SIAM J. OPTIM.
Vol. 30, No. 3, pp. 1822–1849

INEXACT SEQUENTIAL QUADRATIC OPTIMIZATION WITH PENALTY PARAMETER UPDATES WITHIN THE QP SOLVER*

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 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, inexact matrix-free methods, infeasibility detection

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

DOI. 10.1137/18M1176488

1. Introduction.

In this paper, we consider the use of sequential quadratic optimization (SQP) methods for solving large-scale nonlinear optimization problems (NLPs) [1, 2, 3, 5, 9, 13, 16]. 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 [8]. However, the practical efficiency of such an approach remains an open question. A critical aspect of their implementation is the choice of a subproblem solver since it 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

*Received by the editors March 20, 2018; accepted for publication (in revised form) April 7, 2020; published electronically July 2, 2020.
https://doi.org/10.1137/18M1176488
†Department of Mathematics, University of Washington, Seattle, WA 98195 (jvburke@uw.edu).
‡Department Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015 (frank.e.curtis@gmail.com).
§School of Information Science and Technology, ShanghaiTech University, Shanghai, 201210 (wanghao1@shanghaitech.edu.cn).
¶Department of Mathematics, University of Washington, Sunnyvale, CA 94085 (jsw1119@math.washington.edu).

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 previously proposed techniques that only update the penalty parameter after a sequence of iterations, in hindsight at the end of an iteration [1, 9, 10], or at the expense of numerous subproblem solves within a single iteration [3, 6, 7].

To provide some context about how the algorithm proposed in this paper compares to other recently proposed SQP-type methods in the literature, let us contrast our approach with those proposed in [3, 8]. The penalty SQP method proposed in [3] was motivated by the desire to formulate an SQP approach that attains strong convergence guarantees when solving problems regardless of whether they involve constraints that are feasible or infeasible. Toward this end, the approach involved a novel dynamic updating scheme for the penalty parameter that, e.g., quickly drives the algorithm toward constraint violation minimization when infeasibility is detected. The approach relies on exact solves of two QP subproblems per iteration; the first determines the reduction that can be obtained in a local model of an infeasibility measure while the second minimizes a local model of the objective while ensuring that the reduction in a local model of the infeasibility measure is proportional to that attained by the solution of the first QP. In this manner, rapid convergence can be attained when solving either a feasible or infeasible problem, although a high price is paid by needing exact subproblem solutions. The method in [8] overcomes this obstacle by allowing inexact subproblem solves. However, it also potentially requires (approximate) solutions of two QPs per iteration, one aimed at minimizing constraint violation and one aimed at reducing the objective subject to an appropriate bound on constraint violation. The approach proposed in this paper also allows inexactness in the QP solves, but only requires solving a *single* QP in each iteration. This is made possible by a new strategy for dynamically updating the penalty parameter *within* the QP solver. This dynamic penalty parameter updating strategy is the focus of our investigation. We prove that our algorithm does not reduce the penalty parameter unnecessarily and that one can ensure convergence to an optimal solution (when a given problem is feasible) or to an infeasible stationary point (when a given problem is infeasible).

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 automatic 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 section 2, we introduce a basic penalty-SQP algorithm. Our penalty parameter updating strategy is detailed in section 3. A complete algorithm is presented and analyzed in section 4. The results of numerical experiments are presented in section 5. Concluding remarks are provided in section 6.

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 $\operatorname{mid}\{a, b, c\}$, 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 $\mathbb{N}$. The extended real number line is defined as $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$.

For a set of scalars $b_i \in \mathbb{R}$ for $i \in \{1, \ldots, m\}$, we 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 ith component of $\mathbf{y}$ (an element of $\mathbb{R}^{d_i}$) is denoted y^i while the jth element of y^i is written as y^i_j. For convex sets $C_i \in \mathbb{R}^{d_i}$ for $i \in \{1, \ldots, m\}$, the distance functions are defined as

$$\operatorname{dist}_2\left(y^i \mid C_i\right) := \inf_{z^i \in C_i} \|y^i - z^i\|_2.$$

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

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 $C := \{d : \langle a, d\rangle + b = 0\}$ (resp., half space $C = \{d : \langle a, d\rangle + b \leq 0\}$), one finds that $\delta^*(y|C) < \infty$ if and only if $\langle y, a\rangle = \pm\|y\|_2\|a\|_2$ (resp., $\langle y, a\rangle = \|y\|_2\|a\|_2$). In this case,

$$y = \zeta a \quad \text{with} \quad \zeta = \frac{1}{\|a\|_2^2}\langle y, a\rangle, \quad \text{meaning that} \quad \delta^*(y|C) = -\zeta b. \tag{1.1}$$

For an iterative algorithm, we use superscript k to indicate the iteration number for vectors and subscript k for scalars to avoid confusion with the kth power of the scalar, e.g., x^k and ρ_k. For an algorithm for solving the subproblem, we use superscript (j) to indicate the iteration number for vectors and subscript (j) for scalars.

2. A penalty-SQP framework. Consider the following NLP 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:

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

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 a 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 [8].

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 over a convex set $X \subseteq \mathbb{R}^n$ containing $\{0\}$ by

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

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 $\rho_k \in (0, \rho_{k-1}]$, i.e.,

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

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,

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

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

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

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

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 section 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 section 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

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

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.

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

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

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. 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

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

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

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

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

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

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

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

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

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

We close this subsection by noting that the projection onto the set C_i,

$$
P_{C_i}(y^i) := \arg\min_{z^i \in C_i} \left\| z^i - y^i \right\|_2,
$$

is easy to compute for any $i \in \{1,\dots,m\}$; in particular,

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

3.2. Updating the penalty parameter. Given $\rho \ge 0$, 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

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

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

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

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

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

Let $J_\omega^{(0)} := J^{(0)} + \omega$ for any scalar $\omega \in (0, \infty)$. (As will be 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:

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

(Referring 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

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

we define the complementarity measure

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

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,

$$r_c^{(j)} := 1 - \sqrt{\frac{\chi^{(j)}}{J_\omega^{(0)}}} \geq \beta_v. \tag{Rc}$$

In our strategy, if the optimality QP subproblem is solved sufficiently accurately, then we turn to verify whether feasibility has also been improved to a satisfactory extent. Therefore, the key idea here is to determine a criterion reflecting that the optimality QP has been solved sufficiently accurately. Making this determination requires us to check a measure of complementarity. In particular, if the initial objective $J^{(0)}$ is far from optimal, then $r_\phi^{(j)} \approx 1$ might not indicate that the subproblem solution is nearly primal-dual optimal since a large $J^{(0)}$ can cause the numerator of $r_\phi^{(j)}$ to be very close to the denominator, even though the dual value is far from dual optimality. As a result, the updating strategy may be triggered too early, so that ρ is inappropriately driven to zero. Therefore, we need a certification showing the progress achieved by the dual estimates, which can be reflected by the complementary condition (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 conclude 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 occurrence 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 $r^{(j)}_v > \beta_v$ 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
$$\frac{J^{(0)}_\omega - l(d^*_0, 0)}{J^{(0)}_\omega - (D(\mathbf{u}^*_0, 0))_+} = \frac{\omega}{\omega} = 1,$$
meaning that $r^{(j)}_v > \beta_v$ 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
$$r^{(j)}_v = \frac{\omega}{\omega + J^{(0)} - (D(\mathbf{w}^{(j)}, 0))_+}.$$
There are now two cases to consider. If $r^{(j)}_v < \beta_v$, then (DUST) decreases the penalty parameter, as is appropriate. Otherwise, if $r^{(j)}_v \geq \beta_v$, then—with a sufficiently accurate dual solution—(DUST) returns a null step to the penalty-SQP framework. (In a later subproblem solved with a smaller ω, one would find that either (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.) 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 $j \in \mathbb{N}$,*

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

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* $j \in \mathbb{N}$*,*

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

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

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

$$\begin{aligned} \kappa_2 &:= \|g\|_2\kappa_1 + \tfrac{1}{2}\|H_f\|_2\kappa_1^2 \\ \text{and} \quad \kappa_3 &:= \frac{\kappa_0 + \rho_{(0)}\|g\|_2}{2\underline{\lambda}}(\kappa_0\|H_0^{-1}\|_2\|H_f\|_2 + \|g\|_2) + \tfrac{1}{2}\kappa_0\|H_0^{-1}\|_2\|g\|_2. \end{aligned}$$

Proof. For the primal values, it holds true that

$$\begin{aligned} |J(d^{(j)}, \rho_{(j)}) - J(d^{(j)}, 0)| &= |\rho_{(j)}\langle g, d^{(j)}\rangle + \tfrac{1}{2}\langle d^{(j)}, H_{\rho_{(j)}} d^{(j)}\rangle - \tfrac{1}{2}\langle d^{(j)}, H_0 d^{(j)}\rangle| \\ &= |\rho_{(j)}\langle g, d^{(j)}\rangle + \tfrac{1}{2}\rho_{(j)}\langle d^{(j)}, H_f d^{(j)}\rangle| \\ &\leq \rho_{(j)}(\|g\|_2\|d^{(j)}\|_2 + \tfrac{1}{2}\|H_f\|_2\|d^{(j)}\|_2^2), \end{aligned}$$

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

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

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. Let*

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

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

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

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

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

Thus, for any

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

it follows from the first inequality of (3.16) that

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

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 was 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:

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

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.

The intuition of this posterior updating strategy is to detect whether the iterate may be near an infeasible stationary point. If a step has achieved improvement on optimality but not very much on feasibility, then the algorithm should decrease ρ to reduce the effect of the objective in the penalty function. This is the typical approach used by penalty methods that update the penalty parameter in hindsight at the end of an iteration. 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* [12]. Our analysis, which focuses on primal convergence, works with any such choice as long as the sequence of dual estimates remains bounded (see below).

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 qualification 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 section 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. *The compact convex set $X \subset \mathbb{R}^n$ with $0 \in \operatorname{int}(X)$ is used in defining all subproblems, and there exist positive scalar constants $\underline{\Lambda}, \overline{\Lambda}$ and K_0 with $\underline{\Lambda} \leq \overline{\Lambda}$ such that the following hold true.*

(i) *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 $\{x^k + d^k\}$.*

(ii) *For all $k \in \mathbb{N}$ and any $\rho \in [0, \rho_0]$,*

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

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 $(x^0, \eta^0) \in \mathbb{R}^n \times \mathbb{R}^m$.
for $k \in \mathbb{N}$ **do**
    Set $\rho_{(0)} \leftarrow \rho_{k-1}$
    for $j \in \mathbb{N}$ **do**
        Generate a primal-dual feasible solution estimate $(d^{(j)}, \mathbf{u}^{(j)}, \mathbf{w}^{(j)})$
        Set $\rho_{(j+1)}$ by applying (DUST)
    Set $d^k \leftarrow d^{(j)}$ and $\tilde{\rho}_k \leftarrow \rho_{(j)}$.
    Set ρ_k by applying (PSST)
    Let α^k be the largest value in $\{\gamma^0, \gamma^1, \gamma^2, \dots\}$ such that

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

    Choose $\omega_{k+1} \in (0, \theta_\omega \omega_k]$.
    Set $x^{k+1} \leftarrow x^k + \alpha_k d^k$ and choose $\eta \in \mathbb{R}^m$.

(iii) $\kappa_{0,k} \le K_0$ *for all* $k \in \mathbb{N}$.
(iv) $\|\nabla c_i(x^k)\|_2 > 0$ *for all* $k \in \mathbb{N}$ *and* $i \in \{1, \dots, m\}$.
(v) $\{\eta^k\}$ *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

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

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* $\rho \ge 0$.
(ii) $\Delta l(d^*(0, x^k, \eta^k); x^k) \ge 0$, *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

$$\begin{aligned} &J(d^*(0; x^k, \eta^k), 0; x^k, \eta^k) \le J(0, 0; x^k, \eta^k) \\ \text{and} \quad &J(d^*(\rho_k; x^k, \eta^k), \rho_k; x^k, \eta^k) \le J(0, 0; x^k, \eta^k) \end{aligned}$$

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 satisfies* $\alpha_k \ge C\Delta l(d^k, \rho_k; x^k)$ *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

$$\phi(x^k + \alpha d^k, \rho_k) - \phi(x^k, \rho_k) \le -\alpha \Delta l(d^k, \rho_k; x^k) + \tau\alpha^2 \|d^k\|_2^2.$$

Thus, for any $\alpha \in [0, (1 - \theta_\alpha)\Delta l(d^k, \rho_k; x^k)/(\tau\|d^k\|_2^2)]$, it follows that

$$-\alpha \Delta l(d^k, \rho_k; x^k) + \tau\alpha^2 \|d^k\|_2^2 \le -\alpha\theta_\alpha \Delta l(d^k, \rho_k; x^k),$$

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

$$\alpha_k \ge \gamma(1 - \theta_\alpha)\Delta l(d^k, \rho_k; x^k)/(\tau\|d^k\|_2^2).$$

The result follows from this inequality since $\{\|d^k\|_2\}$ 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

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

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

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

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

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

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

which implies

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

The result then follows from this inequality, the fact that $\{\rho_k\}$ is monotonically decreasing, and since $f(x^{k+1}) \ge \underline{f}$ 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 Lemmas 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

$$\begin{aligned} 0 &= \lim_{k\to\infty} -\Delta J(d^k, 0; x^k, \eta^k) = \lim_{k\to\infty} -\Delta l(d^k, 0; x^k) + \tfrac{1}{2}\langle d^k, H(0; x^k, \eta^k) d^k\rangle \\ &= \lim_{k\to\infty} \tfrac{1}{2}\langle d^k, H(0; x^k, \eta^k) d^k\rangle \geq \lim_{k\to\infty} \tfrac{1}{2}\underline{\Delta}\|d^k\|_2^2. \end{aligned}$$

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 $\{J(0, 0; x^k, \eta^k)\}_{k\in\mathcal{S}\cap\mathcal{D}} = \{v(x^k)\}_{k\in\mathcal{S}\cap\mathcal{D}} \to 0$ 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 second inequality is by (Rphi) and the third 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

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

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

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

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

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

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

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

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 infinite.

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,

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

The dual form [14] of MFCQ states that $\zeta^i = 0, i \in \mathcal{E} \cup \mathcal{A}(x)$ is the unique solution of the linear system

$$\sum_{i \in \mathcal{E} \cup \mathcal{A}(x)} \zeta^i \nabla c_i(x) = 0, \ \zeta^i \geq 0, i \in \mathcal{A}(x).$$

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 $\mathbf{u}^*$ 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 that $d = 0$ lies in the interior of X that $u^*_{m+1} = 0$ and

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

meaning that $\delta^*(u_i^* | C_i^*) = -\zeta_*^i \frac{c_i(x^*)}{\|\nabla c_i(x^*)\|_2}$ 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

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

Therefore, x^* violates the dual form of the MFCQ because $\zeta_*^i, i\in\mathcal{E}\cup\mathcal{A}(x^*)$ are not all zero. 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. Numerical experiments. We present numerical experiments that illustrate the impact of the central contribution of this paper, namely, our dynamic penalty parameter updating strategy. With this in mind, the choice of the QP subproblem solver only requires that it generate both primal and dual solution estimates. For this purpose, a coordinate descent algorithm is used to generate the primal and dual variables to solve the subproblem [4]. Experimental results are given for both feasible and infeasible test sets. Our code is implemented using Python and tested on a 2014 MacBook Air with 4 GB memory and a 1.4 GHz Intel Core i5 processor.

5.1. Feasible test. First, we tested on 126 CUTEr Hock–Schittkowski (`hs`) problems [11] which are all feasible. 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 10^6, while a maximum iteration limit for Algorithm 2 was set to be 200. We defined the maximum constraint violation $v_\infty(x)$ and the optimality KKT error $\epsilon_{opt}(x)$ as

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

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

where $\circ$ denotes elementwise product. We terminate the algorithm if $v_\infty(x) \le 10^{-5}$ and $\epsilon_{opt}(x) \le 10^{-4}$, or the maximum iteration number 200 is reached. These 126 problems are of small size; hence we use the exact Hessian in our implementation. If the Hessian, call it H, is not positive definite, then we apply the following modification to adjust its negative eigenvalues. Let $H = U\Lambda U^T$ be the eigendecomposition of H, where $\Lambda = \mathrm{diag}\{\lambda_1, \dots, \lambda_n\}$. For a prescribed constant $\tau > 0$ (e.g., we use $\tau = 10^{-4}$ in these experiments), we reset $\lambda_i \leftarrow \max\{\lambda_i, \tau\}$ and replace H with $U\tilde{\Lambda}U^T$ where $\tilde{\Lambda}$ is the corresponding modification of Λ. We also perform the following modification to control the condition number of the Hessian (approximation) employed in the algorithm: If $\mathrm{cond}(H) > t_c > 0$ (e.g., we use $t_c = 10^6$ in these experiments), then we replace H by $\alpha H + (1-\alpha)I$ where α is the largest value in $[0, 1]$ such that the resulting matrix has a condition number less than or equal to t_c.

Table 1

Performance comparison of SQuID and the proposed algorithm on feasible problems.

Problem type	Algorithm	Succeed	Fail	Infeasible	Total
Feasible `hs` problems	SQuID	110 (90.16%)	11 (9.02%)	1 (0.82%)	122
	Proposed	115 (91.20%)	11 (8.80%)	0	126

For these experiments, we have the following observations.

- Out of 126 CUTEr `hs` problems, our algorithm successfully solved 115, which is a success rate of about 91% ≈ 115/126.[1] Our proposed method outperforms the SQuID algorithm proposed in [3], which is also a penalty-SQP method with automatic infeasibility detection, although it requires two exact QP solves per iteration. The comparison statistics[2] are shown in Table 1. For the complete set of numerical test results, see [4].
- Our (DUST) updating strategy works very well in these experiments, and does not cause ρ to become excessively small for most cases. 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 did not require much tuning. We used $\omega_0 = 10^{-2}$ across all problems and achieved our 91% success rate. We also ran the experiment with $\omega_0 = 10^{-1}$ and saw the same set of 115 problems solved successfully.
- We test the sensitivity of our algorithm with respect to the parameter β_ϕ. We ran the same experiments with $\beta_\phi = 0.5$ and $\beta_\phi = 0.99$. We have 113 successful cases for $\beta_\phi = 0.5$, and 111 successful cases for $\beta_\phi = 0.99$. The additional failure cases in $\beta_\phi = 0.5$ and $\beta_\phi = 0.99$ compared to $\beta_\phi = 0.7$ are all due to the subproblem exceeding the maximum iteration number.
- Coordinate descent performs poorly on ill-conditioned subproblems. We observed that some subproblems require more than 5×10^5 steps to reach the specified accuracy. Since the focus of this paper is on the ρ update strategy, we did not explore other subproblem solvers that might have performed better. Instead, we used a large iteration limit for the subproblem solver.

[1]The termination criterion of SQuID in [3] is based on the relative KKT residual scaled by ρ.

[2]The performance statistics for SQuID is obtained from [3], where the overall number of `hs` problems is 122 due to compiling errors.

- In a few cases, the Hessian modification strategy described above did not work well. For example, for problems `hs72` and `hs75`, we had to reduce the modification constant to 10^{-8} to achieve convergence, since the scale of the Hessian for both problems is around 10^{-4}. For problem `hs93`, convergence is observed with modification constant 10^{-2}.

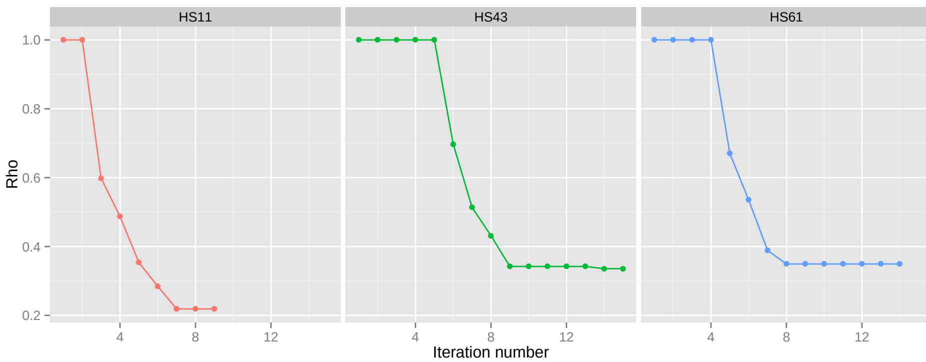

FIG. 1. *ρ values for problems* `hs`11*,* `hs`43*, and* `hs`61*.*

5.2. Infeasible test. As in [3], we modified the 126 CUTEr `hs` problems by adding bound constraints $x_1 \leq 0$ and $x_1 \geq 1$ to make all `hs` problems infeasible; we refer to these problems as `hs_inf`. All of the parameters used for this infeasible test set are the same as mentioned for the feasible test set, except we increase the maximum iteration limit for the subproblem solver to 20000. Defining the feasibility KKT error $\epsilon_{fea}(x)$ as

$$\epsilon_{fea}(x) := \max \left\{ \left\| \sum_{i=1}^{m} \eta_i \nabla c_i(x) \right\|_\infty , \left\| (e - \eta^{\mathcal{E}}) \circ (c_{\mathcal{E}}(x))_+ \right\|_\infty , \left\| (e + \eta^{\mathcal{E}}) \circ (c_{\mathcal{E}}(x))_- \right\|_\infty , \right.$$
$$\left. \left\| (e - \eta^{\mathcal{I}}) \circ (c_{\mathcal{I}}(x))_+ \right\|_\infty , \left\| \eta^{\mathcal{I}} \circ (c_{\mathcal{I}}(x))_- \right\|_\infty \right\},$$

we use the same stopping criteria as in [3], except that we do not necessarily need to drive ρ to 0; hence we drop "$\rho \leq 10^{-8}$" from the stopping criteria used in [3].

TABLE 2
Performance comparison of SQuID and the proposed algorithm on infeasible problems.

Problem type	Algorithm	Succeed	Fail	Total
Infeasible `hs` problems (`hs_inf`)	SQuID	111 (90.24%)	12 (9.76%)	123
	Proposed	116 (92.10%)	10 (7.90%)	126

For these experiments, we have the following observations.

- Out of 126 `hs_inf` problems, our algorithm successfully solved 116, which is a success rate of about 92% ≈ 116/126. Our proposed method also outperforms SQuID on infeasible problems. The comparison statistics[3] are shown in Table 2.

[3]The performance statistics for SQuID is obtained from [3], where the overall number of `hs` problems is 123 due to compiling errors.

- In a few cases, the Hessian modification strategy described above did not work well. For example, for problems `hs104_inf`, `hs114_inf`, `hs8_inf`, `hs23_inf`, and `hs93_inf`, convergence is observed when we increase the modification constant from 10^{-4} to 10^{-2}.

TABLE 3
CUTEr 13 large scale problems.

Problem	# constraints	# variables	# equalities
DTOC1NA	3996	5998	3996
DTOC1NB	3996	5998	3996
DTOC1ND	3996	5998	3996
EG3	20000	10001	1
GILBERT	1	5000	1
JANNSON4	2	10000	0
LUKVLE1	9998	10000	9998
LUKVLE10	9998	10000	9998
LUKVLE3	2	10000	2
LUKVLE6	4999	9999	4999
LUKVLI13	6664	9998	0
LUKVLI3	2	10000	0
LUKVLI6	4999	9999	0

5.3. Large scale test. We also applied our implementation to solve some large scale problems from the CUTEr test set; see Table 3. The parameter settings used were the same as those in section 5.1, except that we set the iteration limit for the subproblem solver to be 2000. We used L-BFGS for the Hessian approximations which pairs well with the coordinate descent algorithm giving a $O(n + \ell)$ total complexity for each coordinate update. Table 4 presents the results for successful runs. For the remaining problems not shown, the coordinate descent QP algorithm could not reach the desired accuracy within the maximum number of subproblem iterations. We leave further investigation into the most effective iterative QP solver for these problems to future work since this is beyond the scope of this paper.

TABLE 4
Test results on CUTEr 13 large scale problems.

Problem	# iter	# f	$f(x^*)$	$v(x^*)$	KKT	Final ρ	CPU (s)
DTOC1NA	13	13	4.138866e+00	2.2154e−06	1.8787e−05	0.751447	1.7
DTOC1NB	13	13	7.138849e+00	4.8350e−07	3.5507e−05	0.849347	1.3
DTOC1ND	14	19	4.760303e+01	1.7999e−07	4.8070e−05	0.815373	1.5
EG3	10	10	8.048306e−06	0.0000e+00	7.3171e−05	0.479603	49.9
GILBERT	74	74	2.459468e+03	2.1702e−08	4.0472e−06	0.024360	2.6
JANNSON4	79	80	9.801970e+03	6.9569e−08	1.8301e−05	0.009923	24.7
LUKVLE1	13	25	4.821043e−14	3.0876e−08	5.3643e−05	0.960000	1.9
LUKVLE10	191	191	3.534934e+03	2.2246e−09	9.7836e−05	0.282103	64.2
LUKVLE3	41	49	2.758658e+01	9.7477e−14	4.9497e−05	0.318856	1.7
LUKVLE6	39	68	6.286441e+05	1.4360e−12	6.9166e−05	0.360397	3.6
LUKVLI13	65	76	1.321855e+02	3.2121e−09	7.0523e−05	0.293858	9.0
LUKVLI3	70	78	1.157754e+01	9.0105e−13	6.6447e−05	0.442002	3.1
LUKVLI6	43	63	6.286441e+05	1.3907e−11	6.7668e−05	0.195366	28.6

To recognize the benefits of our proposed algorithm compared to an alternative approach, let us consider the CPU times required to run the experiments whose results are shown in Table 4 compared to the CPU times that would be required by SQuID from [3]. The aforementioned implementation of SQuID was not able to terminate successfully on any of the problems in Table 3 within ten minutes. The primary expense is solving the QP subproblems to high accuracy in each iteration. By contrast, the result shown in Table 4 that required the most CPU time was the run for problem `LUKVLE10`, where the entire run terminated in 64 seconds. The benefits of our proposed algorithm are clear when solving large scale problems. (On a contemporary laptop computer, the state-of-the-art code Ipopt [15] solves problem `LUKVLE10` in only a couple of seconds, but that code benefits from two decades of software development.)

6. 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 guarantee the effectiveness of our updating strategy under reasonable assumptions. The empirical effects of our strategy are 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.

REFERENCES

[1] J. V. Burke, *A Sequential quadratic programming algorithm for potentially infeasible mathematical programs*, J. Math. Anal. Appl., 139 (1989), pp. 319–351.

[2] J. V. Burke, *A robust trust region method for constrained nonlinear programming problems*, SIAM J. Optim., 2 (1992), pp. 325–347, https://doi.org/10.1137/0802016.

[3] J. V. Burke, F. E. Curtis, and H. Wang, *A sequential quadratic optimization algorithm with rapid infeasibility detection*, SIAM J. Optim., 24 (2014), pp. 839–872, https://doi.org/10.1137/120880045.

[4] J. V. Burke, F. E. Curtis, H. Wang, and J. Wang, *Inexact Sequential Quadratic Optimization with Penalty Parameter Updates Within the QP Solve: Extended Version*, https://arxiv.org/abs/1803.09224, 2018.

[5] J. V. Burke and S.-P. Han, *A robust sequential quadratic programming method*, Math. Programming, 43 (1989), pp. 277–303.

[6] R. H. Byrd, G. Lopez-Calva, and J. Nocedal, *A line search exact penalty method using steering rules*, Math. Programming, 133 (2012), pp. 39–73.

[7] R. H. Byrd, J. Nocedal, and R. A. Waltz, *Steering exact penalty methods for nonlinear programming*, Optim. Methods Softw., 23 (2008), pp. 197–213.

[8] F. E. Curtis, T. Johnson, D. P. Robinson, and A. Wächter, *An inexact sequential quadratic optimization algorithm for nonlinear optimization*, SIAM J. Optim., 24 (2014), pp. 1041–1074, https://doi.org/10.1137/130918320.

[9] S. P. Han, *A globally convergent method for nonlinear programming*, J. Optim. Theory Appl., 22 (1977), pp. 297–309.

[10] S. P. Han and O. L. Mangasarian, *Exact penalty functions in nonlinear programming*, Math. Programming, 17 (1979), pp. 251–269.

[11] W. Hock and K. Schittkowski, *Test examples for nonlinear programming codes*, J. Optim. Theory Appl., 30 (1980), pp. 127–129.

[12] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, 2nd ed., Springer, New York, 2006.

[13] M. J. D. Powell, *A fast algorithm for nonlinearly constrained optimization calculations*, in Numerical Analysis, Lecture Notes in Math. 630, Springer, Berlin, 1978, pp. 144–157.

[14] M. V. Solodov, *Constraint Qualifications*, in Wiley Encyclopedia of Operations Research and Management Science, John Wiley & Sons, London, 2010.

[15] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Programming, 106 (2006), pp. 25–57, https://doi.org/10.1007/s10107-004-0559-y.

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