Journal of Computational and Applied Mathematics 300 (2016) 245–258

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A line search exact penalty method with bi-object strategy for nonlinear constrained optimization

Zhongwen Chen [a,*], Yu-Hong Dai [b]

[a] School of Mathematics Science, Soochow University, Suzhou, 215006, PR China

[b] LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

HIGHLIGHTS

- Penalty factor is only related to the information at the current iterate point.
- The sequence of the penalty parameter is non-monotone.
- The search direction is related to the penalty factor.
- The acceptable criterion is not related to the penalty factor.
- Method can handle degenerate problems and inconsistent constraint linearizations.

ARTICLE INFO

Article history:
Received 2 October 2015
Received in revised form 23 December 2015

MSC:
65K05
49M37
90C30

Keywords:
Nonlinear constrained optimization
Exact penalty methods
Bi-object strategy
Global convergence

ABSTRACT

The exact penalty methods are very popular because of their ability to handle degenerate problems and inconsistent constraint linearizations. This paper presents a line search exact penalty method with bi-object strategy (LSBO) for nonlinear constrained optimization. In the algorithm LSBO, the penalty parameter is selected at every iteration such that the sufficient progress toward feasibility and optimality is guaranteed along the search direction. In contrast with classical exact penalization approaches, LSBO method has two goals to determine whether the current iteration is successful or not. One is improving the feasibility and the other is reducing the value of the objective function. Moreover, the penalty parameter is only related to the information at the current iterate point. The sequence of the penalty parameter is non-monotone, which does not affect the global convergence in theory and is seen to be advantageous in practice. It is shown that the algorithm enjoys favorable global convergence properties under the weaker assumptions. Numerical experiments illustrate the behavior of the algorithm on various difficult situations.

1. Introduction

In this paper, we develop a line search exact penalty method with two-object strategy for finding a local solution of the following nonlinear programming problem

$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & c_i(x) \geq 0, \ i \in \mathcal{I} = \{1, \ldots, m\}, \end{aligned} \tag{1.1}$$

where we assume $f : R^n \to R$ and $c_i : R^n \to R^m$ $(i \in \mathcal{I})$ are twice continuously differentiable.

* Corresponding author.
E-mail addresses: zwchen@suda.edu.cn (Z. Chen), dyh@lsec.cc.ac.cn (Y.-H. Dai).

http://dx.doi.org/10.1016/j.cam.2015.12.033
0377-0427/

There are many practical methods for solving problem (1.1), for example, the sequential quadratic programming (SQP) methods [1], the trust-region SQP methods [2], the interior point methods [3] and so on. All optimization methods generate a sequence of trial steps, which are computed as solutions of some quadratic or linear–quadratic model. The criteria for accepting or rejecting trial steps are the strategy for guaranteeing global convergence. One of the strategies is to use some merit function or some penalty function to measure the quality of a trial step. The penalty function is usually a linear combination of the objective function and some measure of constraint violation, where the objective function minimization and the constraint satisfaction are treated together within the framework of a single penalty function minimization problem.

The main difficulty associated with the use of penalty functions is the choice of the penalty parameter. There is usually a threshold value below which the penalty function does not have a local minimum at the solution to (1.1). This threshold value is unknown in advance. If the initial choice of the penalty parameter is too small, the iterates may move away from the solution which may result in an infeasible point of (1.1) or even an unbounded below in the penalty function. On the other hand, if the penalty parameter is excessively large, the penalty function may be difficult to minimize, as emphasizing constraint feasibility too much may lead to small trial steps (or even the rejection of good trial steps) on the curved boundary of the feasible region of (1.1).

In order to handle the selection of the penalty parameter, some researchers present the various techniques without the penalty function, which are called the penalty-free-type methods, for example, see [4–9] and the references therein. Some other people research the updating strategies of the penalty parameter adaptively, e.g., [10–19]. Among those penalty-type methods which use any penalty function, the exact penalty methods are very popular because of their ability to handle degenerate problems and inconsistent constraint linearizations [11,12]. Exact penalty methods have also been used successfully to solve mathematical programs with complementarity constraints (MPCCs) [17], a class of problems that do not satisfy the Mangasarian–Fromovitz constraint qualification at any feasible point.

The idea in this paper is motivated by the penalty-type methods and the penalty-free-type methods. The trial step (the search direction) is computed by a piecewise quadratic model of the exact penalty function. The penalty parameter is selected at every iteration so that the sufficient progress toward feasibility and optimality is guaranteed along the search direction, which is called steering rules [11]. This requires that an auxiliary subproblem (a linear program) be solved in certain cases. Byrd et al. [11] present a line search exact penalty method using steering rules, which requires that the penalty function is descent sufficiently along the search direction at every iteration. However, the method presented here is very different from Byrd's method. The new method has two goals to determine whether the current iteration is successful or not. One is improving the feasibility and the other is reducing the value of the objective function. The new method allows for a certain amount of non-monotonicity on the objective function and on the measure of constraint violation compared to a penalty function approach. Gould et al. [16] present a filter method for nonlinear optimization, where every trial step is computed from subproblems that value reducing both the constraint violation and the objective function. The new method does not use filter technique and its search direction is similar to Byrd et al. [11] but is different from Gould et al. [16]. Moreover, the penalty parameter is only related to the information at the current iterate point. The sequence of the penalty parameter is non-monotone. This property does not affect the global convergence in theory and is seen to be advantageous in practice. Byrd et al. [12] point out clearly that many of the failures caused by large values of the penalty parameter seemed to occur because, near the feasible region, there are often small increases in infeasibility due to nonlinearities in constraints or roundoff error in even linear constraints. Because of the large value of the penalty parameter, these increases dominated the objective function improvement, and forced the penalty method to take very small steps, and sometimes completely prevented further progress. To restrict using excessively large penalty parameter is valuable in practice, which is also confirmed by the numerical examples in this paper.

This paper is divided into four sections. The next section describes the new algorithm. The well definedness of the algorithm is analyzed in Section 3. In Section 4, we analyze the global convergence under the weaker assumptions. Finally, numerical experiments are reported.

2. Description of algorithm

The new algorithm divides two parts. The first part will determine a search direction and the other part will provide a strategy which judge the iteration to be successful or not. An appropriate choice of the penalty parameter will guarantee that the search direction can improve the objective function or the measure of constraint violation. The choice of penalty parameter and the determination of the search direction are similar to Byrd et al. [12]. For the convenience, we simply describe it.

Consider the following unconstrained optimization with l_1 exact penalty function

$$\min_{x\in R^n} P(x,\sigma) = f(x) + \sigma v(x), \tag{2.1}$$

where, $\sigma > 0$ is a penalty parameter and

$$v(x) = \sum_{i\in\mathcal{I}} [c_i(x)]^-,$$

where $[c_i(x)]^- = \max\{0, -c_i(x)\}$. $v(x)$ means the measure of constraint violation.

Define the piecewise linear model of the measure of constraint violation $v(x)$ at an iterate x_k as follows

$$m_k(d) = \sum_{i\in\mathcal{I}}[c_i(x_k) + \nabla c_i(x_k)^T d]^- \tag{2.2}$$

and a piecewise quadratic model of the penalty function $P(x, \sigma)$ at x_k

$$q_k(d,\sigma) = f_k + g_k^T d + \frac{1}{2}d^T B_k d + \sigma m_k(d), \tag{2.3}$$

where B_k is obtained through a quasi-Newton update formula or by adding (if necessary) a multiple of the identity to the Hessian of the Lagrangian function of problem (1.1) at x_k.

At the current iterate point x_k with the corresponding to the penalty factor σ_k, we compute the search direction d_k by solving the subproblem

$$\min q_k(d, \sigma_k). \tag{2.4}$$

In practice, we recast (2.4) as the smooth quadratic programming by introducing the slack variables $t_i (i \in \mathcal{I})$,

$$\begin{aligned} \min\quad & f_k + g_k^T d + \frac{1}{2}d^T B_k d + \sigma_k \sum_{i\in\mathcal{I}} t_i, \\ \text{s.t.}\quad & c_i(x_k) + \nabla c_i(x_k)^T d \geq -t_i, \quad i \in \mathcal{I}, \\ & t \geq 0. \end{aligned} \tag{2.5}$$

Denote the solution as $d_k(\sigma_k)$. If $m_k(d_k(\sigma_k)) = 0$, then the linearized constraints are satisfied. Let $\sigma_+ = \sigma_k$ and the current search direction $d_k = d_k(\sigma_+)$. In fact, d_k coincides with the classical SQP search direction in this case. If $m_k(d_k(\sigma_k)) > 0$, then we need to assess the adequacy of the current penalty parameter by computing the lowest possible violation of the linearized constraints in a neighborhood of the current iterate. This is done by solving the problem

$$\begin{aligned} \min\quad & m_k(d), \\ \text{s.t.}\quad & \|d\|_\infty \leq \Delta_k, \end{aligned} \tag{2.6}$$

where $\Delta_k > 0$ is given. The problem (2.6) is equivalent to the following linear programming

$$\begin{aligned} \min\quad & \sum_{i\in\mathcal{I}} t_i, \\ \text{s.t.}\quad & c_i(x_k) + \nabla c_i(x_k)^T d \geq -t_i, \quad i \in \mathcal{I}, \\ & t \geq 0, \qquad \|d\|_\infty \leq \Delta_k. \end{aligned} \tag{2.7}$$

Denote its solution as $d_k(\sigma_\infty)$. If necessary, we choose a new penalty parameter $\sigma_+ > \sigma_k$ such that the solution $d_k(\sigma_+)$ of (2.5) yields a sufficient improvement in linearized feasibility, that is, $d_k(\sigma_+)$ satisfies the following condition

$$m_k(0) - m_k(d_k(\sigma_+)) \geq \delta_1[m_k(0) - m_k(d_k(\sigma_\infty))], \quad \delta_1 \in (0, 1). \tag{2.8}$$

Specifically, $m_k(d_k(\sigma_\infty)) = 0$ when $m_k(d_k(\sigma_+)) = 0$, which implies that (2.8) holds.

Moreover, we should choose a sufficiently large penalty parameter σ_+ such that the solution $d_k(\sigma_+)$ of (2.5) yields a sufficient improvement in the quadratic model $q_k(d, \sigma)$ of the penalty function, that is, $d_k(\sigma_+)$ satisfies the following condition

$$q_k(0, \sigma_+) - q_k(d_k(\sigma_+), \sigma_+) \geq \delta_2\sigma_+[m_k(0) - m_k(d_k(\sigma_\infty))], \quad \delta_2 \in (0, 1). \tag{2.9}$$

If $d_k(\sigma_+)$ satisfies (2.8) and (2.9), then set the search direction $d_k = d_k(\sigma_+)$. Byrd's method [11] execute line search along the search direction d_k by means of the l_1 exact penalty function $P(x, \sigma)$. In other words, let $0 < \alpha_k \leq 1$ be the first member of the sequence $\{1, \beta, \beta^2, \ldots\}$ $(0 < \beta < 1)$ such that

$$P(x_k, \sigma_+) - P(x_k + \alpha_k d_k, \sigma_+) \geq \eta\alpha_k(q_k(0, \sigma_+) - q_k(d_k, \sigma_+)), \quad \eta \in (0, 1).$$

Then set $x_{k+1} = x_k + \alpha_k d_k$.

The strategy of the method presented here is very different from Byrd's one. The new method has two goals to determine whether the current iteration is successful or not. One is improving the feasibility and the other is reducing the value of the objective function. In other words, if

$$g_k^T d_k < 0 \quad \text{and} \quad -\alpha g_k^T d_k > \delta[v_k]^{s_v}, \quad (\delta > 0,\ s_v > 1,\ \alpha \in (0, 1]) \tag{2.10}$$

hold, then, when

$$f(x_k + \alpha d_k) \leq f(x_k) + \eta_f \alpha g_k^T d_k, \quad \eta_f \in (0, 1), \tag{2.11}$$

$$v(x_k + \alpha d_k) \leq v_k^{\max} \tag{2.12}$$

all hold, $x_k(\alpha) = x_k + \alpha d_k$ is acceptable, where $v_k^{\max}$ is the current upper bound on the measure of constraint violation. The current iteration corresponding to the step length α is successful. Let $x_{k+1} = x_k + \alpha d_k$. The current iteration is also called as f-type iteration. At the f-type iteration, the main purpose is to reduce the value of the objective function while keeping the measure of constraint violation within a reasonable bound $v_k^{\max}$.

If (2.10) does not hold, then, when

$$v(x_k) - v(x_k + \alpha d_k) \geq \alpha \eta_v (m_k(0) - m_k(d_k)), \quad \eta_v \in (0, 1) \tag{2.13}$$

holds, $x_k(\alpha)$ is also acceptable. The current iteration corresponding to the step length α is successful. Let $x_{k+1} = x_k + \alpha d_k$. The current iteration is called as c-type iteration. At the c-type iteration, the main purpose is to improve the feasibility. Therefore, the value of the objective function may increase.

If (2.10) does not hold, the upper bound $v_k^{\max}$ on the measure of constraint violation should be updated. The algorithm does it in the following way

$$v_{k+1}^{\max} := max\{\beta_1 v_k^{\max}, v_{k+1} + \beta_2(v_k - v_{k+1})\}, \tag{2.14}$$

where, $\beta_1, \beta_2 \in (0, 1)$.

Now the detailed algorithm is described as follows.

Line Search method with Bi-Object strategy (**LSBO**)

Step 0. Given $x_0 \in R^n$, $B_0 \in R^{n\times n}$, $\delta_1, \delta_2 \in (0, 1)$, $\delta > 0$, $s_v > 1$, $\eta_f, \eta_v, \epsilon, \beta, \beta_1, \beta_2 \in (0, 1)$, $\sigma_{\max} \geq \sigma_0 \geq \sigma_{\min} > 0$, $\Delta_{\max} \geq \Delta_0 \geq \Delta_{\min} > 0$, $k := 0$.

Step 1. Solve (2.5) to get its solution $d_k(\sigma_k)$. If $\|d_k(\sigma_k)\| \leq \epsilon$ and $v_k \leq \epsilon$, then x_k is an approximate solution. Stop.

Step 2. If $m_k(d_k(\sigma_k)) = 0$ and

$$q_k(0, \sigma_k) - q_k(d_k(\sigma_k), \sigma_k) \geq \delta_2 \sigma_k m_k(0),$$

then set $\sigma_+ = \sigma_k$. Go to Step 6.

Step 3. If $m_k(d_k(\sigma_k)) > 0$, then solve (2.7) to get its solution $d_k(\sigma_\infty)$. If $m_k(0) = m_k(d_k(\sigma_\infty)) > 0$, then x_k is an infeasible stationary point. Stop.

Step 4. If $m_k(d_k(\sigma_\infty)) = 0$, then choose $\sigma_+ > \sigma_k$ such that $m_k(d_k(\sigma_+)) = 0$.

Step 5. If (2.8) does not hold, then choose a new $\sigma_+ > \sigma_k$ such that (2.8) holds.

Step 6. Execute line search along $d_k := d_k(\sigma_+)$.

6.1. $\alpha_{k,0} := 1$, $l := 0$;

6.2. If $v(x_k + \alpha_{k,l} d_k) > v_k^{\max}$, then go to Step 6.5;

6.3. If (2.10) holds but (2.11) does not hold, then go to Step 6.5; If (2.10) and (2.11) hold, then go to Step 7;

6.4. If (2.10) and (2.13) do not hold, then go to Step 6.5; If (2.10) does not hold but (2.13) holds, then go to Step 7;

6.5. $\alpha_{k,l+1} := \beta \alpha_{k,l}$, $l := l + 1$. Go to Step 6.2.

Step 7. $\alpha_k := \alpha_{k,l}$, $x_{k+1} := x_k + \alpha_k d_k$, $\sigma_{k+1} = \sigma_+$.

Step 8. Update the trust region radius Δ_k to $\Delta_{k+1} \in [\Delta_{\min}, \Delta_{\max}]$ and $\sigma_{k+1} \in [\sigma_{\min}, \sigma_{\max}]$.

Step 9. If (2.10) does not hold, then update $v_k^{\max}$ by (2.14).

Step 10. Update B_k to B_{k+1}. Set $k := k + 1$. Go to Step 1.

Remark. At the beginning of each iteration, the penalty parameter σ lies in the interval $[\sigma_{\min}, \sigma_{\max}]$, σ_k is a penalty factor which corresponds to the successful f-type or c-type iteration. In other words, the search direction $d_k(\sigma_k)$ satisfies (2.8) and (2.9). Therefore, the penalty parameter is only related to the information at the current iterate point. The sequence of the penalty parameter $\{\sigma_k\}$ is non-monotone, which will avoid some faults resulted from excessively large penalty parameter generated because of unappropriate evaluate at some iterate. Moreover, the sequence of the objective function $\{f(x_k)\}$ and the sequence of the measure of constraint violation $\{v_k\}$ all may be non-monotone. But, the upper bound of the measure of constraint violation $\{v_k^{\max}\}$ is nonincreased monotonely (see Lemma 3.1).

3. Well definedness

From now on, we study the global convergence properties of the algorithm LSBO. We make the following assumptions about the sequence of iterates $\{x_k\}$ and the matrices $\{B_k\}$ generated by the algorithm.

Assumption A. A1 $f(x)$, $c_i(x) (i \in \mathcal{I})$ are twice continuously differentiable.

A2 There exists a bounded convex closed set $\Omega \subseteq R^n$ such that $x_k \in \Omega$ for all k.

A3 The matrices B_k are uniformly positive definite and bounded above, i.e., there exist two constants $0 < \mu_{\min} < \mu_{\max}$ such that

$$\mu_{\min}\|p\|^2 \leq p^T B_k p \leq \mu_{\max}\|p\|^2, \; p \in R^n. \tag{3.1}$$

The following lemma shows the property on the sequence $\{v_k^{\max}\}$ (see [7]).

Lemma 3.1. *The sequence* $\{v_k^{\max}\}$ *is nonincreased monotonely and for any k,* $v_k^{\max} > 0$, $0 \le v_k \le v_k^{\max}$.

The next two results are very important whose proof can be found in [11].

Lemma 3.2. *The following three statements are true.*

(a) $d_k(\sigma) = 0$ *if and only if* x_k *is a stationary point of* $P(x, \sigma)$*;*
(b) $v_k = 0$ *and* x_k *is a stationary point of* $P(x, \sigma)$*, then* x_k *is a stationary point of* (1.1)*;*
(c) x_k *is a stationary point of* $v(x)$ *if and only if* $m_k(0) = m_k(d_k(\sigma_\infty))$.

Lemma 3.3. *Suppose that the algorithm LSBO does not terminate at* x_k*, then there exists* $\sigma_+ \ge \sigma$ *such that* (2.8) *and* (2.9) *hold.*

Now we prove that the algorithm LSBO is well defined.

Lemma 3.4. *Under Assumption A, the algorithm LSBO is well defined, that is, if the algorithm does not terminate at* x_k*, then after reducing the step length finite times, the algorithm can find a successful iterate point.*

Proof. Consider two cases.

(I) $m_k(d_k(\sigma_\infty)) = 0$.

(i) $m_k(0) = m_k(d_k(\sigma_\infty)) = 0$.

Then it follows from the algorithm LSBO that $m_k(d_k(\sigma_+)) = 0$. Thus, x_k is a feasible point and is a stationary point of $v(x)$. Since the algorithm does not stop, $d_k(\sigma_+) \ne 0$. Therefore,

$$q_k(0, \sigma_+) > q_k(d_k(\sigma_+), \sigma_+),$$

which follows that

$$g_k^T d_k(\sigma_+) < -\frac{1}{2} d_k(\sigma_+)^T B_k d_k(\sigma_+) \le -\frac{1}{2}\mu_{\min}\|d_k(\sigma_+)\|^2 < 0.$$

Therefore, (2.10) always holds. Moreover,

$$\begin{aligned} f(x_k + \alpha d_k(\sigma_+)) &= f(x_k) + \alpha g_k^T d_k(\sigma_+) + \frac{\alpha^2}{2} d_k(\sigma_+)^T \nabla^2 f(\xi) d_k(\sigma_+) \\ &= f(x_k) + \alpha\eta_f g_k^T d_k(\sigma_+) + \alpha(1-\eta_f) g_k^T d_k(\sigma_+) + \frac{\alpha^2}{2} d_k(\sigma_+)^T \nabla^2 f(\xi) d_k(\sigma_+) \\ &\le f(x_k) + \alpha\eta_f g_k^T d_k(\sigma_+) - \frac{1}{2}\alpha(1-\eta_f)\mu_{\min}\|d_k(\sigma_+)\|^2 + \frac{\alpha^2}{2} M_f \|d_k(\sigma_+)\|^2, \end{aligned}$$

where, ξ lies between x_k and $x_k + \alpha d_k(\sigma_+)$, $M_f = \sup_{x\in\Omega} \|\nabla^2 f(x)\|$. (2.11) holds as long as $\alpha \le (1-\eta_f)\mu_{\min}/M_f$. It follows from $v_k = 0$ and $v_k^{\max} > 0$ that (2.12) always holds for all sufficiently small $\alpha > 0$. In a word, the algorithm can generate a successful f-type iteration in this case.

(ii) $m_k(0) > m_k(d_k(\sigma_\infty)) = 0$.

By the algorithm LSBO, $m_k(d_k(\sigma_+)) = 0$ and (2.8), (2.9) hold. Suppose, without loss of generality, that α is small enough and (2.10) does not hold. Thus, we only check the condition (2.13). Noting that

$$ared_k^c(\alpha) = v(x_k) - v(x_k + \alpha d_k(\sigma_+)).$$
$$pred_k^c(\alpha) = m_k(0) - m_k(\alpha d_k(\sigma_+)).$$

$$\begin{aligned} |ared_k^c(\alpha) - pred_k^c(\alpha)| &\le \frac{\alpha^2}{2} \sum_{i\in\mathcal{I}} |d_k(\sigma_+)^T \nabla^2 c_i(\xi_{ki}) d_k(\sigma_+)| \\ &\le \frac{\alpha^2}{2} m M_c \|d_k(\sigma_+)\|^2, \end{aligned} \tag{3.2}$$

where ξ_{ki} lies between x_k and $x_k + \alpha d_k(\sigma_+)$, $M_c = \max_{i\in\mathcal{I}} \sup_{x\in\Omega} \|\nabla^2 c_i(x)\|$. It follows from the convexity of the function $m_k(d)$ that

$$\begin{aligned} \frac{|ared_k^c(\alpha) - pred_k^c(\alpha)|}{|pred_k^c(\alpha)|} &\le \frac{0.5\alpha^2 m M_c \|d_k(\sigma_+)\|^2}{m_k(0) - m_k(\alpha d_k(\sigma_+))} \\ &\le \frac{0.5\alpha^2 m M_c \|d_k(\sigma_+)\|^2}{\alpha[m_k(0) - m_k(d_k(\sigma_+))]} \\ &= \frac{0.5\alpha m M_c \|d_k(\sigma_+)\|^2}{m_k(0)}. \end{aligned} \tag{3.3}$$

If α is small enough, then

$$\begin{aligned} v(x_k) - v(x_k + \alpha d_k(\sigma_+)) &\ge \eta_v (m_k(0) - m_k(\alpha d_k(\sigma_+))) \\ &\ge \alpha \eta_v (m_k(0) - m_k(d_k(\sigma_+))), \end{aligned}$$

which implies that the algorithm can generate a successful c-type iteration in this case.

(II) $m_k(d_k(\sigma_\infty)) > 0$.

Since the algorithm does not stop, then $m_k(0) > m_k(d_k(\sigma_\infty)) > 0$. Suppose, without loss of generality, that α is small enough and (2.10) does not hold. Thus, we only check the condition (2.13). Note that (2.8) and (2.9) hold. Similar to (3.3),

$$\begin{aligned} \frac{|ared_k^c(\alpha) - pred_k^c(\alpha)|}{|pred_k^c(\alpha)|} &\le \frac{0.5\alpha^2 m M_c \|d_k(\sigma_+)\|^2}{m_k(0) - m_k(\alpha d_k(\sigma_+))} \\ &\le \frac{0.5\alpha m M_c \|d_k(\sigma_+)\|^2}{\delta_1 [m_k(0) - m_k(d_k(\sigma_\infty))]}. \end{aligned}$$

If α is small enough, then $ared_k^c(\alpha) \ge \eta_v pred_k^c(\alpha)$, which implies that the algorithm can generate a successful c-type iteration in this case.

Therefore, the result is proved. □

4. Global convergence

Denote the index set

$$\mathcal{C} = \{k |\ v_{k+1}^{\max} \ne v_k^{\max}\}.$$

Definition 4.1. A point x^* satisfies the MFCQ condition if x^* is feasible for (1.1) and there is a unit direction s such that

$$\nabla c_i(x^*)^T s > 0, \quad i \in \mathcal{A}(x^*),$$

where $\mathcal{A}(x^*) = \{i \in \mathcal{I} | c_i(x^*) = 0\}$.

Lemma 4.2. *If $|\mathcal{C}| < +\infty$, then $\lim_k v(x_k) = 0$, which implies that any accumulation point of $\{x_k\}$ is feasible.*

Proof. By $|\mathcal{C}| < +\infty$, there exists an index k_0 such that $v_k^{\max} = v_{k_0}^{\max} > 0$ for all $k \ge k_0$, which implies that all iterations are f-type iteration for $k \ge k_0$. By (2.10) and (2.11),

$$f_k - f_{k+1} \ge \eta_f (-\alpha_k g_k^T d_k) > \eta_f \delta v_k^{s_v},$$

which follows that

$$f_{k_0} - f_{k+1} > \eta_f \delta \sum_{i=k_0}^{k} v_i^{s_v}.$$

So $\lim_{k\to\infty} v_k = 0$. □

Lemma 4.3. *Let x^* be a feasible point that satisfies MFCQ. If x^* is not a KKT point, then there exist a neighborhood $\mathcal{N}(x^*)$ of x^* and constants $b_1(x^*) > 0$, $\bar{\alpha} \in (0, 1]$ such that*

$$g(x_k)^T d_k \le -0.5 \bar{\alpha} b_1(x^*)$$

holds for all $x_k \in \mathcal{N}(x^*)$.

Proof. By the assumption, there exist a neighborhood $\mathcal{N}(x^*)$ of x^*, a constant $b_1(x^*) > 0$ and a unit vector s such that, for all $x_k \in \mathcal{N}(x^*)$,

$$\begin{aligned} &g(x_k)^T s \le -b_1(x^*), \\ &\nabla c_i(x_k)^T s \ge b_1(x^*), \quad i \in \mathcal{A}(x^*). \end{aligned}$$

If $i \in \mathcal{A}(x^*)$, then

$$c_i(x_k) + \nabla c_i(x_k)^T (\alpha s) \ge 0$$

as long as $\alpha \ge v_k / b_1(x^*)$. If $i \notin \mathcal{A}(x^*)$, then there exists a constant $b_1'(x^*) > 0$ such that

$$c_i(x_k) \ge b_1'(x^*), \quad \forall x_k \in \mathcal{N}(x^*).$$

It follows from

$$c_i(x_k) + \nabla c_i(x_k)^T(\alpha s) \geq b_1'(x^*) - \alpha\|\nabla c_i(x_k)\| \geq b_1'(x^*) - \alpha M_c$$

that $c_i(x_k) + \nabla c_i(x_k)^T(\alpha s) \geq 0$ as long as $\alpha \leq b_1'(x^*)/M_c$. Let

$$\bar{\alpha} = \min\left\{1, \frac{b_1'(x^*)}{M_c}, \frac{b_1(x^*)}{\mu_{\max}}\right\}.$$

If the neighborhood $\mathcal{N}(x^*)$ is sufficiently small, then $v_k/b_1(x^*) < \bar{\alpha}$. Therefore,

$$\begin{aligned} g_k^T d_k &\leq g_k^T d_k + \frac{1}{2}d_k^T B_k d_k + \sigma_k m_k(d_k) \\ &\leq \frac{1}{2}g_k^T(\bar{\alpha}s) + \frac{1}{2}g_k^T(\bar{\alpha}s) + \frac{\bar{\alpha}^2}{2}s^T B_k s + \sigma_k m_k(\bar{\alpha}s) \\ &\leq -\frac{\bar{\alpha}}{2}b_1(x^*) - \frac{\bar{\alpha}}{2}[b_1(x^*) - \bar{\alpha}\mu_{\max}] \\ &\leq -\frac{\bar{\alpha}}{2}b_1(x^*). \end{aligned}$$

The proof is completed. □

Lemma 4.4. *Suppose that $|\mathcal{C}| < +\infty$. Then any limit point of $\{x_k\}$ is either a KKT point of (1.1) or is a feasible limit point where MFCQ fails.*

Proof. Let x^* be any limit point of $\{x_k\}$. Then there exists an infinite subsequence $\{x_{k_i}\}$ such that $\lim_{i\to\infty} x_{k_i} = x^*$. By Lemma 4.2, $v(x^*) = 0$. Suppose that x^* is not a KKT point and MFCQ holds at x^*. By Lemma 4.3, there is a neighborhood $\mathcal{N}(x^*)$ of x^* and constants $b_1(x^*) > 0, \bar{\alpha} \in (0, 1]$ such that

$$g(x_{k_i})^T d_{k_i} \leq -0.5\bar{\alpha}b_1(x^*)$$

holds for all $x_{k_i} \in \mathcal{N}(x^*)$. By $|\mathcal{C}| < +\infty$, there exists an index k_0 such that $v_k^{\max} = v_{k_0}^{\max} > 0$ for all $k \geq k_0$, which implies that all iterations are f-type iteration for $k \geq k_0$. Suppose, without loss of generality, that $k_1 \geq k_0$. Therefore,

$$\begin{aligned} f(x_{k_{i+1}}) - f(x_{k_i}) &\leq f(x_{k_i+1}) - f(x_{k_i}) \\ &\leq \eta_f \alpha_{k_i} g(x_{k_i})^T d_{k_i} \leq -0.5\eta_f b_1(x^*)\bar{\alpha}\alpha_{k_i}, \end{aligned}$$

which implies that $\lim_{i\to+\infty} \alpha_{k_i} = 0$. Since $\lim_{k_i\to\infty} v(x_{k_i}) = 0$ and $\|d_k\|$ is bounded above,

$$v(x_{k_i} + \alpha_{k_i} d_{k_i}) \leq v_{k_0}^{\max}$$

holds for all sufficiently large k_i. Moreover,

$$\begin{aligned} f(x_{k_i} + \alpha d_{k_i}) &= f(x_{k_i}) + \alpha g_{k_i}^T d_{k_i} + \frac{\alpha^2}{2} d_{k_i}^T \nabla^2 f(\xi) d_{k_i} \\ &\leq f(x_{k_i}) + \alpha\eta_f g_{k_i}^T d_{k_i} + \alpha(1-\eta_f) g_{k_i}^T d_{k_i} + \frac{\alpha^2}{2} M_f M_d^2 \\ &\leq f(x_{k_i}) + \alpha\eta_f g_{k_i}^T d_{k_i} - 0.5\alpha(1-\eta_f)\bar{\alpha}b_1(x^*) + \frac{\alpha^2}{2} M_f M_d^2 \\ &\leq f(x_{k_i}) + \alpha\eta_f g_{k_i}^T d_{k_i} \end{aligned}$$

as long as $\alpha \leq (1-\eta_f)\bar{\alpha}b_1(x^*)/(M_f M_d^2)$, where ξ is between x_{k_i} and $x_{k_i} + \alpha d_{k_i}$, $\|d_k\| \leq M_d$ for all k.

By the algorithm LSBO,

$$f(x_{k_i} + \alpha_{k_i} d_{k_i}) \leq f(x_{k_i}) + \eta_f \alpha_{k_i} g_{k_i}^T d_{k_i}$$

and

$$f(x_{k_i} + \beta^{-1}\alpha_{k_i} d_{k_i}) > f(x_{k_i}) + \eta_f \beta^{-1}\alpha_{k_i} g_{k_i}^T d_{k_i},$$

which contradicts with $\lim_{i\to\infty} \beta^{-1}\alpha_{k_i} = 0$. So the result is true. □

Lemma 4.5. *Suppose that $|\mathcal{C}| = +\infty$ and $\lim_{k\to\infty} v_k^{\max} = \tilde{v}$. If x^* is any limit point of $\{x_k\}_{k\in\mathcal{C}}$, then $v(x^*) = \tilde{v}$.*

Proof. Since x^* is a limit point of $\{x_k\}_{k\in\mathcal{C}}$, there exists an infinite index subset $\mathcal{K} \subseteq \mathcal{C}$ such that $\lim_{k\in\mathcal{K}} x_k = x^*$. It follows from $\lim_{k\to\infty} v_k^{\max} = \tilde{v}$ that there exists a positive integer k_1 such that, for all $k \in \mathcal{C}, k \geq k_1$,

$$\begin{aligned} v_{k+1}^{\max} &= \beta_2 v_k + (1-\beta_2)v_{k+1} \\ &\leq \beta_2 v_k + (1-\beta_2)v_k - (1-\beta_2)\alpha_k\eta_v[m_k(0) - m_k(d_k)] \\ &\leq v_k - \alpha_k(1-\beta_2)\eta_v\delta_1[m_k(0) - m_k(d_k(\sigma_\infty))], \end{aligned}$$

which follows that

$$0 \leq \alpha_k(1-\beta_2)\eta_v\delta_1[m_k(0) - m_k(d_k(\sigma_\infty))] \leq v_k - v_{k+1}^{\max}, \quad k \in \mathcal{C},\ k \geq k_1. \tag{4.1}$$

Let $k \in \mathcal{K}, k \to \infty$, then

$$0 \leq v(x^*) - \tilde{v}.$$

On the other hand, $v(x_k) \leq v_k^{\max}$ for all k. So $v(x^*) \leq \tilde{v}$. Thus, the result is true. □

Lemma 4.6. *Suppose that $|\mathcal{C}| = +\infty$ and $\lim_{k\to\infty} v_k^{\max} = \tilde{v} > 0$. Then any limit point of $\{x_k\}_{k\in\mathcal{C}}$ is an infeasible stationary point.*

Proof. Let x^* be a limit point of $\{x_k\}_{k\in\mathcal{C}}$, then there exists an infinite index subset $\mathcal{K} \subseteq \mathcal{C}$ such that $\lim_{k\in\mathcal{K}} x_k = x^*$. By Lemma 4.5, $v(x^*) = \tilde{v} > 0$. If x^* is not an infeasible stationary point, then $m_*(0) - m_*(d_*) > 0$, where d_* is a solution of the following subproblem

$$\begin{aligned} \min \quad & m_*(d) = \sum_{i\in\mathcal{I}} [c_i(x^*) + \nabla c_i(x^*)^T d]^-, \\ \text{s.t.} \quad & \|d\|_\infty \leq \Delta_{\min}. \end{aligned}$$

Obviously, d_* is a feasible solution of (2.6). So, for all sufficiently large $k \in \mathcal{K}$,

$$m_k(0) - m_k(d_k(\sigma_\infty)) \geq m_k(0) - m_k(d_*) \geq \frac{1}{2}(m_*(0) - m_*(d_*)) > 0,$$

which implies from (4.1) that $\lim_{k\in\mathcal{K}} \alpha_k = 0$. According to the algorithm LSBO, for all sufficiently large $k \in \mathcal{K} \subseteq \mathcal{C}$,

$$v(x_k) - v(x_k + \alpha_k d_k) \geq \alpha_k \eta_v [m_k(0) - m_k(d_k)]$$

and

$$v(x_k) - v(x_k + \beta^{-1}\alpha_k d_k) < \beta^{-1}\alpha_k\eta_v[m_k(0) - m_k(d_k)]. \tag{4.2}$$

Moreover,

$$\begin{aligned} pred_k^c(\beta^{-1}\alpha_k) &= m_k(0) - m_k(\beta^{-1}\alpha_k d_k) \\ &\geq \beta^{-1}\alpha_k(m_k(0) - m_k(d_k)) \\ &\geq \delta_1\beta^{-1}\alpha_k(m_k(0) - m_k(d_k(\sigma_\infty))). \end{aligned} \tag{4.3}$$

Since x^* is not a stationary point of $v(x)$, then,

$$m_*(0) > m_*(d_*(\Delta_{\min})) \geq 0,$$

where $m_*(d) = \sum_{i\in\mathcal{I}}[c_i(x^*) + \nabla c_i(x^*)^T d]^-$. Similar to (3.2), we have that

$$\begin{aligned} \frac{|ared_k^c(\beta^{-1}\alpha_k) - pred_k^c(\beta^{-1}\alpha_k)|}{|pred_k^c(\beta^{-1}\alpha_k)|} &\leq \frac{0.5\beta^{-2}\alpha_k^2 m M_c \|d_k\|^2}{\beta^{-1}\alpha_k\delta_1[m_k(0) - m_k(d_k(\sigma_\infty))]} \\ &\leq \frac{\beta^{-1}\alpha_k m M_c M_d^2}{\delta_1[m_*(0) - m_*(d_*(\Delta_{\min}))]}, \end{aligned}$$

which follows that

$$ared_k^c(\beta^{-1}\alpha_k) \geq \eta_v pred_k^c(\beta^{-1}\alpha_k)$$

holds for all sufficiently large $k \in \mathcal{K}$, which is a contradiction with (4.2). So $m_*(0) = m_*(d_*(\Delta_{\min})) > 0$, i.e., x^* is an infeasible stationary point of $v(x)$. □

Lemma 4.7. *Suppose that $|\mathcal{C}| = +\infty$ and $\lim_{k\to\infty} v_k^{\max} = 0$. Then any limit point of $\{x_k\}_{k\in\mathcal{C}}$ is either a KKT point of (1.1) or is a feasible limit point where MFCQ fails.*

Proof. Let $\mathcal{K} \subseteq \mathcal{C}$ be an infinite index subset and $\lim_{k\in\mathcal{K}} x_k = x^*$. By Lemma 4.5, $v(x^*) = 0$. Suppose, by contradiction, that x^* is not a KKT point and MFCQ holds at x^*. By Lemma 4.3, there is a neighborhood $\mathcal{N}(x^*)$ of x^* and constants $b_1(x^*) > 0$, $\bar{\alpha} \in (0, 1]$ such that

$$g(x_k)^T d_k \le -0.5\bar{\alpha} b_1(x^*) \tag{4.4}$$

holds for all $x_k \in \mathcal{N}(x^*)$.

For $x_k \in \mathcal{N}(x^*)$ and $k \in \mathcal{K}$, if $v(x_k) = 0$, then, by (4.4),

$$-\alpha g(x_k)^T d_k > \delta v_k^{\delta_v} = 0, \quad \forall \alpha \in (0, 1],$$

which implies that the current iteration is f-type iteration. This contradicts $k \in \mathcal{C}$. Therefore, $v(x_k) > 0$.

By the proof of Lemma 4.3, there exists a unit vector s such that, for all $x_k \in \mathcal{N}(x^*)$,

$$g(x_k)^T s \le -b_1(x^*),$$
$$\nabla c_i(x_k)^T s \ge b_1(x^*), \quad i \in \mathcal{A}(x^*).$$

If $i \in \mathcal{A}(x^*)$, then

$$c_i(x_k) + \nabla c_i(x_k)^T(\alpha s) \ge 0$$

as long as $\alpha \ge v_k/b_1(x^*)$. If $i \notin \mathcal{A}(x^*)$, then there exists a constant $b_1'(x^*) > 0$ such that

$$c_i(x_k) \ge b_1'(x^*), \quad \forall x_k \in \mathcal{N}(x^*),\ k \in \mathcal{K}.$$

It follows from

$$c_i(x_k) + \nabla c_i(x_k)^T(\alpha s) \ge b_1'(x^*) - \alpha\|\nabla c_i(x_k)\| \ge b_1'(x^*) - \alpha M_c$$

that $c_i(x_k) + \nabla c_i(x_k)^T(\alpha s) \ge 0$ as long as $\alpha \le b_1'(x^*)/M_c$. Therefore, if

$$v_k/b_1(x^*) \le \alpha \le \min\{b_1'(x^*)/M_c, \Delta_{\min}\},$$

then $0 \le m_k(d_k(\sigma_\infty)) \le m_k(\alpha s) = 0$, i.e., $m_k(d_k(\sigma_\infty)) = 0$, which implies that $m_k(d_k) = 0$. By (3.2), it follows that

$$\begin{aligned} v(x_k + \alpha d_k) &\le m_k(\alpha d_k) + 0.5\alpha^2 m M_c \|d_k\|^2 \\ &\le (1-\alpha)v_k + 0.5\alpha^2 m M_c M_d^2. \end{aligned}$$

If $\alpha \le v_k/(m M_c M_d^2)$, then

$$\begin{aligned} v(x_k + \alpha d_k) &\le (1-0.5\alpha)v_k - 0.5\alpha(v_k - \alpha m M_c M_d^2) \\ &\le (1-0.5\alpha)v_k \le v(x_k) \le v_k^{\max}, \end{aligned}$$

i.e., (2.12) holds. It follows from (4.4) that

$$-\alpha g(x_k)^T d_k \ge 0.5\bar{\alpha}\alpha b_1(x^*) > \delta v_k^{s_v}$$

holds as long as $\alpha > 2\delta v_k^{s_v}/(\bar{\alpha} b_1(x^*))$. By $\lim_k v(x_k) = 0$ and $s_v > 1$,

$$\frac{2\delta v_k^{s_v}}{\bar{\alpha} b_1(x^*)} < \alpha < \frac{v_k}{m M_c M_d^2} < \bar{\alpha} \tag{4.5}$$

holds for sufficiently large $k \in \mathcal{K}$, which shows that (2.10) holds if α satisfies (4.5). Moreover,

$$\begin{aligned} f(x_k + \alpha d_k) &\le f(x_k) + \alpha\eta_f g_k^T d_k - 0.5\alpha(1-\eta_f)\bar{\alpha} b_1(x^*) + \frac{\alpha^2}{2} M_f M_d^2 \\ &\le f(x_k) + \alpha\eta_f g_k^T d_k \end{aligned}$$

holds if $\alpha \le (1-\eta_f)\bar{\alpha} b_1(x^*)/(M_f M_d^2)$. Let

$$\hat{\alpha} = \min\{\bar{\alpha}, (1-\eta_f)\bar{\alpha} b_1(x^*)/(M_f M_d^2)\}.$$

If $k \in \mathcal{K}$ is sufficiently large such that

$$\frac{2\delta v_k^{s_v}}{\bar{\alpha} b_1(x^*)} < \alpha < \frac{v_k}{m M_c M_d^2} < \hat{\alpha},$$

then αd_k satisfies (2.10)–(2.12). Hence, the current iteration is f-type iteration. This contradicts $k \in \mathcal{C}$. Thus the lemma is proved. □

Table 5.1
Summary of results.

Name	Alg.	it	nf	ng	Qps	Feas	KKT	$f(x_k)$
Degenerate2	LSBO	9	10	10	12	0.00e+00	8.67e−07	0.0260
$x_0 = (1, 1)$	LSEP	9	10	10	13	0.00e+00	8.67e−07	0.0260
Degenerate3	LSBO	9	10	10	22	1.19e−16	8.67e−07	0.0006
$x_0 = (1, 1)$	LSEP	9	10	10	12	1.19e−16	8.67e−07	0.0006
Infeasible1	LSBO	2	3	3	3	1.00e+00	1.00e+00	0.0000
$x_0 = 10$	LSEP	2	3	3	3	1.00e+00	1.00e+00	0.0000
Infeasible2	LSBO	6	7	7	17	0.00e+00	6.54e−06	1.0000
$x_0 = (0, -1)$	LSEP	6	7	7	10	0.00e+00	6.54e−06	1.0000
Infeasible3	LSBO	3	4	4	5	1.50e+00	2.13e−14	5.0000
$x_0 = (0, 0)$	LSEP	4	8	5	6	1.50e+00	3.55e−15	5.0000
Mpcc1	LSBO	3	4	4	6	1.87e−16	1.53e−09	1.0000
$x_0 = (0.1, 0.9)$	LSEP	3	4	4	6	1.87e−16	1.53e−09	1.0000
Mpcc3	LSBO	8	9	9	15	6.88e−33	4.44e−16	−2.0000
$x_0 = (-0.1, 0.5)$	LSEP	8	9	9	11	7.19e−33	1.48e−14	−2.0000
Mpcc5	LSBO	7	8	8	16	1.37e−20	7.79e−07	0.5000
$x_0 = (0.1, 0.1)$	LSEP	8	9	9	12	5.66e−24	5.06e−07	0.5000
Mpcc6	LSBO	10	11	11	12	9.53e−07	4.76e−07	−0.0000
$x_0 = (1, 1)$	LSEP	10	11	11	12	9.53e−07	4.76e−07	−0.0000
NLP1	LSBO	5	6	6	6	4.51e−12	1.12e−12	−1.4142
$x_0 = (0, 0)$	LSEP	5	6	6	6	4.51e−12	1.12e−12	−1.4142
NLP2	LSBO	1	2	2	3	5.55e−17	2.22e−16	0.3125
$x_0 = (0, 0)$	LSEP	1	2	2	3	5.55e−17	2.22e−16	0.3125
Switch-off1	LSBO	1	3	2	4	0.00e+00	1.77e−15	1.0000
$x_0 = (0, 0)$	LSEP	1	3	2	3	0.00e+00	1.77e−15	1.0000
Switch-off2	LSBO	3	4	4	8	0.00e+00	1.96e−07	6.0039
$x_0 = (0, 5)$	LSEP	3	4	4	8	0.00e+00	1.96e−07	6.0039
Switch-off3	LSBO	11	12	12	13	4.62e−09	4.66e−17	−1.0000
$x_0 = (0, 0)$	LSEP	38	213	39	45	8.51e−12	2.27e−07	−0.0019
Switch-off4	LSBO	9	10	10	13	9.80e−13	4.61e−12	−29.3137
$x_0 = (0, 0)$	LSEP	10	16	11	12	4.89e−12	7.24e−12	−29.3137
Unbounded1	LSBO	1	2	2	4	0.00e+00	2.22e−16	2.0000
$x_0 = 0$	LSEP	1	2	2	3	0.00e+00	2.22e−16	2.0000
Unbounded2	LSBO	2	3	3	4	0.00e+00	4.44e−16	−1.0000
$x_0 = 0$	LSEP	2	3	3	4	0.00e+00	4.44e−16	−1.0000
Wächter1	LSBO	3	4	4	5	0.00e+00	1.11e−16	1.0000
$x_0 = -3$	LSEP	3	4	4	5	0.00e+00	7.77e−16	1.0000
Wächter2	LSBO	4	5	5	6	0.00e+00	1.11e−16	1.0000
$x_0 = -3$	LSEP	4	5	5	6	3.33e−16	7.77e−16	1.0000

We can now give the main convergence result.

Theorem 4.8. *Suppose that the algorithm LSBO generates an infinite sequence of iterates* $\{x_k\}$ *and Assumption A holds. Then,*

(a) *If* $|\mathcal{C}| < +\infty$, *any limit point of* $\{x_k\}$ *is either a KKT point of (1.1) or is a feasible limit point where MFCQ fails.*
(b) *If* $|\mathcal{C}| = +\infty$ *and* $\lim_{k\to\infty} v_k^{\max} > 0$, *any limit point of* $\{x_k\}_{k\in\mathcal{C}}$ *is an infeasible stationary point.*
(c) *If* $|\mathcal{C}| = +\infty$ *and* $\lim_{k\to\infty} v_k^{\max} = 0$, *any limit point of* $\{x_k\}_{k\in\mathcal{C}}$ *is either a KKT point of (1.1) or is a feasible limit point where MFCQ fails.*

5. Numerical experiments

In this section, we developed a matlab implementation of the algorithm LSBO and tested its performance on several difficult situations in [17]. We report the numerical results on all examples with only inequality constraints in [17]. The test set is divided into five groups of problems: (1) the problems that give rise to inconsistent linearization of the constraints; (2) the problems that MFCQ is violated at the solution; (3) the problems that is infeasible; (4) the problems where the l_1 exact penalty function is unbounded for small penalty parameters; (5) some regular and simple problems. For comparison, we also run Algorithm I in [11]—a line search exact penalty (LSEP) method.

To solve the subproblems (2.5) and (2.7), we employed the codes provided by the matlab optimization toolbox. The linear program (2.7) was solved using linprog and the quadratic program (2.5), using quadprog. The matrix B_k is symmetric positive definite which is obtained similarly to [11], i.e., by adding (if necessary) a multiple of the identity to the Hessian matrix of

Table 5.2
Results of Mpcc1 for $x_0 = (0.1, 0.9)$.

σ_0	Alg.	it	nf	ng	Qps	Feas	KKT	$f(x_k)$
1	LSBO	3	4	4	6	1.87e−16	1.53e−09	1.0000
	LSEP	3	4	4	6	1.87e−16	1.53e−09	1.0000
10	LSBO	2	3	3	3	0.00e+00	8.45e−07	1.0000
	LSEP	3	4	4	4	0.00e+00	1.05e−09	1.0000
10^2	LSBO	3	4	4	4	0.00e+00	1.16e−14	1.0000
	LSEP	3	4	4	4	0.00e+00	1.16e−08	1.0000
10^3	LSBO	3	4	4	4	0.00e+00	4.49e−08	1.0000
	LSEP	8	9	9	9	0.00e+00	6.65e−12	1.0000
10^4	LSBO	3	4	4	4	0.00e+00	7.98e−08	1.0000
	LSEP	58	59	59	59	0.00e+00	1.81e−12	1.0000
10^5	LSBO	3	4	4	4	0.00e+00	8.40e−08	1.0000
	LSEP	556	557	557	557	0.00e+00	1.45e−11	1.0000
10^6	LSBO	3	4	4	4	0.00e+00	8.45e−08	1.0000
	LSEP	F	F	F	F	0.00e+00	1.00	1.0000
10^7	LSBO	3	4	4	4	0.00e+00	8.45e−08	1.0046
	LSEP	F	F	F	F	0.00e+00	1.00	1.0000
10^8	LSBO	3	4	4	4	0.00e+00	8.45e−08	1.0055
	LSEP	F	F	F	F	0.00e+00	1.00	1.0055

the Lagrangian function of the problem (1.1) at x_k. The initial penalty σ_0 is set to 1 in all tests. We multiply σ by 10 every time it has to be increased. We trigger the solution of the linear programming if $m_k(d_k) > 1.0\text{e}{-6}$. For the algorithm LSBO, we have used the following parameter settings

$$\delta_1 = \delta_2 = 0.1, \quad \delta = 10, \quad s_v = 2.1, \quad \eta_f = \eta_v = 1.0\text{e}{-4},$$
$$\beta = 0.5, \quad \beta_1 = 0.9, \quad \beta_2 = 0.75, \quad \sigma_{\min} = 1, \quad \sigma_{\max} = 1.0\text{e}{+8}.$$

For LSEP in [11], we have used the following parameter settings which are the same as [11].

$$\delta_1 = \delta_2 = 0.1, \quad \beta = 0.5, \quad \eta = 1.0\text{e}{-4}, \quad \eta_1 = 0.25, \quad \eta_2 = 0.75.$$

Set $\Delta_{\min}$, Δ_0, $\Delta_{\max}$ to 10^{-3}, 1, 10^3, respectively.

The numerical results are summarized in Table 5.1, which report the number of iterations (it), the number of the objective function estimations (nf), the number of the gradient estimations (ng), the number of quadratic programs solved (Qps), the KKT error (KKT, infinity norm) and feasibility errors (Feas, l_1 norm), as well as the value of the objective function $f(x_k)$ when the algorithm stops. The algorithms LSBO and LSEP stop if

$$\max\{Feas, KKT\} \le \epsilon = 10^{-6} \quad \text{or} \quad it > 1000.$$

"F" means that the algorithm fails, i.e., $it > 1000$.

In Table 5.1, LSBO can find the solutions of all problems although the number of quadratic programs solved (Qps) by LSBO is slightly more than it by LSEP. LSEP does not obtain the solution of problem Switch-off 3. In order to test the performance of the algorithm LSBO further, we run two algorithms LSBO and LSEP from other initial point and the various initial penalty settings $\sigma_0 = 10^0, 10^1, \dots, 10^8$. LSEP is very sensitive to the initial point and the choice of parameters for some test problems such as Mpcc1, Switch-off 2, Switch-off 3 and Switch-off 4. Tables 5.2–5.5 report their numerical results. The results in Tables 5.2–5.5 suggest that setting large values for the penalty parameter, a strategy that has no theoretical drawbacks and that is sometimes to consider a "safe" approach in practice, may harm the performance of an exact penalty method.

For the problem Mpcc1 (see Table 5.2), LSBO and LSEP all can find the solution of Mpcc1 but the number of iterations for LSEP is more and more as the enlargement of σ_0 even over 1000 iterations. For the problem Switch-off 2 (see Table 5.3), LSEP all cannot find the solution of Switch-off 2 when it runs from the initial point $x_0 = (0.001, 5.001)$ and the various initial penalty settings $\sigma_0 = 10^0, 10^1, \dots, 10^8$ although they stop because of $\max\{Feas, KKT\} \le 10^{-6}$. In fact, the iterate sequence $\{x_k\}$ generated by LSEP all goes around the point $(0, 5)$. However, the algorithm LSBO all can get the solution normally in the various cases. For the problem Switch-off 3 (see Tables 5.4a–5.4f), LSBO all can find the solution easily for the different initial points and the various initial penalty settings. LSEP cannot find the solution from $x_0 = (0, 0)$ and can find the solution from $x_0 = (0.1, -0.1), (-0.1, 0.1)$ only for the smaller initial penalty settings. For the problem Switch-off 4 (see Table 5.5), LSEP fails for the larger initial penalty settings while LSBO always succeeds.

6. Conclusion

In this paper, we have proposed a line search exact penalty method with bi-object strategy (LSBO) for nonlinear constrained optimization. The search direction is computed by a piecewise quadratic model of the exact penalty function.

Table 5.3
Results of Switch-off 2 for $x_0 = (0.001, 5.001)$.

σ_0	Alg.	it	nf	ng	Qps	Feas	KKT	$f(x_k)$
1	LSBO	5	6	6	14	0.00e+00	6.45e−08	6.0012
	LSEP	1	10	2	6	0.00e+00	1.77e−07	10.0044
10	LSBO	5	6	6	13	0.00e+00	6.45e−08	6.0012
	LSEP	1	10	2	5	0.00e+00	1.77e−07	10.0044
10^2	LSBO	5	6	6	12	0.00e+00	6.45e−08	6.0012
	LSEP	1	10	2	4	0.00e+00	1.77e−07	10.0044
10^3	LSBO	5	6	6	11	0.00e+00	6.45e−08	6.0012
	LSEP	1	10	2	3	0.00e+00	1.77e−07	10.0044
10^4	LSBO	5	6	6	10	0.00e+00	6.45e−08	6.0012
	LSEP	1	10	2	2	0.00e+00	1.77e−07	10.0044
10^5	LSBO	5	6	6	10	0.00e+00	6.45e−08	6.0012
	LSEP	1	10	2	2	0.00e+00	1.77e−07	10.0044
10^6	LSBO	5	6	6	10	0.00e+00	6.45e−08	6.0012
	LSEP	1	10	2	2	0.00e+00	1.77e−07	10.0044
10^7	LSBO	5	6	6	10	0.00e+00	6.45e−08	6.0012
	LSEP	1	10	2	2	0.00e+00	1.77e−07	10.0044
10^8	LSBO	5	6	6	10	0.00e+00	6.45e−08	6.0012
	LSEP	1	10	2	2	0.00e+00	1.77e−07	10.0044

Table 5.4a
Results of Switch-off 3 for LSBO. $x_0 = (0, 0)$, $\sigma_0 = 1, 10, \ldots, 10^8$.

it	nf	ng	Qps	Feas	KKT	$f(x_k)$
11	12	12	13	4.62e−09	4.66e−17	−1.0000

Table 5.4b
Results of Switch-off 3 for LSEP. $x_0 = (0, 0)$.

σ_0	it	nf	ng	Qps	Feas	KKT	$f(x_k)$
1	38	213	39	45	8.51e−12	2.27e−07	−0.0019
10	38	213	39	44	8.51e−12	2.27e−07	−0.0019
10^2	38	213	39	43	8.51e−12	2.27e−07	−0.0019
10^3	38	213	39	42	8.51e−12	2.27e−07	−0.0019
10^4	38	213	39	41	8.51e−12	2.27e−07	−0.0019
10^5	38	213	39	40	8.51e−12	2.27e−07	−0.0019
10^6	38	213	39	39	8.51e−12	2.27e−07	−0.0019
10^7	43	281	44	44	3.13e−15	1.48e−08	−0.0004
10^8	46	283	47	47	5.79e−15	1.80e−09	−0.0001

Table 5.4c
Results of Switch-off 3 for LSBO. $x_0 = (0.1, -0.1)$, $\sigma_0 = 1, 10, \ldots, 10^8$.

it	nf	ng	Qps	Feas	KKT	$f(x_k)$
11	12	12	14	1.58e−09	0.00e+00	−1.0000

Table 5.4d
Results of Switch-off 3 for LSEP. $x_0 = (0.1, -0.1)$.

σ_0	it	nf	ng	Qps	Feas	KKT	$f(x_k)$
1	20	48	21	22	6.29e−11	4.44e−16	−1.0000
10	20	48	21	21	3.37e−10	3.73e−16	−1.0000
10^2	39	268	40	44	2.14e−12	9.22e−08	−0.0012
10^3	39	268	40	43	2.14e−12	9.22e−08	−0.0012
10^4	39	268	40	42	2.13e−12	9.22e−08	−0.0012
10^5	39	268	40	41	2.14e−12	9.22e−08	−0.0012
10^6	39	268	40	40	2.12e−12	9.22e−08	−0.0012
10^7	43	284	44	44	1.27e−18	2.10e−08	−0.0005
10^8	42	223	43	43	2.22e−18	2.44e−09	−0.0001

Table 5.4e
Results of Switch-off 3 for LSBO. $x_0 = (-0.1, 0.1)$, $\sigma_0 = 1, 10, \dots, 10^8$.

it	nf	ng	Qps	Feas	KKT	$f(x_k)$
12	13	13	16	8.40e−09	2.22e−16	−1.0000

Table 5.4f
Results of Switch-off 3 for LSEP. $x_0 = (-0.1, 0.1)$.

σ_0	it	nf	ng	Qps	Feas	KKT	$f(x_k)$
1	21	49	22	23	5.51e−10	1.71e−14	−1.0000
10	12	20	13	13	1.10e−09	0.00e+00	−1.0000
10^2	103	633	104	104	2.78e−07	2.98e−15	−1.0000
10^3	32	157	33	36	8.70e−12	4.50e−08	−0.0008
10^4	32	157	33	35	8.70e−12	4.50e−08	−0.0008
10^5	32	157	33	34	8.69e−12	4.50e−08	−0.0008
10^6	32	157	33	33	8.69e−12	4.50e−08	−0.0008
10^7	42	274	43	43	2.01e−14	1.73e−08	−0.0005
10^8	42	255	43	43	6.62e−18	1.90e−09	−0.0001

Table 5.5
Results of Switch-off 4 for $x_0 = (0, 0)$.

σ_0	Alg.	it	nf	ng	Qps	Feas	KKT	$f(x_k)$
1	LSBO	9	10	10	13	9.80e−13	4.61e−12	−29.3137
	LSEP	10	16	11	12	4.89e−12	7.24e−12	−29.3137
10	LSBO	8	9	9	12	3.80e−09	5.62e−09	−29.3137
	LSEP	9	14	10	10	6.83e−12	1.01e−11	−29.3137
10^2	LSBO	6	7	7	9	8.79e−10	1.29e−09	−29.3137
	LSEP	116	119	117	117	1.12e−11	1.65e−11	−29.3137
10^3	LSBO	6	7	7	9	1.26e−07	1.86e−07	−29.3137
	LSEP	F	F	F	F	0.00e+00	8.97e+00	−28.9602
10^4	LSBO	6	7	7	9	7.71e−08	3.67e−07	−29.3137
	LSEP	F	F	F	F	0.00e+00	7.75e+00	−25.6806
10^5	LSBO	6	7	7	9	7.33e−08	1.08e−07	−29.3137
	LSEP	F	F	F	F	0.00e+00	7.35e+00	−25.0755
10^6	LSBO	6	7	7	9	7.29e−08	1.07e−07	−29.3137
	LSEP	F	F	F	F	0.00e+00	7.30e+00	−25.0076
10^7	LSBO	6	7	7	9	7.29e−08	1.07e−07	−29.3137
	LSEP	F	F	F	F	0.00e+00	7.29e+00	−25.0007
10^8	LSBO	6	7	7	9	7.29e−08	1.07e−07	−29.3137
	LSEP	F	F	F	F	0.00e+00	7.29e+00	−25.0000

The penalty parameter is selected at every iteration whose function is only to guarantee the sufficient progress toward feasibility and optimality along the search direction. The new method has two goals to determine whether the current iteration is successful or not whose strategy is independent of the penalty parameter. Therefore, the penalty parameter can be any optional positive numbers at the beginning of each iteration. The analysis in this paper indicates that the algorithm should be very robust, and some examples are done to test that robustness in cases where the theory applies and in cases that go beyond the theory. However, it may be necessary to unravel further the more exact behavior of such methods and all their characteristics.

Acknowledgments

The authors would like to thank the referees whose comments have helped them improve the presentation. The first author was supported by Chinese NSF grant 11371273. The second author was supported by the Chinese Natural Science Foundation (Nos. 11331012, 71331001), the National Funds for Distinguished Young Scientists (No. 11125107) and the National 973 Program of China (No. 2015CB856000).

References

[1] P.T. Boggs, J.W. Tolle, Sequential quadratic programming, Acta Numer. 4 (1996) 1–51.
[2] R. Fletcher, N.I.M. Gould, S. Lyffer, A. Wächter, Ph.L. Toint, Global convergence of trust-region SQP-filter algorithms for general nonlinear programming, SIAM J. Optim. 13 (2003) 635–659.

[3] P.T. Boggs, A.J. Kearsley, J.W. Tolle, A global convergence analysis of an algorithm for large-scale nonlinear optimization problems, SIAM J. Optim. 4 (1999) 833–862.
[4] Z.W. Chen, S.Q. Qiu, Global and local convergence of a penalty-free method for nonlinear programming, Comput. Math. Appl. 65 (2013) 589–608.
[5] Z.W. Chen, S.Q. Qiu, Y.J. Jiao, A penalty-free method for equality constrained optimization, J. Ind. Manag. Optim. 9 (2) (2013) 891–409.
[6] R. Fletcher, S. Leyffer, Nonlinear programming without a penalty function, Math. Program. 91 (2002) 239–269.
[7] N.I.M. Gould, PH.L. Toint, Nonlinear programming without a penalty function or a filter, Math. Program. 122 (2010) 155–196.
[8] M. Ulbrich, S. Ulbrich, Non-monotone trust region methods for nonlinear equality constrained optimization without a penalty function, Math. Program. 95 (2003) 103–135.
[9] A. Wächter, L.T. Biegler, Line search filter methods for nonlinear programming: Motivation and global convergence, SIAM J. Optim. 16 (2005) 1–31.
[10] E.G. Birgin, J.M. Martínez, Augmented Lagrangian method with nonmonotone penalty parameters for constrained optimization, Comput. Optim. Appl. 51 (2012) 941–965.
[11] R.H. Byrd, G. Lopez-Calva, J. Nocedal, A line search exact penalty method using steering rules, Math. Program. 133 (2012) 39–73.
[12] R.H. Byrd, J. Nocedal, R.A. Waltz, Steering exact penalty methods for nonlinear programming, Optim. Methods Softw. 23 (2) (2008) 97–213.
[13] L. Chen, D. Goldfarb, Interior-point l_2 penalty methods for nonlinear programming with strong global convergence properties, Math. Program. 108 (1) (2006) 1–36.
[14] Z.W. Chen, X.S. Zhang, A nonmonotone trust-region algorithm with nonmonotone penalty parameters for constrained optimization, J. Comput. Appl. Math. 172 (2004) 7–39.
[15] F.A.M. Gomes, M.C. Maciel, J.M. Martínez, Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters, Math. Program. 84 (1999) 161–200.
[16] N.I.M. Gould, Y. Loh, D.P. Robinson, A filter method with unified step computation for nonlinear optimization, SIAM Optim. 24 (2014) 175–209.
[17] G. Lopez-Calva, Exact-penalty methods for nonlinear programming (PhD thesis), Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA, 2005.
[18] M. Mongeau, A. Sartenaer, Automatic decrease of the penalty parameter in exact penalty function methods, European J. Oper. Res. 83 (3) (1995) 686–699.
[19] L.F. Niu, Y.X. Yuan, A trust-region algorithm for nonlinear constrained optimization, J. Comput. Math. 28 (1) (2010) 72–86.