SIAM J. OPTIM.
Vol. 34, No. 4, pp. 3592–3622

SEQUENTIAL QUADRATIC OPTIMIZATION FOR STOCHASTIC OPTIMIZATION WITH DETERMINISTIC NONLINEAR INEQUALITY AND EQUALITY CONSTRAINTS*

FRANK E. CURTIS†, DANIEL P. ROBINSON†, AND BAOYU ZHOU‡

Abstract. A sequential quadratic optimization algorithm for minimizing an objective function defined by an expectation subject to nonlinear inequality and equality constraints is proposed, analyzed, and tested. The context of interest is when it is tractable to evaluate constraint function and derivative values in each iteration, but it is intractable to evaluate the objective function or its derivatives in any iteration, and instead, an algorithm can only make use of stochastic objective gradient estimates. Under loose assumptions, including that the gradient estimates are unbiased, the algorithm is proved to possess convergence guarantees in expectation. The results of numerical experiments are presented to demonstrate that the proposed algorithm can outperform an alternative approach that relies on the ability to compute more accurate gradient estimates and can outperform a stochastic algorithm that employs a penalty method to enforce the constraints.

Key words. nonlinear optimization, stochastic optimization, sequential quadratic optimization

MSC codes. 49M05, 49M37, 65K05, 65K10, 90C15, 90C30, 90C55

DOI. 10.1137/23M1556149

1. Introduction.

We propose a sequential quadratic optimization (commonly known as SQP) algorithm for minimizing an objective function defined by an expectation subject to nonlinear inequality and equality constraints. Such optimization problems arise in a plethora of application areas, including, but not limited to, machine learning [30], network optimization [7], resource allocation [27], portfolio optimization [39], risk-averse partial-differential-equation-constrained optimization [29], maximum-likelihood estimation [26], and multistage optimization [43].

The design and analysis of deterministic algorithms for solving continuous optimization problems involving inequality and equality constraints has been a well-studied topic for decades. Numerous types of such algorithms, such as penalty methods, interior-point methods, and SQP methods, have been designed to solve such problems. Penalty methods are based on the idea of using unconstrained optimization algorithms to minimize a weighted sum—determined by a penalty parameter—of the objective and a measure of constraint violation; e.g., see [11, 20, 49] for algorithms that make use of nondifferentiable (exact) penalty functions, and see [15, 16, 22, 50] for algorithms that make use of differentiable (exact) penalty functions. While they are able to offer convergence guarantees from remote starting points, the numerical performance of penalty methods often suffers from ill-conditioning of the penalty functions and/or sensitivity of the algorithm's performance on the particular scheme employed for updating the penalty parameter [36]. Interior-point methods [17] are

*Received by the editors February 28, 2023; accepted for publication (in revised form) July 30, 2024; published electronically November 12, 2024.
https://doi.org/10.1137/23M1556149
Funding: This material is based upon work supported by the U.S. NSF under award CCF-2139735 and by the Office of Naval Research under award N00014-21-1-2532.
†Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015 USA (frank.e.curtis@lehigh.edu, daniel.p.robinson@lehigh.edu).
‡School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85281 USA (baoyu.zhou@asu.edu).

designed to use barrier functions to guide the algorithm along a central path through the interior of the feasible region (or, at least, the interior of a set defined by bounds on a subset of the variables) to a solution [9, 10, 31, 32, 47, 48]. Such algorithms have been shown to be very effective in practice, which is why many state-of-the-art software packages for continuous nonlinear optimization are built on interior-point methods; see, e.g., [10, 45]. Overall, both penalty and interior-point methods involve the use of additional objective terms to handle the presence of inequality constraints.

Alternatively, in this paper, we present, analyze, and demonstrate the numerical performance of an SQP method for solving continuous nonlinear optimization problems. The SQP paradigm is based on the idea of, at each iterate, solving a subproblem (or subproblems) defined based on a local linearization of the constraint function and a local quadratic approximation of the objective or Lagrangian. Unlike in the deterministic setting, for which numerous SQP algorithms have been proposed (see, e.g., [19, 21, 25, 36]), there have been few stochastic algorithms proposed for solving optimization problems with nonlinear constraints. That said, in the past few years, a couple of classes of stochastic SQP methods have been designed for optimization subject to nonlinear *equality* constraints. For example, [3] proposes an SQP algorithm that uses stochastic objective gradient estimates for solving such problems that employs an adaptive step-size policy based on Lipschitz constants (or estimates of them). For an alternative setting in which one is willing to compute objective value estimates as well, and to refine objective function and gradient estimates within a given iteration until probabilistic conditions of accuracy are satisfied, [33] proposes a line-search stochastic SQP method. There have subsequently been multiple extensions of the methods in [3] and [33], as well as work on different but related algorithmic strategies—still for the setting of only nonlinear equality constraints. There has been work on relaxing constraint qualifications [2], allowing matrix-free and inexact solves of the arising linear systems [14], using a trust-region methodology [18], incorporating noisy (potentially biased) function and gradient estimates [5, 37], employing variance-reduction strategies [1, 4], considering sketch-and-project techniques [35], and analyzing the worst-case complexity (see [13]) of the method proposed in [3].

Unlike the setting of equality constraints only, to our knowledge, there has been very little work on the design and analysis of stochastic algorithms for optimization subject to nonlinear (nonconvex) inequality and equality constraints. Three exceptions are the active-set line-search SQP algorithms proposed in [34] and (very recently) in [41] and the momentum-based augmented Lagrangian method (a penalty method) proposed in [44]. We expect that our proposed SQP algorithm will perform well in comparison to a stochastic–gradient-based penalty method. We demonstrate with numerical experiments that our approach can outperform the algorithm proposed in [34]. We remark in passing that interior-point methods often outperform SQP methods in the deterministic setting, but as far as we are aware, there exists no interior-point method designed for the stochastic setting that we consider.

1.1. Contributions. In this paper, we build on the algorithmic strategy and analysis in [3] to propose and analyze an adaptive stochastic SQP algorithm for solving nonlinear optimization problems subject to (deterministic) inequality and equality constraints. This work involves significant advancements beyond [3] that are necessary since, unlike in the setting of only having equality constraints, the presence of inequality constraints automatically guarantees that, at a given iterate, the search direction computed in a stochastic SQP method will be a biased estimate of the "true" search direction, i.e., the one that would be computed if the actual gradient of the objective function were available. This necessitates a distinct change in the design of the algorithm, as well as distinct alterations to the convergence analysis, since the analysis in [3] relies heavily on the search directions being (conditionally) unbiased estimators of their "true" counterparts. The algorithm from the literature that can be seen as the nearest alternative approach is the algorithm in [34]. However, there are substantial differences between the algorithm and analysis in [34] and those presented in this paper. Like in [33] for the equality-only case, the algorithm in [34] is designed for the setting in which one is willing to refine function and gradient estimates within an iteration until probabilistic conditions of accuracy are satisfied, and in this manner, the analysis of that algorithm offers guarantees that are relatively closer to those offered for a deterministic algorithm. By contrast, the algorithm in this paper, like the algorithm in [3], is designed to allow the stochastic gradient estimates to be potentially much less accurate, and in such a context, we are satisfied with offering convergence guarantees in expectation. We compare the numerical performance of our proposed algorithm with that in [34] to demonstrate that there are settings in which our proposed approach has advantages in practice. We also compare our method with that in [46] and a stochastic subgradient method employed to minimize a penalty function.

1.2. Notation. We use $\mathbb{R}$ to denote the set of real numbers, $\overline{\mathbb{R}}$ to denote the set of extended-real numbers (i.e., $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$), and $\mathbb{R}_{\geq a}$ (resp., $\mathbb{R}_{>a}$) to denote the set of real numbers greater than or equal to (resp., greater than) $a \in \mathbb{R}$. We append a superscript to such a set to denote the space of vectors or matrices whose elements are restricted to the indicated set; e.g., we use $\mathbb{R}^n$ to denote the set of n-dimensional real vectors and $\mathbb{R}^{m\times n}$ to denote the set of m-by-n-dimensional real matrices. We use $\mathbb{N} := \{1, 2, \dots\}$ to denote the set of positive integers, and, given $n \in \mathbb{N}$, we use $[n] := \{1, \dots, n\}$ to denote the set of positive integers less than or equal to n. Given $(a, b) \in \mathbb{R}^n \times \mathbb{R}^n$, we write $a \perp b$ to mean—with a_i and b_i denoting the ith elements of a and b, respectively—that $a_i = 0$ and/or $b_i = 0$ for all $i \in [n]$. Given real symmetric matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$, we write $A \succeq B$ (resp., $A \succ B$) to indicate that $A - B$ is positive semidefinite (resp., positive definite). Given $H \in \mathbb{R}^{n\times n}$ with $H \succ 0$ and $a \in \mathbb{R}^n$, we denote the norm $\|a\|_H := \sqrt{a^T H a}$.

Our problem of interest is defined with respect to a variable $x \in \mathbb{R}^n$, and the algorithm that we propose and analyze is iterative, meaning that, in any run, it generates an iterate sequence that we denote as $\{x_k\}$ with $x_k \in \mathbb{R}^n$ for all generated $k \in \mathbb{N}$; i.e., $\{x_k\} \subset \mathbb{R}^n$. (We use such notation throughout the paper when the elements of sequence are contained within a given set. We say "for all *generated* $k \in \mathbb{N}$" since our proposed algorithm might terminate finitely. Whether a subscript is being used to indicate the element of a vector or the index number of a sequence is always made clear by the context. The ith element of an iterate x_k is denoted $[x_k]_i$.) We use subscripts similarly to denote other quantities corresponding to each iteration of the algorithm; e.g., we introduce a merit parameter denoted as $\tau \in \mathbb{R}_{>0}$ whose value in iteration $k \in \mathbb{N}$ is denoted as $\tau_k \in \mathbb{R}_{>0}$, and, corresponding to a constraint function c (see problem (2.1) below), we denote its value at x_k as $c_k := c(x_k)$.

The iteration-dependent quantities mentioned in the previous paragraph—and additional ones introduced in the description of our algorithm—represent realizations of the random variables in a stochastic process generated by the algorithm. Specifically, the behavior of our algorithm is dictated by prescribed initial conditions and a sequence of stochastic objective gradient estimators that we denote by $\{G_k\}$. After proving preliminary results that hold for every run of the algorithm, we present our ultimate convergence theory for our algorithm in terms of a filtration defined in terms of σ-algebras dependent on the initial conditions of the algorithm and $\{G_k\}$.

1.3. Organization. A statement of our problem of interest and preliminary assumptions about its objective and constraint functions, as well as about user-defined quantities in our proposed algorithm, are given in section 2. A description of our proposed algorithm is provided in section 3. Convergence in expectation of the algorithm is proved under reasonable assumptions in section 4. The results of numerical experiments are presented in section 5, and concluding remarks are given in section 6.

2. Setting. We formulate our problem of interest as

$$\min_{x\in\mathbb{R}^n} f(x) \text{ subject to (s.t.) } c(x)=0 \text{ and } x\geq 0 \text{ with } f(x)=\mathbb{E}_\omega[F(x,\omega)], \tag{2.1}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ and $c:\mathbb{R}^n\to\mathbb{R}^m$ are continuously differentiable, ω is a random variable with associated probability space $(\Omega,\mathcal{F},\mathbb{P}_\omega)$, $F:\mathbb{R}^n\times\Omega\to\mathbb{R}$, and $\mathbb{E}_\omega$ denotes expectation taken with respect to $\mathbb{P}_\omega$. Our algorithm and analysis extend easily to the setting in which the nonnegativity constraint in (2.1) is generalized to $l\leq x\leq u$ for some $(l,u)\in\overline{\mathbb{R}}^n\times\overline{\mathbb{R}}^n$ with $l_i\leq u_i$ for all $i\in[n]$; we merely consider nonnegativity in (2.1) for the sake of notational simplicity. It is also worth mentioning that any smooth constrained optimization problem can be reformulated as (2.1) (or at least as such a problem with generalized bound constraints); e.g., inequality constraints $c_{\mathcal{I}}(x)\leq 0$, where $c_{\mathcal{I}}:\mathbb{R}^n\to\mathbb{R}^{m_{\mathcal{I}}}$ is continuously differentiable, can be reformulated to fit into the form of (2.1) through the incorporation of slack variables, say, $s\in\mathbb{R}^{m_{\mathcal{I}}}$, to have the constraints $c_{\mathcal{I}}(x)+s_{\mathcal{I}}=0$ and $s_{\mathcal{I}}\geq 0$.

We make the following assumption throughout the remainder of the paper pertaining to the functions in problem (2.1) and our proposed algorithm. As seen in the following section, our algorithm seeks feasibility and stationarity with respect to (2.1) by generating an iterate sequence that stays feasible with respect to the bound constraints, meaning that, in any run of the algorithm, $x_k\in\mathbb{R}^n_{\geq 0}$ for all generated $k\in\mathbb{N}$.

Assumption 2.1. Let $\mathcal{X}\subset\mathbb{R}^n$ be an open convex set that almost surely contains the iterate sequence $\{x_k\}\subset\mathbb{R}^n_{\geq 0}$ generated in any realization of a run of the algorithm. The objective function $f:\mathbb{R}^n\to\mathbb{R}$ is continuously differentiable and bounded below over $\mathcal{X}$, and the objective gradient function $\nabla f:\mathbb{R}^n\to\mathbb{R}^n$ is Lipschitz continuous and bounded in norm over $\mathcal{X}$. Similarly, for all $i\in[m]$, the constraint function $c_i:\mathbb{R}^n\to\mathbb{R}$ is continuously differentiable and bounded over $\mathcal{X}$, and the constraint gradient function $\nabla c_i:\mathbb{R}^n\to\mathbb{R}^n$ is Lipschitz continuous and bounded in norm over $\mathcal{X}$. Finally, the constraint Jacobian $\nabla c^T:\mathbb{R}^n\to\mathbb{R}^{m\times n}$ has full row rank over $\mathcal{X}$.

Under Assumption 2.1, there exists $f_{\text{inf}}\in\mathbb{R}$ and a tuple of positive constants $(\kappa_{\nabla f},\kappa_c,\kappa_{\nabla c},L,\Gamma)\in\mathbb{R}_{>0}\times\mathbb{R}_{>0}\times\mathbb{R}_{>0}\times\mathbb{R}_{>0}\times\mathbb{R}_{>0}$ such that, for all $x\in\mathcal{X}$, one has that

$$f(x)\geq f_{\text{inf}},\quad \|\nabla f(x)\|_2\leq\kappa_{\nabla f},\quad \|c(x)\|_2\leq\kappa_c,\quad \text{and}\quad \|\nabla c(x)\|_2\leq\kappa_{\nabla c} \tag{2.2}$$

and, for all $(x,\bar{x})\in\mathcal{X}\times\mathcal{X}$, one has that

$$\|\nabla f(x)-\nabla f(\bar{x})\|_2\leq L\|x-\bar{x}\|_2 \quad\text{and}\quad \|\nabla c(x)^T-\nabla c(\bar{x})^T\|_2\leq\Gamma\|x-\bar{x}\|_2. \tag{2.3}$$

In addition, due to the continuous differentiability of the objective and constraint functions, it follows that, at any local minimizer of (2.1) at which the Jacobian of the active constraints (i.e., equality constraints and inequality constraints active at their bounds) has full row rank, call it $x\in\mathbb{R}^n$, there exists $y\in\mathbb{R}^m$ and $z\in\mathbb{R}^n$ such that the following Karush–Kuhn–Tucker (KKT) conditions are satisfied:

$$\nabla f(x)+\nabla c(x)y-z=0,\quad c(x)=0,\quad 0\leq x\perp z\geq 0. \tag{2.4}$$

We refer to any $x \in \mathbb{R}^n$ such that there exists $(y,z) \in \mathbb{R}^m \times \mathbb{R}^n$ satisfying (2.4) as a first-order stationary point (or KKT point) with respect to (2.1).

Since our algorithm generates iterates that are feasible with respect to the bound constraints, but not necessarily with respect to the equality constraints, we need to account for the possible existence of points that are infeasible for (2.1) but are stationary with respect to the minimization of a constraint violation measure over $\mathbb{R}^n_{\geq 0}$. We refer to a point that is infeasible for (2.1) as an infeasible stationary point if it is stationary with respect to the minimization of $\frac{1}{2}\|c(x)\|_2^2$ subject to $x \in \mathbb{R}^n_{\geq 0}$, meaning

$$0 \leq x \perp \nabla c(x) c(x) \geq 0. \tag{2.5}$$

Each iteration of our algorithm requires a stochastic estimate of the gradient of the objective at the current iterate. In a given run at iteration $k \in \mathbb{N}$, the realization of the iterate and gradient estimate is (x_k, g_k), which, later in our analysis, we denote as a realization of the pair of random variables (X_k, G_k). (See section 4.3 for a complete description of a stochastic process that we analyze.) With respect to the gradient estimators, we make Assumption 2.2 below. For the prescribed (i.e., not random) sequence $\{\rho_k\} \subset \mathbb{R}_{>0}$ referenced in the assumption, we state precise conditions that it must satisfy in section 4.3. In the assumption and throughout the remainder of the paper, we use $\mathbb{E}_k[\cdot]$ to denote expectation taken with respect to the distribution of ω conditioned on a trace σ-algebra of an event $\mathcal{E}$, denoted by $\mathcal{F}_k$; see section 4.3.

Assumption 2.2. *For a prescribed $\{\rho_k\} \subset \mathbb{R}_{>0}$, one finds, for all $k \in \mathbb{N}$, that*

$$\mathbb{E}_k[G_k] = \nabla f(X_k) \quad \text{and} \quad \mathbb{E}_k[\|G_k - \nabla f(X_k)\|_2^2] \leq \rho_k. \tag{2.6}$$

One might relax the latter condition in (2.6) and obtain guarantees that are similar to those that we prove; see, e.g., [38]. We employ (2.6) for simplicity since it is sufficient for demonstrating the guarantees that our algorithmic approach can offer. We remark that our introduction of the sequence $\{\rho_k\}$—rather than a constant—is needed since one of our main theoretical results requires $\{\rho_k\} \to 0$. We further discuss this requirement, which is stronger than is needed for the equality-constrained setting (see, e.g., [2, 3]), immediately after Lemma 4.20, where it can be best explained.

Each iteration of our algorithm also makes use of a symmetric and positive-definite (SPD) matrix, denoted as $H_k \in \mathbb{R}^{n \times n}$ for iteration $k \in \mathbb{N}$, to define a quadratic term in the subproblem that is solved for computing the search direction. For simplicity, we assume that the sequence $\{H_k\}$ is prescribed; e.g., one may consider $H_k = I$ for all $k \in \mathbb{N}$. More generally, one could consider a more sophisticated scheme such as setting, for all $k \in \mathbb{N}$, the matrix H_k as a stochastic estimate of the Hessian of the objective function and/or a Lagrangian function as long as it is sufficiently positive definite and bounded and the choice is made to be conditionally uncorrelated with the stochastic gradient estimate. However, since considering such a loose requirement would only obfuscate our analysis without adding significant value, we assume, for simplicity, that $\{H_k\}$ is prescribed and merely satisfies the following.

Assumption 2.3. *There exists $(\kappa_H, \zeta) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$ with $\kappa_H \geq \zeta$ such that, for all $k \in \mathbb{N}$, the SPD matrix $H_k \in \mathbb{R}^{n \times n}$ has $\kappa_H I \succeq H_k \succeq \zeta I$.*

Observe from Assumption 2.3 that we are not assuming that accurate second-order information is being used by the algorithm. Hence, our convergence guarantees are of the type that may be expected for a first–order-type algorithm, although, in situations when it is computationally tractable, one might find better performance if H_k incorporates some (approximate) second-order derivative information.

3. Algorithm.

In this section, we present our proposed algorithm. We state the algorithm in terms of a particular realization of it (e.g., denoting the iterate for each $k \in \mathbb{N}$ as x_k), although our subsequent analysis of it (starting in section 4.3) will be written in terms of the stochastic process that the algorithm defines.

Each iteration of our algorithm proceeds as follows. First, given the current iterate $x_k \in \mathbb{R}^n_{\geq 0}$, the algorithm computes a direction whose purpose is to determine the progress that can be made in terms of reducing a measure of violation of a linearization of the equality constraints subject to the bound constraints. This is done in a manner that regularizes the component of the direction that lies in the null space of the constraint Jacobian. Specifically, the iteration commences by computing a direction $v_k := u_k + \nabla c(x_k) w_k \in \mathbb{R}^n$, where $u_k \in \text{Null}(\nabla c(x_k)^T)$ and $\nabla c(x_k) w_k \in \text{Range}(\nabla c(x_k))$, by solving the quadratic optimization subproblem

$$\begin{aligned} \min_{u \in \mathbb{R}^n, w \in \mathbb{R}^m} \ & \frac{1}{2}\|c_k + \nabla c(x_k)^T \nabla c(x_k) w\|_2^2 + \frac{1}{2}\mu_k \|u\|_2^2 \\ \text{s.t.} \ & \nabla c(x_k)^T u = 0 \ \text{ and } \ x_k + u + \nabla c(x_k) w \geq 0, \end{aligned} \tag{3.1}$$

where $\mu_k \in \mathbb{R}_{>0}$ is a user-prescribed parameter. Observe that, since $x_k \in \mathbb{R}^n_{\geq 0}$, this subproblem is always feasible, and by construction, it is convex. Generally, the solution of (3.1) might not be unique, but in our setting, it is unique since $\nabla c(x_k)^T$ has full row rank. In our analysis, we show that the solution of subproblem (3.1) is given by $(u_k, w_k) = (0, 0)$ if and only if the current iterate x_k is stationary with respect to the minimization of $\frac{1}{2}\|c(x)\|_2^2$ over $x \in \mathbb{R}^n_{\geq 0}$. This means, e.g., that if $c_k \neq 0$, but the solution of (3.1) is $(u_k, w_k) = (0, 0)$—which, by the fundamental theorem of linear algebra, occurs if and only if $v_k = u_k + \nabla c(x_k) w_k = 0$—then it is reasonable to terminate since x_k is an infeasible stationary point (see (2.5)), as in our algorithm.

We do not expect the value of the regularization parameter $\mu_k \in \mathbb{R}_{>0}$ to have a significant impact on the performance of the algorithm as long as it is set small relative to the constraint violation. (If μ_k is set too large relative to the constraint violation, then the regularization term might cause the algorithm to compute normal steps that are small in norm, which might slow progress.) The primary role of positive μ_k is to ensure that subproblem (3.1) has a unique solution since it might not if this parameter were set to zero. Nonetheless, for the sake of generality in the statement of our algorithm and analysis, we introduce the generic sequence $\{\mu_k\}$.

After computing $v_k \in \mathbb{R}^n$ by solving (3.1) and generating a stochastic objective gradient estimate $g_k \in \mathbb{R}^n$ (see Assumption 2.2), the algorithm next computes a search direction $d_k \in \mathbb{R}^n$ by solving the quadratic optimization subproblem

$$\min_{d \in \mathbb{R}^n} \ g_k^T d + \frac{1}{2} d^T H_k d \ \text{ s.t. } \ \nabla c(x_k)^T d = \nabla c(x_k)^T v_k \text{ and } x_k + d \geq 0. \tag{3.2}$$

By construction, this subproblem is feasible—indeed, by construction of (3.1) and $v_k = u_k + \nabla c(x_k) w_k$, it follows that $d = v_k$ is feasible for (3.2)—and, under Assumption 2.3, it is convex. The search direction d_k is designed to achieve the same progress toward linearized feasibility within the nonnegative orthant that is achieved by v_k; then, within the null space of $\nabla c(x_k)^T$ and the nonnegative orthant, it aims to minimize a (stochastically estimated) local quadratic approximation of the objective at x_k.

The remainder of the kth iteration proceeds in a similar manner as in [3, 14] with the primary goal of setting parameters and choosing a step size so as to achieve expected decrease in a merit function. In an algorithmic framework such as ours, one could employ an ℓ_p-norm merit function for any $p \in [1, \infty)$ and achieve similar algorithmic behavior. Common choices in the literature are $p = 1$ and $p = 2$. For example, in [3], the ℓ_1-norm merit function is used. For our purposes here, we employ the ℓ_2-norm merit function, namely, $\phi : \mathbb{R}^n \times \mathbb{R}_{>0} \to \mathbb{R}$ defined by $\phi(x,\tau) = \tau f(x) + \|c(x)\|_2$. We make this choice since it simplifies expressions in our analysis; e.g., since subproblem (3.1) employs a squared ℓ_2-norm for the first term in its objective function, it is consistent to employ an ℓ_2-norm merit function.

Back to the description of our algorithm, with the ℓ_2-norm merit function in mind, the algorithm next sets a value for the merit parameter $\tau_k \in \mathbb{R}_{>0}$. This is done by considering a local model of this merit function, namely, $l : \mathbb{R}^n \times \mathbb{R}_{>0} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined by $l(x,\tau,g,d) = \tau(f(x) + g^T d) + \|c(x) + \nabla c(x)^T d\|_2$ and, in particular, the reduction in this model defined for all $k \in \mathbb{N}$ by

$$\begin{aligned}\Delta l(x_k,\tau_k,g_k,d_k) &:= l(x_k,\tau_k,g_k,0) - l(x_k,\tau_k,g_k,d_k)\\ &= -\tau_k g_k^T d_k + \|c_k\|_2 - \|c_k + \nabla c(x_k)^T d_k\|_2\end{aligned} \tag{3.3}$$

and setting τ_k such that this reduction is sufficiently large. Specifically, with user-prescribed $(\epsilon_\tau, \sigma) \in (0,1) \times (0,1)$, the algorithm first sets

$$\tau_k^{\text{trial}} \leftarrow \begin{cases} \infty & \text{if } g_k^T d_k + \dfrac{1}{2} d_k^T H_k d_k \le 0,\\ \dfrac{(1-\sigma)(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T d_k\|_2)}{g_k^T d_k + \frac{1}{2} d_k^T H_k d_k}, & \text{otherwise,}\end{cases} \tag{3.4}$$

then sets the merit parameter value as

$$\tau_k \leftarrow \begin{cases} \tau_{k-1} & \text{if } \tau_{k-1} \le \tau_k^{\text{trial}},\\ \min\{(1-\epsilon_\tau)\tau_{k-1}, \tau_k^{\text{trial}}\}, & \text{otherwise.}\end{cases} \tag{3.5}$$

(The value $\tau_0 \in \mathbb{R}_{>0}$ is also prescribed by the user.) We show in our analysis (see Lemma 4.9) that this procedure for setting τ_k ensures that $\Delta l(x_k,\tau_k,g_k,d_k)$ is sufficiently large relative to the squared norm of the search direction and the improvement offered toward linearized feasibility. For use in the step-size procedure, the algorithm next sets a value $\xi_k \in \mathbb{R}_{>0}$ (referred to as the ratio parameter) that acts as an estimate for a lower bound of the ratio between the model reduction and a multiple of the squared norm of the search direction. Specifically, if $d_k \neq 0$, it sets

$$\xi_k^{\text{trial}} \leftarrow \frac{\Delta l(x_k,\tau_k,g_k,d_k)}{\tau_k \|d_k\|_2^2}, \quad \text{then} \quad \xi_k \leftarrow \begin{cases} \xi_{k-1} & \text{if } \xi_{k-1} \le \xi_k^{\text{trial}},\\ \min\{(1-\epsilon_\xi)\xi_{k-1}, \xi_k^{\text{trial}}\}, & \text{otherwise,}\end{cases} \tag{3.6}$$

where $(\xi_0, \epsilon_\xi) \in \mathbb{R}_{>0} \times (0,1)$ are user-prescribed parameters; see [3, 14] for further motivation. On the other hand, if $d_k = 0$, then it sets $\xi_k^{\text{trial}} \leftarrow \infty$ and $\xi_k \leftarrow \xi_{k-1}$.

The step-size selection procedure, which, for all $k \in \mathbb{N}$, chooses the step size $\alpha_k \in \mathbb{R}_{>0}$, can now be summarized as follows. First, if $d_k = 0$, then the algorithm simply sets all step-size values to 1. Second, suppose that $d_k \neq 0$. With user-prescribed $\eta \in (0,1)$, $\theta \in \mathbb{R}_{>0}$, and $\{\beta_k\}$ with $\beta_k \in (0,1]$ for all $k \in \mathbb{N}$ such that

$$\alpha_k^{\min} \leftarrow \frac{2(1-\eta)\beta_k \xi_k \tau_k}{\tau_k L + \Gamma} \in (0,1] \quad \text{for all} \quad k \in \mathbb{N} \tag{3.7}$$

and with the strongly convex function $\varphi_k : \mathbb{R}_{\ge 0} \to \mathbb{R}$ defined by

$$\begin{aligned}\varphi_k(\alpha) = (\eta - 1)\alpha\beta_k \Delta l(x_k,\tau_k,g_k,d_k) + \|c_k + \alpha \nabla c(x_k)^T d_k\|_2 - \|c_k\|_2\\ + \alpha(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T d_k\|_2) + \frac{1}{2}(\tau_k L + \Gamma)\alpha^2 \|d_k\|_2^2,\end{aligned} \tag{3.8}$$

Algorithm 3.1. Stochastic SQP.

Require: $x_1 \in \mathbb{R}^n_{\geq 0}$; $\{\mu_k\} \subset \mathbb{R}_{>0}$; $\{H_k\} \subset \mathbb{R}^{n\times n}$ satisfying Assumption 2.3; $\tau_0 \in \mathbb{R}_{>0}$; $\xi_0 \in \mathbb{R}_{>0}$; $\{\sigma, \eta, \epsilon_\tau, \epsilon_\xi\} \subset (0,1)$; $\{\beta_k\} \subset (0,1]$ satisfying (3.7); $\theta \in \mathbb{R}_{>0}$; $\{\rho_k\} \subset \mathbb{R}_{>0}$; Lipschitz constants $L \in \mathbb{R}_{>0}$ and $\Gamma \in \mathbb{R}_{>0}$ (see (2.3))

```
for k ∈ ℕ, do
    compute v_k ∈ ℝ^n by solving (3.1)
    if c_k ≠ 0 and v_k = 0, then terminate and return x_k (infeasible stationary)
    compute g_k ∈ ℝ^n (recall Assumption 2.2)
    compute d_k ∈ ℝ^n by solving (3.2)
    set τ_k^trial by (3.4) and τ_k by (3.5)
    if d_k = 0, then
        set ξ_k^trial ← ∞, ξ_k ← ξ_{k-1}, α_k^min ← 1, α_k^φ ← 1, α_k^max ← 1, and α_k ← 1
    else
        set ξ_k^trial and ξ_k by (3.6), α_k^min by (3.7), and both α_k^φ and α_k^max by (3.9)
        choose α_k ∈ [α_k^min, α_k^max]
    end if
    Set x_{k+1} ← x_k + α_k d_k
end for
```

the algorithm sets the values

$$\alpha_k^{\varphi} \leftarrow \max\{\alpha \in \mathbb{R}_{\geq 0} : \varphi_k(\alpha) \leq 0\} \quad \text{and} \quad \alpha_k^{\max} \leftarrow \min\{1, \alpha_k^{\varphi}, \alpha_k^{\min} + \theta\beta_k\}. \tag{3.9}$$

The algorithm then chooses the step size α_k as any value in $[\alpha_k^{\min}, \alpha_k^{\max}]$. Overall, this strategy involves the computation of a minimal, conservative step-size value ($\alpha_k^{\min}$) that could simply be used as the step size to ensure our convergence guarantees. However, so that the algorithm may take larger step sizes to improve practical performance while still ensuring our convergence guarantees, the procedure computes a maximal step-size value ($\alpha_k^{\max}$) that ensures that the resulting step satisfies a sufficient–decrease-type condition. This can be seen in our analysis in Lemma 4.12.

A complete statement of our algorithm is given as Algorithm 3.1.

4. Analysis.

In this section, we provide theoretical results for Algorithm 3.1. We begin by introducing common assumptions under which one can establish stationarity measures for problem (2.1) that are defined by solutions of (3.1) and/or (3.2). These stationarity measures allow us to connect our convergence guarantees for Algorithm 3.1 with stationarity conditions for (2.1). Then, under Assumptions 2.1 and 2.3, we prove generally applicable results pertaining to the behavior of algorithmic quantities in any run of the algorithm. These results reveal that the algorithm is well defined in the sense that any run will either terminate and return an infeasible stationary point or generate an infinite sequence of iterates. We then consider the convergence properties of the algorithm in the event that the (monotonically nonincreasing) merit parameter sequence eventually produces values that are sufficiently small, yet bounded away from zero, which, as shown in our analysis, means that the sequence ultimately becomes constant at a sufficiently small value. This analysis, which includes our main convergence results for the algorithm, is provided under Assumption 4.15 stated in section 4.3. We follow this analysis with a section on theoretical results related to the occurrence of the event in Assumption 4.15. As in [3] for the equality-constraints-only setting, this discussion illuminates the fact that, while the event in Assumption 4.15 is not always guaranteed to occur due to the looseness of our assumptions about properties of the stochastic gradient estimates, the event represents likely behavior in practice, which shows that our convergence results about the algorithm are meaningful for real-world situations. We conclude this section with a discussion of the behavior of the algorithm in the deterministic setting, i.e., when the true gradient of the objective is employed in all iterations. This discussion is meant to provide confidence to a user that our algorithm is based on one that has state-of-the-art convergence properties under common assumptions in the deterministic setting.

4.1. Subproblems and stationarity measures. We begin by showing that subproblem (3.1) yields a zero solution if and only if the point defining the subproblem is feasible for problem (2.1) or an infeasible stationary point.

LEMMA 4.1. *Suppose that Assumption* 2.1 *holds and* $x \in \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$, *and, given* $\mu \in \mathbb{R}_{>0}$, *consider the quadratic optimization problem* (*recall* (3.1))

$$\begin{aligned} \min_{u\in\mathbb{R}^n, w\in\mathbb{R}^m}\ & \frac{1}{2}\|c(x)+\nabla c(x)^T\nabla c(x)w\|_2^2+\frac{1}{2}\mu\|u\|_2^2 \\ \text{s.t.}\ & \nabla c(x)^T u = 0 \quad \text{and} \quad x+u+\nabla c(x)w \geq 0. \end{aligned} \tag{4.1}$$

Then, the unique optimal solution of problem (4.1) *is* $(u,w)=(0,0)$ *if and only if* x *is feasible for problem* (2.1) *or an infeasible stationary point* (*i.e., it satisfies* (2.5)), *whereas* $(u,w)\neq(0,0)$ *if and only if* $\|c(x)\|_2 > \|c(x)+\nabla c(x)^T\nabla c(x)w\|_2$.

Proof. Suppose that the conditions of the lemma hold, and let (u,w) be the unique optimal solution of (4.1). Since $x \in \mathbb{R}^n_{\geq 0}$, it follows that $(0,0)$ is feasible for (4.1). In addition, necessary and sufficient optimality conditions for (4.1) are that, corresponding to $(u,w)\in\mathbb{R}^n\times\mathbb{R}^m$, there exists $(\gamma,\delta)\in\mathbb{R}^m\times\mathbb{R}^n$ with

$$\begin{aligned} &\nabla c(x)^T\nabla c(x)c(x)+\nabla c(x)^T\nabla c(x)\nabla c(x)^T\nabla c(x)w-\nabla c(x)^T\delta=0, \\ &\mu u+\nabla c(x)\gamma-\delta=0, \quad \nabla c(x)^T u=0, \quad \text{and} \quad 0\leq\delta\perp x+u+\nabla c(x)w\geq 0. \end{aligned} \tag{4.2}$$

If $(u,w)=(0,0)$, then it follows from (4.2) that

$$\nabla c(x)^T\nabla c(x)c(x)-\nabla c(x)^T\delta=0, \quad \nabla c(x)\gamma-\delta=0, \quad \text{and} \quad 0\leq\delta\perp x\geq 0. \tag{4.3}$$

Since $\nabla c(x)^T$ has full row rank, (4.3) implies that $\gamma=(\nabla c(x)^T\nabla c(x))^{-1}\nabla c(x)^T\delta = c(x)$, $\delta=\nabla c(x)c(x)$, and $0\leq\nabla c(x)c(x)\perp x\geq 0$, which, from (2.5), means that x is either feasible or an infeasible stationary point, as desired. On the other hand, if x is either feasible or an infeasible stationary point, meaning that $0\leq\nabla c(x)c(x)\perp x\geq 0$, then $u=0$, $w=0$, $\gamma=c(x)$, and $\delta=\nabla c(x)c(x)$ satisfy (4.2), and this solution (i.e., $(u,w)=(0,0)$) is unique since the objective of (4.1) is strongly convex.

Now, let us show that the unique optimal solution of (4.1) is $(u,w)\neq(0,0)$ if and only if $\|c(x)\|_2>\|c(x)+\nabla c(x)^T\nabla c(x)w\|_2$. If $\|c(x)\|_2>\|c(x)+\nabla c(x)^T\nabla c(x)w\|_2$, then $w\neq 0$ follows trivially, giving the desired conclusion. To prove the reverse implication, let us consider two cases. If $u\neq 0$, then, since $(0,0)$ is feasible for (4.1), we have that $\frac{1}{2}\|c(x)\|_2^2 \geq \frac{1}{2}\|c(x)+\nabla c(x)^T\nabla c(x)w\|_2^2+\frac{1}{2}\mu\|u\|_2^2 > \frac{1}{2}\|c(x)+\nabla c(x)^T\nabla c(x)w\|_2^2$, as desired. Second, if $u=0$ and $w\neq 0$, then w is the minimizer of the strongly convex objective $\frac{1}{2}\|c(x)+\nabla c(x)^T\nabla c(x)w\|_2^2$ subject to $x+\nabla c(x)w\geq 0$. Since 0 is feasible for this problem, $w\neq 0$ means that $\frac{1}{2}\|c(x)\|_2^2>\frac{1}{2}\|c(x)+\nabla c(x)^T\nabla c(x)w\|_2^2$, as desired. ◻

We now show that, under common assumptions and given $x_k\in\mathbb{R}^n_{\geq 0}$, the quantity $\|v_k\|_2^2$, where $v_k\in\mathbb{R}^n$ solves subproblem (3.1), represents a stationarity measure with respect to the problem to minimize $\frac{1}{2}\|c(x)\|_2^2$ subject to $x \in \mathbb{R}^n_{\geq 0}$. (The assumption in the lemma that $\mu_k = \mu \in \mathbb{R}_{>0}$ for all $k \in \mathbb{N}$ could be relaxed; see Remark 4.6 at the end of this subsection. We consider this case for the sake of brevity.)

LEMMA 4.2. *Suppose that Assumption* 2.1 *holds and there exists infinite* $\mathcal{S} \subseteq \mathbb{N}$ *such that, for some sequence* $\{x_k\} \subset \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$, *one finds that* $\{x_k\}_{k\in\mathcal{S}} \to x_*$ *for some* $x_* \in \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$, *where, with* $\mathcal{A}(x) := \{i \in [n] : x_i = 0\}$, $I_{\mathcal{A}(x)}$ *denoting the matrix composed of rows of* $I \in \mathbb{R}^{n\times n}$ *corresponding to indices in* $\mathcal{A}(x)$, *and* $\nabla c(x)_{\mathcal{A}(x)}$ *denoting the matrix composed of rows of* $\nabla c(x)$ *corresponding to indices in* $\mathcal{A}(x)$, *one finds that*

(i) $[\nabla c(x_*)c(x_*)]_i > 0$ *for all* $i \in \mathcal{A}(x_*)$ *and*

(ii) *the following matrix has full row rank:* $\begin{bmatrix} 0 & \nabla c(x_*)^T \\ \nabla c(x_*)_{\mathcal{A}(x_*)} & I_{\mathcal{A}(x_*)} \end{bmatrix}$.

Then, with $\mu_k = \mu \in \mathbb{R}_{>0}$ *for all* $k \in \mathbb{N}$ *and with* (u_k, w_k) *solving subproblem* (3.1) *and* $v_k := u_k + \nabla c(x)w_k$ *for all* $k \in \mathbb{N}$, *it follows that* x_* *satisfies the stationarity conditions* (2.5) *if and only if* $\{v_k\}_{k\in\mathcal{S}} \to 0$.

Proof. Let $\mathcal{A}_* := \mathcal{A}(x_*)$ and $j(x) := \nabla c(x)^T$, and consider the linear system

$$\begin{bmatrix} j(x)j(x)^T j(x)j(x)^T & 0 & 0 & -j(x)_{\mathcal{A}_*} \\ 0 & \mu I & j(x)^T & -I_{\mathcal{A}_*}^T \\ 0 & j(x) & 0 & 0 \\ j(x)_{\mathcal{A}_*}^T & I_{\mathcal{A}_*} & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ u \\ \gamma \\ \delta_{\mathcal{A}_*} \end{bmatrix} = \begin{bmatrix} -j(x)j(x)^T c(x) \\ 0 \\ 0 \\ -x_{\mathcal{A}_*} \end{bmatrix}.$$

Since, under the conditions of the lemma, the matrix in this linear system is nonsingular when $x = x_*$ (e.g., this follows from [42, Theorem 1.5.1] and (ii)), it follows that there exists an open ball $\mathcal{B}_*$ centered at x_* such that, for each $x \in \mathcal{B}_* \cap \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$, this linear system has a unique solution, call it $(w(x), u(x), \gamma(x), \delta_{\mathcal{A}_*}(x))$, and—due to continuity of the left-hand-side matrix and right-hand-side vector with respect to x—this solution varies continuously over $\mathcal{B}_* \cap \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$. If x_* satisfies (2.5), then it follows that $(0, 0, c(x_*), [j(x_*)^T c(x_*)]_{\mathcal{A}_*})$ (with $[j(x_*)^T c(x_*)]_{\mathcal{A}_*} > 0$) is the unique solution of the system at $x = x_*$ and, for all $x \in \mathcal{B}_* \cap \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$, the solution of the system in conjunction with $\delta_i = 0$ for all $i \notin \mathcal{A}_*$ satisfies (4.2), meaning that the components $(u(x), w(x))$ represent the unique optimal solution of problem (4.1). Hence, with respect to the quantities in the lemma and using Assumption 2.1, one finds that $\{v_k\}_{k\in\mathcal{S}} \to 0$, as desired. To prove the reverse inclusion, suppose that $\{v_k\}_{k\in\mathcal{S}} \to 0$, from which it follows by the fundamental theorem of linear algebra and (ii) in Lemma 4.2 that $\{(u_k, w_k)\}_{k\in\mathcal{S}} \to 0$. For all $k \in \mathcal{S}$, let $(u_k, w_k, \gamma_k, \delta_k)$ be a primal-dual optimal solution of (3.1) (satisfying optimality conditions of the form in (4.2)). One finds, under the conditions of the lemma, that, for all sufficiently large $k \in \mathcal{S}$, this solution has $[\delta_k]_i = 0$ for all $i \notin \mathcal{A}(x_*)$ whereas $(u_k, w_k, \gamma_k, [\delta_k]_{\mathcal{A}_*})$ solves the linear system above at $x = x_k$. Since, by the arguments above, this solution varies continuously within $\mathcal{B}_* \cap \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$, the fact that $\{x_k\}_{k\in\mathcal{S}} \to x_*$ implies that x_* satisfies (2.5), as desired. ∎

In fact, under the conditions of the prior lemma, the quantity $\|c_k\|_2 - \|c_k + \nabla c(x_k)^T v_k\|_2$ also represents a stationarity measure for the problem to minimize $\frac{1}{2}\|c(x)\|_2^2$ subject to $x \in \mathbb{R}^n_{\geq 0}$. This is shown in the following lemma.

LEMMA 4.3. *Suppose that Assumption* 2.1 *holds,* $\mu_k = \mu \in \mathbb{R}_{>0}$ *for all* $k \in \mathbb{N}$, *and there exist* $\lambda \in \mathbb{R}_{>0}$ *and infinite* $\mathcal{S}_\lambda \subseteq \mathbb{N}$ *such that, for some* $\{x_k\} \subset \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$, *one finds that* $\nabla c(x_k)^T \nabla c(x_k) \succeq \lambda I$ *for all* $k \in \mathcal{S}_\lambda$. *Then, there exists* $\kappa_{v,2} \in \mathbb{R}_{>0}$ *such that*

$$\|c_k\|_2 - \|c_k + \nabla c(x_k)^T v_k\|_2 \geq \kappa_{v,2}\|v_k\|_2^2 \quad \text{for all} \quad k \in \mathcal{S}_\lambda, \tag{4.4}$$

where $v_k = u_k + \nabla c(x_k) w_k$ *with* (u_k, w_k) *being the unique optimal solution of* (3.1). *Consequently, under the conditions of Lemma* 4.2, *if* $\mathcal{S}$ *is defined as in Lemma* 4.2 *and there exists* $\lambda \in \mathbb{R}_{>0}$ *and infinite* $\mathcal{S}_\lambda \subseteq \mathcal{S}$ *such that, for some* $\{x_k\} \subset \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$, *one finds that* $\nabla c(x_k)^T \nabla c(x_k) \succeq \lambda I$ *for all* $k \in \mathcal{S}_\lambda$, *then it follows that* $\{v_k\}_{k \in \mathcal{S}} \to 0$ *if and only if* $\{\|c_k\|_2 - \|c_k + \nabla c(x_k)^T v_k\|_2\}_{k \in \mathcal{S}} \to 0$.

Proof. Consider arbitrary $k \in \mathcal{S}_\lambda$. Under the stated conditions with $j_k := \nabla c(x_k)^T$, Lemma 4.1 implies that $\|c_k + j_k v_k\|_2 \leq \|c_k\|_2$. Hence, by Assumption 2.1,

$$\begin{aligned}\|c_k\|_2^2 - \|c_k + j_k v_k\|_2^2 &= (\|c_k\|_2 + \|c_k + j_k v_k\|_2)(\|c_k\|_2 - \|c_k + j_k v_k\|_2) \\ &\leq 2\|c_k\|_2(\|c_k\|_2 - \|c_k + j_k v_k\|_2) \leq 2\kappa_c(\|c_k\|_2 - \|c_k + j_k v_k\|_2).\end{aligned} \tag{4.5}$$

If $v_k = 0$, then (4.4) follows trivially. Hence, we may proceed under the assumption that $v_k \neq 0$, which, by $v_k = u_k + j_k^T w_k$ and the fundamental theorem of linear algebra, means that $u_k \neq 0$ and/or $w_k \neq 0$. If $w_k = 0$, then it follows, by construction of (3.1), that $u_k = 0$ as well. Hence, we may conclude from $v_k \neq 0$ that, in fact, $w_k \neq 0$. Since (u_k, w_k) is the unique optimal solution of (3.1), it follows that $\alpha_k^* = 1$ is the optimal solution of the strongly convex quadratic optimization problem

$$\min_{\alpha \in [0,1]} \ \frac{1}{2}\|c_k + \alpha j_k j_k^T w_k\|_2^2 + \frac{1}{2}\mu_k \|\alpha u_k\|_2^2, \tag{4.6}$$

which further implies (since an optimality condition of (4.6) is that the derivative of its objective function with respect to α is less than or equal to zero at $\alpha_k^* = 1$) that $-c_k^T j_k j_k^T w_k \geq \|j_k j_k^T w_k\|_2^2 + \mu_k \|u_k\|_2^2$. Consequently, one finds that

$$\begin{aligned}\|c_k\|_2^2 - \|c_k + j_k v_k\|_2^2 &= \|c_k\|_2^2 - \|c_k + j_k j_k^T w_k\|_2^2 \\ &= -2c_k^T j_k j_k^T w_k - \|j_k j_k^T w_k\|_2^2 \geq \|j_k j_k^T w_k\|_2^2 + 2\mu_k \|u_k\|_2^2.\end{aligned} \tag{4.7}$$

With (4.5) and (4.7), it follows from Assumption 2.1, the conditions of the lemma, and the Cauchy–Schwarz inequality implying that $\|w_k\|_2 \geq \|j_k^T w_k\|_2 / \|j_k^T\|_2$ that

$$\begin{aligned}\|c_k\|_2 - \|c_k + j_k v_k\|_2 &\geq (2\kappa_c)^{-1}(\|c_k\|_2^2 - \|c_k + j_k v_k\|_2^2) \\ &\geq (2\kappa_c)^{-1}(\|j_k j_k^T w_k\|_2^2 + 2\mu_k \|u_k\|_2^2) \geq (2\kappa_c)^{-1}(\lambda^2 \|w_k\|_2^2 + 2\mu_k \|u_k\|_2^2) \\ &\geq (2\kappa_c)^{-1}\left(\frac{\lambda^2}{\kappa_{\nabla c}^2}\|j_k^T w_k\|_2^2 + 2\mu_k \|u_k\|_2^2\right) \\ &\geq (2\kappa_c)^{-1}\min\left\{\frac{\lambda^2}{\kappa_{\nabla c}^2}, 2\mu_k\right\}(\|j_k^T w_k\|_2^2 + \|u_k\|_2^2) \\ &= (2\kappa_c)^{-1}\min\left\{\frac{\lambda^2}{\kappa_{\nabla c}^2}, 2\mu\right\}\|v_k\|_2^2 =: \kappa_{v,2}\|v_k\|_2^2,\end{aligned}$$

which gives (4.4), as desired. ∎

Next, we show that, if the point defining subproblem (3.2) is not an infeasible stationary point for problem (2.1), then the subproblem with $g_k = \nabla f(x_k)$ yields a zero solution if and only if the point defining the subproblem is stationary for (2.1).

LEMMA 4.4. *Suppose that Assumption* 2.1 *holds and, with respect to* $x \in \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$, *one finds that* $c(x) = 0$. *Given* $H \in \mathbb{R}^{n \times n}$ *with* $H \succ 0$, *consider* (*recall* (3.2))

$$\min_{d \in \mathbb{R}^n} \ \nabla f(x)^T d + \frac{1}{2} d^T H d \ \text{ s.t. } \ c(x) + \nabla c(x)^T d = 0 \ \ \text{and} \ \ x + d \geq 0. \tag{4.8}$$

Then, one finds that the optimal solution of problem (4.8) *is* $d = 0$ *if and only if* x *is a KKT point* (*i.e., first-order stationary point*) *for problem* (2.1).

Proof. Suppose that the conditions of the lemma hold, and let d be the optimal solution of (4.8). Since $x \in \mathbb{R}^n_{\geq 0}$ and $c(x) = 0$, it follows that the zero vector is feasible for (4.8). In addition, necessary and sufficient optimality conditions for subproblem (4.8) are that, corresponding to $d \in \mathbb{R}^n$, there exist $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ such that

$$\nabla f(x) + Hd + \nabla c(x)y - z = 0, \quad \nabla c(x)^T d = 0, \quad \text{and} \quad 0 \leq x + d \perp z \geq 0. \tag{4.9}$$

If $d = 0$, then, since $c(x) = 0$, it follows that (x, y, z) satisfies (2.4), as desired. On the other hand, if x is a KKT point for (2.1), then there exist $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ such that (x, y, z) satisfies (2.4), which, in turn, means that $d = 0$, along with (y, z), satisfies (4.9), and this solution is unique since the objective of (4.8) is strongly convex. ☐

We conclude this subsection by showing that, under common assumptions and given $x_k \in \mathbb{R}^n_{\geq 0}$, the quantity $\|d_k\|_2^2$, where $d_k \in \mathbb{R}^n$ solves subproblem (3.2) with $g_k = \nabla f(x_k)$, represents a stationarity measure with respect to (2.1). (The assumption in the lemma that $H_k = H$ for some $H \succ 0$ for all $k \in \mathbb{N}$ could be relaxed; see Remark 4.6 at the end of this subsection. We consider this case for the sake of brevity.)

LEMMA 4.5. *Suppose that Assumption* 2.1 *holds and there exists infinite* $\mathcal{S} \subseteq \mathbb{N}$ *such that, for some sequence* $\{x_k\} \subset \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$, *one finds that* $\{x_k\}_{k \in \mathcal{S}} \to x_*$ *for some* $x_* \in \mathcal{X} \cap \mathbb{R}^n_{\geq 0}$ *with* $c(x_*) = 0$ *and, with the notation in Lemma* 4.2, *one finds that*

(i) $-\nabla f(x_*) = \nabla c(x_*)y - I^T_{\mathcal{A}(x_*)} z_{\mathcal{A}(x_*)}$ *for some* $(y, z_{\mathcal{A}(x_*)}) \in \mathbb{R}^m \times \mathbb{R}^{|\mathcal{A}(x_*)|}_{>0}$ *and*

(ii) *the following matrix has full row rank:* $\begin{bmatrix} \nabla c(x_*)^T \\ I_{\mathcal{A}(x_*)} \end{bmatrix}$.

Then, with $H_k = H$ *for some* $H \succ 0$ *for all* $k \in \mathbb{N}$ *and with* d_k *solving* (3.2) *with* $g_k = \nabla f(x_k)$ *for all* $k \in \mathbb{N}$, x_* *satisfies* (2.4) *if and only if* $\{\|d_k\|_2^2\}_{k \in \mathcal{S}} \to 0$.

Proof. Letting $\mathcal{A}_* := \mathcal{A}(x_*)$ and considering the linear system of equations

$$\begin{bmatrix} H & \nabla c(x) & -I^T_{\mathcal{A}_*} \\ \nabla c(x)^T & 0 & 0 \\ I_{\mathcal{A}_*} & 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ y \\ z_{\mathcal{A}_*} \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \\ -x_{\mathcal{A}_*} \end{bmatrix},$$

the proof follows under the conditions of the lemma using the same line of deduction as the proof of Lemma 4.2, which we omit for the sake of brevity. ☐

Remark 4.6. One might relax the condition in Lemma 4.2 that $\mu = \mu_k$ for all $k \in \mathbb{N}$ and similarly relax the condition in Lemma 4.5 that $H_k = H \succ 0$ for all $k \in \mathbb{N}$, such as by requiring merely that $\{\mu_k\}_{k \in \mathcal{S}}$ and $\{H_k\}_{k \in \mathcal{S}}$ have bounded subsequences that converge to some $\mu \in \mathbb{R}_{>0}$ and $H \succ 0$, respectively. In these cases, the "if and only if" statements would be replaced by "if" statements, which, in fact, is all that is needed for our subsequent analysis and discussions. Nevertheless, for brevity in the proofs, we provide the conditions that offer the stronger conclusions in these lemmas.

4.2. General algorithm behavior. We now prove generally applicable results that hold for arbitrary $k \in \mathbb{N}$ in every run of Algorithm 3.1. Our initial results in this section presume that iteration $k \in \mathbb{N}$ is reached, at which point certain properties hold, e.g., $x_k \in \mathbb{R}^n_{\geq 0}$. Ultimately, we combine these results to prove inductively that, in fact, these properties are guaranteed in any run for any generated $k \in \mathbb{N}$; see Lemma 4.14. It is worthwhile to emphasize that the results in this section merely require that $g_k \in \mathbb{R}^n$ for all $k \in \mathbb{N}$, which means, for example, that Assumption 2.2 is not needed in this section. All results that depend on the properties and effects of the stochastic gradient estimates are found in the subsequent subsection, i.e., section 4.3.

Our first lemma follows directly from Lemma 4.1, so it is stated without proof.

LEMMA 4.7. *Suppose that Assumption* 2.1 *holds. Then, in any run of the algorithm such that iteration* $k \in \mathbb{N}$ *is reached and* $x_k \in \mathbb{R}^n_{\geq 0}$, *it holds that* $v_k = 0$ *if and only if* x_k *satisfies* (2.5); *i.e.,* x_k *is either feasible or an infeasible stationary point, whereas* $v_k \neq 0$ *if and only if* $\|c_k\|_2 > \|c_k + \nabla c(x_k)^T v_k\|_2$.

Our next result shows that, in any iteration in which the current iterate x_k is in the nonnegative orthant and $\tau_{k-1} > 0$, the merit parameter is either kept at the same value or decreased, and, if it is decreased, then it is decreased below a constant fraction times its former value. As in other SQP methods with such a feature, this ensures that, if the merit parameter sequence does not vanish (i.e., its limiting value is nonzero), then it eventually remains at a constant positive value; see Lemma 4.14.

LEMMA 4.8. *Suppose that Assumption* 2.1 *holds. In any run of the algorithm such that line* 4 *of iteration* $k \in \mathbb{N}$ *is reached,* $x_k \in \mathbb{R}^n_{\geq 0}$, *and* $\tau_{k-1} \in \mathbb{R}_{>0}$, *it holds that* $0 < \tau_k \leq \tau_{k-1}$, *where, if* $\tau_k < \tau_{k-1}$, *then* $\tau_k \leq (1-\epsilon_\tau)\tau_{k-1}$.

Proof. Consider an arbitrary run in which line 4 of iteration $k \in \mathbb{N}$ is reached, $x_k \in \mathbb{R}^n_{\geq 0}$, and $\tau_{k-1} \in \mathbb{R}_{>0}$. Let us show that $0 < \tau_k \leq \tau_{k-1}$, in which case the fact that $\tau_k < \tau_{k-1}$ implies $\tau_k \leq (1-\epsilon_\tau)\tau_{k-1}$ follows from (3.5). Toward this end, let us next show that $\tau_k^{\text{trial}} > 0$. By the constraints of (3.2), (3.4), and Lemma 4.7, one finds that $\tau_k^{\text{trial}} > 0$ whenever $\|c_k\|_2 - \|c_k + \nabla c(x_k)^T v_k\|_2 > 0$. Hence, to show that one always finds $\tau_k^{\text{trial}} > 0$, all that remains is to consider the case when $\|c_k\|_2 - \|c_k + \nabla c(x_k)^T v_k\|_2 = 0$. In this case, it follows from Lemma 4.7 that $v_k = 0$, meaning that $d = 0$ is feasible for (3.2). This, in turn, means that $g_k^T d_k + \frac{1}{2} d_k^T H_k d_k \leq 0$, so, by (3.4), one finds that $\tau_k^{\text{trial}} = \infty > 0$. Since it has been shown that $\tau_k^{\text{trial}} > 0$, the fact that $0 < \tau_k \leq \tau_{k-1}$ now follows directly from (3.5), completing the proof. ∎

We now show that the model reduction offered by the computed search direction satisfies a lower bound with the properties stated in our algorithm development.

LEMMA 4.9. *Suppose that Assumptions* 2.1 *and* 2.3 *hold. In any run of the algorithm such that line* 4 *is reached in iteration* $k \in \mathbb{N}$, $x_k \in \mathbb{R}^n_{\geq 0}$, *and* $\tau_k \in \mathbb{R}_{>0}$, *one finds, with* ζ *from Assumption* 2.3, *that*

$$\Delta l(x_k, \tau_k, g_k, d_k) \geq \frac{1}{2}\tau_k \zeta \|d_k\|_2^2 + \sigma(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T d_k\|_2), \tag{4.10}$$

and, if $d_k \neq 0$, *then* $\Delta l(x_k, \tau_k, g_k, d_k) > 0$.

Proof. Consider an arbitrary run in which line 4 of iteration $k \in \mathbb{N}$ is reached, $x_k \in \mathbb{R}^n_{\geq 0}$, and $\tau_k \in \mathbb{R}_{>0}$. By (3.3) and Assumption 2.3, (4.10) is implied by

$$(1-\sigma)(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T d_k\|_2) \geq \tau_k \left(g_k^T d_k + \frac{1}{2} d_k^T H_k d_k \right). \tag{4.11}$$

If $g_k^T d_k + \frac{1}{2} d_k^T H_k d_k \leq 0$, then (4.11) holds due to Lemma 4.7 and the fact that (3.2) ensures $\nabla c(x_k)^T v_k = \nabla c(x_k)^T d_k$. On the other hand, if $g_k^T d_k + \frac{1}{2} d_k^T H_k d_k > 0$, then one finds, by (3.4) and (3.5), that $\tau_k \leq \tau_k^{\text{trial}} = \frac{(1-\sigma)(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T d_k\|_2)}{g_k^T d_k + \frac{1}{2} d_k^T H_k d_k}$, from which (4.11) follows again. Finally, that $d_k \neq 0$ implies $\Delta l(x_k, \tau_k, g_k, d_k) > 0$ follows from (4.10), $\tau_k \in \mathbb{R}_{>0}$, and $\zeta \in \mathbb{R}_{>0}$ in Assumption 2.3. ∎

Our next result is that, under the same conditions as Lemma 4.9 and under the assumption that $\xi_{k-1} \in \mathbb{R}_{>0}$, the ratio parameter is either kept at the same value or decreased, and, like the merit parameter, if it is decreased, then it is decreased at least below a constant fraction times its previous value.

Lemma 4.10. *Suppose that Assumptions* 2.1 *and* 2.3 *hold. In any run of the algorithm such that line 4 is reached in iteration* $k \in \mathbb{N}$, $x_k \in \mathbb{R}^n_{\geq 0}$, $\tau_k \in \mathbb{R}_{>0}$, *and* $\xi_{k-1} \in \mathbb{R}_{>0}$, *it holds that* $0 < \xi_k \leq \xi_{k-1}$, *where, if* $\xi_k < \xi_{k-1}$, *then* $\xi_k \leq (1-\epsilon_\xi)\xi_{k-1}$.

Proof. Consider an arbitrary run in which line 4 of iteration $k \in \mathbb{N}$ is reached, $x_k \in \mathbb{R}^n_{\geq 0}$, $\tau_k \in \mathbb{R}_{>0}$, and $\xi_{k-1} \in \mathbb{R}_{>0}$. Let us show that $0 < \xi_k \leq \xi_{k-1}$, in which case the fact that $\xi_k < \xi_{k-1}$ implies $\xi_k \leq (1-\epsilon_\xi)\xi_{k-1}$ follows from (3.6). Toward this end, observe that, if $d_k = 0$, then the algorithm sets $\xi_k \leftarrow \xi_{k-1} > 0$, which is consistent with the desired conclusion. On the other hand, if $d_k \neq 0$, then, by (3.6), $\tau_k \in \mathbb{R}_{>0}$, Lemma 4.7, the fact that (3.2) ensures $\nabla c(x_k)^T v_k = \nabla c(x_k)^T d_k$, and Lemma 4.9,

$$\xi_k^{\text{trial}} = \frac{\Delta l(x_k, \tau_k, g_k, d_k)}{\tau_k \|d_k\|_2^2} \geq \frac{\frac{1}{2}\tau_k \zeta \|d_k\|_2^2}{\tau_k \|d_k\|_2^2} = \frac{1}{2}\zeta > 0. \tag{4.12}$$

Hence, by (3.6), the desired conclusion follows. ∎

Next, we prove bounds for the step size computed in the algorithm.

Lemma 4.11. *Suppose that Assumptions* 2.1 *and* 2.3 *hold. In any run of the algorithm such that line 4 is reached in iteration* $k \in \mathbb{N}$, $x_k \in \mathbb{R}^n_{\geq 0}$, $\tau_k \in \mathbb{R}_{>0}$, *and* $\xi_k \in \mathbb{R}_{>0}$, *it holds that* $0 < \alpha_k^{\min} \leq \alpha_k^{\max} \leq \min\{1, \alpha_k^\varphi\}$, *and so,* $x_{k+1} \in \mathbb{R}^n_{\geq 0}$.

Proof. Consider an arbitrary run of the algorithm in which line 4 of iteration $k \in \mathbb{N}$ is reached, $x_k \in \mathbb{R}^n_{\geq 0}$, $\tau_k \in \mathbb{R}_{>0}$, and $\xi_k \in \mathbb{R}_{>0}$. Let us show that $0 < \alpha_k^{\min} \leq \alpha_k^{\max} \leq 1$, in which case the fact that $x_{k+1} \in \mathbb{R}^n_{\geq 0}$ follows from $x_k \in \mathbb{R}^n_{\geq 0}$, the fact that the constraints of (3.2) ensure that $x_k + d_k \in \mathbb{R}^n_{\geq 0}$, and the step size having $\alpha_k \in [\alpha_k^{\min}, \alpha_k^{\max}] \subset (0,1]$. Toward this end, observe that, if $d_k = 0$, then the algorithm yields $\alpha_k = \alpha_k^{\min} = \alpha_k^{\max} = \alpha_k^\varphi = 1$, so the conclusion follows trivially. Hence, let us assume $d_k \neq 0$. Observe that, from (3.7), the algorithm uses $\alpha_k^{\min}$ with

$$0 < \alpha_k^{\min} = \frac{2(1-\eta)\beta_k \xi_k \tau_k}{\tau_k L + \Gamma} \leq 1. \tag{4.13}$$

Now, observing (3.9), which shows that $\alpha_k^{\max} \leq \min\{1, \alpha_k^\varphi\}$, one finds that all that remains is to prove that $\alpha_k^{\min} \leq \alpha_k^\varphi$. Let us introduce $\alpha_k^{\text{suff}} := \min\{1, \frac{2(1-\eta)\beta_k \Delta l(x_k, \tau_k, g_k, d_k)}{(\tau_k L + \Gamma)\|d_k\|_2^2}\}$, where $\alpha_k^{\text{suff}} \in (0,1]$ follows by $\beta_k \in (0,1]$, Lemma 4.9, and $d_k \neq 0$. To show that $\alpha_k^{\min} \leq \alpha_k^\varphi$, our aim is to show that $\alpha_k^{\min} \leq \alpha_k^{\text{suff}} \leq \alpha_k^\varphi$. First, from (3.6), one finds that

$$\alpha_k^{\min} = \frac{2(1-\eta)\beta_k \xi_k \tau_k}{\tau_k L + \Gamma} \leq \frac{2(1-\eta)\beta_k \xi_k^{\text{trial}} \tau_k}{\tau_k L + \Gamma} = \frac{2(1-\eta)\beta_k \Delta l(x_k, \tau_k, g_k, d_k)}{(\tau_k L + \Gamma)\|d_k\|_2^2}. \tag{4.14}$$

Combining (4.13) and (4.14), one finds that $\alpha_k^{\min} \leq \alpha_k^{\text{suff}}$, as desired. Now, toward proving that $\alpha_k^{\text{suff}} \leq \alpha_k^\varphi$, let us first show that $\varphi_k(\alpha_k^{\text{suff}}) \leq 0$. From the triangle inequality, the fact that $\alpha_k^{\text{suff}} \in (0,1]$, and (3.8), it follows that

$$\begin{aligned}
\varphi_k(\alpha_k^{\text{suff}}) &= (\eta-1)\alpha_k^{\text{suff}} \beta_k \Delta l(x_k, \tau_k, g_k, d_k) + \|c_k + \alpha_k^{\text{suff}} \nabla c(x_k)^T d_k\|_2 - \|c_k\|_2 \\
&\quad + \alpha_k^{\text{suff}}(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T d_k\|_2) + \frac{1}{2}(\tau_k L + \Gamma)(\alpha_k^{\text{suff}})^2 \|d_k\|_2^2 \\
&\leq (\eta-1)\alpha_k^{\text{suff}} \beta_k \Delta l(x_k, \tau_k, g_k, d_k) + (1-\alpha_k^{\text{suff}})\|c_k\|_2 + \alpha_k^{\text{suff}} \|c_k + \nabla c(x_k)^T d_k\|_2
\end{aligned}$$

$$
\begin{aligned}
&- \|c_k\|_2 + \alpha_k^{\rm suff}(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T d_k\|_2) + \frac{1}{2}(\tau_k L + \Gamma)(\alpha_k^{\rm suff})^2 \|d_k\|_2^2 \\
&= (\eta - 1)\alpha_k^{\rm suff}\beta_k \Delta l(x_k, \tau_k, g_k, d_k) + \frac{1}{2}(\tau_k L + \Gamma)(\alpha_k^{\rm suff})^2\|d_k\|_2^2 \\
&\le (\eta - 1)\alpha_k^{\rm suff}\beta_k \Delta l(x_k, \tau_k, g_k, d_k) \\
&\quad + \frac{1}{2}\alpha_k^{\rm suff}(\tau_k L + \Gamma)\|d_k\|_2^2 \left(\frac{2(1-\eta)\beta_k \Delta l(x_k, \tau_k, g_k, d_k)}{(\tau_k L + \Gamma)\|d_k\|_2^2}\right) = 0.
\end{aligned}
$$

Therefore, by (3.9), it follows that $\alpha_k^{\rm suff} \le \alpha_k^{\varphi}$. ∎

Our next lemma shows an upper bound on the change in the merit function. In the next lemma and throughout the rest of the paper, for any $k \in \mathbb{N}$ such that line 4 is reached, we let $d_k^{\rm true} \in \mathbb{R}^n$ denote the solution of (3.2) when g_k is replaced by $\nabla f(x_k)$.

LEMMA 4.12. *Suppose that Assumptions* 2.1 *and* 2.3 *hold. In any run of the algorithm such that line* 4 *is reached in iteration* $k \in \mathbb{N}$, $x_k \in \mathbb{R}_{\ge 0}$, $\tau_k \in \mathbb{R}_{>0}$, *and* $\alpha_k \in (0, \alpha_k^{\varphi}]$, *it holds that*

$$
\begin{aligned}
\phi(x_k + \alpha_k d_k, \tau_k) - \phi(x_k, \tau_k) \le & -\alpha_k \Delta l(x_k, \tau_k, \nabla f(x_k), d_k^{\rm true}) + \alpha_k \tau_k \nabla f(x_k)^T (d_k - d_k^{\rm true}) \\
& + (1-\eta)\alpha_k \beta_k \Delta l(x_k, \tau_k, g_k, d_k).
\end{aligned}
$$

Proof. Consider an arbitrary run of the algorithm in which line 4 of iteration $k \in \mathbb{N}$ is reached, $x_k \in \mathbb{R}^n_{\ge 0}$, $\tau_k \in \mathbb{R}_{>0}$, and $\alpha_k \in (0, \alpha_k^{\varphi}]$. By Assumption 2.1 (which led to (2.3)), (3.2) (which implies that $c_k + \nabla c(x_k)^T d_k = c_k + \nabla c(x_k)^T d_k^{\rm true}$), (3.3), (3.8), and the fact that $0 < \alpha_k \le \alpha_k^{\varphi}$ (which means that $\varphi_k(\alpha_k) \le 0$), it follows that

$$
\begin{aligned}
&\phi(x_k + \alpha_k d_k, \tau_k) - \phi(x_k, \tau_k) \\
&\quad = \tau_k (f(x_k + \alpha_k d_k) - f_k) + \|c(x_k + \alpha_k d_k)\|_2 - \|c_k\|_2 \\
&\quad \le \alpha_k \tau_k \nabla f(x_k)^T d_k + \|c_k + \alpha_k \nabla c(x_k)^T d_k\|_2 - \|c_k\|_2 + \frac{1}{2}(\tau_k L + \Gamma)\alpha_k^2 \|d_k\|_2^2 \\
&\quad = -\alpha_k \Delta l(x_k, \tau_k, \nabla f(x_k), d_k^{\rm true}) + \alpha_k \tau_k \nabla f(x_k)^T (d_k - d_k^{\rm true}) + \|c_k + \alpha_k \nabla c(x_k)^T d_k\|_2 \\
&\qquad - \|c_k\|_2 + \alpha_k(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T d_k\|_2) + \frac{1}{2}(\tau_k L + \Gamma)\alpha_k^2\|d_k\|_2^2 \\
&\quad \le -\alpha_k \Delta l(x_k, \tau_k, \nabla f(x_k), d_k^{\rm true}) + \alpha_k \tau_k \nabla f(x_k)^T (d_k - d_k^{\rm true}) \\
&\qquad + (1-\eta)\alpha_k \beta_k \Delta l(x_k, \tau_k, g_k, d_k),
\end{aligned}
$$

which shows the desired conclusion. ∎

We now show that each search direction—and, similarly, the search direction that would be computed if the true gradient of the objective function were used in place of the stochastic gradient estimate—can be viewed as a projection of the unconstrained minimizer of the objective of (3.2) onto the feasible region of (3.2).

LEMMA 4.13. *Suppose that Assumptions* 2.1 *and* 2.3 *hold. In any run of the algorithm such that line* 4 *is reached in iteration* $k \in \mathbb{N}$, $x_k \in \mathbb{R}_{\ge 0}$, *and with*

$$
\mathcal{D}_k := \{d \in \mathbb{R}^n : \nabla c(x_k)^T (d - v_k) = 0,\ x_k + d \ge 0\} \text{ and } \mathrm{Proj}_k(\bar{D}) := \operatorname*{argmin}_{d \in \mathcal{D}_k} \|d - \bar{D}\|_{H_k}^2,
$$

it holds that $d_k = \mathrm{Proj}_k(-H_k^{-1} g_k)$ *and* $d_k^{\rm true} = \mathrm{Proj}_k(-H_k^{-1}\nabla f(x_k))$.

Proof. Consider an arbitrary run of the algorithm in which line 4 of iteration $k \in \mathbb{N}$ is reached and $x_k \in \mathbb{R}_{\ge 0}$. The desired conclusion follows from the facts that $\mathcal{D}_k$ is convex and, under Assumption 2.3, H_k is SPD; in particular, one finds that

$$d_k = \underset{d\in\mathcal{D}_k}{\operatorname{argmin}}\ g_k^T d + \frac{1}{2} d^T H_k d = \underset{d\in\mathcal{D}_k}{\operatorname{argmin}}\ \frac{1}{2}\|d + H_k^{-1} g_k\|_{H_k}^2 = \operatorname{Proj}_k(-H_k^{-1} g_k),$$

and similarly with respect to $d_k^{\rm true}$ with g_k replaced by $\nabla f(x_k)$. ∎

We are now prepared to prove Lemma 4.14, which shows that the algorithm is well defined and either terminates finitely with an infeasible stationary point or generates an infinite sequence of iterates with certain critical properties of the simultaneously generated algorithmic sequences. Lemma 4.14 also reveals that the monotonically nonincreasing merit parameter sequence either vanishes or ultimately remains constant, and it reveals that the monotonically nonincreasing ratio parameter sequence ultimately remains constant at a value that is greater than or equal to a positive real number that is defined uniformly across all runs of the algorithm.

LEMMA 4.14. *Suppose that Assumptions* 2.1 *and* 2.3 *hold. In any run, either the algorithm terminates finitely with an infeasible stationary point, or it performs an infinite number of iterations such that, for all* $k \in \mathbb{N}$, *it holds that*

(a) $x_k \in \mathbb{R}^n_{\geq 0}$,
(b) $v_k = 0$ *if and only if* x_k *satisfies* (2.5),
(c) $v_k \neq 0$ *if and only if* $\|c_k\|_2 > \|c_k + \nabla c(x_k)^T v_k\|_2$,
(d) $0 < \tau_k \leq \tau_{k-1} < \infty$,
(e) $\tau_k < \tau_{k-1}$ *if and only if* $\tau_k \leq (1-\epsilon_\tau)\tau_{k-1}$,
(f) (4.10) *holds,*
(g) $d_k \neq 0$ *if and only if* $\Delta l(x_k, \tau_k, g_k, d_k) > 0$,
(h) $0 < \xi_k \leq \xi_{k-1} < \infty$,
(i) $\xi_k < \xi_{k-1}$ *if and only if* $\xi_k \leq (1-\epsilon_\xi)\xi_{k-1}$, *and*
(j) $0 < \alpha_k^{\min} \leq \alpha_k^{\max} \leq \min\{1, \alpha_k^\varphi\}$.

In addition, in any run that does not terminate finitely, it holds that

(k) *either* $\{\tau_k\} \searrow 0$ *or there exists* $k_\tau \in \mathbb{N}$ *and* $\tau_{\min} \in \mathbb{R}_{>0}$ *such that* $\tau_k = \tau_{\min}$ *for all* $k \in \mathbb{N}$ *with* $k \geq k_\tau$, *and*
(l) *there exist* $k_\xi \in \mathbb{N}$ *and* $\xi_{\min} \in \mathbb{R}_{>0}$ *with* $\xi_{\min} \geq \frac{1}{2}\zeta(1-\epsilon_\xi)$ *such that* $\xi_k = \xi_{\min}$ *for all* $k \in \mathbb{N}$ *with* $k \geq k_\xi$.

Proof. Given the initialization of the algorithm, statements (a)–(j) follow by induction from Lemmas 4.7–4.11. Statement (k) follows from statements (d) and (e). Finally, to prove statement (l), consider arbitrary $k \in \mathbb{N}$ in a run that does not terminate finitely and note that, if $d_k = 0$, then $\xi_k^{\rm trial} \leftarrow \infty$, and if $d_k \neq 0$, then $\xi_k^{\rm trial}$ satisfies (4.12), meaning that $\xi_k^{\rm trial} \geq \frac{1}{2}\zeta$. Consequently, by (3.6), $\xi_k < \xi_{k-1}$ only if $\xi_{k-1} > \frac{1}{2}\zeta$. This, along with statements (h) and (i), leads to the conclusion. ∎

4.3. Convergence guarantees. We now turn to prove convergence results under Assumption 4.15 below. Recalling the role of $\frac{1}{2}\zeta(1-\epsilon_\xi) \in \mathbb{R}_{>0}$ in Lemma 4.14(l), the assumption focuses on the following event for some $(k_{\min}, \tau_{\min}, f_{\sup}) \in \mathbb{N} \times \mathbb{R}_{>0} \times \mathbb{R}$, where, for all generated $k \in \mathbb{N}$, we denote $\tau_k^{\rm true,trial}$ as the value of $\tau_k^{\rm trial}$ that would be computed in iteration k if (3.2) were solved with $\nabla f(x_k)$ in place of g_k:

$$\begin{aligned}
&\mathcal{E}(k_{\min}, \tau_{\min}, f_{\sup}) \\
&\quad := \{\text{An infinite number of iterations are performed, } f(x_{k_{\min}}) \leq f_{\sup}, \text{ and} \\
&\qquad\quad \text{there exist } k' \in \mathbb{N} \text{ with } k' \leq k_{\min},\ \tau' \in \mathbb{R}_{>0} \text{ with } \tau' \geq \tau_{\min}, \\
&\qquad\quad \text{and } \xi' \in \mathbb{R}_{>0} \text{ with } \xi' \geq \tfrac{1}{2}\zeta(1-\epsilon_\xi) \text{ such that} \\
&\qquad\quad \tau_k = \tau' \leq \tau_k^{\rm true,trial} \text{ and } \xi_k = \xi' \text{ for all } k \in \mathbb{N} \text{ with } k \geq k'\}.
\end{aligned} \tag{4.15}$$

The following assumption is made in this subsection. We present a discussion and supporting theoretical results about this assumption in section 4.4.

Assumption 4.15. For some $(k_{\min}, \tau_{\min}, f_{\sup}) \in \mathbb{N} \times \mathbb{R}_{>0} \times \mathbb{R}$, the event $\mathcal{E} := \mathcal{E}(k_{\min}, \tau_{\min}, f_{\sup})$ occurs and, conditioned on the occurrence of $\mathcal{E}$, Assumption 2.1 holds (with the same constants as previously presented in (2.2) and (2.3)).

It is not a shortcoming of our analysis that Assumption 4.15, through the definition of $\mathcal{E}$, assumes that (i) an infinite number of iterations are performed, (ii) the objective value is bounded above in iteration $k_{\min}$, and (iii) $\{\xi_k\}$ ultimately becomes a constant sequence with value at least $\frac{1}{2}\zeta(1-\epsilon_\xi) \in \mathbb{R}_{>0}$. After all, (i) Lemma 4.14 shows that the only alternative to an infinite number of iterations being performed is that the algorithm terminates finitely with an infeasible stationary point, in which case there is nothing else to prove; (ii) $f_{\sup} \in \mathbb{R}$ can be arbitrarily large, and knowledge of it is not required by the algorithm, so assuming that it exists is a very loose requirement; and (iii) Lemma 4.14(l) shows that, in any run that does not terminate finitely, $\{\xi_k\}$ is monotonically nonincreasing and bounded below by $\frac{1}{2}\zeta(1-\epsilon_\xi) \in \mathbb{R}_{>0}$, which is a constant (i.e., it is not run-dependent). Overall, the only important restriction of our analysis in this section is the fact that $\mathcal{E}$ includes the requirement that $\{\tau_k\}$ ultimately becomes constant at a value at least $\tau_{\min}$ that is sufficiently small relative to $\{\tau_k^{\text{true,trial}}\}$. This restriction is the subject of section 4.4.

For the remainder of this subsection, we consider the stochastic process corresponding to the statement of Algorithm 3.1. Specifically, the sequence

$$\{(x_k, v_k, g_k, d_k, d_k^{\text{true}}, \tau_k^{\text{trial}}, \tau_k^{\text{true,trial}}, \tau_k, \xi_k^{\text{trial}}, \xi_k, \alpha_k^{\min}, \alpha_k^{\varphi}, \alpha_k^{\max}, \alpha_k)\}$$

generated in any run can be viewed as a realization of the stochastic process

$$\{(X_k, V_k, G_k, D_k, D_k^{\text{true}}, \mathcal{T}_k^{\text{trial}}, \mathcal{T}_k^{\text{true,trial}}, \mathcal{T}_k, \Xi_k^{\text{trial}}, \Xi_k, \mathcal{A}_k^{\min}, \mathcal{A}_k^{\varphi}, \mathcal{A}_k^{\max}, \mathcal{A}_k)\}.$$

Let $\mathcal{G}_1$ denote the σ-algebra defined by the initial conditions of the algorithm and, for all $k \in \mathbb{N}$ with $k \geq 2$, let $\mathcal{G}_k$ denote the σ-algebra generated by the initial conditions and the random variables $\{G_1, \ldots, G_{k-1}\}$. Then, with respect to the event $\mathcal{E}$ in Assumption 4.15, denote the trace σ-algebra of $\mathcal{E}$ on $\mathcal{G}_k$ as $\mathcal{F}_k := \mathcal{G}_k \cap \mathcal{E}$ for all $k \in \mathbb{N}$. It follows that $\{\mathcal{F}_k\}$ is a filtration, and we proceed in our analysis under Assumptions 2.2, 2.3, and 4.15 (which subsumes Assumption 2.1) with the definitions that $\mathbb{P}_k[\cdot] := \mathbb{P}_\omega[\cdot|\mathcal{F}_k]$ and $\mathbb{E}_k[\cdot] := \mathbb{E}_\omega[\cdot|\mathcal{F}_k]$. We also define, with respect to $\mathcal{E}$, the random variables $K' \leq k_{\min}$, $\mathcal{T}' \geq \tau_{\min}$, and $\Xi' \geq \frac{1}{2}\zeta(1-\epsilon_\xi)$, which, for a given run of the algorithm, have the realized values k', τ', and ξ', respectively, defined in (4.15). Conditioned on $\mathcal{E}$, one has, in any run, that

$$\tau_{\min} \leq \mathcal{T}' \leq \tau_0 \quad \text{and} \quad \frac{1}{2}\zeta(1-\epsilon_\xi) \leq \Xi' \leq \xi_0 \tag{4.16}$$

and that $\mathcal{T}'$ and Ξ' are $\mathcal{F}_k$-measurable for $k = k_{\min} \geq K'$.

Our next lemma shows upper bounds on the norm of the difference between the computed search direction and the search direction that would be computed with the true gradient of the objective. (The conclusion of this lemma would hold even without assuming that the event $\mathcal{E}$ occurs, but in the result, we condition on $\mathcal{F}_k := \mathcal{G}_k \cap \mathcal{E}$ so that it may be used directly in our ultimate results under $\mathcal{E}$.)

LEMMA 4.16. *Suppose that Assumptions* 2.2, 2.3, *and* 4.15 *hold. For all* $k \in \mathbb{N}$,

$$\begin{aligned} &\|D_k - D_k^{\text{true}}\|_2 \leq \zeta^{-1}\|G_k - \nabla f(X_k)\|_2 \\ \textit{and}\ \ &\mathbb{E}_k[\|D_k - D_k^{\text{true}}\|_2] \leq \zeta^{-1}\mathbb{E}_k[\|G_k - \nabla f(X_k)\|_2] \leq \zeta^{-1}\sqrt{\rho_k}. \end{aligned}$$

Proof. Consider arbitrary $k \in \mathbb{N}$ under the stated conditions. Lemma 4.13 and the obtuse angle lemma for projections [6, Proposition 1.1.9] imply that $(D_k - D_k^{\rm true})^T H_k(-H_k^{-1}\nabla f(X_k) - D_k^{\rm true}) \leq 0$ and $(D_k^{\rm true} - D_k)^T H_k(-H_k^{-1}G_k - D_k) \leq 0$. Summing these inequalities yields

$$\begin{aligned} 0 &\geq (D_k - D_k^{\rm true})^T H_k(-H_k^{-1}\nabla f(X_k) - D_k^{\rm true}) + (D_k^{\rm true} - D_k)^T H_k(-H_k^{-1}G_k - D_k) \\ &= \|D_k - D_k^{\rm true}\|_{H_k}^2 - (D_k - D_k^{\rm true})^T(\nabla f(X_k) - G_k). \end{aligned}$$

Hence, by the Cauchy–Schwarz inequality, it follows that $\|D_k - D_k^{\rm true}\|_{H_k}^2 \leq (D_k - D_k^{\rm true})^T(\nabla f(X_k) - G_k) \leq \|D_k - D_k^{\rm true}\|_2\|\nabla f(X_k) - G_k\|_2$, which shows, under Assumption 2.3, that $\|D_k - D_k^{\rm true}\|_2 \leq \zeta^{-1}\|G_k - \nabla f(X_k)\|_2$, as desired. Then, from this inequality, Assumption 2.2, and Jensen's inequality, one has that $\mathbb{E}_k[\|G_k - \nabla f(X_k)\|_2] \leq \sqrt{\mathbb{E}_k[\|G_k - \nabla f(X_k)\|_2^2]} \leq \sqrt{\rho_k}$, from which the remainder of the conclusion follows. ◻

We now show an upper bound on the expected difference between inner products involving the true and stochastic gradients and the true and stochastic directions.

Lemma 4.17. *Suppose that Assumptions* 2.2, 2.3, *and* 4.15 *hold. For all* $k \geq k_{\min}$,

$$\begin{aligned} &|\mathbb{E}_k[G_k^T D_k - \nabla f(X_k)^T D_k^{\rm true}]| \leq \zeta^{-1}(\rho_k + \kappa_{\nabla f}\sqrt{\rho_k}) \\ \textit{and}\ \ &\mathbb{E}_k[\Delta l(X_k, \mathcal{T}_k, G_k, D_k)] - \Delta l(X_k, \mathcal{T}', \nabla f(X_k), D_k^{\rm true}) \leq \mathcal{T}'\zeta^{-1}(\rho_k + \kappa_{\nabla f}\sqrt{\rho_k}). \end{aligned}$$

Proof. Consider arbitrary $k \geq k_{\min}$ under the stated conditions. From the triangle and Cauchy–Schwarz inequalities and Lemma 4.16, it holds that

$$\begin{aligned} &|\mathbb{E}_k[G_k^T D_k - \nabla f(X_k)^T D_k^{\rm true}]| \\ &\quad = |\mathbb{E}_k[(G_k - \nabla f(X_k))^T D_k^{\rm true} + (G_k - \nabla f(X_k))^T(D_k - D_k^{\rm true}) \\ &\qquad + \nabla f(X_k)^T(D_k - D_k^{\rm true})]| \\ &\quad = |\mathbb{E}_k[(G_k - \nabla f(X_k))^T(D_k - D_k^{\rm true})] + \mathbb{E}_k[\nabla f(X_k)^T(D_k - D_k^{\rm true})]| \\ &\quad \leq \mathbb{E}_k[\|G_k - \nabla f(X_k)\|_2\|D_k - D_k^{\rm true}\|_2] + \|\nabla f(X_k)\|_2\mathbb{E}_k[\|D_k - D_k^{\rm true}\|_2] \\ &\quad \leq \zeta^{-1}\mathbb{E}_k[\|G_k - \nabla f(X_k)\|_2^2] + \zeta^{-1}\kappa_{\nabla f}\mathbb{E}_k[\|G_k - \nabla f(X_k)\|_2] \leq \zeta^{-1}\rho_k + \zeta^{-1}\kappa_{\nabla f}\sqrt{\rho_k}, \end{aligned}$$

which gives the first result. Then, for $k \geq k_{\min}$, (3.3) and the equation above give

$$\begin{aligned} &\mathbb{E}_k[\Delta l(X_k, \mathcal{T}_k, G_k, D_k)] - \Delta l(X_k, \mathcal{T}', \nabla f(X_k), D_k^{\rm true}) \\ &\quad = \mathcal{T}'\mathbb{E}_k[\nabla f(X_k)^T D_k^{\rm true} - G_k^T D_k] \leq \mathcal{T}'\zeta^{-1}(\rho_k + \kappa_{\nabla f}\sqrt{\rho_k}), \end{aligned}$$

which completes the proof. ◻

Our next lemma shows a lower bound on the true model reduction. In Lemma 4.18 and our subsequent results, we define $J_k := \nabla c(X_k)^T$ for the sake of brevity.

Lemma 4.18. *Suppose that Assumptions* 2.2, 2.3, *and* 4.15 *hold. For all* $k \geq k_{\min}$,

$$\Delta l(X_k, \mathcal{T}_k, \nabla f(X_k), D_k^{\rm true}) \geq \frac{1}{2}\mathcal{T}'\zeta\|D_k^{\rm true}\|_2^2 + \sigma(\|c(X_k)\|_2 - \|c(X_k) + J_k D_k^{\rm true}\|_2) \geq 0.$$

Proof. Consider arbitrary $k \geq k_{\min}$ under the stated conditions. By (3.3), the fact that $\mathcal{T}_k = \mathcal{T}'$, and Assumption 2.3, the first desired conclusion is implied by

$$(1 - \sigma)(\|c(X_k)\|_2 - \|c(X_k) + J_k D_k^{\rm true}\|_2) \geq \mathcal{T}'(\nabla f(X_k)^T D_k^{\rm true} + \frac{1}{2}(D_k^{\rm true})^T H_k D_k^{\rm true}).$$

If $\nabla f(X_k)^T D_k^{\rm true} + \frac{1}{2}(D_k^{\rm true})^T H_k D_k^{\rm true} \leq 0$, then the above holds due to Lemma 4.14 and the fact that $J_k D_k^{\rm true} = J_k V_k$; else, $\nabla f(X_k)^T D_k^{\rm true} + \frac{1}{2}(D_k^{\rm true})^T H_k D_k^{\rm true} > 0$, in which case one finds, from the conditions of Lemma 4.18, (3.4), and (3.5), that

$\mathcal{T}_k = \mathcal{T}' \leq \mathcal{T}_k^{\text{true,trial}} = \frac{(1-\sigma)(\|c(X_k)\|_2 - \|c(X_k)+J_k D_k^{\text{true}}\|_2)}{\nabla f(X_k)^T D_k^{\text{true}} + \frac{1}{2}(D_k^{\text{true}})^T H_k D_k^{\text{true}}}$, from which the displayed inequality above follows again. Finally, the remaining desired conclusion follows from the first conclusion, Lemma 4.14, and $J_k D_k^{\text{true}} = J_k V_k$. ∎

Next, we prove a critical upper bound on the expected value of the second term on the right-hand side of the upper bound proved in Lemma 4.12.

LEMMA 4.19. *Suppose that Assumptions* 2.2, 2.3, *and* 4.15 *hold. For all* $k \geq k_{\min}$, $\mathbb{E}_k[\mathcal{A}_k \mathcal{T}_k \nabla f(X_k)^T (D_k - D_k^{\text{true}})] \leq (\frac{2(1-\eta)\Xi'\mathcal{T}'}{\mathcal{T}'L+\Gamma} + \theta)\beta_k \mathcal{T}' \kappa_{\nabla f} \zeta^{-1}\sqrt{\rho_k}$.

Proof. For arbitrary $k \geq k_{\min}$ under the conditions of Lemma 4.19, (3.7) and (3.9) yield

$$\mathcal{A}_k^{\min} = \beta_k \mathcal{A}' \ \text{ and } \ \mathcal{A}_k^{\max} \leq \mathcal{A}_k^{\min} + \theta\beta_k, \ \text{ where } \ \mathcal{A}' = \frac{2(1-\eta)\Xi'\mathcal{T}'}{\mathcal{T}'L+\Gamma}. \tag{4.17}$$

Letting $\mathcal{P}_k$ denote the event that $\nabla f(X_k)^T (D_k - D_k^{\text{true}}) \geq 0$ and letting $\mathcal{P}_k^c$ denote the event that $\nabla f(X_k)^T (D_k - D_k^{\text{true}}) < 0$, the law of total expectation and the fact that $\mathcal{T}'$ and Ξ' are $\mathcal{F}_k$-measurable for $k \geq k_{\min}$ show that

$$\begin{aligned}
&\mathbb{E}_k[\mathcal{A}_k \mathcal{T}_k \nabla f(X_k)^T (D_k - D_k^{\text{true}})] \\
&\quad = \mathbb{P}_k[\mathcal{P}_k] \cdot \mathbb{E}_k[\mathcal{A}_k \mathcal{T}' \nabla f(X_k)^T (D_k - D_k^{\text{true}}) | \mathcal{P}_k] \\
&\qquad + \mathbb{P}_k[\mathcal{P}_k^c] \cdot \mathbb{E}_k[\mathcal{A}_k \mathcal{T}' \nabla f(X_k)^T (D_k - D_k^{\text{true}}) | \mathcal{P}_k^c] \\
&\quad \leq (\mathcal{A}_k^{\min} + \theta\beta_k)\mathcal{T}' \mathbb{P}_k[\mathcal{P}_k] \cdot \mathbb{E}_k[\nabla f(X_k)^T (D_k - D_k^{\text{true}}) | \mathcal{P}_k] \\
&\qquad + \mathcal{A}_k^{\min} \mathcal{T}' \mathbb{P}_k[\mathcal{P}_k^c] \cdot \mathbb{E}_k[\nabla f(X_k)^T (D_k - D_k^{\text{true}}) | \mathcal{P}_k^c] \\
&\quad = \mathcal{A}_k^{\min} \mathcal{T}' \mathbb{E}_k[\nabla f(X_k)^T (D_k - D_k^{\text{true}})] + \theta\beta_k \mathcal{T}' \mathbb{P}_k[\mathcal{P}_k] \cdot \mathbb{E}_k[\nabla f(X_k)^T (D_k - D_k^{\text{true}}) | \mathcal{P}_k].
\end{aligned}$$

The Cauchy–Schwarz inequality and law of total expectation show that

$$\begin{aligned}
&\mathbb{P}_k[\mathcal{P}_k] \cdot \mathbb{E}_k[\nabla f(X_k)^T (D_k - D_k^{\text{true}}) | \mathcal{P}_k] \leq \mathbb{P}_k[\mathcal{P}_k] \cdot \mathbb{E}_k[\|\nabla f(X_k)\|_2 \|D_k - D_k^{\text{true}}\|_2 | \mathcal{P}_k] \\
&\quad = \mathbb{E}_k[\|\nabla f(X_k)\|_2 \|D_k - D_k^{\text{true}}\|_2] - \mathbb{P}_k[\mathcal{P}_k^c] \cdot \mathbb{E}_k[\|\nabla f(X_k)\|_2 \|D_k - D_k^{\text{true}}\|_2 | \mathcal{P}_k^c] \\
&\quad \leq \mathbb{E}_k[\|\nabla f(X_k)\|_2 \|D_k - D_k^{\text{true}}\|_2],
\end{aligned}$$

so from the above, the Cauchy–Schwarz inequality, Assumption 4.15, and Lemma 4.16,

$$\begin{aligned}
\mathbb{E}_k[\mathcal{A}_k \mathcal{T}_k \nabla f(X_k)^T (D_k - D_k^{\text{true}})] &\leq (\mathcal{A}_k^{\min} + \theta\beta_k)\mathcal{T}' \|\nabla f(X_k)\|_2 \mathbb{E}_k[\|D_k - D_k^{\text{true}}\|_2] \\
&\leq \left(\frac{2(1-\eta)\Xi'\mathcal{T}'}{\mathcal{T}'L+\Gamma} + \theta \right) \beta_k \mathcal{T}' \kappa_{\nabla f} \zeta^{-1} \sqrt{\rho_k},
\end{aligned}$$

which gives the desired conclusion. ∎

We now present, as Lemma 4.20, results pertaining to the asymptotic behavior of the model reductions generated by the algorithm. In the subsequent theorem after Lemma 4.20, these results will be translated in terms of quantities that, as seen in section 4.1, can be connected to stationarity measures related to problem (2.1). We remark that the conditions of the lemma can be satisfied in a run-dependent manner if, every time the merit or ratio parameter is decreased, say, in iteration $\hat{k} \in \mathbb{N}$, the sequence $\{\beta_k\}$ is "restarted" such that, with $\alpha' = 2(1-\eta)\xi_{\hat{k}}\tau_{\hat{k}}/(\tau_{\hat{k}}L+\Gamma)$ and some (run-independent) $\psi \in (0,1]$, one chooses $\beta_k = \beta = \psi\frac{\alpha'}{2(1-\eta)(\alpha'+\theta)}$ for part (a) of Lemma 4.20 and $\beta_k = \frac{1}{k-\hat{k}+1}\psi\frac{\alpha'}{2(1-\eta)(\alpha'+\theta)}$ for part (b); such a scheme was described in [3] as well. Notice that, in this situation, β and $\{\beta_k\}_{k\geq\hat{k}}$ in parts (a) and (b), respectively, are random variables, but importantly, they are $\mathcal{F}_k$-measurable for $k \geq k_{\min}$.

Alternatively, one could choose $\{\beta_k\}$ using the same formulas, but with $\xi_{\min}$ and $\tau_{\min}$ in place of ξ_k and τ_k, respectively, in the formula for α', in which case the choices are run independent. The downside of relying on this latter situation is that it requires knowledge of $\xi_{\min}$ and $\tau_{\min}$, which would not typically be known a priori. Hence, we analyze the former scheme but use run-dependent bounds that, under $\mathcal{E}$, are defined with respect to $\xi_{\min}$ and $\tau_{\min}$. These values are unknown by the algorithm but nonetheless can be employed for our theoretical analysis.

We also remark that, for case (a) in Lemma 4.20, the sequence $\{\rho_k\}$, which bounds the expected squared error in the stochastic gradient estimates, can be a constant sequence. However, for case (b), the relationship between $\{\rho_k\}$ and $\{\beta_k\}$ means that the expected squared error in the gradient estimates must vanish as $k \to \infty$. This requirement, which is stronger than the requirement for the equality-constraints-only case in [3], is needed to overcome the fact that, in the presence of bound constraints, the search directions can be biased estimates of their true counterparts. We discuss this further with an illustrative example after the proof of Lemma 4.20.

LEMMA 4.20. *Under Assumptions* 2.2, 2.3, *and* 4.15, *suppose that* $\{\rho_k\}$ *is chosen such that there exists* $\iota \in \mathbb{R}_{>0}$ *with* $\rho_k \leq \iota\beta_k^2$ *for all* $k \in \mathbb{N}$ *with* $k \geq k_{\min}$, *and define* $\alpha'_{\min} = \frac{2(1-\eta)\xi_{\min}\tau_{\min}}{\tau_{\min}L+\Gamma}$, $\alpha'_{\max} = \frac{2(1-\eta)\xi_0\tau_0}{\tau_0 L+\Gamma}$, *and* $\rho'_{\max} = (\alpha'_{\max} + \theta)\tau_0\zeta^{-1}(\kappa_{\nabla f}\sqrt{\iota} + (1-\eta)(\iota + \kappa_{\nabla f}\sqrt{\iota}))$. *Then, with* $\mathcal{A}'$ *defined in* (4.17) *and* $\mathbb{E}[\cdot|\mathcal{E}]$ *denoting expectation over all realizations of the algorithm conditioned on* $\mathcal{E}$, *the following statements hold true.*

(a) *If* $\beta_k = \beta = \psi\frac{\mathcal{A}'}{2(1-\eta)(\mathcal{A}'+\theta)}$ *for some* $\psi \in (0,1]$ *for all* $k \geq k_{\min}$, *then*

$$\limsup_{k\to\infty}\ \mathbb{E}\left[\frac{1}{k}\sum_{j=k_{\min}}^{k_{\min}+k-1}\Delta l(X_j,\mathcal{T}',\nabla f(X_j),D_j^{\text{true}})\middle|\mathcal{E}\right] \leq \frac{\psi(\alpha'_{\max})^2(\alpha'_{\min}+\theta)\rho'_{\max}}{2(1-\eta)(1-\frac{\psi}{2})(\alpha'_{\min})^2(\alpha'_{\max}+\theta)^2};$$

(b) *if* $\sum_{k=k_{\min}}^{\infty}\beta_k = \infty$, $\sum_{k=k_{\min}}^{\infty}\beta_k^2 < \infty$, *and* $\beta_k \leq \psi\frac{\mathcal{A}'}{2(1-\eta)(\mathcal{A}'+\theta)}$ *for some* $\psi \in (0,1]$ *for all* $k \geq k_{\min}$, *it holds that*

$$\mathbb{E}\left[\frac{1}{\sum_{j=k_{\min}}^{k_{\min}+k-1}\beta_j}\sum_{j=k_{\min}}^{k_{\min}+k-1}\beta_j\Delta l(X_j,\mathcal{T}',\nabla f(X_j),D_j^{\text{true}})\middle|\mathcal{E}\right]\xrightarrow{k\to\infty}0.$$

Proof. For arbitrary $k \geq k_{\min}$ under the conditions, it follows from Lemma 4.12, Lemma 4.18 (which shows that $\Delta l(X_k,\mathcal{T}_k,\nabla f(X_k),D_k^{\text{true}}) \geq 0$), (4.17), the fact that $\mathcal{A}_k \geq \mathcal{A}_k^{\min} = \mathcal{A}'\beta_k$, Lemma 4.19, the fact that $\mathcal{A}_k \leq \mathcal{A}_k^{\max} \leq \mathcal{A}_k^{\min} + \theta\beta_k = (\mathcal{A}'+\theta)\beta_k$, Lemma 4.16, Lemma 4.17, and $\beta_k \in (0,1]$ that

$$\begin{aligned}
&\mathbb{E}_k[\phi(X_{k+1},\mathcal{T}_k) - \phi(X_k,\mathcal{T}_k)] = \mathbb{E}_k[\phi(X_k+\mathcal{A}_kD_k,\mathcal{T}_k) - \phi(X_k,\mathcal{T}_k)]\\
&\quad\leq \mathbb{E}_k[-\mathcal{A}_k\Delta l(X_k,\mathcal{T}_k,\nabla f(X_k),D_k^{\text{true}})\\
&\qquad + \mathcal{A}_k\mathcal{T}_k\nabla f(X_k)^T(D_k - D_k^{\text{true}}) + (1-\eta)\mathcal{A}_k\beta_k\Delta l(X_k,\mathcal{T}_k,G_k,D_k)]\\
&\quad\leq -\mathcal{A}'\beta_k\Delta l(X_k,\mathcal{T}',\nabla f(X_k),D_k^{\text{true}}) + (\mathcal{A}'+\theta)\beta_k\mathcal{T}'\kappa_{\nabla f}\zeta^{-1}\sqrt{\rho_k}\\
&\qquad + (1-\eta)(\mathcal{A}'+\theta)\beta_k^2(\Delta l(X_k,\mathcal{T}',\nabla f(X_k),D_k^{\text{true}}) + \mathcal{T}'\zeta^{-1}(\rho_k + \kappa_{\nabla f}\sqrt{\rho_k}))\\
&\quad\leq -\mathcal{A}'\beta_k\Delta l(X_k,\mathcal{T}',\nabla f(X_k),D_k^{\text{true}}) + (\mathcal{A}'+\theta)\beta_k\mathcal{T}'\kappa_{\nabla f}\zeta^{-1}\sqrt{\iota}\beta_k\\
&\qquad + (1-\eta)(\mathcal{A}'+\theta)\beta_k^2(\Delta l(X_k,\mathcal{T}',\nabla f(X_k),D_k^{\text{true}}) + \mathcal{T}'\zeta^{-1}(\iota\beta_k^2 + \kappa_{\nabla f}\sqrt{\iota}\beta_k))\\
&\quad\leq -\beta_k(\mathcal{A}' - (1-\eta)(\mathcal{A}'+\theta)\beta_k)\Delta l(X_k,\mathcal{T}',\nabla f(X_k),D_k^{\text{true}}) + R'\beta_k^2,
\end{aligned}\tag{4.18}$$

where $R' = (\mathcal{A}' + \theta)\mathcal{T}'\zeta^{-1}(\kappa_{\nabla f}\sqrt{\iota} + (1-\eta)(\iota + \kappa_{\nabla f}\sqrt{\iota}))$. Now, from Assumption 4.15 (which subsumes Assumption 2.1), there exists $\phi_{\min} \in \mathbb{R}$ such that $\phi(X_k, \mathcal{T}') \geq \phi_{\min}$ for all $k \geq k_{\min}$. One also finds that $\alpha'_{\min} \leq \mathcal{A}' \leq \alpha'_{\max}$ due to the monotonicity of $\frac{2(1-\eta)\Xi'\tau}{\tau L + \Gamma}$ with respect to τ. Therefore, under part (a), in which case one finds, for $k \geq k_{\min}$, that $\psi\frac{\alpha'_{\min}}{2(1-\eta)(\alpha'_{\min}+\theta)} \leq \beta \leq \psi\frac{\alpha'_{\max}}{2(1-\eta)(\alpha'_{\max}+\theta)}$, it follows from above that

$$\begin{aligned}
&\mathbb{E}_k[\phi(X_{k+1}, \mathcal{T}_k) - \phi(X_k, \mathcal{T}_k)] \\
&\leq -\frac{\psi(1-\frac{\psi}{2})(\alpha'_{\min})^2}{2(1-\eta)(\alpha'_{\min}+\theta)}\Delta l(X_k, \mathcal{T}', \nabla f(X_k), D_k^{\text{true}}) \\
&\quad + \rho'_{\max}\left(\psi\frac{\alpha'_{\max}}{2(1-\eta)(\alpha'_{\max}+\theta)}\right)^2;
\end{aligned}$$

so, by taking total expectation conditioned on the event $\mathcal{E}$, one finds that

$$\begin{aligned}
&\phi_{\min} - \mathbb{E}[\phi(X_{k_{\min}}, \mathcal{T}')|\mathcal{E}] \\
&\quad \leq \mathbb{E}[\phi(X_{k_{\min}+k}, \mathcal{T}') - \phi(X_{k_{\min}}, \mathcal{T}')|\mathcal{E}] = \mathbb{E}\left[\sum_{j=k_{\min}}^{k_{\min}+k-1}(\phi(X_{j+1}, \mathcal{T}') - \phi(X_j, \mathcal{T}'))\middle|\mathcal{E}\right] \\
&\quad \leq -\frac{\psi(1-\frac{\psi}{2})(\alpha'_{\min})^2}{2(1-\eta)(\alpha'_{\min}+\theta)}\mathbb{E}\left[\sum_{j=k_{\min}}^{k_{\min}+k-1}\Delta l(X_j, \mathcal{T}', \nabla f(X_j), D_j^{\text{true}})\middle|\mathcal{E}\right] \\
&\qquad + \frac{k\rho'_{\max}(\psi\alpha'_{\max})^2}{(2(1-\eta)(\alpha'_{\max}+\theta))^2}.
\end{aligned}$$

Observing that $\mathbb{E}[\phi(X_{k_{\min}}, \mathcal{T}')|\mathcal{E}]$ is bounded above under Assumption 4.15 and considering $k \to \infty$, the conclusion of part (a) follows. On the other hand, under the conditions of part (b), it follows in a similar manner that, for any $k \in \mathbb{N}$, one finds

$$\begin{aligned}
&\phi_{\min} - \mathbb{E}[\phi(X_{k_{\min}}, \mathcal{T}')|\mathcal{E}] \\
&\quad \leq \mathbb{E}[\phi(X_{k_{\min}+k}, \mathcal{T}') - \phi(X_{k_{\min}}, \mathcal{T}')|\mathcal{E}] = \mathbb{E}\left[\sum_{j=k_{\min}}^{k_{\min}+k-1}(\phi(X_{j+1}, \mathcal{T}') - \phi(X_j, \mathcal{T}'))\middle|\mathcal{E}\right] \\
&\quad \leq \mathbb{E}\left[\sum_{j=k_{\min}}^{k_{\min}+k-1}(-\beta_j(\mathcal{A}' - (1-\eta)(\mathcal{A}'+\theta)\beta_j)\Delta l(X_j, \mathcal{T}', \nabla f(X_j), D_j^{\text{true}}) + R'\beta_j^2)\middle|\mathcal{E}\right].
\end{aligned}$$

Taking limits as $k \to \infty$, the conclusion of part (b) follows. ◻

Let us now provide further justification for the introduction of the sequence $\{\rho_k\}$ in Assumption 2.2, specifically, the need for this sequence to vanish in part (b) of Lemma 4.20. Algebraically, the need for this sequence to vanish in part (b) can be seen in (4.18), wherein the requirement that $\rho_k \leq \iota\beta_k^2$ for all $k \in \mathbb{N}$ with $k \geq k_{\min}$ is relevant. In particular, this choice ensures that the expected value of the inner product term, namely, $\mathcal{A}_k\mathcal{T}_k\nabla f(X_k)^T(D_k - D_k^{\text{true}})$, can be bounded as $\mathcal{O}(\beta_k^2)$; see Lemma 4.19 as the preliminary result for bounding this term in this manner. A similar bound is needed for such a term in the equality-constrained setting in [3]; see Lemma 3.11 in that paper. However, in the equality-constrained setting, the fact that the search directions are unbiased estimates of their true counterparts allows the expected value of this inner product term to be bounded more tightly in terms of the error in the stochastic gradient estimates. By contrast, in our present setting, the bias in the search directions allows us only to prove Lemma 4.16, where the latter bound involves $\sqrt{\rho_k}$. Consequently, in the present setting, in order to ensure convergence for diminishing $\{\beta_k\}$, the algorithm requires diminishing $\{\rho_k\}$ as well.

To understand the bias in the search directions that arises in the presence of inequality constraints, various examples can be constructed. For example, suppose that $x_k = (1,0)$ and the linearized equality constraints require that $[x_k + d_k]_1 = 0$; i.e., the search direction moves to the vertical axis in $\mathbb{R}^2$. Suppose also that, with only equality constraints, the search direction takes the value $(-1,-1)$ with probability 0.5 and takes the value $(-1,1)$, otherwise. This means that the expected search direction is $(-1,0)$, which means that $x_k + d_k = 0$. However, if the inequality constraints $x_k + d_k \geq 0$ are present, then, with the same stochastic gradient distribution, the expected search direction is $(-1,0.5)$, which is a biased estimate of the true search direction. In Lemma 4.20(b), $\{\rho_k\}$ needs to vanish in order to account for the presence of this bias.

We now present our main convergence theorem for Algorithm 3.1, which is essentially a translation of Lemma 4.20 from results about model reductions to results about quantities connected to measures of stationarity for problem (2.1).

THEOREM 4.21. *Suppose that the conditions of Lemma* 4.20 *hold. Then,*
(a) *under the conditions of Lemma* 4.20(a)*, there exists* $C \in \mathbb{R}_{>0}$ *such that*

$$\limsup_{k\to\infty} \mathbb{E}\left[\frac{1}{k}\sum_{j=k_{\min}}^{k_{\min}+k-1}\left(\frac{1}{2}\mathcal{T}'\zeta\|D_j^{\text{true}}\|_2^2 + \sigma(\|c(X_j)\|_2 - \|c(X_j) + J_j D_j^{\text{true}}\|_2)\right)\middle|\mathcal{E}\right] = C;$$

(b) *under the conditions of Lemma* 4.20(b) *with* $B_k := \sum_{j=k_{\min}}^{k_{\min}+k-1}\beta_j$*,*

$$\mathbb{E}\left[\frac{1}{B_k}\sum_{j=k_{\min}}^{k_{\min}+k-1}\beta_j\left(\frac{1}{2}\mathcal{T}'\zeta\|D_j^{\text{true}}\|_2^2 + \sigma(\|c(X_j)\|_2 - \|c(X_j) + J_j D_j^{\text{true}}\|_2)\right)\middle|\mathcal{E}\right] \xrightarrow{k\to\infty} 0,$$

which further implies that $\liminf_{k\to\infty} \mathbb{E}[\|D_k^{\text{true}}\|_2^2 + (\|c(X_k)\|_2 - \|c(X_k) + J_k D_k^{\text{true}}\|_2)|\mathcal{E}] = 0$*.*

Proof. The desired conclusions follow from Lemmas 4.18 and 4.20. ∎

One might be able to strengthen the conclusion in Theorem 4.21(b), say, to an almost-sure convergence guarantee; see, e.g., [8]. However, we are satisfied with Theorem 4.21(b), which is sufficient for revealing the favorable properties of Algorithm 3.1 under Assumptions 2.2, 2.3, and 4.15. Theorem 4.21(a) shows, under Assumptions 2.2, 2.3, and 4.15, that if the latter condition in (2.6) holds with $\rho_k = \rho$ for some $\rho \in \mathbb{R}_{>0}$ for all $k \in \mathbb{N}$ and $\{\beta_k\} = \{\beta\}$ is chosen as a (sufficiently small) constant sequence, then the limit superior of the expectation of the average of quantities connected to stationarity measures for problem (2.1) is bounded above by a constant proportional to β. Intuitively, this shows that the iterates generated by the algorithm ultimately remain in a region in which these stationarity measures are small. On the other hand, Theorem 4.21(b) shows, under Assumption 4.15, that if $\{\rho_k\}$ and $\{\beta_k\}$ vanish with $\rho_k = \mathcal{O}(\beta_k^2)$, then a subsequence of iterates exists over which the expected values of these stationarity measures vanish. As seen in Lemma 4.3, if there exists a subsequence of iterates, say, indexed by $\mathcal{S} \subseteq \mathbb{N}$, that converges to a point satisfying certain regularity conditions, then $\{\|c_k\|_2 - \|c_k + \nabla c(x_k)^T v_k\|_2\}_{k\in\mathcal{S}} \to 0$ means that the limit point is stationary with respect to the problem to minimize $\frac{1}{2}\|c(x)\|_2^2$ subject to $x \in \mathbb{R}^n_{\geq 0}$. Similarly, as seen in Lemma 4.5, if there exists such a subsequence and the limit point is feasible with respect to problem (2.1), then $\{d_k^{\text{true}}\}_{k\in\mathcal{S}} \to 0$ means that the limit point is stationary with respect to (2.1). These situations are not guaranteed to occur, but this discussion shows that Theorem 4.21 is meaningful.

4.4. Nonvanishing merit parameter. Our main convergence result in the previous section, namely, Theorem 4.21, requires Assumption 4.15, which, in turn, requires that the merit parameter sequence ultimately becomes a sufficiently small, positive constant sequence. (Recall the discussion after Assumption 4.15.) To show that this corresponds to a realistic event for practical purposes, we next show conditions under which one finds that the merit parameter would not vanish.

We begin by showing a generally applicable result about the solution of (3.1). It is related to that in Lemma 4.3 but is stronger due to an additional assumption.

LEMMA 4.22. *Suppose that the conditions of Lemma* 4.3 *hold and there exists* $\kappa_w \in [0,1)$ *such that, for all generated* $k \in \mathbb{N}$ *in any run of the algorithm, one has* $\|c_k + \nabla c(x_k)^T v_k\|_2 \leq \kappa_w \|c_k\|_2$. *Then, there exists* $\kappa_v \in \mathbb{R}_{>0}$ *such that, in any run of the algorithm such that iteration* $k \in \mathbb{N}$ *is reached, one finds that*

$$\|c_k\|_2 - \|c_k + \nabla c(x_k)^T v_k\|_2 \geq \kappa_v \|v_k\|_2. \tag{4.19}$$

Proof. Consider an arbitrary run of the algorithm in which the conditions of the lemma hold and iteration $k \in \mathbb{N}$ is reached. If $c_k = 0$, then it follows by construction of (3.1) that $v_k = 0$, in which case (4.19) follows trivially. Hence, we may proceed under the assumption that $c_k \neq 0$, which, by the conditions of Lemma 4.22, Assumption 2.1 (see (2.2)), and the triangle inequality gives $\kappa_{\nabla c}\|v_k\|_2 \geq \|\nabla c(x_k)^T v_k\|_2 \geq \|c_k\|_2 - \|c_k + \nabla c(x_k)^T v_k\|_2 \geq (1-\kappa_w)\|c_k\|_2$. Consequently, from (4.6), (4.7), and a similar derivation as in Lemma 4.3, one finds that $2\|c_k\|_2(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T v_k\|_2) \geq \|c_k\|_2^2 - \|c_k + \nabla c(x_k)^T v_k\|_2^2 \geq \min\{\frac{\lambda^2}{\kappa_{\nabla c}^2}, 2\mu\}\|v_k\|_2^2 \geq \min\{\frac{\lambda^2}{\kappa_{\nabla c}^2}, 2\mu\}(\frac{1-\kappa_w}{\kappa_{\nabla c}})\|c_k\|_2\|v_k\|_2$, from which the desired conclusion in (4.19) follows. ∎

We now show that, under common conditions and when the norm of the stochastic gradient estimate is bounded uniformly, the denominator of the formula for τ_k^{trial} in (3.4) is bounded proportionally to $\|v_k\|_2$.

LEMMA 4.23. *Suppose that Assumptions* 2.1 *and* 2.3 *hold and that there exist* $(\lambda, \mu, \kappa_g) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0}$ *such that, for all generated* $k \in \mathbb{N}$ *in any run, one has* $\nabla c(x_k)^T \nabla c(x_k) \succeq \lambda I$, $\mu_k \geq \mu$, *and* $\|g_k\|_2 \leq \kappa_g$. *Then, there exists* $\kappa_{g,H} \in \mathbb{R}_{>0}$ *such that, in any run such that iteration* $k \in \mathbb{N}$ *is reached,* $g_k^T d_k + \frac{1}{2} d_k^T H_k d_k \leq \kappa_{g,H}\|v_k\|_2$.

Proof. Consider an arbitrary run in which the conditions of Lemma 4.23 hold and iteration $k \in \mathbb{N}$ is reached. By Lemma 4.14, $(u, w) = (0, 0)$ is feasible for (3.1), so

$$\begin{aligned}\max&\left\{\frac{1}{2}\|c_k + \nabla c(x_k)^T \nabla c(x_k) w_k\|_2^2, \frac{1}{2}\mu_k\|u_k\|_2^2\right\}\\ &\leq \frac{1}{2}\|c_k + \nabla c(x_k)^T \nabla c(x_k) w_k\|_2^2 + \frac{1}{2}\mu_k\|u_k\|_2^2 \leq \frac{1}{2}\|c_k\|_2^2.\end{aligned}$$

Since $\frac{1}{2}\|c_k + \nabla c(x_k)^T \nabla c(x_k) w_k\|_2^2 \leq \frac{1}{2}\|c_k\|_2^2$, it follows that $\|\nabla c(x_k)^T \nabla c(x_k) w_k\|_2^2 \leq -2c_k^T \nabla c(x_k)^T \nabla c(x_k) w_k \leq 2\|c_k\|_2\|\nabla c(x_k)^T \nabla c(x_k) w_k\|_2$, which, along with Assumption 2.1 (see (2.2)), shows that $\|\nabla c(x_k) w_k\|_2 \leq \kappa_{\nabla c}\|w_k\|_2 \leq \frac{\kappa_{\nabla c}}{\lambda}\|\nabla c(x_k)^T \nabla c(x_k) w_k\|_2$

$\leq 2\frac{\kappa_{\nabla c}}{\lambda}\|c_k\|_2 \leq 2\frac{\kappa_{\nabla c}}{\lambda}\kappa_c$. On the other hand, since $\frac{1}{2}\mu_k\|u_k\|_2^2 \leq \frac{1}{2}\|c_k\|_2^2$, it follows, under Assumption 2.1, that $\|u_k\|_2 \leq \frac{1}{\sqrt{\mu_k}}\|c_k\|_2 \leq \frac{1}{\sqrt{\mu}}\kappa_c$. Therefore, we have that $\|v_k\|_2 = \sqrt{\|\nabla c(x_k)w_k\|_2^2 + \|u_k\|_2^2} \leq (\sqrt{4(\frac{\kappa_{\nabla c}}{\lambda})^2 + \frac{1}{\mu}})\kappa_c$. Now, since $v_k = \nabla c(x_k)w_k + u_k$ is a feasible solution of (3.2) while d_k is the optimal solution of (3.2), under the conditions of Lemma 4.23,

$$\begin{aligned} g_k^T d_k + \frac{1}{2}d_k^T H_k d_k &\leq g_k^T v_k + \frac{1}{2}v_k^T H_k v_k \\ &\leq \kappa_g\|v_k\|_2 + \frac{1}{2}\kappa_H\|v_k\|_2^2 \leq \left(\kappa_g + \frac{1}{2}\kappa_H\left(\sqrt{4(\frac{\kappa_{\nabla c}}{\lambda})^2 + \frac{1}{\mu}}\right)\kappa_c\right)\|v_k\|_2, \end{aligned}$$

which leads to the desired conclusion. □

We now prove conditions under which the merit parameter does not vanish.

THEOREM 4.24. *Suppose that Assumptions* 2.1 *and* 2.3 *hold and that there exist* $(\lambda, \mu, \kappa_g, \kappa_w) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0} \times [0,1)$ *such that, for all generated* $k \in \mathbb{N}$ *in any run of the algorithm, one has that* $\nabla c(x_k)^T\nabla c(x_k) \succeq \lambda I$, $\mu_k \geq \mu$, $\|g_k\|_2 \leq \kappa_g$, *and* $\|c_k + \nabla c(x_k)^T v_k\|_2 \leq \kappa_w\|c_k\|_2$. *Then, in any run that does not terminate finitely, the latter event in Lemma* 4.14(*k*) *occurs* (*i.e.*, $\{\tau_k\}$ *does not vanish*) *with* $\tau_{\min} \geq \frac{(1-\sigma)\kappa_v}{\kappa_{g,H}}(1-\epsilon_\tau)$.

Proof. Consider arbitrary $k \in \mathbb{N}$ in a run that does not terminate finitely, and note that, if $d_k = 0$ or $g_k^T d_k + \frac{1}{2}d_k^T H_k d_k \leq 0$, then $\tau_k^{\text{trial}} \leftarrow \infty$, and otherwise, τ_k^{trial} is set by (3.4). Hence, under the conditions of Theorem 4.24 and by Lemmas 4.22 and 4.23,

$$\begin{aligned} \tau_k^{\text{trial}} &\geq \frac{(1-\sigma)(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T d_k\|_2)}{g_k^T d_k + \frac{1}{2}d_k^T H_k d_k} = \frac{(1-\sigma)(\|c_k\|_2 - \|c_k + \nabla c(x_k)^T v_k\|_2)}{g_k^T d_k + \frac{1}{2}d_k^T H_k d_k} \\ &\geq \frac{(1-\sigma)\kappa_v}{\kappa_{g,H}} =: \tau_*. \end{aligned}$$

Consequently, by the merit parameter update in (3.5), $\tau_k < \tau_{k-1}$ only if $\tau_{k-1} > \tau_*$. This, along with Lemma 4.14(d)–(e), leads to the conclusion. □

Since ∇f is bounded in norm over the set $\mathcal{X}$ in Assumption 2.1, Theorem 4.24 shows that, among the other stated conditions, if $\|g_k - \nabla f(x_k)\|_2$ is bounded uniformly over all $k \in \mathbb{N}$ in any, then the merit parameter sequence always remains bounded below by a positive number. Under such conditions, the only potentially poor behavior of the merit parameter sequence is that, in a given run, it ultimately remains constant at a value that is too large. We claim that, under certain assumptions about the distribution of the stochastic gradient estimates, this behavior can be shown to occur with probability zero. (We do not prove such a result here, but refer the interested reader to Proposition 3.16 in [3] to see such a result for the equality-constraints-only setting, in which case the behavior of the merit parameter is similar.) On the other hand, if $\|g_k - \nabla f(x_k)\|_2$ is not bounded uniformly in this manner, then it is possible for the merit parameter sequence to vanish unnecessarily. This issue is one that should be noted by a user of the algorithm. In particular, if, in a run of the algorithm, one chooses $\mu_k \geq \mu$ for some $\mu \in \mathbb{R}_{>0}$ for all $k \in \mathbb{N}$ and finds, for some $(\lambda, \kappa_w) \in \mathbb{R}_{>0} \times (0,1)$, that generated $k \in \mathbb{N}$ yield $\nabla c(x_k)^T\nabla c(x_k) \succeq \lambda I$ and $\|c_k + \nabla c(x_k)^T v_k\|_2 \leq \kappa_w\|c_k\|_2$, yet τ_k has become exceedingly small, then Theorem 4.24 shows that this must be due to the stochastic gradient estimates tending to become significantly large in norm, in which case the performance of the algorithm may improve with more accurate stochastic gradient estimates.

4.5. Deterministic algorithm. We conclude this section with a statement of a convergence result that we claim to hold for Algorithm 3.1 if it were to be run with $g_k = \nabla f(x_k)$ for all $k \in \mathbb{N}$. Due to space considerations, we do not provide a proof of the result, although we offer the proposition for reference for the reader and claim that it holds from results proved in this paper for the stochastic setting as well as other similar results for SQP methods for deterministic continuous nonlinear optimization.

PROPOSITION 4.25. *Suppose that Assumptions* 2.1 *and* 2.3 *hold and Algorithm* 3.1 *is run with $g_k = \nabla f(x_k)$ for all $k \in \mathbb{N}$. If, for all large $k \in \mathbb{N}$, there exists $\kappa_w \in [0,1)$ such that $\|c_k + \nabla c(x_k)^T v_k\|_2 \le \kappa_w \|c_k\|_2$, then $\{x_k\} \subset \mathbb{R}^n_{\ge 0}$, $\{\tau_k\}$ is bounded away from zero, and, with $y_k \in \mathbb{R}^m$ and $z_k \in \mathbb{R}^n_{\ge 0}$ defined as the optimal multipliers corresponding to the solution of subproblem* (3.2) *for all $k \in \mathbb{N}$, it follows that*

$$\left\{\left\|\begin{bmatrix}(\nabla f(x_k) + \nabla c(x_k) y_k - z_k)^T & c_k^T & x_k^T z_k\end{bmatrix}\right\|\right\} \to 0.$$

Otherwise, $\{x_k\} \subset \mathbb{R}^n_{\ge 0}$, $\{\min\{\nabla c(x_k) c_k, 0\}\} \to 0$, and $\{|x_k^T \nabla c(x_k) c_k|\} \to 0$, and, if $\{\tau_k\}$ is bounded away from zero, then $\left\{\left\|\begin{bmatrix}(\nabla f(x_k) + \nabla c(x_k) y_k - z_k)^T & x_k^T z_k\end{bmatrix}\right\|\right\} \to 0$.

5. Numerical results. In this section, we provide results demonstrating the performance of a MATLAB implementation of Algorithm 3.1 when solving a subset of problems from CUTEst [23] and a couple of fair machine learning test problems, where Gurobi is used to solve the arising subproblems [24]. The purposes of our experiments with the CUTEst problems are twofold. First, on a subset of problems, we compare the performance of our method against that of the Julia implementation provided by the authors of [34, Algorithm 1]. Second, on a larger subset, we demonstrate that, for our method, one should aim to trade off the cost of more accurate gradient estimates and the cost of solving the arising subproblems. These experiments also allow us to demonstrate that there are settings in which our approach with relatively less accurate gradient estimates can be more computationally efficient than one that relies on highly accurate gradient estimates (i.e., an approach that is nearly deterministic). From all inequality-constrained problems in CUTEst, we selected those such that (i) $m \le n \le 1000$, (ii) $f(x_k) \ge -10^{20}$ for all $k \in \mathbb{N}$ in all runs of our algorithm, and (iii) Gurobi did not report any errors. This resulted in a set of 323 test problems. The purpose of our experiments with fair machine learning test problems is to demonstrate the performance of our algorithm on problems derived from a real-world application and to show that, on such problems, it can outperform approaches that handle the constraints by moving them to the objective through penalty terms.

We begin by presenting our results pertaining to CUTEst. For each test problem, both our code and that for the Julia implementation of [34, Algorithm 1] used the same initial iterate and generated stochastic gradient estimates in the same manner. Specifically, for all $k \in \mathbb{N}$ in each run, the codes set $g_k = \mathcal{N}(\nabla f(x_k), \epsilon_g(I + ee^T))$, where e is the all-ones vector and $\epsilon_g \in \{10^{-8}, 10^{-4}, 10^{-2}, 10^{-1}\}$ was fixed for each run (see below). If a problem had only inequality constraints (i.e., $m = 0$), then our code explicitly computed α_k^φ (as defined in (3.9)) and set $\alpha_k \leftarrow \alpha_k^{\max}$ for all $k \in \mathbb{N}$. Otherwise, the code set $\alpha_k \leftarrow \min\{1, (1.1)^{t_k} \alpha_k^{\min}, \alpha_k^{\min} + \theta \beta_k\}$, where $t_k \leftarrow \max\{t \in \mathbb{N} : \varphi_k((1.1)^t \alpha_k^{\min}) \le 0\}$. This guaranteed that $\alpha_k \in [\alpha_k^{\min}, \alpha_k^{\max}]$ for all $k \in \mathbb{N}$. The other user-defined parameters of Algorithm 3.1 were selected as $\sigma = \tau_0 = 0.1$, $\eta = 0.5$, $\xi_0 = 1$, $\epsilon_\tau = \epsilon_\xi = 10^{-2}$, $\theta = 10^4$, $\mu_k = \max\{10^{-8}, 10^{-4}\|c_k\|_2^2\}$, $\beta_k = 1$, and $H_k = I$ for all $k \in \mathbb{N}$.

The Lipschitz constants L and Γ were estimated every 100 iterations by differences of stochastic gradients at 10 samples around the current iterate. Meanwhile, we ran the Julia code for [34, Algorithm 1] with the `AdapGD` option and its default parameter settings as described in [34, section 4].

For our comparison of our code with the Julia implementation of [34, Algorithm 1], each code terminated as soon as 10^4 stochastic gradient samples were evaluated or a 12-hour CPU time limit was reached. Let $\texttt{FeasErr}(x)$ be the ∞-norm constraint violation at x, and let $\texttt{KKTErr}(x,y,z)$ be the ∞-norm violation of the KKT conditions (recall (2.4)) at a primal-dual iterate (x,y,z). Each run of our MATLAB implementation of Algorithm 3.1 generates $\{x_k\} \subset \mathbb{R}^n$. For each $k \in \mathbb{N}$, let $y_k^{\text{true}} \in \mathbb{R}^m$ and $z_k^{\text{true}} \in \mathbb{R}^n$ denote the optimal Lagrange multipliers corresponding to the equality and inequality constraints when (3.2) is solved with $g_k = \nabla f(x_k)$. For each run of Algorithm 3.1, we determined the best iterate as $x_{k_{\texttt{best}}}$, where

$$k_{\texttt{best}} = \begin{cases} \arg\min\limits_{k\in\mathbb{N}} \ \texttt{FeasErr}(x_k) & \text{if } \texttt{FeasErr}(x_k) > 10^{-4} \text{ for all } k \in \mathbb{N}, \\ \arg\min\limits_{k\in\mathbb{N}} \ \{\texttt{KKTErr}(x_k, y_k^{\text{true}}, z_k^{\text{true}}) : \texttt{FeasErr}(x_k) \le 10^{-4}\}, & \text{otherwise.} \end{cases}$$

We determined the best iterate in a run of [34, Algorithm 1] using the same formula with the sequence of iterates and Lagrange multiplier estimates that are computed as part of the algorithm. Our results for four noise levels, provided in Figure 5.1 below, are presented in terms of $\texttt{FeasErr}(x_{k_{\texttt{best}}})$ as the feasibility error and $\texttt{KKTErr}(x_{k_{\texttt{best}}}, y_{k_{\texttt{best}}}^{\text{true}}, z_{k_{\texttt{best}}}^{\text{true}})$ as the KKT error for each run of both algorithms.

Since the Julia code for [34, Algorithm 1] is only set up to solve CUTEst problems without simple bound constraints, the plots in Figure 5.1 are only based on problems that code was able to handle. In particular, there are 57 problems for which both algorithms were set up to run, and in Figure 5.1, the two box plots show the best feasibility and KKT errors achieved by both codes, where each problem is run 5 times each (since the behaviors of the algorithms are stochastic). These results show that our approach performs very well in comparison to the method from [34].

Let us now provide some additional results for the implementation of our algorithm employed to solve all 323 problems in our subset from CUTEst. Our aim in this experiment is to demonstrate potential trade-offs between accuracy in the gradient estimates and the cost of solving the subproblems in our algorithm. We set up the experiment as follows. First, we defined a *unit* as the cost of a stochastic gradient estimate with noise level $\epsilon_g = 10^{-1}$. Second, to represent the extra cost of more accurate estimates, we suppose that the cost of a stochastic gradient estimate with

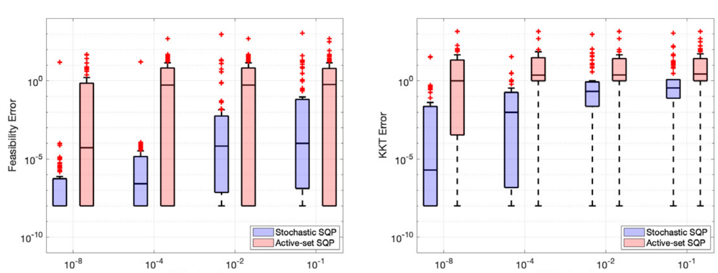

FIG. 5.1. *Box plots comparing the best feasibility errors (left) and KKT errors (right) of a MATLAB implementation of Algorithm* 3.1 *("Stochastic SQP") and the Julia implementation provided by the authors of* [34, Algorithm 1] *("Active-set SQP") when solving* 57 *CUTEst problems.*

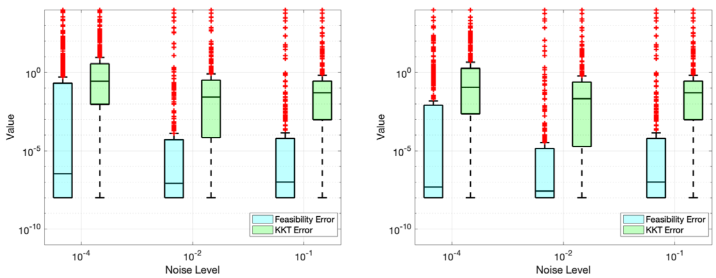

FIG. 5.2. *Histogram of final* $\log_{10}(\tau)$.

$\epsilon_g = 10^{-2}$ is 10 units and that with $\epsilon_g = 10^{-4}$ is 1000 units. (These relative costs are not exact in general; we simply made these choices for the purposes of demonstrating one realistic setting.) Third, we considered two relative costs for the subproblem solves in an iteration: two units ("2x") or five units ("5x"). These represent two reasonable possibilities in practice. Recall that, generally speaking, the cost of solving the subproblems depends on the numbers of variables and constraints, whereas the cost of more accurate gradient estimates depends on the variance of the estimates, which, for example, when f is defined by an average of functions, could depend on the number of terms in the average—which could be extremely large—and the cost of evaluating the gradient for each term in the average. We consider here situations when the cost of obtaining accurate gradient estimates is not trivial so that the relative cost of a subproblem solve is a few times that of a stochastic gradient estimate with $\epsilon_g = 10^{-1}$.

For each setting, namely, 2x and 5x, we ran our algorithm with noise levels $\epsilon_g \in \{10^{-4}, 10^{-2}, 10^{-1}\}$ with the same computational budget in terms of units. For example, for the 2x case, the cost per iteration with $\epsilon_g = 10^{-1}$ is $1+2=3$ units, whereas the cost per iteration with $\epsilon_g = 10^{-2}$ is $10+2=12$ units. This means that the latter run (with more accurate gradient estimates) is only able to run 1/4th the number of iterations as the former run (with less accurate gradient estimates). We always ran 2×10^5 iterations for the $\epsilon_g = 10^{-1}$ setting, and we determined computational unit budgets for the 2x and 5x settings based on this benchmark iteration budget for the $\epsilon_g = 10^{-1}$ setting. The results are presented in Figure 5.2, where again, we present box plots for feasibility errors and KKT errors for the best iterates found using the same criteria as our comparisons earlier in this section. Overall, the results show that there exist settings—namely, the ones that we consider here—where one does not obtain the best results by employing highly accurate gradient estimates. Instead, the trade-off between gradient accuracy and subproblem cost can be such that, for a limited computational budget, one obtains the best results by allowing some inaccuracy in the stochastic gradient estimates. We remark that, in these experiments, our algorithm regularly did not reduce the merit parameter to small values. Rather, low feasibility errors were generally attained with moderate τ values; see Figure 5.3 for a histogram of final $\log_{10}(\tau)$ values with respect to the noisiest setting, namely, $\epsilon_g = 10^{-1}$.

We close this section with the results of experiments on a couple of fair machine learning test problems. In particular, we consider logistic regression problems where the constraints bound a surrogate for disparate impact; specifically, we consider problem (3) from [12] with $\epsilon = 0.1$. We use the `Adult` and `German` datasets that are available from [28], where gender is the sensitive feature. For each dataset, we randomly

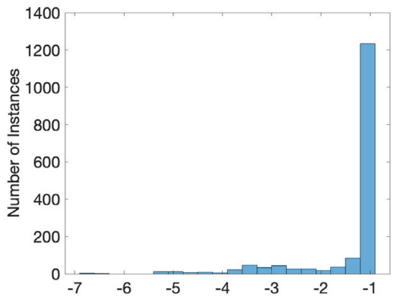

FIG. 5.3. *Box plots comparing the best feasibility errors and KKT errors of a MATLAB implementation of Algorithm* 3.1 *("Stochastic SQP") with different noise levels all subject to the same computational budget. For the results on the left, the cost of solving the subproblems in an iteration is presumed to be twice that of a relatively inaccurate stochastic gradient estimate, and on the right, the cost of the subproblem solves is presumed to be five times the cost of such a gradient estimate.*

TABLE 5.1
Performance of "Stochastic SQP" versus the algorithm of Wang and Spall [46] *and a stochastic subgradient method applied to minimize a penalty function for the fairness-constrained logistic regression problem* (3) *from* [12]*, here using the* `Adult` *data with gender as the sensitive attribute.*

Algorithm	Training infeasibility error	Training accuracy	Testing infeasibility error	Testing accuracy
Stochastic SQP	1.9e-08	83.3%	5.1e-02	82.9%
Wang and Spall [46]	5.8e-02	63.9%	1.3e-01	63.2%
Subgradient (10^{-1})	8.8e-05	63.7%	0.0e+00	63.1%
Subgradient (10^{-4})	3.5e-05	72.4%	0.0e+00	72.3%
Subgradient (10^{-7})	6.5e-05	72.4%	0.0e+00	72.3%

selected data points for a training set (to define the optimization problem that we solve) and a testing set: For `Adult`, of the 45222 data points, 35000 were selected for training with the remaining used for testing, and for `German`, of the 1000 data points, 800 were selected for training, and the remaining were used for testing. For `Adult`, 1000 of the training data points were randomly selected to define the constraints, and a minibatch size of 1000 was used for stochastic gradient estimates. For `German`, 100 of the training data points were randomly selected to define the constraints, and a minibatch size of 100 was used for stochastic gradient estimates. Since our stochastic SQP method has already been shown to compare favorably against [34, Algorithm 1] in our experiments with CUTEst problems (see Figure 5.1), in this set of experiments, we compare our proposed method with two other algorithms: (i) the algorithm proposed by Wang and Spall in [46] with the settings from section 4 of that paper (see also [40]) and (ii) a stochastic subgradient method employed to minimize an ℓ_2-norm exact penalty function when the logistic loss objective is weighted by 10^{-1}, 10^{-4}, or 10^{-7}. For a fixed budget of 10000 iterations for stochastic SQP and 100000 iterations for each of the other algorithms, the results for the best iterates in terms of training infeasibility, training classification accuracy, testing infeasibility, and testing accuracy are provided in Tables 5.1 and 5.2. The results show that, despite the additional iterations offered to the other methods, our stochastic SQP method generally outperforms them in terms of these metrics. In addition, we provide in Figures 5.4 and 5.5 plots of these metrics as a function of CPU time for a single representative run of a few of the algorithms with each dataset. (We do not plot the performance of "subgradient

Table 5.2

Performance of "Stochastic SQP" versus the algorithm of Wang and Spall [46] *and a stochastic subgradient method applied to minimize a penalty function for the fairness-constrained logistic regression problem* (3) *from* [12], *here using the* **German** *data with gender as the sensitive attribute.*

Algorithm	Training infeasibility error	Training accuracy	Testing infeasibility error	Testing accuracy
Stochastic SQP	3.2e-08	73.8%	0.0e+00	75.0%
Wang and Spall	4.5e-01	60.6%	3.1e-01	70.0%
Subgradient (10^{-1})	5.5e-01	56.6%	5.5e-01	58.0%
Subgradient (10^{-4})	5.7e-01	49.8%	6.4e-01	48.0%
Subgradient (10^{-7})	5.7e-01	49.8%	6.4e-01	48.0%

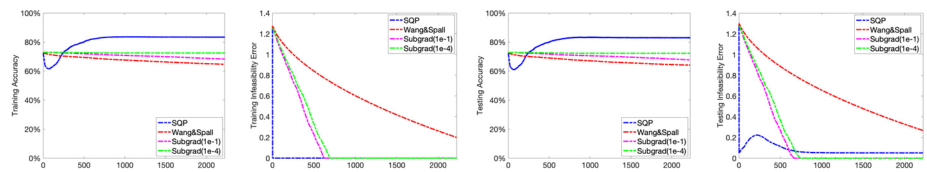

Fig. 5.4. *CPU time versus training accuracy, training infeasibility error, testing accuracy, and testing infeasibility error for a representative run of SQP, Wang and Spall* [46], *subgradient* (10^{-1}), *and subgradient* (10^{-4}) *with the* Adult *dataset.*

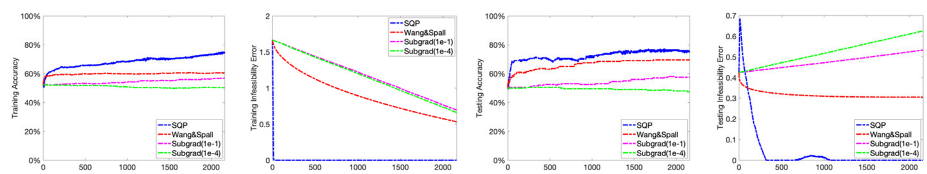

Fig. 5.5. *CPU time versus training accuracy, training infeasibility error, testing accuracy, and testing infeasibility error for a representative run of SQP, Wang and Spall* [46], *subgradient* (10^{-1}), *and subgradient* (10^{-4}) *with the* German *dataset.*

(10^{-7})" since its performance was similar to that of "subgradient (10^{-4})" in terms of these plots.) These results further show that our SQP method has desirable practical performance.

6. Conclusion.

We have proposed, analyzed, and tested an algorithm for solving continuous optimization problems. The algorithm requires that constraint function and derivative values can be computed in each iteration but does not require exact objective function and derivative values; rather, the algorithm merely requires that a stochastic objective gradient estimate is computed to satisfy relatively loose assumptions in each iteration. The theoretical convergence guarantees of the algorithm require knowledge of Lipschitz constants for the objective gradient and constraint Jacobian, although, in practice, these constants can be estimated. Our numerical experiments show that our proposed algorithm can outperform an alternative algorithm that relies on the ability to compute more accurate gradient estimates. We have provided comments throughout the paper on how the assumptions that are required for our theoretical convergence guarantees might be loosened further.

Acknowledgments. The authors are grateful to Sen Na for providing consultation about the Julia implementation provided by the authors of [34, Algorithm 1]. The authors also thank the editors and reviewers for their helpful comments.

REFERENCES

[1] A. S. Berahas, R. Bollapragada, and B. Zhou, *An Adaptive Sampling Sequential Quadratic Programming Method for Equality Constrained Stochastic Optimization*, preprint, arXiv:2206.00712, 2022.

[2] A. S. Berahas, F. E. Curtis, M. J. O'Neill, and D. P. Robinson, *A stochastic sequential quadratic optimization algorithm for nonlinear equality constrained optimization with rank-deficient Jacobians*, Math. Oper. Res., (2023), online ahead of print, https://doi.org/10.1287/moor.2021.0154.

[3] A. S. Berahas, F. E. Curtis, D. Robinson, and B. Zhou, *Sequential quadratic optimization for nonlinear equality constrained stochastic optimization*, SIAM J. Optim., 31 (2021), pp. 1352–1379.

[4] A. S. Berahas, J. Shi, Z. Yi, and B. Zhou, *Accelerating stochastic sequential quadratic programming for equality constrained optimization using predictive variance reduction*, Comput. Optim. Appl., 86 (2023), pp. 79–116, https://doi.org/10.1007/s10589-023-00483-2.

[5] A. S. Berahas, M. Xie, and B. Zhou, *A Sequential Quadratic Programming Method with High Probability Complexity Bounds for Nonlinear Equality Constrained Stochastic Optimization*, preprint, arXiv:2301.00477, 2023.

[6] D. Bertsekas, *Convex Optimization Theory*, Athena Scientific, Nashua, NH, 2009.

[7] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*, Athena Scientific, Nashua, NH, 1998.

[8] D. P. Bertsekas and J. N. Tsitsiklis, *Gradient convergence in gradient methods with errors*, SIAM J. Optim., 10 (2000), pp. 627–642.

[9] R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Math. Program., 89 (2000), pp. 149–185.

[10] R. H. Byrd, M. E. Hribar, and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming*, SIAM J. Optim., 9 (1999), pp. 877–900.

[11] A. R. Conn, *Constrained optimization using a nondifferentiable penalty function*, SIAM J. Numer. Anal., 10 (1973), pp. 760–784.

[12] F. E. Curtis, S. Liu, and D. P. Robinson, *Fair machine learning through constrained stochastic optimization and an ϵ-constraint method*, Opt. Lett., (2023), online ahead of print, https://doi.org/10.1007/s11590-023-02024-6.

[13] F. E. Curtis, M. J. O'Neill, and D. P. Robinson, *Worst-case complexity of an SQP method for nonlinear equality constrained stochastic optimization*, Math. Program., 205 (2024), pp. 431–483, https://doi.org/10.1007/s10107-023-01981-1.

[14] F. E. Curtis, D. P. Robinson, and B. Zhou, *A stochastic inexact sequential quadratic optimization algorithm for nonlinear equality-constrained optimization*, INFORMS J. Optim., (2024), online ahead of print, https://doi.org/10.1287/ijoo.2022.0008.

[15] G. Di Pillo and L. Grippo, *A continuously differentiable exact penalty function for nonlinear programming problems with inequality constraints*, SIAM J. Control Optim., 23 (1985), pp. 72–84.

[16] G. Di Pillo and L. Grippo, *Exact penalty functions in constrained optimization*, SIAM J. Control Optim., 27 (1989), pp. 1333–1360.

[17] I. Dikin, *Iterative solution of problems of linear and quadratic programming*, in Dokl. Akad. Nauk SSSR, 174 (1967), pp. 747–748.

[18] Y. Fang, S. Na, M. W. Mahoney, and M. Kolar, *Fully Stochastic Trust-Region Sequential Quadratic Programming for Equality-Constrained Optimization Problems*, preprint, arXiv:2211.15943, 2022.

[19] R. Fletcher, *An exact penalty function for nonlinear programming with inequalities*, Math. Program., 5 (1973), pp. 129–150.

[20] R. Fletcher, *Practical Methods of Optimization*, Wiley, Hoboken, NJ, 2013.

[21] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Rev., 47 (2005), pp. 99–131.

[22] T. Glad and E. Polak, *A multiplier method with automatic limitation of penalty growth*, Math. Program., 17 (1979), pp. 140–155.

[23] N. I. Gould, D. Orban, and P. L. Toint, *CUTEst: A constrained and unconstrained testing environment with safe threads for mathematical optimization*, Comput. Optim. Appl., 60 (2015), pp. 545–557.

[24] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/10.0/refman.pdf (February 2023).

[25] S.-P. Han, *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Program., 11 (1976), pp. 263–282.
[26] R. J. Hathaway, *A constrained formulation of maximum-likelihood estimation for normal mixture distributions*, Ann. Statist., 13 (1985), pp. 795–800.
[27] T. Ibaraki and N. Katoh, *Resource Allocation Problems: Algorithmic Approaches*, MIT Press, Cambridge, MA, 1988.
[28] M. Kelly, R. Longjohn, and K. Nottingham, *The UCI Machine Learning Repository*, https://archive.ics.uci.edu (November 2023).
[29] D. P. Kouri and T. M. Surowiec, *Risk-averse PDE-constrained optimization using the conditional value-at-risk*, SIAM J. Optim., 26 (2016), pp. 365–396.
[30] G. Lan, *First-Order and Stochastic Optimization Methods for Machine Learning*, Springer Ser. Data Sci., Springer, Cham, 2020.
[31] L. Lasdon, A. Waren, and R. Rice, *An interior penalty method for inequality constrained optimal control problems*, IEEE Trans. Automat. Control, 12 (1967), pp. 388–395.
[32] R. McGill, *Optimum control, inequality state constraints, and the generalized Newton–Raphson algorithm*, J. SIAM Control, 3 (1965), pp. 291–298.
[33] S. Na, M. Anitescu, and M. Kolar, *An adaptive stochastic sequential quadratic programming with differentiable exact augmented Lagrangians*, Math. Program., 199 (2023), pp. 721–791.
[34] S. Na, M. Anitescu, and M. Kolar, *Inequality constrained stochastic nonlinear optimization via active-set sequential quadratic programming*, Math. Program., 202 (2023), pp. 279–353.
[35] S. Na and M. W. Mahoney, *Asymptotic Convergence Rate and Statistical Inference for Stochastic Sequential Quadratic Programming*, preprint, arXiv:2205.13687, 2022.
[36] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, Cham, 2006.
[37] F. Oztoprak, R. Byrd, and J. Nocedal, *Constrained optimization in the presence of noise*, SIAM J. Optim., 33 (2023), pp. 2118–2136.
[38] V. Patel and S. Zhang, *Stochastic Gradient Descent on Nonconvex Functions with General Noise Models*, preprint, arXiv:2104.00423, 2021.
[39] A. F. Perold, *Large-scale portfolio optimization*, Manag. Sci., 30 (1984), pp. 1143–1160.
[40] G. C. Pflug, *On the convergence of a penalty-type stochastic optimization procedure*, J. Inform. Optim. Sci., 2 (1981), pp. 249–258.
[41] S. Qiu and V. Kungurtsev, *A Sequential Quadratic Programming Method for Optimization with Stochastic Objective Functions, Deterministic Inequality Constraints and Robust Subproblems*, preprint, arXiv:2302.07947, 2023.
[42] D. P. Robinson, *Primal-Dual Methods for Nonlinear Optimization*, Ph.D. thesis, University of California, San Diego, CA, 2007.
[43] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*, MOS-SIAM Ser. Optim., SIAM, Philadelphia, 2021.
[44] Q. Shi, X. Wang, and H. Wang, *A Momentum-based Linearized Augmented Lagrangian Method for Nonconvex Constrained Stochastic Optimization*, https://optimization-online.org/?p=19870 (October 2022).
[45] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program., 106 (2006), pp. 25–57.
[46] I.-J. Wang and J. C. Spall, *A constrained simultaneous perturbation stochastic approximation algorithm based on penalty functions*, in Proceedings of the 1999 Amer. Cont. Conference, Vol. 1, IEEE, 1999, pp. 393–399.
[47] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.
[48] H. Yamashita, *A globally convergent primal-dual interior point method for constrained optimization*, Optim. Methods Softw., 10 (1998), pp. 443–469.
[49] W. I. Zangwill, *Nonlinear programming via penalty functions*, Manag. Sci., 13 (1967), pp. 344–358.
[50] V. M. Zavala and M. Anitescu, *Scalable nonlinear programming via exact differentiable penalty functions and trust-region newton methods*, SIAM J. Optim., 24 (2014), pp. 528–558.