SIAM J. OPTIM.
Vol. 20, No. 5, pp. 2281–2299

INFEASIBILITY DETECTION AND SQP METHODS FOR NONLINEAR OPTIMIZATION*

RICHARD H. BYRD†, FRANK E. CURTIS‡, AND JORGE NOCEDAL§

Abstract. This paper addresses the need for nonlinear programming algorithms that provide fast local convergence guarantees regardless of whether a problem is feasible or infeasible. We present a sequential quadratic programming method derived from an exact penalty approach that adjusts the penalty parameter automatically, when appropriate, to emphasize feasibility over optimality. The superlinear convergence of such an algorithm to an optimal solution is well known when a problem is feasible. The main contribution of this paper, however, is a set of conditions under which the superlinear convergence of the same type of algorithm to an infeasible stationary point can be guaranteed when a problem is infeasible. Numerical experiments illustrate the practical behavior of the method on feasible and infeasible problems.

Key words. nonlinear programming, constrained optimization, infeasibility

AMS subject classifications. 90C26, 90C30, 90C55

DOI. 10.1137/080738222

1. Introduction.

Constrained optimization algorithms are confronted with two tasks—the minimization of a function and the satisfaction of constraints. In cases when feasible points exist and one or more are optimal, a number of rapidly convergent methods can be employed. However, when many of these methods are applied to infeasible problem instances, the progress of the iteration can be very slow, and a great number of function evaluations are often required before a declaration of infeasibility can be made.

The primary focus of this paper is to address the need for optimization algorithms that can both efficiently solve feasible problems and rapidly detect when a given problem instance is infeasible. Fast detection of infeasibility has become increasingly important due to the central role it plays in branch-and-bound methods for mixed-integer nonlinear programming and in parametric studies of optimization models, but it is also a concern in its own right for general nonlinear programming problems that may include constraint incompatibilities.

One way to contend with the possibility that a problem could be infeasible is to employ a switch to decide whether the current iteration should seek a solution of the nonlinear program or, alternatively, a minimizer of some measure of infeasibility. Such an approach has been advocated by Fletcher and Leyffer [18] and has the benefit that infeasibility can be declared when a minimizer of the infeasibility measure is found that violates one or more constraints. The main difficulty faced by this type of

*Received by the editors October 16, 2008; accepted for publication (in revised form) March 6, 2010; published electronically May 21, 2010.
http://www.siam.org/journals/siopt/20-5/73822.html

†Department of Computer Science, University of Colorado, Boulder, CO 80309 (Richard.Byrd@colorado.edu). This author was supported by National Science Foundation grant CMMI 0728190.

‡Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015 (frank.e.curtis@gmail.com). This author was supported in part by National Science Foundation grant DMS 0602235 while at the Courant Institute of Mathematical Sciences, New York University, New York, NY.

§Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 (nocedal@eecs.northwestern.edu). This author was supported by National Science Foundation grant DMS-0810213 and by Department of Energy grant DE-FG02-87ER25047-A004.

approach, however, lies in the design of effective criteria for determining when such a switch should be made, since an inappropriate technique can lead to inefficiencies. In particular, since the objective function is ignored during iterations that care only about minimizing a measure of infeasibility, the iterates may stray from an optimal solution, thus delaying the optimization process.

In this paper we study an alternative approach involving a single optimization strategy. We show that it is effective for finding an optimal feasible solution (when one exists) or finding the minimizer of an infeasibility measure (when no feasible point exists). Our algorithm is an exact penalty method that uses the penalty parameter to emphasize infeasibility detection over optimization when appropriate. It belongs to the class of penalty sequential quadratic programming (SQP) methods proposed by Fletcher [17] that compute steps by minimizing a piecewise quadratic model of a penalty function subject to a linearization of the constraints. An important feature of this type of approach is that with an accurate estimate of the set of constraints satisfied as equalities at a solution point, the asymptotic convergence rate of the iteration is primarily influenced by the choice of penalty parameter. Thus, in this paper we pay careful attention to the procedure for updating the penalty parameter at every iteration, particularly for infeasible problem instances.

The problem of interest is formulated as

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \geq 0, \quad i \in \mathcal{I} = \{1,\ldots,t\}, \end{aligned} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and the $g_i : \mathbb{R}^n \to \mathbb{R}$ are smooth functions. Since the results and algorithms presented in this paper can be extended without difficulty to the case when the nonlinear program contains both equality and inequality constraints, we restrict our attention here, for simplicity, to problem (1.1). When feasible points of (1.1) do not exist, the algorithm should return a solution of the problem

$$\min_{x\in\mathbb{R}^n} v(x) \triangleq \sum_{i\in\mathcal{I}} \max\{-g_i(x), 0\}. \tag{1.2}$$

More specifically, we would like to design the optimization algorithm so that, when the problem is infeasible, the iterates converge quickly to an *infeasible stationary point* $\hat{x}$, which is defined as a stationary point of problem (1.2) such that $v(\hat{x}) > 0$. We say that problem (1.1) is *locally infeasible* if there is an infeasible stationary point for it.

Let us motivate the penalty approach studied in this paper by reviewing the properties of a basic SQP method when applied to infeasible problems. In a basic SQP approach, the search direction d_k is defined as the solution to the quadratic programming (QP) subproblem

$$\min_{d\in\mathbb{R}^n} \nabla f(x_k)^T d + \tfrac{1}{2} d^T \nabla^2_{xx} L(x_k,\lambda_k) d \tag{1.3a}$$

$$\text{s.t.} \quad g_i(x_k) + \nabla g_i(x_k)^T d \geq 0, \quad i \in \mathcal{I}, \tag{1.3b}$$

where $L(x,\lambda) \triangleq f(x) - \lambda^T g(x)$ is the Lagrangian of (1.1) and

$$\nabla^2_{xx} L(x,\lambda) = \nabla^2 f(x) - \sum_{i\in\mathcal{I}} \lambda_i \nabla^2 g_i(x)$$

is its corresponding Hessian matrix. Such a subproblem is known to be effective when it is feasible and when the constraints of (1.1) satisfy certain regularity properties.

However, it can be shown that at an infeasible stationary point $\hat{x}$, subproblem (1.3) is infeasible. Moreover, in a neighborhood of $\hat{x}$, the constraints (1.3b) are either inconsistent or ill-conditioned. Thus, if the SQP iterates $\{x_k\}$ approach $\hat{x}$, the steps $\{d_k\}$ may not be well defined, and even if they are, they will become extremely large in norm. In a line search framework, this will cause the steplengths to tend to zero, and in a trust region framework it will produce steps of insufficient descent, meaning that in either framework convergence to $\hat{x}$ can be extremely slow. In section 5 we provide an illustration of this kind of behavior.

The penalty SQP method proposed by Fletcher, otherwise known as an Sℓ_1QP method [16, 17] or elastic SQP method [19, 4], was designed to overcome the difficulties posed by incompatibility of the constraints (1.3b). The subproblem in such a penalty SQP method is, in fact, always feasible, meaning that the search direction is always well defined. What the method lacks, however, are fast local convergence guarantees when (1.1) is infeasible. The goal of this paper is to provide such guarantees.

The regularization benefits of exact penalty methods have been recognized for a long time; see, for example, [35, 29, 12, 17, 15, 21, 11] and the literature on optimization problems with complementarity constraints [27, 34, 1, 25, 26]. For example, the SNOPT software package [19] reverts to an ℓ_1 penalty approach when the Lagrange multipliers are deemed too large or when the quadratic subproblem is inconsistent. However, in contrast to the algorithm proposed in this paper, the algorithm in SNOPT employs a switch as previously described; it starts as a regular SQP method based on the subproblem (1.3) and invokes a penalty approach only if difficulties are encountered. In addition, SNOPT makes no attempt to achieve a fast rate of convergence to stationary points in the infeasible case.

We note at the outset that infeasibility detection is difficult in the nonconvex case. Indeed, for a nonconvex problem, infeasibility detection has many of the difficulties inherent in global optimization since, even if an algorithm identifies an infeasible point where constraint violations are locally minimized, there may exist feasible points in other regions of the search space. Nevertheless, the techniques discussed in this paper can be used in conjunction with global optimization methods [33, 24] to determine if a problem is in fact *globally infeasible*.

The paper is organized into five sections. In section 2 we motivate and present a general form of our penalty SQP method. In section 3 we show that this type of approach yields fast convergence guarantees to infeasible stationary points with certain properties when the penalty parameter is handled appropriately. In section 4 we describe a practical technique for updating the penalty parameter, and in section 5 we present the results of some numerical experiments. Throughout the paper, $\|\cdot\|$ denotes any vector or vector-induced norm. We generally use the subscript i to denote the element number of a vector and a subscript k or $k+1$ to denote the iteration number of an algorithm.

2. A penalty SQP framework.

In the penalty SQP method proposed by Fletcher [16, 17], each search direction is computed as the solution of the subproblem

$$\min_{d\in\mathbb{R}^n,\, s\in\mathbb{R}^t} \quad \rho[\nabla f(x_k)^T d + \tfrac{1}{2} d^T \nabla_{xx}^2 L(x_k,\lambda_k) d] + \sum_{i\in\mathcal{I}} s_i \tag{2.1a}$$

$$\text{s.t.} \quad g_i(x_k) + \nabla g_i(x_k)^T d + s_i \ge 0,\ \ s_i \ge 0, \quad i \in \mathcal{I}, \tag{2.1b}$$

where $\rho \ge 0$ is a penalty parameter and s is a vector of slack variables. This subproblem can be viewed as a relative of (1.3) where the constraints have been relaxed (so that they are always feasible), and the balance between the nonlinear objective and constraint terms can be manipulated through the parameter ρ. The algorithm we describe in this section, which is designed to provide fast convergence when (1.1) is feasible or infeasible, can be formulated in terms of an algorithm that solves (2.1) to compute search directions. However, such an algorithm would require adjustments in the multipliers λ_k when changes to the penalty parameter are made. Thus, we prefer an alternative formulation of the penalty SQP method in which ρ is the only control parameter. We derive our formulation as follows.

Consider the minimization of the ℓ_1 exact penalty function

$$\phi(x;\rho) \triangleq \rho f(x) + v(x),$$

where $v(x)$ is defined in (1.2). As is well known, if the penalty parameter ρ is sufficiently small, stationary points of the nonlinear program (1.1) are also stationary points of the penalty function ϕ; see, e.g., [23]. Moreover, the problem of minimizing ϕ can be written as the smooth constrained problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n,\, r\in\mathbb{R}^t} \ & \rho f(x) + \sum_{i\in\mathcal{I}} r_i \\ \text{s.t. } & g_i(x) + r_i \geq 0, \quad r_i \geq 0, \quad i \in \mathcal{I}, \end{aligned} \tag{2.2}$$

where r are slack variables. (Note that if $\rho = 0$, then (2.2) is equivalent to (1.2).) A basic SQP subproblem for (2.2) is equivalent to

$$\min_{d\in\mathbb{R}^n,\, s\in\mathbb{R}^t} \quad \rho_k \nabla f(x_k)^T d + \tfrac{1}{2} d^T W(x_k,\lambda_k;\rho_k) d + \sum_{i\in\mathcal{I}} s_i \tag{2.3a}$$

$$\text{s.t.} \quad g_i(x_k) + \nabla g_i(x_k)^T d + s_i \geq 0, \quad i \in \mathcal{I}, \tag{2.3b}$$

$$s_i \geq 0, \quad i \in \mathcal{I}, \tag{2.3c}$$

where s is again a vector of slack variables and $W(x_k,\lambda_k;\rho_k)$ is the Hessian matrix

$$W(x_k,\lambda_k;\rho_k) \triangleq \rho_k \nabla^2 f(x_k) - \sum_{i\in\mathcal{I}} \lambda_k^i \nabla^2 g_i(x_k). \tag{2.4}$$

An alternative motivation for (2.3) is that it is equivalent to minimizing the local model

$$\begin{aligned} q_k(d;\rho_k) \triangleq \ & \rho_k \nabla f(x_k)^T d + \tfrac{1}{2} d^T W(x_k,\lambda_k;\rho_k) d \\ & + \sum_{i\in\mathcal{I}} \max\{-g_i(x_k) - \nabla g_i(x_k)^T d, 0\} \end{aligned} \tag{2.5}$$

of the penalty function $\phi(\cdot;\rho_k)$ about x_k. Note that the last term in $q_k(d;\rho_k)$ may be viewed as a piecewise linear approximation to $v(x)$.

Subproblem (2.3) is the focal point of our approach. Note that it differs from (2.1) only in the choice of Hessian matrices in their quadratic objective functions. In (2.1), the penalty parameter multiplies the entire Hessian matrix, while in (2.3) it multiplies the Hessian of the objective and not the term involving second derivatives of the constraints. The structured Hessian (2.4) enables us to achieve rapid convergence by controlling only the penalty parameter ρ.

A general form of our approach is presented as Algorithm I below. For concreteness, we describe a line search algorithm with a merit function given by $\phi(x;\rho)$, but since in our local analysis we assume that the steplength is always $\alpha_k = 1$ (see section 3), our results apply equally to a trust region approach. With a solution d_k to problem (2.3), the iterate is updated as

$$x_{k+1} \leftarrow x_k + \alpha_k d_k,$$

where α_k is a steplength parameter that ensures sufficient reduction in $\phi(\cdot;\rho_k)$, and the new dual variables are given by

$$\lambda_{k+1} \leftarrow \text{optimal multipliers associated with the constraints (2.3b).} \tag{2.6}$$

Algorithm I: Penalty SQP Method.
Initialize: (x_0, λ_0), $\tau \in (0,1)$, and $\eta \in (0,1)$.
For $k = 0, 1, 2, \ldots$, or until x_k solves either (1.1) or (1.2):

1. Determine an appropriate value ρ_k for the penalty parameter.
2. Compute the Hessian $W(x_k, \lambda_k; \rho_k)$ given by (2.4).
3. Compute (d_k, λ_{k+1}) by solving the subproblem (2.3).
4. Let $0 < \alpha_k \le 1$ be the first member of the sequence $\{1, \tau, \tau^2, \ldots\}$ such that

$$\phi(x_k;\rho_k) - \phi(x_k + \alpha_k d_k;\rho_k) \ge \eta\alpha_k[q_k(0;\rho_k) - q_k(d_k;\rho_k)].$$

5. Let $x_{k+1} \leftarrow x_k + \alpha_k d_k$.

Our remaining task in the presentation of our approach is to design a mechanism for setting the penalty parameter ρ_k during each iteration. We confront this issue in section 4 after considering the local behavior of Algorithm I in the next section.

3. Fast convergence to infeasible stationary points. In this section, we illustrate the local convergence properties of Algorithm I under certain common conditions. Since such properties for penalty SQP algorithms applied to feasible problems have been studied elsewhere [17], we focus our analysis here on the infeasible case. Our interests are in asymptotic rate of convergence results, and therefore, it is convenient to assume that the steplength is $\alpha_k = 1$. In doing so, we assume that a mechanism, such as a watchdog technique [10] or a second-order correction [17], is employed to ensure that unit steplengths are accepted by the merit function $\phi(x;\rho)$.

3.1. Local analysis. Let x^ρ denote a first-order optimal solution of problem (2.2) for a given ρ. Then, there exist slack variables r^ρ and Lagrange multipliers $(\lambda^\rho, \sigma^\rho)$ such that $(x^\rho, \lambda^\rho, r^\rho, \sigma^\rho)$ satisfies the KKT system

$$\rho\nabla f(x) - \sum_{i\in\mathcal{I}} \lambda_i \nabla g_i(x) = 0, \tag{3.1a}$$

$$1 - \lambda_i - \sigma_i = 0, \quad i \in \mathcal{I}, \tag{3.1b}$$

$$\lambda_i(g_i(x) + r_i) = 0, \quad i \in \mathcal{I}, \tag{3.1c}$$

$$\sigma_i r_i = 0, \quad i \in \mathcal{I}, \tag{3.1d}$$

$$g_i(x) + r_i \ge 0, \quad i \in \mathcal{I}, \tag{3.1e}$$

$$r, \lambda, \sigma \ge 0. \tag{3.1f}$$

We note, in particular, that if for $\rho > 0$ such a solution has $r^\rho = 0$, then x^ρ is a first-order optimal solution of the nonlinear program (1.1).

We provide an alternative way of characterizing solutions of the penalty problem (2.2) in the following well-known result; see, e.g., [17].

LEMMA 3.1. *Suppose that* $(x^\rho, \lambda^\rho, r^\rho, \sigma^\rho)$ *is a primal-dual KKT point for problem* (2.2) *and that the strict complementarity conditions*

$$r_i^\rho + \sigma_i^\rho > 0, \quad \lambda_i^\rho + (g_i(x^\rho) + r_i^\rho) > 0 \tag{3.2}$$

hold for all $i \in \mathcal{I}$. *Then* (x^ρ, λ^ρ) *satisfies the system*

$$\rho \nabla f(x) - \sum_{i \in \mathcal{I}} \lambda_i \nabla g_i(x) = 0, \tag{3.3a}$$

$$\textit{and either} \quad g_i(x) < 0 \textit{ and } \lambda_i = 1, \textit{ or} \tag{3.3b}$$

$$g_i(x) > 0 \textit{ and } \lambda_i = 0, \textit{ or} \tag{3.3c}$$

$$g_i(x) = 0 \textit{ and } \lambda_i \in (0,1). \tag{3.3d}$$

Conversely, if (x, λ) *satisfies* (3.3), *then it also satisfies* (3.1) *together with* $r_i = \max(0, -g_i(x))$ *and* $\sigma_i = 1 - \lambda_i$.

Throughout this section, it will be useful to distinguish between three distinct sets of constraint indices defined with respect to a given point x. Specifically, at x we define the sets of *active*, *violated*, and *strictly satisfied* constraints, respectively, as

$$\mathcal{A}(x) \triangleq \{i \in \mathcal{I} : g_i(x) = 0\}; \qquad \mathcal{V}(x) \triangleq \{i \in \mathcal{I} : g_i(x) < 0\}; \\ \mathcal{S}(x) \triangleq \{i \in \mathcal{I} : g_i(x) > 0\}. \tag{3.4}$$

Let us now characterize stationary points for the infeasibility measure v defined in (1.2). Note that problem (1.2) can be recast as (2.2) for $\rho = 0$. Let us denote a stationary point of problem (2.2), for $\rho = 0$, as $\hat{x}$, i.e.,

$$\hat{x} \triangleq x^{\rho=0}.$$

Such a point satisfies conditions (3.1) (with $\rho = 0$), for some nonnegative vectors $\hat{\lambda}$, $\hat{r}$, and $\hat{\sigma}$. If we have that $\hat{r} \neq 0$, then one can show that $g_i(\hat{x}) < 0$ for some i, and therefore $\hat{x}$ is an infeasible stationary point of the nonlinear program (1.1).

Another convenient way of describing an infeasible stationary point $\hat{x}$ is to note that it is a first-order optimal solution of the auxiliary problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & \sum_{i \in \hat{\mathcal{V}}} -g_i(x) \\ \text{s.t. } & g_i(x) = 0, \quad i \in \hat{\mathcal{A}}, \end{aligned} \tag{3.5}$$

where from now on $\hat{\mathcal{A}}$ is shortand for $\mathcal{A}(\hat{x})$, and similarly for $\hat{\mathcal{V}}$ and $\hat{\mathcal{S}}$. To verify this claim, we note by Lemma 3.1 that since the conditions (3.1) can be recast as (3.3), then if $(\hat{x}, \hat{\lambda})$ solves (3.1) for $\rho = 0$, we have that $\hat{\lambda}_i \in (0,1)$ for $i \in \hat{\mathcal{A}}$, $\hat{\lambda}_i = 1$ for $i \in \hat{\mathcal{V}}$, and $\hat{\lambda}_i = 0$ for $i \in \hat{\mathcal{S}}$. As a result, the pair $(\hat{x}, \hat{\lambda})$ satisfies the system

$$F(x, \lambda_{\hat{\mathcal{A}}}, \rho) \triangleq \begin{bmatrix} \rho \nabla f(x) - \sum_{i \in \hat{\mathcal{A}}} \lambda_i \nabla g_i(x) - \sum_{i \in \hat{\mathcal{V}}} \nabla g_i(x) \\ g_{\hat{\mathcal{A}}}(x) \end{bmatrix} = 0 \tag{3.6}$$

for $\rho = 0$, where we have defined the subvectors

$$\lambda_{\hat{\mathcal{A}}} \triangleq [\lambda_i]_{i \in \hat{\mathcal{A}}} \quad \text{and} \quad g_{\hat{\mathcal{A}}} \triangleq [g_i]_{i \in \hat{\mathcal{A}}}.$$

Since (3.6) (with $\rho = 0$) are the KKT conditions of problem (3.5), we have verified that $\hat{x}$ is a first-order optimal point for (3.5).

There are various types of infeasible stationary points. Some are strict isolated local minimizers of the infeasibility measure $v(x)$, while others may belong to a set of minimizers of v or may simply be stationary points of v. Formulation (3.5) is useful in identifying those stationary points for which we are able to establish a fast rate of convergence. In our analysis, we require that $\hat{x}$ is a point satisfying second-order sufficiency for problem (3.5). Our complete set of assumptions is as follows.

Assumptions 3.2. The point $\hat{x}$ is an infeasible stationary point of the feasibility problem (1.2), $\hat{\lambda}$ is a vector of Lagrange multipliers such that $(\hat{x}, \hat{\lambda})$ solves (3.3), and the following conditions hold:

(a) *Smoothness.* The functions f and g are twice continuously differentiable in an open convex set containing $\hat{x}$.

(b) *Regularity.* The Jacobian of active constraints $\nabla g_{\hat{\mathcal{A}}}(\hat{x})^T$ has full row rank, where

$$\nabla g_{\hat{\mathcal{A}}}(x)^T \triangleq [\nabla g_i(x)]_{i\in\hat{\mathcal{A}}}. \tag{3.7}$$

(c) *Strict complementarity.* The multipliers satisfy $\hat{\lambda}_i \in (0,1)$ for $i \in \hat{\mathcal{A}}$.

(d) *Second-order sufficiency.* For $\rho = 0$, the Hessian matrix defined in (2.4) satisfies

$$d^T W(\hat{x}, \hat{\lambda}; 0) d = -d^T \left[\sum_{i\in\hat{\mathcal{A}}} \hat{\lambda}_i \nabla^2 g_i(\hat{x}) + \sum_{i\in\hat{\mathcal{V}}} \nabla^2 g_i(\hat{x})\right] d > 0$$

for all $d \neq 0$ such that $\nabla g_{\hat{\mathcal{A}}}(\hat{x})^T d = 0$.

Our approach for establishing superlinear convergence to an infeasible stationary point $\hat{x}$ is as follows. Let z_k denote the current primal-dual pair (x_k, λ_k) of Algorithm I and $z_{k+1} = (x_{k+1}, \lambda_{k+1})$ the next iterate. Also, recall that we have defined z^ρ as a first-order optimal solution of the penalty problem (2.2) for a given value of the penalty parameter ρ. In a series of lemmas we show that, if z_k is close to $\hat{z}$ and ρ is small, then

$$\begin{aligned}\|z_{k+1} - \hat{z}\| &\leq \|z_{k+1} - z^\rho\| + \|z^\rho - \hat{z}\| \\ &\leq C_1 \|z_k - z^\rho\|^2 + O(\rho) \\ &\leq C_2 \|z_k - \hat{z}\|^2 + O(\rho)\end{aligned}$$

for some constants $C_1, C_2 > 0$ independent of ρ. Thus, to achieve superlinear convergence, we need only ensure that, as the iterates approach an infeasible stationary point $\hat{x}$, Algorithm I decreases ρ fast enough.

Our first result quantifies the distance between x^ρ and $\hat{x}$ as a function of ρ.

LEMMA 3.3. *Suppose that Assumptions* 3.2 *are satisfied. Then, for all ρ sufficiently small, the penalty problem* (2.2) *has a solution x^ρ with the same sets of active, violated, and strictly satisfied constraints as $\hat{x}$ (i.e., $\mathcal{A}(x^\rho) = \hat{\mathcal{A}}$, $\mathcal{V}(x^\rho) = \hat{\mathcal{V}}$, and $\mathcal{S}(x^\rho) = \hat{\mathcal{S}}$), and we have that*

$$\lambda_i^\rho \in (0,1),\ i \in \hat{\mathcal{A}}, \quad \lambda_i^\rho = 1,\ i \in \hat{\mathcal{V}}, \quad \lambda_i^\rho = 0,\ i \in \hat{\mathcal{S}}, \tag{3.8}$$

and

$$\left\|\begin{bmatrix} x^\rho - \hat{x} \\ \lambda^\rho - \hat{\lambda}\end{bmatrix}\right\| \leq C\rho \tag{3.9}$$

for some constant $C > 0$ independent of ρ.

Proof. We have shown above that if $(\hat{x}, \hat{\lambda})$ is a KKT point for problem (2.2) with $\rho = 0$, then it satisfies system (3.6) with $\rho = 0$. Under Assumptions 3.2, F is a continuously differentiable mapping about the point $(\hat{x}, \hat{\lambda}_{\hat{\mathcal{A}}}, 0)$ that satisfies $F(\hat{x}, \hat{\lambda}_{\hat{\mathcal{A}}}, 0) = 0$. Differentiating F, we find

$$F'(x, \lambda_{\hat{\mathcal{A}}}, \rho) \triangleq \frac{\partial F(x, \lambda_{\hat{\mathcal{A}}}, \rho)}{\partial (x, \lambda_{\hat{\mathcal{A}}})} = \begin{bmatrix} G(x, \lambda_{\hat{\mathcal{A}}}, \rho) & -\nabla g_{\hat{\mathcal{A}}}(x) \\ \nabla g_{\hat{\mathcal{A}}}(x)^T & 0 \end{bmatrix}, \tag{3.10}$$

where

$$G(x, \lambda_{\hat{\mathcal{A}}}, \rho) \triangleq \rho \nabla^2 f(x) - \sum_{i \in \hat{\mathcal{A}}} \lambda_i \nabla^2 g_i(x) - \sum_{i \in \hat{\mathcal{V}}} \nabla^2 g_i(x). \tag{3.11}$$

By Assumptions 3.2(b) and (d), the matrix (3.10) is nonsingular at $(\hat{x}, \hat{\lambda}_{\hat{\mathcal{A}}}, 0)$. We can thus apply the implicit function theorem [32, Theorem 9.28] and state that there exist open sets $\mathcal{N}_x \in \mathbb{R}^n$, $\mathcal{N}_\lambda \in \mathbb{R}^{|\hat{\mathcal{A}}|}$, and $\mathcal{N}_\rho \in \mathbb{R}$ containing $\hat{x}$, $\hat{\lambda}_{\hat{\mathcal{A}}}$, and 0, respectively, and continuously differentiable functions $\bar{x} : \mathcal{N}_\rho \to \mathcal{N}_x$ and $\bar{\lambda}_{\hat{\mathcal{A}}} : \mathcal{N}_\rho \to \mathcal{N}_\lambda$ such that

$$\bar{x}(0) = \hat{x}, \quad \bar{\lambda}_{\hat{\mathcal{A}}}(0) = \hat{\lambda}_{\hat{\mathcal{A}}}, \quad \text{and} \quad F(\bar{x}(\rho), \bar{\lambda}_{\hat{\mathcal{A}}}(\rho), \rho) = 0 \ \text{ for all } \ \rho \in \mathcal{N}_\rho. \tag{3.12}$$

By the second equation in (3.6), and since the inequalities in the definitions (3.4) of $\hat{\mathcal{V}}$ and $\hat{\mathcal{S}}$ are strict, we have that for ρ sufficiently small

$$g_i(\bar{x}(\rho)) = 0, \ i \in \hat{\mathcal{A}}, \quad g_i(\bar{x}(\rho)) > 0, \ i \in \hat{\mathcal{S}}, \quad g_i(\bar{x}(\rho)) < 0, \ i \in \hat{\mathcal{V}}. \tag{3.13}$$

Also, since $\hat{\lambda}_{\hat{\mathcal{A}}} \in (0,1)$, we have that $\bar{\lambda}_{\hat{\mathcal{A}}}(\rho) \in (0,1)$ for sufficiently small ρ. If we define

$$\bar{\lambda}_i(\rho) = 1, \ i \in \hat{\mathcal{V}}, \quad \text{and} \quad \bar{\lambda}_i(\rho) = 0, \ i \in \hat{\mathcal{S}}, \tag{3.14}$$

then it follows that $(\bar{x}(\rho), \bar{\lambda}(\rho))$ satisfies (3.3). Lemma 3.1 then implies that $(\bar{x}(\rho), \bar{\lambda}(\rho))$ satisfies (3.1) (together with $\bar{r}(\rho) = \max(0, -g(\bar{x}(\rho))$ and $\bar{\sigma}(\rho) = 1 - \bar{\lambda}(\rho))$ and is therefore a first-order optimal point for the penalty problem (2.2) for small ρ. Thus we can write $(x^\rho, \lambda^\rho) = (\bar{x}(\rho), \bar{\lambda}(\rho))$ for small ρ, and by (3.13), we have that x^ρ has the same sets of active, violated, and strictly satisfied constraints as $\hat{x}$. This proves the first part of the lemma.

We now establish the bound (3.9). From the differentiability of the functions $x^\rho = \bar{x}(\rho)$ and $\lambda^\rho_{\hat{\mathcal{A}}} = \bar{\lambda}(\rho)$ and the expressions for their derivatives given by the implicit function theorem, we have for ρ sufficiently small

$$\begin{bmatrix} x^\rho \\ \lambda^\rho_{\hat{\mathcal{A}}} \end{bmatrix} = \begin{bmatrix} \hat{x} \\ \hat{\lambda}_{\hat{\mathcal{A}}} \end{bmatrix} - F'_{x,\lambda_{\mathcal{A}}}(\hat{x}, \hat{\lambda}_{\mathcal{A}}, 0)^{-1} F'_\rho(\hat{x}, \hat{\lambda}_{\mathcal{A}}, 0)\rho + o(\rho). \tag{3.15}$$

Therefore, we have that (3.9) is satisfied. ∎

We now describe an asymptotic property of the steps generated by the penalty SQP method. Note that we do not assume that $W(x_k, \lambda_k; \rho_k)$ is positive definite; Assumption 3.2(d) states only that $W(\hat{x}, \hat{\lambda}; 0)$ is positive definite on the tangent space of the active constraints. Therefore, the QP (2.3) could have several solutions. In the following result, we show that at least one of these solutions has certain desirable properties. (This nonuniqueness issue is discussed further in section 3.3.)

LEMMA 3.4. *Suppose that Assumptions* 3.2 *hold. Then, there exist* $\hat{\rho} > 0$ *and* $\hat{\epsilon} > 0$ *such that, for all* $\rho_k \in [0, \hat{\rho}]$ *and* (x_k, λ_k) *satisfying* $\|(x_k, \lambda_k) - (\hat{x}, \hat{\lambda})\| \leq \hat{\epsilon}$, *there is a local solution* d_k *of the SQP subproblem* (2.3) *that yields the same sets of active, violated, and strictly satisfied constraints as* $\hat{x}$ *and can be obtained via a solution to the linear system*

$$\begin{bmatrix} W(x_k, \lambda_k; \rho_k) & -\nabla g_{\hat{\mathcal{A}}}(x_k) \\ \nabla g_{\hat{\mathcal{A}}}(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} d \\ [\lambda_{k+1}]_{\hat{\mathcal{A}}} \end{bmatrix} = - \begin{bmatrix} \rho_k \nabla f(x_k) - \sum_{i \in \hat{\mathcal{V}}} \nabla g_i(x_k) \\ g_{\hat{\mathcal{A}}}(x_k) \end{bmatrix}. \tag{3.16}$$

Moreover, the multipliers in (2.6) *corresponding to the solution* d_k *satisfy*

$$[\lambda_{k+1}]_i \in (0,1),\ i \in \hat{\mathcal{A}}, \quad [\lambda_{k+1}]_i = 1,\ i \in \hat{\mathcal{V}}, \quad [\lambda_{k+1}]_i = 0,\ i \in \hat{\mathcal{S}}. \tag{3.17}$$

Proof. The first-order optimality conditions for the SQP subproblem (2.3), at some point (x, λ) and for some value ρ, are

$$\begin{aligned} W(x, \lambda; \rho) d - \sum_{i \in \mathcal{I}} \gamma_i \nabla g_i(x) &= -\rho \nabla f(x), \\ 1 - \gamma_i - \omega_i &= 0, \quad i \in \mathcal{I}, \\ \gamma_i (g_i(x) + \nabla g_i(x)^T d + s_i) &= 0, \quad i \in \mathcal{I}, \\ \omega_i s_i &= 0, \quad i \in \mathcal{I}, \\ g_i(x) + \nabla g_i(x)^T d + s_i &\geq 0, \quad i \in \mathcal{I}, \\ s, \gamma, \omega &\geq 0, \end{aligned} \tag{3.18}$$

where (γ, ω) are Lagrange multipliers. Note that if we set $d = 0$, then this system is identical to (3.1), and since $(\hat{x}, \hat{\lambda})$ satisfies (3.1) for $\rho = 0$, it follows that system (3.18) for $\rho = 0$ is solved at $(x, \lambda) = (\hat{x}, \hat{\lambda})$ by $(d, \hat{\gamma}) = (0, \hat{\lambda})$ and $(\hat{\omega}, \hat{s}) = (\hat{\sigma}, \hat{r})$. By Lemma 3.1, we have that

$$\hat{\lambda}_i \in (0,1),\ i \in \hat{\mathcal{A}}, \quad \hat{\lambda}_i = 1,\ i \in \hat{\mathcal{V}}, \quad \hat{\lambda}_i = 0,\ i \in \hat{\mathcal{S}}, \tag{3.19}$$

and hence by (3.1a) (with $\rho = 0$)

$$\sum_{i \in \hat{\mathcal{A}}} \hat{\gamma}_i \nabla g_i(\hat{x}) + \sum_{i \in \hat{\mathcal{V}}} \nabla g_i(\hat{x}) = 0.$$

Therefore, for $\rho = 0$, the linear system

$$\begin{bmatrix} W(x, \lambda; \rho) & -\nabla g_{\hat{\mathcal{A}}}(x) \\ \nabla g_{\hat{\mathcal{A}}}(x)^T & 0 \end{bmatrix} \begin{bmatrix} d \\ \gamma_{\hat{\mathcal{A}}} \end{bmatrix} = - \begin{bmatrix} \rho \nabla f(x) - \sum_{i \in \hat{\mathcal{V}}} \nabla g_i(x) \\ g_{\hat{\mathcal{A}}}(x) \end{bmatrix} \tag{3.20}$$

is satisfied at $(x, \lambda) = (\hat{x}, \hat{\lambda})$ by $(d, \gamma_{\hat{\mathcal{A}}}) = (0, \hat{\lambda}_{\hat{\mathcal{A}}})$. Moreover, under Assumptions 3.2, the matrix in (3.20) is nonsingular at $(\hat{x}, \hat{\lambda})$ for $\rho = 0$, and hence the solution of (3.20) varies continuously in a neighborhood of $(\hat{x}, \hat{\lambda}, 0)$. We also have that $W(x, \lambda; \rho)$ is positive definite on the null space of $\nabla g_{\hat{\mathcal{A}}}(x)^T$ in a neighborhood of $(\hat{x}, \hat{\lambda}, 0)$.

It follows from these observations that for all (x, λ) sufficiently close to $(\hat{x}, \hat{\lambda})$ and for ρ sufficiently small, the solution $(d, \gamma_{\hat{\mathcal{A}}})$ to (3.20) is close to $(0, \hat{\lambda}_{\hat{\mathcal{A}}})$ and therefore satisfies

$$\begin{aligned} \gamma_i \in (0,1) &\text{ for } i \in \hat{\mathcal{A}}, \\ g_i(x) + \nabla g_i(x)^T d < 0 &\text{ for } i \in \hat{\mathcal{V}}, \\ g_i(x) + \nabla g_i(x)^T d > 0 &\text{ for } i \in \hat{\mathcal{S}}. \end{aligned} \tag{3.21}$$

By construction, such a solution $(d, \gamma_{\hat{\mathcal{A}}})$ satisfies (3.20) and therefore satisfies (3.18) together with $\gamma_i = 1$ for $i \in \hat{\mathcal{V}}$, $\gamma_i = 0$ for $i \in \hat{\mathcal{S}}$, $w_i = 1 - \gamma_i$, and $s_i = \max(0, -g_i(x) - \nabla g_i(x)^T d)$. Therefore, (d, γ) is a KKT point of the SQP subproblem (2.3) (in fact, by Assumption 3.2(d) it is a minimizer of that problem), and by (3.21) it has the same sets of active, violated, and strictly satisfied constraints as $\hat{x}$. ∎

Lemma 3.4 shows that, near a stationary point, the SQP step d_k is given by the system (3.16), and by conditions (3.17) we have that for all iterates close to $(\hat{x}, \hat{\lambda})$, the multiplier estimates generated by Algorithm I satisfy

$$[\lambda_k]_i \in (0,1),\ i \in \hat{\mathcal{A}}, \quad [\lambda_k]_i = 1,\ i \in \hat{\mathcal{V}}, \quad [\lambda_k]_i = 0,\ i \in \hat{\mathcal{S}}. \tag{3.22}$$

Based on these observations, we study the effect of one step of the algorithm on the penalty problem (2.2) when ρ is small. We show that quadratic convergence of a pure Newton iteration to the solution of each penalty problem is uniform with respect to ρ. (This result is similar to that proved by Cores and Tapia in Lemma 2.1 of [13].)

LEMMA 3.5. *Let (x^ρ, λ^ρ) denote a primal-dual pair satisfying the KKT conditions* (3.1) *for the penalty problem* (2.2). *Then, for ρ sufficiently small and for all (x_k, λ_k) sufficiently close to $(\hat{x}, \hat{\lambda})$ with λ_k satisfying* (3.22), *the iterates generated by Algorithm* I *satisfy*

$$\left\| \begin{bmatrix} x_{k+1} - x^\rho \\ \lambda_{k+1} - \lambda^\rho \end{bmatrix} \right\| \le C' \left\| \begin{bmatrix} x_k - x^\rho \\ \lambda_k - \lambda^\rho \end{bmatrix} \right\|^2 \tag{3.23}$$

for some constant C' independent of ρ.

Proof. By Lemma 3.4, if (x_k, λ_k) is sufficiently close to $(\hat{x}, \hat{\lambda})$ and ρ is sufficiently small, the step d_k generated by Algorithm I is obtained via a solution of the system (3.16). Moreover, since $W(x_k, \lambda_k; \rho_k)$ is given by (2.4), and since conditions (3.22) are satisfied, we have from (3.11) that $W(x_k, \lambda_k; \rho_k) = G(x_k, \lambda_{\hat{\mathcal{A}}}, \rho_k)$. Therefore, system (3.16) constitutes the Newton iteration applied to the nonlinear system $F(x, \lambda_{\hat{\mathcal{A}}}, \rho) = 0$ for fixed ρ, where F is defined in (3.6).

We can now apply standard Newton analysis (see, for example, [14]). By Assumption 3.2(a) we have that F', given in (3.10), is continuously differentiable, and hence F' is Lipschitz continuous in a neighborhood of $(\hat{x}, \hat{\lambda})$ for each ρ. Moreover, since ρ is bounded, this Lipschitz constant, which we denote by κ_1, is independent of ρ. Next, by Assumptions 3.2(b) and (d), the matrix F' is nonsingular at $(\hat{x}, \hat{\lambda}, 0)$, and hence its inverse exists and is bounded in norm by a constant κ_2 in a neighborhood of that point. By Theorem 5.2.1 of [14], if (x_k, λ_k) satisfies

$$\left\| \begin{bmatrix} x_k - x^\rho \\ \lambda_k - \lambda^\rho \end{bmatrix} \right\| \le \frac{1}{2\kappa_1\kappa_2}, \tag{3.24}$$

then we have

$$\left\| \begin{bmatrix} x_{k+1} - x^\rho \\ \lambda_{k+1} - \lambda^\rho \end{bmatrix} \right\| \le \kappa_1\kappa_2 \left\| \begin{bmatrix} x_k - x^\rho \\ \lambda_k - \lambda^\rho \end{bmatrix} \right\|^2. \tag{3.25}$$

(This inequality contains all components of x and λ, and not just those in the active set $\hat{\mathcal{A}}$, because from (3.8) and (3.22) we have that $[x_k]_i = x_i^\rho$ and $[\lambda_k]_i = \lambda_i^\rho$ for $i \in \hat{\mathcal{V}} \cup \hat{\mathcal{S}}$.)

Finally, if ρ is sufficiently small that (x^ρ, λ^ρ) satisfies

$$\left\| \begin{bmatrix} x^\rho - \hat{x} \\ \lambda^\rho - \hat{\lambda} \end{bmatrix} \right\| \le \frac{1}{4\kappa_1\kappa_2}$$

and (x_k, λ_k) is chosen so that

$$\left\| \begin{bmatrix} x_k - \hat{x} \\ \lambda_k - \hat{\lambda} \end{bmatrix} \right\| \leq \frac{1}{4\kappa_1\kappa_2},$$

then (3.24) is satisfied. ◻

We are now ready to prove the main result of this section—quadratic or superlinear convergence to an infeasible stationary point. Recall that $z = (x, \lambda)$ denotes a primal-dual pair.

THEOREM 3.6. *Suppose that $\hat{x}$ is an infeasible stationary point for the nonlinear program* (1.1) *and that Assumptions* 3.2 *hold. If an iterate $z_k = (x_k, \lambda_k)$ is sufficiently close to $\hat{z} = (\hat{x}, \hat{\lambda})$ and the penalty parameter is selected so that $\rho_k = O(||z_k - \hat{z}||^2)$, then the sequence $\{z_k\}$ in Algorithm* I *converges to $\hat{z}$ quadratically. If, instead, $\rho_k = o(||z_k - \hat{z}||)$, then the convergence rate is superlinear.*

Proof. If $\rho_k = O(\|z_k - \hat{z}\|^2)$ or $\rho_k = o(\|z_k - \hat{z}\|)$, then z_k sufficiently close to $\hat{z}$ implies that both $\|z_k - \hat{z}\|$ and ρ_k are sufficiently small to satisfy the conditions of Lemmas 3.3, 3.4, and 3.5. Defining $z_{k+1} = (x_{k+1}, \lambda_{k+1})$ and $z^{\rho_k} = (x^{\rho_k}, \lambda^{\rho_k})$, we may apply Lemmas 3.3 and 3.5 to the inequality

$$\|z_{k+1} - \hat{z}\| \leq \|z_{k+1} - z^{\rho_k}\| + \|z^{\rho_k} - \hat{z}\|$$

to obtain

$$||z_{k+1} - \hat{z}|| \leq C'||z_k - z^{\rho_k}||^2 + C\rho_k. \tag{3.26}$$

Then,

$$\begin{aligned} ||z_k - z^{\rho_k}||^2 &\leq (||z_k - \hat{z}|| + ||\hat{z} - z^{\rho_k}||)^2 \\ &\leq (||z_k - \hat{z}|| + C\rho_k)^2 \\ &= ||z_k - \hat{z}||^2 + C^2\rho_k^2 + 2C||z_k - \hat{z}||\rho_k. \end{aligned}$$

If $\rho = O(||z_k - \hat{z}||^2)$, then this relation and (3.26) imply that the iteration is locally quadratically convergent, while if $\rho = o(||z_k - \hat{z}||)$, then the rate of convergence is superlinear. ◻

In summary, by using the structured Hessian approximation (2.4), a penalty SQP method will achieve a fast rate of convergence simply by driving the penalty parameter to zero fast enough in a neighborhood of an infeasible stationary point. Of course, a practical algorithm must also be efficient in the feasible case, and a strategy for controling ρ in both cases is an essential part of a general purpose algorithm. In section 4 we present one such strategy. Before doing so, we discuss two issues related to the convergence analysis just presented.

3.2. Problems with n active constraints. An interesting case to consider is when n constraints are active at the infeasible stationary point $\hat{x}$, where n is the number of variables. In this case, Lemma 3.4 implies that for all ρ smaller than a threshold value $\hat{\rho} > 0$, the SQP subproblem (2.3) identifies all n constraints as active. The full rank assumption then implies that the solution of $g_{\mathcal{A}}(x) = 0$ is locally unique and must be $\hat{x}$. Thus, in this case, the algorithm can declare that the problem is locally infeasible without driving the penalty parameter ρ to zero.

3.3. Positive definiteness of the Hessian. The analysis we have presented assumes that the exact Hessian $W_k \triangleq W(x_k, \lambda_k; \rho_k)$ defined in (2.4) is used to form the quadratic model $q_k(d; \rho_k)$. In general, however, this Hessian will not be positive definite in $\mathbb{R}^n$. We can only expect that, in a neighborhood of an infeasible stationary point, the Hessian $W(x_k, \lambda_k; 0)$ is positive definite on the null space of $\nabla g_{\mathcal{A}}(\hat{x})^T$.

If W_k is indefinite, then the algorithm requires the solution of an indefinite QP that may have several local minimizers. The results presented above apply provided the QP solver finds the minimizer satisfying Lemma 3.4. One way to achieve this property near an infeasible stationary point $\hat{x}$ is to find the QP minimizer that is closest to the current iterate x_k.

The situation becomes simpler if W_k is defined to be positive definite over all of $\mathbb{R}^n$, since in this case the QP (2.3) has a unique solution. Our analysis then applies directly without giving further consideration to the solution obtained by the QP solver. (Note that W_k will be positive definite in $\mathbb{R}^n$ if the problem is convex.) Otherwise, we could modify or redefine the Hessian to ensure this positive definiteness property, but this must be done carefully to avoid interfering with fast convergence.

Suppose that a positive definite approximate Hessian W_k is employed. Since in a neighborhood of $\hat{x}$ our approach is equivalent to applying an SQP method to the equality constrained problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & \rho f(x) + \sum_{i \in \hat{\mathcal{V}}} -g_i(x) \\ \text{s.t. } & g_i(x) = 0, \quad i \in \hat{\mathcal{A}}, \end{aligned} \tag{3.27}$$

whose Lagrangian Hessian is G (see (3.11)), we can apply the characterization result for superlinear convergence given in [3, 30]. It states that this iteration is superlinear if and only if the modified Hessian W_k satisfies the condition

$$\frac{\| P_k (W_k - G(\hat{x}, \hat{\lambda}_{\hat{\mathcal{A}}}, \rho)) \|}{\| x_{k+1} - x_k \|} \to 0, \tag{3.28}$$

where P_k denotes the $n \times n$ orthogonal projection onto the null space of $\nabla g_{\hat{\mathcal{A}}}^T(x_k)$. One way to achieve this limit is to define W_k as the augmented Lagrangian Hessian

$$\rho_k \nabla^2 f(x_k) - \sum_{i \in \mathcal{I}} [\lambda_k]_i \nabla^2 g_i(x_k) + \sigma \nabla g_{\hat{\mathcal{A}}}(x_k) \nabla g_{\hat{\mathcal{A}}}(x_k)^T$$

for a sufficiently large parameter $\sigma > 0$. In addition, quasi-Newton updating methods have been developed that generate positive definite approximations to the Hessian of the augmented Lagrangian that yield R-superlinear convergence; see [31, 8].

Thus, although the discussion in this paper has focused on the exact Hessian as defined in (2.4), a variety of other practical methods will converge rapidly to minimizers of infeasibility that satisfy Assumptions 3.2.

4. A practical algorithm. In the previous section, we have provided conditions on the sequence of penalty parameters in Algorithm I that ensure a fast rate of convergence in the infeasible case. In practice, however, the penalty parameter update must also ensure global convergence from remote starting points and, preferably, fast local convergence to optimal solutions. In this section we present an instance of Algorithm I that accomplishes all of these goals.

In our approach, the penalty parameter update follows the spirit of the *steering rules* that are reviewed in [6] and incorporated into a line search method in [9]. Algorithm II, given below, modifies these rules so that they conform with the requirements of the analysis presented in section 3. For ease of exposition, we define the linear model of the constraints appearing in (2.5) as

$$m_k(d) \triangleq \sum_{i\in\mathcal{I}} \max\{-g_i(x_k) - \nabla g_i(x_k)^T d, 0\},$$

so that the quadratic model of the penalty function can be written as

$$q_k(d;\rho) = \rho\nabla f(x_k)^T d + \tfrac{1}{2}d^T W(x_k,\lambda_k;\rho)d + m_k(d). \tag{4.1}$$

In addition, we define a measure for the KKT error, with respect to (3.3), as

$$E_k(\lambda,\rho) \triangleq \left\| \rho\nabla f(x_k) - \sum_{i\in\mathcal{I}} \lambda_i \nabla g_i(x_k)\right\|_1 + \sum_{i\in\mathcal{S}_k} |g_i(x_k)\lambda_i| + \sum_{i\in\mathcal{V}_k} |g_i(x_k)(1-\lambda_i)|. \tag{4.2}$$

Here, $\mathcal{S}_k$ and $\mathcal{V}_k$ are the sets of strictly satisfied and violated constraints evaluated at x_k.

Algorithm II: Penalty SQP Method with Penalty Parameter Update Strategy.

Initialize (x_0,λ_0), $\rho_0 > 0$, and all of τ, η, ϵ_1, and ϵ_2 in $(0,1)$.

For $k = 0,1,2,\ldots$, or until x_k solves (1.1) or (1.2):

a. Solve (2.3) to obtain (d_k,λ_{k+1}).

b. If $m_k(d_k) = 0$, then go to step g.

c. Solve (2.3) with $\rho_k = 0$ to obtain $(\bar{d}_k,\bar{\lambda}_{k+1})$.

d. Decrease ρ_k until the solution (d_k,λ_{k+1}) to (2.3) satisfies

$$\begin{cases} m_k(d_k) = 0 & \text{if } m_k(\bar{d}_k) = 0,\\ m_k(0) - m_k(d_k) \geq \epsilon_1(m_k(0) - m_k(\bar{d}_k)) & \text{otherwise.}\end{cases} \tag{4.3}$$

e. Further decrease ρ_k, if necessary, until the solution (d_k,λ_{k+1}) to (2.3) satisfies

$$q_k(0;\rho_k) - q_k(d_k;\rho_k) \geq \epsilon_2(q_k(0;0) - q_k(\bar{d}_k;0)). \tag{4.4}$$

f. Set $\rho_k = \min\{\rho_k, [E_k(\bar{\lambda}_{k+1},0)/\max\{1,v(x_k)\}]^2\}$. If ρ_k has decreased, then solve (2.3) to obtain (d_k,λ_{k+1}).

g. Let $\alpha_k \in (0,1]$ be the first member of the sequence $\{1,\tau,\tau^2,\ldots\}$ such that

$$\phi(x_k;\rho_k) - \phi(x_k+\alpha_k d_k;\rho_k) \geq \eta\alpha_k[q_k(0;\rho_k) - q_k(d_k;\rho_k)]. \tag{4.5}$$

h. Let $x_{k+1} \leftarrow x_k + \alpha_k d_k$ and $\rho_{k+1} \leftarrow \rho_k$.

The essential effects of this strategy are the following. First, if the SQP subproblem for the most recent value of the penalty parameter yields a linearly feasible solution (i.e., $m(d_k) = 0$), then we follow this direction. This is to ensure that regularization of the constraints does not ruin the progress of the algorithm when it is not needed. Second, if the SQP solution is linearly infeasible, then we decrease the penalty parameter sufficiently so that the new direction provides sufficient progress in linearized feasibility. This is to ensure that the algorithm does not diverge from minimizers of the infeasibility measure. The condition (4.4) ensures that as the algorithm progresses, the penalty parameter is small enough so that minimizers of $\phi(x;\rho)$

correspond to minimizers of the infeasibility measure v. Finally, by reducing ρ_k sufficiently in step f when the optimality error for the feasibility problem $E_k(\bar{\lambda}_{k+1}, 0)$ is small, Theorem 3.6 states that the algorithm can converge quadratically.

The steering rules described in [6] were designed to ensure global convergence (even in the infeasible case) and have proved to be effective in practice [5, 9] but were not designed to yield a fast rate of convergence in the infeasible case. The conditions in Algorithm II, on the other hand, may be regarded as alternative steering rules that are designed to be efficient for both feasible and infeasible problems. In particular, the original steering rules were based on subproblem (1.3), meaning that the subproblem is linear when $\rho = 0$ (i.e., during the computation of $\bar{d}_k$). Alternatively, the strategy described in Algorithm II can be seen as an extension of the steering rules that maintains the Hessian matrix for the constraints in the subproblem even for varying values of ρ.

A possible drawback of Algorithm II is that multiple QP subproblems may need to be solved during each iteration. However, our numerical evidence has shown that the steering rules only occasionally require more than one QP solve on well-behaved problems and rarely require more than two or three QP solves on badly scaled or infeasible problems, and this additional cost is more than compensated for by a savings in the total number of iterations. A numerical exploration of the performance of our penalty SQP method on a standard collection of feasible and infeasible problems is outside the scope of this paper as it requires a sophisticated software implementation of the algorithm. For our purposes, we simply present Algorithm II as one that yields good performance on the problems presented in the next section and fits into the framework of Algorithm I for ensuring fast local convergence.

We close this section by mentioning that other techniques have been proposed in the literature for updating the penalty parameter; see, e.g., [20, 2, 28, 11]. In those techniques, however, the update is based on the behavior of the algorithm over several iterations, while the steering rules proposed here and in [6] explore the properties of the problem at a fixed iterate in order to update ρ.

5. Numerical tests. We developed a prototype MATLAB implementation of Algorithm II in order to observe its performance on some illustrative examples. In the code, we ensure that W_k is positive definite by adding a multiple of the identity matrix to the matrix (2.4) so that $q_k(d;\rho)$ has a unique minimizer for each $\rho \in [0, \rho_{k-1}]$. The QP (2.3) is solved using the MATLAB `quadprog` routine, and the chosen input parameters are given as follows: $\rho_0 = 1$, $\tau = 0.5$, $\eta = 10^{-8}$, and $\epsilon_1 = \epsilon_2 = 0.1$.

Our code terminates when either of the following conditions holds:

$$E_k(\lambda_k, \rho_k) \leq 10^{-6} \quad \text{and} \quad v_k \leq 10^{-6} \quad \text{(optimality)}$$

or

$$E_k(\lambda_k, 0) \leq 10^{-6} \quad \text{and} \quad v_k > 10^{-6} \quad \text{(infeasibility)}.$$

Note that we use λ_k instead of $\bar{\lambda}_k$ when computing $E_k(\cdot, 0)$ since $\bar{\lambda}_k$ is available only if the algorithm performed step c in the previous iteration. However, either choice is valid (when $\bar{\lambda}_k$ is available) when checking for infeasibility.

Example 1. We refer to the following problem as `unique`:

$$\begin{aligned} &\min\ x_1 + x_2 \\ &\text{s.t. } x_2 - x_1^2 - 1 \geq 0, \\ &\qquad 0.3(1 - e^{x_2}) \geq 0. \end{aligned} \tag{5.1}$$

This problem is infeasible and the point $\hat{x} = (0, 1)$ is a strict minimizer of the infeasibility measure v at which Assumptions 3.2 are satisfied. The second constraint simply imposes the bound $x_2 \leq 0$, but we have written it as a nonlinear expression; the weight 0.3 is chosen so that only the first constraint is active at $\hat{x}$. Running our algorithm on this problem, starting from the point $(3, 2)$, yields the results given in Table 1. We report the iteration number k, the function value f_k, the value of the infeasibility measure v_k, the KKT (optimality) error at the start of the iteration $E_k(\lambda_k, \rho_k)$, the KKT (feasibility) error $E_k(\lambda_k, 0)$, the value of the penalty parameter ρ_k used to compute the search direction (i.e., the value when step g is reached), the norm of the search direction $\|d_k\|$, and the steplength α_k from (4.5).

Table 1
Output for Example 1 *(*`unique`*). QPs solved:* 24.

k	f_k	v_k	$E_k(\lambda_k, \rho_k)$	$E_k(\lambda_k, 0)$	ρ_k	$\|\|d_k\|\|$	α_k
0	+5.00e+00	9.92e+00	8.84e+00	7.51e+00	1.00e+00	1.71e+00	1.00e+00
1	+2.66e+00	2.82e+00	3.51e+00	2.23e+00	1.00e-02	1.16e+00	1.00e+00
2	+1.04e+00	1.06e+00	9.14e-01	9.07e-01	1.00e-02	1.13e+00	5.00e-01
3	+3.20e-01	8.23e-01	4.55e-01	4.53e-01	1.00e-02	7.76e-01	1.00e+00
4	+6.74e-01	6.09e-01	4.33e-01	4.38e-01	1.00e-02	4.66e-01	1.00e+00
5	+9.72e-01	5.81e-01	2.84e-01	2.73e-01	1.00e-02	2.29e-01	1.00e+00
6	+9.39e-01	5.22e-01	4.52e-02	5.49e-02	1.00e-02	3.90e-02	1.00e+00
7	+9.94e-01	5.16e-01	1.66e-03	8.46e-03	2.69e-04	4.93e-03	1.00e+00
8	+1.00e+00	5.15e-01	9.17e-05	3.61e-04	4.91e-07	2.20e-04	1.00e+00
9	+1.00e+00	5.15e-01	1.18e-07	6.08e-07	--------	--------	--------

We mentioned at the end of section 1 that algorithms that impose strict linearizations of the constraints in the step computation are likely to be inefficient in a neighborhood of infeasible stationary points. An example of such a method is the line search interior point method implemented in the KNITRO/DIRECT solver [7]. For problem `unique`, this method requires 42 iterations and 151 function evaluations to declare the problem locally infeasible. As can be seen from the large ratio of function evaluations to iterations, the step has to be cut considerably, particularly as the iteration approaches the infeasible stationary point.

Example 2. Problem `robot` is from the CUTEr collection [22]. We ran the version with 14 variables. The original model has three equality constraints (call them c_1, c_2, and c_3), but we added the additional constraint

$$c_4(x) = c_1^2(x) + 1$$

so that the resulting model is infeasible. The results from our code are given in Table 2.

Example 3. We refer to the following problem as `isolated`:

$$\begin{aligned} &\min\; x_1 + x_2 \\ &\text{s.t. } -x_1^2 + x_2 - 1 \geq 0, \\ &\quad -x_1^2 - x_2 - 1 \geq 0, \\ &\quad x_1 - x_2^2 - 1 \geq 0, \\ &\quad -x_1 - x_2^2 - 1 \geq 0. \end{aligned} \tag{5.2}$$

This problem is infeasible, and the point $\hat{x} = (0, 0)$ is a strict minimizer of the infeasibility measure v where none of the constraints are active. Running our algorithm with the starting point $(3, 2)$ yields the results in Table 3.

Table 2
Output for Example 2 (`robot`). *QPs solved:* 34.

k	f_k	v_k	$E_k(\lambda_k, \rho_k)$	$E_k(\lambda_k, 0)$	ρ_k	$\|\|d_k\|\|$	α_k
0	+0.00e+00	1.12e+01	1.00e+00	4.00e+00	1.00e+00	1.25e+00	1.00e+00
1	+1.56e+00	5.06e+00	1.70e+00	2.80e+00	1.00e+00	5.79e-01	1.00e+00
2	+2.12e+00	2.89e+00	3.34e-01	7.62e-01	1.00e-01	7.39e-01	1.00e+00
3	+4.19e+00	1.42e+00	5.13e-01	6.49e-01	1.00e-01	3.25e-01	1.00e+00
4	+4.75e+00	1.19e+00	2.95e-01	5.05e-01	1.00e-01	1.83e-01	1.00e+00
5	+5.20e+00	1.10e+00	1.30e-01	4.49e-01	1.00e-01	1.10e-01	1.00e+00
6	+5.45e+00	1.06e+00	5.96e-02	3.67e-01	1.00e-02	2.04e+00	6.25e-02
7	+5.50e+00	1.05e+00	2.37e-01	2.54e-01	1.00e-02	1.77e-01	1.00e+00
8	+6.01e+00	1.01e+00	1.20e-01	1.42e-01	1.00e-02	2.44e-01	5.00e-01
9	+6.09e+00	1.01e+00	5.65e-02	7.93e-02	1.00e-03	9.37e-02	1.00e+00
10	+6.19e+00	1.01e+00	9.57e-03	6.76e-03	4.57e-05	2.72e-02	1.00e+00
11	+6.15e+00	1.00e+00	2.79e-04	1.59e-04	2.52e-08	2.81e-03	1.00e+00
12	+6.15e+00	1.00e+00	7.53e-06	7.46e-06	5.56e-11	3.17e-05	1.00e+00
13	+6.15e+00	1.00e+00	3.96e-10	2.48e-10	--------	--------	--------

Table 3
Output for Example 3 (`isolated`). *QPs solved:* 20.

k	f_k	v_k	$E_k(\lambda_k, \rho_k)$	$E_k(\lambda_k, 0)$	ρ_k	$\|\|d_k\|\|$	α_k
0	+5.00e+00	3.00e+01	1.15e+01	1.15e+01	1.00e+00	2.34e+00	1.00e+00
1	+1.70e+00	7.63e+00	4.49e+00	4.15e+00	1.00e+00	2.22e+00	1.00e+00
2	-1.44e+00	7.15e+00	3.09e+00	4.62e+00	1.00e+00	1.05e+00	1.00e+00
3	-1.85e-01	4.07e+00	1.00e+00	7.38e-01	1.00e-01	2.54e-01	1.00e+00
4	+4.32e-02	4.01e+00	3.73e-01	2.73e-01	1.00e-02	7.42e-02	1.00e+00
5	-5.00e-03	4.00e+00	7.69e-16	1.00e-02	1.00e-04	3.50e-03	1.00e+00
6	-5.00e-05	4.00e+00	1.57e-15	1.00e-04	1.00e-08	3.54e-05	1.00e+00
7	-5.00e-09	4.00e+00	6.16e-16	1.00e-08	--------	--------	--------

Example 4. In this example, we simulate a situation that occurs in nonlinear branch-and-bound methods for mixed-integer nonlinear programming. Problem `batch`, from the CUTEr collection, is a problem with 46 variables that has feasible solutions, at least one of which is locally optimal. Running this problem with our code, we find that our penalty parameter updating strategy does not impede the progress of the algorithm when applied to a feasible problem. In fact, the method performs quite well, quadratic convergence is obtained, and the penalty parameter is kept constant in the last iterations; see Table 4. (The final value of ρ is small because the optimal objective value f is on the order of 10^5.)

Adding the single bound constraint $\mathrm{tl}[1] \geq 5$ to problem `batch`, however, makes the problem infeasible. We refer to this resulting problem as `batch1`. Running this problem with our implementation, using the same initial conditions as when problem `batch` is solved, yields the results in Table 5.

Example 5. We refer to our final example problem as `nactive`:

$$
\begin{aligned}
&\min\ x_1 \\
&\text{s.t. } \tfrac{1}{2}(-x_1 - x_2^2 - 1) \geq 0, \\
&\qquad x_1 - x_2^2 \geq 0, \\
&\qquad -x_1 + x_2^2 \geq 0.
\end{aligned}
\tag{5.3}
$$

This problem is infeasible with $n = 2$ active constraints at the minimizer of infeasibility $\hat{x} = (0, 0)$. We ran this problem with the starting point $(-20, 10)$, and the results are provided in Table 6.

TABLE 4
Output for Example 4(a) *(*`batch`*). QPs solved:* 28.

k	f_k	v_k	$E_k(\lambda_k, \rho_k)$	$E_k(\lambda_k, 0)$	ρ_k	$\|\|d_k\|\|$	α_k
0	+6.00e+02	4.70e+02	2.43e+02	2.27e+02	1.00e+00	2.48e+01	1.00e+00
1	+2.36e+02	2.45e+02	6.35e+01	8.24e+01	1.00e+00	2.89e+00	1.00e+00
2	+1.04e+02	2.24e+02	1.03e+01	2.76e+01	1.00e-01	7.69e+00	5.00e-01
3	+4.42e+02	1.58e+02	3.21e+00	1.63e+01	1.00e-02	1.10e+01	5.00e-01
4	+2.86e+03	8.21e+01	2.64e+00	1.00e+01	1.00e-03	5.64e+00	1.00e+00
5	+2.97e+04	1.05e+01	3.42e+00	4.83e+00	1.00e-04	3.05e+00	1.00e+00
6	+5.69e+04	3.08e+00	4.90e-01	2.53e+00	1.00e-04	1.66e+00	1.00e+00
7	+6.28e+04	1.89e+00	4.16e-02	1.65e+00	1.00e-05	1.50e+00	1.00e+00
8	+9.33e+04	2.88e-01	2.50e-01	3.44e-01	1.00e-05	9.55e-01	1.00e+00
9	+1.03e+05	2.07e-02	1.86e-02	2.44e-02	1.00e-05	2.17e-01	1.00e+00
10	+1.04e+05	4.24e-04	3.59e-04	4.99e-04	1.00e-05	1.54e-03	1.00e+00
11	+1.04e+05	1.31e-07	7.42e-08	--------	--------	--------	--------

TABLE 5
Output for Example 4(b) *(*`batch1`*). QPs solved:* 40.

k	f_k	v_k	$E_k(\lambda_k, \rho_k)$	$E_k(\lambda_k, 0)$	ρ_k	$\|\|d_k\|\|$	α_k
0	+6.00e+02	4.75e+02	2.43e+02	2.25e+02	1.00e+00	2.85e+01	1.00e+00
1	+2.36e+02	2.46e+02	6.25e+01	8.13e+01	1.00e+00	3.38e+00	1.00e+00
2	+1.04e+02	2.25e+02	1.03e+01	2.84e+01	1.00e-01	1.03e+01	5.00e-01
3	+4.35e+02	1.61e+02	3.35e+00	1.78e+01	1.00e-02	1.33e+01	5.00e-01
4	+2.85e+03	8.52e+01	2.75e+00	1.09e+01	1.00e-03	6.41e+00	1.00e+00
5	+3.01e+04	1.39e+01	3.55e+00	3.58e+00	1.00e-04	3.30e+00	1.00e+00
6	+5.72e+04	6.13e+00	5.00e-01	1.78e+00	1.00e-04	1.25e+00	1.00e+00
7	+6.35e+04	4.94e+00	3.71e-02	1.29e+00	1.00e-05	1.45e+00	1.00e+00
8	+9.40e+04	3.68e+00	1.70e-01	5.17e-01	1.00e-05	6.01e-01	1.00e+00
9	+1.09e+05	3.45e+00	1.58e-02	4.72e-01	1.00e-06	1.68e+00	1.00e+00
10	+1.63e+05	3.24e+00	1.22e-01	1.55e-01	1.00e-06	1.65e+00	1.00e+00
11	+2.27e+05	3.09e+00	3.92e-02	3.41e-02	1.00e-06	3.68e-01	1.00e+00
12	+2.27e+05	3.07e+00	3.87e-04	4.32e-02	1.00e-07	7.74e-01	1.00e+00
13	+2.66e+05	3.04e+00	6.76e-03	6.79e-03	1.00e-07	2.24e-01	1.00e+00
14	+2.66e+05	3.04e+00	1.19e-05	1.20e-05	1.44e-10	1.30e-01	1.00e+00
15	+2.66e+05	3.04e+00	7.82e-09	6.02e-09	--------	--------	--------

TABLE 6
Output for Example 5 *(*`nactive`*). QPs solved:* 17.

k	f_k	v_k	$E_k(\lambda_k, \rho_k)$	$E_k(\lambda_k, 0)$	ρ_k	$\|\|d_k\|\|$	α_k
0	-2.00e+01	1.60e+02	6.05e+01	1.15e+02	1.00e-01	2.01e+01	1.00e+00
1	-5.00e-01	3.79e+01	2.37e+01	2.51e+01	1.00e-01	1.47e+01	1.00e+00
2	-1.47e+01	1.57e+01	1.43e+01	1.53e+01	1.00e-01	1.37e+01	1.00e+00
3	-1.03e+00	1.03e+00	4.12e-01	5.15e-01	1.00e-01	1.03e+00	1.00e+00
4	-1.61e-10	5.00e-01	6.09e-05	5.54e-05	3.06e-09	2.52e-05	1.00e+00
5	-6.27e-10	5.00e-01	5.03e-06	5.03e-06	2.53e-11	2.52e-06	1.00e+00
6	-6.33e-12	5.00e-01	3.21e-12	3.21e-12	--------	--------	--------

By the commentary at the end of section 3, the algorithm would have converged to the infeasible stationary point $\hat{x}$ even if the penalty parameter was not driven to zero. However, our method for ensuring fast convergence, namely, step f of Algorithm II, always forces ρ to zero on infeasible problems, which is apparent in Table 6.

To verify that it is not necessary to drive ρ to zero for this problem, we disabled step f of Algorithm II and report the output in Table 7. Note that ρ is never changed and that quadratic convergence is obtained. In practice, it may be advantageous to identify the cases when the algorithm is converging to an infeasible stationary point

Table 7
Output for Example 5 *(*`nactive` *with modified* ρ *update). QPs solved:* 13.

k	f_k	v_k	$E_k(\lambda_k, \rho_k)$	$E_k(\lambda_k, 0)$	ρ_k	$\|\|d_k\|\|$	α_k
0	-2.00e+01	1.60e+02	6.05e+01	1.15e+02	1.00e-01	2.01e+01	1.00e+00
1	-5.00e-01	3.79e+01	2.37e+01	2.51e+01	1.00e-01	1.47e+01	1.00e+00
2	-1.47e+01	1.57e+01	1.43e+01	1.53e+01	1.00e-01	1.37e+01	1.00e+00
3	-1.03e+00	1.03e+00	4.12e-01	5.15e-01	1.00e-01	1.03e+00	1.00e+00
4	-1.61e-10	5.00e-01	6.09e-05	5.54e-05	1.00e-01	2.77e-05	1.00e+00
5	-7.66e-10	5.00e-01	3.07e-10	3.84e-10	--------	--------	--------

where n or more constraints are active and not impose the quadratic decrease in ρ, but for general-purpose implementations the advantage of carefully driving $\rho \to 0$ is illustrated by all of the experiments in this section.

Acknowledgments. The authors would like to thank Roger Fletcher for having stressed throughout the years the need to build fast infeasibility detection capabilities in optimization algorithms. They also thank the referees for several suggestions that improved the presentation.

REFERENCES

[1] M. Anitescu, *Global convergence of an elastic mode approach for a class of mathematical programs with complementarity constraints*, SIAM J. Optim., 16 (2005), pp. 120–145.

[2] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[3] P. T. Boggs, J. W. Tolle, and P. Wang, *On the local convergence of quasi-Newton methods for constrained optimization*, SIAM J. Control Optim., 20 (1982), pp. 161–171.

[4] E. G. Boman, *Infeasibility and Negative Curvature in Optimization*, Ph.D. thesis, Scientific Computing and Computational Mathematics, Stanford University, Palo Alto, CA, 1999.

[5] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz, *An algorithm for nonlinear optimization using linear programming and equality constrained subproblems*, Math. Program. Ser. B, 100 (2004), pp. 27–48.

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

[7] R. H. Byrd, J. Nocedal, and R. A. Waltz, *KNITRO: An integrated package for nonlinear optimization*, in Large-Scale Nonlinear Optimization, G. di Pillo and M. Roma, eds., Springer, New York, 2006, pp. 35–59.

[8] R. H. Byrd, R. A. Tapia, and Y. Zhang, *An SQP augmented Lagrangian BFGS algorithm for constrained optimization*, SIAM J. Optim., 2 (1992), pp. 210–241.

[9] R. H. Byrd, J. Nocedal, and G. López-Calva, *A Line Search Exact Penalty Method Using Steering Rules*, Technical report, Optimization Center, Northwestern University, Evanston, IL, 2009.

[10] R. M. Chamberlain, M. J. D. Powell, C. Lemaréchal, and H. C. Pedersen, *The watchdog technique for forcing convergence in algorithms for constrained optimization*, Math. Programming Stud., 16 (1982), pp. 1–17.

[11] L. Chen and D. Goldfarb, *Interior-point ℓ_2 penalty methods for nonlinear programming with strong global convergence properties*, Math. Program., 108 (2006), pp. 1–36.

[12] T. F. Coleman and A. R. Conn, *Second-order conditions for an exact penalty function*, Math. Programming, 19 (1980), pp. 178–185.

[13] D. Cores and R. A. Tapia, *Perturbation lemma for the Newton method with application to the SQP Newton method*, J. Optim. Theory Appl., 97 (1998), pp. 271–280.

[14] J. E. Dennis, Jr., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983 (reprinted as Classics Appl. Math. 16, SIAM, Philadelphia, 1996).

[15] J.-P. Dussault, *Numerical stability and efficiency of penalty algorithms*, SIAM J. Numer. Anal., 32 (1995), pp. 296–317.

[16] R. Fletcher, *An ℓ_1 penalty method for nonlinear constraints*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 26–40.

[17] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley and Sons, Chichester, UK, 1987.
[18] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.
[19] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM J. Optim., 12 (2002), pp. 979–1006.
[20] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.
[21] N. I. M. Gould, D. Orban, and Ph. Toint, *An Interior-Point L1-Penalty Method for Nonlinear Optimization*, Technical report RAL-TR-2003-022, Rutherford Appleton Laboratory, Chilton, Oxfordshire, UK, 2003.
[22] N. I. M. Gould, D. Orban, and Ph. L. Toint, CUTEr *and* sifdec*: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.
[23] S. P. Han and O. L. Mangasarian, *Exact penalty functions in nonlinear programming*, Math. Programming, 17 (1979), pp. 251–269.
[24] R. Horst and H. Tuy, *Global Optimization*, 3rd ed., Springer, New York, 1996.
[25] X. M. Hu and D. Ralph, *Convergence of a penalty method for mathematical programming with complementarity constraints*, J. Optim. Theory Appl., 123 (2004), pp. 365–390.
[26] S. Leyffer, G. López-Calva, and J. Nocedal, *Interior methods for mathematical programs with complementarity constraints*, SIAM J. Optim., 17 (2006), pp. 52–77.
[27] O. L. Mangasarian and J. S. Pang, *Exact penalty functions for mathematical programs with linear complementarity constraints*, Optimization, 42 (1997), pp. 1–8.
[28] M. Mongeau and A. Sartenaer, *Automatic decrease of the penalty parameter in exact penalty function methods*, European J. Oper. Res., 83 (1995), pp. 686–699.
[29] T. Pietrzykowski, *An exact potential method for constrained maxima*, SIAM J. Numer. Anal., 6 (1969), pp. 299–304.
[30] M. J. D. Powell, *Variable metric methods for constrained optimization*, in Mathematical Programming: The State of the Art (Bonn, 1982), A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 288–311.
[31] M. J. D. Powell, *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, in Nonlinear Programming 3, O. Mangasarian, R. Meyer, and S. Robinson, eds., Academic Press, London, 1978, pp. 27–63.
[32] W. Rudin, *Principles of Mathematical Analysis*, McGraw–Hill, New York, 1976.
[33] N. V. Sahinidis and M. Tawarmalani, *BARON* 7.2.5*: Global Optimization of Mixed-Integer Nonlinear Programs, User's Manual*, 2005.
[34] S. Scholtes and M. Stöhr, *Exact penalization of mathematical programs with equilibrium constraints*, SIAM J. Control Optim., 37 (1999), pp. 617–652.
[35] W. I. Zangwill, *Non-linear programming via penalty functions*, Management Sci., 13 (1967), pp. 344–358.