SIAM J. SCI. COMPUT.
Vol. 22, No. 2, pp. 517–534

A ROBUST ALGORITHM FOR OPTIMIZATION WITH GENERAL EQUALITY AND INEQUALITY CONSTRAINTS*

XIN-WEI LIU† AND YA-XIANG YUAN†

Abstract. An algorithm for general nonlinearly constrained optimization is presented, which solves an unconstrained piecewise quadratic subproblem and a quadratic programming subproblem at each iterate. The algorithm is robust since it can circumvent the difficulties associated with the possible inconsistency of QP subproblem of the original SQP method. Moreover, the algorithm can converge to a point which satisfies a certain first-order necessary optimality condition even when the original problem is itself infeasible, which is a feature of Burke and Han's methods [*Math. Programming*, 43 (1989), pp. 277–303]. Unlike Burke and Han's methods, our algorithm does not introduce additional bound constraints. The algorithm solves the same subproblems as the Han–Powell SQP algorithm at feasible points of the original problem. Under certain assumptions, it is shown that the algorithm coincides with the Han–Powell method when the iterates are sufficiently close to the solution. Some global convergence results are proved and locally superlinear convergence results are also obtained. Preliminary numerical results are reported.

Key words. SQP algorithm, constrained optimization, convergence

AMS subject classifications. 65K10, 95C30, 90C45

PII. S1064827598334861

1. Introduction.

We consider the optimization problem with general equality and inequality constraints

$$\min f(x) \tag{1.1}$$

$$\text{subject to (s.t.) } c_i(x) = 0, \quad i \in E, \tag{1.2}$$

$$c_i(x) \geq 0, \quad i \in I, \tag{1.3}$$

where $f(x) : R^n \to R$ and $c_i(x) : R^n \to R(i \in E \cup I)$ are continuously differentiable functions, $E = \{1, 2, \ldots, m_e\}$, $I = \{m_e + 1, \ldots, m\}$, m_e and m are two positive integers, and $m \geq m_e$.

SQP algorithms for constrained optimization are iterative and generate a new approximate to the solution by the procedure

$$x^+ = x + sd, \tag{1.4}$$

where x is the current point, d is a search direction which minimizes a quadratic model subject to linearized constraints, and s is the step-length along the direction such as [8, 13, 21]. For $k \geq 1$, the original SQP method developed by Wilson, Han, and Powell solves the following QP subproblem

$$\min g_k^T d + \frac{1}{2} d^T B_k d \tag{1.5}$$

$$\text{s.t. } c_i(x_k) + \nabla c_i(x_k)^T d = 0, \quad i \in E, \tag{1.6}$$

$$c_i(x_k) + \nabla c_i(x_k)^T d \geq 0, \quad i \in I, \tag{1.7}$$

*Received by the editors March 2, 1998; accepted for publication (in revised form) March 14, 2000; published electronically July 25, 2000. This research was partially supported by Chinese NSF grants 19525101, 19731010, and by National 9-5 key project 96-221-04-02-02.

http://www.siam.org/journals/sisc/22-2/33486.html

†LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, P.O. Box 2719, Beijing, 100080, China (lxw@lsec.cc.ac.cn, yyx@lsec.cc.ac.cn).

at the kth iterate, where $g_k = \nabla f(x_k)$ is the gradient of the objective function and B_k is an estimate of the Hessian of the Lagrangian function

$$L(x, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x), \tag{1.8}$$

and $(\lambda_1, \lambda_2, \ldots, \lambda_m)^T$ is a multiplier vector approximation.

Because of its nice convergence properties (for example, see [8, 13, 14] and [1]), the SQP method has been attracting the attention of many researchers. It has been extended to problems other than optimization [12, 20].

The requisite consistency of the linearized constraints of the QP subproblem (1.5)–(1.7) is a serious limitation of the SQP method. Within the framework of the SQP method, Powell suggests solving a modified subproblem at each iterate (see [13, 19]):

$$\min\ g_k^T d + \frac{1}{2} d^T B_k d + \frac{1}{2}\delta_k (1-\mu)^2 \tag{1.9}$$

$$\text{s.t. } \mu c_i(x_k) + \nabla c_i(x_k)^T d = 0, \quad i \in E, \tag{1.10}$$

$$\mu_i c_i(x_k) + \nabla c_i(x_k)^T d \geq 0, \quad i \in I, \tag{1.11}$$

where

$$\mu_i = \begin{cases} 1, & c_i(x_k) > 0 \\ \mu, & c_i(x_k) \leq 0 \end{cases}$$

and $0 \leq \mu \leq 1$, $\delta_k > 0$ is a penalty parameter. With some other technique, the computational investigation provided by Schittkowski [17, 18] shows that this modification works very well.

A simple example presented by Burke and Han [3] and Burke [2] indicates this approach may not be the best one. Assume there are two constraints on R:

$$c_1(x) = 1 - e^x = 0, \tag{1.12}$$

$$c_2(x) = x = 0, \tag{1.13}$$

with any objective function $f(x)$ on R. For any infeasible point $x \neq 0$, the linearized constraints are inconsistent, and the only solution of the modified constraints (1.10)–(1.11) is $\mu = 0$ and $d = 0$. Although this example is too specialized to make a general claim, it shows that the problem caused by the inconsistency of the linearized constraints can not always be solved by using (1.10)–(1.11).

Based on a trust region strategy, Fletcher [6, 7] developed the Sl_1QP method for (1.1)–(1.3). Fletcher's approach solves the following QP subproblem at the kth iteration:

$$\min\ g_k^T d + \frac{1}{2} d^T B_k d + \delta_k ||(c(x_k) + \nabla c(x_k)^T d)_-||_1 \tag{1.14}$$

$$\text{s.t. } ||d||_\infty \leq \beta_k, \tag{1.15}$$

where $c(x_k) = (c_1(x_k), \ldots, c_m(x_k))^T$, $(c(x_k) + \nabla c(x_k)^T d)_- \in R^m$ with

$$(c_i(x_k) + \nabla c_i(x_k)^T d)_- = c_i(x_k) + \nabla c_i(x_k)^T d, \quad i \in E, \tag{1.16}$$

$$(c_i(x_k) + \nabla c_i(x_k)^T d)_- = \min(0, c_i(x_k) + \nabla c_i(x_k)^T d), \quad i \in I, \tag{1.17}$$

δ_k is a penalty parameter, β_k is a positive constant. It has been shown that under certain assumptions the search direction generated by (1.14)–(1.15) is locally identical to that by (1.5)–(1.7).

Burke and Han [3] show that Fletcher's approach is still incomplete. One of the reasons is that the search direction may point away from the optimal point.

Similar to the method of Sahba [16], Burke and Han [3] and Burke [2] present an approach to overcome difficulties associated with the inconsistency of the QP subproblem (1.5)–(1.7). Their methods are also similar to the methods of Powell [13] and Fletcher [6, 7]. A feature different from the other methods is that even when (1.1)–(1.3) is itself infeasible their methods can converge to a point which meets a certain first-order necessary optimality condition. However, Burke and Han's method is conceptual.

In this paper, we describe an implementable algorithm which is a modification to the SQP method. Our motivation is to explore further techniques for overcoming the inconsistency of the QP subproblem to derive an efficient reliable SQP algorithm. The line search direction of our algorithm consists of two directions: one is computed by solving a special nonsmooth l_1QP subproblem that depends on only active constraints defined by an active technique; another is obtained by solving a simplified QP problem which is always feasible even when the QP subproblem of the standard SQP method is infeasible. Our algorithm is a generalization of the algorithm presented by Liu and Yuan [10], which is also similar to Burke and Han's method [3]. However, unlike their method, we do not introduce additional bound constraints. Our algorithm obtains a direction which can be a nonzero descent direction of the merit function even if (1.5)–(1.7) is infeasible. At a feasible point of (1.1)–(1.3), the algorithm solves the same subproblem as (1.5)–(1.7). Moreover, under certain assumptions, our algorithm generates the same iterates as the Han–Powell method. Some global convergence results are proved and locally superlinear convergence is derived.

Our algorithm can be easily combined with the trust region approach. Thus, the algorithm can be extended to a trust region algorithm for optimization with general constraints.

The paper is organized as follows. We present our algorithm in section 2. The stationary properties of the algorithm are given in section 3. In section 4 some global convergence results are proved. We discuss the local properties of the algorithm in section 5. In section 6, some preliminary numerical results are reported.

2. The algorithm. Define the penalty function associated with (1.1)–(1.3),

$$\phi(x,r) = f(x) + r||c(x)_-||, \tag{2.1}$$

where $||\cdot||$ is any given convex norm on R^m, $r > 0$ is a penalty parameter, and $c(x)_- \in R^m$ with

$$c_i(x)_- = c_i(x), \quad i \in E, \tag{2.2}$$

$$c_i(x)_- = \min(0, c_i(x)), \quad i \in I. \tag{2.3}$$

It is straightforward to see that $||c(x)_-|| = 0$ if and only if x is a feasible point of (1.1)–(1.3). If the norm $||\cdot||$ is the l_1 norm, (2.1) is the l_1 exact penalty function, which is also a merit function employed by Han [8] and Powell [13, 14]. Throughout this paper if the norm is not specified, it is the same as that used in (2.1).

Define the index sets

$$I_k = \{i \in I : c_i(x_k) \le 0\}, \tag{2.4}$$

$$\bar{I}_k = \{i \in I : c_i(x_k) > 0\}, \tag{2.5}$$

$$J_k = I_k \cup E. \tag{2.6}$$

These index sets are related to the current iterate x_k and can be identified easily. Under some assumptions, we will show that J_k tends to be the index set of the active constraints of (1.1)–(1.3).

Our algorithm solves two subproblems at each iterate: one is an unconstrained piecewise quadratic subproblem (see [13, 21]), and the other is a quadratic programming subproblem. At the kth iteration the unconstrained subproblem has the following form:

$$\min_{d \in R^n} \psi_k(d) = \frac{1}{2} d^T B_k d + r_k ||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T d)_-||, \tag{2.7}$$

where B_k positive definite is an estimate of the Lagrangian Hessian of (1.1)–(1.3), $c_{J_k}(x_k) \in R^{|J_k|}$ is a vector whose components are $c_i(x_k)(i \in J_k)$, $|J_k|$ is the cardinality of the index set J_k, and r_k is the penalty parameter. Let d_{k1} be the solution of (2.7). If x_k is feasible, we have $d_{k1} = 0$. If $d_{k1} \neq 0$, d_{k1} is a descent direction of $\phi(x_k, r_{k+1})$ for sufficiently large r_{k+1}. Moreover, there is a $\tau_k \in (0, 1]$ such that $c_i(x_k) + \delta \nabla c_i(x_k)^T d_{k1} \geq 0$ for all $\delta \in [0, \tau_k]$ and $i \in \bar{I}_k$. In fact, we can let $\tau_k = \min\{1, \hat{\tau}_k\}$, where

$$\hat{\tau}_k = \min\{-c_i(x_k)/(\nabla c_i(x_k)^T d_{k1}) : \quad i \in \bar{I}_k \text{ and } \nabla c_i(x_k)^T d_{k1} < 0\}. \tag{2.8}$$

Let $\hat{c}_i(x_k) = c_i(x_k) + \nabla c_i(x_k)^T \tau_k d_{k1}$ for $i \in \bar{I}_k$. We generate d_{k2} by solving the QP subproblem

$$\min \ g_k^T d + \frac{1}{2} d^T B_k d \tag{2.9}$$

$$\text{s.t. } \nabla c_i(x_k)^T d = 0, \quad i \in E, \tag{2.10}$$

$$\nabla c_i(x_k)^T d \geq 0, \quad i \in I_k, \tag{2.11}$$

$$\hat{c}_i(x_k) + \nabla c_i(x_k)^T d \geq 0, \quad i \in \bar{I}_k, \tag{2.12}$$

and let $d_k = \tau_k d_{k1} + d_{k2}$ be the search direction. It will be shown that d_k is a descent direction for the penalty function where the penalty parameter is updated automatically. Therefore, (2.1) can be employed as a merit function to force the global convergence of the algorithm.

The updating of penalty parameter for the SQP approach is important. In order to obtain the global convergence, Han [8] and Powell [13] require that

$$r \geq ||\lambda_k||_\infty \tag{2.13}$$

for all $k \geq 1$, where λ_k is an estimate of the Lagrangian multiplier vector at x_k. However, (2.13) is generally replaced by some updating procedure when practically implementing an SQP algorithm because we do not know any information about the multiplier vector of (1.8). Similar to Powell [13] and Burke and Han [3], a penalty parameter updating procedure is employed in our algorithm. Since d_{k2} is not related to the constraint violation, the object of updating the penalty parameter is to force d_{k1} to be a descent direction of (2.1). Thus, at the kth iteration we let r_k remain unchanged if d_{k1} is a descent direction; otherwise, r_k is increased in the following way:

$$r_{k+1} = \max \left\{ 2r_k + \rho, \quad \frac{g_k^T d_{k1} + d_{k1}^T B_k d_{k1}}{||(c_{J_k})_-|| - ||(c_{J_k} + \nabla c_{J_k}^T d_{k1})_-||} \right\}, \tag{2.14}$$

where ρ is a positive number.

Now we can state our algorithm as follows.

ALGORITHM 2.1 (a robust algorithm for optimization).

Step 1 [Step 0.] Given the initial approximate x_0, *an* $n \times n$ *symmetric positive definite matrix* B_0, *an initial penalty parameter* $r_0 > 0$, *and some positive scalars* ρ, β *and* μ, *where* $\beta < 1$ *and* $\mu < \frac{1}{2}$; $k = 0$;

If the stopping criterion is satisfied, stop;
Solve subproblem (2.7) *to generate* d_{k1} *and subproblem* (2.9)–(2.12) *to generate* d_{k2};

Step 2. Update penalty parameter. If

$$g_k^T d_{k1} + \frac{1}{2} d_{k1}^T B_k d_{k1} + r_k(||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T d_{k1})_-|| - ||(c_k(x_k))_-||) \leq 0, \tag{2.15}$$

let $r_{k+1} = r_k$; *Otherwise,* r_k *is updated by* (2.14).

Step 3. $d_k = \tau_k d_{k1} + d_{k2}$. *Select the smallest positive integer* s *such that*

$$\phi(x_k + \beta^s d_k, r_{k+1}) - \phi(x_k, r_{k+1}) \leq \mu\beta^s(g_k^T d_k + r_{k+1}(||(c(x_k) + \nabla c(x_k)^T d_k)_-|| - ||(c(x_k))_-||)). \tag{2.16}$$

Let $t_k = \beta^s$ *and* $x_{k+1} = x_k + t_k d_k$;

Step 4. Generate B_{k+1}. *Set* $k = k+1$ *and goto Step* 1.

The stopping criterion is not given in the algorithm. Generally, $||d_k||_2 = 0$ can be used as the stopping criterion. Since no assumption on regularity of the constraints is made, it is possible that d_k does not tend to zero for $k \to \infty$. Thus, we use the condition $||x_{k+1} - x_k||_2 = 0$ as the stopping criterion. In practical implementation, a positive tolerance number will be introduced.

Algorithm 2.1 is similar to the methods proposed by Burke and Han [3] and Burke [2]. Since no additional bound constraints are employed, the algorithm can be implemented in the same way as *SQP* algorithms.

It should be noted that our algorithm solves the same subproblem as (1.5)–(1.7) at a feasible point of (1.1)–(1.3).

Two examples presented by Burke and Han [3] can help us to understand the above algorithm and the differences between our algorithm and Burke and Han's methods.

Example 2.2. The constraint function $c : R \to R^2$ has the form

$$c(x) = \begin{pmatrix} 1 - e^x \\ x \end{pmatrix} \tag{2.17}$$

and $m_e = m = 2$. The norm is the l_1 norm.

For this problem, (2.7) has the form

$$\min_{d \in R} \quad \frac{1}{2} B_k d^2 + r_k(|1 - e^x - e^x d| + |x + d|). \tag{2.18}$$

For any $x_k = x \neq 0$, by direct calculations, $d_{k2} = 0$ and

$$d_{k1} = \begin{cases} e^{-x} - 1 \text{ or } -\frac{r_k}{B_k}(e^x + 1) & \text{if } x > 0, \\ \frac{r_k}{B_k}(e^x + 1) \text{ or } -x & \text{if } x < 0, \\ = 0 & \text{if } x = 0. \end{cases} \tag{2.19}$$

It is easily found that d_{k1} has the following properties:

$$d_{k1} \begin{cases} > 0 & \text{for } x < 0, \\ < 0 & \text{for } x > 0, \\ = 0 & \text{for } x = 0. \end{cases} \tag{2.20}$$

From (2.20), it is easy to see that our algorithm will converge to the solution $x = 0$ from any starting point.

Example 2.3. The constraint function $c : R \to R^2$ is given by

$$c(x) = \begin{pmatrix} -x^2 - 1 \\ -x \end{pmatrix} \tag{2.21}$$

and $m_e = 0$, $m = 2$. Any problem with $c(x)$ as its constraint is infeasible as $c_1(x) = -x^2 - 1 = 0$ has no solution. Let the norm be the l_1 norm.

For constraints (2.21), we have that

$$d_{k1} = \begin{cases} \max\left(-\frac{2r_k}{B_k}x, -\frac{x^2+1}{2x}\right) & \text{if } x < 0, \\ -x, -\frac{x^2+1}{2x}, -\frac{r_k(2x+1)}{B_k} \text{ or } -\frac{2r_k}{B_k}x & \text{if } 0 < x < 1, \\ -x, -\frac{x^2+1}{2x}, -\frac{r_k}{B_k} \text{ or } -\frac{r_k}{B_k}(2x+1) & \text{if } x > 1, \\ \max\left(-\frac{3r_k}{B_k}, -1\right) & \text{if } x = 1, \\ 0 & \text{if } x = 0, \end{cases} \tag{2.22}$$

and

$$d_{k2} \begin{cases} = 0 & \text{if } x < 0, \\ \leq 0 & \text{if } x \geq 0. \end{cases} \tag{2.23}$$

Thus, the search direction generated by our algorithm always points toward the origin, of which the image under c is the closest point to R^2_+ for the l_1 norm.

Algorithm 2.1 can also solve the problem (8.1) of Burke and Han [3] successfully since $d_{k2} = 0$ and d_{k1} directs to the optimal solution for any iteration point $x \neq 0$.

3. Stationary properties of the algorithm. Examples 2.2 and 2.3 display some properties of Algorithm 2.1. These properties are favorable in practice because much information, such as consistency for (1.1)–(1.3), is not known beforehand. Since no restrictions are imposed on the constraint functions, a cluster point of the sequence generated by our algorithm can be one of three different types of points. Similar to Yuan [23], we give their definitions and their stationary properties.

DEFINITION 3.1. *$x \in R^n$ is called*

(1) *a strong stationary point of* (1.1)–(1.3) *if x is feasible and there exists a vector $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T \in R^m$ such that*

$$g(x) - \sum_{i=1}^{m} \lambda_i \nabla c_i(x) = 0, \tag{3.1}$$

$$\lambda_i \geq 0, \quad \lambda_i c_i(x) = 0, \qquad i \in I; \tag{3.2}$$

(2) *an infeasible stationary point of* (1.1)–(1.3) *if x is infeasible and*

$$\min_{d \in R^n} ||(c(x) + \nabla c(x)^T d)_-|| = ||(c(x))_-||; \tag{3.3}$$

(3) *a singular stationary point of* (1.1)–(1.3) *if x is feasible and there exists an infeasible sequence $\{v_k\}$ converging to x such that*

$$\lim_{k\to\infty} \frac{\min_{d\in R^n} ||(c(v_k)+\nabla c(v_k)^T d)_-||}{||(c(v_k))_-||} = 1. \tag{3.4}$$

Definition 3.1 is related to our algorithm closely. It should be noted that there are some differences between our definition and that of [23], for example, the definition on the singular stationary point.

A strong stationary point defined above is precisely a $K-T$ point of (1.1)–(1.3). If $||(c(x_k))_-|| = 0$ and $d_{k2} = 0$, by the first-order $K-T$ condition of (2.9)–(2.12), x_k is a strong stationary point of (1.1)–(1.3).

Throughout this report, we make the following assumption.

Assumption 3.2. (1) $f(x)$ and $c_i(x)$, $i \in E \cup I$, are twice continuously differentiable functions; (2) the approximation B_k of the Lagrangian Hessian is positive definite and there exists two positive constants M_1 and M_2 such that

$$M_1||d||_2^2 \le d^T B_k d \le M_2||d||_2^2 \tag{3.5}$$

holds for all $d \in R^n$ and all $k \ge 1$.

LEMMA 3.3. *The following statements hold:*

(i) *If* (3.3) *holds at x_k, then $d = 0$ solves* (2.7) *uniquely;*

(ii) *if $\{x_k\}$ and $\{r_k\}$ are bounded, then $\{d_{k1}\}$ is also bounded.*

Proof. (i) For any $d \ne 0$, by (3.3), there exists $t > 0$ sufficiently small such that

$$\begin{aligned} \psi_k(td) &= (1/2)t^2 d^T B_k d + r_k||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T(td))_-|| \\ &= (1/2)t^2 d^T B_k d + r_k||(c(x_k) + \nabla c(x_k)^T(td))_-|| \\ &\ge (1/2)t^2 d^T B_k d + r_k||(c(x_k))_-|| > \psi_k(0). \end{aligned} \tag{3.6}$$

Because $\psi_k(d)$ is convex, we can see that $d = 0$ is the unique solution of (2.7).

(ii) The definition of d_{k1} shows that

$$\begin{aligned} \psi_k(0) &\ge \psi_k(d_{k1}) \\ &\ge (1/2)M_1||d_{k1}||_2^2 + r_k||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T d_{k1})_-|| \\ &\ge (1/2)M_1||d_{k1}||_2^2. \end{aligned} \tag{3.7}$$

Therefore,

$$||d_{k1}||_2^2 \le (2/M_1)\psi_k(0) = (2/M_1)r_k||(c(x_k))_-||. \qquad \square \tag{3.8}$$

LEMMA 3.4. *If $x \in R^n$ is an infeasible stationary point or a singular stationary point as defined above, then there exist $\lambda_0 \ge 0$ and $\lambda \in R^m$ such that the first-order necessary optimality condition*

$$\lambda_0 g(x) - \sum_{i=1}^m \lambda_i \nabla c_i(x) = 0, \tag{3.9}$$

$$\lambda_i \ge 0, \quad i \in I, \tag{3.10}$$

holds.

Proof. Suppose that $d(x)$ minimizes the unconstrained problem

$$\min_{d\in R^n} \frac{1}{2}d^T Bd + ||(c(x) + \nabla c(x)^T d)_-|| \tag{3.11}$$

at the iteration point x, where B is any positive definite matrix. Then, the first-order optimality condition at x gives that

$$Bd + \nabla c(x)\mu(x) = 0, \tag{3.12}$$

$$\mu(x) \in \partial ||u|| \big|_{u=(c(x)+\nabla c(x)^T d)_-}, \tag{3.13}$$

where $\mu(x) \in R^m$. It follows directly from (3.13) that $(\mu(x))_i \le 0$ for $i \in I$.

If x is an infeasible stationary point, similar to the proof of Lemma 3.3, we have that $d(x) = 0$. Let $\lambda_0 = 0$ and $\lambda_i = -(\mu(x))_i$, which gives (3.9).

Now suppose that x is a singular stationary point, $\{x_k : k \in K\}$ is a subsequence, and $x_k \to x$ for $k \to \infty (k \in K)$. Suppose that $d(x_k)$ is a solution of (3.11) at x_k; then (3.12)–(3.13) holds at x_k and

$$\begin{aligned} \min_{d\in R^n} ||(c(x_k) + \nabla c(x_k)^T d)_-|| - ||(c(x_k))_-|| \\ \le -\frac{1}{2} d(x_k)^T Bd(x_k) \le 0. \end{aligned} \tag{3.14}$$

Combining (3.4), we have

$$\lim_{k\to\infty, k\in K} \frac{d(x_k)^T Bd(x_k)}{||c(x_k)_-||} = 0. \tag{3.15}$$

Thus, for $k \in K$,

$$\lim_{k\to\infty} ||d(x_k)|| = 0. \tag{3.16}$$

It follows from (3.16) and (3.12) that

$$\lim_{k\to\infty, k\in K} \nabla c(x_k)\mu(x_k) = 0. \tag{3.17}$$

Because $||\mu(x_k)||_0 \le 1$ for all k (where $||\cdot||_0$ is the dual norm of $||\cdot||$), there is a cluster point $\mu^* \in R^m$ with $(\mu^*)_i \le 0$ for $i \in I$. We see that (3.9) holds if we let $\lambda_0 = 0$ and $\lambda_i = -(\mu^*)_i$ for $i \in E \cup I$. This completes our proof. ∎

4. Global convergence. First we show that if our algorithm stops after finite many iterations, the last iterate point must be a strong stationary point or an infeasible stationary point of (1.1)–(1.3).

LEMMA 4.1. *Suppose that d_{k1} is a solution of* (2.7) *and d_{k2} solves* (2.9)–(2.12). *If $d_{k1} = 0$ and $d_{k2} = 0$, then x_k is either a strong stationary point or an infeasible stationary point of* (1.1)–(1.3).

Proof. If $d_{k1} = 0$ and $d_{k2} = 0$, it follows from the first-order Kuhn–Tucker condition of (2.9)–(2.12) that there exists $\lambda_k \in R^m$ such that

$$g_k - \nabla c(x_k)\lambda_k = 0, \tag{4.1}$$

$$(\lambda_k)_i c_i(x_k) = 0 \quad \text{for } i \in \bar{I}_k, \tag{4.2}$$

$$(\lambda_k)_i \ge 0 \quad \text{for } i \in I. \tag{4.3}$$

If $||(c(x_k))_-|| = 0$, then by (4.1)–(4.3) and Definition 3.1(1), x_k is a strong stationary point of (1.1)–(1.3).

Suppose that $||(c(x_k))_-|| \neq 0$. We want to prove that (3.3) holds for x_k. If it is not the case, then there exist $\tilde{d}_k \neq 0$ and $0 < \tau_k \leq 1$ such that

$$\begin{aligned} \min_{d \in R^n} ||(c(x_k) + \nabla c(x_k)^T d)_-|| &= ||(c(x_k) + \nabla c(x_k)^T \tilde{d}_k)_-|| \\ &< ||(c(x_k))_-|| \end{aligned} \tag{4.4}$$

and

$$||(c(x_k) + \nabla c(x_k)^T (\tau_k \tilde{d}_k))_-|| = ||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T (\tau_k \tilde{d}_k))_-||. \tag{4.5}$$

Let $\hat{d}_k = \tau_k \tilde{d}_k$; then it follows that

$$r_k(||(c_{J_k}(x_k))_-|| - ||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T \hat{d}_k)_-||) \leq \frac{1}{2} \hat{d}_k^T B_k \hat{d}_k. \tag{4.6}$$

Define

$$t_0 = \frac{r_k}{2} \frac{||(c_{J_k}(x_k))_-|| - ||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T \hat{d}_k)_-||}{\hat{d}_k^T B_k \hat{d}_k}; \tag{4.7}$$

then by (4.6), $0 < t_0 \leq \frac{1}{4}$ and

$$\begin{aligned} &\psi_k(t_0 \hat{d}_k) - \psi_k(d_{k1}) \\ &\quad \leq \frac{1}{2} t_0^2 \hat{d}_k^T B_k \hat{d}_k + r_k t_0 \{||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T \hat{d}_k)_-|| - ||(c_{J_k}(x_k))_-||\} \\ &\quad \leq \frac{3}{4} t_0 \tau_k r_k \{||(c(x_k) + \nabla c(x_k)^T \tilde{d}_k)_-|| - ||(c(x_k))_-||\} < 0, \end{aligned} \tag{4.8}$$

which gives a contradiction. $\square$

The following result shows that the line search procedure is well defined in the algorithm.

LEMMA 4.2. *Suppose that at least one of d_{k1} and d_{k2} is nonzero; then τ_k is defined by* (2.8). *Then $\tau_k d_{k1} + d_{k2}$ is a descent direction of the penalty function* (2.1) *and the line search condition* (2.17) *is well defined.*

Proof. Let $q(x) = ||c(x)_-||$; then by Lemma 4.1 of Burke and Han [4],

$$q^{'}(x; d) \leq ||(c(x) + \nabla c(x)^T d)_-|| - ||c(x)_-||. \tag{4.9}$$

Define $d_k = \tau_k d_{k1} + d_{k2}$; then

$$\phi^{'}(x_k, r_{k+1}; d_k) \leq g_k^T d_k + r_{k+1}(||(c(x_k) + \nabla c(x_k)^T d_k)_-|| - ||c(x_k)_-||). \tag{4.10}$$

By (2.9)–(2.12) and the convexity of the norm,

$$\begin{aligned} &||(c(x_k) + \nabla c(x_k)^T d_k)_-|| - ||c(x_k)_-|| \\ &\quad \leq \tau_k(||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T d_{k1})_-|| - ||c(x_k)_-||). \end{aligned} \tag{4.11}$$

Thus,

$$\begin{aligned} \phi^{'}(x_k, r_{k+1}; d_k) \leq{}& g_k^T d_{k2} + \tau_k \{g_k^T d_{k1} \\ &+ r_{k+1}(||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T d_{k1})_-|| - ||c(x_k)_-||)\}. \end{aligned} \tag{4.12}$$

It follows from Step 2 of the algorithm that

$$\phi'(x_k, r_{k+1}; d_k) \le -\frac{1}{2} d_{k2}^T B_k d_{k2} - \frac{1}{2}\tau_k d_{k1}^T B_k d_{k1} < 0. \tag{4.13}$$

Now we prove that the line search condition (2.17) is well defined. By the mean value theorem, for any $t > 0$, there exists $\alpha \in (0, t)$ such that

$$f(x_k + td_k) - f(x_k) = tg(x_k + \alpha d_k)^T d_k. \tag{4.14}$$

Similarly there exist $\alpha_i \in (0, t)$ such that

$$c_i(x_k + td_k) - c_i(x_k) = t\nabla c_i(x_k + \alpha_i d_k)^T d_k. \tag{4.15}$$

Define $A_k = (\nabla c_1(x_k + \alpha_1 d_k), \nabla c_2(x_k + \alpha_2 d_k), \ldots, \nabla c_m(x_k + \alpha_m d_k))$; then

$$\begin{aligned} \phi(x_k + td_k, r_{k+1}) - \phi(x_k, r_{k+1}) \le {} & tg(x_k + \alpha d_k)^T d_k \\ & + tr_{k+1}(||(c(x_k) + A_k^T d_k)_-|| - ||c(x_k)_-||). \end{aligned} \tag{4.16}$$

Since

$$||(c(x_k) + A_k^T d_k)_-|| - ||(c(x_k) + \nabla c(x_k)^T d_k)_-|| \le ||(A_k - \nabla c(x_k))^T d_k||, \tag{4.17}$$

it follows from the first part of the proof that there always exists a sufficiently small $t_0 > 0$ such that for all $t \in (0, t_0)$, $\alpha \in (0, t)$,

$$\begin{aligned} & (g(x_k + \alpha d_k) - g_k)^T d_k + r_{k+1}(||(A_k - \nabla c(x_k))^T d_k||) \\ & \quad + (1 - \mu)(g_k^T d_k + r_{k+1}(||(c(x_k) + \nabla c(x_k)^T d_k)_-|| - ||c(x_k)_-||)) < 0, \end{aligned} \tag{4.18}$$

which completes the proof. □

Assumption 4.3. $\{x_k\}$ and $\{d_k\}$ are uniformly bounded.

The assumption on $\{x_k\}$ is common in analyses on convergence of the algorithms. Since the objective function (2.9) is coercive, and $d = 0$ is feasible for (2.10)–(2.12), d_{k2} is bounded. If $r_k \to \infty$, in place of (2.7), we use the following subproblem:

$$\min \frac{1}{2} d^T B_k d + r_k ||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T d)_-|| \tag{4.19}$$

$$\text{s.t. } ||d||_2 \le R, \tag{4.20}$$

where $R > 0$ is a constant, and all analyses still hold since the norm is convex.

If $r_k \to \infty$, by Lemma 4.2 of [23], $\lim_{k\to\infty} ||c(x_k)_-||$ exists.

LEMMA 4.4. *If* $r_k \to \infty$ *and* $\lim_{k\to\infty} ||c(x_k)_-|| \ne 0$, *then there exists a convergent subsequence of* $\{x_k\}$ *which converges to an infeasible stationary point of* (1.1)–(1.3).

Proof. Let S be the set of the accumulation points of $\{x_k\}$. If the lemma is not true, for any $x \in S$, $||c(x)_-|| \ne 0$ and (3.3) does not hold. Thus, there exists a $v > 0$ such that for k large enough,

$$\min_{||d||_2 \le \delta} ||(c(x_k) + \nabla c(x_k)^T d)_-|| \le ||c(x_k)_-|| - v, \tag{4.21}$$

where δ is a positive constant.

Let $\hat{d}_k$ be a vector such that $\|\hat{d}_k\| \le \delta$ and that

$$||(c(x_k) + \nabla c(x_k)^T \hat{d}_k)_-|| = \min_{||d||_2 \le \delta} ||(c(x_k) + \nabla c(x_k)^T d)_-||. \tag{4.22}$$

The fact that $\|\hat{d}_k\| \leq \delta$,

$$||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T \hat{d}_k)_-|| \leq ||(c(x_k) + \nabla c(x_k)^T \hat{d}_k)_-||, \tag{4.23}$$

$r_k \to \infty$, and that d_{k1} solves (2.7) implies that inequality

$$\begin{aligned}
&g_k^T d_{k1} + \frac{1}{2} d_{k1}^T B_k d_{k1} + r_k(||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T d_{k1})_-|| - ||(c(x_k))_-||) \\
&\leq g_k^T d_{k1} + \frac{1}{2} \hat{d}_k^T B_k \hat{d}_k + r_k(||(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T \hat{d}_k)_-|| - ||(c(x_k))_-||) \\
&\leq g_k^T d_{k1} + \frac{1}{2} M\delta^2 - r_k v < 0
\end{aligned} \tag{4.24}$$

holds for all sufficiently large k, which contradicts the parameter updating procedure. ∎

Similarly, we have the following result.

LEMMA 4.5. *If* $r_k \to \infty$ *and* $\lim_{k\to\infty} ||c(x_k)_-|| = 0$*, then there exists a convergent subsequence of* $\{x_k\}$ *which converges to a singular stationary point of* (1.1)–(1.3).

Proof. Let x be any accumulation point of $\{x_k\}$. Then x is a feasible point of (1.1)–(1.3). The condition $r_k \to \infty$ implies that there exists an infinite subsequence $\{x_k : k \in \mathcal{K}\}$ such that $||(c(x_k))_-|| \neq 0$ for $k \in \mathcal{K}$.

If the result is not true, then for any convergent subsequence $\{x_k : k \in \tilde{K}\}(\tilde{K} \subset \mathcal{K})$, (3.4) does not hold. Hence, there exists a positive number v such that (4.21) holds. Similar to Lemma 4.4, the proof can be completed. ∎

The above two lemmas imply that r_k is bounded if no subsequence of $\{x_k\}$ converges to an infeasible stationary point or a singular stationary point of (1.1)–(1.3).

LEMMA 4.6. *Suppose that* $r_k = r$ *(*r *is a positive constant) for all* k *large enough,* $\{x_k\}$ *is an infinite sequence, and* $\{x_k : k \in \hat{K}\}$ *is a convergent subsequence. Then* $d_k \to 0$ *for* $k \in \hat{K}$ *and* $k \to \infty$.

Proof. We proceed by contradiction. Without loss of generality, assume that $r_k = r$ for all k.

Suppose that there exist an infinite subset $K' \subset \hat{K}$ and a positive constant η such that $||d_k||_2 \geq \eta$ for $k \in K'$. By Lemma 4.2, there exists $\hat{\eta} > 0$ such that

$$\nabla_t \phi(x_k + td_k, r)|_{t=0} \leq -\hat{\eta} < 0. \tag{4.25}$$

Thus, there exists a constant $\sigma > 0$ and sufficiently small $t_k > 0$ such that for $k \in K'$,

$$\phi(x_k + t_k d_k, r) \leq \phi(x_k, r) - \sigma. \tag{4.26}$$

The above inequality implies that

$$\Sigma_{k\in K'}(\phi(x_k + t_k d_k, r) - \phi(x_k, r)) \leq -\Sigma_{k\in K'}\sigma = -\infty, \tag{4.27}$$

which is a contradiction. This completes the proof. ∎

In the following theorem, we assume that (d_{k2}, λ_k) is a Kuhn–Tucker pair of (2.9)–(2.12) at x_k, where $\lambda_k \in R^m$ is a Lagrange multiplier vector associated with d_{k2}.

THEOREM 4.7. *Suppose that* $\{x_k\}$ *is an infinite sequence generated by the algorithm,* $\{r_k\}$ *and* $\{\lambda_k\}$ *are bounded, and* $\{x_k : k \in \hat{K}\}$ *is a subsequence converging to* x^**. If* $||c(x^*)_-|| = 0$*, then* x^* *is a strong stationary point of* (1.1)–(1.3).

Proof. Since (d_{k2}, λ_k) is a Kuhn–Tucker pair of (2.9)–(2.12) at x_k, we have

$$g_k + B_k d_{k2} - \nabla c(x_k)\lambda_k = 0, \tag{4.28}$$

$$(\lambda_k)_i \nabla c_i(x_k)^T d_{k2} = 0 \text{ for } i \in I_k, \tag{4.29}$$

$$(\lambda_k)_i(\hat{c}_i(x_k) + \nabla c_i(x_k)^T d_{k2}) = 0 \text{ for } i \in \bar{I}_k, \tag{4.30}$$

$$(\lambda_k)_i \geq 0 \text{ for } i \in I, \tag{4.31}$$

and (2.10)–(2.12) hold. Moreover, (d_{k1}, μ_{J_k}) satisfies that

$$B_k d_{k1} + r_k \nabla c_{J_k}(x_k)\mu_{J_k} = 0, \tag{4.32}$$

$$\mu_{J_k} \in \partial ||u|| \big|_{u=(c_{J_k}(x_k)+\nabla c_{J_k}(x_k)^T d_{k1})_-}, \tag{4.33}$$

where $\mu_{J_k} \in R^{|J_k|}$ is a vector with $(\mu_{J_k})_i (i \in J_k)$ as its components. It follows from (4.33) that $(\mu_{J_k})_i \leq 0$ for $i \in I_k$. Thus, we have

$$g_k + B_k d_k - \nabla c(x_k) u_k = 0, \tag{4.34}$$

$$(u_k)_i(c_i(x_k) + \nabla c_i(x_k)^T d_k) = 0 \text{ for } i \in \bar{I}_k, \tag{4.35}$$

$$(u_k)_i \geq 0 \text{ for } i \in I, \tag{4.36}$$

where

$$(u_k)_i = (\lambda_k)_i - r_k \tau_k (\mu_{J_k})_i \quad \text{ for } i \in J_k, \tag{4.37}$$

and

$$(u_k)_i = (\lambda_k)_i \quad \text{ for } i \in \bar{I}_k. \tag{4.38}$$

Let $\mathcal{I}(x^*) = \{i : i \in I_k \text{ for infinitely many } k \in \hat{K}\}$, $I(x^*) = \{i \in I : c_i(x^*) = 0\}$. Then $I(x^*) \supset \mathcal{I}(x^*)$.

By (4.33), $||\mu_{J_k}||_0 \leq 1$, where $|| \cdot ||_0$ is the dual norm of $|| \cdot ||$ defined by (2.1). Then it follows from Lemma 4.6 that there exists a cluster point $u^* \in R^m$ of $\{u_k\}$ such that

$$g(x^*) - \nabla c(x^*) u^* = 0, \tag{4.39}$$

$$(u^*)_i c_i(x^*) = 0 \text{ for } i \in I, \tag{4.40}$$

with $(u^*)_i \geq 0$ for $i \in I$. ◻

The condition on λ_k is not restrictive. The boundedness of $\{r_k\}$ implies that (2.15) holds for sufficiently large k. Thus, by (4.32)–(4.33) and (4.28), we have

$$r_k \geq ((\nabla c(x_k)^T d_{k1})^T \lambda_k)/||\nabla c(x_k)^T d_{k1}|| \tag{4.41}$$

for sufficiently large k. On the other hand, if we suppose the Mangasarian–Fromovitz condition holds at x^*, it can be proved that λ_k is bounded.

It should be noted that the above convergence results do not rely on any linear independence assumption of the gradients of the constraints. Thus, the algorithm may terminate at some iteration, which is not a Kuhn–Tucker point of (1.1)–(1.3), even if the penalty parameter is bounded. A simple example will demonstrate this case.

Example 4.8. Consider the problem

$$\min\ y_1 + (1/2)y_2^2 \tag{4.42}$$

$$\text{s.t. } (1/2)y_1^2 = 0, \tag{4.43}$$

$$y_1 + y_2^3 - 3/2 = 0. \tag{4.44}$$

Let the penalty parameter $r = 1$; the algorithm will terminate at $(1, 0)$, which is not a Kuhn–Tucker point of (4.42)–(4.44).

5. Local convergence. To study local convergence properties of the algorithm, we make the following assumption.

Assumption 5.1. (1) $x_k \to x^*$, where x^* is a Kuhn–Tucker point of (1.1)–(1.3); (2) let $I^* = \{i \in I : c_i(x^*) = 0\}$; $\nabla c_i(x^*)(i \in E \cup I^*)$ are linearly independent; (3) $r_k = r$ for $k \geq \hat{k}$, where $r > 0$ is a constant, and $\hat{k}$ is a sufficiently large positive integer.

The definitions of (2.4)–(2.5) imply that for infinitely many k, there exists a small $\epsilon > 0$ such that $c_i(x_k) \geq \epsilon$ for $i \in \bar{I}_k$. Thus, by Assumption 4.3 and (2.8), we have $\tau_k \geq \tau_0$ for infinitely many k, where $\tau_0 > 0$ is a constant.

For sufficiently large k, by definitions of (2.4)–(2.5), $\bar{I}_k \supseteq \bar{I}_{k+1}$. Thus, $I_k = I^*$ for sufficiently large k. Moreover, under Assumption 5.1, it follows from (3.8) that $||d_{k1}||_2 \to 0$ for $k \to \infty$. Therefore, $\tau_k \to 1$ for sufficiently large k.

LEMMA 5.2. *Under Assumption* 5.1*, suppose that d_{k1} is a solution of* (2.7) *at the point x_k; then there exists a sufficiently large $k^{'}$, such that for $k \geq k^{'}$,*

$$(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T d_{k1})_- = 0. \tag{5.1}$$

Proof. The first-order necessary condition of (2.7) imply that (4.32)–(4.33) hold and $||\mu_{J_k}||_0 \leq 1$ with $||\cdot||_0$ being the dual norm of $||\cdot||$ defined by (2.1). If at the kth iteration $(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T d_{k1})_- \neq 0$, then $||\mu_{J_k}||_0 = 1$.

Define $p_k = ||B_k d_{k1} + r_k \nabla c_{J_k}(x_k)\mu_{J_k}||_2$ and $J^* = E \cup I^*$. If for sufficiently large k, $(c_{J_k}(x_k) + \nabla c_{J_k}(x_k)^T d_{k1})_- \neq 0$, then it follows from Lemma 4.6 and the equivalence of the norm that

$$\begin{aligned} p_k &= ||r_k \nabla c_{J^*}(x^*)\mu_{J_k}||_2 + O(||x_k - x^*||_2) + O(||d_{k1}||_2) \\ &\geq r_k ||\mu_{J_k}||_2 / ||\nabla c_{J^*}(x^*)^+||_2 + o(1) \\ &\geq c_0 r_k ||\mu_{J_k}||_0 / ||\nabla c_{J^*}(x^*)^+||_2 + o(1) \\ &\geq c_0 r / (2||\nabla c_{J^*}(x^*)^+||_2), \end{aligned} \tag{5.2}$$

where $\nabla c_{J^*}(x^*) \in R^{n \times |J^*|}$ is a matrix with $\nabla c_i(x^*)$ $(i \in J^*)$ as its volume vectors, $\nabla c_{J^*}(x^*)^+$ is its generalized inverse matrix, and $c_0 > 0$ is a constant. Equation (5.2) contradicts (4.32). This completes our proof. $\square$

The above lemma shows that there exists a sufficient large integer k_0, such that for $k \geq k_0$, the piecewise quadratic subproblem (2.7) is equivalent to the following quadratic programming problem:

$$\min\ \frac{1}{2} d^T B_k d \tag{5.3}$$

$$\text{s.t. } c_i(x_k) + \nabla c_i(x_k)^T d = 0, \quad i \in E, \tag{5.4}$$

$$c_i(x_k) + \nabla c_i(x_k)^T d \geq 0, \quad i \in I_k. \tag{5.5}$$

Assumption 5.3. Suppose that λ^* is a Lagrangian multiplier vector associated with x^*: (1) The strict complementarity condition holds at (x^*, λ^*); (2) $\nabla^2 L(x^*, \lambda^*)$ is positive definite for all nonzero d in the null space $\{d: \ \nabla c_i(x^*)^T d = 0, \ \ i \in E \cup I^*\}$, where $L(x, \lambda)$ is defined as (1.8).

It follows from Assumption 5.3 that (x^*, λ^*) is an isolated Kuhn–Tucker pair of (1.1)–(1.3). If the conditions in Assumption 5.3 hold, then for sufficiently large k, d_{k1} derived by (5.3)–(5.5) is a solution of the problem

$$\min \frac{1}{2} d^T B_k d \tag{5.6}$$

$$\text{s.t. } c_i(x_k) + \nabla c_i(x_k)^T d = 0 \quad \text{ for } i \in E \cup I^*, \tag{5.7}$$

and d_{k2} generated by (2.9)–(2.12) solves

$$\min g_k^T d + \frac{1}{2} d^T B_k d \tag{5.8}$$

$$\text{s.t. } \nabla c_i(x_k)^T d = 0 \quad \text{ for } i \in E \cup I^*. \tag{5.9}$$

Let $\nabla c_{J^*}(x_k)$ be an $n \times |J^*|$ matrix with $\nabla c_i(x_k) (i \in J^*)$ as its components. By direct calculations, it follows from (5.6)–(5.7) that

$$d_{k1} = -B_k^{-1} \nabla c_{J^*}(x_k) (\nabla c_{J^*}(x_k)^T B_k^{-1} \nabla c_{J^*}(x_k))^{-1} c_{J^*}(x_k), \tag{5.10}$$

and by (5.8)–(5.9),

$$\begin{aligned} d_{k2} = {} & B_k^{-1} \nabla c_{J^*}(x_k) (\nabla c_{J^*}(x_k)^T B_k^{-1} \nabla c_{J^*}(x_k))^{-1} \nabla c_{J^*}(x_k)^T B_k^{-1} g_k \\ & - B_k^{-1} g_k. \end{aligned} \tag{5.11}$$

Thus, $d_{k1} + d_{k2}$ is a solution of the problem

$$\min g_k^T d + \frac{1}{2} d^T B_k d \tag{5.12}$$

$$\text{s.t. } c_i(x_k) + \nabla c_i(x_k)^T d = 0 \quad \text{ for } i \in J^*. \tag{5.13}$$

The above discussion can be stated as the following lemma.

LEMMA 5.4. *If the conditions in Assumptions* 5.1 *and* 5.3 *hold, then there exists a sufficiently large* k_1, $k_1 \geq k_0$, *such that for* $k \geq k_1$, *the algorithm generates identical directions with the Han–Powell method.*

By Lemma 5.4 and the related results of the *SQP* method (for example, see [1] and [22]), the superlinear convergence of the algorithm is a direct result.

LEMMA 5.5. *Suppose that the conditions in Assumption* 5.3 *hold. If*

$$\lim_{k \to \infty} \frac{||P^*(B_k - \nabla^2 L(x^*, \lambda^*)) d_k||_2}{||d_k||_2} = 0, \tag{5.14}$$

where P^* *is a projection matrix on the null space* $\{d: \ \nabla c_i(x^*)^T d = 0, \ \ i \in E \cup I^*\}$; *then*

$$\lim_{k \to \infty} \frac{||x_k + d_k - x^*||}{||x_k - x^*||} = 0. \tag{5.15}$$

A superlinear convergence step may be truncated due to the nonsmoothness of the merit function, which is known as "the Marotos effect" (for example, see [22, 24]). In order to avoid this difficulty, the second-order correction technique is considered by Mayne and Polak [11], Coleman and Conn [5], Fletcher [7], and so on. For our problem, when $||c_{J_k}(x_k)|| \leq \epsilon$ (ϵ is a prescribed number), we solve the subproblem

$$\min_{d \in R^n} \frac{1}{2} d^T B_k d + r_k ||(c_{J_k}(x_k + d_k) + \nabla c_{J_k}(x_k)^T d)_-|| \tag{5.16}$$

to generate the second-order correction step $\tilde{d}_k$. The algorithm with the second-order correction technique is presented, which is a modification to Algorithm 2.1.

ALGORITHM 5.6.

Step 1 [Step 0.] Given $x_0 \in R^n$, $B_0 \in R^{n \times n}$, $r_0 > 0$, $0 < \mu < \frac{1}{2}$, $0 < \beta < 1$, $\rho > 0$, ϵ_0, ϵ_1, ϵ_2, $\epsilon_3 > 0$, $k := 0$.

Generate J_k *and* $\bar{I}_k$ *by* (2.4)–(2.6), *solve* (2.7) *giving* d_{k1};
Calculate τ_k *by* (2.8); *Solve* (2.9) *giving* d_{k2}; $d_k = \tau_k d_{k1} + d_{k2}$;
If $||d_k|| \leq \epsilon_0$, *stop;*
If $||c_{J_k}(x_k)|| \leq \epsilon_1$ *solve* (5.16) *giving* $\tilde{d}_k$; *else* $\tilde{d}_k = 0$.

Step 2. $r_{k+1} := r_k$ *if* (2.15) *holds, otherwise compute* r_{k+1} *by* (2.14).

Step 3. $s = 0, 1, 2, \ldots$; *If*

$$\begin{aligned} \phi(x_k + \beta^s d_k + \beta^{2s} \tilde{d}_k, r_{k+1}) - \phi(x_k, r_{k+1}) \leq \mu \beta^s (g_k^T d_k \\ + r_{k+1}(||(c(x_k) + \nabla c(x_k)^T d_k)_-|| - ||(c(x_k))_-||)). \end{aligned} \tag{5.17}$$

Let $t_k = \beta^s$ *and* $x_{k+1} = x_k + t_k d_k + t_k^2 \tilde{d}$.

Step 4. *If* $||x_{k+1} - x_k|| \leq \epsilon_3$, *stop.*

Step 5. *Compute the values of* $f(x)$, $c(x)$, $\nabla f(x)$, *and* $\nabla c(x)$ *at* x_{k+1};
Generate B_{k+1}; $k := k + 1$ *and goto Step* 1.

Similar to the above discussion on Lemmas 5.2 and 5.4, and to the analyses in [11, 22, 24], we have the following result.

THEOREM 5.7. *Under Assumptions* 5.1 *and* 5.3, *suppose that* (5.14) *holds,* $\epsilon_i = 0 (i = 0, 1, 2, 3)$, *and* $\{x_k\}$ *is an infinite sequence generated by Algorithm* 5.6. *Then*

$$\lim_{k \to \infty} \frac{||x_k + d_k + \tilde{d}_k - x^*||}{||x_k - x^*||} = 0, \tag{5.18}$$

and there exists a sufficiently large k_2 *such that for* $k \geq k_2$, $t_k = 1$. *Thus,* $\{x_k\}$ *converges Q-superlinearly.*

6. Numerical results. A FORTRAN subroutine was programmed to test our algorithm. Our experiments were done on an SGI indigo workstation at the State Key Laboratory of Scientific and Engineering Computing, Chinese Academy of Sciences, Beijing.

The norm in (2.1) is selected to be the l_∞ norm. We solve the piecewise quadratic subproblem (2.7) by reformulating it as follows:

$$\min \frac{1}{2} w^T Q_k w + p_k^T w \tag{6.1}$$

$$\text{s.t. } y - \nabla c_i(x_k)^T d \geq c_i(x_k) \quad \text{for } i \in E, \tag{6.2}$$

$$y + \nabla c_i(x_k)^T d \geq -c_i(x_k) \quad \text{for } i \in J_k, \tag{6.3}$$

where $Q_k = \begin{pmatrix} B_k & \\ & 0 \end{pmatrix} \in R^{(n+1)\times(n+1)}$, $w = \begin{pmatrix} d \\ y \end{pmatrix} \in R^{n+1}$, $p_k = \begin{pmatrix} 0 \\ r_k \end{pmatrix} \in R^{n+1}$. Problem (6.1)–(6.3) is a convex quadratic programming problem, which is, in our algorithm, solved by Powell's subroutine ZQPCVD [15]. The second-order correction subproblem (5.16) is solved similarly.

The first test problem that we solved is taken from [16]:

$$\min\ x_1x_2 \tag{6.4}$$

$$\text{s.t.}\ -\sin(x_1) \geq 0, \tag{6.5}$$

$$\cos(x_1) \geq 0, \tag{6.6}$$

$$-x_1^2 - x_2^2 + \pi/2 \geq 0, \tag{6.7}$$

$$x_1 + \pi \geq 0, \tag{6.8}$$

$$x_2 + \pi/2 \geq 0, \tag{6.9}$$

and the standard starting point is $x_0 = (0, 5)^T$. Sahba's algorithm terminates at the point $x^* = (0, -1.25331)^T$, which is an approximate Kuhn–Tucker point and not the approximate minimum point of (6.4)–(6.9). The other test problems are from Hock and Schittkowski [9].

In our implementation of the algorithm, a small positive tolerance number ϵ is introduced and the index sets

$$I_k(\epsilon) = \{i \in I : c_i(x_k) \leq \epsilon\}, \tag{6.10}$$

$$\bar{I}_k(\epsilon) = \{i \in I : c_i(x_k) > \epsilon\}, \tag{6.11}$$

$$J_k(\epsilon) = I_k(\epsilon) \cup E \tag{6.12}$$

are employed instead of (2.4)–(2.6).

For each problem, the standard initial point is used. We choose initial parameters $\mu = 0.1$, $\beta = 0.5$, $\rho = 1$, and $\epsilon_i = 10^{-6}$ for $i = 0, 1, 2, 3$. The choice of the initial penalty parameter is scale dependent and $r_0 = 1$ is chosen for our test problems. The initial Lagrangian Hessian estimate $B_0 = I$ and B_k is updated by the damped BFGS formula (see [14]):

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}, \tag{6.13}$$

where

$$y_k = \begin{cases} \hat{y}_k, & \hat{y}_k^T s_k \geq 0.2 s_k^T B_k s_k, \\ \theta_k \hat{y}_k + (1-\theta_k) B_k s_k, & \text{otherwise}, \end{cases} \tag{6.14}$$

and $\hat{y}_k = g_{k+1} - g_k + (\nabla c(x_{k+1}) - \nabla c(x_k))\lambda_k$, $s_k = x_{k+1} - x_k$, $\theta_k = 0.8 s_k^T B_k s_k / (s_k^T B_k s_k - s_k^T \hat{y}_k)$, λ_k is a multiplier associated with (2.9)–(2.12).

The test problems are also solved by Powell's subroutine VMCWD, which is a very successful algorithm for many nonlinear programming problems. The error tolerance for VMCWD is 10^{-8}.

The subroutine VMCWD failed to solve Sahba's problem (6.4)–(6.9) since the constraints seem to be inconsistent after the first iteration. The numerical results given by our algorithm are presented in Table 1, where $R-KT$ and $R-CV$ represent the l_2 norms of the gradient of the Lagrangian and the violation of the constraints,

TABLE 1

k	$x_k^{(1)}$	$x_k^{(2)}$	r_k	τ_k	t_k	$R-KT$	$R-CV$
0	0.0	5.0	1.0	1.0	1.0	5.0	2.3429E+1
1	-3.1416	2.6571	1.0	1.0	1.0	3.2415	1.5391E+1
2	-1.1116	2.3552	1.0	1.0	1.0	1.7142	5.2119
3	-0.8262	1.3834	1.0	1.0	1.0	8.8489E-1	1.0257
4	-0.9236	0.9546	1.0	1.0	1.0	4.4570E-2	1.9339E-1
5	-0.8899	0.8858	1.0	1.0	1.0	6.1008E-3	5.8572E-3
6	-0.8870	0.8854	1.0	1.0	1.0	2.4008E-3	7.7911E-6
7	-0.8863	0.8862	1.0	1.0	1.0	1.3274E-4	0.0

TABLE 2

				VMCWD		Our algorithm	
Problem	n	m_e	m	NI-NF-NG	Residual	NI-NF-NG	Residual
HS7	2	1	1	12-14-14	4.94E-08	9-18-10	3.85E-08
HS14	2	1	2	5-6-6	7.90E-11	4-5-5	2.98E-07
HS22	2	0	2	5-7-7	3.18E-10	23-46-24	4.58E-08
HS38	4	0	8	81-104-104	8.96E-04	38-64-39	6.68E-05
HS43	4	0	3	12-15-15	5.14E-06	12-23-13	4.48E-06
HS52	5	3	3	5-9-9	2.21E-05	16-21-17	2.43E-12
HS63	3	2	5	8-9-9	6.72E-07	7-8-8	2.41E-07
HS76	4	0	7	5-6-6	1.45E-04	6-7-7	2.12E-07
HS86	5	0	15	4-6-6	1.78E-04	4-7-5	6.22E-05
HS113	10	0	8	12-17-17	6.46E-06	14-20-15	4.07E-05

respectively. The algorithm terminates at the approximate minimum point of (6.4)–(6.9).

Some numerical results for equality constrained optimization problems have been reported in Liu and Yuan [10]. It has been noticed that our algorithm can overcome the difficulties associated with the linear dependence of the gradients of the constraints, since an unconstrained subproblem is solved at each iterate.

The numerical results for other test problems are listed in Table 2. The problems are numbered in the same way as in Hock and Schittkowski [9]. For example, "HS43" is problem 43 in Hock and Schittkowski [9]. NI, NF, and NG represent the numbers of iterations, function, and gradient calculations, respectively.

The numerical results show that our algorithm is comparable to VMCWD. However, our algorithm requires slightly more function evaluations.

Acknowledgement. We would like to thank an anonymous referee for valuable comments.

REFERENCES

[1] 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.
[2] J.V. BURKE, *A sequential quadratic programming method for potentially infeasible mathematical programs*, J. Math. Anal. Appl., 139 (1989), pp. 319–351.
[3] J.V. BURKE AND S.P. HAN, *A robust sequential quadratic programming method*, Math. Programming, 43 (1989), pp. 277–303.
[4] J.V. BURKE AND S.P. HAN, *A Gauss-Newton approach to solving generalized inequalities*, Math. Oper. Res., 11 (1986), pp. 632–643.

[5] T.F. Coleman and A.R. Conn, *Nonlinear programming via an exact penalty function: Asymptotic analysis*, Math. Programming, 24 (1982), pp. 123–136.
[6] R. Fletcher, *Practical Methods for Optimization*, *Vol.* 2, *Constrained Optimization*, John Wiley and Sons, Chichester, UK, 1981.
[7] R. Fletcher, *A model algorithm for composite nondifferentiable optimization problems*, Math. Programming Stud., 17 (1982), pp. 67–76.
[8] S.P. Han, *A globally convergent method for nonlinear programming*, J. Optim. Theory Appl., 22 (1977), pp. 297–309.
[9] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Econ. and Math. Systems 187, Springer-Verlag, Berlin, New York, 1981.
[10] X. Liu and Y. Yuan, *A Globally Convergent, Locally Superlinearly Convergent Algorithm for Equality Constrained Optimization*, Research Report, ICM-97-84, Institute of Computational Mathematical and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing, China.
[11] D.Q. Mayne and E. Polak, *A superlinearly convergent algorithm for constrained optimization problems*, Math. Programming Stud., 16 (1982), pp. 45–61.
[12] J.S. Pang and S.A. Gabriel, *NE/SQP: A robust algorithm for the nonlinear complementarity problem*, Math. Programming, 60 (1993), pp. 295–337.
[13] M.J.D. Powell, *A fast algorithm for nonlinearly constrained optimization calculations*, in Proceedings 1977 Dundee Biennial Conference on Numerical Analysis, G.A.Watson, ed., Springer-Verlag, Berlin, 1978, pp. 144–157.
[14] M.J.D. Powell, *The convergence of variable metric methods for nonlinear constrained optimization calculations*, in Nonlinear Programming 3, O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, eds., Academic Press, New York, 1978, pp. 27–63.
[15] M.J.D. Powell, *ZQPCVX: A Fortran Subroutine for Convex Quadratic Programming*, Report DAMTP 1983/NA17, University of Cambridge, UK, 1983.
[16] M. Sahba, *Globally convergent algorithm for nonlinearly constrained optimization problems*, J. Optim. Theory Appl., 52 (1987), pp. 291–309.
[17] K. Schittkowski, *The nonlinear programming method of Wilson, Han and Powell with an augmented Lagrangian type line search function, Part* 1: *Convergence analysis*, Numer. Math., 38 (1981), pp. 83–114.
[18] K. Schittkowski, *On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function*, Math. Operationsforsch. Statist. Ser. Optim., 14 (1983), pp. 197–216.
[19] J. Stoer, *Principles of sequential quadratic programming methods for solving nonlinear programs*, in Computational Mathematical Programming, NATO Advanced Science Institute Series F: Computer and Systems Sciences 15, K. Schittkowski, ed., Springer-Verlag, Berlin, 1985, pp. 166–207.
[20] K. Taji and M. Fukushima, *A new merit function and a successive quadratic programming algorithm for variational inequality problems*, SIAM J. Optim., 6 (1996), pp. 704–713.
[21] R.B. Wilson, *A Simplicial Algorithm for Concave Programming*, Ph.D. thesis, Harvard University, Cambridge, MA, 1963.
[22] Y. Yuan, *Numerical Methods for Nonlinear Programming*, Modern Mathematics Series, Shanghai Scientific and Technical Publishers, Shanghai, China, 1993 (in Chinese).
[23] Y. Yuan, *On the convergence of a new trust region algorithm*, Numer. Math., 70 (1995), pp. 515–539.
[24] Y. Yuan and W. Sun, *Theories and methods for optimization*, Scientific Press, Beijing, 1997 (in Chinese).