A DYNAMIC PENALTY PARAMETER UPDATING STRATEGY FOR MATRIX-FREE SEQUENTIAL QUADRATIC OPTIMIZATION

JAMES V. BURKE*, FRANK E. CURTIS†, HAO WANG‡, AND JIASHAN WANG§

Abstract. This paper focuses on the design of sequential quadratic optimization (commonly known as SQP) methods for solving large-scale nonlinear optimization problems. The most computationally demanding aspect of such an approach is the computation of the search direction during each iteration, for which we consider the use of matrix-free methods. In particular, we develop a method that requires an inexact solve of a single QP subproblem to establish the convergence of the overall SQP method. It is known that SQP methods can be plagued by poor behavior of the global convergence mechanism. To confront this issue, we propose the use of an exact penalty function with a dynamic penalty parameter updating strategy to be employed *within* the subproblem solver in such a way that the resulting search direction predicts progress toward both feasibility and optimality. We present our parameter updating strategy and prove that, under reasonable assumptions, the strategy does not modify the penalty parameter unnecessarily. We also discuss a matrix-free subproblem solver in which our updating strategy can be incorporated. We close the paper with a discussion of the results of numerical experiments that illustrate the benefits of our proposed techniques.

Key words. nonlinear optimization, sequential quadratic optimization, exact penalty functions, convex composite optimization, alternating direction methods, coordinate descent methods

AMS subject classifications. 49M20, 49M29, 49M37, 65K05, 65K10, 90C06, 90C20, 90C25

1. Introduction. In this paper, we consider the use of sequential quadratic optimization (commonly known as SQP) methods for solving large-scale nonlinear optimization problems (NLPs) [1, 2, 4, 8, 14, 17]. While they have proved to be effective for solving small- to medium-scale problems, SQP methods have traditionally faltered in large-scale settings due to the expense of (accurately) solving large-scale quadratic subproblems (QPs) during each iteration. However, with the use of matrix-free methods for solving these subproblems, one may consider the acceptance of inexact subproblem solutions. Such a feature offers the possibility of terminating the subproblem solver early, perhaps well before an accurate solution has been computed. This characterizes the type of strategy that we propose in this paper.

Some work has been done to provide global convergence guarantees for SQP methods that allow inexact subproblem solves [7]. However, the practical efficiency of such an approach remains an open question. A critical aspect of any implementation of such an approach is the choice of subproblem solver. This is the case as the solver must be able to provide good inexact solutions quickly, as well as have the ability to compute highly accurate solutions—say, by exploiting well-chosen starting points—in the neighborhood of a solution of the NLP. In addition, while a global convergence mechanism such as a merit function or filter is necessary to guarantee convergence from remote starting points, any NLP algorithm can suffer when such a mechanism does not immediately guide the algorithm toward promising regions of the search space. To confront this issue when an exact penalty function is used as a merit function, we propose a dynamic penalty parameter updating strategy to be incorporated *within* the subproblem solver so that each computed search direction predicts progress toward both feasibility and optimality. This strategy represents a stark contrast to

*Dept. of Mathematics, University of Washington; jvburke@uw.edu
†Dept. of Ind. and Sys. Engr., Lehigh University; frank.e.curtis@gmail.com
‡Sch. of Inf. Sci. and Tech., ShanghaiTech University; wanghao1@shanghaitech.edu.cn
§Dept. of Mathematics, University of Washington; jsw1119@math.washington.edu.

previously proposed techniques that only update the penalty parameter after a sequence of iterations, in hindsight at the end of an iteration [1, 8, 9], or at the expense of numerous subproblem solves within a single iteration [3, 5, 6].

Overall, the contributions in this paper can be summarized as the following.

- Our proposed SQP technique is specifically designed to be effective in large-scale settings. In particular, it allows for the use of iterative methods for solving the QP subproblems, allowing inexactness in the subproblem solves.
- Our technique involves a dynamic penalty parameter updating strategy to be employed *within* the subproblem solve. This makes the approach efficient while not having to accurately solve multiple QPs in a single iteration.
- By ensuring that each computed step predicts progress toward minimizing constraint violation, our technique allows for rapid infeasibility detection.

1.1. Organization. In the remainder of this section, we outline our notation and introduce various concepts that will be employed throughout the paper. In §2, we introduce a basic penalty-SQP algorithm that will form the framework for which we introduce our penalty parameter updating strategy in §3. A complete algorithm is presented and analyzed in §4. A description of important details of an implementation of our method is presented in §5. The results of numerical experiments are presented in §6. Concluding remarks are provided in §7.

1.2. Notation. Let $\mathbb{R}^n$ be the space of real n-vectors, $\mathbb{R}^n_+$ be the nonnegative orthant of $\mathbb{R}^n$ (i.e., $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x \geq 0\}$), and $\mathbb{R}^n_{++}$ be the interior of $\mathbb{R}^n_+$ (i.e., $\mathbb{R}^n_{++} := \{x \in \mathbb{R}^n : x > 0\}$). The set of $m \times n$ real matrices is denoted $\mathbb{R}^{m\times n}$. On $\mathbb{R}^n$, the ℓ_2 (i.e., Euclidean) norm is indicated as $\|\cdot\|_2$, with the unit ℓ_2-norm ball defined as $\mathbb{B}_2 := \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$. For a pair of vectors $(u, v) \in \mathbb{R}^n \times \mathbb{R}^n$, their inner product is written as $\langle u, v\rangle := u^T v$ and the line segment between them is written as $[u, v]$. The middle value operator applied to $(a, b, c) \in \mathbb{R}\times\mathbb{R}\times\mathbb{R}$, denoted by $\text{mid}\{a, b, c\}$, returns the median of $\{a, b, c\}$. For a scalar a, let $(a)_+ := \max\{a, 0\}$ and $(a)_- := \min\{a, 0\}$. The set of nonnegative integers is denoted by $\mathbb{N}$.

For a set of scalars $b_i \in \mathbb{R}$ for $i \in \{1, \ldots, m\}$, we use boldface to denote the vector $\mathbf{b} = [b_1, b_2, \ldots, b_m]^T \in \mathbb{R}^m$. For convenience, we use $\mathbf{1}_n$ to denote the n-vector of all ones and $\mathbf{0}_n$ to denote the n-vector of all zeros. Given vectors $y^i \in \mathbb{R}^{d_i}$ for $i \in \{1, \ldots, m\}$, we use boldface to denote the element $\mathbf{y} = (y^1, \ldots, y^m)$ on the product space $\mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_m}$. Conversely, given $\mathbf{y} \in \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_m}$, the i-th component of $\mathbf{y}$ (an element of $\mathbb{R}^{d_i}$) is denoted y^i while the j-th element of y^i is written as y^i_j. In this product space, we define the norm and its dual norm

$$\|\mathbf{y}\| = \left\|(y^1, \ldots, y^m)\right\| := \sum_{i=1}^m \left\|y^i\right\|_2 \quad \text{and} \quad \|\mathbf{y}\|_* = \sup_{i\in\{1,\ldots,m\}} \left\|y^i\right\|_2 .$$

For convex sets $C_i \in \mathbb{R}^{d_i}$ for $i \in \{1, \ldots, m\}$, we define the product set

$$\mathbf{C} := C_1 \times \cdots \times C_m \subset \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_m}.$$

The distance functions are defined as

$$\text{dist}_2\left(y^i \mid C_i\right) := \inf_{z^i\in C_i} \|y^i - z^i\|_2 \;\text{ and }\; \text{dist}\left(\mathbf{y} \mid \mathbf{C}\right) := \sum_{i=1}^m \text{dist}_2\left(y^i \mid C_i\right),$$

as well as the corresponding projection operators

$$P_{C_i}(y^i) := \arg\min_{z^i \in C_i} \left\|z^i - y^i\right\|_2 \quad \text{and} \quad P_{\mathbf{C}}(\mathbf{y}) := \arg\min_{\boldsymbol{z} \in \mathbf{C}} \|\boldsymbol{z} - \mathbf{y}\|.$$

The interior of a set C is denoted by $\text{int}(C)$.

For an extended-real-valued function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$, the Legendre-Fenchel conjugate of f is denoted as $f^\star$. For a convex set $X \subseteq \mathbb{R}^n$, we define the characteristic function $\delta(d|X)$ which evaluates to 0 if $d \in X$ and evaluates to ∞ otherwise. The conjugate of $\delta(\cdot|X)$ is the support function of X, which we denote by $\delta^*(y|X) = \sup_{d \in X} \langle y, d\rangle$. For example, for a hyperplane $C := \{d : \langle a, d\rangle + b = 0\}$ (respectively, half space $C = \{d : \langle a, d\rangle + b \le 0\}$), one finds that $\delta^*(y|C) < \infty$ if and only if $\langle y, a\rangle = \pm\|y\|_2\|a\|_2$ (respectively, $\langle y, a\rangle = \|y\|_2\|a\|_2$). In this case,

$$y = \zeta a \quad \text{with} \quad \zeta = \frac{1}{\|a\|_2}\langle y, a\rangle, \quad \text{meaning that} \quad \delta^*(y|C) = -\zeta b. \tag{1.1}$$

If f is convex, then the subdifferential of f at x is the set

$$\partial f(x) := \{y \in \mathbb{R}^n : f(x) + \langle y, z - x\rangle \le f(z) \text{ for all } z \in \mathbb{B}_2\}.$$

For example, the subdifferentials of our distance functions are given by (see [15])

$$\partial \text{dist}_2\left(y^i \mid C_i\right) := \begin{cases} \frac{(I - P_{C_i})y^i}{\|(I - P_{C_i})y^i\|_2} & \text{if } y^i \notin C_i \\ \mathbb{B}_2 \cap N(y^i|C_i) & \text{if } y^i \in C_i, \end{cases}$$
$$\text{and} \quad \partial \text{dist}\left(\mathbf{y} \mid C\right) := \partial \text{dist}_2\left(y^1 \mid C_1\right) \times \cdots \times \partial \text{dist}_2\left(y^m \mid C_m\right).$$

The normal cone to C_i at $y^i \in C_i$ is defined by

$$N(y^i|C_i) := \{z^i \in \mathbb{R}^{d_i} : \langle z^i, p^i - y^i\rangle \le 0 \text{ for all } p^i \in C_i\}.$$

2. A Penalty-SQP Framework. Consider the following nonlinear optimization problem with equality and inequality constraints where we assume that the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are continuously differentiable:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & f(x) \\ \text{s.t. } & c_i(x) = 0 \quad \text{for all} \quad i \in \{1, \ldots, \overline{m}\}; \\ & c_i(x) \le 0 \quad \text{for all} \quad i \in \{\overline{m} + 1, \ldots, m\}. \end{aligned} \tag{NLP}$$

Our penalty-SQP framework uses two functions for use in the algorithm and for characterizing first-order stationary solutions. First, with a penalty parameter $\rho \in \mathbb{R}_+$, we define the measure of infeasibility and exact penalty function

$$v(x) = \sum_{i=1}^{\overline{m}} |c_i(x)| + \sum_{i=\overline{m}+1}^{m} (c_i(x))_+ \quad \text{and} \quad \phi(x, \rho) = \rho f(x) + v(x).$$

Generally speaking, our penalty-SQP framework aims to solve (NLP) through systematic minimization of $\phi(\cdot, \rho)$ for appropriately chosen values of $\rho \in \mathbb{R}_{++}$. However, if the constraints of (NLP) are infeasible, then the algorithm is designed to return an infeasibility certificate in the form of a stationary point for the *feasibility problem*

$$\min_{x \in \mathbb{R}^n} \phi(x, 0), \quad \text{where} \quad \phi(x, 0) = v(x). \tag{2.1}$$

Given $\rho \in \mathbb{R}_+$ and $\eta \in \mathbb{R}^m$, we define the Fritz John function for (NLP) by

$$F(x, \rho, \eta) = \rho f(x) + \langle \eta, c(x) \rangle.$$

Note that $\rho \in \mathbb{R}_+$ plays the double role as penalty parameter in ϕ and objective multiplier in F. This makes sense from both theoretical and practical perspectives. First-order stationarity conditions for (NLP) can be written in terms of ∇F, the constraint function c, and bounds on the dual variables [7].

In the kth iteration of our penalty-SQP framework, the search direction computation is based on a local model of the penalty function about a primal iterate $x^k \in \mathbb{R}^n$ that can make use of a dual iterate $\eta^k \in \mathbb{R}^m$. We define this model by

$$J(d, \rho; x^k, \eta^k) := l(d, \rho; x^k) + \tfrac{1}{2}\langle d, H(\rho; x^k, \eta^k) d \rangle,$$

where l is a linearized model of the penalty function (ignoring $\rho f(x^k)$) defined by

$$l(d, \rho; x^k) = \rho \langle \nabla f(x^k), d \rangle + \sum_{i=1}^{\overline{m}} \left| c_i(x^k) + \langle \nabla c_i(x^k), d \rangle \right| + \sum_{i=\overline{m}+1}^{m} (c_i(x^k) + \langle \nabla c_i(x^k), d \rangle)_+$$

and H represents an approximation of $\nabla^2_{xx} F$ with

$$H(\rho; x^k, \eta^k) \approx \nabla^2_{xx} F(\rho; x^k, \eta^k) = \rho \nabla^2_{xx} f(x^k) + \sum_{i=1}^{m} \eta_i^k \nabla^2_{xx} c_i(x^k).$$

In particular, the search direction d^k is computed as an approximate minimizer of $J(\cdot, \rho_k; x^k, \eta^k)$ for some $\rho_k \in (0, \rho_{k-1}]$ over a convex set $X \subseteq \mathbb{R}^n$ containing $\{0\}$, i.e.,

$$\text{(QP)} \qquad d^k \approx \arg\min_{d \in X} \; J(d, \rho_k; x^k, \eta^k) \;\; \text{for some} \;\; \rho_k \in (0, \rho_{k-1}].$$

We introduce the set X to allow for the possibility of employing, e.g., a trust region constraint; e.g., for some $\Delta \in \mathbb{R}_+$, one may define X such that $X \subset \{d : \|d\|_2 \leq \Delta\}$.

The value $\rho_k \in (0, \rho_{k-1}]$ is computed *during* the iterative solve of (QP). Roughly speaking, we aim to adjust this value so that the (inexact) solution d^k to (QP) predicts progress toward both feasibility and optimality. In particular, this occurs if the reduction in a linearized model of the feasibility measure,

$$\text{(2.2)} \qquad \Delta l(d^k, 0; x^k) := l(0, 0; x^k) - l(d^k, 0; x^k),$$

$$\text{(2.3)} \qquad \text{where generally} \;\; \Delta l(d^k, \rho_k; x^k) := l(0, \rho_k; x^k) - l(d^k, \rho_k; x^k),$$

and the reduction in the local model of the penalty function,

$$\text{(2.4)} \qquad \Delta J(d^k, \rho_k; x^k, \eta^k) := J(0, \rho_k; x^k, \eta^k) - J(d^k, \rho_k; x^k, \eta^k),$$

are sufficiently positive, in which case d^k represents a direction of sufficient descent for both v and $\phi(\cdot, \rho_k)$ from x^k. However, if x^k is (nearly) stationary for v and/or for $\phi(\cdot, \rho_k)$, then requiring both of these reductions to be positive can force the algorithm to compute a highly accurate solution of (QP) when one is not entirely needed. Therefore, the precise conditions that (d^k, ρ_k) must satisfy—introduced in the next section—involve margins that allow one or both of these reductions to be small or even negative for an acceptable step.

Overall, the kth iteration of our penalty-SQP strategy proceeds as in Algorithm 1. First, a search direction and penalty parameter pair (d^k, ρ_k) is computed by a subproblem solver such that d^k yields reductions in the local models of the penalty function and measure of infeasibility that satisfy our conditions in §3. Then, a line search is performed with respect to the merit function $\phi(\cdot, \rho_k)$ from x^k along the search direction d^k, yielding a stepsize $\alpha_k \in \mathbb{R}_{++}$. Finally, the new iterate is set as $x^{k+1} \leftarrow x^k + \alpha_k d^k$ and the algorithm proceeds to the $(k+1)$st iteration. We discuss choices for the new dual iterate η^{k+1} with the complete algorithm in §4.

Algorithm 1 Penalty-SQP Algorithm (Preliminary)

Require: $(\gamma, \theta) \in (0,1)$ and $\rho_{-1} \in (0, \infty)$.
1: Choose $(x^0, \eta^0) \in \mathbb{R}^n \times \mathbb{R}^m$.
2: **for all** $k \in \mathbb{N}$ **do**
3: Solve (approximately) (QP) to obtain $(d^k, \rho_k) \in \mathbb{R}^n \times (0, \rho_{k-1}]$.
4: Let α^k be the largest value in $\{\gamma^0, \gamma^1, \gamma^2, \dots\}$ such that

$$\phi(x^k + \alpha_k d^k, \rho_k) - \phi(x^k, \rho_k) \le -\theta \alpha_k \Delta l(d^k, \rho_k; x^k).$$

5: Set $x^{k+1} \leftarrow x^k + \alpha_k d^k$ and choose $\eta^{k+1} \in \mathbb{R}^m$.

Before proceeding, it is worthwhile to emphasize the benefit of ignoring the term $\rho f(x^k)$ in our definitions of the models J and l above. It is valid to do this since this term has no effect on the solution of (QP), and since its presence would not affect the model reduction values in (2.2) and (2.4). On the other hand, ignoring this term simplifies our presentation and analysis significantly since it allows us to avoid the fact that, if this term were not ignored, then the optimal value of (QP) for a given x^k would shift with changes in the penalty parameter.

3. A Dynamic Penalty Parameter Updating Strategy.

In this section, we present a dynamic penalty parameter updating strategy. As mentioned, the method is novel since the update is employed *within* a solver for the subproblem arising in our penalty-SQP framework. A potential pitfall of such an approach is that, since the penalty parameter dictates the weight between the objective terms in (QP), one may disrupt typical convergence guarantees of the subproblem solver by manipulating this weight during the solution process. However, under reasonable assumptions, we prove that for sufficiently small values of the penalty parameter, our updating strategy will no longer be triggered. Consequently, once the penalty parameter reaches a sufficiently small value, it will remain fixed and the subproblem solver will effectively be applied to solve (QP) for a fixed value ρ_k. The updating strategy is described in a manner that allows it to be incorporated into various subproblem solvers; see §5.2.

3.1. Preliminaries.

For ease of exposition in this section, we drop the dependence of certain quantities on the iteration number:

$$\begin{aligned} &g = \nabla f(x^k),\ a^i = \nabla c_i(x^k),\ b_i = c_i(x^k),\ A = [a^1, \cdots, a^m]^T, \\ &H_f \approx \nabla^2_{xx} f(x^k),\ H_0 \approx \sum_{i=1}^m \eta_i^k \nabla^2_{xx} c_i(x^k),\ \text{and } H_\rho = \rho H_f + H_0. \end{aligned} \tag{3.1}$$

We also temporarily drop the dependence of the functions J, l, etc. on the kth iterate.

We make the following assumption about the subproblem data.

Assumption 1. *The subproblem data matrices A, H_f, and H_0 are such that*
(i) H_ρ is positive definite for any $\rho \in [0, \rho_{k-1}]$; and
(ii) $\|a^i\|_2 > 0$ for all $i \in \{1, \dots, m\}$.

We claim that this assumption is reasonable due to the following considerations. First, in large-scale contexts, it is typically impractical to construct complete second-derivative matrices. Hence, as indicated in (3.1), one can assume that H_f and H_0 represent (limited memory) Hessian approximations with at least H_0 being positive definite. (See §5.1 for further discussion.) Second, if $a^i = 0$ for any $i \in \{1, \dots, m\}$, then the model of the ith constraint is constant with respect to d, meaning that the ith constraint can be removed from the subproblem. Such a phenomenon can be detected during a preprocessing phase before solving the subproblem, so for simplicity, we assume that each constraint gradient is nonzero. Under Assumption 1, we define the scaled quantities $\bar{a}^i := a^i/\|a^i\|_2$ and $\bar{b}_i := b_i/\|a^i\|_2$ for all $i \in \{1, \dots, m\}$.

Of central importance in the subproblems are the convex sets

$$
\begin{aligned}
C_i &:= \{d \in \mathbb{R}^n : \langle \bar{a}^i, d\rangle + \bar{b}_i = 0\} \quad \text{for all} \quad i \in \{1, \dots, \overline{m}\} \\
\text{and} \quad C_i &:= \{d \in \mathbb{R}^n : \langle \bar{a}^i, d\rangle + \bar{b}_i \le 0\} \quad \text{for all} \quad i \in \{\overline{m}+1, \dots, m\}.
\end{aligned}
$$

The quadratic and penalty terms in J can be written, respectively, as

$$
\varphi(d, \rho) = \rho\langle g, d\rangle + \tfrac{1}{2}\langle d, H_\rho d\rangle \quad \text{and} \quad l(d, 0) = \sum_{i=1}^m \|a^i\|_2 \operatorname{dist}_2(d \mid C_i),
$$

meaning that we may rewrite the penalty-SQP subproblem (QP) as

$$
\text{(QPrho)} \qquad \min_{d \in \mathbb{R}^n} J(d, \rho), \quad \text{where} \quad J(d, \rho) = \varphi(d, \rho) + l(d, 0) + \delta(d|X).
$$

We refer to (QPrho) with $\rho > 0$ as a *penalty subproblem* and we refer to (QPrho) with $\rho = 0$ as the *feasibility subproblem*. The Fenchel–Rockafellar dual of (QPrho) is

$$
\text{(DQPrho)} \qquad \max_{\mathbf{u} \in \mathbb{R}^n \times \cdots \times \mathbb{R}^n} D(\mathbf{u}, \rho) \quad \text{s.t.} \quad u^0 + \sum_{i=1}^m \|a^i\|_2 u^i + u^{m+1} = 0
$$
$$
\text{and} \quad u^i \in \mathbb{B}_2 \quad \text{for all} \quad i \in \{1, \dots, m\},
$$

where the dual objective function is given by

$$
D(\mathbf{u}, \rho) = -\tfrac{1}{2}\langle u^0 - \rho g, H_\rho^{-1}(u^0 - \rho g)\rangle - \sum_{i=1}^m \|a^i\|_2 \delta^*(u^i|C_i) - \delta^*(u^{m+1}|X).
$$

Letting $\zeta_i(\mathbf{u}) := \langle u^i, \bar{a}^i\rangle$ for a dual feasible $\mathbf{u}$, one finds from (1.1) and the constraint in (DQPrho) that $D(\mathbf{u}, \rho)$ is finite if and only if

$$
\begin{aligned}
& u^i = \zeta_i(\mathbf{u})\bar{a}^i, \\
\text{which means} \quad & \zeta_i(\mathbf{u}) \in \begin{cases} [-1, 1] & \text{for all } i \in \{1, ..., \bar{m}\} \\ [0, 1] & \text{for all } i \in \{\bar{m}+1, ..., m\}, \end{cases} \\
\text{and} \quad & \delta^*(u^i|C_i) = -\zeta_i(\mathbf{u})\bar{b}^i.
\end{aligned} \tag{3.2}
$$

An interesting aspect of the dual subproblem (DQPrho) is that the penalty parameter appears only in the objective. Thus, if $\mathbf{u}$ satisfies the constraints of (DQPrho), then it is dual-feasible regardless of the value of ρ appearing in the subproblem. As a result, by weak duality, we have for any primal-dual feasible pair $(d, \mathbf{u})$ that both

$$D(\mathbf{u}, 0) \leq J(d, 0) \quad \text{and} \quad D(\mathbf{u}, \rho) \leq J(d, \rho). \tag{3.3}$$

We close this subsection by noting that the projection onto the set C_i is easy to compute for any $i \in \{1, \ldots, m\}$; in particular,

$$P_{C_i}(d) = \begin{cases} d - (\langle \bar{a}^i, d\rangle + \bar{b}_i)\bar{a}^i & \text{for all } \ i \in \{1, \ldots, \overline{m}\} \\ d - (\langle \bar{a}^i, d\rangle + \bar{b}_i)_+\bar{a}^i & \text{for all } \ i \in \{\overline{m}+1, \ldots, m\}. \end{cases}$$

3.2. Updating the penalty parameter. Given $\rho \geq 0$, let $(d^*_\rho, \mathbf{u}^*_\rho)$ represent an optimal primal-dual pair for the penalty subproblem (QPrho) corresponding to ρ; in particular, $(d^*_0, \mathbf{u}^*_0)$ represents an optimal primal-dual pair for the feasibility subproblem. The algorithm is presented in the context of a subproblem solver that generates two sequences of iterates: the first sequence, call it $\{(d^{(j)}, \mathbf{u}^{(j)})\}$, is a sequence of primal-dual feasible solution estimates for a penalty subproblem, while the second sequence, call it $\{\mathbf{w}^{(j)}\}$, is a sequence of dual feasible solution estimates for the feasibility subproblem. (In our strategy, we do not make separate use of a sequence of primal solution estimates for the feasibility subproblem; rather, the sequence $\{d^{(j)}\}$ plays this role as well.) Without loss of generality, we assume that the jth primal solution estimate $d^{(j)}$ represents a better (or no worse) primal solution estimate for the penalty subproblem than a zero step in the sense that

$$J(d^{(j)}, \rho_{(j)}) \leq J(0, \rho_{(j)}). \tag{3.4}$$

Similarly, we assume that the dual solution estimate $\mathbf{w}^{(j)}$ represents a better (or no worse) dual solution estimate for the feasibility subproblem than $\mathbf{u}^{(j)}$, and that each dual solution estimate $\mathbf{u}^{(j)}$ is no worse than the feasible $\mathbf{u}^{(0)}$, in that

$$D(\mathbf{w}^{(j)}, 0) \geq D(\mathbf{u}^{(j)}, 0) \geq D(\mathbf{u}^{(0)}, 0) > -\infty. \tag{3.5}$$

These are both reasonable assumptions since if (3.4) (resp. (3.5)) were not to hold, then one could consider $d^{(j)} = 0$ (resp. $\mathbf{w}^{(j)} = \mathbf{u}^{(j)} = \mathbf{u}^{(0)}$) for the jth iterate (even if the subproblem solver works with a different estimate in its internal operations).

Observe that, by the definition of the model J, we have for any $\rho \in (0, \infty)$ that

$$J^{(0)} := J(0, \rho) = J(0, 0) = l(0, 0) = \sum_{i=1}^{\bar{m}} |b_i| + \sum_{i=\bar{m}+1}^{m} (b_i)_+ \geq 0.$$

Let $J^{(0)}_\omega := J^{(0)} + \omega$ for any scalar $\omega \in (0, \infty)$. (As discussed later, ω is held fixed during a given subproblem solve, but will sequentially be reduced to zero over the course of the overall penalty-SQP framework.) We then define the following ratios corresponding to the jth subproblem solver iterate:

$$r_v^{(j)} := \frac{J^{(0)}_\omega - l(d^{(j)}, 0)}{J^{(0)}_\omega - (D(\mathbf{w}^{(j)}, 0))_+} \quad \text{and} \quad r_\phi^{(j)} := \frac{J^{(0)}_\omega - J(d^{(j)}, \rho_{(j)})}{J^{(0)}_\omega - D(\mathbf{u}^{(j)}, \rho_{(j)})}. \tag{3.6}$$

(Referring back to our discussion surrounding (2.2) and (2.4), note that the numerators of these ratios are $\Delta l(d^{(j)}, 0) + \omega$ and $\Delta J(d^{(j)}, \rho^{(j)}) + \omega$, respectively.) The critical property of these ratios is that, if they are sufficiently large, then the corresponding subproblem solver iterates must yield reductions in the feasibility and penalty function models that are proportional to those obtained by corresponding exact subproblem solutions. In particular, suppose that for some prescribed $\beta_v \in (0,1)$ we have

$$r_v^{(j)} \geq \beta_v. \tag{Rv}$$

Then the reduction in the linearized constraint violation model obtained by the subproblem solver iterate $d^{(j)}$ relative to a zero step satisfies

$$\begin{aligned} J_\omega^{(0)} - l(d^{(j)}, 0) &\geq \beta_v \left(J_\omega^{(0)} - (D(\mathbf{w}^{(j)}, 0))_+ \right) \\ &\geq \beta_v \left(J_\omega^{(0)} - D(\mathbf{u}_0^*, 0) \right) = \beta_v \left(J_\omega^{(0)} - J(d_0^*, 0) \right), \end{aligned} \tag{3.7}$$

where the first inequality follows by (Rv), the second follows by the optimality of $\mathbf{u}_0^*$ with respect to the feasibility subproblem (for which it is known that $D(\mathbf{u}_0^*, 0) \geq 0$), and the last follows by strong duality. Similarly, if for $\beta_\phi \in (0,1)$ we have

$$r_\phi^{(j)} \geq \beta_\phi, \tag{Rphi}$$

then it follows that

$$\begin{aligned} J_\omega^{(0)} - J(d^{(j)}, \rho_{(j)}) &\geq \beta_\phi (J_\omega^{(0)} - D(\mathbf{u}^{(j)}, \rho_{(j)})) \\ &\geq \beta_\phi (J_\omega^{(0)} - D(\mathbf{u}_{\rho_{(j)}}^*, \rho_{(j)})) = \beta_\phi (J_\omega^{(0)} - J(d_{\rho_{(j)}}^*, \rho_{(j)})). \end{aligned} \tag{3.8}$$

The last component of our updating strategy involves an estimate of the complementarity of a primal-dual solution estimate. This is needed since we only reduce the penalty parameter if a primal-dual solution estimate is approximately complementary. We do this in the following manner. First, defining the index sets

$$\begin{aligned} \mathcal{E}_+(d) &:= \{i \in \{1, \ldots, \bar{m}\} : \langle \bar{a}^i, d \rangle + \bar{b}_i > 0\}, \\ \mathcal{E}_-(d) &:= \{i \in \{1, \ldots, \bar{m}\} : \langle \bar{a}^i, d \rangle + \bar{b}_i < 0\}, \\ \text{and } \ \mathcal{I}_+(d) &:= \{i \in \{\bar{m}+1, \ldots, m\} : \langle \bar{a}^i, d \rangle + \bar{b}_i > 0\}, \end{aligned}$$

we define the complementarity measure

$$\chi(d, \mathbf{u}) := \sum_{i \in \mathcal{E}_+ \cup \mathcal{I}_+} (1 - \zeta_i(\mathbf{u})) \|a^i\|_2 \text{dist}\left(d \mid C_i\right) + \sum_{i \in \mathcal{E}_-} (1 + \zeta_i(\mathbf{u})) \|a^i\|_2 \text{dist}\left(d \mid C_i\right).$$

To reduce the penalty parameter, we require that $(d^{(j)}, \mathbf{u}^{(j)})$ satisfies

$$\chi^{(j)} := \chi(d^{(j)}, \mathbf{u}^{(j)}) \leq (1 - \beta_v)^2 J_\omega^{(0)},$$

or, equivalently,

$$r_c^{(j)} := \sqrt{\frac{J_\omega^{(0)} - \chi^{(j)}}{J_\omega^{(0)}}} \geq \beta_v. \tag{Rc}$$

Overall, our penalty parameter strategy is motivated by the desire to ensure that if the jth iterate of the subproblem solver offers a sufficiently accurate solution of the penalty subproblem for $\rho_{(j)} > 0$, then it should also offer a sufficiently accurate solution of the feasibility subproblem; otherwise, the penalty parameter should be reduced. Specifically, choosing parameters

$$0 < \beta_v < \beta_\phi < 1, \tag{3.9}$$

we initialize $\rho_{(0)} \leftarrow \rho_{k-1}$ (from the preceding iteration of the penalty-SQP framework) and apply the subproblem solver to (QPrho) to initialize $\{(d^{(j)}, \mathbf{u}^{(j)}, \mathbf{w}^{(j)})\}$. If, at the end of the jth subproblem solver iteration we have that (Rphi) or (Rc) is not satisfied, then we continue to iterate toward solving (QPrho) with $\rho = \rho_{(j)}$. Otherwise, if (Rphi) and (Rc) hold but (Rv) does not, then we reduce the penalty parameter by setting

$$\rho_{(j+1)} \leftarrow \theta_\rho \rho_{(j)} \tag{3.10}$$

for some prescribed $\theta_\rho \in (0, 1)$. (A special case that one should consider occurs when (Rphi), (Rc), and (Rv) all hold with $d^{(j)} = 0$. For simplicity in our presentation, in such a case, we have the subproblem solver terminate with $d^{(j)} = 0$, causing the penalty-SQP framework to take a null step in the primal space. As previously mentioned, this would be followed by a decrease in ω, prompting the penalty-SQP framework to eventually make further progress or terminate with a stationarity certificate. In practice, this decrease in ω in this scenario need not occur over a sequence of iterations. It can occur immediately within a subproblem solve. We merely state the occurence of a null step for simplicity in our discussions.)

We state our *d*ynamic *u*pdating *st*rategy (DUST) as:

(DUST)

Given $\rho_{(j)}$ and the jth iterate $(d^{(j)}, \mathbf{u}^{(j)}, \mathbf{w}^{(j)})$, perform the following:
• if (Rphi), (Rc), and (Rv) hold, then terminate;
• else if (Rphi) and (Rc) hold, but (Rv) does not, then apply (3.10);
• else set $\rho_{(j+1)} \leftarrow \rho_{(j)}$.

We formally analyze (DUST) in the following subsections. We begin with the following intuitive arguments to motivate the strategy for adjusting the penalty parameter in a few cases of interest. These cases depend on properties of the kth iterate of the penalty-SQP framework, namely, x^k, with respect to the constraint violation measure and the penalty function.

- First, observe that with an optimal primal-dual solution $(d^*_\rho, \mathbf{u}^*_\rho)$ for a penalty subproblem, one has $\zeta_i(\mathbf{u}^*_\rho) = 1$ for $i \in \mathcal{E}_+(d^*_\rho)$, $\zeta_i(\mathbf{u}^*_\rho) = -1$ for $i \in \mathcal{E}_-(d^*_\rho)$, and $\zeta_i(\mathbf{u}^*_\rho) = 1$ for $i \in \mathcal{I}_+(d^*_\rho)$, from which it follows that $\chi(d^*_\rho, \mathbf{u}^*_\rho) = 0$. Therefore, for a given $\omega \in (0, \infty)$, the condition (Rc) will hold for sufficiently accurate primal-dual solutions of the penalty subproblem.
- If x^k is not stationary with respect to $\phi(\cdot, \rho)$ for any $\rho \in (0, \rho_{k-1}]$, then, with $(d^{(j)}, \mathbf{u}^{(j)}, \rho_{(j)}) = (d_{\rho_*}, \mathbf{u}_{\rho_*}, \rho)$ for any such ρ, one finds that $r^{(j)}_\phi = 1 > \beta_\phi$. In turn, this means that (Rphi) holds for any $(d^{(j)}, \mathbf{u}^{(j)})$ in a neighborhood of $(d_{\rho_*}, \mathbf{u}_{\rho_*})$. If, in addition, x^k is not stationary with respect to v, then one should expect that for a sufficiently small $\rho_{(j)}$ the condition (Rv) would also be satisfied for such a $d^{(j)}$. This should be expected since for $(d^*_0, \mathbf{u}^*_0)$ one has

$$\frac{J^{(0)}_\omega - l(d^*_0, 0)}{J^{(0)}_\omega - (D(\mathbf{u}^*_0, 0))_+} \geq \frac{J^{(0)}_\omega - J(d^*_0, 0)}{J^{(0)}_\omega - D(\mathbf{u}^*_0, 0)} = 1,$$

meaning that $r_v^{(j)} > \beta_v$ for $(d^{(j)}, \mathbf{w}^{(j)})$ in a neighborhood of $(d_0^*, \mathbf{u}_0^*)$. Overall, in this case, one should expect that (DUST) would only reduce the penalty parameter a finite number of times, if at all.

- If x^k is not stationary with respect to $\phi(\cdot, \rho)$ for any $\rho \in (0, \rho_{k-1}]$, but is stationary with respect to v, then for $(d_0^*, \mathbf{u}_0^*)$ one has

$$\frac{J_\omega^{(0)} - l(d_0^*, 0)}{J_\omega^{(0)} - (D(\mathbf{u}_0^*, 0))_+} = \frac{\omega}{\omega} = 1,$$

meaning that $r_v^{(j)} > \beta_v$ for $(d^{(j)}, \mathbf{w}^{(j)})$ in a neighborhood of $(d_0^*, \mathbf{u}_0^*)$. Hence, as in the previous bullet, one should expect that (DUST) would only reduce the penalty parameter a finite number of times.

- If x^k is stationary with respect to $\phi(\cdot, \rho_{(j)})$ for $\rho_{(j)} > 0$ encountered during the subproblem solve, then, under Assumption 1, the only primal iterate satisfying (Rphi) is $d^{(j)} = 0$. For this value, one finds that

$$r_v^{(j)} = \frac{\omega}{\omega + J^{(0)} - (D(\mathbf{w}^{(j)}, 0))_+}.$$

There are now two cases to consider. If $r_v^{(j)} < \beta_v$, then (DUST) decreases the penalty parameter, as is appropriate. Otherwise, if $r_v^{(j)} \geq \beta_v$, then—with a sufficiently accurate dual solution—(DUST) returns a null step to the penalty-SQP framework. (In a later subproblem solve with a smaller ω, one would either find that (Rphi) holds for $d^{(j)} = 0$—and a sufficiently accurate dual solution—but (Rv) does not, prompting a decrease of the penalty parameter, or—again with a sufficiently accurate dual solution—one would terminate the overall algorithm with certificate of stationarity for x^k.)

We close this subsection by making a few practical remarks regarding the use of (DUST) within a subproblem solver for (QPrho). In particular, while we have defined the sequence $\{(d^{(j)}, \mathbf{u}^{(j)}, \mathbf{w}^{(j)})\}$ as being generated by the solver, it may be reasonable to reinitialize the solver—or at least perform some auxiliary computations—after any iteration in which (3.10) is invoked. (Such auxiliary computations may involve scaling vectors and/or matrices due to the change in the penalty parameter; e.g., see the discussion of the Hessian approximation strategy in §5.1.) That being said, it is reasonable to assume that, during any sequence of iterations in which the penalty parameter does not change, the subproblem solver would be applied as if it were being applied to a static instance of (QPrho). In such a manner, any convergence guarantees for the subproblem solver would hold if/when the penalty parameter stabilizes at a fixed value, as is guaranteed to occur under common conditions described next.

3.3. Finite Updates for a Single Subproblem. The purpose of this subsection is to show that if (DUST) is employed within an algorithm for solving (QPrho), then, under reasonable assumptions on the subproblem data, for any $\rho_{(j)} \in (0, \tilde{\rho}]$ for some sufficiently small $\tilde{\rho} > 0$ whose value depends only on the subproblem data, if (Rphi) and (Rc) are satisfied, then (Rv) is also satisfied. In other words, after a finite number of iterations, the update (3.10) will never be triggered. Let $\underline{\lambda}_0$ and $\overline{\lambda}_0$ be the smallest and largest eigenvalues of H_0, and similarly for $\underline{\lambda}_\rho$ and $\overline{\lambda}_\rho$ with respect to the matrix H_ρ. Notice that, since $\rho_{(j)} \in (0, \rho_{(0)}]$, it follows that

$$\underline{\lambda}_{\rho_{(j)}} \geq \underline{\lambda} := \min\{\underline{\lambda}_{\rho_{(0)}}, \underline{\lambda}_0\} \quad \text{and} \quad \overline{\lambda}_{\rho_{(j)}} \leq \overline{\lambda} := \max\{\overline{\lambda}_{\rho_{(0)}}, \overline{\lambda}_0\}. \tag{3.11}$$

We formalize our assumption for this analysis as the following.

Assumption 2. *For all $j \in \mathbb{N}$, the sequence $\{(d^{(j)}, \mathbf{u}^{(j)}, \mathbf{w}^{(j)})\}$ has $d^{(j)} \in X$, (3.4) and (3.5) hold, and $\mathbf{u}^{(j)}$ and $\mathbf{w}^{(j)}$ are feasible for (DQPrho).*

We first show that the dual sequences $\{\mathbf{u}^{(j)}\}$ and $\{\mathbf{w}^{(j)}\}$ are bounded in norm.

Lemma 3. *Under Assumption 1, there exists $\kappa_0 > 0$ such that, for all $j \in \mathbb{N}$,*

$$\|\mathbf{u}^{(j)}\|_2 \leq \kappa_0 \quad \textit{and} \quad \|\mathbf{w}^{(j)}\|_2 \leq \kappa_0.$$

Proof. Since $\mathbf{u}^{(j)}$ is feasible for (DQPrho), the elements $\{(u^i)^{(j)}\}$ for all $i \in \{1, \ldots, m\}$ are bounded in norm by 1. Therefore, by the first constraint of (DQPrho), it suffices to show that $\{(u^0)^{(j)}\}$ is bounded. We show this by contradiction. Suppose there exists an infinite index set $\mathcal{J}$ such that $\{\|(u^0)^{(j)}\|_2\}_{j\in\mathcal{J}} \nearrow \infty$. Notice that for $(u^{m+1})^{(j)}$ it holds that $\delta^*((u^{m+1})^{(j)}|X) = \sup_{x\in X}\langle (u^{m+1})^{(j)}, x\rangle \geq 0$ since it is assumed that $0 \in X$. All together, with these facts and Assumption 1, we may conclude that $\{D(\mathbf{u}^{(j)}, 0)\}_{j\in\mathcal{J}} \to -\infty$, which contradicts (3.5). Therefore, $\{(u^0)^{(j)}\}$ must be bounded, so overall the sequence $\{\mathbf{u}^{(j)}\}$ is bounded.

Following the same argument for $\mathbf{w}^{(j)}$, it follows that $\{\mathbf{w}^{(j)}\}$ is bounded. ∎

We now show that the primal variables $\{d^{(j)}\}$ are also bounded in norm.

Lemma 4. *Under Assumptions 1 and 2, it follows that, for all $j \in \mathbb{N}$,*

$$\|d^{(j)}\|_2 \leq \kappa_1 := \left(\rho_{(0)}\|g\|_2 + \sqrt{\rho_{(0)}^2\|g\|_2^2 + 2\overline{\lambda}J^{(0)}}\right)/\underline{\lambda}. \tag{3.12}$$

Proof. By Assumption 2, it follows that $d^{(j)} \in X$ for all $j \in \mathbb{N}$, which implies that $\delta(d^{(j)}|X) = 0$ for all $j \in \mathbb{N}$. By (3.4), every $(d^{(j)}, \mathbf{u}^{(j)}, \rho_{(j)})$ for $j \in \mathbb{N}$ must satisfy

$$\rho_{(j)}\langle g, d^{(j)}\rangle + \tfrac{1}{2}\langle d^{(j)}, H_{\rho_{(j)}} d^{(j)}\rangle \leq J(d^{(j)}, \rho_{(j)}) \leq J(0, \rho_{(j)}) = J^{(0)}.$$

It follows that

$$\tfrac{1}{2}\underline{\lambda}_{\rho_{(j)}}\|d^{(j)}\|_2^2 \leq J^{(0)} + |\rho_{(j)}\langle g, d^{(j)}\rangle| \leq J^{(0)} + \rho_{(0)}\|g\|_2\|d^{(j)}\|_2,$$

which, using the quadratic formula, implies that

$$\|d^{(j)}\|_2 \leq \left(\rho_{(0)}\|g\|_2 + \sqrt{\rho_{(0)}^2\|g\|_2^2 + 2\underline{\lambda}_{\rho_{(j)}}J^{(0)}}\right)/\underline{\lambda}_{\rho_{(j)}}.$$

Together with (3.11), this proves (3.12), as desired. ∎

The next lemma shows that the differences between the primal and dual values of the penalty and feasibility subproblems are bounded with respect to ρ.

Lemma 5. *Under Assumptions 1 and 2, it follows that, for any $j \in \mathbb{N}$,*

$$|J(d^{(j)}, \rho_{(j)}) - J(d^{(j)}, 0)| \leq \kappa_2\rho_{(j)} \tag{3.13a}$$

$$\textit{and} \quad |D(\mathbf{u}^{(j)}, \rho_{(j)}) - D(\mathbf{u}^{(j)}, 0)| \leq \kappa_3\rho_{(j)}, \tag{3.13b}$$

where, with $\kappa_1 > 0$ defined in Lemma 4,

$$\kappa_2 := \|g\|_2\kappa_1 + \tfrac{1}{2}\|H_f\|_2\kappa_1^2$$

$$\textit{and} \quad \kappa_3 := \frac{\kappa_0 + \rho_{(0)}\|g\|_2}{2\underline{\lambda}}(\kappa_0\|H_0^{-1}\|_2\|H_f\|_2 + \|g\|_2) + \tfrac{1}{2}\kappa_0\|H_0^{-1}\|_2\|g\|_2.$$

Proof. For the primal values, it holds true that

$$\begin{aligned}|J(d^{(j)},\rho_{(j)})-J(d^{(j)},0)| &= |\rho_{(j)}\langle g,d^{(j)}\rangle+\tfrac{1}{2}\langle d^{(j)},H_{\rho_{(j)}}d^{(j)}\rangle-\tfrac{1}{2}\langle d^{(j)},H_0d^{(j)}\rangle| \\ &= |\rho_{(j)}\langle g,d^{(j)}\rangle+\tfrac{1}{2}\rho_{(j)}\langle d^{(j)},H_fd^{(j)}\rangle| \\ &\leq \rho_{(j)}(\|g\|_2\|d^{(j)}\|_2+\tfrac{1}{2}\|H_f\|_2\|d^{(j)}\|_2^2),\end{aligned}$$

which combined with Lemma 4 proves (3.13a).

We now aim to prove (3.13b). Toward this goal, let $\hat{y}^{(j)}:=H_{\rho_{(j)}}^{-1}(u_0^{(j)}-\rho_{(j)}g)$ and $\bar{y}^{(j)}:=H_0^{-1}u_0^{(j)}$. Then, by Assumption 2, it follows that

$$\|\hat{y}^{(j)}\|_2\leq(\kappa_0+\rho_{(j)}\|g\|_2)/\underline{\lambda}_{\rho_{(j)}}\leq(\kappa_0+\rho_{(0)}\|g\|_2)/\underline{\lambda}.$$

In addition, it follows that

$$\rho_{(j)}g=u_0^{(j)}-(u_0^{(j)}-\rho_{(j)}g)=H_0\bar{y}^{(j)}-H_{\rho_{(j)}}\hat{y}^{(j)}=H_0(\bar{y}^{(j)}-\hat{y}^{(j)})-\rho_{(j)}H_f\hat{y}^{(j)},$$

which implies that, for all $j\in\mathbb{N}$,

$$\begin{aligned}\|\bar{y}^{(j)}-\hat{y}^{(j)}\|_2 &= \|\rho_{(j)}H_0^{-1}(H_f\hat{y}^{(j)}+g)\|_2 \\ &\leq \rho_{(j)}\|H_0^{-1}\|_2\|H_f\hat{y}^{(j)}+g\|_2 \\ &\leq \rho_{(j)}\|H_0^{-1}\|_2\left(\|H_f\|_2\frac{\kappa_0+\rho_{(0)}\|g\|_2}{\underline{\lambda}}+\|g\|_2\right).\end{aligned}\tag{3.14}$$

The difference between the dual values is then given by

$$\begin{aligned}&|D(\mathbf{u}^{(j)},\rho_{(j)})-D(\mathbf{u}^{(j)},0)| \\ &= |-\tfrac{1}{2}\langle u_0^{(j)}-\rho_{(j)}g,H_{\rho_{(j)}}^{-1}(u_0^{(j)}-\rho_{(j)}g)\rangle+\tfrac{1}{2}\langle u_0^{(j)}),H_0^{-1}u_0^{(j)}\rangle| \\ &= |\tfrac{1}{2}\langle\bar{y}^{(j)}-\hat{y}^{(j)},u_0^{(j)}\rangle+\tfrac{1}{2}\rho_{(j)}\langle g,\hat{y}^{(j)}\rangle| \\ &\leq \tfrac{1}{2}\|\bar{y}^{(j)}-\hat{y}^{(j)}\|_2\|u_0^{(j)}\|_2+\tfrac{1}{2}\rho_{(j)}\|g\|_2\|\hat{y}^{(j)}\|_2 \\ &\leq \rho_{(j)}\left(\tfrac{1}{2}\|H_0^{-1}\|_2\left(\|H_f\|_2\frac{\kappa_0+\rho_{(0)}\|g\|_2}{\underline{\lambda}}+\|g\|_2\right)\kappa_0+\tfrac{1}{2}\|g\|_2\frac{\kappa_0+\rho_{(0)}\|g\|_2}{\underline{\lambda}}\right) \\ &= \rho_{(j)}\left(\frac{\kappa_0+\rho_{(0)}\|g\|_2}{2\underline{\lambda}}\left(\kappa_0\|H_0^{-1}\|_2\|H_f\|_2+\|g\|_2\right)+\tfrac{1}{2}\kappa_0\|H_0^{-1}\|_2\|g\|_2\right),\end{aligned}$$

where the last inequality follows by (3.14) and Assumption 2. ∎

Let us now define

$$\mathcal{U}=\{j:(d^{(j)},\mathbf{u}^{(j)})\text{ satisfies (Rphi) and (Rc) but not (Rv)}\},$$

meaning that $\mathcal{U}$ is the set of subproblem iterations in which (3.10) is triggered. Now we are ready to prove our main result in this section.

THEOREM 6. *Suppose Assumptions 1 and 2 hold and let*

$$\kappa_4:=\inf_{j\in\mathcal{U}}\{J^{(0)}-J(d^{(j)},\rho_{(j)})\}\geq 0\quad\textit{and}\quad\kappa_5:=\inf_{j\in\mathcal{U}}\{J^{(0)}-D(\mathbf{u}^{(j)},0)\}\geq 0.$$

Then, for $\rho_{(j)}\in(0,\tilde{\rho}]$*, where*

$$\tilde{\rho}:=\frac{\omega+\min\{\kappa_4,\kappa_5\}}{\max\{\kappa_2,\kappa_3\}}\left(1-\sqrt{\beta_v/\beta_\phi}\right),\tag{3.15}$$

if $(d^{(j)}, \mathbf{u}^{(j)})$ *satisfies* (Rphi) *and* (Rc)*, then* $(d^{(j)}, \mathbf{w}^{(j)})$ *satisfies* (Rv)*. In other words, for any* $\rho_{(j)} \in (0, \tilde{\rho}]$*, the update* (3.10) *is never triggered by* (DUST).

Proof. In order to derive a contradiction, suppose that $\mathcal{U}$ is infinite, meaning that the subproblem solver is never terminated and $\rho_{(j)} \to 0$. We have from (3.13a) that

$$-\kappa_2 \rho_{(j)} \leq J(d^{(j)}, \rho_{(j)}) - J(d^{(j)}, 0) \leq \kappa_2 \rho_{(j)} \quad \text{for any} \quad j \in \mathcal{U},$$

which, after adding and dividing through by $J_\omega^{(0)} - J(d^{(j)}, \rho_{(j)})$, yields for $j \in \mathcal{U}$ that

$$1 - \frac{\kappa_2 \rho_{(j)}}{J_\omega^{(0)} - J(d^{(j)}, \rho_{(j)})} \leq \frac{J_\omega^{(0)} - J(d^{(j)}, 0)}{J_\omega^{(0)} - J(d^{(j)}, \rho_{(j)})} \leq 1 + \frac{\kappa_2 \rho_{(j)}}{J_\omega^{(0)} - J(d^{(j)}, \rho_{(j)})}. \tag{3.16}$$

Thus, for any

$$\rho_{(j)} \leq \frac{\omega + \kappa_4}{\kappa_2} \left(1 - \sqrt{\frac{\beta_v}{\beta_\phi}}\right) \leq \frac{J_\omega^{(0)} - J(d^{(j)}, \rho_{(j)})}{\kappa_2} \left(1 - \sqrt{\frac{\beta_v}{\beta_\phi}}\right),$$

it follows from the first inequality of (3.16) that

$$\frac{J_\omega^0 - J(d^{(j)}, 0)}{J_\omega^0 - J(d^{(j)}, \rho_{(j)})} \geq \sqrt{\frac{\beta_v}{\beta_\phi}}. \tag{3.17}$$

Following an argument similar to that for (3.13b), we have that for any

$$\rho_{(j)} \leq \frac{\omega + \kappa_5}{\kappa_3} \left(1 - \sqrt{\frac{\beta_v}{\beta_\phi}}\right) \leq \frac{J_\omega^{(0)} - D(\mathbf{u}^{(j)}, 0)}{\kappa_3} \left(1 - \sqrt{\frac{\beta_v}{\beta_\phi}}\right),$$

one finds that

$$\frac{J_\omega^0 - D(\mathbf{u}^{(j)}, \rho_{(j)})}{J_\omega^0 - D(\mathbf{u}^{(j)}, 0)} \geq \sqrt{\frac{\beta_v}{\beta_\phi}}. \tag{3.18}$$

Overall, we have shown that for any $\rho_{(j)} \leq \tilde{\rho}$ with $\tilde{\rho}$ defined in (3.15), it follows that (3.17) and (3.18) both hold true and, since $D(\mathbf{w}^{(j)}, 0) \geq D(\mathbf{u}^{(j)}, 0)$, that

$$\frac{J_\omega^0 - D(\mathbf{u}^{(j)}, \rho_{(j)})}{J_\omega^0 - D(\mathbf{w}^{(j)}, 0)} \geq \frac{J_\omega^0 - D(\mathbf{u}^{(j)}, \rho_{(j)})}{J_\omega^0 - D(\mathbf{u}^{(j)}, 0)} > \sqrt{\frac{\beta_v}{\beta_\phi}}. \tag{3.19}$$

Since our supposition that $\mathcal{U}$ is infinite implies that $\rho_{(j)} \to 0$, we may now proceed under the assumption that $j \in \mathcal{U}$ with $\rho_{(j)} \in (0, \tilde{\rho}]$. Let us now define the ratios

$$\hat{r}_v^{(j)} := \frac{J_\omega^{(0)} - J(d^{(j)}, 0)}{J_\omega^{(0)} - (D(\mathbf{w}^{(j)}, 0))_+} \quad \text{and} \quad \bar{r}_v^{(j)} := \frac{J_\omega^{(0)} - J(d^{(j)}, 0)}{J_\omega^{(0)} - D(\mathbf{w}^{(j)}, 0)},$$

where, since $J(d^{(j)}, 0) = l(d^{(j)}, 0) + \frac{1}{2}\langle d^{(j)}, H_0 d^{(j)} \rangle \geq l(d^{(j)}, 0)$ and by the definition of the operator $(\cdot)_+$, it follows that $r_v^{(j)} \geq \hat{r}_v^{(j)} \geq \bar{r}_v^{(j)}$. From (3.17) and (3.19),

$$\frac{\bar{r}_v^{(j)}}{r_\phi^{(j)}} = \frac{J_\omega^0 - J(d^{(j)}, 0)}{J_\omega^0 - J(d^{(j)}, \rho_{(j)})} \frac{J_\omega^0 - D(\mathbf{u}^{(j)}, \rho_{(j)})}{J_\omega^0 - D(\mathbf{w}^{(j)}, 0)} \geq \frac{\beta_v}{\beta_\phi},$$

yielding

$$r_v^{(j)} \geq \bar{r}_v^{(j)} \geq \frac{\beta_v}{\beta_\phi} r_\phi^{(j)} \geq \beta_v.$$

However, this contradicts the fact that $j \in \mathcal{U}$. Overall, since we have reached a contradiction, we may conclude that $\mathcal{U}$ is finite. □

4. A Complete Penalty-SQP Algorithm. In the previous section, a dynamic penalty parameter updating strategy is proposed to guarantee that the computed search direction simultaneously offers progress toward reducing the penalty function and reducing infeasibility. In this section, a complete algorithm for solving (NLP) that employs this strategy is proposed and analyzed. It follows the general strategy in Algorithm 1, but includes additional details.

Our complete algorithm involves an additional check of the penalty parameter after the search direction has been computed as is similarly done in various algorithms that employ a penalty function as a merit function. Let $\tilde{\rho}_k$ be the value of the penalty parameter obtained by applying (DUST) within the kth subproblem solve. Then, given a constant $\beta_l \in (0, \beta_\phi(1-\beta_v)]$, we require $\rho_k \in (0, \tilde{\rho}_k]$ so that

$$\Delta l(d^k, \rho_k; x^k) + \omega_k \geq \beta_l(\Delta l(d^k, 0; x^k) + \omega_k), \tag{4.1}$$

where the right-hand side of this inequality is guaranteed to be positive due to (Rv). More precisely, we employ the following Posterior Subproblem STrategy:

$$\text{(PSST)} \qquad \boxed{\rho_k \leftarrow \begin{cases} \tilde{\rho}_k & \text{if this yields (4.1)} \\ \dfrac{(1-\beta_l)(\Delta l(d^k, 0; x^k) + \omega_k)}{\langle \nabla f(x^k), d^k \rangle + \frac{1}{2}\langle d^k, H(\rho_k; x^k, \eta^k) d^k \rangle} & \text{otherwise.} \end{cases}}$$

Observe that if the choice $\rho_k = \tilde{\rho}_k$ does not yield (4.1), then, by setting ρ_k according to the latter formula in (PSST), it follows (since $H(\rho_k; x^k, \eta^k) \succeq 0$) that

$$\rho_k \langle \nabla f(x^k), d^k \rangle \leq (1-\beta_l)(\Delta l(d^k, 0; x^k) + \omega_k),$$

which means that

$$\Delta l(d^k, \rho_k; x^k) + \omega_k = \Delta l(d^k, 0; x^k) - \rho_k \langle \nabla f(x^k), d^k \rangle + \omega_k \geq \beta_l(\Delta l(d^k, 0; x^k) + \omega_k),$$

implying that (4.1) holds. This idea is similar to the updating strategy in [3]. A novel aspect of (PSST), however, is that this model reduction condition is imposed inexactly (due to the presence of $\omega_k > 0$). In fact, for a relatively large ω_k, the model reduction in $l(\cdot, \rho_k; x^k)$ is not necessarily at least a fraction of that in $l(\cdot, 0; x^k)$. This difference makes (PSST) more suitable for an inexact penalty-SQP framework.

Our complete algorithm employing (DUST) and (PSST) is given as Algorithm 2. While we do not complicate the notation by making the dependence explicit on $k \in \mathbb{N}$, it should be clear that in the inner loop (over j) one is solving a subproblem with quantities dependent on the kth iterate; see (3.1). Also, while our analysis does not depend on this choice, we remark that a reasonable choice for η^{k+1} for all $k \in \mathbb{N}$ are the *QP multipliers*, i.e., $\eta^{k+1} = \zeta(\mathbf{u}^{(j)})$, where $\zeta(\mathbf{u})$ is defined prior to (3.2). We do not specify this choice since one might also consider using, e.g., *least squares multipliers* [13]. Our analysis, which focuses on primal convergence, works with any such choice as long as the sequence of dual estimates remains bounded (see below).

Algorithm 2 Penalty-SQP with a Dynamic Penalty Parameter Updating Strategy

Require: $(\gamma, \theta_\rho, \theta_\alpha, \theta_\omega, \beta_v, \beta_\phi) \in (0,1)$, $\beta_l \in (0, \beta_\phi(1-\beta_v))$, and $(\rho_{-1}, \omega_0) \in (0, \infty)$

```
Choose (x^0, η^0) ∈ R^n × R^m.
for k ∈ N do
    Set ρ_(0) ← ρ_{k−1}
    for j ∈ N do
        Generate a primal-dual feasible solution estimate (d^(j), u^(j), w^(j))
        Set ρ_(j+1) by applying (DUST)
    Set d^k ← d^(j) and ρ̃_k ← ρ_(j).
    Set ρ_k by applying (PSST)
    Let α^k be the largest value in {γ^0, γ^1, γ^2, ...} such that
```

$$\phi(x^k + \alpha_k d^k, \rho_k) - \phi(x^k, \rho_k) \le -\theta_\alpha \alpha_k \Delta l(d^k, \rho_k; x^k). \tag{4.2}$$

```
    Choose ω_{k+1} ∈ (0, θ_ω ω_k].
    Set x^{k+1} ← x^k + α_k d^k and choose η ∈ R^m.
```

In the remainder of this section, we show that if (DUST) and (PSST) are employed within a penalty-SQP algorithm for solving (NLP), then, under reasonable assumptions, the algorithm converges from any starting point. Specifically, if (DUST) and (PSST) are only triggered a finite number of times, then every limit point of the iterates is either infeasible stationary or first-order stationary for (NLP). Otherwise, if (DUST) and (PSST) are triggered an infinite number of times, driving the penalty parameter to zero, then every limit point of the iterates is either an infeasible stationary point or a feasible point at which a constraint qualfication fails to hold.

For our analysis in this section, we extend our use of the sub/superscript k to denote the value of quantities associated with iteration $k \in \mathbb{N}$. For example, $\mathcal{U}^k$ denotes the set $\mathcal{U}$ defined in §3.3 while solving the kth subproblem and $\kappa_{0,k}$ is the constant κ_0 in Assumption 2 for the kth subproblem.

We make the following assumption throughout this analysis.

Assumption 7. *There exist a compact convex set $X \subset \mathbb{R}^n$ with $0 \in \text{int}\,(X)$ and positive scalar constants $\underline{\Lambda}_0$, $\overline{\Lambda}_0$, $\underline{\Lambda}_\rho$, $\overline{\Lambda}_\rho$, and K_0 such that the following hold true.*

(i) $X^k = X$ *for all* $k \in \mathbb{N}$.

(ii) f *and* c_i *for all* $i \in \{1, \ldots, m\}$*, and their first- and second-order derivatives, are all bounded in an open convex set containing* $\{x^k\}$ *and* $\{x^k + d^k\}$.

(iii) There exist constants $\overline{\Lambda} \ge \underline{\Lambda} > 0$ *such that for all* $k \in \mathbb{N}$ *and any* $\rho \in [0, \rho_0]$,

$$0 < \underline{\Lambda} \le \underline{\lambda}_{0,k} \le \overline{\lambda}_{0,k} \le \overline{\Lambda} \quad \textit{and} \quad 0 < \underline{\Lambda} \le \underline{\lambda}_{\rho,k} \le \overline{\lambda}_{\rho,k} \le \overline{\Lambda}.$$

(iv) $\kappa_{0,k} \le K_0$ *for all* $k \in \mathbb{N}$.

(v) $\|\nabla c_i(x^k)\|_2 > 0$ *for all* $k \in \mathbb{N}$ *and* $i \in \{1, \ldots, m\}$.

(vi) $\{\eta^k\}$ *is bounded.*

Recalling Lemmas 4 and 5, it follows under Assumption 1, 2, and 7 that there exist positive scalar constants K_1, K_2, and K_3 such that

$$0 < \kappa_{1,k} \le K_1, \quad 0 < \kappa_{2,k} \le K_2, \quad \text{and} \quad 0 < \kappa_{3,k} \le K_3 \quad \text{for all} \quad k \in \mathbb{N}. \tag{4.3}$$

Let us define the index set

$$\mathcal{D} := \{k \in \mathbb{N} : \mathcal{U}^k \ne \emptyset\}.$$

Moreover, for every $k \in \mathcal{D}$, let j_k be the subproblem iteration number corresponding to the value of the smallest ratio r_v, i.e., such that

$$r_v^{(j_k)} \leq r_v^{(i_k)} \quad \text{for any} \quad i_k \in \mathcal{U}^k.$$

Let us also define the index set

$$\mathcal{T} := \{k \in \mathbb{N} : \rho_k \text{ is reduced by (PSST)}\}.$$

It follows from these definitions that $\rho_k < \rho_{k-1}$ if and only if $k \in \mathcal{D} \cup \mathcal{T}$.

Before analyzing the behavior of the iterates of our algorithm, we first provide a couple results related to our subproblem and its solutions. For this result and the remainder of this section, let $d^*(\rho; x, \eta)$ denote a minimizer of $J(d, \rho; x, \eta)$. From [3, Lemma 4.2, 4.3, and 4.4], we have the properties stated in the following lemma.

LEMMA 8. *Under Assumption 7, the following hold at any* (x^k, η^k)*.*
(i) The minimizer of $J(\cdot, \rho; x^k, \eta^k)$ *is unique for any* $\rho \geq 0$*.*
(ii) $\Delta l(d^*(0, x^k, \eta^k); x^k) \geq 0$ *where equality holds if and only if* $d^*(0; x^k, \eta^k) = 0$*.*
(iii) $d^*(0; x^k, \eta^k) = 0$ *if and only if* x^k *is stationary for* v*.*
(iv) If $d^*(\rho; x^k, \eta^k) = 0$ *for* $\rho > 0$ *and* $v(x^k) = 0$*, then* x^k *is stationary for* (NLP)*.*

We also have the following fact about the subproblem solutions.

LEMMA 9. *Under Assumption 7,* $\{d^*(0; x^k, \eta^k)\}$ *and* $\{d^*(\rho_k; x^k, \eta^k)\}$ *are bounded.*

Proof. The proof follows the same line of argument for bounding each primal step in norm as is used in the proof of Lemma 4, where the facts that

$$J(d^*(0; x^k, \eta^k), 0; x^k, \eta^k) \leq J(0, 0; x^k, \eta^k)$$
$$\text{and} \quad J(d^*(\rho_k; x^k, \eta^k), \rho_k; x^k, \eta^k) \leq J(0, 0; x^k, \eta^k)$$

follow from the definitions of $d^*(0; x^k, \eta^k)$ and $d^*(\rho_k; x^k, \eta^k)$. ◻

We now prove a useful lower bound for the stepsize in each iteration.

LEMMA 10. *Under Assumption 7, it follows that, for all* $k \in \mathbb{N}$*, the stepsize satisifies* $\alpha_k \geq C\Delta l(d^k, \rho_k; x^k)$ *for some constant* $C > 0$ *independent of* k*.*

Proof. If $d^k = 0$, then (4.2) holds with $\alpha^k = \gamma^0 = 1$. Hence, for the remainder of the proof, let us assume that $d^k \neq 0$. Under Assumption 7, applying Taylor's theorem and [3, Lemma 4.2], we have that for all positive α that are sufficiently small, there exists $\tau > 0$ such that

$$\phi(x^k + \alpha d^k, \rho_k) - \phi(x^k, \rho_k) \leq -\alpha \Delta l(d^k, \rho_k; x^k) + \tau \alpha^2 \|d^k\|^2.$$

Thus, for any $\alpha \in [0, (1 - \theta_\alpha)\Delta l(d^k, \rho_k; x^k)/(\tau \|d^k\|^2)]$, it follows that

$$-\alpha \Delta l(d^k, \rho_k; x^k) + \tau \alpha^2 \|d^k\|^2 \leq -\alpha \theta_\alpha \Delta l(d^k, \rho_k; x^k),$$

meaning that the sufficient decrease condition (4.2) holds. During the line search, the stepsize is multiplied by γ until (4.2) holds, so we know by the above inequality that the backtracking procedure terminates with

$$\alpha_k \geq \gamma(1 - \theta_\alpha)\Delta l(d^k, \rho_k; x^k)/(\tau \|d^k\|^2).$$

The result follows from this inequality since $\{\|d^k\|\}$ is bounded above by K_1. ◻

Next we show that the reductions in the models of the constraint violation and the penalty function both vanish in the limit. For this purpose, it will be convenient to work with the shifted penalty function

$$\varphi(x,\rho) := \rho(f(x) - \underline{f}) + v(x) \geq 0,$$

where $\underline{f}$ is the infimum of f over the smallest convex set containing $\{x^k\}$. The existance of $\underline{f}$ follows from Assumption 7(ii). The function φ possesses a useful monotonicity property proved in the following lemma.

Lemma 11. *Under Assumption 7, it holds that, for all $k \in \mathbb{N}$,*

$$\varphi(x^{k+1}, \rho_{k+1}) \leq \varphi(x^k, \rho_k) - \theta_\alpha \alpha_k \Delta l(d^k, \rho_k; x^k).$$

Proof. By the line search condition (4.2), it follows that

$$\varphi(x^{k+1}, \rho_k) \leq \varphi(x^k, \rho_k) - \theta_\alpha \alpha_k \Delta l(d^k, \rho_k; x^k),$$

which implies

$$\varphi(x^{k+1}, \rho_{k+1}) \leq \varphi(x^k, \rho_k) - (\rho_k - \rho_{k+1})(f(x^{k+1}) - \underline{f}) - \theta_\alpha \alpha_k \Delta l(d^k, \rho_k; x^k).$$

The result then follows from this inequality, the fact that $\{\rho_k\}$ is monotonically decreasing, and since $f(x^{k+1}) \geq \underline{f}$ for all $k \in \mathbb{N}$. ∎

We now show that the model reductions and duality gap all vanish asymptotically.

Lemma 12. *Under Assumption 7, the following limits hold.*

(i) $0 = \lim_{k\to\infty} \Delta l(d^k, \rho_k; x^k) = \lim_{k\to\infty} \Delta J(d^k, \rho_k; x^k, \eta^k)$,

(ii) $0 = \lim_{k\to\infty} \Delta l(d^k, 0; x^k) = \lim_{k\to\infty} \Delta J(d^k, 0; x^k, \eta^k)$,

(iii) $0 = \lim_{k\to\infty} \Delta J(d^*(0; x^k, \eta^k), 0; x^k, \eta^k) = \lim_{k\to\infty} \Delta J(d^*(\rho_k; x^k, \eta^k), \rho_k; x^k, \eta^k)$,

(iv) $0 = \lim_{k\to\infty} [J(0, \rho_k; x^k, \eta^k) - D(\mathbf{u}^k, \rho_k; x^k, \eta^k)]$,

(v) $0 = \lim_{k\to\infty} [J(0, 0; x^k, \eta^k) - D(\mathbf{w}^k, 0; x^k, \eta^k)]$.

Proof. Let us first prove (i) by contradiction. Suppose that $\Delta l(d^k, \rho_k; x^k)$ does not converge to 0. Then, there exists a constant $\epsilon > 0$ and an infinite $\mathcal{K} \subseteq \mathbb{N}$ such that $\Delta l(d^k, \rho_k; x^k) \geq \epsilon$ for all $k \in \mathcal{K}$. It then follows from Lemma 10 and 11 that $\varphi(x^k; \rho_k) \to -\infty$, which contradicts the fact that $\{\varphi(x^k, \rho_k)\}$ is bounded below by zero. Therefore, $\Delta l(d^k, \rho_k; x^k) \to 0$. The second limit in ($i$) then follows from the first limit, the fact that $H(\rho_k; x^k, \eta^k) \succeq 0$ for all $k \in \mathbb{N}$, and the fact that

$$\begin{aligned} \Delta l(d^k, \rho_k; x^k) &= \Delta J(d^k, \rho_k; x^k, \eta^k) + \tfrac{1}{2}\langle d^k, H(\rho_k; x^k, \eta^k) d^k \rangle \\ &\geq \Delta J(d^k, \rho_k; x^k, \eta^k). \end{aligned} \tag{4.4}$$

Next, from (4.1) and (4.4), it follows that

$$\Delta l(d^k, \rho_k; x^k) + \omega_k \geq \beta_l(\Delta l(d^k, 0; x^k) + \omega_k) \geq \beta_l(\Delta J(d^k, 0; x^k, \eta^k) + \omega_k).$$

The limits in (ii) follow from these inequalities, the first limit in (i), and the fact that $\{\omega_k\} \to 0$. Finally, the limits in (iii), (iv), and (v) follow from the limits in parts (i) and (ii) along with the inequalities in (3.7) and (3.8). ∎

We now show that the primal steps and the exact subproblem solutions vanish.

LEMMA 13. *Suppose Assumption 7 holds and $\{\rho_k\} \to \rho_*$. Then, $\{d^k\} \to 0$ and for any limit point x^* of $\{x^k\}$ it follows that $d^*(0; x^*, \cdot) = 0$ and $d^*(\rho_*; x^*, \cdot) = 0$.*

Proof. From Lemma 12(*ii*), it follows that

$$0 = \lim_{k\to\infty} -\Delta J(d^k, 0; x^k, \eta^k) = \lim_{k\to\infty} -\Delta l(d^k, 0; x^k) + \tfrac{1}{2}\langle d^k, H(0; x^k, \eta^k) d^k\rangle$$
$$= \lim_{k\to\infty} \tfrac{1}{2}\langle d^k, H(0; x^k, \eta^k) d^k\rangle \geq \lim_{k\to\infty} \tfrac{1}{2}\underline{\Delta}\|d^k\|^2.$$

This implies that $\{d^k\} \to 0$, as desired. Next, from Lemma 12(*iii*) and continuity, it follows that $\Delta J(d^*(0; x^*, \cdot), 0; x^*, \cdot) = 0$, from which it follows that

$$J(d^*(0; x^*, \cdot), 0; x^*, \cdot) = J(0, 0; x^*, \cdot).$$

From the strong convexity of $J(\cdot, 0; x^*, \cdot)$ and the fact that $d^*(0; x^*, \cdot)$ is its minimizer, it follows that $d^*(0; x^*, \cdot) = 0$. Using a similar argument and Lemma 12(*iii*) again, it follows that $d^*(\rho_*; x^*, \cdot) = 0$, completing the proof. ∎

Our first global convergence theorem follows.

THEOREM 14. *Under Assumption 7, the following statements hold.*
- *(i) Any limit point of $\{x^k\}$ is first-order stationary for v, i.e., it is feasible or an infeasible stationary point for (NLP).*
- *(ii) If $\rho_k \to \rho_*$ for some $\rho_* > 0$ and $v(x^k) \to 0$, then any limit point x^* of $\{x^k\}$ with $v(x^*) = 0$ is a KKT point for (NLP).*
- *(iii) If $\rho_k \to 0$, then either all limit points of $\{x^k\}$ are feasible for (NLP) or all are infeasible.*

Proof. Part (*i*) follows by combining Lemma 13 with Lemma 8(*iii*). Similarly, part (*ii*) follows by combining Lemma 13 with Lemma 8(*iv*).

We prove (*iii*) by contradiction. Suppose there exist infinite $\mathcal{K}^* \subseteq \mathbb{N}$ and $\mathcal{K}^\times \subseteq \mathbb{N}$ such that $\{x^k\}_{k\in\mathcal{K}^*} \to x^*$ with $v(x^*) = 0$ and $\{x^k\}_{k\in\mathcal{K}^\times} \to x^\times$ with $v(x^\times) = \epsilon > 0$. Since $\rho_k \to 0$, there exists $k^* \geq 0$ such that for all $k \in \mathcal{K}^*$ and $k \geq k^*$ one has that $\rho_{k+1}(f(x^k) - \underline{f}) < \epsilon/4$ and $v(x^k) < \epsilon/4$, meaning that $\varphi(x^k, \rho_{k+1}) < \epsilon/2$. On the other hand, it follows that $\rho_{k+1}(f(x^k) - \underline{f}) \geq 0$ for all $k \in \mathbb{N}$ and there exists $k^\times \in \mathbb{N}$ such that $v(x^k) \geq \epsilon/2$ for all $k \geq k^\times$ with $k \in \mathcal{K}^\times$, meaning that $\varphi(x^k, \rho_{k+1}) \geq \epsilon/2$. This contradicts Lemma 11, which shows that $\varphi(x^k, \rho_{k+1})$ is monotonically decreasing. Thus, the set of limit points of $\{x^k\}$ must be all feasible or all infeasible. ∎

Theorem 14 is satisfactory in the case when $\rho_k \to \rho_* > 0$, since it shows that any limit point of the primal sequence is a KKT point for (NLP). But more needs to be said when $\rho_k \to 0$. We now address this case, showing that it only occurs if a limit point of the algorithm is either an infeasible stationary point or a feasible point at which a constraint qualification fails to hold. We begin with the following lemma.

LEMMA 15. *Suppose Assumption 7 holds and $\rho_k \to 0$. Let x^* be a limit point of $\{x^k\}_{k\in\mathcal{D}\cup\mathcal{T}}$ that is feasible for (NLP) with infinite $\mathcal{S} \subseteq \mathcal{D}\cup\mathcal{T}$ such that $\{x^k\}_{k\in\mathcal{S}} \to x^*$. Then, the following hold true.*
- *(i) $|\mathcal{S}\cap\mathcal{D}|$ is finite or $\{\Delta J(d^{(j_k)}, \rho_{(j_k)}; x^k, \eta^k)\}_{k\in\mathcal{S}\cap\mathcal{D}} \to 0$;*
- *(ii) $|\mathcal{S}\cap\mathcal{D}|$ is finite or $\{d^{(j_k)}\}_{k\in\mathcal{S}\cap\mathcal{D}} \to 0$;*
- *(iii) any limit point of $\{\mathbf{u}^{(j_k)}\}_{k\in\mathcal{S}\cap\mathcal{D}} \cup \{\mathbf{u}^k\}_{k\in\mathcal{S}\cap\mathcal{T}}$ is optimal for $D(\cdot, 0; x^*, \cdot)$;*
- *(iv) $\{\mathbf{u}^{(j_k)}\}_{k\in\mathcal{S}\cap\mathcal{D}} \cup \{\mathbf{u}^k\}_{k\in\mathcal{S}\cap\mathcal{T}}$ has a nonzero limit point.*

Proof. For part (i), if $|\mathcal{S}\cap\mathcal{D}|$ is finite, then there is nothing left to prove. Hence, let us assume that $|\mathcal{S}\cap\mathcal{D}|=\infty$. Observe that, for all $k\in\mathbb{N}$, it holds that

$$\begin{aligned}
0 &\leq \Delta J(d^{(j_k)},\rho_{(j_k)};x^k,\eta^k)\\
&= v(x^k)-\rho_{(j_k)}\langle\nabla f(x^k),d^{(j_k)}\rangle-\tfrac{\rho_{(j_k)}}{2}\langle d^{(j_k)},H_f(x^k)d^{(j_k)}\rangle-J(d^{(j_k)},0;x^k,\eta^k)\\
&\leq v(x^k)-\rho_{(j_k)}\langle\nabla f(x^k),d^{(j_k)}\rangle-\tfrac{\rho_{(j_k)}}{2}\langle d^{(j_k)},H_f(x^k)d^{(j_k)}\rangle,
\end{aligned}$$

where the first inequality follows from (3.4) and the second inequality follows from the definition of J, which ensures that $J(d^{(j_k)},0;x^k,\eta^k)\geq 0$. In addition, $\{d^{(j_k)}\}$ is bounded due to Lemma 4 and Assumption 7(ii)-(iii). Consequently, since $|\mathcal{S}\cap\mathcal{D}|=\infty$ and $\{v(x^k)\}_{k\in\mathcal{S}\cap\mathcal{D}}\to 0$ with $\rho^{(j_k)}\to 0$, the limit in part (i) holds.

For part (ii), again, if $|\mathcal{S}\cap\mathcal{D}|$ is finite, then there is nothing left to prove. Otherwise, since $\{J(0,0;x^k,\eta^k)\}_{k\in\mathcal{S}\cap\mathcal{D}}=\{v(x^k)\}_{\mathcal{S}\cap\mathcal{D}}\to 0$ and $\rho^{(j_k)}\to 0$, the limit in part (ii) holds due to Lemma 4 and Assumption 7(ii)-(iii).

Now consider part (iii). If $|\mathcal{S}\cap\mathcal{D}|$ is infinite, then for a limit point $\mathbf{u}^*$ there must exist an infinite $\mathcal{S}_{\mathcal{D}}\subseteq\mathcal{S}\cap\mathcal{D}$ such that $\{\mathbf{u}^{(j_k)}\}_{k\in\mathcal{S}_D}\to\mathbf{u}^*$. Then, it follows that

$$\begin{aligned}
0 &\leq J(0,0;x^*,\cdot)-D(\mathbf{u}^*,0;x^*,\cdot)\\
&= \lim_{\substack{k\in\mathcal{S}_{\mathcal{D}}\\k\to\infty}} J(0,\rho_{(j_k)};x^k,\cdot)-D(\mathbf{u}^{(j_k)},\rho_{(j_k)};x^k,\cdot)\\
&\leq \lim_{\substack{k\in\mathcal{S}_{\mathcal{D}}\\k\to\infty}} \beta_\phi[J(0,\rho_{(j_k)};x^k,\cdot)-J(d^{(j_k)},\rho_{(j_k)};x^k,\cdot)]\\
&= \lim_{\substack{k\in\mathcal{S}_{\mathcal{D}}\\k\to\infty}} \beta_\phi[J(0,0;x^k,\cdot)-J(d^{(j_k)},0;x^k,\cdot)]\leq \lim_{\substack{k\in\mathcal{S}_{\mathcal{D}}\\k\to\infty}} \beta_\phi J(0,0;x^k,\cdot)=0,
\end{aligned}\tag{4.5}$$

where the first inequality is by (Rphi) and the second inequality is by the fact that $J(d^{(j_k)},0;x^k,\cdot)\geq 0$. This means that $\mathbf{u}^*$ is optimal for $D(\cdot,0;x^*,\cdot)$. On the other hand, if $|\mathcal{S}\cap\mathcal{D}|$ is finite, then $|\mathcal{S}\cap\mathcal{T}|$ must be infinite, in which case for a limit point $\mathbf{u}^*$ there must exist an infinite $\mathcal{S}_{\mathcal{T}}\subseteq\mathcal{S}\cap\mathcal{T}$ such that $\{\mathbf{u}^k\}_{k\in\mathcal{S}_{\mathcal{T}}}\to\mathbf{u}^*$. Then, again from Lemma 12 and (4.5), it follows that $\mathbf{u}^*$ is optimal for $D(\cdot,0;x^*,\cdot)$.

For part (iv), first observe that

$$l(d,0;x^k)=\sum_{i\in\mathcal{E}_+(d)\cup\mathcal{E}_-(d)\cup\mathcal{I}_+(d)}\|\nabla c_i(x^k)\|_2\text{dist}\left(d\;\middle|\;C_i^k\right),$$

and that $\chi(d,\mathbf{u};x^k)$ can be viewed as a weighted variant of this sum with weights

$$1-\zeta_i(\mathbf{u})\ \text{ for all }\ i\in\mathcal{E}_+(d)\cup\mathcal{I}_+(d)\ \text{ and }\ 1+\zeta_i(\mathbf{u})\ \text{ for all }\ i\in\mathcal{E}_-(d).$$

Also observe that (Rc) holds at any primal-dual point

$$(d,\mathbf{u})\in\{(d^{(j_k)},\mathbf{u}^{(j_k)})\}_{k\in\mathcal{S}\cap\mathcal{D}}\cup\{(d^k,\mathbf{u}^k)\}_{k\in\mathcal{S}\cap\mathcal{T}}$$

due to the facts that

$$\chi(d^{(j_k)},\mathbf{u}^{(j_k)};x^k)\leq(1-\beta_v)^2(v(x^k)+\omega_k)\ \text{ for all }\ k\in\mathcal{S}\cap\mathcal{D}\ \text{ and}\tag{4.6}$$

$$\chi(d^k,\mathbf{u}^k;x^k)\leq(1-\beta_v)^2(v(x^k)+\omega_k)\ \text{ for all }\ k\in\mathcal{S}\cap\mathcal{T}.\tag{4.7}$$

We now consider three cases.

Case (a): Assume there exists an infinite $\mathcal{S}_{\mathcal{D}} \subseteq \mathcal{S} \cap \mathcal{D}$ such that

$$l(d^{(j_k)}, 0; x^k) > (1 - \beta_v)(v(x^k) + \omega_k) \quad \text{for all} \quad k \in \mathcal{S}_D. \tag{4.8}$$

Then, $\|\zeta(\mathbf{u}^{(j_k)})\|_\infty \geq \beta_v$ for all $k \in \mathcal{S}_{\mathcal{D}}$; indeed, if this were not the case, then for some $k \in \mathcal{S}_{\mathcal{D}}$ one would find from the definition of χ and (4.8) that

$$\chi(d^{(j_k)}, \mathbf{u}^{(j_k)}; x^k) \geq (1 - \beta_v) l(d^{(j_k)}, 0; x^k) > (1 - \beta_v)^2 (v(x^k) + \omega_k),$$

contradicting (4.6). In this case, combining Lemma 3, Assumption 7(iv), and the fact that $\|\zeta(\mathbf{u}^{(j_k)})\|_\infty \geq \beta_v$ for all $k \in \mathcal{S}_{\mathcal{D}}$ shows that $\{\mathbf{u}^{(j_k)}\}_{k \in \mathcal{S} \cap \mathcal{D}}$ has a nonzero limit point, proving part (iv), as desired.

Case (b): Assume there exists an infinite $\mathcal{S}_{\mathcal{T}} \subseteq \mathcal{S} \cap \mathcal{T}$ such that

$$l(d^k, 0; x^k) > (1 - \beta_v)(v(x^k) + \omega_k) \quad \text{for all} \quad k \in \mathcal{S}_{\mathcal{T}}. \tag{4.9}$$

Then, $\|\zeta(\mathbf{u}^k)\|_\infty \geq \beta_v$ for all $k \in \mathcal{S}_{\mathcal{T}}$; indeed, if this were not the case, then for some $k \in \mathcal{S}_{\mathcal{T}}$ one would find from the definition of χ and (4.8) that

$$\chi(d^k, \mathbf{u}^k; x^k) \geq (1 - \beta_v) l(d^k, 0; x^k) > (1 - \beta_v)^2 (v(x^k) + \omega_k),$$

contradicting (4.7). In this case, combining Lemma 3, Assumption 7(iv), and the fact that $\|\zeta(\mathbf{u}^k)\|_\infty \geq \beta_v$ for all $k \in \mathcal{S}_{\mathcal{T}}$ shows that $\{\mathbf{u}^k\}_{k \in \mathcal{S} \cap \mathcal{T}}$ has a nonzero limit point, proving part (iv), as desired.

Case (c): Suppose that (4.8) and (4.9) only hold for finite subsets of $\mathcal{S} \cap \mathcal{D}$ and $\mathcal{S} \cap \mathcal{T}$. In this case, there exists a sufficiently large $\bar{k} \in \mathbb{N}$ such that

$$l(d^{(j_k)}, 0; x^k) \leq (1 - \beta_v)(v(x^k) + \omega_k) \text{ for all } k \in \mathcal{S} \cap \mathcal{D} \text{ with } k \geq \bar{k}; \tag{4.10}$$

$$l(d^k, 0; x^k) \leq (1 - \beta_v)(v(x^k) + \omega_k) \text{ for all } k \in \mathcal{S} \cap \mathcal{T} \text{ with } k \geq \bar{k}. \tag{4.11}$$

We can further assume that

$$\begin{aligned} \|\zeta(\mathbf{u}^{(j_k)})\|_\infty < \beta_v &\text{ for all } k \in \mathcal{S} \cap \mathcal{D} \text{ with } k \geq \bar{k} \text{ and} \\ \|\zeta(\mathbf{u}^k)\|_\infty < \beta_v &\text{ for all } k \in \mathcal{S} \cap \mathcal{T} \text{ with } k \geq \bar{k}; \end{aligned}$$

since otherwise, as in Cases (a) and (b), respectively, part (iv) would hold. Now, for $k \geq \bar{k}$ with $k \in \mathcal{S} \cap \mathcal{D}$, it follows from (4.10) that

$$\begin{aligned} & J(0, 0; x^k, \eta^k) + \omega_k - l(d^{(j_k)}, 0; x^k) \\ \geq\ & v(x^k) + \omega_k - (1 - \beta_v)(v(x^k) + \omega_k) \\ =\ & \beta_v (v(x^k) + \omega_k) \\ \geq\ & \beta_v [v(x^k) + \omega_k - (D(\mathbf{w}^{(j_k)}, 0; x^k, \eta^k))_+], \end{aligned}$$

from which it follows that

$$r_v^{(j_k)} = \frac{J(0, 0; x^k, \eta^k) + \omega_k - l(d^{(j_k)}, 0; x^k)}{v(x^k) + \omega_k - (D(\mathbf{w}^{(j_k)}, 0; x^k, \eta^k))_+} \geq \beta_v.$$

This indicates that (DUST) is not triggered at any iteration $k \geq \bar{k}$ with $k \in \mathcal{S} \cap \mathcal{D}$. By the definition of $\mathcal{D}$, this implies that $\mathcal{S} \cap \mathcal{D}$ is finite. On the other hand, for $k \in \mathcal{S} \cap \mathcal{T}$ with $k \geq \bar{k}$, it holds that

$$
\begin{aligned}
& J(0,0;x^k,\eta^k) - D(\mathbf{u}^k,\rho_k;x^k,\eta^k) \\
& \geq v(x^k) + \sum_{i=1}^{m} \|\nabla c_i(x^k)\|_2 \delta^*(u_i^k|C_i^k) \\
& = \sum_{i=1}^{\bar{m}} |c_i(x^k)| + \sum_{i=\bar{m}+1}^{m} (c_i(x^k))_+ - \sum_{i=1}^{m} \|\nabla c_i(x^k)\|_2 \zeta_k^i \frac{c_i(x^k)}{\|\nabla c_i(x^k)\|_2} \\
& = \sum_{i=1}^{\bar{m}} |c_i(x^k)| + \sum_{i=\bar{m}+1}^{m} (c_i(x^k))_+ - \sum_{i=1}^{m} \zeta_k^i c_i(x^k) \\
& = \sum_{i=1}^{\bar{m}} [|c_i(x^k)| - \zeta_k^i c_i(x^k)] + \sum_{i=\bar{m}+1}^{m} [(c_i(x^k))_+ - \zeta_k^i c_i(x^k)] \\
& \geq \sum_{i=1}^{\bar{m}} (1-|\zeta_k^i|)|c_i(x^k)| + \sum_{i=\bar{m}+1}^{m} (1-|\zeta_k^i|)(c_i(x^k))_+ \\
& \geq (1-\beta_v) \sum_{i=1}^{\bar{m}} |c_i(x^k)| + (1-\beta_v) \sum_{i=\bar{m}+1}^{m} (c_i(x^k))_+ = (1-\beta_v) v(x^k),
\end{aligned}
\tag{4.12}
$$

where the first inequality is from the positive definiteness of $H(0,x^k,\eta^k)$ and $\delta^*(u_{m+1}^k|X) = \sup_{d\in X}\langle u_{m+1}^k, d\rangle \geq 0$, and the first equality is from (3.2). Since (Rphi) is satisfied, the first inequality in (3.8) and (4.12) imply

$$
\begin{aligned}
& \Delta J(d^k,\rho_k;x^k,\eta^k) + \omega_k = J(0,0;x^k,\eta^k) - J(d^k,\rho_k;x^k,\eta^k) + \omega_k \\
& \geq \beta_\phi[J(0,0;x^k,\eta^k) - D(\mathbf{u}^k,\rho_k;x^k,\eta^k) + \omega_k] \\
& \geq \beta_\phi[(1-\beta_v)v(x^k) + \omega_k] \geq \beta_\phi(1-\beta_v)(v(x^k)+\omega_k) \\
& \geq \beta_l(v(x^k)+\omega_k) \geq \beta_l(\Delta l(d^k,0;x^k)+\omega_k).
\end{aligned}
$$

which, together with (4.4), yields

$$
\Delta l(d^k,\rho^k;x^k) + \omega_k \geq \Delta J(d^k,\rho_k;x^k,\eta^k) + \omega_k \geq \beta_l(\Delta l(d^k,0;x^k)+\omega_k).
$$

Therefore, (PSST) is not triggered in any iteration $k \in \mathcal{S}\cap\mathcal{T}$ with $k \geq \bar{k}$. By the definition of $\mathcal{T}$, this means that $\mathcal{S}\cap\mathcal{T}$ is finite. Overall, we have shown in this case that $\mathcal{S}\cap\mathcal{D}$ and $\mathcal{S}\cap\mathcal{T}$ are finite, meaning $\mathcal{S}$ is finite. However, this contradicts the statement of the lemma, which defines $\mathcal{S}$ to be finite.

Overall, since Case (c) leads to a contradiction, it follows that either Case (a) or (b) must occur, which proves part (iv). ∎

We are now prepared to prove a theorem about the behavior of the algorithm when the penalty parameter is driven to zero. The theorem involves a statement about points satisfying the well-known Mangasarian-Fromovitz constraint qualificatioin (MFCQ). Defining $\mathcal{E} = \{1,\ldots,\bar{m}\}$, $\mathcal{I} = \{\overline{m}+1,\ldots,m\}$,

$$
\begin{aligned}
& \mathcal{A}(x) = \{i \in \{\bar{m}+1,\ldots,m\} : c_i(x) = 0\}, \\
\text{and} \quad & \mathcal{N}(x) = \{i \in \{\bar{m}+1,\ldots,m\} : c_i(x) < 0\},
\end{aligned}
$$

we now recall this qualification then state and prove our theorem.

Definition 16. *A point x satisfies the MFCQ for problem (NLP) if $v(x) = 0$, $\{\nabla c_i(x) : i \in \mathcal{E}\}$ are linearly independent, and there exists $d \in \mathbb{R}^n$ such that*

$$\begin{aligned} c_i(x) + \langle \nabla c_i(x), d\rangle = 0 \quad &\textit{for all} \quad i \in \mathcal{E} \\ \textit{and} \quad c_i(x) + \langle \nabla c_i(x), d\rangle < 0 \quad &\textit{for all} \quad i \in \mathcal{I}, \end{aligned}$$

or, equivalently,

$$\langle \nabla c_i(x), d\rangle = 0 \quad \textit{for all} \quad i \in \mathcal{E} \quad \textit{and} \quad \langle \nabla c_i(x), d\rangle < 0 \quad \textit{for all} \quad i \in \mathcal{A}(x).$$

Theorem 17. *Suppose Assumption 7 holds and $\rho_k \to 0$. Then, every limit point of $\{x^k\}_{k\in\mathcal{D}\cup\mathcal{T}}$ is either an infeasible stationary point or a feasible point where the MFCQ does not hold.*

Proof. By Theorem 14(i), any limit point of $\{x^k\}_{k\in\mathcal{D}\cup\mathcal{T}}$ is either feasible or an infeasible stationary point. If any such point is infeasible, then there is nothing left to prove. We may thus proceed by letting x^* represent a feasible limit point of $\{x^k\}_{k\in\mathcal{D}\cup\mathcal{T}}$. Our goal is to show that the MFCQ fails to hold at x^*.

Let $\mathcal{S} \subseteq \mathcal{D}\cup\mathcal{T}$ be an infinite set such that $\{x^k\}_{k\in\mathcal{S}} \to x^*$. By Theorem 15($iv$), it follows that there exists a nonzero limit point of $\{\mathbf{u}^{(j_k)}\}_{k\in\mathcal{S}\cap\mathcal{D}} \cup \{\mathbf{u}^k\}_{k\in\mathcal{S}\cap\mathcal{T}}$. In addition, from Lemma 13, it follows that $(d, \mathbf{u}) = (0, \mathbf{u}^*)$ is stationary for the feasibility subproblem at x^*. Therefore, it follows from (3.2) and the fact under Assumption 7(i) that $d = 0$ lies in the interior of X that $u^*_{m+1} = 0$ and

$$u_i^* = \begin{cases} \zeta_*^i \frac{\nabla c_i(x^*)}{\|\nabla c_i(x^*)\|_2} & \text{with} \quad \zeta_*^i \in [-1,1] \quad \text{for all } i \in \mathcal{E} \\ \zeta_*^i \frac{\nabla c_i(x^*)}{\|\nabla c_i(x^*)\|_2} & \text{with} \quad \zeta_*^i \in [0,1] \quad \text{for all } i \in \mathcal{I} \end{cases}$$

$$\text{meaning that} \quad \delta^*(u_i^*|C_i^*) = -\zeta_*^i \frac{c_i(x^*)}{\|\nabla c_i(x^*)\|_2} \quad \text{for all } i \in \mathcal{E}\cup\mathcal{I}.$$

It follows that

$$\begin{aligned} 0 = v(x^*) &= J(0,0;x^*,\cdot) = D(\mathbf{u}^*, 0; x^*, \cdot) \\ &= -\tfrac{1}{2}\langle u_0^*, H(0;x^*,\cdot)^{-1}u_0^*\rangle - \sum_{i\in\mathcal{E}\cup\mathcal{I}} \|\nabla c_i(x^*)\|_2 \delta^*(u_i^*|C_i) - \delta^*(u^*_{m+1}|X) \\ &= -\tfrac{1}{2}\langle u_0^*, H(0;x^*,\cdot)^{-1}u_0^*\rangle + \sum_{i\in\mathcal{E}\cup\mathcal{I}} \zeta_*^i c_i(x^*) \\ &= -\tfrac{1}{2}\langle u_0^*, H(0;x^*,\cdot)^{-1}u_0^*\rangle + \sum_{i\in\mathcal{N}(x^*)} \zeta_*^i c_i(x^*). \end{aligned}$$

Since $H(0;x^*,\cdot)$ is positive definite and $\sum_{i\in\mathcal{N}(x^*)} \zeta_*^i c_i(x^*) \le 0$, it follows that

$$\tfrac{1}{2}\langle u_0^*, H(0;x^*,\cdot)^{-1}u_0^*\rangle = 0 \quad \text{and} \quad \sum_{i\in\mathcal{N}(x^*)} \zeta_*^i c_i(x^*) = 0,$$

yielding $u_0^* = 0$ and $\zeta_*^i = 0$ for all $i \in \mathcal{N}(x^*)$. Overall, we have shown that the constraints of (DQPrho) imply that

$$\sum_{i\in\mathcal{E}\cup\mathcal{A}(x^*)} \zeta_*^i \nabla c_i(x^*) = 0. \tag{4.13}$$

□

(4.13) implies that x^* violates the dual form of the MFCQ [16]. Since we have reached a contradiction, it follows that the MFCQ cannot hold at x^*, as desired.

We summarize the results of all of our theorems in the following corollary.

COROLLARY 18. *Suppose Assumption 7 holds. Then, one of the following occurs.*

(i) $\rho_k \to \rho_*$ *for some constant* $\rho_* > 0$ *and each limit point of* $\{x^k\}$ *either corresponds to a KKT point or an infeasible stationary point for problem (NLP).*

(ii) $\rho_k \to 0$ *and all limit points of* $\{x^k\}$ *are infeasible stationary points for (NLP).*

(iii) $\rho_k \to 0$*, all limit points of* $\{x^k\}$ *are feasible for (NLP), and the MFCQ fails to hold at all limit points of* $\{x^k\}_{k\in\mathcal{D}\cup\mathcal{T}}$.

5. Implementation.

In this section, we discuss techniques that can be used for implementing our method. In §5.1, we describe details about how low-rank Hessian approximations could be updated. In §5.2, we introduce a coordinate descent method as an example subproblem solver that could be used with our method.

5.1. Low-rank approximation.

In large-scale settings, it is often intractable to compute and store exact Hessians. Instead, low-rank approximations of the Hessian could be used, e.g., based on L-BFGS [12]. In this section, we describe how to update the low-rank Hessian approximation and its inverse when ρ is updated by (DUST).

Assume the Hessian approximations have the form

$$H_\rho = \sigma I + \Psi\Sigma^{-1}\Psi^T \quad \text{and} \quad H_0 = \gamma I + \Phi\Gamma^{-1}\Phi^T,$$

where $\Psi \in \mathbb{R}^{n\times r}$ with $r \ll n$ and $\Phi \in \mathbb{R}^{n\times l}$ with $l \ll n$ are low rank matrices, and $\Sigma \in \mathbb{R}^{r\times r}$ and $\Gamma \in \mathbb{R}^{l\times l}$ are invertible. We investigate the inverse of H_ρ by using the following generalized matrix inversion formula. For any given invertible $A \in \mathbb{R}^{n\times n}$, invertible $S \in \mathbb{R}^{l\times l}$, and $U, V \in \mathbb{R}^{n\times l}$, the Sherman-Morrison formula yields

$$(A + USV^T)^{-k} = A^{-k} - A^{-k}U(S^{-1} + V^TA^{-k}U)^{-1}WV^TA^{-k}\,, \tag{5.1}$$

where

$$W = \sum_{j=0}^{k-1}(S^{-1}(S^{-1} + V^TA^{-k}U)^{-1})^{(j)}\,.$$

In particular,

$$(A + USV^T)^{-1} = A^{-1} - A^{-1}U(S^{-1} + V^TA^{-1}U)^{-1}V^TA^{-1}. \tag{5.2}$$

Using (5.2), the inverses of H_0 and H_ρ are given by

$$H_0^{-1} = \frac{1}{\gamma}\left[I - \Phi(\gamma\Gamma + \Phi^T\Phi)^{-1}\Phi^T\right] \quad \text{and} \quad H_\rho^{-1} = \frac{1}{\sigma}\left[I - \Psi(\sigma\Sigma + \Psi^T\Psi)^{-1}\Psi^T\right].$$

These can be rewritten in a compact form. Defining

$$\Theta^T = (\gamma\Gamma + \Phi^T\Phi)^{-1}\Phi^T, \quad \Theta_1^T = (\sigma\Sigma + \Psi^T\Psi)^{-1}\Psi^T,$$
$$\text{and} \quad \Theta_2^T = \left[\left(\sigma + \frac{m}{\mu}\right)\Sigma + \Psi^T\Psi\right]^{-1}\Psi^T,$$

it follows that

$$H_0^{-1} = \frac{1}{\gamma}(I - \Phi\Theta^T) \quad \text{and} \quad H_\rho^{-1} = \frac{1}{\sigma}\left[I - \Psi\Theta_1^T\right]. \tag{5.3}$$

After reducing ρ to a smaller value $\bar\rho < \rho$, one finds that

$$H_{\bar\rho} = \bar\rho H_f + H_0 = \frac{\bar\rho}{\rho}(H_0 + \rho H_f) + (1 - \frac{\bar\rho}{\rho})H_0$$

$$
\begin{aligned}
&= \tau H_\rho + (1-\tau) H_0 \\
&= \bar{\sigma} I + \tau \Psi \Sigma^{-1} \Psi^T + (1-\tau)\Phi\Gamma^{-1}\Phi^T \\
&= H_\tau + (1-\tau)\Phi\Gamma^{-1}\Phi^T,
\end{aligned}
$$

with

$$\tau = \frac{\bar{\rho}}{\rho}, \quad \bar{\sigma} = \tau\sigma + (1-\tau)\gamma, \quad \text{and} \quad H_\tau = \bar{\sigma} I + \tau \Psi \Sigma^{-1} \Psi^T.$$

Therefore, we have

$$
\begin{aligned}
& H_\tau^{-1} = \frac{1}{\bar{\sigma}}[I - \Psi\Theta_3^T] && \text{with} \quad \Theta_3^T = (\frac{\bar{\sigma}}{\tau}\Sigma + \Psi^T\Psi)^{-1}\Psi^T, \\
\text{and} \quad & H_{\bar{\rho}}^{-1} = H_\tau^{-1} - H_\tau^{-1}\Phi\Theta_4^T H_\tau^{-1} && \text{with} \quad \Theta_4^T = \left[\frac{1}{1-\tau}\Gamma + \Phi^T H_\tau^{-1}\Phi\right]^{-1}\Phi^T.
\end{aligned}
$$

5.2. Subproblem Solver. As an example of a subproblem solver that can be used within our approach, we present a coordinate descent algorithm to solve (QPrho). For simplicity, let us assume that $X = \mathbb{R}^n$. We have the following two subproblems:

$$\min_{x\in\mathbb{R}^n} J(x;\rho) := \frac{1}{2}x^T H_\rho x + \rho g^T x + \sum_{i=1}^{\overline{m}} |a_i^T x + b_i| + \sum_{i=\overline{m}+1}^{m} (a_i^T x + b_i)_+ \tag{5.4}$$

$$\text{and} \quad \min_{z\in\mathbb{R}^n} J(z;0) := \frac{1}{2}z^T H_0 z + \sum_{i=1}^{\overline{m}} |a_i^T z + b_i| + \sum_{i=\overline{m}+1}^{m} (a_i^T z + b_i)_+. \tag{5.5}$$

Lagrangian duals of (5.4) and (5.5) are, respectively,

$$\max_{l\le\eta\le c} D(\eta;\rho) := -\frac{1}{2}(A^T\eta - \rho g)^T H_\rho^{-1}(A^T\eta - \rho g) + \eta^T b \tag{5.6}$$

$$\text{and} \quad \max_{l\le\lambda\le c} D(\lambda;0) := -\frac{1}{2}\lambda^T A H_0^{-1} A^T \lambda + \lambda^T b \tag{5.7}$$

where $l = [-\mathbf{1}_{\overline{m}}^T, \mathbf{0}_{m-\overline{m}}^T]^T$ and $c = \mathbf{1}_m$. The solutions of (5.4) and (5.5) can be recovered by those of (5.6) and (5.7) as $x = -H_\rho^{-1}(\rho g + A^T\eta)$ and $z = -H_0^{-1}A^T\lambda$, respectively. If we solve the feasibility dual problem (5.7), this will give us a better estimate of r_v at the extra cost of solving for λ. If this cost becomes prohibitive, we can use η instead of λ in the calculation of r_v. This might lead to more iterations for the subproblem solver. Algorithm 3 shows one iteration update of a coordinate descent algorithm. Note that subproblems (5.8) and (5.9) minimize one dimensional quadratics over a box constraint; hence, these have closed form solutions.

We will now discuss how to make one sweep over all coordinates in an efficient manner when we use low-rank Hessian approximations. Since (5.8) and (5.9) have similar structure, we will use (5.9) to demonstrate the implementation details.

Using (5.3), subproblem (5.7) can be written as

$$\max_{l\le\lambda\le c} D(\lambda;0) := -\frac{1}{2\gamma}\lambda^T AA^T\lambda + \frac{1}{2\gamma}\lambda^T A\Phi\Theta^T A^T\lambda + \lambda^T b. \tag{5.10}$$

In large scale settings, it is not practical to calculate and store AA^T. Usually, A will have a nice sparse structure, while AA^T does not. Defining $Q := A\Phi$ and $\tilde{Q} := A\Theta$, subproblem (5.10) becomes

$$\max_{l\le\lambda\le c} D(\lambda;0) := -\frac{1}{2\gamma}\lambda^T AA^T\lambda + \frac{1}{2\gamma}\lambda^T Q\tilde{Q}^T\lambda + \lambda^T b. \tag{5.11}$$

Algorithm 3 Coordinate Descent Algorithm

1: **for** $i = 1, \ldots, m$ **do**

2: Set

$$\eta_i^k := \underset{l_i \le \eta_i \le c_i}{\operatorname{argmin}} D(\eta_1^k, \ldots, \eta_{i-1}^k, \eta_i, \eta_{i+1}^{k-1}, \ldots, \eta_m^{k-1}; \rho^{k-1}) \tag{5.8}$$

$$\text{and} \quad \lambda_i^k := \underset{l_i \le \lambda_i \le c_i}{\operatorname{argmin}} D(\lambda_1^k, \ldots, \lambda_{i-1}^k, \lambda_i, \lambda_{i+1}^{k-1}, \ldots, \lambda_m^{k-1}; 0). \tag{5.9}$$

3: Update $x^k := -H_\rho^{-1}(\rho g + A^T \eta^k)$.

4: Set ρ_k by applying (DUST).

The partial derivative of $D(\lambda; 0)$ with respect to λ_i is given by

$$\frac{\partial D(\lambda; 0)}{\partial \lambda_i} := \frac{1}{\gamma} \sum_{j=1}^m (-a_i^T a_j + q_i^T \tilde{q}_j^T)\lambda_j + b_i, \tag{5.12}$$

where q_i and $\tilde{q}_i$ are the i-th row of Q and $\tilde{Q}$ respectively. Then, (5.9) becomes

$$\lambda_i^k = \begin{cases} l_i & \text{if } a_i^T a_i - q_i^T \tilde{q}_i^T = 0 \text{ and } \frac{\partial D(\lambda;0)}{\partial \lambda_i} < 0 \\ [l_i, c_i] & \text{if } a_i^T a_i - q_i^T \tilde{q}_i^T = 0 \text{ and } \frac{\partial D(\lambda;0)}{\partial \lambda_i} = 0 \\ c_i & \text{if } a_i^T a_i - q_i^T \tilde{q}_i^T = 0 \text{ and } \frac{\partial D(\lambda;0)}{\partial \lambda_i} > 0 \\ \mu_i & \text{if } a_i^T a_i - q_i^T \tilde{q}_i^T \ne 0, \end{cases} \tag{5.13}$$

where

$$\mu_i := \operatorname{mid} \left\{ \frac{\gamma b_i - \sum_{j=1}^{i-1} (a_i^T a_j - q_i^T \tilde{q}_j^T)\lambda_j^k - \sum_{j=i+1}^{n} (a_i^T a_j - q_i^T \tilde{q}_j^T)\lambda_j^{k-1}}{a_i^T a_i - q_i^T \tilde{q}_i^T}, \; l_i, \; c_i \right\}.$$

Hence, the main calculation for the solution of (5.9) is the partial derivative (5.12). Direct computation of (5.12) takes $O(n^2 + nr)$ operations. In [11], it is shown that coordinate descent will become competitive if there is an efficient way to compute the partial derivative. Here, if we keep track of the vectors $v := \sum_{j=1}^n \lambda_j a_j$ and $p := \sum_{j=1}^n \lambda_j \tilde{q}_j$, then the complexity of the update of the derivative becomes $O(n+r)$ which is much better than $O(n^2 + nr)$. First notice that if we have v and p for the most recent λ, then

$$\frac{\partial D(\lambda; 0)}{\partial \lambda_i} = \frac{1}{\gamma}(-a_i^T v + q_i^T p) + b_i,$$

i.e. given v and p, calculating (5.12) takes only $O(n + r)$ operations. Next let us see how to update v and p. Assume we update λ_i^{k-1} to λ_i^k, then

$$v \leftarrow v + (\lambda_i^k - \lambda_i^{k-1})a_i \quad \text{and} \quad p \leftarrow p + (\lambda_i^k - \lambda_i^{k-1})\tilde{q}_i.$$

This shows the update of v and p is $O(n + r)$. In summary, the total complexity for each coordinate update is $O(n+r)$. Moreover, if A is a sparse matrix with an average of n_s nonzeros per row, then the complexity becomes $O(n_s + r)$.

6. Numerical Experiments. In this section, we present our experimental results on 126 CUTEr Hock-Schittkowski (`hs`) problems [10]. The coordinate descent algorithm described in §5.2 is used to solve the subproblems. We set the parameters stated in Algorithm 2 as $\gamma = 0.5$, $\rho_{(-1)} = 1$, $\beta_\phi = 0.7$, $\beta_v = 0.1$, $\beta_l = 0.6\beta_\phi(1-\beta_v)$, $\omega_0 = 10^{-2}$, $\theta_\rho = 0.9$, $\theta_\omega = 0.7$, $\theta_\alpha = 10^{-4}$, and $\eta^0 = \mathbf{0}_m$ with x^0 set as defined for each CUTEr problem. The maximum iteration limit for the subproblem solver was set as 2000, while a maximum iteration limit for Algorithm 2 is set to be 200. Define the maximum constraint violation $v_\infty(x)$ and the optimality error $\epsilon_{kkt}(x)$ as

$$v_\infty(x) := \max\{|c_i(x)| \ i = 1, \cdots, \bar{m}, (c_i(x))_+ \ i = \bar{m}+1, \cdots, m\},$$

$$\epsilon_{kkt}(x) := \max\left\{\left\|\nabla f(x) + \sum_{i=1}^m \eta_i \nabla c_i(x)\right\|_\infty, \|\eta \circ c(x)\|_\infty\right\},$$

where $\circ$ denotes element-wise product. We terminate the algorithm if $v_\infty(x) \leq 10^{-5}$ and $\epsilon_{kkt}(x) \leq 10^{-4}$, or the maximum iteration number 200 reached. These 126 problems are of small size, hence we use the exact Hessian in our implementation. If the Hessian is not positive definite, we apply the following modification to adjust its negative eigenvalues. Let $H = U\Lambda U^T$ be the eigen-decomposition of H, where $\Lambda = \text{diag}\{\lambda_1, \cdots, \lambda_n\}$. We replace all $\lambda_i < 0$ with a pre-determined small constant; in particular, 10^{-4} for these experiments. Let $\tilde{H}$ be the modified positive definite Hessian. If the condition number $cond(\tilde{H})$ of the modified Hessian is still greater than $t_c = 10^6$, then we use $H_{mod} = (t_c/cond(\tilde{H}))\tilde{H}$. Otherwise we use $H_{mod} = \tilde{H}$.

Table 1

Iteration summary for `hs93`. We need set the subproblem maximum iteration to be 2×10^5.

Iter	KKT	Violation	ρ	$f(x)$	r_c	r_v	r_ϕ	ω	Sub Iter
0	22.548600	0.012045	0.016423	136.003	0.917353	0.371655	0.956975	0.007000	90
1	3.872910	0.033563	0.014781	132.837	0.829690	0.498155	0.816669	0.004900	3381
2	0.253776	0.032065	0.013303	132.824	0.962329	0.181379	0.700017	0.003430	26287
3	0.174009	0.027869	0.013303	133.112	0.976902	0.229839	0.700412	0.002401	34210
4	0.197790	0.023335	0.013303	133.428	0.980734	0.244409	0.700065	0.001681	39342
5	11.072500	0.018771	0.012185	133.748	0.983078	0.264604	0.700024	0.001176	44832
6	41.793300	0.004973	0.009061	134.775	0.975333	0.785888	0.700039	0.000824	69876
7	0.341450	0.002241	0.009061	134.938	0.922223	0.654138	0.700010	0.000576	97147
8	0.118744	0.001101	0.009061	135.008	0.914334	0.623505	0.700020	0.000404	73176
9	0.048035	0.000613	0.009061	135.038	0.907883	0.598038	0.700036	0.000282	83228
10	0.022491	0.000379	0.009061	135.052	0.903235	0.579164	0.700007	0.000198	92409
11	0.012125	0.000250	0.009061	135.060	0.900605	0.568286	0.700010	0.000138	100744
12	0.007334	0.000170	0.009061	135.065	0.899296	0.562757	0.700013	0.000097	108530
13	0.004773	0.000118	0.009061	135.069	0.898687	0.560101	0.700006	0.000068	116196
14	0.003229	0.000082	0.009061	135.071	0.898416	0.558871	0.700010	0.000047	123518
15	0.002227	0.000057	0.009061	135.072	0.898312	0.558352	0.700054	0.000033	130668
16	0.001547	0.000040	0.009061	135.073	0.898242	0.558001	0.700017	0.000023	138177
17	0.001080	0.000028	0.009061	135.074	0.898214	0.557843	0.700003	0.000016	145662
18	0.000755	0.000020	0.009061	135.075	0.898217	0.557822	0.700035	0.000011	152807
19	0.000528	0.000014	0.009061	135.075	0.898204	0.557747	0.700017	0.000008	160168
20	0.000370	0.000010	0.009061	135.075	0.898198	0.557708	0.700006	0.000006	167328
21	0.000259	0.000007	0.009061	135.076	0.898198	0.557694	0.700007	0.000004	174759
22	0.000181	0.000005	0.009061	135.076	0.898218	0.557770	0.700061	0.000003	182035
23	0.000127	0.000003	0.009061	135.076	0.898197	0.557680	0.700015	0.000002	189285
24	0.000089	0.000002	0.009061	135.076	0.898194	0.557662	0.700006	0.000001	196618

For these experiments, we have the following observations.

- Out of 126 CUTEr `hs` problems, our algorithm successfully solved 116, which is a success rate of about $92\% \approx 116/126$. Tables 2, 3, and 4 summarize the detailed output for these 116 successful cases.

- Our (DUST) updating strategy works very well in these experiments. To illustrate the behavior of the penalty parameter updates, we plot ρ values for three sample problems—`hs11`, `hs43`, and `hs61`—in Figure 1.
- The parameter ω does not require much tuning. We used $\omega_0 = 10^{-2}$ across all problems and achieved our 92% success rate. We also ran the experiment with $\omega_0 = 10^{-1}$ and we see the same set of 116 problems solved successfully.
- The coordinate descent algorithm performs poorly on ill-conditioned subproblems. For example, we needed to increase the maximum number of iterations to be 2×10^5 to make `hs93` converge. Table 1 shows the detailed iteration information. Since the focus of this paper is on the ρ update strategy, we did not fully explore other subproblem solvers that might have performed better. Instead, we simply increased the iteration limit for this problem.
- In a few cases, the Hessian modification strategy described above might not work well. For example, for problems `hs70`, `hs72`, and `hs75`, we had to reduce the modification constant to 10^{-8} to achieve convergence.

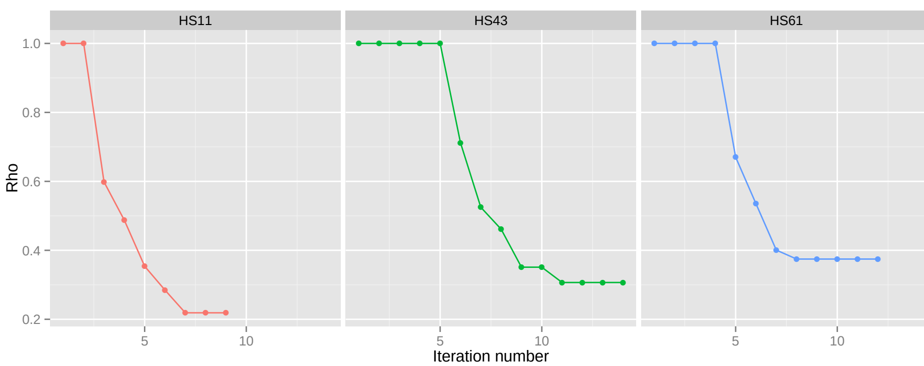

FIG. 1. *ρ values for problems* `hs11`*,* `hs43` *and* `hs61`*.*

7. Conclusion. In this paper, we have proposed a penalty-SQP framework for solving nonlinear optimization problems. The novelty of this work is a dynamic penalty parameter updating strategy that is carried out within the QP subproblem solver, so that at the end of the QP solve, a search direction and a new penalty parameter are both obtained. The key idea is to force improvement toward feasibility whenever optimality and complementarity are sufficiently improved. This enables the SQP algorithm to finish penalty parameter updating and infeasibility detection via *inexact* solves for only *one* subproblem in each iteration, a feature which is not shared with most contemporary solvers which require two subproblem solves per iteration.

The convergence properties that we have proved for our algorithm guarantees the effectiveness of our updating strategy under reasonable assumptions. The empirical effects of our strategy is demonstrated in numerical results on small CUTEr examples. We remark, however, that the performance could be further enhanced with the development of a more efficient QP subproblem solver and a more robust approach to addressing ill-conditioning of the Hessian approximation.

[1]Increased sub-problem maximum number of iteration to be 10^4

[2]Decrease the Hessian modification constant to be 10^{-8}, since the scale of this problem's Hessian is around 10^{-4}

[3]Subproblem is ill-conditioned. Increased subproblem maximum iteration number to be 2×10^5.

REFERENCES

[1] J. V. Burke, *A Sequential Quadratic Programming Algorithm for Potentially Infeasible Mathematical Programs*, Journal of Mathematical Analysis and Applications, 139 (1989), pp. 319–351.

[2] J. V. Burke, *A Robust Trust Region Method for Constrained Nonlinear Programming Problems*, SIAM Journal on Optimization, 2 (1992), pp. 325–347.

[3] J. V. Burke, F. E. Curtis, and H. Wang, *A Sequential Quadratic Optimization Algorithm with Rapid Infeasibility Detection*, SIAM Journal on Optimization, 24 (2014), pp. 839–872.

[4] J. V. Burke and S.-P. Han, *A Robust Sequential Quadratic Programming Method*, Mathematical Programming, 43 (1989), pp. 277–303.

[5] R. H. Byrd, G. Lopez-Calva, and J. Nocedal, *A Line Search Exact Penalty Method Using Steering Rules*, Mathematical Programming, 133 (2012), pp. 39–73.

[6] R. H. Byrd, J. Nocedal, and R. A. Waltz, *Steering Exact Penalty Methods for Nonlinear Programming*, Optimization Methods and Software, 23 (2008), pp. 197–213.

[7] F. E. Curtis, T. Johnson, D. P. Robinson, and A. Wächter, *An Inexact Sequential Quadratic Optimization Algorithm for Nonlinear Optimization*, SIAM Journal on Optimization, 24 (2014), pp. 1041–1074.

[8] S. P. Han, *A Globally Convergent Method for Nonlinear Programming*, Journal of Optimization Theory and Applications, 22 (1977), pp. 297–309.

[9] S. P. Han and O. L. Mangasarian, *Exact Penalty Functions in Nonlinear Programming*, Mathematical Programming, 17 (1979), pp. 251–269.

[10] W. Hock and K. Schittkowski, *Test examples for nonlinear programming codes*, Journal of Optimization Theory and Applications, 30 (1980), pp. 127–129.

[11] Y. Nesterov, *Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems*, SIAM Journal on Optimization, 22 (2012), pp. 341–362.

[12] J. Nocedal, *Updating Quasi-Newton Matrices with Limited Storage*, Mathematics of Computation, 35 (1980), pp. 773–782, http://www.ams.org/journals/mcom/1980-35-151/S0025-5718-1980-0572855-7/S0025-5718-1980-0572855-7.pdf.

[13] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, Second ed., 2006.

[14] M. J. D. Powell, *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations*, in Numerical Analysis, Lecture Notes in Mathematics, Springer, Berlin, Heidelberg, 1978, pp. 144–157.

[15] R. Rockafellar and R. Wets, *Variational Analysis*, vol. 317, Springer, 1998.

[16] M. V. Solodov, *Constraint qualifications*, Wiley Encyclopedia of Operations Research and Management Science, (2010).

[17] R. B. Wilson, *A Simplicial Algorithm for Concave Programming*, PhD thesis, Graduate School of Business Administration, Harvard University, Cambridge, MA, USA, 1963.

Table 2

CUTEr test result, 116 successful cases out of 126 CUTEr `hs` *problems.*

Problem	Iter #	$f(x)$ #	$f(x)$	Violation	KKT Error	Final ρ
`hs1`	24	34	4.215353e−17	0.000000e+00	2.983621e−09	1.000000e+00
`hs10`	8	9	−1.000001e+00	1.551523e−06	3.360370e−06	1.000000e+00
`hs100`	12	24	6.806301e+02	2.248202e−14	5.168706e−06	4.800507e−01
`hs100lnp`	11	42	6.806301e+02	4.923944e−06	1.323450e−05	1.350852e−01
`hs100mod`	7	23	6.786796e+02	3.594508e−09	4.942095e−09	5.738871e−01
`hs101`	41	79	1.809764e+03	2.190603e−07	2.074525e−05	1.327445e−04
`hs102`	46	97	9.118752e+02	3.199962e−06	3.555227e−05	2.836799e−04
`hs103`	37	65	5.436673e+02	5.349532e−07	2.350039e−06	4.882164e−04
`hs104`	20	48	4.200000e+00	1.354694e−11	6.367180e−09	5.625000e−02
`hs105`	46	53	1.044612e+03	1.191199e−10	6.704045e−05	2.737157e−03
`hs107`	21	90	5.054990e+03	4.531648e−06	2.228662e−05	1.821769e−05
`hs108`	20	31	−8.660114e−01	6.632213e−06	2.884348e−06	5.062500e−02
`hs109`	146	428	5.362069e+03	9.145833e−08	9.484826e−05	1.025156e−02
`hs11`	8	9	−8.498465e+00	1.225127e−07	1.349376e−06	2.187265e−01
`hs110`	3	5	−4.577848e+01	0.000000e+00	6.067373e−09	1.000000e+00
`hs111`	21	34	−4.776117e+01	7.478155e−06	1.445093e−05	4.413721e−02
`hs111lnp`	19	28	−4.776110e+01	8.713650e−06	2.404894e−05	4.153435e−02
`hs112`	19	23	−4.776104e+01	7.853528e−06	2.875205e−07	3.634359e−03
`hs113`	14	15	2.430624e+01	5.839225e−06	7.816360e−06	4.304672e−01
`hs117`	17	89	3.234887e+01	5.725331e−06	1.417529e−05	3.695221e−03
`hs118`	20	21	9.329922e+02	7.826075e−06	1.803395e−06	7.569401e−02
`hs119`	21	22	2.448995e+02	5.612386e−06	1.250750e−06	1.605909e−01
`hs12`	5	9	−3.000000e+01	3.991066e−07	2.934243e−07	1.000000e+00
`hs14`	19	51	1.393447e+00	9.569376e−06	6.579044e−06	3.723179e−01
`hs15`	46	48	3.064999e+02	9.332233e−08	3.255544e−05	2.130850e−06
`hs16`	20	21	2.314422e+01	6.835587e−06	7.748629e−05	7.390441e−03
`hs17`	9	11	1.000000e+00	0.000000e+00	6.917612e−05	1.824800e−02
`hs18`	8	11	5.000000e+00	3.778853e−08	2.030860e−08	1.000000e+00
`hs19` [1]	32	40	-6.961825e+03	9.324828e−06	7.484845e−06	5.080023e−04
`hs2`	17	43	4.941229e+00	2.360334e−13	4.601504e−05	1.000000e+00
`hs20`	34	39	4.019865e+01	4.047174e−07	5.688441e−05	2.441406e−04
`hs21`	2	3	−9.996000e+01	4.440892e−16	4.999500e−09	1.000000e+00
`hs21mod`	8	9	−9.596000e+01	0.000000e+00	5.982811e−07	1.628848e−01
`hs22`	8	15	9.999971e−01	4.352097e−06	3.243715e−05	1.000000e+00
`hs23`	16	18	2.000089e+00	0.000000e+00	7.371917e−05	7.119141e−04
`hs24`	17	52	−9.998614e−01	0.000000e+00	7.296459e−05	1.250000e−01
`hs25`	129	530	1.818451e−16	0.000000e+00	4.306988e−07	2.222948e−13
`hs26`	13	28	2.172765e−10	4.361497e−06	2.031413e−07	1.000000e+00
`hs268`	10	11	1.818989e−11	4.440892e−16	3.004539e−05	1.000000e+00
`hs27`	7	11	4.000000e−02	1.781389e−19	5.580537e−05	1.000000e+00
`hs28`	2	3	1.117108e−13	0.000000e+00	2.034355e−07	1.000000e+00
`hs29`	8	9	−2.262742e+01	5.095551e−10	1.211382e−05	6.804834e−01
`hs3`	7	8	2.338799e−04	0.000000e+00	9.672226e−05	1.000000e+00
`hs30`	12	56	1.000051e+00	0.000000e+00	3.892105e−05	2.500000e−01
`hs31`	7	9	6.000000e+00	4.216536e−10	1.079204e−07	1.086945e−01
`hs32`	19	54	9.999618e−01	9.550907e−06	1.213808e−06	3.177332e−02
`hs33`	4	5	−4.000000e+00	0.000000e+00	4.135340e−06	8.862938e−02
`hs34`	19	25	−8.340328e−01	8.637830e−06	3.359038e−07	9.000000e−01
`hs35`	1	2	1.111111e−01	0.000000e+00	8.332954e−05	1.000000e+00
`hs35i`	1	2	1.111111e−01	0.000000e+00	8.332954e−05	1.000000e+00
`hs35mod`	3	38	2.500000e−01	2.741989e−09	1.234391e−08	4.050000e−01
`hs36`	19	20	−3.299992e+03	0.000000e+00	6.395368e−05	1.090995e−03
`hs37`	7	8	−3.456000e+03	5.320189e−11	9.128910e−05	6.362685e−03
`hs38`	37	56	1.686699e−12	0.000000e+00	6.564919e−07	1.000000e+00

Table 3
Continued

Problem	Iter #	$f(x)$ #	$f(x)$	Violation	KKT Error	Final ρ
hs39	34	35	−1.000005e+00	5.137450e−06	1.876909e−06	3.906250e−03
hs3mod	2	3	8.026142e−14	0.000000e+00	1.162326e−07	1.000000e+00
hs4	2	3	2.666667e+00	0.000000e+00	3.149394e−15	2.287679e−01
hs40	19	20	−2.499981e−01	7.807879e−06	6.256268e−06	2.500000e−01
hs41	16	138	1.925926e+00	3.332050e−06	2.240422e−06	1.099376e−01
hs42	4	18	1.385786e+01	2.177929e−10	1.588578e−05	3.134193e−01
hs43	13	15	−4.399979e+01	2.561947e−09	6.005441e−05	2.816279e−01
hs44	19	54	−1.500004e+01	7.714744e−06	6.233836e−07	7.148998e−03
hs44new	19	20	−1.500003e+01	7.210908e−06	1.791073e−06	7.943331e−03
hs45	13	14	1.000370e+00	1.010592e−11	6.605379e−05	1.535793e−01
hs46	18	19	3.203845e−09	8.879526e−06	2.695075e−07	1.000000e+00
hs47	16	18	3.525500e−11	8.142011e−06	1.967257e−07	1.000000e+00
hs48	9	10	1.344490e−20	2.894060e−06	3.215622e−10	1.000000e+00
hs49	11	12	5.353471e−07	5.126566e−12	6.018853e−05	1.000000e+00
hs5	5	8	−1.913223e+00	0.000000e+00	2.071583e−08	5.904900e−01
hs50	12	15	4.191775e−19	3.619227e−06	1.220630e−09	7.290000e−01
hs51	2	3	4.819461e−17	2.397612e−08	9.999561e−09	1.000000e+00
hs52	23	24	5.326616e+00	5.142620e−06	2.504244e−07	3.906250e−03
hs53	23	24	4.092989e+00	7.977733e−06	2.091871e−07	7.812500e−03
hs54	4	5	−1.561094e−01	1.250555e−10	7.557178e−05	1.000000e+00
hs55	19	20	6.666665e+00	6.970546e−06	2.205215e−07	7.812500e−03
hs56	11	14	−3.456001e+00	8.092971e−07	1.346679e−05	4.536772e−01
hs57	1	2	3.064627e−02	0.000000e+00	2.696159e−06	1.000000e+00
hs59	19	51	−7.802789e+00	0.000000e+00	7.464629e−06	4.500000e−01
hs6	9	24	8.091820e−10	1.605262e−07	2.842575e−05	1.000000e+00
hs60	5	6	3.256820e−02	9.958889e−08	1.894968e−07	1.000000e+00
hs61	11	47	−1.436461e+02	5.941746e−06	5.540538e−05	1.834697e−01
hs62	5	7	−2.627251e+04	1.526557e−16	4.647296e−07	1.455578e−03
hs63	19	21	9.617152e+02	7.907387e−06	4.228438e−06	3.543676e−03
hs64	43	44	6.299843e+03	1.254110e−08	7.139929e−05	4.183808e−02
hs65	6	7	9.535287e−01	2.046497e−06	1.686632e−07	1.000000e+00
hs66	6	10	5.181592e−01	6.147406e−06	3.681782e−06	9.000000e−01
hs67	16	17	−1.162119e+03	0.000000e+00	6.464876e−05	1.000000e+00
hs69	21	23	−9.567131e+02	4.634015e−06	7.811305e−05	1.767413e−03
hs7	7	8	−1.732051e+00	9.245062e−07	1.247472e−06	1.000000e+00
hs70 [2]	33	52	7.498464e−03	0.000000e+00	8.572072e−05	4.398314e−09
hs71	35	57	1.701402e+01	4.918665e−06	1.485627e−06	1.953125e−03
hs72 [2]	56	57	7.276855e+02	4.211156e−11	5.434002e−05	1.245512e−05
hs73	22	27	2.989858e+01	2.871592e−13	5.942612e−05	7.693523e−04
hs74	26	29	5.126498e+03	8.289811e−07	6.799524e−05	1.012367e−01
hs75 [2]	147	675	5.174413e+03	4.500867e−06	9.285981e−05	3.541118e−04
hs76	7	77	−4.681696e+00	0.000000e+00	7.902726e−05	1.328603e−01
hs76i	8	45	−4.681622e+00	6.703622e−16	4.214496e−05	2.152336e−01
hs77	19	57	2.415044e−01	8.793855e−06	1.880529e−07	2.500000e−01
hs78	21	22	−2.919704e+00	5.430606e−06	4.818101e−07	2.278125e−02
hs79	17	18	7.877670e−02	4.575711e−06	7.649023e−08	1.000000e+00
hs8	15	17	−1.000000e+00	6.620267e−06	1.631257e−10	1.000000e+00
hs80	16	49	5.394959e−02	6.516877e−06	9.739155e−08	5.000000e−01
hs81	19	20	5.394957e−02	7.186109e−06	9.438093e−09	4.923855e−02
hs86	18	157	−3.234877e+01	8.333681e−06	2.522773e−07	2.649447e−03
hs88	33	37	1.362657e+00	2.312826e−12	3.800417e−07	6.445449e−04
hs89	31	66	1.362657e+00	6.336861e−13	1.383187e−07	6.934900e−04
hs9	2	3	−5.000000e−01	6.821210e−13	6.842740e−05	1.000000e+00

TABLE 4
Continued

Problem	Iter #	$f(x)$ #	$f(x)$	Violation	KKT Error	Final ρ
hs90	33	49	1.362657e+00	7.690347e−12	4.771747e−06	6.524855e−04
hs91	34	48	1.362657e+00	8.400233e−14	3.786938e−08	6.600099e−04
hs92	33	44	1.362657e+00	1.792880e−11	1.663324e−05	6.774808e−04
hs93 [3]	24	28	1.350758e+02	2.302123e−06	8.869173e−05	9.060500e−03
hs95	35	267	1.562139e−02	5.984454e−14	9.168869e−06	5.500098e−06
hs96	33	154	1.581460e−02	0.000000e+00	8.797320e−05	1.204551e−05
hs97	32	35	4.071188e+00	2.094652e−07	8.714763e−05	1.002393e−06
hs98	33	37	4.071231e+00	5.368689e−08	7.785792e−05	1.804307e−06
hs99	19	20	−8.310799e+08	7.325137e−06	6.074152e−07	5.000000e−01