bounded feasible set or a particular choice of starting point (as in [30]), and we do not assume any specific rate of growth of f as x moves away from the solution set. Still, our computational complexity rates match (for small n) or improve on (for large n) those in [30]. Practically speaking, our algorithm has the appealing feature that it puts minimal restrictions on the step size, allowing the line search to take steps that are much closer to the boundary than the current iterate.

3 Approximate Optimality Conditions We now discuss first- and second-order optimality criteria for (1) in a form that can be related to the approximate optimality criteria (6) that are targeted by our algorithm. We show that points satisfying these necessary conditions are the limits of sequences of points that satisfy our approximate criteria (6). We then compare our approximate criteria with similar conditions that have been proposed previously, and argue that ours are more appropriate.

3.1 Deriving Approximate Optimality Conditions from Exact Conditions First-order conditions for x to be a solution of (1) are that there exists a vector $s^* \in \mathbb{R}^n$ such that

$$\nabla f(x) - s^* = 0, \quad (x, s^*) \geq 0, \quad x_i s_i^* = 0 \quad \text{for all } i = 1, 2, \ldots n. \tag{8}$$

Our second-order condition is a modified version of the condition derived in [2]. It requires the existence of a vector θ^* such that

$$\nabla^2 f(x) + \text{diag}\,(\theta^*) \succeq 0, \quad \theta^* \geq 0, \quad x_i^2 \theta_i^* = 0 \quad \text{for all } i = 1, 2, \ldots n. \tag{9}$$

This is equivalent to a "weak" form of second-order necessary conditions for (1), namely $[\nabla^2 f(x)]_{\mathcal{I}(x)\mathcal{I}(x)} \succeq 0$, where $\mathcal{I}(x) := \{i \,|\, x_i > 0\}$. The more satisfactory "strong" second-order conditions require testing that $d^\top \nabla^2 f(x) d \geq 0$ for all d in the cone defined by

$$\{d \in \mathbb{R}^n \,|\, d_i = 0 \text{ when } x_i = 0,\, [\nabla f(x)]_i > 0;\ d_i \geq 0 \text{ when } x_i = 0,\, [\nabla f(x)]_i = 0\}.$$

This is known to be an NP-hard problem [35].

The following result shows that a local minimizer x^* can be expressed in terms of the limit of sequences that satisfy approximate forms of these two optimality conditions.

THEOREM 1. *Let f be twice continuously differentiable on the interior of $\mathbb{R}^n_+$. Let x^* be a local solution of (1). Then there exists a sequence of approximate solutions*

$\{x^k\}$ *with* $x^k > 0$*; sequences of approximate Lagrange multipliers* $\{s^k\}$ *and* $\{\theta^k\}$*, with* $s^k \geq 0$ *and* $\theta^k \geq 0$*; and a sequence of scalars* $\{\delta_k\}$ *with* $\delta_k > 0$ *and* $\delta_k \to 0$ *such that the following conditions hold:*

$$x^k > 0 \text{ for all } k \text{ and } x^k \to x^*, \tag{10a}$$

$$\nabla f(x^k) - s^k \to 0, \tag{10b}$$

$$\min\{x_i^k, 1\} s_i^k \to 0 \quad \text{for all } i = 1, 2, \ldots n, \tag{10c}$$

$$\nabla^2 f(x^k) + \text{diag}\,(\theta^k) + \delta_k I \succeq 0, \tag{10d}$$

$$\min\{x_i^k, 1\}^2 \theta_i^k \to 0 \quad \text{for all } i = 1, 2, \ldots n. \tag{10e}$$

The proof of this result follows directly from that of [30, Theorem 1] by noting that $\min\{x_i^k, 1\} s_i^k \leq x_i^k s_i^k$ and $\min\{x_i^k, 1\}^2 \theta_i^k \leq (x_i^k)^2 \theta_i^k$ trivially hold for all i and k.

Theorem 1 suggests that we should declare $x > 0$ to be an approximate interior solution of (1) when there exist $s \in \mathbb{R}^n_+$ and $\theta \in \mathbb{R}^n_+$ such that

$$\|\nabla f(x) - s\|_\infty \leq \epsilon_g, \tag{11a}$$

$$\|\bar{X} s\|_\infty \leq \epsilon_g, \tag{11b}$$

$$\nabla^2 f(x) + \text{diag}\,(\theta) + \epsilon_H I \succeq 0, \tag{11c}$$

$$\|\bar{X}^2 \theta\|_\infty \leq \epsilon_H. \tag{11d}$$

We will now describe the connection between our approximate optimality conditions (6) and the conditions (11).

THEOREM 2. *Let* x *be a point satisfying (6). Then there exist* $s \in \mathbb{R}^n_+$ *and* $\theta \in \mathbb{R}^n_+$ *such that (11) holds at* x*.*

Proof. Let $s_i := \max\{0, [\nabla f(x)]_i\}$ for $i = 1 \ldots n$, so that $s \in \mathbb{R}^n_+$ and, by direct substitution, we have (11a) and (11b). Our second-order condition (6d) is that

$$d^\top \left(\bar{X} \nabla^2 f(x) \bar{X} + \epsilon_H I \right) d \geq 0, \quad \text{for all } d \in \mathbb{R}^n.$$

Since $\bar{X}^{-1}$ exists and is positive definite, we have

$$d^\top \left(\nabla^2 f(x) + \sum_{i=1}^n \frac{\epsilon_H}{\min\{x_i, 1\}^2} e_i e_i^\top \right) d \geq 0, \quad \text{for all } d \in \mathbb{R}^n.$$

Therefore, by choosing $\theta_i = \epsilon_H / \min\{x_i, 1\}^2$ for all $i = 1, 2, \ldots n$, we have that $\theta \geq 0$ and that (11c) and (11d) are both satisfied. ∎

3.2 Comparison with Previously Proposed Approximate Conditions

The conditions (8) and (9) directly motivate the approximate optimality conditions for $x > 0$ used in the interior-point method of [30], which are

$$\nabla f(x) \geq -\epsilon_g e, \tag{12a}$$

$$\|X \nabla f(x)\|_\infty \leq \epsilon_g, \tag{12b}$$

$$d^\top \left(X \nabla^2 f(x) X + \sqrt{\epsilon_g} I \right) d \geq 0. \tag{12c}$$

The scaled first-order condition (12b) and scaled second-order condition (12c) are commonly used optimality conditions for (1) [3, 21]. However, these two conditions alone are insufficient to guarantee that a sequence of points that satisfies these conditions as $\epsilon_g \to 0$ converges to a KKT point for f [30]. For this reason the condition (12a) is added in [30], motivated by the first-order optimality conditions (8).

These conditions can be overly stringent for coordinates i in which $x_i \gg 0$. In this case, the complementarity condition (12b), requires $|[\nabla f(x)]_i|$ to be very small. Similarly, (12c) requires that the Hessian in the subspace spanned by these coordinates can have only minimal negative curvature. Such requirements contrast sharply with the case of unconstrained minimization. In the limiting scenario in which all of the coordinates of x are far from the boundary, these approximate first-order conditions are significantly harder to satisfy than in the (equivalent) unconstrained formulation.

To remedy this situation, our approximate optimality conditions (6) contain scalings by x_i only when $x_i \in (0, 1]$. Our conditions thus interpolate between the bound-constrained case (when x_i is small) and the unconstrained case (when x_i is large) while also controlling the norm of the matrix used in our optimality conditions.

4 Log-Barrier Newton-CG Algorithm We now give an overview of our Log-Barrier Newton-CG (LBNCG) algorithm, defined in Algorithm 1, along with its component parts.

The main branch in each iteration is conditional on the approximate first-order optimality conditions, (6b) and (6c). When one or both of these conditions are not satisfied, the Capped CG method (Algorithm 2) is applied to the damped, preconditioned Newton system

$$\left(\bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k + 2\epsilon_H I\right) d = \bar{X}_k \nabla \phi_\mu(x^k), \tag{14}$$

where according to the definition (2) of the barrier function ϕ_μ, we have

$$\nabla \phi_\mu(x) = \nabla f(x) - \mu X^{-1} e \quad \text{ and } \quad \nabla^2 \phi_\mu(x) = \nabla^2 f(x) + \mu X^{-2}.$$

Algorithm 2, which is described further in Section 4.1 and in the earlier paper [36], returns either an approximate solution to the linear system (14), or else a direction of sufficient negative curvature for $\bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k$.

Alternatively, when (6b) and (6c) are satisfied, a "Minimum Eigenvalue Oracle" (Procedure 3) is invoked to certify either that the second-order optimality condition (6d) holds at the current iterate or, if not, to return a direction v of sufficient negative curvature for $\bar{X}_k \nabla f(x^k) \bar{X}_k$. Procedure 3 may be implemented by a randomized procedure, with some probability of failure δ, in which it incorrectly certifies that (6d) is satisfied. Further discussion of this procedure appears in Section 4.2.

However the search direction is chosen, it is scaled to obtain a step d^k that satisfies $\|X_k^{-1} \bar{X}_k d^k\|_\infty \le \beta < 1$. This condition guarantees that for $x^k > 0$, we have

$$x^{k+1} = x^k + \bar{X}_k d^k = X_k \left(e + X_k^{-1} \bar{X}_k d^k\right) \ge x^k(1-\beta) > 0,$$

so that all iterates lie strictly inside the positive orthant. A backtracking linesearch is performed along the direction $\bar{X}_k d^k$ to ensure sufficient decrease in ϕ_μ. We note that a value of β close to its upper bound of 1 results in aggressive steps that may approach the zero bounds closely. Steps of this kind are favored in practical interior-point methods. We will see in later sections that a factor $(1-\beta)$ emerges in the complexity results, leading to weaker bounds if β is *too* close to 1. Though we are mindful of this effect, our focus is on the dependence on the tolerance ϵ_g. The choice of β is independent of ϵ_g; we would not expect β to be updated in response to a change in the tolerance ϵ_g.

Algorithm 1 Log-Barrier Newton-Conjugate-Gradient

Inputs: Tolerance $\epsilon_g \in (0,1)$; backtracking parameter $\theta \in (0,1)$; starting point $x^0 > 0$; accuracy parameters $\zeta_r \in (0,1)$ and $\bar{\zeta} \in (0,1)$; maximum step scaling $\beta \in [\epsilon_g^{1/2}, 1)$; step acceptance parameter $\eta \in (0,1)$;
Optional input: Scalar $\hat{M} > 0$ such that $\|\nabla^2 f(x)\| \le \hat{M}$ for all x (set $\hat{M} = 0$ if not provided);
Set $\epsilon_H = \epsilon_g^{1/2}$, $\mu = \epsilon_g/4$, $c_\mu = \bar{\zeta}\mu$, $M_\mu = \hat{M} + \mu$;
for $k = 0, 1, 2, \ldots$ **do**
  if $[\nabla f(x^k)]_i \le -\epsilon_g$ for some coordinate i or $\|\bar{X}_k \nabla f(x^k)\|_\infty > \epsilon_g$ **then**
    Call Algorithm 2 with $H = \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k$, $\epsilon = \epsilon_H$, $g = \bar{X}_k \nabla \phi_\mu(x^k)$, accuracy parameters ζ_r and c_μ, and bound $M = M_\mu$, to obtain outputs $\hat{d}^k$, d_type;
    if {d_type=NC} **then**
      $d^k \leftarrow -\mathrm{sgn}(g^\top \hat{d}^k) \min\left\{ \frac{|(\hat{d}^k)^\top \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k \hat{d}^k|}{\|\hat{d}^k\|^3}, \frac{\beta}{\|X_k^{-1} \bar{X}_k \hat{d}^k\|_\infty} \right\} \hat{d}^k$;
    else {d_type=SOL}
      $d^k \leftarrow \min\left\{ 1, \frac{\beta}{\|X_k^{-1} \bar{X}_k \hat{d}^k\|_\infty} \right\} \hat{d}^k$;
    end if
    Go to **Line Search**;
  else
    Call Procedure 3 with $H = \bar{X}_k \nabla^2 f(x^k) \bar{X}_k$, $\epsilon = \epsilon_H$, and $M = \hat{M}$ (if provided);

    if Procedure 3 certifies that $\lambda_{\min}(\bar{X}_k \nabla^2 f(x^k) \bar{X}_k) \ge -\epsilon_H$ **then**
      Terminate;
    else {direction of sufficient negative curvature v returned by Procedure 3}
      Set $d^k \leftarrow -\mathrm{sgn}(v^\top \bar{X}_k \nabla \phi_\mu(x^k)) \min\left\{ |v^\top \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k v|, \frac{\beta}{\|X_k^{-1} \bar{X}_k v\|_\infty} \right\} v$;
      Go to **Line Search**;
    end if
  end if
  Line Search: Compute a step length $\alpha_k = \theta^{j_k}$, where j_k is the smallest nonnegative integer such that

$$\phi_\mu(x^k + \alpha_k \bar{X}_k d^k) < \phi_\mu(x^k) - \frac{\eta}{6} \alpha_k^3 \|d^k\|^3; \tag{13}$$

  $x^{k+1} \leftarrow x^k + \alpha_k \bar{X}_k d^k$;
end for

We set a number of parameters at the beginning of the algorithm, including the particular choice $\epsilon_H = \epsilon_g^{1/2}$. This choice is commonly made in the unconstrained optimization literature too, for purposes of aligning two different complexity expressions. In our current context, this choice is embedded more deeply into the analysis, but we keep the distinction between ϵ_H and ϵ_g to maintain the generality of individual results. The particular choice $\mu = \epsilon_g/4$ of the barrier parameter is key to the complexity result. Finally, we note that when $\hat{M}$ is an upper bound on $\|\nabla^2 f(x)\|$ for all x of interest, we have

$$\|\bar{X} \nabla^2 \phi_\mu(x) \bar{X}\| \le \|\bar{X} \nabla^2 f(x) \bar{X}\| + \mu \|\bar{X} X^{-2} \bar{X}\| \le \|\nabla^2 f(x)\| + \mu \le \hat{M} + \mu, \tag{15}$$

so that $\|H\| \le M_\mu$ for H defined as the input of Algorithm 2 in Algorithm 1.

4.1 Capped Conjugate Gradient Algorithm 2 is a safeguarded version of the conjugate gradient (CG) procedure for either solving the linear system $(H + 2\epsilon I)y = -g$, or else detecting a direction d such that $d^\top H d \le -\epsilon\|d\|^2$. This method, which was described in [36], consists of classical CG iterations plus various checks to determine whether (a) the upper bound M on $\|H\|$ is adequate, and (b) negative curvature in H has been detected. One of the techniques for detecting negative curvature is the too-slow-convergence criterion $\|r^j\| > \sqrt{T}\tau^{j/2}\|r^0\|$ (where T and τ both depend on the bound M). By Theorem 6, this behavior can occur only when there exists some $i \in \{0, \dots, j-1\}$ such that $(y^{j+1} - y^i)^\top \bar{H}(y^{j+1} - y^i) < \epsilon\|y^{j+1} - y^i\|^2$ holds. In this situation, Algorithm 2 returns $d = y^{j+1} - y^i$ as a direction of sufficient negative curvature.

Algorithm 2 is called from Algorithm 1 with $H = \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k$ which, as we note in (15), has norm bounded by $M_\mu = \hat{M} + \mu$, where $\hat{M}$ is the bound on $\|\nabla^2 f(x^k)\|$. Hence the value of M in Algorithm 2 will never be larger than this value.

Altogether, the safeguards mentioned above and the diagonal preconditioning strategy guarantee that Capped CG requires $\min\{n, \tilde{\mathcal{O}}(\epsilon^{-1/2})\}$ iterations to terminate. A derivation of this bound is given in Section 5.1.

4.2 Minimum Eigenvalue Oracle The Minimum Eigenvalue Oracle (Procedure 3) is called when the approximate first-order conditions (6b), (6c) are satisfied. This procedure either verifies that the approximate second-order condition (6d) is satisfied as well (in which case the algorithm terminates), or else returns a direction of sufficient negative curvature for the scaled Hessian $\bar{X}_k \nabla^2 f(x^k) \bar{X}_k$, along which further progress can be made in reducing the barrier function ϕ_μ.

This procedure can be implemented via any method that finds the smallest eigenvalue of H to an absolute precision of $\epsilon/2$ with probability at least $1 - \delta$. (A deterministic implementation based on a full eigenvalue decomposition would have $\delta = 0$.) In Section 5.3, we will establish complexity results under this general setting, and analyze the impact of the threshold δ.

Several possibilities for implementing Procedure 3 have been proposed in the literature, with various guarantees. In our setting, in which Hessian-vector products and vector operations are the fundamental operations, Procedure 3 can be implemented using the Lanczos method with a random starting vector (see [11]). The following result from [36, Lemma 2] verifies its effectiveness.

LEMMA 3. *Suppose that the Lanczos method is used to estimate the smallest eigenvalue of H starting with a random vector uniformly generated on the unit sphere, where $\|H\| \le M$. For any $\delta \in [0, 1)$, this approach finds the smallest eigenvalue of H to an absolute precision of $\epsilon/2$, together with a corresponding direction v, in at most*

$$\min\left\{n, 1 + \left\lceil \frac{1}{2}\ln(2.75n/\delta^2)\sqrt{\frac{M}{\epsilon}} \right\rceil\right\} \quad \textit{iterations}, \tag{17}$$

with probability at least $1 - \delta$.

Procedure 3 can be implemented by outputting the approximate eigenvalue λ for H, determined by the randomized Lanczos process, along with the corresponding direction v, provided that $\lambda \le -\epsilon/2$. When $\lambda > -\epsilon/2$, Procedure 3 returns the certificate that $\lambda_{\min}(H) \ge -\epsilon$, a conclusion that is correct with probability at least $1-\delta$. Conjugate gradient with a random right-hand side can be used as an alternative

Algorithm 2 Capped Conjugate Gradient

Inputs: Symmetric matrix $H \in \mathbb{R}^{n\times n}$; vector $g \neq 0$; damping parameter $\epsilon \in (0,1)$; desired relative accuracy parameter $\zeta_r \in (0,1)$; desired accuracy $c_\mu \in (0,1)$;
Optional input: scalar $M \geq 0$ such that $\|H\| \leq M$ (set to 0 if not provided);
Outputs: d_type, d;
Secondary outputs: final values of M, κ, $\hat{\zeta}_r$, τ, and T;
Set

$$\bar{H} := H + 2\epsilon I, \quad \kappa := \frac{M+2\epsilon}{\epsilon}, \quad \hat{\zeta}_r := \frac{\zeta_r}{3\kappa}, \quad \tau := \frac{\sqrt{\kappa}}{\sqrt{\kappa}+1}, \quad T := \frac{4\kappa^4}{(1-\sqrt{\tau})^2};$$

$y^0 \leftarrow 0$, $r^0 \leftarrow g$, $p^0 \leftarrow -g$, $j \leftarrow 0$;
if $(p^0)^\top \bar{H} p^0 < \epsilon\|p^0\|^2$ **then**
  Set $d = p^0$ and terminate with d_type=NC;
else if $\|Hp^0\| > M\|p^0\|$ **then**
  Set $M \leftarrow \|Hp^0\|/\|p^0\|$ and update $\kappa, \hat{\zeta}_r, \tau, T$ accordingly;
end if
while TRUE **do**
  $\alpha_j \leftarrow (r^j)^\top r^j/(p^j)^\top \bar{H} p^j$; {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 $\|Hp^j\| > M\|p^j\|$ **then**
    Set $M \leftarrow \|Hp^j\|/\|p^j\|$ and update $\kappa, \hat{\zeta}_r, \tau, T$ accordingly;
  else if $\|Hy^j\| > M\|y^j\|$ **then**
    Set $M \leftarrow \|Hy^j\|/\|y^j\|$ and update $\kappa, \hat{\zeta}_r, \tau, T$ accordingly;
  else if $\|Hr^j\| > M\|r^j\|$ **then**
    Set $M \leftarrow \|Hr^j\|/\|r^j\|$ and update $\kappa, \hat{\zeta}_r, \tau, T$ accordingly;
  end if
  if $(y^j)^\top \bar{H} y^j < \epsilon\|y^j\|^2$ **then**
    Set $d \leftarrow y^j$ and terminate with d_type=NC;
  else if $\|r^j\| \leq \hat{\zeta}_r\|r^0\|$ and $\|r^j\|_\infty \leq c_\mu$ **then**
    Set $d \leftarrow y^j$ and terminate with d_type=SOL;
  else if $(p^j)^\top \bar{H} p^j < \epsilon\|p^j\|^2$ **then**
    Set $d \leftarrow p^j$ 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 $i \in \{0,\ldots,j-1\}$ such that

$$\frac{(y^{j+1}-y^i)^\top \bar{H}(y^{j+1}-y^i)}{\|y^{j+1}-y^i\|^2} < \epsilon; \tag{16}$$

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

Procedure 3 Minimum Eigenvalue Oracle

Inputs: Symmetric matrix $H \in \mathbb{R}^{n \times n}$, tolerance $\epsilon > 0$;
Optional input: Scalar $M > 0$ such that $\|H\| \le M$;
Outputs: An estimate λ of $\lambda_{\min}(H)$ such that $\lambda \le -\epsilon/2$, and vector v with $\|v\| = 1$ such that $v^\top H v = \lambda$ OR a certificate that $\lambda_{\min}(H) \ge -\epsilon$. In the latter case, when the certificate is output, it is false with probability at most δ, for some $\delta \in [0, 1)$.

to randomized Lanczos, with essentially the same properties; see [36, Appendices A and B].

5 Complexity Analysis This section presents complexity results for Algorithm 1. Section 5.1 describes the iteration complexity of Capped CG (Algorithm 2) and the properties of its outputs. Section 5.2 shows that Algorithm 1 *deterministically* finds a point satisfying the approximate first-order optimality conditions (6b), (6c) in at most $\tilde{\mathcal{O}}(n\epsilon_g^{-1/2} + \epsilon_g^{-3/2})$ iterations. We also show that these conditions are satisfied in at most $\tilde{\mathcal{O}}(n\epsilon_g^{-3/4} + \epsilon_g^{-7/4})$ gradient evaluations and/or Hessian-vector products when n is large and $\tilde{\mathcal{O}}(n\epsilon_g^{-3/2})$ operations when n is small. Finally, Section 5.3 shows that the same type of complexity bound holds (differing in the constants) for finding a point which satisfies all approximate optimality conditions in (6) with high probability (rather than deterministically).

5.1 Properties of Capped CG We begin this subsection by finding a lower bound on the norm of the right-hand side in the Newton system of Algorithm 1 (Lemma 4). We then derive a bound on the maximum number of iterations of the Capped CG method that can occur before returning a direction $\hat{d}^k$, which is either an approximate solution of (14) or a negative curvature direction for the diagonally scaled Hessian of the log-barrier function (Lemma 5). Theorem 6 verifies that the direction returned in the case of too-slow-decrease is in fact a vector with the required negative curvature properties. Finally, we present a number of properties of the search direction d^k computed from the vector returned by Algorithm 2, which will be instrumental in the complexity analysis of the following sections (Lemma 7).

LEMMA 4. *Let $\mu = \epsilon_g/4$ and suppose that either (6b) or (6c) is violated at x^k. Then,*

$$\|\bar{X}_k \nabla \phi_\mu(x^k)\| \ge \mu. \tag{18}$$

Proof. By definition of $\nabla \phi_\mu(x^k)$, we have

$$\|\bar{X}_k \nabla \phi_\mu(x^k)\| = \|\bar{X}_k \nabla f(x^k) - \mu \bar{X}_k X_k^{-1} e\|. \tag{19}$$

Suppose first that (6b) is not satisfied at x^k. Thus, there exists at least one coordinate i such that $[\nabla f(x^k)]_i < -\epsilon_g < 0$. If $x_i^k \le 1$, it follows that

$$\bar{x}_i^k [\nabla f(x^k)]_i - \frac{\bar{x}_i^k}{x_i^k}\mu = \bar{x}_i^k [\nabla f(x^k)]_i - \mu < -\mu.$$

If $x_i^k > 1$, we have $\bar{x}_i^k = 1$ so that

$$\bar{x}_i^k [\nabla f(x^k)]_i - \frac{\bar{x}_i^k}{x_i^k}\mu < [\nabla f(x^k)]_i < -\epsilon_g = -4\mu.$$

In either case, we have from (19) that

$$\|\bar{X}_k \nabla \phi_\mu(x^k)\| \geq \mu.$$

Now, suppose that (6c) does not hold, so that $|\bar{x}_i^k [\nabla f(x^k)]_i| > \epsilon_g$ for some i. Thus, we have

$$\|\bar{X}_k \nabla \phi_\mu(x^k)\| \geq \left| \bar{x}_i^k [\nabla f(x^k)]_i - \mu \frac{\bar{x}_i^k}{x_i^k} \right| \geq |\bar{x}_i^k [\nabla f(x^k)]_i| - \mu \frac{\bar{x}_i^k}{x_i^k} \geq \epsilon_g - \mu \geq 3\mu,$$

proving the result. $\square$

We now find the iteration bound on Algorithm 2 that was foreshadowed in Section 4.1. The precise bound in the following lemma is based on a quantity $J(M, \epsilon, \zeta_r, c_\mu)$, for which the estimate in terms of the accuracy parameter is given following the lemma.

LEMMA 5. *The number of iterations of Algorithm 2 is bounded by*

$$\min\{n, J(M, \epsilon, \zeta_r, c_\mu)\},$$

where $J = J(M, \epsilon, \zeta_r, c_\mu)$ *is the smallest integer such that*

$$\sqrt{T}\tau^{J/2}\|r^0\| \leq \min\left\{\hat{\zeta}_r \|r^0\|, c_\mu\right\}, \tag{20}$$

where M, $\hat{\zeta}_r$, T, *and* τ *are the values returned by the algorithm. If all iterates* y_i *generated by Algorithm 2 are stored, the number of matrix-vector multiplications required is bounded by* $\min\{n, J(M, \epsilon, \zeta_r, c_\mu)\} + 1$. *If the iterates* y_i *must be regenerated in order to define the direction* d *returned after* (16), *this bound becomes* $2\min\{n, J(M, \epsilon, \zeta_r, c_\mu)\} + 1$.

Proof. We omit a detailed proof, as the result and proof are identical to [36, Lemma 1] modulo a new definition of J. We need only consider the case in which $J < n$, where J is the index defined in the lemma. If $\|r^J\| > \sqrt{T}\tau^{J/2}\|r^0\|$, the last termination test in Algorithm 2 ensures termination at iteration J. In the alternative case $\|r^J\| \leq \sqrt{T}\tau^{J/2}\|r^0\|$, we have by definition of J that

$$\|r^J\| \leq \sqrt{T}\tau^{J/2}\|r^0\| \leq \min\left\{\hat{\zeta}_r \|r^0\|, c_\mu\right\}.$$

Therefore, $\|r^J\| \leq \hat{\zeta}_r \|r^0\|$ and $\|r^J\|_\infty \leq \|r^J\| \leq c_\mu$ both hold. Thus, by the termination tests in Algorithm 2, termination occurs in this case as well, completing the proof. $\square$

We can now estimate $J(M, \epsilon, \zeta_r, c_\mu)$ when Algorithm 2 is called by Algorithm 1 and Assumption 3 holds. Here, we have $r^0 = \bar{X}_k \nabla \phi_\mu(x^k)$ and $c_\mu = \bar{\zeta}\mu$, so that the right-hand side of condition (20) is

$$\min\left\{\hat{\zeta}_r \|\bar{X}_k \nabla \phi_\mu(x^k)\|, \bar{\zeta}\mu\right\}. \tag{21}$$

Using the same argument as in [36], when the minimum in (21) is achieved by the first argument, we have

$$J(M, \epsilon, \zeta_r, c_\mu) \leq \left\lceil \left(\sqrt{\kappa} + \frac{1}{2}\right) \ln\left(\frac{144\left(\sqrt{\kappa}+1\right)^2 \kappa^6}{\zeta_r^2}\right)\right\rceil = \tilde{\mathcal{O}}\left(\epsilon^{-1/2}\right). \tag{22}$$

On the other hand, when the minimum in (21) is achieved by the second argument, an argument of [36] along with the bound

$$\|\bar{X}_k \nabla \phi_\mu(x^k)\| \le \|\bar{X}_k \nabla f(x^k)\| + \mu \|\bar{X}_k X_k^{-1} e\| \le U_g + \mu\sqrt{n},$$

shows that

$$\begin{aligned} J(M,\epsilon,\zeta_r,c_\mu) &\le \left\lceil \left(\sqrt{\kappa}+\frac{1}{2}\right) \ln\left(\frac{16\left(\sqrt{\kappa}+1\right)^2 \kappa^4 (U_g+\mu\sqrt{n})^2}{\bar{\zeta}^2\mu^2}\right)\right\rceil \\ &= \tilde{\mathcal{O}}(\epsilon^{-1/2}). \end{aligned} \tag{23}$$

Therefore, in either case, we have that $J(M,\epsilon,\zeta_r,c_\mu) \le \tilde{\mathcal{O}}(\epsilon^{-1/2})$, as claimed in Section 4.1.

The following theorem shows that when Algorithm 2 is terminated because of the test $\|r^j\| > \sqrt{T}\tau^{j/2}\|r^0\|$, then (16) will hold for some $i = 0,1,\ldots,j$, so that the outputs of Algorithm 2 are well defined.

THEOREM 6. *Suppose that the main loop of Algorithm 2 terminates with $j = \hat{J}$, where*

$$\hat{J} \in \{1,\ldots,\min\{n, J(M,\epsilon,\zeta_r,c_\mu)\}\},$$

(where $J(M,\epsilon,\zeta_r,c_\mu)$ is defined in Lemma 5) because the fourth termination test is satisfied and the three earlier conditions do not hold, that is, $(y^{\hat{J}})^\top \bar{H} y^{\hat{J}} \ge \epsilon\|y^{\hat{J}}\|^2$, $(p^{\hat{J}})^\top \bar{H} p^{\hat{J}} \ge \epsilon\|p^{\hat{J}}\|^2$,

$$\|r^{\hat{J}}\| > \hat{\zeta}_r\|r^0\| \quad \textit{and/or} \quad \|r^{\hat{J}}\|_\infty > c_\mu,$$

and

$$\|r^{\hat{J}}\| > \sqrt{T}\tau^{\hat{J}/2}\|r^0\| \tag{24}$$

where M, T, and τ are the values returned by Algorithm 2. Then $y^{\hat{J}+1}$ is computed by Algorithm 2, and we have

$$\frac{(y^{\hat{J}+1}-y^i)^\top \bar{H}(y^{\hat{J}+1}-y^i)}{\|y^{\hat{J}+1}-y^i\|^2} < \epsilon, \quad \textit{for some } i \in \{0,\ldots,\hat{J}-1\}. \tag{25}$$

Proof. This result follows directly from [36, Theorem 2] after noting that the properties of $\hat{J}$ used in the proof do not depend on the definition of $J(M,\epsilon,\zeta_r,c_\mu)$. In particular, $\hat{J}$ simply needs to be an index such that (24) holds and the CG process has not stopped iterating before reaching $\hat{J}$. Thus, the result holds once we account for the additional stopping criterion $\|r^{\hat{J}}\|_\infty \le c_\mu$ in the new definition of $J(M,\epsilon,\zeta_r,c_\mu)$. ∎

We focus now on the main output of Algorithm 2, which is denoted by $\hat{d}^k$ in Algorithm 1. The properties of d^k, which is obtained by scaling $\hat{d}^k$, are essential to the first- and second-order complexity analysis of later sections.

LEMMA 7. *Let Assumption 1 hold and suppose that Algorithm 2 is invoked at iteration k of Algorithm 1. Let d^k be the vector obtained in Algorithm 1 from the output $\hat{d}^k$ of Algorithm 2. For each of the two possible settings of output flag d_type, we have the following.*

1. *When* d_type=SOL*, the direction* d^k *satisfies*

$$\epsilon_H \|d^k\|^2 \le (d^k)^\top \left(\bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k + 2\epsilon_H I\right) d^k, \tag{26a}$$

$$\|d^k\| \le 1.1 \epsilon_H^{-1} \|\bar{X}_k \nabla \phi_\mu(x^k)\|, \tag{26b}$$

$$(d^k)^\top \bar{X}_k \nabla \phi_\mu(x^k) = -\gamma_k (d^k)^\top \left(\bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k + 2\epsilon_H I\right) d^k, \tag{26c}$$

where $\gamma_k = \max\left\{\frac{\|X_k^{-1}\bar{X}_k \hat{d}^k\|_\infty}{\beta}, 1\right\}$*. If* $\|X_k^{-1}\bar{X}_k \hat{d}^k\|_\infty \le \beta$ *holds, then* d^k *also satisfies*

$$\|\hat{r}^k\| \le \frac{1}{2}\epsilon_H \zeta_r \|d^k\|, \tag{27}$$

where $\hat{r}^k$ *is the residual of the scaled Newton system, defined by*

$$\hat{r}^k := \left(\bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k + 2\epsilon_H I\right) \hat{d}^k + \bar{X}_k \nabla \phi_\mu(x^k). \tag{28}$$

2. *When* d_type=NC*, the direction* d^k *satisfies* $(d^k)^\top \bar{X}_k \nabla \phi_\mu(x^k) \le 0$ *and*

$$\frac{(d^k)^\top \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k d^k}{\|d^k\|^2} \le -\|d^k\| \le -\epsilon_H. \tag{29}$$

Proof. For simplicity of notation, we use the following shorthand in the proof:

$$H = \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k, \quad g = \bar{X}_k \nabla \phi_\mu(x^k).$$

Since Algorithm 1 invoked Algorithm 2, at least one of the conditions (6b) or (6c) must be violated at x^k. Thus, by Lemma 4, we have $\|g\| \ge \mu > 0$, so the iterates of Algorithm 2 are well defined.

Consider first the case of d_type=SOL. The bounds (26a) and (26b) follow by the same argument as in the first part of the proof of [36, Lemma 3]. We now prove (26c). The residual $\hat{r}^k$ at the final iteration of CG procedure is orthogonal to all previous search directions, so that $(\hat{d}^k)^\top \hat{r}^k = 0$ (see [36, Appendix A]). Since $\hat{d}^k$ and d^k are collinear, we have $(d^k)^\top \hat{r}^k = 0$, so from (28) it follows that

$$(d^k)^\top g = -(d^k)^\top (H + 2\epsilon_H I)\hat{d}^k. \tag{30}$$

When $\|X_k^{-1}\bar{X}_k \hat{d}^k\|_\infty \le \beta$, we have $d^k = \hat{d}^k$, so

$$(d^k)^\top g = -(d^k)^\top (H + 2\epsilon_H I)\hat{d}^k = -(d^k)^\top (H + 2\epsilon_H I) d^k,$$

proving (26c) in this case. When $\|X_k^{-1}\bar{X}_k \hat{d}^k\|_\infty > \beta$, we have

$$d^k = \frac{\beta}{\|X_k^{-1}\bar{X}_k \hat{d}^k\|_\infty} \hat{d}^k$$

and thus

$$(d^k)^\top g = -(d^k)^\top (H + 2\epsilon_H I)\hat{d}^k = -\frac{\|X_k^{-1}\bar{X}_k \hat{d}^k\|_\infty}{\beta} (d^k)^\top (H + 2\epsilon_H I) d^k,$$

proving (26c) for this case as well.

Turning to (27), we note first that from termination conditions of Algorithm 2 that $\|\hat{r}^k\| \le \hat{\zeta}_r \|g\|$. Thus, using (28), we have that

$$\|\hat{r}^k\| \le \hat{\zeta}_r \|g\| \le \hat{\zeta}_r \left(\|(H + 2\epsilon_H I)\hat{d}^k\| + \|\hat{r}^k\| \right) \le \hat{\zeta}_r \left((M + 2\epsilon_H)\|\hat{d}^k\| + \|\hat{r}^k\| \right),$$

where M is the value that is returned by Algorithm 2, so that

$$\|\hat{r}^k\| \le \frac{\hat{\zeta}_r}{1 - \hat{\zeta}_r}(M + 2\epsilon_H)\|\hat{d}^k\|.$$

Using again that $\hat{\zeta}_r = \zeta_r/(3\kappa) < 1/6$ and the definition of $\hat{\zeta}_r$ in Algorithm 2, we have

$$\frac{\hat{\zeta}_r}{1 - \hat{\zeta}_r}(M + 2\epsilon_H) \le \frac{6}{5}\hat{\zeta}_r(M + 2\epsilon_H) = \frac{6}{5}\frac{\zeta_r \epsilon_H}{3} < \frac{1}{2}\zeta_r \epsilon_H,$$

which yields (27) when we note that $d^k = \hat{d}^k$ when $\|X_k^{-1}\bar{X}_k \hat{d}^k\|_\infty \le \beta$.

In the case of d_type=NC, we recall that Algorithm 1 defines

$$d^k = -\mathrm{sgn}(g^\top \hat{d}^k) \min\left\{ \frac{|(\hat{d}^k)^\top H \hat{d}^k|}{\|\hat{d}^k\|^3}, \frac{\beta}{\|X_k^{-1}\bar{X}_k \hat{d}^k\|_\infty} \right\} \hat{d}^k. \tag{31}$$

We have from positivity of the ratios in the $\min\{\cdot,\cdot\}$ expression that

$$\mathrm{sgn}(g^\top d_k) = -\mathrm{sgn}(g^\top \hat{d}^k)^2 = -1,$$

so that $g^\top d_k \le 0$. Next, since $\hat{d}^k$ and d^k are collinear, we have

$$\frac{(d^k)^\top (H + 2\epsilon_H I)(d^k)}{\|d^k\|^2} = \frac{(\hat{d}^k)^\top (H + 2\epsilon_H I)(\hat{d}^k)}{\|\hat{d}^k\|^2} \le \epsilon_H,$$

so that

$$\frac{(d^k)^\top H (d^k)}{\|d^k\|^2} \le -\epsilon_H. \tag{32}$$

When the min in (31) is achieved by the first term, we have

$$\|d^k\| = \frac{|(\hat{d}^k)^\top H \hat{d}^k|}{\|\hat{d}^k\|^2} \ge \epsilon_H,$$

proving (29) in this case. Otherwise, when the min in (31) is achieved by the second term, we have

$$\beta = \|X_k^{-1}\bar{X}_k d^k\|_\infty \le \|X_k^{-1}\bar{X}_k d^k\| \le \|X_k^{-1}\bar{X}_k\|\|d^k\| \le \|d^k\|.$$

Using this bound, along with (32) and the fact that $\beta \ge \epsilon_H$ (by definition), we have

$$\|d^k\| \ge \min\left\{ \frac{|(\hat{d}^k)^\top H \hat{d}^k|}{\|\hat{d}^k\|^2}, \beta \right\} = \min\left\{ \frac{|(d^k)^\top H (d^k)|}{\|d^k\|^2}, \beta \right\} \ge \min\{\epsilon_H, \beta\} = \epsilon_H.$$

In either case of the min in (31), we have $\|d^k\| \le -(d^k)^\top H d^k / \|d^k\|^2$, so that

$$\frac{(d^k)^\top H d^k}{\|d^k\|^2} \le -\|d^k\| \le -\epsilon_H,$$

proving (29). ∎

5.2 First-Order Complexity Analysis We now derive a worst-case complexity result for the first-order optimality condtions (6b) and (6c). We show that when Algorithm 2 returns d_type=SOL and a unit step is taken by the line search procedure in Algorithm 1 (that is, $\alpha_k = 1$), either the first-order optimality conditions hold at x^{k+1}, or else $\|d^k\|$ is large enough to make significant progress in reducing the function ϕ_μ. Theorem 13 and Corollary 14 state first-order complexity results in terms of the number of iterations of Algorithm 1 and the number of gradient evaluations and/or Hessian vector products, respectively.

Our results depend on the following technical result concerning the decrease of the log-barrier term in ϕ_μ. Its proof can be found in Appendix A.1.

LEMMA 8. *Given $x > 0$, define X, $\bar{X}$ as in (5), and suppose that $d \in \mathbb{R}^n$ is such that $\|X^{-1}\bar{X}d\|_\infty \le \beta < 1$. Then,*

$$-\sum_{i=1}^{n} \log(x_i + \bar{x}_i d_i) + \sum_{i=1}^{n} \log(x_i) \le -e^\top X^{-1}\bar{X}d + \frac{1}{2}d^\top \bar{X}X^{-2}\bar{X}d + \frac{2-\beta}{6(1-\beta)^2}\|d\|^3. \tag{33}$$

Our first result deals with the case in which a full step ($\alpha_k = 1$) is taken in Algorithm 1.

LEMMA 9. *Let Assumption 1 hold and suppose that Algorithm 2 is invoked at an iterate x^k of Algorithm 1, and returns d_type = SOL. Then, when the unit step is taken (that is, $x^{k+1} = x^k + \bar{X}_k d^k$), we have either*

$$\|d^k\| \ge c_d \epsilon_H, \quad \textit{where } c_d = \min\left\{\frac{1-\bar{\zeta}}{9}, \left(\frac{3}{2L_H}\right)^{1/2}, \frac{1}{2\left(L_H + 9/2 + \zeta_r\right)}\right\}, \tag{34}$$

or else

$$\nabla f(x^{k+1}) \ge -\epsilon_g e \quad \textit{and} \quad \|\bar{X}_{k+1}\nabla f(x^{k+1})\|_\infty \le \epsilon_g. \tag{35}$$

Proof. We begin by noting that if the output $\hat{d}^k$ from Algorithm 2 satisfies $\|X_k^{-1}\bar{X}_k\hat{d}^k\|_\infty \ge \beta$ then

$$\epsilon_H \le \beta = \|X_k^{-1}\bar{X}_k d^k\|_\infty \le \|X_k^{-1}\bar{X}_k d^k\| \le \|X_k^{-1}\bar{X}_k\|\|d^k\| \le \|d^k\|,$$

so the claim (34) holds, since $c_d \le 1$. Thus, we assume for the remainder of the proof that $\|X_k^{-1}\bar{X}_k\hat{d}^k\|_\infty < \beta$ and $d^k = \hat{d}^k$, and that $\|d^k\| < c_d\epsilon_H$. We show that the conditions (35) hold in this case.

We start by establishing that $\nabla f(x^{k+1}) \ge -\epsilon_g e$. Since d_type = SOL, we have that $\bar{\zeta}\mu \ge \|\hat{r}^k\|_\infty$ where $\hat{r}^k$ is defined in (28). Using $\|\bar{X}_k X_k^{-2}\bar{X}_k\| \le 1$ and $\epsilon_H\|d^k\| < c_d\epsilon_H^2 = c_d\epsilon_g$, it follows that

$$\begin{aligned}
\bar{\zeta}\mu &\ge \|\left(\bar{X}_k\nabla^2\phi_\mu(x^k)\bar{X}_k + 2\epsilon_H I\right)d^k + \bar{X}_k\nabla\phi_\mu(x^k)\|_\infty \\
&= \|\bar{X}_k\left(\nabla^2 f(x^k)\bar{X}_k d^k + \nabla\phi_\mu(x^k)\right) + \mu\bar{X}_k X_k^{-2}\bar{X}_k d^k + 2\epsilon_H d^k\|_\infty \\
&\ge \|\bar{X}_k\left(\nabla^2 f(x^k)\bar{X}_k d^k + \nabla\phi_\mu(x^k)\right)\|_\infty - \mu\|\bar{X}_k X_k^{-2}\bar{X}_k d^k\|_\infty - 2\epsilon_H\|d^k\|_\infty \\
&\ge \|\bar{X}_k\left(\nabla^2 f(x^k)\bar{X}_k d^k + \nabla\phi_\mu(x^k)\right)\|_\infty - \mu\|\bar{X}_k X_k^{-2}\bar{X}_k\|\|d^k\| - 2\epsilon_H\|d^k\| \\
&\ge \|\bar{X}_k\left(\nabla^2 f(x^k)\bar{X}_k d^k + \nabla\phi_\mu(x^k)\right)\|_\infty - \mu\|d^k\| - 2\epsilon_H\|d^k\| \\
&> \|\bar{X}_k\left(\nabla^2 f(x^k)\bar{X}_k d^k + \nabla f(x^k)\right) - \mu\bar{X}_k X_k^{-1}e\|_\infty - c_d\epsilon_H\mu - 2c_d\epsilon_g. \qquad (36)
\end{aligned}$$

Since $\epsilon_H < 1$ and $\mu = \epsilon_g/4$, we have

$$\bar{\zeta}\mu + c_d\epsilon_H\mu + 2c_d\epsilon_g \le \bar{\zeta}\mu + c_d\mu + 2c_d\epsilon_g = \mu\left(\bar{\zeta} + 9c_d\right).$$

Then, by the definition of c_d, $\bar{\zeta} + 9c_d \le 1$ so that $\bar{\zeta}\mu + c_d\epsilon_H\mu + 2c_d\epsilon_g \le \mu$. Thus, by substituting into (36), we obtain

$$\mu > \|\bar{X}_k\left(\nabla^2 f(x^k)\bar{X}_k d^k + \nabla f(x^k)\right) - \mu\bar{X}_k X_k^{-1}e\|_\infty. \tag{37}$$

By considering each component $i = 1, 2, \ldots, n$ in turn, we now show that

$$\nabla^2 f(x^k)\bar{X}_k d^k + \nabla f(x^k) > -\mu e. \tag{38}$$

When $0 < x_i^k \le 1$, it follows that $\bar{x}_i^k/x_i^k = 1$, so

$$\left|\left[\bar{X}_k\left(\nabla^2 f(x^k)\bar{X}_k d^k + \nabla f(x^k)\right)\right]_i - \mu\right| < \mu,$$

so that

$$\left[\bar{X}_k\left(\nabla^2 f(x^k)\bar{X}_k d^k + \nabla f(x^k)\right)\right]_i > 0,$$

establishing (38) for this component i. When $x_i^k > 1$, we have $\bar{x}_i^k = 1$ and $0 < \bar{x}_i^k/x_i^k < 1$, so from (37), we have

$$\begin{aligned}-\mu < \left[\bar{X}_k\left(\nabla^2 f(x^k)\bar{X}_k d^k + \nabla f(x^k)\right)\right]_i - \frac{\bar{x}_i^k}{x_i^k}\mu &< \left[\bar{X}_k\left(\nabla^2 f(x^k)\bar{X}_k d^k + \nabla f(x^k)\right)\right]_i \\ &= \left[\nabla^2 f(x^k)\bar{X}_k d^k + \nabla f(x^k)\right]_i,\end{aligned}$$

establishing (38) for this component too.

Finally, using (3), $\mu = \epsilon_g/4$, $\|d^k\| < c_d\epsilon_H$, $c_d \le \sqrt{3/(2L_H)}$, and $\epsilon_H^2 = \epsilon_g$, together with $\|\bar{X}_k\| \le 1$, we have from (38) that

$$\begin{aligned}\nabla f(x^{k+1}) &= \nabla f(x^{k+1}) - \nabla^2 f(x^k)\bar{X}_k d^k - \nabla f(x^k) + \nabla^2 f(x^k)\bar{X}_k d^k + \nabla f(x^k) \\ &> -\|\nabla f(x^{k+1}) - \nabla^2 f(x^k)\bar{X}_k d^k - \nabla f(x^k)\|e - \mu e \\ &\ge -\frac{L_H}{2}\|\bar{X}_k\|^2\|d^k\|^2 e - \mu e \\ &> -\left(\frac{L_H}{2}c_d^2 + \frac{1}{4}\right)\epsilon_g e \ge -\epsilon_g e.\end{aligned}$$

We now focus on the second condition, $\|\bar{X}_{k+1}\nabla f(x^{k+1})\| \le \epsilon_g$. To begin, we show that

$$\|\bar{X}_{k+1}\nabla f(x^{k+1})\|_\infty \le 2\|\bar{X}_k\nabla f(x^{k+1})\|_\infty. \tag{39}$$

First, assume that $x_i^k \le 1$ holds. Then, $\bar{x}_i^k = x_i^k$ so that $d_i^k = \left(\bar{x}_i^k/x_i^k\right)d_i^k \le \beta < 1$, so

$$\bar{x}_i^{k+1} \le x_i^{k+1} = x_i^k + \bar{x}_i^k d_i^k = \bar{x}_i^k(1 + d_i^k) < 2\bar{x}_i^k.$$

When $x_i^k > 1$, we have

$$\bar{x}_i^{k+1} \le 1 = \bar{x}_i^k < 2\bar{x}_i^k.$$

Applying these two cases for each coordinate i, we obtain (39). Now, recall from the conditions stated at the start of the proof that $\|X_k^{-1}\bar{X}_k\hat{d}^k\|_\infty < \beta$, so that $d^k = \hat{d}^k$, where $\hat{d}^k$ is the output of Algorithm 2 at iteration k. We thus have for $\hat{r}^k$ defined by

(28) that (27) holds, by Lemma 7. Therefore, by (3), (27), (39), $\|\bar{X}_k X_k^{-1} e\|_\infty \le 1$, and $\|\bar{X}_k\| \le 1$, we have

$$
\begin{aligned}
&\|\bar{X}_{k+1}\nabla f(x^{k+1})\|_\infty \\
&\le 2\|\bar{X}_k \nabla f(x^{k+1})\|_\infty && \text{by (39)} \\
&= 2\|\bar{X}_k \nabla f(x^{k+1}) - \bar{X}_k \nabla f(x^k) + \bar{X}_k \nabla f(x^k)\|_\infty \\
&= 2\|\bar{X}_k \nabla f(x^{k+1}) - \bar{X}_k \nabla f(x^k) - \bar{X}_k \nabla^2 \phi_\mu(x^k)\bar{X}_k d^k \\
&\quad - 2\epsilon_H d^k + \mu \bar{X}_k X_k^{-1} e + \hat{r}^k\|_\infty && \text{by (28)} \\
&\le 2\|\bar{X}_k \left(\nabla f(x^{k+1}) - \nabla f(x^k) - \nabla^2 f(x^k)\bar{X}_k d^k\right)\|_\infty \\
&\quad + 2\mu\|\bar{X}_k X_k^{-2}\bar{X}_k d^k\|_\infty + 4\epsilon_H\|d^k\|_\infty \\
&\quad + 2\mu\|\bar{X}_k X_k^{-1} e\|_\infty + 2\|\hat{r}^k\|_\infty && \text{by definition of } \phi_\mu \\
&\le 2\|\bar{X}_k\|\|\nabla f(x^{k+1}) - \nabla f(x^k) - \nabla^2 f(x^k)\bar{X}_k d^k\| \\
&\quad + 2\mu\|\bar{X}_k X_k^{-2}\bar{X}_k d^k\| + 4\epsilon_H\|d^k\| + 2\mu + 2\|\hat{r}^k\| && \text{since } \|\bar{X}_k X_k^{-1} e\|_\infty \le 1 \\
&\le L_H\|\bar{X}_k d^k\|^2 + 2\mu\|\bar{X}_k X_k^{-2}\bar{X}_k\|\|d^k\| \\
&\quad + 4\epsilon_H\|d^k\| + 2\mu + \zeta_r \epsilon_H\|d^k\| && \text{by (3), (27), and } \|\bar{X}_k\| \le 1 \\
&< L_H c_d^2 \epsilon_g + 2\mu c_d \epsilon_H + 4 c_d \epsilon_g + \epsilon_g/2 + \zeta_r c_d \epsilon_g,
\end{aligned}
$$

where we used $\|\bar{X}_k\| \le 1$, $\|\bar{X}_k^{-1} X_k\| \le 1$, $\|d^k\| < c_d \epsilon_H$, $\epsilon_H^2 = \epsilon_g$, and $\mu = \epsilon_g/4$ for the last inequality. Finally, since $\epsilon_H < 1$, $c_d \le 1$, and $c_d \le 1/\left(2\left(L_H + 9/2 + \zeta_r\right)\right)$, it follows that

$$
\begin{aligned}
\|\bar{X}_{k+1}\nabla f(x^{k+1})\|_\infty &< L_H c_d^2 \epsilon_g + 2\mu c_d \epsilon_H + 4 c_d \epsilon_g + \epsilon_g/2 + \zeta_r c_d \epsilon_g \\
&\le L_H c_d \epsilon_g + 2\mu c_d + 4 c_d \epsilon_g + \epsilon_g/2 + \zeta_r c_d \epsilon_g \\
&\le L_H c_d \epsilon_g + c_d \epsilon_g/2 + 4 c_d \epsilon_g + \epsilon_g/2 + \zeta_r c_d \epsilon_g \\
&\le c_d \epsilon_g \left(L_H + 9/2 + \zeta_r\right) + \epsilon_g/2 \\
&\le \epsilon_g/2 + \epsilon_g/2 = \epsilon_g,
\end{aligned}
$$

completing the proof. □

Lemma 9 is useful in the following line search argument, because we need only consider cases in which $\|d^k\| \ge c_d \epsilon_H$. We now show that a sufficiently long step is taken whenever d_type=SOL and x^{k+1} does not satisfy the approximate first-order conditions (6b) and (6c).

Lemma 10. *Suppose that Assumption 1 holds. Suppose that at iteration k of Algorithm 1, we have either $[\nabla f(x^k)]_i \le -\epsilon_g$ for some coordinate i or $\|\bar{X}_k \nabla f(x^k)\|_\infty \ge \epsilon_g$, so that Algorithm 2 is called. When Algorithm 2 outputs a direction $\hat{d}^k$ with d_type=SOL, then either*

(A) the backtracking line search terminates with $\alpha_k = 1$ and both (6b) and (6c) hold at x^{k+1}, or

(B) the backtracking line search requires at most $j_k \le j_{\rm sol} + 1$ iterations, where

$$
(40) \qquad j_{\rm sol} = \left[\frac{1}{2}\log_\theta\left(\frac{6(1-\beta)^2}{(L_H+\eta)(1-\beta)^2 + (2-\beta)}\ \frac{\epsilon_H^2}{1.1(U_g + \mu\sqrt{n})}\right)\right]_+,
$$

and

$$
(41) \qquad \alpha_k\|d^k\| \ge c_{\rm sol}\epsilon_H,
$$

where

$$c_{\rm sol} = \min\left\{c_d, \frac{6(1-\beta)^2\theta^2}{(L_H+\eta)(1-\beta)^2+(2-\beta)}\right\},$$

and c_d *is defined in (34).*

Proof. This result follows by largely the same argument as that of the proof of [37, Lemma 13]. The main difference is due to the result of Lemma 8 which, together with (4), implies

(42)
$$\phi_\mu(x^k+\theta^j \bar{X}_k d^k) - \phi_\mu(x^k) \le \theta^j g^\top d^k + \frac{\theta^{2j}}{2}(d^k)^\top H d^k + \frac{L_H(1-\beta)^2+(2-\beta)}{6(1-\beta)^2}\theta^{3j}\|d^k\|^3,$$

where the notation $g = \bar{X}_k \nabla\phi_\mu(x^k)$ and $H = \bar{X}_k\nabla^2\phi_\mu(x^k)\bar{X}_k$ is used once more. Replacing the Taylor series expansion around f in the proof of [37, Lemma 13] with this expression yields the result. We provide a full proof in Appendix A.2. ◻

Now we show that a sufficiently long step always occurs when d_type=NC.

LEMMA 11. *Suppose that Assumption 1 holds. Suppose that at iteration* k *of Algorithm 1, we have either* $[\nabla f(x^k)]_i \le -\epsilon_g$ *for some coordinate* i *or* $\|\bar{X}_k \nabla f(x^k)\|_\infty \ge \epsilon_g$*, so that Algorithm 2 is called. When Algorithm 2 outputs a direction* $\hat{d}^k$ *with d_type=NC, then the backtracking line search requires at most* $j_k \le j_{\rm nc}+1$ *iterations, where*

$$j_{\rm nc} \;=\; \left[\log_\theta\left(\frac{3(1-\beta)^2}{(L_H+\eta)(1-\beta)^2+(2-\beta)}\right)\right]_+, \tag{43}$$

and

$$\alpha_k\|d^k\| \ge c_{\rm nc}\epsilon_H, \tag{44}$$

where

$$c_{\rm nc} = \min\left\{1, \frac{3(1-\beta)^2\theta}{(L_H+\eta)(1-\beta)^2+(2-\beta)}\right\}.$$

Proof. This result follows from the same argument as the proof of [37, Lemma 1]. The main difference in the proof once again revolves around the use of (42) in place of the Taylor expansion around f. A full proof is provided in Appendix A.3. ◻

Next, we bound the maximum decrease in the logarithmic terms over the iterations of Algorithm 1.

LEMMA 12. *Let* ω *be such that* $\|x^0\|_\infty \le \omega$*. Then for any* $k \ge 0$*, we have*

$$\sum_{i=1}^n \left(-\log x_i^{k+1} + \log x_i^0\right) \ge -n\left(\log\omega - \min_i \log x_i^0\right) - \frac{\sqrt{n}}{\omega}\sum_{j=0}^k \alpha_j\|d^j\|. \tag{45}$$

Proof. We focus on a single coordinate i, and show that the following holds for any $k \ge 0$:

$$-\log x_i^{k+1} + \log x_i^0 \ge -\log\omega + \log x_i^0 - \frac{1}{\omega}\sum_{j=0}^k \alpha_j|d_i^j|. \tag{46}$$

We consider three cases.

1: $x_i^{k+1} \le \omega$. Here we have $-\log x_i^{k+1} \ge -\log \omega$, so (46) is satisfied trivially.
2: $x_i^{k+1} > \omega$ and $x_i^k \le \omega$. Here, we have

$$\begin{aligned} -\log x_i^{k+1} = -\log\left(x_i^k + \alpha_k \bar{x}_i^k d_i^k\right) &\ge -\log\left(\omega + \alpha_k \bar{x}_i^k d_i^k\right) \\ &= -\log\left(\omega\left(1 + \frac{1}{\omega}\alpha_k \bar{x}_i^k d_i^k\right)\right) \\ &= -\log \omega - \log\left(1 + \frac{1}{\omega}\alpha_k \bar{x}_i^k d_i^k\right) \\ &\ge -\log \omega - \frac{1}{\omega}\alpha_k \bar{x}_i^k d_i^k \\ &\ge -\log \omega - \frac{1}{\omega}\alpha_k |d_i^k|, \end{aligned}$$

where the second to last inequality follows by $\log(1+x) \le x$ and the last by $\bar{x}_i^k \le 1$. Therefore, we have

$$-\log(x_i^{k+1}) + \log(x_i^0) \ge -\log(\omega) + \log(x_i^0) - \frac{1}{\omega}\alpha_k |d_i^k|, \tag{47}$$

so (46) is satisfied again.
3: $x_i^{k+1} > \omega$ and $x_i^k > \omega$. For this case, we have

$$\begin{aligned} -\log(x_i^{k+1}) = -\log\left(x_i^k + \alpha_k \bar{x}_i^k d_i^k\right) &= -\log\left(x_i^k\left(1 + \alpha_k \frac{\bar{x}_i^k}{x_i^k} d_i^k\right)\right) \\ &= -\log(x_i^k) - \log\left(1 + \alpha_k \frac{\bar{x}_i^k}{x_i^k} d_i^k\right) \\ &\ge -\log(x_i^k) - \alpha_k \frac{\bar{x}_i^k}{x_i^k} d_i^k \\ &\ge -\log(x_i^k) - \frac{1}{\omega}\alpha_k |d_i^k|, \end{aligned} \tag{48}$$

where the second to last inequality follows by $\log(1+x) \le x$ and the last by $\bar{x}_i^k \le 1$ and $x_i^k \ge \omega$. We define $\bar{k}$ to be the smallest index such that $x_i^j > \omega$ for all $j = \bar{k}, \bar{k}+1, \ldots, k+1$. We have that $\bar{k}$ exists, and lies in the range $\{1, 2, \ldots, k\}$. Moreover, we have that

$$x_i^{\bar{k}} > \omega, \quad x_i^{\bar{k}-1} \le \omega. \tag{49}$$

Since (48) holds when k is replaced by any $j = \bar{k}, \ldots, k$, we have

$$-\log x_i^{k+1} + \log x_i^{\bar{k}} = \sum_{j=\bar{k}}^{k}\left(-\log x_i^{j+1} + \log x_i^j\right) \ge -\frac{1}{\omega}\sum_{j=\bar{k}}^{k}\alpha_j |d_i^j|. \tag{50}$$

Since $\bar{k}-1$ is in Case 2, because of (49), we have

$$-\log x_i^{\bar{k}} \ge -\log \omega - \frac{1}{\omega}\alpha_{\bar{k}-1}|d_i^{\bar{k}-1}|.$$

By adding this expression to (50), and adding $\log x_i^0$ to both sides, we obtain

$$-\log x_i^{k+1} + \log x_i^0 \ge -\log \omega + \log x_i^0 - \frac{1}{\omega}\sum_{j=\bar{k}-1}^{k}\alpha_j |d_i^j|,$$

which implies (46).

By summing (46) over all coordinates i, we obtain

$$\begin{aligned}\sum_{i=1}^{n}\left(-\log(x_i^{k+1})+\log(x_i^0)\right) &\geq -\sum_{i=1}^{n}\left(\log(\omega)-\log(x_i^0)\right)-\frac{1}{\omega}\sum_{j=0}^{k}\sum_{i=1}^{n}\alpha_j|d_i^j| \\ &= -\sum_{i=1}^{n}\left(\log(\omega)-\log(x_i^0)\right)-\frac{1}{\omega}\sum_{j=0}^{k}\alpha_j\|d^j\|_1 \\ &\geq -n\left(\log(\omega)-\min_i \log(x_i^0)\right)-\frac{\sqrt{n}}{\omega}\sum_{j=0}^{k}\alpha_j\|d^j\|,\end{aligned}$$

which proves the result. ∎

Now we are ready to bound the maximum number of iterations of Algorithm 1 that can occur before the approximate first-order optimality conditions (6a), (6b), and (6c) are satisfied.

THEOREM 13. *Let Assumptions 1 and 2 hold. Then, some iterate x^k generated by Algorithm 1, where $k = 0, 1, \dots, \bar{K}_1 + 1$ and*

$$\begin{aligned}\bar{K}_1 &:= \left\lceil \frac{12\left(\mu n\left(\log(\omega_1)-\min_i \log(x_i^0)\right)+f(x^0)-f_{\text{low}}\right)}{\eta c_{\text{all}}^3}\epsilon_g^{-3/2}\right\rceil, \\ \omega_1 &:= \max\left\{\frac{3\sqrt{n}}{\eta c_{\text{all}}^2}, \|x^0\|_\infty\right\}, \\ c_{\text{all}} &:= \min\{c_{\text{sol}}, c_{\text{nc}}\},\end{aligned}$$

will satisfy the conditions

$$\nabla f(x^k) \geq -\epsilon_g e, \quad \|\bar{X}_k \nabla f(x^k)\|_\infty \leq \epsilon_g. \tag{51}$$

Proof. Suppose for contradiction that at least one of the conditions in (51) is violated for all $k = 0, 1, \dots, \bar{K}_1 + 1$, so that case A of Lemma 10 does not occur for all $k = 0, 1, \dots, \bar{K}_1$. Algorithm 2 will be invoked at each of the first $\bar{K}_1 + 1$ iterates of Algorithm 1. For each iteration $l = 0, 1, \dots, \bar{K}_1$ for which Algorithm 2 returns d_type=SOL, we have from Lemma 10, and the fact that case A does not occur, that $\alpha_k\|d^k\| \geq c_{\text{sol}}\epsilon_H$. For each iteration $l = 0, 1, \dots, \bar{K}_1$ for which Algorithm 2 returns d_type=NC, we have by Lemma 11 that $\alpha_k\|d^k\| \geq c_{\text{nc}}\epsilon_H$. Thus, for either type of step, we have

$$\alpha_k\|d^k\| \geq \min\{c_{\text{sol}}, c_{\text{nc}}\}\epsilon_H = c_{\text{all}}\epsilon_H. \tag{52}$$

Now, by (13), we have

$$-\frac{\eta}{6}\alpha_k^3\|d^k\|^3 \geq \phi_\mu(x^{k+1})-\phi_\mu(x^k) = f(x^{k+1})-f(x^k)+\mu\sum_{i=1}^{n}\left(-\log(x_i^{k+1})+\log(x_i^k)\right).$$

By summing this bound over $k = 0, 1, \dots, \bar{K}_1$, and telescoping both terms on the right-hand size, we obtain

$$-\frac{\eta}{6}\sum_{k=0}^{\bar{K}_1}\alpha_k^3\|d^k\|^3 \geq f(x^{\bar{K}_1+1})-f(x^0)+\mu\sum_{i=1}^{n}\left(-\log(x_i^{\bar{K}_1+1})+\log(x_i^0)\right).$$

By applying Lemma 12 with $\omega = \omega_1$, we have

(53)

$$-\frac{\eta}{6}\sum_{k=0}^{\bar{K}_1}\alpha_k^3\|d^k\|^3 \geq f(x^{\bar{K}_1+1}) - f(x^0) - \mu n\left(\log(\omega_1) - \min_i \log(x_i^0)\right) - \mu\frac{\sqrt{n}}{\omega_1}\sum_{k=0}^{\bar{K}_1}\alpha_k\|d^k\|.$$

From the definition of ω_1, we obtain

$$-\mu\frac{\sqrt{n}}{\omega_1}\sum_{k=0}^{\bar{K}_1}\alpha_k\|d^k\| \geq -\frac{\mu\eta c_{\text{all}}^2}{3}\sum_{k=0}^{\bar{K}_1}\alpha_k\|d^k\| = -\frac{\eta c_{\text{all}}^2\epsilon_H^2}{12}\sum_{k=0}^{\bar{K}_1}\alpha_k\|d^k\|,$$

where the final equality is due to $\mu = \epsilon_g/4 = \epsilon_H^2/4$. It follows that

$$\begin{aligned}
\frac{\eta}{6}\sum_{k=0}^{\bar{K}_1}\alpha_k^3\|d^k\|^3 - \mu\frac{\sqrt{n}}{\omega_1}\sum_{k=0}^{\bar{K}_1}\alpha_k\|d^k\| &\geq \frac{\eta}{6}\sum_{k=0}^{\bar{K}_1}\alpha_k\|d^k\|\left(\alpha_k^2\|d^k\|^2 - \frac{c_{\text{all}}^2\epsilon_H^2}{2}\right)\\
&\geq \frac{\eta}{12}\sum_{k=0}^{\bar{K}_1}\alpha_k\|d^k\|c_{\text{all}}^2\epsilon_H^2\\
&\geq \frac{\eta}{12}\sum_{k=0}^{\bar{K}_1}c_{\text{all}}^3\epsilon_H^3\\
&= \frac{\eta}{12}\left(\bar{K}_1+1\right)c_{\text{all}}^3\epsilon_H^3,
\end{aligned}$$

where the second and third inequalities follow by (52). By combining this inequality with (53), we have

$$\begin{aligned}
&f(x^0) - f(x^{\bar{K}_1+1}) + \mu n\left(\log(\omega_1) - \min_i \log(x_i^0)\right)\\
&\qquad \geq \left(\bar{K}_1+1\right)\frac{\eta}{12}\epsilon_H^3 c_{\text{all}}^3\\
&\qquad > \mu n\left(\log(\omega_1) - \min_i \log(x_i^0)\right) + f(x^0) - f_{\text{low}},
\end{aligned}$$

where we used the definition of $\bar{K}_1$ and $\epsilon_H = \epsilon_g^{1/2}$ for the final inequality. This inequality contradicts the definition of f_{low} (in Assumption 2), so our claim is proved. ∎

Recalling that the workload of Algorithm 2 in terms of Hessian-vector products depends on the index J defined in Lemma 5, we obtain the following corollary. (Note the mild assumption on the value of M used at each instance of Algorithm 2, which is satisfied provided that this algorithm is always invoked with an initial estimate of M in the range $[0, U_H + \mu]$.)

Corollary 14. *Suppose that Assumptions 1, 2, and 3 hold, and let $\bar{K}_1$ be defined as in Theorem 13 and $J(M, \epsilon_H, \zeta_r, c_\mu)$ be as defined in Lemma 5. Suppose that the values of M used or calculated at each instance of Algorithm 2 satisfy $M \leq U_H + \mu$. Then the number of Hessian-vector products and/or gradient evaluations required by Algorithm 1 to output an iterate satisfying (51) is at most*

$$(2\min\{n, J(U_H+\mu, \epsilon_H, \zeta_r, c_\mu)\} + 2)\,(\bar{K}_1 + 1). \tag{54}$$

If $J(U_H + \mu, \epsilon_H, \zeta_r, c_\mu) < n$, this bound is

$$\tilde{\mathcal{O}}(\epsilon_g^{-7/4} + n\epsilon_g^{-3/4}), \tag{55}$$

while if $J(U_H + \mu, \epsilon_H, \zeta_r, c_\mu) \geq n$*, it is*

$$\tilde{\mathcal{O}}(n\epsilon_g^{-3/2}). \tag{56}$$

Proof. From Lemma 5, the number of Hessian-vector multiplications in the main loop of Algorithm 2 is bounded by $\min\{n, J(U_H, \epsilon_H, \zeta_r, c_\mu)\} + 1$. An additional $\min\{n, J(U_H, \epsilon_H, \zeta_r, c_\mu)\}$ Hessian-vector products may be needed to return a direction satisfying (16), if Algorithm 2 does not store its iterates y_j. Each iteration also requires a single evaluation of the gradient ∇f, giving a bound of $(2\min\{n, J(U_H, \epsilon_H, \zeta_r, c_\mu)\} + 2)$ on the workload per iteration of Algorithm 1. Per Theorem 13, we obtain the result (54) by multiplying this quantity by $\bar{K}_1 + 1$.

To obtain the estimate (55), we note from $\mu = \epsilon_g/4$ that

$$\bar{K}_1 = \tilde{\mathcal{O}}(n\epsilon_g^{-1/2} + \epsilon_g^{-3/2}),$$

while from (22) and (23), using $\epsilon = \epsilon_H = \epsilon_g^{1/2}$, we have for $J(U_H + \mu, \epsilon_H, \zeta_r, c_\mu) < n$ that

$$J(U_H + \mu, \epsilon_H, \zeta_r, c_\mu) = \tilde{\mathcal{O}}(\epsilon_H^{-1/2}) = \tilde{\mathcal{O}}(\epsilon_g^{-1/4}).$$

We obtain (55) by substituting these estimates into (54). For (56), we have from $J(U_H + \mu, \epsilon_H, \zeta_r, c_\mu) \geq n$ together with (22) and (23) that $n \leq \tilde{\mathcal{O}}\left(\epsilon_g^{-1/4}\right)$. Therefore, computational complexity is bounded by

$$\tilde{\mathcal{O}}(n(n\epsilon_g^{-1/2} + \epsilon_g^{-3/2})) \leq \tilde{\mathcal{O}}(n(\epsilon_g^{-3/4} + \epsilon_g^{-3/2})) = \tilde{\mathcal{O}}(n\epsilon_g^{-3/2}),$$

as claimed ∎

5.3 Second-Order Complexity Analysis We now find bounds on iteration and computational complexity of finding a point that satisfies all of the approximate optimality conditions in (6). In this section, as well as using results from Sections 5.1 and 5.2, we need to use the properties of the minimum eigenvalue oracle, Procedure 3. To this end, we make the following generic assumption.

ASSUMPTION 4. *For every iteration k at which Algorithm 1 calls Procedure 3, and for a specified failure probability δ with $0 \leq \delta \ll 1$, Procedure 3 either certifies that $\bar{X}_k \nabla^2 f(x_k) \bar{X}_k \succeq -\epsilon_H I$ or finds a vector of curvature smaller than $-\epsilon_H/2$ in at most*

$$N_{\text{meo}} := \min\left\{n, 1 + \left\lceil \mathcal{C}_{\text{meo}} \epsilon_H^{-1/2} \right\rceil\right\} \tag{57}$$

Hessian-vector products, with probability $1 - \delta$, where $\mathcal{C}_{\text{meo}}$ depends at most logarithmically on δ and ϵ_H.

Assumption 4 encompasses the strategies we mentioned in Section 4.2. Assuming the bound U_H on $\|H\|$ is available, for both the Lanczos method with a random starting vector and the conjugate gradient algorithm with a random right-hand side, (57) holds with $\mathcal{C}_{\text{meo}} = \ln(2.75n/\delta^2)\sqrt{U_H}/2$. When a bound on $\|H\|$ is not available in advance, it can be estimated efficiently with minimal effect on the complexity bounds; see Appendix B.3 of [36].

The next lemma guarantees termination of the backtracking line search for a negative curvature direction. As for Lemma 10, the result is deterministic.

LEMMA 15. *Suppose that Assumptions 1 and 4 hold. Suppose that at iteration k of Algorithm 1, the search direction d^k is of negative curvature type, obtained either directly from Procedure 3 or as the output of Algorithm 2 with d_type=NC. Then the backtracking line search terminates with step length $\alpha_k = \theta^{j_k}$ with $j_k \le j_{\rm nc}+1$, where $j_{\rm nc}$ is defined as in Lemma 11, and the decrease in the function value resulting from the chosen step length satisfies*

$$\alpha_k \|d^k\| \ge \frac{1}{4} c_{\rm nc}\epsilon_H, \tag{58}$$

with $c_{\rm nc}$ is defined in Lemma 11.

Proof. Lemma 11 shows that the claim holds (with a factor of 1/4 to spare) when the direction of negative curvature is obtained from Algorithm 2. When the direction v is obtained from Procedure 3, we have by $\|v\|=1$ that

$$v^\top \bar{X}_k \nabla^2 f(x^k) \bar{X}_k v \le -\frac{1}{2}\epsilon_H.$$

Then, since $v^\top \bar{X}_k X_k^{-2} \bar{X}_k v \le 1$, we have

$$\begin{aligned} v^\top \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k v &= v^\top \bar{X}_k \nabla^2 f(x^k) \bar{X}_k v + \mu v^\top \bar{X}_k X_k^{-2} \bar{X}_k v \\ &\le -\frac{1}{2}\epsilon_H + \mu \le -\frac{1}{4}\epsilon_H, \end{aligned} \tag{59}$$

where the last inequality follows from $\mu = \epsilon_g/4 = \epsilon_H^2/4$ and $\epsilon_H < 1$. Now, when

$$\min\left\{ |v^\top \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k v|, \frac{\beta}{\|X_k^{-1}\bar{X}_k v\|_\infty} \right\} = |v^\top \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k v|,$$

we have $\|d^k\| = |v^\top \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k v| \ge \epsilon_H/4$. Otherwise, we have

$$\beta = \|X_k^{-1}\bar{X}_k d^k\|_\infty \le \|X_k^{-1}\bar{X}_k d^k\| \le \|X_k^{-1}\bar{X}_k\|\|d^k\| \le \|d^k\|.$$

By combining the two cases, and using $\beta \ge \epsilon_H$, we have

$$\|d^k\| \ge \min\left\{\frac{1}{4}\epsilon_H, \beta\right\} = \frac{1}{4}\epsilon_H.$$

Finally, we note that in either case, we have

$$\|d^k\| \le -v^\top \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k v = -\frac{(d^k)^\top \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k d^k}{\|d^k\|^2}.$$

Therefore, we have

$$\frac{(d^k)^\top \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k d^k}{\|d^k\|^2} \le -\|d^k\| \le -\frac{1}{4}\epsilon_H.$$

The result can now be obtained by following the proof of Lemma 11, with $\frac{1}{4}\epsilon_H$ replacing ϵ_H. ∎

We are now ready to state our iteration complexity result for Algorithm 1.

THEOREM 16. *Suppose that Assumptions 1, 2, and 4 hold and define*

$$\omega_2 := \max\left\{\frac{96\sqrt{n}}{\eta c_{\mathrm{all}}^2},\ \|x^0\|_\infty\right\}, \tag{60}$$

and

$$\begin{aligned}\bar{K}_2 &:= \left\lceil \frac{1536\left(f(x^0) - f_{\mathrm{low}} + \mu n\left(\log(\omega_2) - \min_i \log(x_i^0)\right)\right)}{\eta c_{\mathrm{all}}^3}\epsilon_g^{-3/2}\right\rceil + 2 \\ &= \tilde{\mathcal{O}}(n\epsilon_g^{-1/2} + \epsilon_g^{-3/2}),\end{aligned} \tag{61}$$

where the constant c_{all} is defined in Theorem 13. Then with probability at least $(1-\delta)^{\bar{K}_2}$, Algorithm 1 terminates at a point satisfying (6) in at most $\bar{K}_2$ iterations. (With probability at most $1-(1-\delta)^{\bar{K}_2}$, it terminates incorrectly within $\bar{K}_2$ iterations at a point for which (6a), (6b), and (6c) hold but (6d) does not.)

Proof. Algorithm 1 terminates incorrectly with probability δ at any iteration at which Procedure 3 is called, when Procedure 3 certifies erroneously that $\lambda_{\min}(\bar{X}_k\nabla^2 f(x^k)\bar{X}_k) \geq -\epsilon_H$. Such an erroneous certificate only leads to termination. Therefore, an erroneous certificate at iteration k means that Procedure 3 did not produce an erroneous certificate at iterations 0 to $k-1$. By a disjunction argument, we have that the overall probability of terminating with an erroneous certificate during the first $\bar{K}_2$ iterations is bounded by $1-(1-\delta)^{\bar{K}_2}$. Therefore, with probability at least $(1-\delta)^{\bar{K}_2}$, no incorrect termination occurs in the first $\bar{K}_2$ iterations.

Suppose now for contradiction that Algorithm 1 runs for $\bar{K}_2$ iterations without terminating. That is, for all $l = 0, 1, \ldots, \bar{K}_2$, we have at least one of: $[\nabla f(x^l)]_i < -\epsilon_g$ for some coordinate i, $\|\bar{X}_l\nabla f(x^l)\|_\infty > \epsilon_g$, or $\lambda_{\min}(\bar{X}_l\nabla^2 f(x^l)\bar{X}_l) < -\epsilon_H$. Consider the following partition of the set of iteration indices:

$$\mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_3 = \{0, 1, \ldots, \bar{K}_2 - 1\}, \tag{62}$$

where $\mathcal{K}_1$, $\mathcal{K}_2$, and $\mathcal{K}_3$ are defined as follows.

Case 1: $\mathcal{K}_1 := \{l = 0, 1, \ldots, \bar{K}_2 - 1 : \nabla f(x^l) \geq -\epsilon_g e \text{ and } \|\bar{X}_l\nabla f(x^l)\|_\infty \leq \epsilon_g\}$.

Case 2: $\mathcal{K}_2 := \{l = 0, 1, \ldots, \bar{K}_2 - 1 : [\nabla f(x^l)]_i < -\epsilon_g$ for some coordinate i and/or $\|\bar{X}_l\nabla f(x^l)\|_\infty > \epsilon_g$ and $\alpha_l\|d^l\| \geq (c_{\mathrm{all}}/4)\epsilon_H\}$.

Case 3: $\mathcal{K}_3 := \{l = 0, 1, \ldots, \bar{K}_2 - 1 : [\nabla f(x^l)]_i < -\epsilon_g$ for some coordinate i and/or $\|\bar{X}_l\nabla f(x^l)\|_\infty > \epsilon_g$ and $\alpha_l\|d^l\| < (c_{\mathrm{all}}/4)\epsilon_H\}$.

Then, for all $l \in \mathcal{K}_1 \cup \mathcal{K}_2$, the fact that the algorithm does not satisfy (6) at iteration $l+1$ together with Lemmas 10, 11, and 15 guarantee that

$$\alpha_l\|d^l\| \geq \min\{c_{\mathrm{sol}}, c_{\mathrm{nc}}/4\}\epsilon_H \geq (c_{\mathrm{all}}/4)\epsilon_H. \tag{63}$$

On the other hand, for $l \in \mathcal{K}_3$, case A of Lemma 10 must have occured. Therefore, for any $l \in \mathcal{K}_3$, we must have $\nabla f(x^{l+1}) \geq -\epsilon_g e$ and $\|\bar{X}_{l+1}\nabla f(x^{l+1})\|_\infty \leq \epsilon_g$, so that $l+1 \in \mathcal{K}_1$ for $l < \bar{K}_2 - 1$. Thus, a sufficiently long step will be taken at the *next* iteration, and we have

$$|\mathcal{K}_3| \leq |\mathcal{K}_1| + 1 \leq |\mathcal{K}_1| + |\mathcal{K}_2| + 1. \tag{64}$$

Now, by a similar argument to Theorem 13 that led to (53), we have

$$-\frac{\eta}{6}\sum_{j=0}^{\bar{K}_2-1}\alpha_j^3\|d^j\|^3 \geq f(x^{\bar{K}_2}) - f(x^0) - \mu\frac{\sqrt{n}}{\omega_2}\sum_{l=0}^{\bar{K}_2-1}\alpha_l\|d^l\| - \mu n\left(\log(\omega_2) - \min_i \log(x_i^0)\right). \tag{65}$$

Using the definition of ω_2, we have

$$-\mu\frac{\sqrt{n}}{\omega_2}\sum_{l=0}^{\bar{K}_2-1}\alpha_l\|d^l\| \geq -\frac{\mu\eta c_{\text{all}}^2}{96}\sum_{l=0}^{\bar{K}_2-1}\alpha_l\|d^l\| = -\frac{\eta c_{\text{all}}^2\epsilon_H^2}{384}\sum_{l=0}^{\bar{K}_2-1}\alpha_l\|d^l\|,$$

where the second equality is due to $\mu = \epsilon_g/4 = \epsilon_H^2/4$. Therefore, we have

$$\begin{aligned}
&\frac{\eta}{6}\sum_{l=0}^{\bar{K}_2-1}\alpha_l^3\|d^l\|^3 - \mu\frac{\sqrt{n}}{\omega_2}\sum_{l=0}^{\bar{K}_2-1}\alpha_l\|d^l\| \\
&\geq \frac{\eta}{6}\sum_{l=0}^{\bar{K}_2-1}\left(\alpha_l^3\|d^l\|^3 - \frac{c_{\text{all}}^2\epsilon_H^2}{64}\alpha_l\|d^l\|\right) \\
&= \frac{\eta}{6}\sum_{j\in\mathcal{K}_1\cup\mathcal{K}_2}\alpha_j\|d^j\|\left(\alpha_j^2\|d^j\|^2 - \frac{c_{\text{all}}^2\epsilon_H^2}{64}\right) + \frac{\eta}{6}\sum_{l\in\mathcal{K}_3}\left(\alpha_l^3\|d^l\|^3 - \frac{c_{\text{all}}^2\epsilon_H^2}{64}\alpha_l\|d^l\|\right) \\
&\geq \frac{3\eta}{384}\sum_{j\in\mathcal{K}_1\cup\mathcal{K}_2}\alpha_j\|d^j\|c_{\text{all}}^2\epsilon_H^2 - \frac{\eta}{1536}\sum_{l\in\mathcal{K}_3}c_{\text{all}}^3\epsilon_H^3 \\
&\geq (|\mathcal{K}_1|+|\mathcal{K}_2|)\frac{3\eta}{1536}c_{\text{all}}^3\epsilon_H^3 - (|\mathcal{K}_1|+|\mathcal{K}_2|+1)\frac{\eta}{1536}c_{\text{all}}^3\epsilon_H^3 \\
&\geq \left(|\mathcal{K}_1|+|\mathcal{K}_2|-\frac{1}{2}\right)\frac{\eta}{768}c_{\text{all}}^3\epsilon_H^3,
\end{aligned}$$

where the second inequality follows by (63) and the definition of $\mathcal{K}_3$, while the third inequalities follows by (63) and (64).

Thus, this inequality, (65) and $|\mathcal{K}_1|+|\mathcal{K}_2|+|\mathcal{K}_3|-2 \leq 2(|\mathcal{K}_1|+|\mathcal{K}_2|-1/2)$, imply

$$\begin{aligned}
&f(x^0) - f(x^{\bar{K}_2}) + \mu n\left(\log(\omega_2) - \min_i \log(x_i^0)\right) \\
&\geq (|\mathcal{K}_1|+|\mathcal{K}_2|-1/2)\frac{\eta}{768}c_{\text{all}}^3\epsilon_H^3 \\
&\geq (|\mathcal{K}_1|+|\mathcal{K}_2|+|\mathcal{K}_3|-2)\frac{\eta}{1536}c_{\text{all}}^3\epsilon_H^3 \\
&\geq \left(\bar{K}_2-1\right)\frac{\eta}{1536}c_{\text{all}}^3\epsilon_H^3 \\
&> f(x^0) - f_{\text{low}} + \mu n\left(\log(\omega_2) - \min_i \log(x_i^0)\right).
\end{aligned}$$

where the final inequality follows from the definition of $\bar{K}_2$ and $\epsilon_H = \epsilon_g^{1/2}$. The final inequality implies that $f_{\text{low}} > f(x^{\bar{K}_2})$, which contradicts the definition of f_{low}, proving the claim.

The estimate $\bar{K}_2 = \tilde{\mathcal{O}}(n\epsilon_g^{-1/2} + \epsilon_g^{-3/2})$ follows directly from $\mu = \epsilon_g/4$. $\square$

Finally, we provide a computational complexity result, a bound on the number of Hessian-vector products and gradient evaluations necessary for Algorithm 1 to find a point that satisfies (6).

COROLLARY 17. *Suppose that Assumptions 1, 2, 3, and 4 hold, and let $\bar{K}_2$ be defined as in (61). Suppose that the values of M used or calculated at each instance of Algorithm 2 satisfy $M \leq U_H + \mu$. Then with probability at least $(1-\delta)^{\bar{K}_2}$, Algorithm 1 terminates at a point satisfying (6) after at most*

$$(\max\{2\min\{n, J(U_H+\mu, \epsilon_H, \zeta_r, c_\mu)\}+2, N_{\text{meo}}\})\,\bar{K}_2 \tag{66}$$

Hessian-vector products and/or gradient evaluations. (With probability at most $1 - (1-\delta)^{\bar{K}_2}$*, it terminates incorrectly with this complexity at a point for which (6a), (6b), and (6c) hold but (6d) does not.)*

Proof. The proof follows by combining Theorem 16 (which bounds the number of iterations) with Lemma 5 and Assumption 4 (which bound the workload per iteration). ∎

For large n, the operation bound (66) is $\tilde{\mathcal{O}}(\epsilon_g^{-7/4} + n\epsilon_g^{-3/4})$, because the multiplier of $\bar{K}_2$ in (66) is $\tilde{\mathcal{O}}(\epsilon_g^{-1/4})$ while $\bar{K}_2$ is $\tilde{\mathcal{O}}(n\epsilon_g^{-1/2} + \epsilon_g^{-3/2})$. For small n, the multiplier of $\bar{K}_2$ in (66) is $\mathcal{O}(n)$, and the dominant term in $\bar{K}_2$ is $\epsilon_g^{-3/2}$, leading to a computational complexity bound of $\tilde{\mathcal{O}}(n\epsilon_g^{-3/2})$ for this case.

These computational complexity bounds are the same as those obtained for *unconstrained* smooth minimization discussed in Section 2, except for the inclusion of the $n\epsilon_g^{-3/4}$ term for the case of large n. In the latter case, our algorithm acheives a superior worst-case computational complexity bound to that of [30], whose worst-case computational complexity appear to be $\mathcal{O}(n\epsilon_g^{-3/2})$. The $n\epsilon_g^{-3/4}$ term is a consequence of using the log-barrier term to monitor descent. It may be avoided by making an additional assumption that f grows rapidly enough to overcome the improvement in the logarithmic term of ϕ_μ, as x moves away from the solution set for (1) and becomes large. Indeed, we made such an assumption in an earlier version of the paper. It makes the analysis somewhat more straightforward in that it allows us assume that the iterates $\{x^k\}$ are bounded. However, prompted by a referee's comment and a desire for generality, we have dropped this assumption in the current version.

6 Discussion

We have presented a log-barrier Newton-CG algorithm which combines recent advances in complexity of algorithms for large-scale unconstrained optimization with results on the primal log-barrier function for bound constraints. Our algorithm uses the Capped CG method of [36] to compute Newton-type steps for the log-barrier function, while monitoring convexity during the CG iterations to detect possible directions of negative curvature. Once the algorithm has found a point satisfying the first-order optimality conditions, a Minimum Eigenvalue Oracle is used to find a direction of negative curvature for the scaled Hessian matrix or to certify (with high probability) that the second-order optimality conditions hold at the current iterate. Both types of steps can be computed using efficient iterative solvers, enabling good overall computational complexity results. The resulting method finds a point satisfying (6) in at most $\mathcal{O}(\epsilon_g^{-3/2} + n\epsilon_g^{-1/2})$ iterations, with at most $\tilde{\mathcal{O}}(n\epsilon_g^{-3/2})$ gradient evaluations and/or Hessian vector products when n is small and at most $\tilde{\mathcal{O}}(\epsilon_g^{-7/4} + n\epsilon_g^{-3/4})$ gradient evaluations and/or Hessian vector products for n sufficiently large. This overall computational complexity compares favorably with the worst-case bounds of recently proposed methods.

There are a number of ways to align our algorithm more closely with the interior-point methods in common use. One possible extension is to embed this method in a primal-dual interior-point framework, which is more widely used than the primal log-barrier framework. A second is to extend the log-barrier approach to minimize ϕ_μ for a decreasing positive sequence of values of μ, rather than the "one-shot" approach using a small fixed value of μ that we describe in this paper. Finally, generalizations of our approach to problems with more complex constraint sets, such as problems with general linear constraints, remains an open problem.

Acknowledgement We thank Clément Royer for his valuable advice and many suggestions during the preparation of this manuscript. We are also grateful for the very helpful comments of the associate editor and two referees on an earlier version.

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 (STOC 2017), PMLR, 2017.

[2] R. Andreani, G. Haeser, A. Ramos, and P. Silva, *A second-order sequential optimality condition associated to the convergence of optimization algorithms*, IMA Journal of Numerical Analysis, 37 (2017), pp. 1902–1929.

[3] W. Bian, X. Chen, and Y. Ye, *Complexity analysis of interior point algorithms for non-lipschitz and nonconvex minimization*, Mathematical Programming, 149 (2015), pp. 301–327.

[4] E. G. Birgin, J. L. 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 Journal on Optimization, 26 (2016), pp. 951–967.

[5] E. G. Birgin and J. M. Martínez, *The use of quadratic regularization with a cubic descent condition for unconstrained optimization*, SIAM J. Optim., 27 (2017), pp. 1049–1074.

[6] ———, *On regularization and active-set methods with complexity for constrained optimization*, SIAM Journal on Optimization, 28 (2018), pp. 1367–1395.

[7] ———, *Complexity and performance of an augmented lagrangian algorithm*, arXiv preprint arXiv:1907.02401, (2019).

[8] ———, *A newton-like method with mixed factorizations and cubic regularization for unconstrained minimization*, Computational Optimization and Applications, 73 (2019), pp. 707–753.

[9] Y. Carmon, J. Duchi, O. Hinder, and A. Sidford, *Lower bounds for finding stationary points i*, arXiv preprint arXiv:1710.11606, (2017).

[10] 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 Volume 70: International Conference on Machine Learning, 6-11 August 2017, International Convention Centre, Sydney, Australia, PMLR, 2017, pp. 654–663.

[11] ———, *Accelerated methods for non-convex optimization*, SIAM J. Optim., 28 (2018), pp. 1751–1772.

[12] 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*, Journal of Complexity, 53 (2019), pp. 68–94.

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

[14] ———, *Optimal Newton-type methods for nonconvex optimization*, Tech. Rep. naXys-17-2011, Dept of Mathematics, FUNDP, Namur (B), 2011.

[15] ———, *An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity*, IMA Journal of Numerical Analysis, 32 (2012), pp. 1662–1695.

[16] ———, *Complexity bounds for second-order optimality in unconstrained optimization*, Journal of Complexity, 28 (2012), pp. 93–108.

[17] ———, *On the evaluation complexity of constrained nonlinear least-squares and general constrained nonlinear optimization using second-order methods*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 836–851.

[18] ———, *Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints*, arXiv preprint arXiv:1811.01220, (2018).

[19] ———, *Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization*, in Proceedings of the 2018 International Conference of Mathematics, 2018.

[20] ———, *Universal regularization methods: varying the power, the smoothness and the accuracy*, SIAM Journal on Optimization, 29 (2019), pp. 595–615.

[21] X. Chen, F. Xu, and Y. Ye, *Lower bound theory of nonzero entries in solutions of ℓ_2-ℓ_p minimization*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2832–2852.

[22] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, MPS-SIAM Series on

Optimization, Society for Industrial and Applied Mathematics, Philadelphia, 2000.
[23] F. E. Curtis, D. P. Robinson, and M. Samadi, *A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}\left(\epsilon^{-3/2}\right)$ for nonconvex optimization*, Math. Program., 162 (2017), pp. 1–32.
[24] ———, *Complexity analysis of a trust funnel algorithm for equality constrained optimization*, SIAM Journal on Optimization, 28 (2018), pp. 1533–1563.
[25] ———, *An inexact regularized Newton framework with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization*, IMA J. Numer. Anal., (2018).
[26] J.-P. Dussault and D. Orban, *Scalable adaptative cubic regularization methods*, tech. rep., 2015.
[27] A. V. Fiacco and G. P. McCormick, *Nonlinear programming: sequential unconstrained minimization techniques*, vol. 4, Siam, 1990.
[28] G. N. Grapiglia and Y. Yuan, *On the complexity of an augmented lagrangian method for nonconvex optimization*, arXiv preprint arXiv:1906.05622, (2019).
[29] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, second ed., 2008.
[30] G. Haeser, H. Liu, and Y. Ye, *Optimality condition and complexity analysis for linearly-constrained optimization without differentiability on the boundary*, Mathematical Programming, (2018).
[31] O. Hinder and Y. Ye, *Worst-case iteration bounds for log barrier methods for problems with nonconvex constraints*, arXiv preprint arXiv:1807.00404, (2018).
[32] C. Jin, P. Netrapalli, and M. I. Jordan, *Accelerated gradient descent escapes saddle points faster than gradient descent*, in Proceedings of the 31st Conference On Learning Theory, PMLR, 2018, pp. 1042–1085.
[33] J. M. Martínez and M. Raydan, *Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization*, J. Global Optim., 68 (2017), pp. 367–385.
[34] Y. Nesterov and B. T. Polyak, *Cubic regularization of Newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.
[35] M. Nouiehed, J. D. Lee, and M. Razaviyayn, *Convergence to second-order stationarity for constrained non-convex optimization*, arXiv preprint arXiv:1810.02024, (2018).
[36] C. W. Royer, M. O'Neill, and S. J. Wright, *A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization*, Math. Program., (2019).
[37] C. W. Royer and S. J. Wright, *Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization*, SIAM J. Optim., 28 (2018), pp. 1448–1477.
[38] Y. Xie and S. J. Wright, *Complexity of proximal augmented lagrangian for nonconvex optimization with nonlinear equality constraints*, arXiv preprint arXiv:1908.00131, (2019).

Appendix A. Proofs of Technical Results.

A.1 Proof of Lemma 8. *Proof.* For scalar $y > -1$, define $g(y) = -\log(1+y)$. We have $g'(y) = -1/(1+y)$, $g''(y) = 1/(1+y)^2$ and $g^{(3)}(y) = -2/(1+y)^3$. By Taylor's theorem, we have

$$g(y) = g(0) + yg'(0) + \frac{1}{2}y^2 g''(0) + \frac{1}{2}\int_0^y (y-t)^2 g^{(3)}(t)dt. \tag{67}$$

Substituting $t = yu$ and using $|y| \le \beta < 1$, we have

$$\frac{1}{2}\int_0^y (y-t)^2 g^{(3)}(t)dt = -y\int_0^1 (y-yu)^2 \frac{du}{(1+yu)^3} \le |y|^3 \int_0^1 (1-u)^2 \frac{du}{(1-\beta u)^3}.$$

Now, since $(1-u)^2$ is monotonically decreasing in u and $1/(1-\beta u)^3$ is monotonically increasing in u, we can apply Chebyshev's integral inequality:

$$\begin{aligned}|y|^3 \int_0^1 (1-u)^2 \frac{du}{(1-\beta u)^3} &\leq |y|^3 \left[\int_0^1 (1-u)^2 du\right] \left[\int_0^1 \frac{du}{(1-\beta u)^3}\right] \\ &= \frac{|y|^3}{3} \left[\int_0^1 \frac{du}{(1-\beta u)^3}\right] \\ &= \frac{|y|^3}{6} \frac{1}{\beta} \left(\frac{1}{(1-\beta)^2} - 1\right) = \frac{|y|^3}{6} \frac{2-\beta}{(1-\beta)^2}.\end{aligned}$$

By combining with (67), we obtain

$$-\log(1+y) \leq -y + \frac{1}{2} y^2 + \frac{|y|^3}{6} \frac{2-\beta}{(1-\beta)^2}.$$

Now, for some coordinate i, let $y = (\bar{x}_i / x_i)\, d_i$. Clearly, we have $|y| \leq \beta$ so

$$\begin{aligned}-\log\left(1 + \frac{\bar{x}_i}{x_i} d_i\right) &\leq -\frac{\bar{x}_i}{x_i} d_i + \frac{1}{2}\left(\frac{\bar{x}_i}{x_i} d_i\right)^2 + \frac{|\frac{\bar{x}_i}{x_i} d_i|^3}{6} \frac{2-\beta}{(1-\beta)^2} \\ &\leq -\frac{\bar{x}_i}{x_i} d_i + \frac{1}{2}\left(\frac{\bar{x}_i}{x_i} d_i\right)^2 + \frac{|d_i|^3}{6} \frac{2-\beta}{(1-\beta)^2}\end{aligned} \tag{68}$$

holds. By the properties of logarithms, we have

$$-\log\left(x_i\left(1 + \frac{\bar{x}_i}{x_i} d_i\right)\right) = -\log(x_i + \bar{x}_i d_i) = -\log(x_i) - \log\left(1 + \frac{\bar{x}_i}{x_i} d_i\right).$$

By rearranging this inequality and substituting from (68), we have

$$-\log\left(x_i + \bar{x}_i d_i\right) + \log(x_i) \leq -\frac{\bar{x}_i}{x_i} d_i + \frac{1}{2}\left(\frac{\bar{x}_i}{x_i} d_i\right)^2 + \frac{|d_i|^3}{6} \frac{2-\beta}{(1-\beta)^2}.$$

By summing this inequality over $i = 1, 2, \dots, n$, we obtain

$$\begin{aligned}&-\sum_{i=1}^n \log\left(x_i + \bar{x}_i d_i\right) + \sum_{i=1}^n \log(x_i) \\ &\leq -e^\top X^{-1} \bar{X} d + \frac{1}{2} d^\top \bar{X} X^{-2} \bar{X} d + \sum_{i=1}^n \frac{|d_i|^3}{6} \frac{2-\beta}{(1-\beta)^2} \\ &= -e^\top X^{-1} \bar{X} d + \frac{1}{2} d^\top \bar{X} X^{-2} \bar{X} d + \frac{2-\beta}{6(1-\beta)^2} \|d\|_3^3 \\ &\leq -e^\top X^{-1} \bar{X} d + \frac{1}{2} d^\top \bar{X} X^{-2} \bar{X} d + \frac{2-\beta}{6(1-\beta)^2} \|d\|^3,\end{aligned}$$

where $\|d\|_3$ denotes the ℓ_3 norm of d. (The final inequality follows from $\|d\|_3 \leq \|d\|_2$). □

A.2 Proof of Lemma 10. *Proof.* For simplicity of notation, we again use $H = \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k$ and $g = \bar{X}_k \nabla \phi_\mu(x^k)$ in the proof.

Suppose first that the unit step length $\alpha_k = 1$ is accepted. Then, if $\|d^k\| < c_d \epsilon_H$, it follows from Lemma 9 that both (6b) and (6c) hold at x^{k+1}, so we are in case A. Otherwise, the statment of case B holds by

$$\alpha_k \|d^k\| = \|d^k\| \geq c_d \epsilon_H \geq c_{\rm sol} \epsilon_H.$$

For the remainder of the proof, we assume that $\alpha_k < 1$. Recall from the statement of Lemma 7 that

$$\gamma_k = \max\left\{\frac{\|X_k^{-1}\bar{X}_k\hat{d}^k\|_\infty}{\beta}, 1\right\}.$$

For any $j \geq 0$ such that the sufficient decrease condition (13) does not hold, we have from (4), (26a), (26c), and Lemma 8 that

$$\begin{aligned}
&-\frac{\eta}{6}\theta^{3j}\|d^k\|^3 \\
&\leq \phi_\mu(x^k + \theta^j\bar{X}_k d^k) - \phi_\mu(x^k) \\
&\leq \theta^j\nabla f(x^k)^\top\bar{X}_k d^k + \frac{\theta^{2j}}{2}(d^k)^\top\bar{X}_k\nabla^2 f(x^k)\bar{X}_k d^k + \frac{L_H}{6}\theta^{3j}\|\bar{X}_k d^k\|^3 && \text{by (4)} \\
&\quad - \mu\theta^j e^\top X_k^{-1}\bar{X}_k d^k + \frac{\mu\theta^{2j}}{2}(d^k)^\top\bar{X}_k X_k^{-2}\bar{X}_k d^k + \frac{\mu(2-\beta)}{6(1-\beta)^2}\theta^{3j}\|d^k\|^3 && \text{by Lemma 8} \\
&= \theta^j g^\top d^k + \frac{\theta^{2j}}{2}(d^k)^\top H d^k + \frac{L_H}{6}\theta^{3j}\|\bar{X}_k d^k\|^3 + \frac{\mu(2-\beta)}{6(1-\beta)^2}\theta^{3j}\|d^k\|^3 \\
&= -\theta^j\gamma_k(d^k)^\top\left(H + 2\epsilon_H I\right)d^k + \frac{\theta^{2j}}{2}(d^k)^\top H d^k && \text{by (26c)} \\
&\quad + \frac{L_H}{6}\theta^{3j}\|\bar{X}_k d^k\|^3 + \frac{\mu(2-\beta)}{6(1-\beta)^2}\theta^{3j}\|d^k\|^3 \\
&= -\theta^j\left(\gamma_k - \frac{\theta^j}{2}\right)(d^k)^\top\left(H + 2\epsilon_H I\right)d^k - \theta^{2j}\epsilon_H\|d^k\|^2 \\
&\quad + \frac{L_H}{6}\theta^{3j}\|\bar{X}_k d^k\|^3 + \frac{\mu(2-\beta)}{6(1-\beta)^2}\theta^{3j}\|d^k\|^3 \\
&\leq -\theta^j\gamma_k\epsilon_H\|d^k\|^2 + \frac{1}{2}\theta^{2j}\epsilon_H\|d^k\|^2 - \theta^{2j}\epsilon_H\|d^k\|^2 && \text{by (26a)} \\
&\quad + \frac{L_H(1-\beta)^2 + (2-\beta)}{6(1-\beta)^2}\theta^{3j}\|d^k\|^3 && \text{by } \mu < 1 \\
&\leq -\theta^j\gamma_k\epsilon_H\|d^k\|^2 + \frac{L_H(1-\beta)^2 + (2-\beta)}{6(1-\beta)^2}\theta^{3j}\|d^k\|^3.
\end{aligned}$$

Therefore, for any $j \geq 0$ at which sufficient decrease is not attained, we have by rearranging terms in the inequality above and using the definition of γ_k that

$$\begin{aligned}
\frac{(L_H+\eta)(1-\beta)^2 + (2-\beta)}{6(1-\beta)^2}\theta^{2j} &\geq \max\left\{\frac{\|X_k^{-1}\bar{X}_k\hat{d}^k\|_\infty}{\beta}, 1\right\}\epsilon_H\|d^k\|^{-1} \\
&\geq \epsilon_H\|d^k\|^{-1}. \qquad (69)
\end{aligned}$$

Evaluating this expression at $j = 0$, we have that

$$\|d^k\| \geq \frac{6(1-\beta)^2}{(L_H+\eta)(1-\beta)^2 + (2-\beta)}\epsilon_H. \qquad (70)$$

From (26b), we have

$$\|d^k\| \leq 1.1\epsilon_H^{-1}\|g\| \leq 1.1\epsilon_H^{-1}(\|\bar{X}_k\nabla f(x^k)\| + \mu\|\bar{X}_k X_k^{-1}e\|) \leq 1.1\epsilon_H^{-1}(U_g + \mu\sqrt{n}),$$

where we used $\|\bar{X}_k\| \le 1$, $\|\nabla f(x^k)\| \le U_g$, and $\|\bar{X}_k X_k^{-1} e\| \le \sqrt{n}$ in the final inequality. Thus, for any $j > j_{\rm sol}$ we have from definition (40) and this bound on $\|d^k\|$ that

$$\begin{aligned}\theta^{2j} < \theta^{2j_{\rm sol}} &\le \frac{6(1-\beta)^2}{(L_H+\eta)(1-\beta)^2+(2-\beta)} \frac{\epsilon_H^2}{1.1(U_g+\mu\sqrt{n})} \\ &\le \frac{6(1-\beta)^2\epsilon_H}{(L_H+\eta)(1-\beta)^2+(2-\beta)} \|d^k\|^{-1}.\end{aligned}$$

Therefore, (69) cannot be satisfied for any $j > j_{\rm sol}$ so the line search must terminate with $\alpha_k = \theta^{j_k}$ for some $1 \le j_k \le j_{\rm sol}+1$. The previous index $j_k - 1$ satisfies (69), so we also have

$$\theta^{2(j_k-1)} = \frac{\theta^{2j_k}}{\theta^2} \ge \frac{6(1-\beta)^2\epsilon_H}{(L_H+\eta)(1-\beta)^2+(2-\beta)} \|d^k\|^{-1}.$$

It follows that

$$\begin{aligned}\alpha_k\|d^k\| = \theta^{j_k}\|d^k\| &\ge \left(\frac{6(1-\beta)^2\theta^2\epsilon_H}{(L_H+\eta)(1-\beta)^2+(2-\beta)}\right)^{1/2} \|d^k\|^{1/2} \\ &\ge \frac{6(1-\beta)^2\theta^2}{(L_H+\eta)(1-\beta)^2+(2-\beta)}\epsilon_H\end{aligned}$$

holds, where the final inequality comes from (70) and $\theta < 1$. Thus, the conclusion holds in this case as well and the proof is complete. □

A.3 Proof of Lemma 11. *Proof.* We again use the notation $H = \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k$ and $g = \bar{X}_k \nabla \phi_\mu(x^k)$ in this proof.

We begin by noting that when the unit step, $\alpha_k = 1$, is taken, we have

$$\alpha_k\|d^k\| = \|d^k\| \ge \epsilon_H,$$

where the inequality follows from (29).

In the remainder of the proof, we assume that the unit step length is not accepted. Then, for any $j \ge 0$ such that (13) does not hold, we have from (4) and (29) along with the result of Lemma 8 that

$$\begin{aligned}
&-\frac{\eta}{6}\theta^{3j}\|d^k\|^3 \\
&\le \phi_\mu(x^k + \theta^j \bar{X}_k d^k) - \phi_\mu(x^k) \\
&\le \theta^j \nabla f(x^k)^\top \bar{X}_k d^k + \frac{\theta^{2j}}{2}(d^k)^\top \bar{X}_k \nabla^2 f(x^k) \bar{X}_k d^k + \frac{L_H}{6}\theta^{3j}\|\bar{X}_k d^k\|^3 \qquad \text{by (4)} \\
&\quad - \mu\theta^j e^\top X_k^{-1}\bar{X}_k d^k + \frac{\mu\theta^{2j}}{2}(d^k)^\top \bar{X}_k X_k^{-2} \bar{X}_k d^k + \frac{\mu(2-\beta)}{6(1-\beta)^2}\theta^{3j}\|d^k\|^3 \qquad \text{by Lemma 8} \\
&= \theta^j g^\top d^k + \frac{\theta^{2j}}{2}(d^k)^\top H d^k + \frac{L_H}{6}\theta^{3j}\|\bar{X}_k d^k\|^3 + \frac{\mu(2-\beta)}{6(1-\beta)^2}\theta^{3j}\|d^k\|^3 \\
&\le -\frac{\theta^{2j}}{2}\|d^k\|^3 + \frac{L_H(1-\beta)^2+(2-\beta)}{6(1-\beta)^2}\theta^{3j}\|d^k\|^3, \quad \text{by (29) and } \mu < 1.
\end{aligned}$$

By rearranging this expression, we have for all such j that

$$\theta^j \ge \frac{3(1-\beta)^2}{(L_H+\eta)(1-\beta)^2+(2-\beta)}$$

which is true only for $j \leq j_{\text{nc}}$. Thus, the line search must terminate for some $j_k \leq j_{\text{nc}} + 1$. Since the line search failed to stop at iteration $j_k - 1$, we must have

$$\theta^{j_k - 1} = \frac{\theta^{j_k}}{\theta} \geq \frac{3(1-\beta)^2}{(L_H + \eta)(1-\beta)^2 + (2-\beta)}.$$

Therefore, using $\|d^k\| \geq \epsilon_H$ from (29), we have that

$$\alpha_k \|d^k\| = \theta^{j_k} \|d^k\| \geq \frac{3(1-\beta)^2 \theta}{(L_H + \eta)(1-\beta)^2 + (2-\beta)} \epsilon_H$$

as required. $\square$