

Global and local convergence of a nonmonotone SQP method for constrained nonlinear optimization

Chungen Shen · Lei-Hong Zhang · Bo Wang ·
Wenqiong Shao

Received: 12 August 2012 / Published online: 16 July 2014
© Springer Science+Business Media New York 2014

Abstract In this paper, we propose a robust sequential quadratic programming (SQP) method for nonlinear programming without using any explicit penalty function and filter. The method embeds the modified QP subproblem proposed by Burke and Han (Math Program 43:277–303, 1989) for the search direction, which overcomes the common difficulty in the traditional SQP methods, namely the inconsistency of the quadratic programming subproblems. A non-monotonic technique is employed further in a framework in which the trial point is accepted whenever there is a sufficient relaxed reduction of the objective function or the constraint violation function. A forcing sequence possibly tending to zero is introduced to control the constraint violation dynamically, which is able to prevent the constraint violation from over-relaxing and plays a crucial role in global convergence and the local fast convergence as well. We prove that the method converges globally without the Mangasarian–Fromovitz constraint qualification (MFCQ). In particular, we show that any feasible limit point that satisfies the relaxed constant positive linear dependence constraint qualification is also a Karush–Kuhn–Tucker point. Under the strict MFCQ and the second order suf-

C. Shen (✉)

Department of Applied Mathematics, Shanghai Finance University, Shanghai 201209, China
e-mail: shenchungen@gmail.com

L.-H. Zhang

Department of Applied Mathematics, Shanghai University of Finance and Economics,
Shanghai 200433, China

B. Wang

School of Applied Mathematics, Nanjing University of Finance and Economics,
Shanghai 200092, China

W. Shao

Industrial and Systems Engineering Department, Wayne State University, Detroit, MI 48202, USA

ficient condition, furthermore, we establish the superlinear convergence. Preliminary numerical results show the efficiency of our method.

Keywords Sequential quadratic programming method · Non-monotonicity · Convergence analysis · RCPLD · Strict MFCQ

Mathematics Subject Classification 90C30

1 Introduction

In this paper, we consider the general constrained optimization problem:

$$(NLP) \quad \begin{cases} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & c_i(x) = 0, i \in \mathcal{E} \\ & c_i(x) \leq 0, i \in \mathcal{I} \end{cases}$$

where $\mathcal{E} = \{1, 2, \dots, m_1\}$, $\mathcal{I} = \{m_1 + 1, m_1 + 2, \dots, m\}$ and the functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$ for all $i \in \mathcal{E} \cup \mathcal{I}$ are continuously differentiable.

Among many efficient approaches [31] for (NLP), the sequential quadratic programming (SQP) method is one of the most important approaches that has been widely used in practice. There are fruitful theoretical results and numerical investigations for (SQP) and the reader can refer to, e.g., [24, 25, 31–33, 45], for general discussion. In applying the (SQP), however, it is known that three main difficulties may arise: Firstly, the quadratic programming (QP) subproblems could be inconsistent; secondly, the associated sequence of search directions could be unbounded, and thirdly, proper values of penalty parameters involved are difficult to set. In tackling these difficulties, Burke and Han [9] and Zhou [46] modified the QP subproblems and overcome the former two difficulties. It is known that some other techniques may also be able to overcome the inconsistency of the QP subproblems, for instance, the elastic technique [17]. However, in general, they need to solve an auxiliary problem with more unknowns than the original problem, and furthermore it might be hard to determine an appropriate penalty parameter. For the third difficulty, Fletcher and Leyffer [14] proposed a trust-region SQP *filter* method. The underlying principle of Fletcher and Leyffer's method is that, instead considering a combination of the objective value and the constraint violation, they accept the trial point if it improves either the objective function or the constraint violation. The method is effective, but numerically, it needs a so-called *feasibility restoration phase* which may be expensive to compute. Recently, Shen et al. [35] made progress on the filter algorithm by proposing a nonmonotone filter SQP algorithm, which converges under standard assumptions. Except for the trust-region *filter* methods, line search *filter* methods for nonlinear programming are proposed and analyzed in [41–43].

There are also several methods without using a penalty function or a filter. For example, Ulbrich and Ulbrich [36] introduced a nonmonotone trust-region method, which does not involve penalty parameters and the feasibility restoration phase. Inspired

by [36], Xue et al. [44] proposed and analyzed a nonmonotone line search method. It is claimed that their method is comparable to the filter method with respect to the flexibility for accepting trial steps. Bielschowsky and Gomes [4] also suggested an algorithm, which dynamically controls infeasibility. Very recently, Gould and Toint [18] (see also [19]) and Liu and Yuan [29] proposed new algorithms without a penalty function or a filter. However, it should be pointed out that all these mentioned methods are only designed for solving equality constrained optimization.

Some researchers proposed second derivative SQP methods which employ the exact Hessian of Lagrangian in the QP subproblem (e.g., [15,20,21]). In [15], the global solution of the QP subproblem is required for proving global convergence. Gould and Robinson [20,21] presented globally and quadratically convergent SQP methods based on QP subproblems that need not be solved globally. An alternative proposed by Morales et al. [30] is to add an equality quadratic programming with exact second-order derivative at each iteration.

Other than handling general optimization problems, some of researchers are interested in degenerate nonlinear programming problems, for which the active constraint gradients at the solution are linearly dependent or the strict complementarity (SC) condition fails to hold (or both). Wright [37] proposed a stabilized SQP algorithm and established the local quadratic convergence without the linear independence constraint qualification (LICQ) condition. But it requires the Mangasarian–Fromovitz constraint qualification (MFCQ), the SC condition and the second-order sufficient condition (SOSC). In fact, if the SC condition holds, the SOSC is equivalent to the stronger SOSC. Hager [23] proved the locally quadratic convergence under only the stronger SOSC instead of the MFCQ and SC conditions. Similar discussions on this topic can also be found in [38–40].

In this paper, we propose a nonmonotone line search SQP method to solve the general constrained optimization problems (NLP) which might be degenerate in the sense that the MFCQ fails to hold at the solution. Different from the nonmonotone line search approach in [44], our method employs a new rule to update the relaxed constraint violation dynamically, which makes our method competitive with some other approaches. In particular, we can prove the global and local convergence under weaker conditions, such as the relaxed constant positive linear dependence (RCPLD) condition [2] for the global convergence, and the strict MFCQ (SMFCQ) for the local convergence. To avoid difficulties caused by degeneracy, the modified QP subproblem [9] and some nonmonotone technique are incorporated in our algorithm, and we will show that, if a feasible limit point of the iterative sequence satisfies the RCPLD condition, then it must be a Karush–Kuhn–Tucker (KKT) point. The RCPLD condition is recently proposed by Andreani et al. [2] as a new constraint qualification, which is the extension of the constant positive linear dependence (CPLD) condition [3,34], firstly introduced by Qi and Wei [34]. It is worth mentioning that the RCPLD condition is weaker than the CPLD (see [2]), the LICQ and the MFCQ conditions as well. Therefore, our global convergence conclusion that a feasible limit point is a KKT point under the RCPLD (or CPLD) condition is stronger than that under the LICQ condition or the MFCQ condition. For the local convergence of our method,

on the other hand, we show it converges superlinearly under the strict MFCQ and SOSC.

The improved convergence results of our SQP method mainly benefit from the development of primal superlinear convergence of some Newtonian methods under SMFCQ and SOSC (see [13]) (Based on the similar technique, Fernández et al. [13] focused only on the local analysis). We will employ this technique to prove the fast local convergence of a global algorithm, which allows iterates to switch from the global phase to the local phase smoothly. In particular, we will show that the unit step-size is accepted when the iterate is close enough to a minimizer.

To give a much clearer picture of our achievement, we summarize the main features of our proposed algorithm as follows:

1. Our algorithm allows the non-monotonicity of the objective values and formulates a subproblem which is always consistent.
2. Our algorithm does not introduce an explicit penalty function or a filter. No penalty parameter needs to be chosen. Compared with the filter SQP methods, it does not need a feasibility restoration phase which might cost numerous computational amount.
3. We establish the global convergence of our algorithm without the CPLD, LICQ or MFCQ assumptions, which implies that our algorithm is able to handle the degenerate problems. As a comparison, we remark that for degenerate problems, [23, 37, 38] introduce algorithms based on the stabilized QP subproblem using exact second-order information, but only local convergence behavior is investigated, and [40] proposes a globally and superlinearly convergent algorithm, which contains a global phase and a local phase but can not switch from one phase to the other smoothly.
4. For the local convergence, instead of LICQ, the SC condition and SOSC, our algorithm possesses the superlinear convergence under SMFCQ and SOSC. The global and local convergence results are stronger than some other SQP methods, such as those proposed in [29, 35, 44, 46]. Moreover, it is worth mentioning that the introduced switching condition allows the algorithm to transit from the global phase to the local phase smoothly and plays important roles in proving the local convergence under such weaker conditions. We believe that this technical rule can make local convergence possible for other merit-function-free algorithms under SMFCQ and SOSC.
5. Our algorithm can be understand as a perturbed feasible SQP method in some sense. Unlike the classical feasible SQP method, our algorithm does not always improve the objective function. Instead, the forcing sequence in our algorithm controls the constraint violation dynamically so that the constraint violation approaches zero normally, with the possible decrease in the objective function (e.g., when some switching condition is satisfied).

The remainder of this paper is organized as follows. Some preliminaries on constraint qualifications and optimality conditions are reviewed in Sect. 2. In Sect. 3, we present the modified QP subproblem to avoid the inconsistency of the original QP subproblem, and describe the non-monotone decrease conditions for the objective function and the constraint violation. The overall algorithm is stated at the end

of Sect. 3. Under some weaker conditions, the global convergence and the fast local convergence of the algorithm are established in Sects. 4 and 5, respectively. Some numerical results are presented in Sect. 6. Finally, we give some concluding remarks in Sect. 7.

Notation Throughout this paper, we make an extensive use of the symbols $o(\cdot)$, $\mathcal{O}(\cdot)$ and $\Theta(\cdot)$. Let $\{a_k\}$ and $\{b_k\}$ be two vanishing sequences, where $a_k, b_k \in \mathbb{R}, k \in \mathbb{N}$. If the sequence of ratios $\{a_k/b_k\}$ approaches zero as $k \rightarrow +\infty$, then we write $a_k = o(b_k)$. If there exists a constant $C > 0$, such that $|a_k| \leq C|b_k|$ for all k sufficiently large, then we write $a_k = \mathcal{O}(b_k)$. If both $a_k = \mathcal{O}(b_k)$ and $b_k = \mathcal{O}(a_k)$, then we write $a_k = \Theta(b_k)$.

To describe the feasibility of a point x , we define

$$h(x) = \left\| \begin{pmatrix} c_{\mathcal{E}}(x) \\ \max\{c_{\mathcal{I}}(x), 0\} \end{pmatrix} \right\|_{\infty},$$

and for $\sigma > 0$,

$$\Phi(x, \sigma) = \min_{\|d\|_{\infty} \leq \sigma} \max_{i \in \mathcal{E}, i \in \mathcal{I}} \{|c_i(x) + \nabla c_i(x)^T d|, \max\{c_i(x) + \nabla c_i(x)^T d, 0\}\}. \tag{1.1}$$

It is easy to check that (1.1) is equivalent to the following linear programming:

$$\left\{ \begin{array}{ll} \min & \gamma \\ d \in \mathbb{R}^n, \gamma \in \mathbb{R} & \\ \text{s.t.} & -\gamma \leq c_i(x) + \nabla c_i(x)^T d \leq \gamma, \quad i \in \mathcal{E} \\ & c_i(x) + \nabla c_i(x)^T d \leq \gamma, \quad i \in \mathcal{I} \\ & \|d\|_{\infty} \leq \sigma, \\ & \gamma \geq 0, \end{array} \right. \tag{1.2}$$

whose solution with $x = x_k$ will be denoted by $(\underline{d}_k, \underline{\gamma}_k)$ and it is true that $\underline{\gamma}_k = \Phi(x_k, \sigma_k)$. With the definition of $h(x)$, the feasible set \mathcal{D} of the problem (NLP) can be expressed as $\mathcal{D} = \{x : c_i(x) = 0, i \in \mathcal{E}, c_i(x) \leq 0, i \in \mathcal{I}\} = \{x : h(x) \leq 0\}$. The function $\Phi(x_k, \sigma_k)$ provides a measure of inconsistency of the linearized constraints at x_k (if we ignore the effect of the trust region constraint $\|d\|_{\infty} \leq \sigma_k$) and the following lemma gives the relationship between $\Phi(x_k, \sigma_k)$ and $h(x_k)$.

Lemma 1.1 *For all k , $\Phi(x_k, \sigma_k) - h(x_k) \leq 0$ holds. Moreover, if $\Phi(x_k, \sigma_k) - h(x_k) = 0$, then $0 \in \partial h(x_k)$, which implies the stationarity of x_k , where $\partial h(x)$ denotes the Clarke subdifferential of h evaluated at x .*

Proof This lemma follows from [9, Lemma 2.1]. □

It follows from this lemma that $h(x_k) = 0$ implies $\Phi(x_k, \sigma_k) = 0$, but conversely, $\Phi(x_k, \sigma_k) = 0$ only requires consistency of the linearized constraints at x_k rather than $h(x_k) = 0$.

2 Preliminaries

We first review some useful constraint qualifications and basic optimality conditions in this section. For a given $x \in \mathbb{R}^n$, we define the index set of the *active inequality constraints* by

$$\mathcal{A}(x) = \{i \mid c_i(x) = 0, i \in \mathcal{I}\},$$

and the well-known Lagrangian is given by

$$L(x, \lambda) = f(x) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x),$$

where $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)^T \in \mathbb{R}^m$ is the vector of *Lagrange multiplier*. A specific *constraint qualification* at a local minimizer x^* , for example the (LICQ, [31]), is an assumption so that there exists a multiplier vector $\lambda^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*)^T \in \mathbb{R}^m$ such that

$$\begin{cases} \nabla_x L(x^*, \lambda^*) = 0, \\ c_i(x^*) = 0, i \in \mathcal{E}, \\ c_i(x^*) \leq 0, \lambda_i^* \geq 0, \lambda_i^* c_i(x^*) = 0, i \in \mathcal{I}. \end{cases} \tag{2.1}$$

This is known as the KKT conditions. We denote the set of optimal Lagrange multipliers λ^* by $\mathcal{M}_\lambda(x^*)$, and the primal-dual optimal set at x^* by $\mathcal{S}(x^*)$; in other words, $\mathcal{M}_\lambda(x^*) = \{\lambda^* \mid \lambda^* \text{ satisfies (2.3)}\}$, and $\mathcal{S}(x^*) = \{x^*\} \times \mathcal{M}_\lambda(x^*)$. We now review some well-known constraint qualifications.

Definition 2.1 (i) A given feasible point $x \in \mathcal{D}$ is said to satisfy the MFCQ condition (see [9]), if there exists a vector $z \in \mathbb{R}^n$ such that the following systems

$$\nabla c_i(x)^T z = 0, i \in \mathcal{E},$$

$$\nabla c_i(x)^T z < 0, i \in \mathcal{A}(x)$$

are satisfied and the gradients $\{\nabla c_i(x), i \in \mathcal{E}\}$ are linearly independent.

(ii) A given KKT point $x \in \mathcal{D}$ is said to satisfy the strict MFCQ (SMFCQ) condition (see [28]), if there exists a multiplier $\lambda \in \mathcal{M}_\lambda(x)$ and a vector $z \in \mathbb{R}^n$ such that the following systems

$$\nabla c_i(x)^T z = 0, i \in \mathcal{E}$$

$$\nabla c_i(x)^T z < 0, i \in \mathcal{A}^0(x)$$

$$\nabla c_i(x)^T z = 0, i \in \mathcal{A}^+(x)$$

are satisfied and the gradients $\{\nabla c_i(x), i \in \mathcal{E} \cup J(x)\}$ are linearly independent, where $\mathcal{A}^0(x) = \{i \in \mathcal{A}(x) | \lambda_i = 0\}$ and $\mathcal{A}^+(x) = \{i \in \mathcal{A}(x) | \lambda_i > 0\}$ are the sets of indices of strongly and weakly active constraints at (x, λ) , respectively.

Remark 2.1 It is known that if the MFCQ holds at a KKT point x , then the multiplier set corresponding to x is bounded. The SMFCQ is a necessary and sufficient condition for the uniqueness of KKT multipliers (see Kyparisis [28, Proposition 1.1]), and moreover $\text{SMFCQ} \Rightarrow \text{MFCQ}$.

Definition 2.2 (i) Let $x \in \mathcal{D}$, $\mathcal{I}_0 \subset \mathcal{A}(x)$, and $\mathcal{E}_0 \subset \mathcal{E}$. We say that $\{\nabla c_i(x)\}_{i \in \mathcal{E}_0 \cup \mathcal{I}_0}$ is positive linearly dependent (PLD) (see [3, 34]) if there exist scalars $\{\beta_i\}_{i \in \mathcal{E}_0}, \{\beta_i\}_{i \in \mathcal{I}_0}$ such that $\beta_i \geq 0$ for all $i \in \mathcal{I}_0$, $\sum_{i \in \mathcal{E}_0} |\beta_i| + \sum_{i \in \mathcal{I}_0} \beta_i > 0$, and

$$\sum_{i \in \mathcal{E}_0} \beta_i \nabla c_i(x) + \sum_{i \in \mathcal{I}_0} \beta_i \nabla c_i(x) = 0.$$

Otherwise, we say that $\{\nabla c_i(x)\}_{i \in \mathcal{E}_0 \cup \mathcal{I}_0}$ is positive linearly independent.

(ii) A given feasible point $x \in \mathcal{D}$ is said to satisfy the CPLD condition (see [3, 34]) if for any $\mathcal{I}_0 \subset \mathcal{A}(x)$ and $\mathcal{E}_0 \subset \mathcal{E}$, whenever $\{\nabla c_i(x)\}_{i \in \mathcal{E}_0 \cup \mathcal{I}_0}$ is positive linearly dependent, there exists an open neighborhood $N(x)$ of x such that for any $y \in N(x)$, $\{\nabla c_i(y)\}_{i \in \mathcal{E}_0 \cup \mathcal{I}_0}$ is linearly dependent.

It should be noted that the MFCQ condition implies the CPLD condition, but not vice versa. A counterexample is given by Andreani et al. [3], who also proved that the CPLD condition implies quasinormality constraint qualification [26], and is therefore a constraint qualification. Our last constraint qualification is closely related to CPLD and is discussed very recently in [2].

Definition 2.3 A given feasible point $x \in \mathcal{D}$ is said to satisfy the RCPLD condition (see [2]) if there exists an open neighborhood $N(x)$ of x such that for any $y \in N(x)$,

- (1) $\nabla c_{\mathcal{E}}(y)$ has the same rank for every $y \in N(x)$, and
- (2) For any $\mathcal{I}_0 \subset \mathcal{A}(x)$, whenever $\{\nabla c_i(x)\}_{i \in \mathcal{E}_0 \cup \mathcal{I}_0}$ is positive linearly dependent, then $\{\nabla c_i(y)\}_{i \in \mathcal{E}_0 \cup \mathcal{I}_0}$ is linearly dependent, where $\mathcal{E}_0 \subset \mathcal{E}$ such that $\{\nabla c_i(x)\}_{i \in \mathcal{E}_0}$ is a maximal linearly independent subset of $\{\nabla c_i(x)\}_{i \in \mathcal{E}}$.

We point out that the RCPLD distinguishes from the CPLD with respect to the index set \mathcal{E}_0 . Let $\text{span}(V)$ denote the subspace spanned by column vectors of V . Compared to original CPLD, the RCPLD only treats the index set \mathcal{E}_0 satisfying $\text{span}(\{\nabla c_i(x)\}_{i \in \mathcal{E}_0}) = \text{span}(\{\nabla c_i(x)\}_{i \in \mathcal{E}})$, without the need to impose restrictions on all their subsets. Moreover, Andreani et al. (see [2, Theorem 1]) proved that the RCPLD is weaker than the CPLD.

We conclude this section by introducing the second-order sufficient condition (SOSC) which will be used in Sect. 5 for the convergence of our algorithm.

Definition 2.4 A KKT point x^* with $(x^*, \lambda^*) \in \mathcal{M}_\lambda(x^*)$ is said to satisfy the SOSC if

$$d^T \nabla_{xx} L(x^*, \lambda^*) d > 0, \quad \text{for any } d \in \mathcal{C}(x^*) \setminus \{0\},$$

where $\mathcal{C}(x^*)$ is the critical cone of the problem (NLP) at x^* , that is,

$$\mathcal{C}(x^*) = \{d \in \mathbb{R}^n \mid \nabla c_i(x^*)^T d = 0, i \in \mathcal{E}; \nabla c_i(x^*)^T d \leq 0, i \in \mathcal{A}(x^*); \nabla f(x^*)^T d \leq 0\}$$

(see [6, (13.5)]).

We remark that the critical cone can also be written in another form (see [6, (13.6)])

$$\begin{aligned} \mathcal{C}(x^*) &= \{d \in \mathbb{R}^n \mid \nabla c_i(x^*)^T d = 0, i \in \mathcal{E}; \nabla c_i(x^*)^T d \\ &= 0, i \in \mathcal{A}^+(x^*); \nabla c_i(x^*)^T d \leq 0, i \in \mathcal{A}^0(x^*)\}, \end{aligned}$$

where $\mathcal{A}^+(x^*)$ and $\mathcal{A}^0(x^*)$ are the sets of indices of strongly and weakly active constraints at (x^*, λ^*) , respectively (see Definition 2.1 (ii)).

3 Algorithm

Based on the framework of traditional SQP methods, in this section, we will develop our algorithm in detail. Let x_k be the current iterate, then the traditional SQP method generates the next iterate $x_{k+1} := x_k + \alpha_k d_k$ by selecting the step size $\alpha_k \in (0, 1]$ and by solving the QP subproblem for the direction d_k

$$\text{QP}(x_k) \begin{cases} \min_{d \in \mathbb{R}^n} & g_k^T d + \frac{1}{2} d^T B_k d \\ \text{s.t.} & c_i(x_k) + \nabla c_i(x_k)^T d = 0, i \in \mathcal{E} \\ & c_i(x_k) + \nabla c_i(x_k)^T d \leq 0, i \in \mathcal{I} \end{cases} \quad (3.1)$$

where $g_k = \nabla f(x_k)$, and B_k is a particular symmetric positive definite matrix. However, it is known that two possible difficulties could arise in (3.1), namely, that the problem $\text{QP}(x_k)$ could be inconsistent and the direction d_k could tend to be unbounded in norm as k tends to infinity if the matrices $\{B_k\}$ tend to singularity. In the next subsection, we will modify (3.1) to handle these difficulties.

3.1 The modified QP subproblem

To get around the troubles in the $\text{QP}(x_k)$ (3.1), we use the technique in [9] by defining a modified QP subproblem

$$\text{MQP}(x_k) \begin{cases} \min_{d \in \mathbb{R}^n} & g_k^T d + \frac{1}{2} d^T B_k d \\ \text{s.t.} & \nabla c_i(x_k)^T d + c_i(x_k) = r_i(x_k, \sigma_k), i \in \mathcal{E}, \\ & \nabla c_i(x_k)^T d + c_i(x_k) \leq \Phi(x_k, \sigma_k), i \in \mathcal{I} \\ & \|d\|_\infty \leq \beta_k, \end{cases} \quad (3.2)$$

where $\beta_k > \sigma_k > 0, r_i(x_k, \sigma_k) = \nabla c_i(x_k)^T \underline{d}_k + c_i(x_k), i \in \mathcal{E}$, and \underline{d}_k is computed by (1.2). Since the optimal value of (1.2) $\underline{\gamma}_k = \Phi(x_k, \sigma_k)$ provides

a measure of inconsistency of the linearized constraints at x_k , and the quantities $r_i(x_k, \sigma_k)$, $i \in \mathcal{E}$ are bounded by $\Phi(x_k, \sigma_k)$ as well, the modified QP subproblem (ignoring the trust region constraint) can be understood as a “small perturbation” of the QP subproblem (3.1) in the sense of inconsistency of the linearized constraints. On one hand, if $QP(x_k)$ is feasible, the quantities $\Phi(x_k, \sigma_k)$ and $r_i(x_k, \sigma_k)$, $i \in \mathcal{E}$ vanish, and then $MQP(x_k)$ reduces to $QP(x_k)$ with a trust region constraint. On the other hand, we note that the modified QP subproblem $MQP(x_k)$ is well-defined as d_k itself is a feasible point, and [9, Lemma 2.2] further claims that $MQP(x_k)$ admits a unique solution, say d_k , bounded by β_k . Therefore, the modified QP subproblem (3.2) overcomes the two potential difficulties in (3.1). Moreover, we will see that the good global and local convergence are ensured by using the search direction d_k computed from (3.2), for which the KKT conditions are

$$\begin{cases} g_k + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_{k,i} \nabla c_i(x_k) + B_k d_k + \lambda_k^u - \lambda_k^l = 0, \\ \lambda_{k,i} (\nabla c_i(x_k)^T d_k + c_i(x_k) - \Phi(x_k, \sigma_k)) = 0, \quad i \in \mathcal{I}, \\ (d_k - \beta_k e_n)^T \lambda_k^u = 0, \quad (d_k + \beta_k e_n)^T \lambda_k^l = 0, \\ \nabla c_i(x_k)^T d_k + c_i(x_k) - r_i(x_k, \sigma_k) = 0, \quad i \in \mathcal{E}, \\ \nabla c_i(x_k)^T d_k + c_i(x_k) \leq \Phi(x_k, \sigma_k), \quad i \in \mathcal{I}, \quad \|d_k\|_\infty \leq \beta_k, \\ \lambda_{k,i} \geq 0, \quad i \in \mathcal{I}, \quad \lambda_{k,i}^u \geq 0, \quad \lambda_{k,i}^l \geq 0, \quad i \in \{1, 2, \dots, n\}, \end{cases} \tag{3.3}$$

where $e_n = (1, 1, \dots, 1)^T \in \mathbb{R}^n$, $\lambda_k = (\lambda_{k,1}, \lambda_{k,2}, \dots, \lambda_{k,m})^T \in \mathbb{R}^m$, $\lambda_k^l = (\lambda_{k,1}^l, \lambda_{k,2}^l, \dots, \lambda_{k,n}^l)^T \in \mathbb{R}^n$ and $\lambda_k^u = (\lambda_{k,1}^u, \lambda_{k,2}^u, \dots, \lambda_{k,n}^u)^T \in \mathbb{R}^n$.

3.2 The nonmonotone decrease condition for the objective function

The general principle of algorithms for the constrained minimization is to minimize the objective function while control the constraint violation in a specific way. This naturally leads to algorithms with monotone conditions. However, algorithms with nonmonotone conditions have been proposed and demonstrated competitive numerical performance recently, because nonmonotone conditions could possibly avoid the undesired *Maratos effect*. In our algorithm, we adopt the nonmonotone conditions by accepting a new trial point whenever certain relaxed requirements on $h(x)$ are fulfilled and some nonmonotone condition $f(x)$ holds, if necessary.

To describe more clearly our strategies, we first state the nonmonotone decrease condition for the objective function in this subsection. Define \hat{f}_k as the maximum value over the past few iterations. For simplicity, in this paper, we choose the past three iterations, that is, $\hat{f}_k = \max_{i=0,1,2} f(x_{k-i})$ and particularly define $\hat{f}_0 = f(x_0)$ and

$\hat{f}_1 = \max\{f(x_0), f(x_1)\}$. Let $\hat{x}_k(\alpha_{k,l}) := x_k + p_k$ be the next trial point, where $p_k = \alpha_{k,l} d_k$, and $\alpha_{k,l} \in (0, 1] (l = 0, 1, 2, \dots)$ is a decreasing sequence converging to 0. Once $\hat{x}_k(\alpha_{k,l})$ is accepted, we set $x_{k+1} := \hat{x}_k(\alpha_{k,l})$ and $\alpha_k := \alpha_{k,l}$. Denote the

relaxed reduction of $f(x_k)$ by

$$\Delta \hat{f}_k = \hat{f}_k - f(x_k + p_k),$$

and the linear reduction by $\Delta l_k = -g_k^T d_k$. For a trial point $\hat{x}_k(\alpha_{k,l})$, if

$$\Delta l_k \geq \xi d_k^T B_k d_k \text{ and } h(x_k) \leq \zeta \|d_k\| \cdot \|\tilde{d}_{k+1}\|, \text{ if FLAG}=1 \ \& \ \alpha_{k,l} = 1 \tag{3.4a}$$

$$\Delta l_k \geq \xi d_k^T B_k d_k, \qquad \text{otherwise} \tag{3.4b}$$

are satisfied, then the relaxed sufficient reduction condition

$$\Delta \hat{f}_k \geq \alpha_{k,l} \sigma \min(\Delta l_k, \tau \|d_k\|^\nu), \tag{3.5}$$

is further required to be satisfied for $\hat{x}_k(\alpha_{k,l})$ to be the accepted iterate, where the constants $\sigma \in (0, \frac{1}{2})$, $\tau > 0$, $\nu \in (2, 3]$, $\xi \in (0, \frac{1}{2})$ and $\zeta > 0$. We have several remarks in order, to explain the conditions (3.4a–3.4b) and (3.5):

1. FLAG=1 in (3.4a) (also see Algorithm A) means that the second-order correction (SOC) step \tilde{d}_{k+1} is computed (Detailed information on the SOC steps is described in Sect. 3.4).
2. If $\hat{x}_k(\alpha_{k,l})$ is accepted as the next iterate satisfying (3.4a) or (3.4b), then either (3.4a–3.4b)–(3.5) or (3.4b–3.5) are fulfilled, depending on the values of FLAG and $\alpha_{k,l}$.
3. Conditions (3.4a) or (3.4b) ensures that d_k is a “sufficiently” descent direction of $f(x)$, but it alone might result in convergence to feasible limit points instead of KKT points, even if the problem is well-posed. Condition (3.5) is therefore further imposed to avoid this situation.
4. The additional requirement $h(x_k) \leq \zeta \|d_k\| \cdot \|\tilde{d}_{k+1}\|$ in (3.4a) is for the local convergence of our algorithm.
5. The right-hand sides in the second inequality of (3.4a) and (3.5) are slight modifications of those in [44], which play important roles in our local convergence analysis.

3.3 The relaxed decrease condition for the constraint violation

Now we discuss the relaxed decrease condition for the constraint violation $h(x)$. Since our algorithm will not use any penalty function to measure contribution of a new iterate with respect to global convergence, we handle the decrease conditions of the constraint violation and the objective function separately. One direct idea is that we can impose a relaxed condition on constraint violation when the objective function is improved. To achieve this goal, we shall introduce a new quantity \mathcal{R}_k to keep the loose control over the constraint violation. The detailed update rules for \mathcal{R}_k are given by Algorithm J.

Algorithm J (Update \mathcal{R}_k at iteration k)

```

1 if  $k=0$  then
2 | Set  $\mathcal{R}_0 = h(x_0) + 1$ ;
3 end
4 if (3.7) does not hold then
5 | Set  $\mathcal{R}_k = h(x_k)$ ;
6 else
7 | Set  $\mathcal{R}_k = \mathcal{R}_{k-1}$ ;
8 end
9 return  $\mathcal{R}_k$ ;

```

It should be noticed that Algorithm J only potentially sets $\mathcal{R}_k \neq \mathcal{R}_{k-1}$ if (3.4) does not hold. If (3.4) is satisfied, the objective function should further fulfill the nonmonotone sufficient reduction condition (3.5). In this case, we prefer to relax the constraint violation rather than update by $h(x_k)$, since we try not to impose too restrictive constraint violation requirement when the objective function has a nonmonotone sufficient reduction. Such strategy is also crucial for the fast local convergence under certain weak conditions as mentioned in Sect. 1.

At each iteration, we impose the following relaxed reduction condition on the constraint violation:

$$\mathcal{R}_k - h(x_k + p_k) \geq \alpha_{k,l} \eta (\mathcal{R}_k - \Phi(x_k, \sigma_k)), \quad (3.6)$$

where $\eta \in (0, \frac{1}{2})$.

We make some remarks for condition (3.6). If $\Phi(x_k, \sigma_k) = 0$, then (3.6) reduces to

$$\mathcal{R}_k - h(x_k + p_k) \geq \alpha_{k,l} \eta \mathcal{R}_k. \quad (3.7)$$

When (3.4) does not hold, this inequality becomes a normal decrease condition on $h(x)$ in the literature. But if (3.4) is satisfied, the quantity $\mathcal{R}_k = \mathcal{R}_{k-1}$ at iteration k , and therefore a relaxed reduction on $h(x)$ is imposed. As the QP subproblem (3.1) might be inconsistent, we introduce a modified one (3.2) in place of (3.1) to overcome this problem. However, in this case we cannot make the constraint violation decrease as much as that in (3.7). Instead, a relaxed condition (3.6) is required in the case of $\Phi(x_k, \sigma_k) > 0$. The quantity $\Phi(x_k, \sigma_k) > 0$ indicates that the QP subproblem is inconsistent, which may be due to a small trust region or ill-conditioning.

3.4 Avoid the Maratos effect

Our last subject is to avoid the Maratos effect. It is well-known that SQP-type methods may suffer from the Maratos effect in which a full SQP step leads to the increase both of the objective function and of the constraint violation. Due to the step acceptance mechanism, the Maratos effect could result in rejection of some good steps towards

an optimal solution. Fletcher and Leyffer [14] employed the SOC step to improve the search direction so that the improved search direction can be accepted at the end. Many other authors have discussed the SOC technique for overcoming the Maratos effect (see e.g., [31] and references therein). Besides, there is an alternative, namely the watch-dog technique [11], which, in our opinion, can also be viewed as an SOC-type technique. As our choice, we adopt an SOC technique in a similar manner of the watch-dog technique by computing the SOC step \tilde{d}_{k+1} from the following subproblem

$$\widetilde{\text{QP}}(x_k + d_k) \begin{cases} \min \nabla f(x_k + d_k)^T d + \frac{1}{2} d^T \tilde{B}_{k+1} d \\ \text{s.t. } c_i(x_k + d_k) + \nabla c_i(x_k + d_k)^T d = 0, \quad i \in \mathcal{E}, \\ \quad c_i(x_k + d_k) + \nabla c_i(x_k + d_k)^T d \leq 0, \quad i \in \mathcal{I}, \end{cases} \quad (3.8)$$

where \tilde{B}_{k+1} is updated using B_k and other information at $x_k + d_k$. One can of course apply certain quasi-Newton formula to proceed this update.

Suppose x_k is the current iterate. Starting from the trial point $x_k + d_k$ (i.e., line 3.5 in Algorithm A) where d_k is computed from the modified QP subproblem, $\text{MQP}(x_k)$ given by (3.2), we describe our detailed steps for updating x_k as follows:

- Step 1. Check whether the trial point $\hat{x}_k(\alpha_{k,l})$ is accepted (i.e., according to the condition given in line 13 in Algorithm A) or not. If it is, then we directly update x_{k+1} according to the value FLAG and $\alpha_{k,l}$ (i.e., lines 26–30 in Algorithm A).
- Step 2. For the case that the trial point $\hat{x}_k(\alpha_{k,l})$ is not accepted, if FLAG=1 (implying the SOC step has been computed) or $\alpha_{k,l} < 1$, then we simply shrink the trial step size $\alpha_{k,l}$ and form a new trial point $\hat{x}_k(\alpha_{k,l})$ (line 22 in Algorithm A) and go to step 1; otherwise, we go to step 3.
- Step 3. Compute \tilde{B}_{k+1} using B_k to form the subproblem $\widetilde{\text{QP}}(x_k + d_k)$ given by (3.8) (i.e., line 15 in Algorithm A) and go to step 4.
- Step 4. If $\widetilde{\text{QP}}(x_k + d_k)$ is consistent, then we compute its solution \tilde{d}_{k+1} and set $\hat{x}_k(1) := x_k + d_k + \tilde{d}_{k+1}$ as the new trial point (i.e., line 17 in Algorithm A) and go to step 1. Otherwise, we simply shrink the trial step size $\alpha_{k,l}$ and form a new trial point $\hat{x}_k(\alpha_{k,l})$ (line 19 in Algorithm A), and go to step 1.

3.5 The overall algorithm

Our previous discussion shows that the trial iterate $\hat{x}_k(\alpha_{k,l}) = x_k + p_k$ is accepted if and only if (3.6) is satisfied and

$$(3.8) \text{ holds or } (3.7) \text{ does not hold,}$$

where $p_k = \alpha_{k,l}d_k$ or $p_k = d_k + \tilde{d}_{k+1}$, depending on if the SOC step is used. Thus, we can now present our overall nonmonotone SQP method in Algorithm A.

Algorithm A

```

1 Given initial parameters:  $x_0 \in \mathbb{R}^n$ ,  $B_0 \in \mathbb{R}^{n \times n}$ ,  $\gamma > 0$ ,  $\zeta > 0$ ,  $\tau > 0$ ,  $\nu \in (2, 3]$ ,
   $\sigma \in (0, \frac{1}{2})$ ,  $\xi \in (0, \frac{1}{2})$ ,  $t \in (0, 1)$ ,  $0 < \underline{\sigma} < \bar{\sigma} < \bar{\beta}$ ,  $\sigma_0 \in [\underline{\sigma}, \bar{\sigma}]$ ,  $\beta_0 \in (\sigma_0, \bar{\beta}]$ ,
  and FLAG=0;
2 Set  $k \leftarrow 0$ ;
3 while  $x_k$  is not optimal do
4   ► Compute the search direction;
5   Compute the search direction  $d_k$  and the associated multiplier  $\lambda_k$  by solving
     MQP( $x_k$ );
6   ► Check for termination;
7   if the KKT condition (2.3) is satisfied, or both  $h(x_k) = \Phi(x_k, \sigma_k)$  and
      $\Phi(x_k, \sigma_k) > 0$  then
8     | Stop;
9   end
10  Calculate  $\mathcal{R}_k$  by calling Algorithm J;
11  ► Perform a line search;
12  Set  $\alpha_{k,0} = 1$ ,  $l \leftarrow 0$ ,  $p_k = \alpha_{k,l}d_k$ , and  $\hat{x}_k(\alpha_{k,l}) = x_k + p_k$ ;
13  while (3.9) is not satisfied or ((3.7) is satisfied and (3.8) is not satisfied) do
14    | if FLAG=0 and  $\alpha_{k,l} = 1$  then
15      | Compute  $\tilde{B}_{k+1}$  from  $B_k$ , and set FLAG=1;
16      | if  $\tilde{Q}\tilde{P}(x_k + d_k)$  is consistent then
17        | Solve  $\tilde{Q}\tilde{P}(x_k + d_k)$  to obtain  $\tilde{d}_{k+1}$  and its multiplier  $\tilde{\lambda}_{k+1}$ , set
18          |  $p_k = d_k + \tilde{d}_{k+1}$  and  $\hat{x}_k(\alpha_{k,l}) = x_k + p_k$ ;
19        | else
20          | Set  $\alpha_{k,l+1} = t\alpha_{k,l}$ ,  $l \leftarrow l + 1$ ,  $p_k = \alpha_{k,l}d_k$ , and  $\hat{x}_k(\alpha_{k,l}) = x_k + p_k$ ;
21        | end
22      | else
23        | Set  $\alpha_{k,l+1} = t\alpha_{k,l}$ ,  $l \leftarrow l + 1$ ,  $p_k = \alpha_{k,l}d_k$ , and  $\hat{x}_k(\alpha_{k,l}) = x_k + p_k$ ;
24      | end
25    | ► Update;
26    | if FLAG=1 and  $\alpha_{k,l} = 1$  then
27      | Set  $\alpha_k = \alpha_{k,l}$ ,  $x_{k+1} = x_k + d_k + \tilde{d}_{k+1}$ , and update  $B_{k+1}$  by  $\tilde{B}_{k+1}$ ,  $\tilde{d}_{k+1}$ 
28        | and  $\tilde{\lambda}_{k+1}$ ;
29      | else
30        | Set  $\alpha_k = \alpha_{k,l}$ ,  $x_{k+1} = x_k + \alpha_k d_k$ , and update  $B_{k+1}$  by  $B_k$ ,  $p_k$  and  $\lambda_k$ ;
31      | end
32    | Choose  $\sigma_{k+1} \in [\underline{\sigma}, \bar{\sigma}]$ ,  $\beta_{k+1} \in (\sigma_{k+1}, \bar{\beta}]$ ;
33    | Set FLAG=0 and  $k \leftarrow k + 1$ ;
34  end

```

As the algorithm looks a little involved, for clarity, we make several remarks:

1. In line 1, parameters $\underline{\sigma}$ and $\bar{\sigma}$ are the lower and upper bounds of σ_k , respectively, and $\bar{\beta}$ is an upper bound of β_k . An evident lower bound of β_k is $\underline{\sigma}$.

2. Conditions in lines 6–9 are the stopping criteria. If (2.1) is satisfied at x_k , then x_k is a KKT point. If both $h(x_k) = \Phi(x_k, \sigma_k)$ and $\Phi(x_k, \sigma_k) > 0$, then by Lemma 1.1, we know that x_k is a locally infeasible stationary point of h .
3. FLAG is used to indicate whether the SOC step is computed (FLAG=1) or not (FLAG=0). It should be noted that FLAG=1 does not mean the success of the SOC correction since we do not know if the SOC step is accepted or not in advance.
4. In lines 15, 27 and 29, we might update \tilde{B}_{k+1} and B_{k+1} by the modified BFGS method [32].
5. We call line 13–line 24 the inner loop and line 3–line 33 the outer loop.

4 Global convergence

Now we are in a position to prove the global convergence of Algorithm A. The followings are our assumptions that will be used in our convergence analysis.

- (A1) Let $\{x_k\}$ be generated by Algorithm A, and $\{x_k\}$ and $\{x_k + d_k\}$ be contained in a closed and bounded set S of \mathbb{R}^n .
- (A2) All the functions $f(x), c_i(x), i \in \mathcal{E} \cup \mathcal{I}$ are twice continuously differentiable on S .
- (A3) The matrices B_k and \tilde{B}_k are uniformly bounded and uniformly positive definite for all k .

Remark 4.1 Assumptions (A1) and (A2) are also used in [15]. Assumption (A3) are commonly used in convergence proofs of line search SQP algorithms (see, e.g., [10]). A consequence of Assumption (A3) is that there exist constants $\delta > 0$ and $M > 0$, independent of k , such that $\delta \|y\|^2 \leq y^T B_k y \leq M \|y\|^2$ for any $y \in \mathbb{R}^n$. Assumptions (A1) and (A2) imply that $\nabla^2 f(x), \nabla^2 c_i(x), i \in \mathcal{E} \cup \mathcal{I}$ are bounded on S . Without loss of generality, we assume that $\|\nabla^2 c_i(x)\| \leq M, \forall i \in \mathcal{E} \cup \mathcal{I}, \|\nabla^2 f(x)\| \leq M, \forall x \in S$.

The following two lemmas show that x_k (or $x_k + d_k$) is a KKT point of the problem (NLP) if both $d_k = 0$ and $h(x_k) = 0$ (or $\tilde{d}_{k+1} = 0$).

Lemma 4.1 *Let $d_k = 0$ be a solution of MQP(x_k) with $\sigma_k \geq \underline{\sigma}$. Then x_k is a stationary point of h . Moreover, if $x_k \in \mathcal{D}$, then x_k is a KKT point of the problem (NLP).*

Proof If $d_k = 0$, it follows from (3.2) and the definition of $h(x)$ that

$$\begin{aligned} h(x_k) &= \max\{|c_i(x_k)|, i \in \mathcal{E}; \max\{c_i(x_k), 0\}, i \in \mathcal{I}\} \\ &\leq \max\{|c_i(x_k) + \nabla c_i(x_k)^T \underline{d}_k|, i \in \mathcal{E}; \Phi(x_k, \sigma_k)\} \\ &\leq \Phi(x_k, \sigma_k), \end{aligned}$$

where the last inequality follows from the definition of $\Phi(x, \sigma)$. Using Lemma 1.1, we have that $\Phi(x_k, \sigma_k) - h(x_k) = 0$, and therefore x_k is a stationary point of h . If $h(x_k) = 0$ and $d_k = 0$, then it follows from [9, Lemma 2.2] that x_k is a KKT point for the problem (NLP). □

Lemma 4.2 *Let $\tilde{d}_{k+1} = 0$ be a solution of $\widetilde{QP}(x_k + d_k)$. Then $x_k + d_k$ is a KKT point of the problem (NLP).*

Proof The conclusion follows immediately from the KKT conditions for $\widetilde{QP}(x_k + d_k)$. □

The following four lemmas give preparations for well-definedness of our algorithm.

Lemma 4.3 *Assume \bar{x} is not a stationary point of h in the sense that $0 \notin \partial h(\bar{x})$. Then there exists a scalar $\bar{\epsilon} > 0$ and a neighborhood $N(\bar{x})$ of \bar{x} , such that $\Phi(x_k, \sigma_k) - h(x_k) < -\bar{\epsilon}$ for all $\sigma_k \geq \underline{\sigma}$ and all $x_k \in N(\bar{x})$.*

Proof By [9, Lemma 2.1], $0 \notin \partial h(\bar{x})$ implies $\Phi(\bar{x}, \underline{\sigma}) - h(\bar{x}) < 0$, where $\underline{\sigma}$ is from Algorithm A. By the continuity of the function $\Phi(\cdot, \underline{\sigma}) - h(\cdot)$ on \mathbb{R}^n , there exists a neighborhood $N(\bar{x})$ and a scalar $\bar{\epsilon} > 0$ such that $\Phi(x_k, \underline{\sigma}) - h(x_k) < -\bar{\epsilon}$ whenever $x_k \in N(\bar{x})$. The condition $\sigma_k \geq \underline{\sigma}$ together with the definition of $\Phi(x_k, \cdot)$ yields $\Phi(x_k, \sigma_k) - h(x_k) \leq \Phi(x_k, \underline{\sigma}) - h(x_k)$. Therefore, $\Phi(x_k, \sigma_k) - h(x_k) < -\bar{\epsilon}$ holds for all $\sigma_k \geq \underline{\sigma}$ and all $x_k \in N(\bar{x})$. □

Remark 4.2 It can be seen that $\bar{\epsilon}$ depends on \bar{x} such that $0 \notin \partial h(\bar{x})$ (i.e., $\bar{\epsilon} = \bar{\epsilon}(\bar{x})$), because it must satisfy $\Phi(x_k, \sigma_k) - h(x_k) < -\bar{\epsilon}$.

Lemma 4.4 *Let Assumptions (A1)–(A3) hold and d_k be the solution of $MQP(x_k)$. Then*

$$h(x_k + \alpha d_k) - h(x_k) \leq \alpha(\Phi(x_k, \sigma_k) - h(x_k)) + \frac{1}{2}\alpha^2 M \|d_k\|^2, \text{ and} \tag{4.1}$$

$$|f(x_k) - f(x_k + \alpha d_k) - \alpha \Delta I_k| \leq \frac{1}{2}\alpha^2 M \|d_k\|^2 \tag{4.2}$$

hold for all $0 < \alpha \leq 1$.

Proof Since $(\underline{d}_k, \underline{\gamma}_k)$ is the solution of the linear program (1.2), and $r_i(x_k, \sigma_k) = \nabla c_i(x_k)^T \underline{d}_k + c_i(x_k)$ for $i \in \mathcal{E}$, it follows from (1.1) that

$$|r_i(x_k, \sigma_k)| \leq \underline{\gamma}_k = \Phi(x_k, \sigma_k). \tag{4.3}$$

Using the Taylor Expansion formula, we have that, for $i \in \mathcal{E}$ and $0 < \alpha \leq 1$,

$$\begin{aligned} & |c_i(x_k + \alpha d_k)| \\ &= |c_i(x_k) + \alpha \nabla c_i(x_k)^T d_k + \frac{1}{2}\alpha^2 d_k^T \nabla^2 c_i(z_i) d_k| \\ &\leq (1 - \alpha)|c_i(x_k)| + \alpha |r_i(x_k, \sigma_k)| \\ &\quad + \frac{1}{2}\alpha^2 |d_k^T \nabla^2 c_i(z_i) d_k| \quad (\text{by the fourth equation in (3.6)}) \\ &\leq (1 - \alpha)h(x_k) + \alpha \Phi(x_k, \sigma_k) \\ &\quad + \frac{1}{2}\alpha^2 M \|d_k\|^2, \quad (\text{by (4.14) and Assumption (A2)}) \end{aligned} \tag{4.4}$$

where the vector z_i is between x_k and $x_k + \alpha d_k$. Similarly, by the Taylor Expansion formula, the fifth equation in (3.3) and Assumption (A2), we have that for all $i \in \mathcal{I}$ and $0 < \alpha \leq 1$,

$$\begin{aligned} c_i(x_k + \alpha d_k) &= c_i(x_k) + \alpha \nabla c_i(x_k)^T d_k + \frac{1}{2} \alpha^2 d_k^T \nabla^2 c_i(z_i) d_k \\ &\leq (1 - \alpha) h(x_k) + \alpha \Phi(x_k, \sigma_k) + \frac{1}{2} \alpha^2 M \|d_k\|^2, \end{aligned}$$

where the vector z_i is between x_k and $x_k + \alpha d_k$. This together with (4.4) implies (4.1). The result of (4.2) follows from [44, Lemma 1]. □

Lemma 4.5 *Let Assumptions (A1)–(A3) hold and d_k be the solution of (3.2). If $h(x_k) = 0$, then (3.4) holds.*

Proof The second inequality (3.4a) is true for the sake of $h(x_k) = 0$. In view of Lemma 1.1, $h(x_k) = 0$ implies $\Phi(x_k, \sigma_k) = 0$, $c_i(x_k) = 0$ with $i \in \mathcal{E}$ and $c_i(x_k) \leq 0$ with $i \in \mathcal{I}$, which together with (1.2) gives $r_i(x_k, \sigma_k) = 0$, $i \in \mathcal{E}$. The KKT conditions for (3.2) give

$$\begin{aligned} g_k^T d_k &= - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_{k,i} \nabla c_i(x_k)^T d_k - d_k^T B_k d_k - (\lambda_k^u - \lambda_k^l)^T d_k \\ &= \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_{k,i} c_i(x_k) - \sum_{i \in \mathcal{E}} \lambda_{k,i} r_i(x_k, \sigma_k) - \sum_{i \in \mathcal{I}} \lambda_{k,i} \Phi(x_k, \sigma_k) \\ &\quad - d_k^T B_k d_k - \beta_k (\|\lambda_k^l\|_1 + \|\lambda_k^u\|_1) \\ &\leq -\xi d_k^T B_k d_k, \end{aligned}$$

where the last inequality follows from that facts $c_i(x_k) = 0$ and $r_i(x_k, \sigma_k) = 0$ with $i \in \mathcal{E}$, $c_i(x_k) \leq 0$ and $\lambda_{k,i} \geq 0$ with $i \in \mathcal{I}$, and $\Phi(x_k, \sigma_k) = 0$. This implies (3.4) holds. □

We remark that $\mathcal{R}_k > 0$ for all k . In fact, the above lemma has shown that $h(x_k) > 0$ if (3.4) is not true at iteration k , and the mechanism of Algorithm J ensures that \mathcal{R}_k either takes $h(x_k) > 0$ in the case that (3.4) is not true, or $\mathcal{R}_k = \mathcal{R}_{k-1}$. Even if (3.4) is satisfied for all k , \mathcal{R}_0 is initialized as $h(x_0) + 1$ (see Algorithm J) which is positive, and then \mathcal{R}_k will remain to be \mathcal{R}_0 for all k .

Lemma 4.6 *Let Assumptions (A1)–(A3) hold and d_k be the solution of (3.2). Then there exists a constant $\tilde{C} \in (0, 1]$ such that (3.5) holds for all $\alpha_{k,l} \in (0, \tilde{C}]$.*

Proof Whatever (3.4b) or (3.4a) holds, the linear reduction $\Delta l_k = -g_k^T d_k \geq \xi d_k^T B_k d_k$ together with Assumption (A3) yields $\Delta l_k \geq \xi \delta \|d_k\|^2$. It follows from Lemma 4.4 that

$$\left| \frac{f(x_k) - f(x_k + \alpha_{k,l} d_k) - \alpha_{k,l} \Delta l_k}{\alpha_{k,l} \Delta l_k} \right| \leq \frac{\alpha_{k,l} M}{2\xi \delta}.$$

We define $\tilde{C} := \min \left\{ 1, \frac{2\delta\xi(1-\sigma)}{M} \right\}$. If $0 < \alpha_{k,l} \leq \tilde{C}$, then

$$\left| \frac{f(x_k) - f(x_k + \alpha_{k,l}d_k)}{\alpha_{k,l}\Delta l_k} - 1 \right| \leq 1 - \sigma,$$

which leads to

$$\Delta \hat{f}_k \geq f(x_k) - f(x_k + \alpha_{k,l}d_k) \geq \alpha_{k,l}\sigma \Delta l_k \geq \alpha_{k,l}\sigma \min\{\Delta l_k, \tau \|d_k\|^v\}.$$

□

With these preparatory results, we can show the well-definedness of Algorithm A in the following theorem.

Theorem 4.1 *Let Assumptions (A1)–(A3) hold. Then the inner loop terminates finitely, that is, there exists an $\bar{\alpha}_k \in (0, 1]$ such that $\hat{x}_k(\alpha_{k,l}) = x_k + \alpha_{k,l}d_k$ is accepted for $0 < \alpha_{k,l} \leq \bar{\alpha}_k$.*

Proof If $d_k = 0$ is the solution of MQP(x_k), then it follows from Lemma 4.1 that x_k is a KKT point of the problem (NLP), and thus Algorithm A terminates without entering the inner loop. If x_k is an infeasible stationary point of h (that is, $\Phi(x_k, \sigma_k) = h(x_k) > 0$), then the inner loop of Algorithm A terminates too. If $x_k + d_k$ or $x_k + d_k + \tilde{d}_{k+1}$ is accepted, then the inner loop terminates.

In the following, we assume that $d_k \neq 0$ for some k , x_k is not an infeasible stationary point of h , and none of $x_k + d_k$ and $x_k + d_k + \tilde{d}_{k+1}$ is accepted. Our following proof will be divided into two scenarios:

Case (i): Condition (3.4) does not hold.

In light of Algorithm J and Lemma 4.5, it follows that $\mathcal{R}_k = h(x_k) > 0$, and using Lemma 4.4 yields

$$h(x_k + \alpha_{k,l}d_k) - \mathcal{R}_k \leq \alpha_{k,l}(\Phi(x_k, \sigma_k) - \mathcal{R}_k) + \frac{1}{2}\alpha_{k,l}^2 M \|d_k\|^2 \tag{4.5}$$

for $\alpha_{k,l} \in (0, 1]$. We define

$$C_k = \min \left\{ 1, \frac{2(1-\eta)(\mathcal{R}_k - \Phi(x_k, \sigma_k))}{M \|d_k\|^2} \right\}. \tag{4.6}$$

We now show that $C_k > 0$ under our assumptions. Since x_k is not an infeasible stationary point of h and $h(x_k) > 0$, it follows that $\Phi(x_k, \sigma_k) < h(x_k) = \mathcal{R}_k$ which leads to $C_k > 0$. Using (4.6), it follows from (4.5) that for $\alpha_{k,l} \in (0, C_k]$,

$$\begin{aligned} h(x_k + \alpha_{k,l}d_k) - \mathcal{R}_k &\leq \alpha_{k,l}(\Phi(x_k, \sigma_k) - \mathcal{R}_k) + \frac{2(1-\eta)(\mathcal{R}_k - \Phi(x_k, \sigma_k))}{M \|d_k\|^2} \frac{\alpha_{k,l}}{2} M \|d_k\|^2 \\ &= -\alpha_{k,l}\eta(\Phi(x_k, \sigma_k) - \mathcal{R}_k), \end{aligned}$$

which means that (3.6) holds for all $\alpha_{k,l} \in (0, C_k]$. In this case, by the mechanism of Algorithm A (see line 3.5 in Algorithm A), the inner loop terminates if $0 < \alpha_{k,l} \leq C_k$.

Case (ii): Condition (3.4) holds.

We now prove that (3.6) also holds for all $\alpha_{k,l} \in (0, C_k]$, where C_k is defined in **Case (i)**. For any k satisfying (3.4), either (3.4) holds at iterations $s = 0, 1, \dots, k - 1$ or there exists an index $k^- < k$ such that (3.4) holds at iterations $s = k^- + 1, k^- + 2, \dots, k$ and (3.4) does not hold at iteration k^- . By Algorithm J, either $\mathcal{R}_k = \mathcal{R}_{k-1} = \dots = \mathcal{R}_0 > 0$ or $\mathcal{R}_k = \mathcal{R}_{k-1} = \dots = \mathcal{R}_{k^-}$. By Lemma 4.4, we obtain that (4.5) is also true for $\alpha_{k,l} \in (0, 1]$ if $\mathcal{R}_k \geq h(x_k)$. In the case that $\mathcal{R}_k = \mathcal{R}_{k-1} = \dots = \mathcal{R}_0 > 0$, it follows from Algorithm J and Lemma 1.1 that $\mathcal{R}_0 > h(x_0) \geq \Phi(x_0, \sigma_0)$ and therefore $C_0 > 0$ (C_0 is above C_k with $k = 0$). Similar to the proof of **Case (i)**, (3.6) holds at iteration 0 for all $0 < \alpha_{0,l} \leq C_0$. Combining it with Lemma 4.6 gives that there exists $\alpha_0 \leq \min\{\tilde{C}, C_0\}$ (\tilde{C} is from Lemma 4.6) such that both (3.6) and (3.5) hold at iteration 0, and therefore $h(x_1) < \mathcal{R}_0$. Similarly, we can prove that $h(x_s) < \mathcal{R}_s$ for $s = 1, 2, \dots, k$. As a result, $\mathcal{R}_k > h(x_k) \geq \Phi(x_k, \sigma_k)$ and $C_k > 0$. Similar to the proof of **Case (i)**, (3.6) holds for all $\alpha_{k,l} \in (0, C_k]$. In another case that $\mathcal{R}_k = \mathcal{R}_{k-1} = \dots = \mathcal{R}_{k^-}$ and (3.4) does not hold at iteration k^- , by Algorithm J and Lemma 4.5, we have that $\mathcal{R}_{k^-} = h(x_{k^-}) > 0$. It should be noted that $h(x_{k^-}) > \Phi(x_{k^-}, \sigma_{k^-})$, otherwise x_{k^-} is an infeasible stationary point and the proposed algorithm stops at iteration k^- . Similar to the proof of **Case (i)**, (3.6) holds for all $\alpha_{k,l} \in (0, C_{k^-}]$, which together with Lemma 4.6 yields that both (3.6) and (3.5) hold at iteration k^- for some $\alpha_{k^-} > 0$. This implies that $h(x_{k^-+1}) < \mathcal{R}_{k^-} = \mathcal{R}_{k^-+1}$. Similarly, we can prove that $h(x_s) < \mathcal{R}_s$ for $s = k^- + 1, k^- + 2, \dots, k$. Similar to the proof of **Case (i)**, (3.6) holds for all $\alpha_{k,l} \in (0, C_k]$.

Thereby, we proved that (3.6) holds for all $\alpha_{k,l} \in (0, C_k]$ in all cases, which together with Lemma 4.6 yields that both (3.6) and (3.5) hold for all $\alpha_{k,l} \in (0, \bar{\alpha}_k]$, where $\bar{\alpha}_k := \min\{C_k, \tilde{C}\}$. Therefore, the inner loop terminates so long as $\alpha_{k,l}$ locates in the interval $(0, \bar{\alpha}_k]$, which implies that the inner loop terminates finitely. \square

Remark 4.3 By Algorithm A (line 13–line 24), α_k is the largest scalar $\alpha_{k,l} \in (0, 1]$ such that (3.6) is satisfied and ((3.4) is not satisfied or (3.5) is satisfied). Theorem 4.1 implies that $\alpha_k \geq t\bar{\alpha}_k$ for all k .

Remark 4.4 Theorem 4.1 has a byproduct that $\mathcal{R}_k \geq h(x_k)$ for all k if all iterates are neither KKT points nor infeasible stationary points. Actually, the proof of **Case (i)** shows that $\mathcal{R}_k = h(x_k)$ in the case that (3.4) holds at iteration k , and the proof of **Case (ii)** shows that $\mathcal{R}_k > h(x_k)$ in the case that (3.4) does not hold at iteration k .

Since the finite termination of the inner loop may not be enough to guarantee the desired convergence results of our algorithm, as a preparation for the ultimate convergence results, we establish the following lemma, which shows that α_k is greater than some positive constant in some situations.

Lemma 4.7 *Let Assumptions (A1)–(A3) hold. Suppose that $\mathcal{R}_k - \Phi(x_k, \sigma_k) \geq \epsilon$ for $k \in \mathcal{K}$, where $\epsilon > 0$ is a scalar and \mathcal{K} is an infinite index set. Then there exists a scalar $\alpha_{\min} > 0$, such that $\alpha_k \geq \alpha_{\min}$ for all $k \in \mathcal{K}$.*

Proof Let $\underline{C} = \frac{2(1-\eta)\epsilon}{M\bar{\beta}^2}$ and $\alpha_{\min} = t \min\{\tilde{C}, \underline{C}\}$, where $\bar{\beta}$ and t are from Algorithm A and \tilde{C} is from Lemma 4.6. Since $\mathcal{R}_k - \Phi(x_k, \sigma_k) \geq \epsilon$ for $k \in \mathcal{K}$, we obtain from the definition of \underline{C} , (4.6), and Remark 4.1 that $\underline{C} \leq C_k$ for all $k \in \mathcal{K}$. It therefore follows that $x_k + \alpha_{k,l}d_k$ is accepted for all $\alpha_{k,l} \leq \frac{1}{t}\alpha_{\min}$. From Remark 4.3, we have that $\alpha_k \geq \alpha_{\min}$ for all $k \in \mathcal{K}$. \square

Next, we prove that, under some conditions, the iteration sequence generated by Algorithm A converges to feasible stationary points of h or there exists some accumulation point of the iteration sequence being an infeasible stationary point of h .

Lemma 4.8 *Let Assumptions (A1)–(A3) hold. If there exist infinite iterations such that (3.4) does not hold and the outer loop of Algorithm A cannot terminate finitely, then either there exists an accumulation point of $\{x_k\}$ which is an infeasible stationary point of h , or*

$$\lim_{k \rightarrow +\infty} h(x_k) = 0. \tag{4.7}$$

Proof If there exists some accumulation point of $\{x_k\}$ being an infeasible stationary point, the conclusion follows. In the following, we suppose that none of the accumulation points of $\{x_k\}$ is an infeasible stationary point. Assume that the subsequence $\{x_{k_i}\}$ of $\{x_k\}$ consists of all iterates such that (3.4) does not hold at all k_i . Algorithm J implies that $h(x_{k_i}) = \mathcal{R}_{k_i}$ for all k_i , and $\mathcal{R}_s = \mathcal{R}_{k_i}$ for any index s satisfying $k_i < s < k_{i+1}$ (if s exists). It follows with Remark 4.4 that (4.7) is equivalent to $\lim_{k_i \rightarrow +\infty} \mathcal{R}_{k_i} = 0$. Noting line 3.5 in Algorithm A, condition (3.6) is required at each iteration. Hence, for each k_i ,

$$\mathcal{R}_{k_{i+1}} - \mathcal{R}_{k_i} \leq \alpha_{k_{i+1}-1}\eta (\Phi(x_{k_{i+1}-1}, \sigma_{k_{i+1}-1}) - \mathcal{R}_{k_i}). \tag{4.8}$$

Remark 4.4 and Lemma 1.1 guarantee the positive right-hand side of (4.8), which concludes that $\{\mathcal{R}_{k_i}\}$ is a decreasing sequence and so is $\{h(x_{k_i})\}$. If (4.7) is not true, then there exist some scalar $\epsilon > 0$ such that

$$\mathcal{R}_{k_i} \geq \epsilon \tag{4.9}$$

for all k_i . By Lemma 1.1, $\Phi(x_{k_{i+1}-1}, \sigma_{k_{i+1}-1}) \leq h(x_{k_{i+1}-1})$. If $h(x_{k_{i+1}-1}) \rightarrow 0$ as $k_i \rightarrow \infty$, it follows that $\Phi(x_{k_{i+1}-1}, \sigma_{k_{i+1}-1}) \rightarrow 0$ as $k_i \rightarrow \infty$ and therefore

$$\Phi(x_{k_{i+1}-1}, \sigma_{k_{i+1}-1}) - \mathcal{R}_{k_i} < -\frac{\epsilon}{2} \tag{4.10}$$

for all sufficiently large k_i . Otherwise, there exists some accumulation point \bar{x} of $\{x_{k_{i+1}-1}\}$ such that $h(\bar{x}) > 0$. Without loss of generality, we assume that $\{x_{k_{i+1}-1}\}$ converges to \bar{x} . Since \bar{x} is not supposed to be an infeasible stationary point, there exists a scalar $\bar{\epsilon} > 0$ such that $\Phi(x_{k_{i+1}-1}, \sigma_{k_{i+1}-1}) - h(x_{k_{i+1}-1}) < -\bar{\epsilon}$ for all sufficiently large k_i . This together with Remark 4.4 yields that

$$\Phi(x_{k_{i+1}-1}, \sigma_{k_{i+1}-1}) - \mathcal{R}_{k_i} < -\bar{\epsilon} \tag{4.11}$$

for all sufficiently large k_i . In view of (4.10) and (4.11), and applying Lemma 4.7, we have that $\alpha_{k_{i+1}-1} \geq \alpha_{\min} > 0$ for all sufficiently large k_i . Substituting this into (4.8) and combining with (4.10) and (4.11), we obtain that

$$\mathcal{R}_{k_{i+1}} - \mathcal{R}_{k_i} \leq -\alpha_{\min}\eta\epsilon_0, \tag{4.12}$$

holds for all sufficiently large k_i , where ϵ_0 is $\frac{\epsilon}{2}$ or $\bar{\epsilon}$. Due to monotonicity and boundedness of the sequence $\{\mathcal{R}_{k_i}\}$, it is convergent, and therefore the left-hand side (4.12) tends to zero as $k_i \rightarrow \infty$, which contradicts the right-hand side of (4.12). Hence, the claim of this lemma is true. \square

Now we are ready to prove our main conclusion in this section.

Theorem 4.2 *Let Assumptions (A1)–(A3) hold. Assume the sequence $\{x_k\}$ is generated by Algorithm A. Then there exists an accumulation point that is either a KKT point, a feasible point at which the RCPLD condition fails to hold or a stationary point of h that is infeasible for the problem (NLP).*

Proof Since Theorem 4.1 implies that the inner loop terminates finitely, we only need consider the case when the outer loop is infinite. For this purpose, we define an index set

$$\mathcal{H} = \{k | (3.4) \text{ does not hold at iteration } k\}.$$

There are two cases to be considered.

Case (i): \mathcal{H} is an infinite set.

By Lemma 4.8, either there exists some accumulation point of $\{x_k\}$ being an infeasible stationary point or $\{h(x_k)\}$ converges to zero. If the accumulation point is an infeasible stationary point for the problem (NLP), then the conclusion has already followed. In the following, we take into account the case that $h(x_k) \rightarrow 0$ as $k \rightarrow \infty$. Since $\{x_k\}_{k \in \mathcal{H}} \subset S$ is bounded, there exists an infinite subset $\mathcal{H}_1 \subseteq \mathcal{H}$, such that $x_k \rightarrow \bar{x}$ for $k \in \mathcal{H}_1$ as $k \rightarrow +\infty$. As $h(x_k) \rightarrow 0$, it follows that \bar{x} is a feasible point. If the the RCPLD condition fails to hold at \bar{x} , the conclusion of this theorem follows. Otherwise, we assume that \bar{x} is a feasible point at which the RCPLD condition holds. We need to prove that \bar{x} is a KKT point for the problem (NLP).

Suppose \mathcal{E}_0 is a subset of \mathcal{E} such that $\{\nabla c_i(\bar{x})\}_{i \in \mathcal{E}_0}$ is a maximal linearly independent subset of $\{\nabla c_i(\bar{x})\}_{i \in \mathcal{E}}$. The RCPLD implies that $\nabla c_{\mathcal{E}}(\bar{x})$ has constant row rank in a small neighborhood of \bar{x} . Hence, $\{\nabla c_i(x_k)\}_{i \in \mathcal{E}_0}$ is also a maximal linearly indepent subset of $\{\nabla c_i(x_k)\}_{i \in \mathcal{E}}$ for all sufficiently large k . It follows that there exists $\tilde{\lambda}_{k,i} \in \mathbb{R}$, $i \in \mathcal{E}_0$ such that

$$\sum_{i \in \mathcal{E}} \lambda_{k,i} \nabla c_i(x_k) = \sum_{i \in \mathcal{E}_0} \tilde{\lambda}_{k,i} \nabla c_i(x_k) \tag{4.13}$$

for all sufficiently large k , where the left-hand side is from the first equality of (3.3). By [2, Lemma 1], for any $k \in \mathcal{H}_1$, there exists an index set $\mathcal{I}_0^k \subseteq \mathcal{I}$ and $0 \leq \tilde{\lambda}_{k,i} \in$

\mathbb{R} , $i \in \mathcal{I}_0^k$, such that $\lambda_{k,i} \neq 0$ for all $i \in \mathcal{I}_0^k$, the vectors $\nabla c_i(x_k)$, $i \in \mathcal{E}_0 \cup \mathcal{I}_0^k$ are linearly independent and

$$\sum_{i \in \mathcal{E}_0} \tilde{\lambda}_{k,i} \nabla c_i(x_k) + \sum_{i \in \mathcal{I}} \lambda_{k,i} \nabla c_i(x_k) = \sum_{i \in \mathcal{E}_0 \cup \mathcal{I}_0^k} \tilde{\lambda}_{k,i} \nabla c_i(x_k), \tag{4.14}$$

where the second term in the left-hand side is from (3.3). Since \mathcal{I} is a finite set, there exists an infinite set $\mathcal{H}_c \subseteq \mathcal{H}_1$ such that $\mathcal{I}_0^k \equiv \mathcal{I}_0$ for all $k \in \mathcal{H}_c$, where \mathcal{I}_0 is a constant subset of \mathcal{I} . Without loss of generality, we assume that $\mathcal{H}_c = \mathcal{H}_1$. Then, $\nabla c_i(x_k)$, $i \in \mathcal{E}_0 \cup \mathcal{I}_0$ are linearly independent, and substituting (4.13) into (4.14) gives

$$\sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_{k,i} \nabla c_i(x_k) = \sum_{i \in \mathcal{E}_0 \cup \mathcal{I}_0} \tilde{\lambda}_{k,i} \nabla c_i(x_k) \tag{4.15}$$

for all sufficiently large $k \in \mathcal{H}_1$. Now we prove that $\{\tilde{\lambda}_k\}_{k \in \mathcal{H}_1}$ is a bounded sequence. If it is not true, without loss of generality, we may assume that

$$\begin{aligned} \lim_{k \rightarrow +\infty, k \in \mathcal{H}_1} \|\tilde{\lambda}_k\| &= +\infty, & \lim_{k \rightarrow +\infty, k \in \mathcal{H}_1} \frac{\tilde{\lambda}_k}{\|\tilde{\lambda}_k\|} &= \mu, \\ \lim_{k \rightarrow +\infty, k \in \mathcal{H}_1} \frac{\lambda_k^u - \lambda_k^l}{\|\tilde{\lambda}_k\|} &= \nu, & \lim_{k \rightarrow +\infty, k \in \mathcal{H}_1} d_k &= \bar{d}, \\ \lim_{k \rightarrow +\infty, k \in \mathcal{H}_1} \sigma_k &= \sigma_H, & \lim_{k \rightarrow +\infty, k \in \mathcal{H}_1} \beta_k &= \beta_H, \end{aligned}$$

and $\|\mu\| = 1$ and $\mu_i \geq 0$, $i \in \mathcal{I}_0$. With the help of $\tilde{\lambda}_k$, \mathcal{E}_0 and \mathcal{I}_0 , the KKT conditions for the problem (3.2) can be rewritten as

$$\begin{cases} g_k + \sum_{i \in \mathcal{E}_0 \cup \mathcal{I}_0} \tilde{\lambda}_{k,i} \nabla c_i(x_k) + B_k d_k + \lambda_k^u - \lambda_k^l = 0, \\ \tilde{\lambda}_{k,i} (\nabla c_i(x_k)^T d_k + c_i(x_k) - \Phi(x_k, \sigma_k)) = 0, \quad i \in \mathcal{I}_0, \\ (d_k - \beta_k e_n)^T \lambda_k^u = 0, \quad (d_k + \beta_k e_n)^T \lambda_k^l = 0, \\ \nabla c_i(x_k)^T d_k + c_i(x_k) - r_i(x_k, \sigma_k) = 0, \quad i \in \mathcal{E}, \\ \nabla c_i(x_k)^T d_k + c_i(x_k) \leq \Phi(x_k, \sigma_k), \quad i \in \mathcal{I}, \quad \|d_k\|_\infty \leq \beta_k, \\ \tilde{\lambda}_{k,i} \geq 0, \quad i \in \mathcal{I}_0, \quad \lambda_{k,i}^u \geq 0, \quad \lambda_{k,i}^l \geq 0, \quad i \in \{1, 2, \dots, n\}, \end{cases} \tag{4.16}$$

where the first equation follows from (4.15), and the second equation follows from the fact $\lambda_{k,i} \neq 0$ for all $k \in \mathcal{I}_0$. Dividing the first equation in (4.16) by $\|\tilde{\lambda}_k\|$ and letting $k \rightarrow +\infty, k \in \mathcal{H}_1$, we have

$$\sum_{i \in \mathcal{E}_0 \cup \mathcal{I}_0} \mu_i \nabla c_i(\bar{x}) + \nu = 0. \tag{4.17}$$

Since \bar{x} is feasible for the problem (NLP), it follows that $c_i(\bar{x}) = 0$ with $i \in \mathcal{E}$, $c_i(\bar{x}) \leq 0$ with $i \in \mathcal{I}$, $r(\bar{x}, \sigma_H) = 0$ and $\Phi(\bar{x}, \sigma_H) = 0$. By dividing the second and

fourth equations in (4.16) by $\|\tilde{\lambda}_k\|$, it follows with the third equation in (4.16) that

$$\mu_i \nabla c_i(\bar{x})^T \bar{d} = -\mu_i c_i(\bar{x}), \quad i \in \mathcal{I}_0, \quad \nabla c_i(\bar{x})^T \bar{d} = 0, \quad i \in \mathcal{E}_0, \tag{4.18}$$

and

$$v^T \bar{d} = \lim_{k \rightarrow +\infty, k \in \mathcal{H}_1} \frac{\beta_H e_n^T (\lambda_k^u + \lambda_k^l)}{\|\tilde{\lambda}_k\|} \geq 0. \tag{4.19}$$

Then multiplying (4.17) by \bar{d}^T and using (4.18), we have $v^T \bar{d} = \sum_{i \in \mathcal{I}_0} \mu_i c_i(\bar{x}) \leq 0$. This together with the right-hand side of (4.19) and the definition of v implies $v = 0$.

Therefore, equation (4.17) reduces to $\sum_{i \in \mathcal{E}_0 \cup \mathcal{I}_0} \mu_i \nabla c_i(\bar{x}) = 0$, which implies that $\{\nabla c_i(\bar{x})\}_{i \in \mathcal{E}_0 \cup \mathcal{I}_0}$ is positive linearly dependent. By the assumptions that the RCPLD condition holds at \bar{x} and $x_k \rightarrow \bar{x}$, we have that $\{\nabla c_i(x_k)\}_{i \in \mathcal{E}_0 \cup \mathcal{I}_0}$ is linearly dependent for all sufficiently large k . This contradicts the fact that $\{\nabla c_i(x_k)\}_{i \in \mathcal{E}_0 \cup \mathcal{I}_0}$ is linearly independent for all $k \in \mathcal{H}_1$.

Consequently, the sequence $\{\tilde{\lambda}_k\}_{k \in \mathcal{H}_1}$ is bounded. We assume that $\|\tilde{\lambda}_k\|_1 \leq \bar{M}$ for some scalar $\bar{M} > 0$ and all $k \in \mathcal{H}_1$. Since $h(x_k) \rightarrow 0$ as $k \rightarrow +\infty$, it follows that $\Phi(x_k, \sigma_k) \rightarrow 0$ as $k \rightarrow +\infty$. If there exists a scalar $\epsilon > 0$, such that $\|d_k\| > \epsilon$ and $\|\tilde{d}_{k+1}\| > \epsilon$ for all $k \in \mathcal{H}_1$, combining with Assumption (A3), it follows that there exists an integer $j_0 > 0$, such that for all $k > j_0, k \in \mathcal{H}_1$,

$$\Phi(x_k, \sigma_k) + h(x_k) \leq \min \left\{ \frac{(1 - \xi)d_k^T B_k d_k}{\bar{M}}, \quad \zeta \|d_k\| \cdot \|\tilde{d}_{k+1}\| \right\}, \tag{4.20}$$

where ξ and ζ are from (3.4). It follows with (4.16) and boundedness of $\{\tilde{\lambda}_k\}_{k \in \mathcal{H}_1}$ that for all $k > j_0$ and $k \in \mathcal{H}_1$,

$$\begin{aligned} g_k^T d_k &= - \sum_{i \in \mathcal{E}_0 \cup \mathcal{I}_0} \tilde{\lambda}_{k,i} \nabla c_i(x_k)^T d_k - d_k^T B_k d_k - \beta_k (\|\lambda_k^l\|_1 + \|\lambda_k^u\|_1) \\ &= \sum_{i \in \mathcal{E}_0 \cup \mathcal{I}_0} \tilde{\lambda}_{k,i} c_i(x_k) - \sum_{i \in \mathcal{E}_0} \tilde{\lambda}_{k,i} r_i(x_k, \sigma_k) - \sum_{i \in \mathcal{I}_0} \tilde{\lambda}_{k,i} \Phi(x_k, \sigma_k) - d_k^T B_k d_k \\ &\quad - \beta_k (\|\lambda_k^l\|_1 + \|\lambda_k^u\|_1) \\ &\leq \bar{M} (h(x_k) + \Phi(x_k, \sigma_k)) - d_k^T B_k d_k, \\ &\leq -\xi d_k^T B_k d_k, \end{aligned}$$

which together with (4.20) contradicts the definition of \mathcal{H} . Thereby, there exists some infinite subset $\mathcal{H}_2 \subset \mathcal{H}_1$ such that

$$\lim_{k \in \mathcal{H}_2, k \rightarrow +\infty} \|d_k\| = 0 \quad \text{or} \quad \lim_{k \in \mathcal{H}_2, k \rightarrow +\infty} \|\tilde{d}_{k+1}\| = 0.$$

Without loss of generality, we assume that $\mathcal{H}_2 = \mathcal{H}_1$. According to Algorithm A, β_k is bounded away from zero for all $k \in \mathcal{H}_1$, and the trust region constraint $\|d_k\|_\infty \leq \beta_k$

must be inactive for all sufficiently large $k \in \mathcal{H}_1$. By (4.16), we have

$$\begin{cases} g_k + \sum_{i \in \mathcal{E}_0 \cup \mathcal{I}_0} \tilde{\lambda}_{k,i} \nabla c_i(x_k) + B_k d_k = 0, \\ \tilde{\lambda}_{k,i} (\nabla c_i(x_k))^T d_k + c_i(x_k) - \Phi(x_k, \sigma_k) = 0, \tilde{\lambda}_{k,i} \geq 0, i \in \mathcal{I}_0, \\ \nabla c_i(x_k)^T d_k + c_i(x_k) - r_i(x_k, \sigma_k) = 0, i \in \mathcal{E}, \\ \nabla c_i(x_k)^T d_k + c_i(x_k) \leq \Phi(x_k, \sigma_k), i \in \mathcal{I}, \end{cases} \quad (4.21)$$

for all sufficiently large $k \in \mathcal{H}_1$. Since $\{\tilde{\lambda}_k\}_{k \in \mathcal{H}_1}$ and $\{B_k\}$ are bounded in norm, both of them have convergent subsequences. We assume, without loss of generality, that

$$\lim_{k \in \mathcal{H}_1, k \rightarrow \infty} \tilde{\lambda}_k = \tilde{\lambda}, \quad \lim_{k \in \mathcal{H}_1, k \rightarrow \infty} B_k = \bar{B}.$$

Using $x_k \rightarrow \bar{x}$, $\|d_k\| \rightarrow 0$, $h(x_k) \rightarrow 0$, and $\Phi(x_k, \sigma_k) \rightarrow 0$ as $k \rightarrow \infty$ and $k \in \mathcal{H}_1$, we obtain from (4.21) and Assumption (A2) that

$$\begin{cases} \nabla f(\bar{x}) + \sum_{i \in \mathcal{E}_0 \cup \mathcal{I}_0} \tilde{\lambda}_i \nabla c_i(\bar{x}) = 0, \\ \tilde{\lambda}_i c_i(\bar{x}) = 0, \tilde{\lambda}_i \geq 0, i \in \mathcal{I}_0, \\ c_i(\bar{x}) = 0, i \in \mathcal{E}, \\ c_i(\bar{x}) \leq 0, i \in \mathcal{I} \end{cases}$$

holds for all sufficiently large $k \in \mathcal{H}_1$. Setting $\bar{\lambda}_i = \begin{cases} \tilde{\lambda}_i, & i \in \mathcal{I}_0 \\ 0, & i \in \mathcal{I} \setminus \mathcal{I}_0 \end{cases}$ yields that the KKT conditions are satisfied at $(\bar{x}, \bar{\lambda})$. That is, \bar{x} is a KKT point.

Case (ii): \mathcal{H} is a finite set.

In this case, there exists an integer $j_0 > 0$ such that (3.4) holds for all $k > j_0$. By the mechanism of Algorithm A, the relaxed reduction condition (3.5) on f is required to be satisfied for all $k > j_0$. It therefore follows with Remark 4.1 that

$$\hat{f}_k - f(x_{k+1}) \geq \alpha_k \sigma \min(\xi d_k^T B_k d_k, \tau \|d_k\|^v) \geq \alpha_k \sigma \min(\xi \delta \|d_k\|^2, \tau \|d_k\|^v). \quad (4.22)$$

Without loss of generality, we assume that (4.22) is true for all k . Similarly to the proof of [22, Theorem], we obtain that $\alpha_k \|d_k\| \rightarrow 0$ as $k \rightarrow +\infty$.

The mechanism of Algorithm J implies that $\mathcal{R}_k = \mathcal{R}_{k-1} = \mathcal{R}_{j_0} (> 0)$ for all $k > j_0$ since (3.4) holds for $k > j_0$. Without loss of generality, we assume that

$$\mathcal{R}_k = \epsilon_0 > 0 \quad (4.23)$$

for all sufficiently large k , where $\epsilon_0 > 0$ is a scalar.

Since $\{x_k\} \subset S$ is bounded, there exists a subsequence $\{x_k\}_{k \in \mathcal{G}} \rightarrow \bar{x}$ where \mathcal{G} is an infinite index set. If $h(\bar{x}) > 0$ and $0 \in \partial h(\bar{x})$, we then have from Lemma 1.1 that \bar{x} is an infeasible stationary point of h , which indicates that the conclusion of this theorem follows. If $h(\bar{x}) > 0$ and $0 \notin \partial h(\bar{x})$, it follows from Lemma 4.3 that there exists a scalar $\epsilon_1 > 0$, such that for all sufficiently large $k \in \mathcal{G}$,

$$\mathcal{R}_k - \Phi(x_k, \sigma_k) \geq h(x_k) - \Phi(x_k, \sigma_k) > \epsilon_1,$$

where the first inequality follows from Remark 4.4. If $h(\bar{x}) = 0$, then

$$\Phi(x_k, \sigma_k) \rightarrow 0$$

as $k \rightarrow +\infty$ and $k \in \mathcal{G}$. Hence, by (4.23) we also have $\mathcal{R}_k - \Phi(x_k, \sigma_k) > \epsilon_2$ for some $\epsilon_2 > 0$ and all sufficiently large $k \in \mathcal{G}$.

According to Lemma 4.7, we obtain that there exists a scalar $\alpha_{\min} > 0$, such that $\alpha_k \geq \alpha_{\min}$ for all sufficiently large $k \in \mathcal{G}$. Hence, $\|d_k\| \rightarrow 0$ as $k \rightarrow +\infty$ and $k \in \mathcal{G}$, which is a contradiction.

From the above two cases, the proof is complete. □

5 Local convergence

In this section, we prove the local convergence of Algorithm A under the conditions of the strict MFCQ and the SOSC. To our knowledge, previous primal superlinear convergence results for SQP methods requires the assumptions of the LICQ, the SC and the SOSC. Very recently, Fernández, Izmailov and Solodov [13] proved sharp primal superlinear convergence results for some Newtonian methods for constrained optimization. However, they did not give local convergence analysis for some specific SQP-type methods. Our algorithm based on some natural globalization procedure instead of hybrid strategies allows iterates to switch from the global phase to the local phase smoothly. In this section, we establish the local convergence of our algorithm under such weaker conditions.

Let x^* be an accumulation point of $\{x_k\}$ generated by Algorithm A, which is a KKT point of the problem (NLP) (the existence of a KKT point has been proven by Theorem 4.2 in Sect. 4). Let $\lambda^* = (\lambda_1^*, \dots, \lambda_m^*)^T$ be the corresponding multiplier vector. It is well-known that the SMFCQ and the SOSC ensure the isolatedness of x^* and the uniqueness of λ^* corresponding to x^* . The following additional assumptions are needed to ensure the fast local convergence of the proposed algorithm.

- (A4) The SMFCQ holds at x^* .
- (A5) The SOSC holds at (x^*, λ^*) .
- (A6) The condition

$$\lim_{k \rightarrow +\infty} \frac{\|P_{C^*} \left((B_k - \nabla_{xx}^2 L(x^*, \lambda^*)) d_k \right)\|}{\|d_k\|} = 0$$

holds, where C^* is the *critical cone* at x^* , and $P_G(y)$ denotes the Euclidean projection of $y \in \mathbb{R}^n$ onto the closed convex set $G \subset \mathbb{R}^n$.

Remark 5.1 By Assumption (A4) and Lemma 5.2, $QP(x_k)$ and $\widetilde{QP}(x_k)$ are consistent for all sufficiently large k . This implies $\Phi(x_k, \sigma_k) = 0$ for all sufficiently large k . Lemma 5.2 and the boundedness of $\{\sigma_k\}$ yield that the constraint $\|d\|_\infty \leq \sigma_k$ is inactive and $\lambda_k^l = \lambda_k^u = 0$ for all sufficiently large k . Consequently, $MQP(x_k)$ reduces to $QP(x_k)$. Correspondingly, the KKT conditions (3.3) reduce to those for $QP(x_k)$;

that is, the system (3.3) changes to

$$\begin{cases} g_k + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_{k,i} \nabla c_i(x_k) + B_k d_k = 0, \\ \nabla c_i(x_k)^T d_k + c_i(x_k) = 0, \quad i \in \mathcal{E}, \\ \nabla c_i(x_k)^T d_k + c_i(x_k) \leq 0, \quad i \in \mathcal{I}, \\ \lambda_{k,i} (\nabla c_i(x_k)^T d_k + c_i(x_k)) = 0, \quad i \in \mathcal{I}, \\ \lambda_{k,i} \geq 0, \quad i \in \mathcal{I}, \end{cases} \tag{5.1}$$

where $\lambda_k = (\lambda_{k,1}, \lambda_{k,2}, \dots, \lambda_{k,m})^T \in \mathbb{R}^m$.

Remark 5.2 As we mentioned before, Assumption (A4) implies that the multiplier vector λ^* is unique and bounded. Assumptions (A4) and (A5) imply that x^* is a strict local minimizer.

Remark 5.3 Assumption (A6) is required to prove the primal superlinear convergence (see [13]). If LICQ holds at x^* , then $\nabla c_{\mathcal{A}^*}(x^*)$ is full row rank, and the Euclidean projection of y onto \mathcal{C}^* equals P^*y , where

$$P^* := I - \nabla c_{\mathcal{A}^*}(x^*)^T [\nabla c_{\mathcal{A}^*}(x^*) \nabla c_{\mathcal{A}^*}(x^*)^T]^{-1} \nabla c_{\mathcal{A}^*}(x^*).$$

In practice, we point out that even in this situation, it is still not easy to find B_k practically to fulfill Assumption (A6). Analogous difficulty arises in the assumption used in [7, Theorem 3.1]. In theory, however, we remark that all these technical assumptions are important for proving the fast local convergence.

In what follows, we focus on iterations satisfying above conclusions with all sufficiently large k . The following lemma shows that the whole primal-dual sequence (x_k, λ_k) converges to (x^*, λ^*) .

Lemma 5.1 *Let Assumptions (A1)–(A6) hold. Then $(x_k, \lambda_k) \rightarrow (x^*, \lambda^*)$ as $k \rightarrow +\infty$.*

Proof It follows from [35, Lemma 4.2] with slight modification. □

We remark that [35, Lemmas 4.1 and 4.2] can also be proved under Assumptions (A1)–(A6) used here.

The following Lemma shows that $d_k \rightarrow 0$ as $k \rightarrow \infty$.

Lemma 5.2 *Let Assumption (A1)–(A6) hold and d_k be the solution of QP(x_k). Then*

$$\lim_{k \rightarrow +\infty} \|d_k\| = 0.$$

Proof Premultiplying the first equation of (5.1) by B_k^{-1} , we obtain that

$$d_k = -B_k^{-1} \left(g_k + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_{k,i} \nabla c_i(x_k) \right). \tag{5.2}$$

Since x^* is a KKT point of the problem (NLP), by Lemma 5.1, (5.2) and Assumptions (A2)–(A3) we have

$$\lim_{k \rightarrow +\infty} \|d_k\| = 0.$$

□

Next, we shows that the full SQP step d_k provides superlinear convergence.

Theorem 5.1 *Let Assumptions (A1)–(A6) hold. Then*

$$\lim_{k \rightarrow +\infty} \frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} = 0. \tag{5.3}$$

Moreover, $\|d_k\| = \Theta(\|x_k - x^*\|)$.

Proof From [13, Theorem 4.1] and Assumptions (A1)–(A6), we have that (5.3) is true. It then follows

$$\frac{\|x_k + d_k - x^*\|}{\|x_k - x^*\|} \geq \left| \frac{\|d_k\|}{\|x_k - x^*\|} - 1 \right| \rightarrow 0, \text{ as } k \rightarrow +\infty.$$

Therefore,

$$\frac{\|d_k\|}{\|x_k - x^*\|} \rightarrow 1, \text{ as } k \rightarrow +\infty,$$

which implies $\|d_k\| = \Theta(\|x_k - x^*\|)$. □

It follows from Theorem 5.1 that our algorithm has the rate of superlinear convergence if the full SQP steps are accepted for all sufficiently large k . Next, we establish some preliminary result for proving acceptance of d_k for all sufficiently large k .

Lemma 5.3 *Let Assumptions (A1)–(A6) hold. Then it follows that*

$$\|\tilde{d}_{k+1}\| = o(\|d_k\|), \text{ and} \tag{5.4}$$

$$h(x_k + d_k + \tilde{d}_{k+1}) = \mathcal{O}(\|\tilde{d}_{k+1}\|^2). \tag{5.5}$$

Proof By the mechanism of Algorithm A, the SOC step \tilde{d}_{k+1} is actually the full SQP step d_{k+1} . The only difference between \tilde{d}_{k+1} and d_{k+1} is that we do not know in advance if \tilde{d}_{k+1} is accepted by the acceptance conditions of Algorithm A. However, it does not interfere with the conclusion of Theorem 5.1. The reason is that \tilde{d}_{k+1} is calculated at $x_k + d_k$ where d_k is also a full SQP step. Therefore, the equation $\|\tilde{d}_{k+1}\| = o(\|d_k\|)$ follows from Theorem 5.1. As for (5.5), we can prove it by applying Taylor Expansion formula directly. For any $i \in \mathcal{E} \cup \mathcal{I}$, Taylor Expansion formula gives

$$c_i(x_k + d_k + \tilde{d}_{k+1}) = c_i(x_k + d_k) + \nabla c_i(x_k + d_k)^T \tilde{d}_{k+1} + \mathcal{O}(\|\tilde{d}_{k+1}\|^2),$$

which together with $\widetilde{\text{QP}}(x_k + d_k)$ and the definition of $h(x)$ yields (5.5). □

In order to prove the local convergence of Algorithm A, we introduce the penalty function

$$\Phi_\rho(x) = f(x) + \rho h(x)$$

which is used just for the proof. Here $\rho > \|\lambda^*\|_\infty$ is the penalty parameter. The following lemma is from [1, Lemma 2.2], whose proof can also be inferred from [8].

Lemma 5.4 *Let Assumptions (A2) and (A5) hold. The MFCQ holds at x^* . Then there exists a sufficiently large ρ and some constant $\bar{C} > 0$ such that*

$$f(x) + \rho h(x) \geq f(x^*) + \bar{C}\|x - x^*\|^2 \tag{5.6}$$

for all x in a neighborhood of x^* .

Since Assumption (A4) implies the MFCQ, we still have the conclusion of Lemma 5.4 if the MFCQ condition in Lemma 5.4 is replaced by the SMFCQ condition.

The following several lemmas give preparations for the acceptance of the full SQP steps ultimately.

Lemma 5.5 *Let Assumptions (A1)–(A6) hold. Then there exists an integer $K_1 > 0$, such that if (3.4) holds for some $k \geq K_1$, then the sufficient reduction condition (3.5) holds for $p_k = d_k + \tilde{d}_{k+1}$.*

Proof We only need to show that if (3.4) holds for some sufficiently large iteration number k , then

$$f(x_k + d_k + \tilde{d}_{k+1}) + \sigma\tau\|d_k\|^v \leq f(x_k), \tag{5.7}$$

where $v \in (2, 3]$ is from Algorithm A. Since $f(x)$ and $c_i(x)$, $i \in \mathcal{E} \cup \mathcal{I}$ are twice continuously differentiable, using Taylor Expansion formula, we have

$$\begin{aligned} & f(x_k + d_k + \tilde{d}_{k+1}) + \lambda^{*T}c(x_k + d_k + \tilde{d}_{k+1}) - f(x^*) \\ &= L(x_k + d_k + \tilde{d}_{k+1}, \lambda^*) - L(x^*, \lambda^*) \\ &= \nabla_x L(x^*, \lambda^*)^T(x_k + d_k + \tilde{d}_{k+1} - x^*) + \mathcal{O}(\|x_k + d_k + \tilde{d}_{k+1} - x^*\|^2) \\ &= \mathcal{O}(\|x_k + d_k + \tilde{d}_{k+1} - x^*\|^2), \end{aligned}$$

where the last equality follows from $\nabla_x L(x^*, \lambda^*) = 0$. We then have

$$\begin{aligned} & f(x_k + d_k + \tilde{d}_{k+1}) + \sigma\tau\|d_k\|^v \\ &= f(x^*) - \lambda^{*T}c(x_k + d_k + \tilde{d}_{k+1}) + \sigma\tau\|d_k\|^v + \mathcal{O}(\|x_k + d_k + \tilde{d}_{k+1} - x^*\|^2) \\ &= f(x^*) - \lambda^{*T}c(x_k + d_k + \tilde{d}_{k+1}) + \mathcal{O}(\|d_k\|^v) + \mathcal{O}(\|x_k + d_k + \tilde{d}_{k+1} - x^*\|^2) \\ &= f(x^*) - \lambda^{*T}c(x_k + d_k + \tilde{d}_{k+1}) + o(\|x_k - x^*\|^2). \quad (\text{by (5.4) and Theorem 5.1}) \end{aligned} \tag{5.8}$$

Similar to the proof of Lemma 5.1, we readily prove that $\tilde{\lambda}_{k+1} \rightarrow \lambda^*$ as $k \rightarrow \infty$, where $\tilde{\lambda}_{k+1}$ is the multiplier vector corresponding to $\widehat{\text{QP}}(x_k + d_k)$. Hence, we conclude that $\tilde{\lambda}_{k,i} > 0$ for sufficiently large k and for any $i \in \mathcal{A}^+(x^*)$, where $\mathcal{A}^+(x^*)$ is the set of indice of strongly active constraints at x^* which is defined at the end of

Sect. 2. It follows from the complementarity condition for $\widetilde{QP}(x_k + d_k)$ that $c_i(x_k + d_k) + \nabla c_i(x_k + d_k)^T \tilde{d}_{k+1} = 0, i \in \mathcal{A}^+(x^*)$ for all sufficiently large k . Using Taylor Expansion formula, we have that

$$\begin{aligned} & c_i(x_k + d_k + \tilde{d}_{k+1}) \\ &= c_i(x_k + d_k) + \nabla c_i(x_k + d_k)^T \tilde{d}_{k+1} + \mathcal{O}(\|\tilde{d}_{k+1}\|^2) \\ &= \mathcal{O}(\|\tilde{d}_{k+1}\|^2) = o(\|x_k - x^*\|^2) \quad (\text{by (5.4) and Theorem 5.1}) \end{aligned}$$

for all $i \in \mathcal{E} \cup \mathcal{A}^+(x^*)$, which leads to $\lambda^{*T} c(x_k + d_k + \tilde{d}_{k+1}) = \sum_{i \in \mathcal{E} \cup \mathcal{A}^+(x^*)} \lambda_i^* c_i(x_k + d_k + \tilde{d}_{k+1}) = o(\|x_k - x^*\|^2)$, where the first equality follows from the fact $\lambda_i^* = 0, i \in \mathcal{I} \setminus \mathcal{A}^+(x^*)$. Substituting this into (5.8) yields

$$\begin{aligned} & f(x_k + d_k + \tilde{d}_{k+1}) + \sigma \tau \|d_k\|^p \\ &= f(x^*) + o(\|x_k - x^*\|^2). \end{aligned} \tag{5.9}$$

Since the MFCQ implies the SMFCQ, it follows from Assumptions (A2), (A4), (A5) and Lemma 5.4 that (5.6) holds when x is close to x^* sufficiently. Then it follows that

$$\begin{aligned} f(x_k) &\geq f(x^*) - \rho h(x_k) + \bar{C} \|x_k - x^*\|^2 \\ &= f(x^*) + \mathcal{O}(\|d_k\| \cdot \|\tilde{d}_{k+1}\|) + \bar{C} \|x_k - x^*\|^2 \quad (\text{by 3.4a}) \\ &\geq f(x^*) + \frac{\bar{C}}{2} \|x_k - x^*\|^2. \quad (\text{by (5.4) and Theorem 5.1}) \end{aligned}$$

Thus, combining with (5.9), we obtain that there exists an integer $K_1 > 0$ such that (5.7) claims if (3.4) holds for some $k \geq K_1$. □

Lemma 5.6 *Let Assumptions (A1)–(A6) hold. Suppose \tilde{d}_{k+1} is computed (That is, FLAG=1). Then there exists a scalar $\mathcal{Q} > 0$ and an integer $K_2(\geq K_1)$ (K_1 is from Lemma 5.5) such that the following statement is true: if (3.4) does not hold for any $k \geq K_2$,*

$$h(x_k) \geq \mathcal{Q} \|d_k\| \cdot \|\tilde{d}_{k+1}\|. \tag{5.10}$$

Proof Lemma 5.1 has showed that $\lambda_k \rightarrow \lambda^*$ and therefore $\{\lambda_k\}$ is bounded. Without loss of generality, we assume that $\|\lambda_k\|_1 \leq M$ for all k , where M is from Remark 4.1. Since $-g_k^T d_k < \xi d_k^T B_k d_k$ implies

$$\begin{aligned} -\xi d_k^T B_k d_k &< g_k^T d_k \\ &= -d^T \nabla c(x_k) \lambda_k - d_k^T B_k d_k, \quad (\text{by (5.1)}) \\ &= \lambda_k^T c(x_k) - d_k^T B_k d_k, \quad (\text{by (5.1)}) \\ &\leq M h(x_k) - d_k^T B_k d_k, \quad (\text{by } \|\lambda_k\|_1 \leq M), \end{aligned}$$

it follows that $h(x_k) \geq \frac{(1-\xi)d_k^T B_k d_k}{M}$. If (3.4) does not hold for some sufficiently large k , we then have

$$h(x_k) \geq \min \left\{ \zeta \|d_k\| \cdot \|\tilde{d}_{k+1}\|, \frac{(1-\xi)d_k^T B_k d_k}{M} \right\} = \mathcal{Q} \|d_k\| \cdot \|\tilde{d}_{k+1}\|,$$

where $\mathcal{Q} := \zeta$, and the equality holds because of Assumption (A3) and Lemma 5.3. Hence, there exists an integer $K_2 (\geq K_1)$ such that if (3.4) fails to hold for any $k \geq K_2$ the inequality (5.10) is satisfied. \square

Lemma 5.7 *Let Assumptions (A1)–(A6) hold. Suppose \tilde{d}_{k+1} is computed (That is, FLAG=1). Then there exists an integer $K_3 (\geq K_2)$ (K_2 is from Lemma 5.6) such that the following statement is true: if*

$$\mathcal{R}_k \geq \mathcal{Q} \|d_k\| \cdot \|\tilde{d}_{k+1}\| \tag{5.11}$$

for any $k \geq K_3$,

$$h(x_k + d_k + \tilde{d}_{k+1}) \leq (1 - \eta)\mathcal{R}_k. \tag{5.12}$$

Proof The conclusion follows from Lemma 5.3. \square

We are now able to show the acceptance of the full SQP steps for all sufficiently large k .

Theorem 5.2 *Let Assumptions (A1)–(A6) hold. Then there exists an integer $K_4 (\geq K_3)$ (K_3 is from Lemma 5.7) such that the trial point $x_k + d_k$ or $x_k + d_k + \tilde{d}_{k+1}$ is accepted for all $k \geq K_4$.*

Proof If $x_k + d_k$ is accepted for all sufficiently large k , then the claim follows trivially. Otherwise, the SOC step \tilde{d}_{k+1} need to be computed. Due to the SMFCQ at x^* , $\widehat{\mathcal{QP}}(x_k + d_k)$ is consistent as x_k approaches x^* . Therefore, \tilde{d}_{k+1} can be computed successfully. In the following, we only need to prove that $x_k + d_k + \tilde{d}_{k+1}$ is accepted for all sufficiently large k . Suppose both x_k and $x_k + d_k$ are not KKT points of the problem (NLP). This means that $d_k \neq 0$ and $\tilde{d}_{k+1} \neq 0$. If there exists only finite iterations such that (3.4) does not hold, it follows from Algorithm J that \mathcal{R}_k will be fixed at some positive constant (say $R_{k_0} > 0$) for all sufficiently large k . By Lemmas 5.2 and 5.3, there exists some $K_4 (\geq K_3)$ such that (5.11) holds for all $k \geq K_4$, and therefore (5.12) is satisfied for all $k \geq K_4$. Without loss of generality, we assume that (3.7) holds for all $k \geq K_4$. It follows from Lemma 5.5 that (3.5) holds. Thereby, for all $k \geq K_4$, $x_k + d_k + \tilde{d}_{k+1}$ is accepted as a new iterate according to Algorithm A.

Next, we consider the case that there exists infinite iterations such that (3.4) does not hold. Let $K_4 (\geq K_3)$ be an integer such that (3.4) does not hold at iteration K_4 , and

$$\mathcal{K} := \{k \geq K_4 | (3.7) \text{ does not hold at iteration } k\}$$

and

$$\mathcal{K}^c := \{k \geq K_4 | (3.7) \text{ holds at iteration } k\}.$$

In this case, \mathcal{K} must be an infinite set. A fact from Algorithm J is that $\mathcal{R}_k = h(x_k)$ for all $k \in \mathcal{K}$. Combining it with Lemmas 5.6 and 5.7, we have that (5.12) for all $k \in \mathcal{K}$, which means that $x_k + d_k + \tilde{d}_{k+1}$ is accepted for all $k \in \mathcal{K}$. If \mathcal{K}^c is a finite set, then the conclusion of this theorem follows immediately. In the following, we assume that \mathcal{K}^c is an infinite set. We only need to prove that $x_k + d_k + \tilde{d}_{k+1}$ is also accepted for all $k \in \mathcal{K}^c$. Due to Lemma 5.5, the sufficient reduction condition (3.5) holds for all $k \in \mathcal{K}^c$. The left task is to prove that (5.12) is satisfied for all $k \in \mathcal{K}^c$.

Let us start from $k = K_4 \in \mathcal{K}$. If $k + 1 \in \mathcal{K}^c$, then set $k^+ = k + 1$; otherwise increase $k := k + 1$ until we find $k \in \mathcal{K}$ and its successor $k^+ := k + 1 \in \mathcal{K}^c$. Since $x_{k+1} = x_k + d_k$ or $x_k + d_k + \tilde{d}_{k+1}$ for this k , we have from (5.4) and Theorem 5.1 that $\|d_{k+1}\| = o(\|d_k\|)$ is true. Without loss of generality, we may assume

$$\mathcal{Q}\|d_{k^+}\| \cdot \|\tilde{d}_{k^++1}\| = \mathcal{Q}\|d_{k+1}\| \cdot \|\tilde{d}_{k+2}\| \leq \mathcal{Q}\|d_k\| \cdot \|\tilde{d}_{k+1}\| \leq h(x_k) = \mathcal{R}_{k^+}, \tag{5.13}$$

where the second inequality follows from Lemma 5.6, and the last equality follows from the update rules of \mathcal{R}_k (see Algorithm J). Applying Lemma 5.7, (5.12) is satisfied and therefore $x_{k^++1} = x_{k^+} + d_{k^+}$ or $x_{k^+} + d_{k^+} + \tilde{d}_{k^++1}$ is accepted. In fact, for all successive indices of k^+ in \mathcal{K}^c (say $k^+ = k + 2, \dots, k_{\max} \in \mathcal{K}^c$, where $k_{\max} + 1 \in \mathcal{K}$), $\mathcal{R}_{k^+} = h(x_k)$ for all $k^+ = k + 2, \dots, k_{\max} \in \mathcal{K}^c$. Since $x_{k+2} = x_{k+1} + d_{k+1}$ or $x_{k+1} + d_{k+1} + \tilde{d}_{k+2}$, without loss of generality, we may assume $\mathcal{Q}\|d_{k^+}\| \cdot \|\tilde{d}_{k^++1}\| \leq \mathcal{Q}\|d_{k+1}\| \cdot \|\tilde{d}_{k+2}\|$, which together with (5.13) leads to

$$\mathcal{Q}\|d_{k^+}\| \cdot \|\tilde{d}_{k^++1}\| \leq \mathcal{R}_{k^+}, \tag{5.14}$$

for $k^+ = k + 2$. Again using Lemma 5.7, we have that $x_{k^++1} = x_{k^+} + d_{k^+}$ or $x_{k^+} + d_{k^+} + \tilde{d}_{k^++1}$ is accepted for $k^+ = k + 2$. Similarly, we can prove that $x_{k^++1} = x_{k^+} + d_{k^+}$ or $x_{k^+} + d_{k^+} + \tilde{d}_{k^++1}$ is accepted for all $k^+ = k + 3, \dots, k_{\max} \in \mathcal{K}^c$ step by step. We now set $k = k_{\max} + 1 \in \mathcal{K}$, and increase k until find $k \in \mathcal{K}$ and $k^+ := k + 1 \in \mathcal{K}^c$. Repeating above process, we have that $x_{k^++1} = x_{k^+} + d_{k^+}$ or $x_{k^+} + d_{k^+} + \tilde{d}_{k^++1}$ is accepted for k^+ and its successive indices in \mathcal{K}^c . By induction, we have that the trial point $x_k + d_k$ or $x_k + d_k + \tilde{d}_{k+1}$ is accepted for all $k \geq K_4$. The proof is complete. \square

Therefore, by Theorem 5.2 the trial point $x_{k+1} = x_k + d_k$ or $x_k + d_k + \tilde{d}_{k+1}$ is accepted as a new iterate for all sufficiently large k . According to Theorem 5.1, we have the following convergence result.

Theorem 5.3 *Let Assumptions (A1)–(A6) hold. Then the sequence $\{x_k\}$ generated by Algorithm A converges to x^* superlinearly.*

6 Numerical results

In this section, we will evaluate the performance of our algorithm on a set of CUTER test problems [5] and provide our preliminary numerical experiments. Our algorithm, Algorithm A, is coded in Matlab 7.9 and run on a PC under OpenSUSE 12.1 system with 2G memory. At each iteration, we use Matlab routines `quadprog` and `linprog`

to solve (3.2) and (1.2), respectively. In our implementation, we choose $\varepsilon = 10^{-6}$ as the tolerance of termination for infeasibility, complementarity and optimality. In other words, we terminate Algorithm A if

$$h(x_k) \leq \varepsilon, \quad \min\{-c_{\mathcal{I}}(x_k), (\lambda_k)_{\mathcal{I}}\} \leq \varepsilon,$$

and

$$\left\| \nabla f(x_k) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_{k,i} \nabla c_i(x_k) \right\|_{\infty} \leq \varepsilon.$$

For the symmetric positive matrix B_k involved in the subproblem $\text{MQP}(x_k)$, we employ the quasi-Newton (BFGS) formula with Powell’s modifications [32] to update B_k with $B_0 = I_n$, which turns out to be a good choice in most cases. Similarly, the quasi-Newton formula is also applied to update \tilde{B}_{k+1} in $\widetilde{\text{QP}}(x_k + d_k)$. Other parameters in Algorithms J and A are as follows: $\eta = 0.01, \sigma = 0.1, \xi = 0.05, \tau = 0.1, \zeta = 1, \nu = 3, t = 0.5, \gamma = 0.01, \mathcal{R}_0 = h(x_0) + 1, \sigma_k = 0.9$ and $\beta_k = 99$ for all k .

6.1 Test for small-size problems from CUTER

Since our current version is coded on Matlab platform where some sophisticated SQP-techniques are not available, we only test some small-size problems with $n \leq 100$. As our algorithm is designed to deal with optimization problems with general constraints, we select minimization problems with nonlinear constrained functions from CUTER test problem collection. The options for our selection are listed in the following table:

Objective function type	: *
Constraints type	: Q O
Regularity	: R
Degree of available derivatives	: *
Problem interest	: *
Explicit internal variables	: *
Number of variables	: in [0, 100]
Number of constraints	: *

where $*$ = everything goes, **Q** and **O** = quadratic or other type (nonlinear; non-constant, etc), and **R** = the problems’ functions are smooth.

With these settings, we conducted on CUTER test environment, and as a result, a set, say \mathcal{P} , of 235 problems are selected. We compare Algorithm A with two state of the art NLP solvers: SNOPT [17] and filterSD (1.0 version) [16]. SNOPT solver is an implementation of a reduced-Hessian SQP algorithm based on a quasi-Newton approximation. The solver filterSD uses a sequential linear constraint programming method which is based on a trust region filter scheme, and moreover, it updates approximate Hessian or reduced Hessian matrices and therefore second-order derivatives are

not required. All experiments for SNOPT and filterSD are conducted in default options (the default tolerance of termination is 10^{-6}).

For comparison purposes, the solvers were compared by using the performance profile suggested by Dolan and Moré [12]. In particular, for a given solver s , we have the value

$$\log_2 \left(\frac{\#\text{iter}(s, \mathbf{p})}{\text{best iter}(\mathbf{p})} \right), \text{ for each problem } \mathbf{p} \in \mathcal{P},$$

where the $\#\text{iter}(s, \mathbf{p})$ denotes the number of iterations (labelled as NIT in the following) that the solver s uses for solving the problem $\mathbf{p} \in \mathcal{P}$, and the “best iter(\mathbf{p})” means the smallest number of iterations, among the three solvers, required to fulfill the termination criterion. We remark that, for a given solver s and a given problem $\mathbf{p} \in \mathcal{P}$, if $\iota = \log_2 \left(\frac{\#\text{iter}(s, \mathbf{p})}{\text{best iter}(\mathbf{p})} \right)$, we can roughly say that the solver s is at worse 2^ι times slower than the best. In a similar manner, we evaluated the numbers of evaluations of the (linear or nonlinear) objective function (labelled as NF) and its gradients (labelled as NG) as well. Finally, counting the evaluations of both linear and nonlinear functions, since the numbers of evaluations of constraint functions and their Jacobian are almost equal to those for the objective function and its gradients, respectively, we did not report these numerical results redundantly.

According to the way for evaluating a set of solvers on a set of tested problems proposed by Dolan and Moré [12], Figs. 1, 2, and 3 show the numerical performance of the tested solvers in terms of NIT, NF and NG, respectively. Here, taking Fig. 1 for example (see more details in [12]), a point (x, y) for a specific solver s is defined by

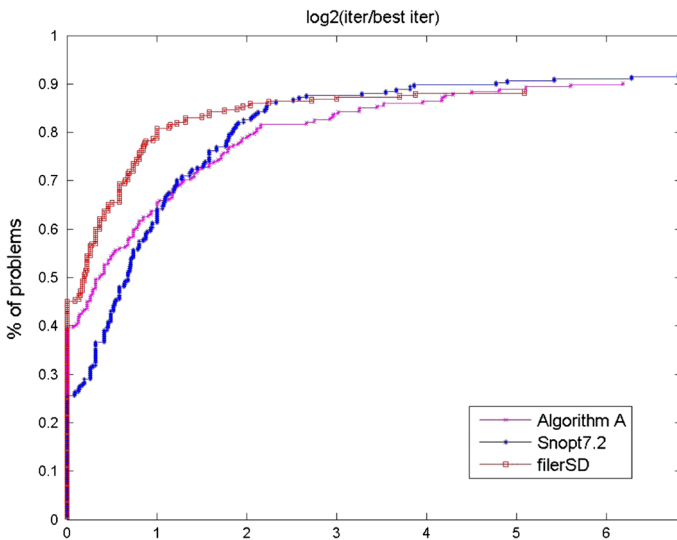


Fig. 1 Performance profile on \mathcal{P} for NIT

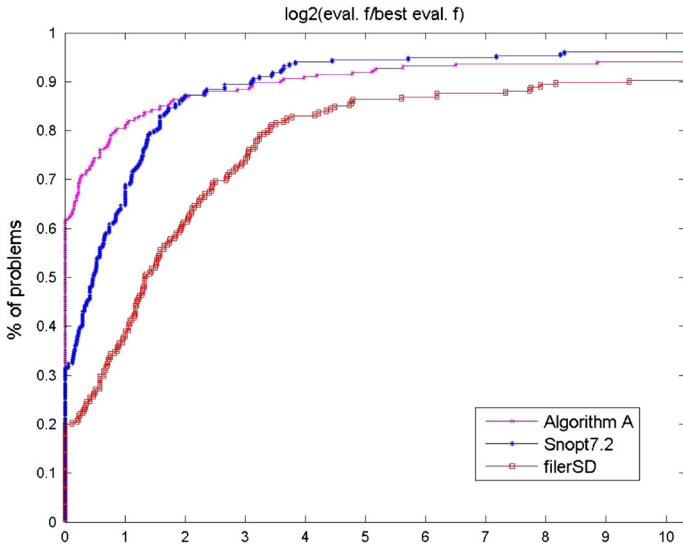


Fig. 2 Performance profile on \mathcal{P} for NF

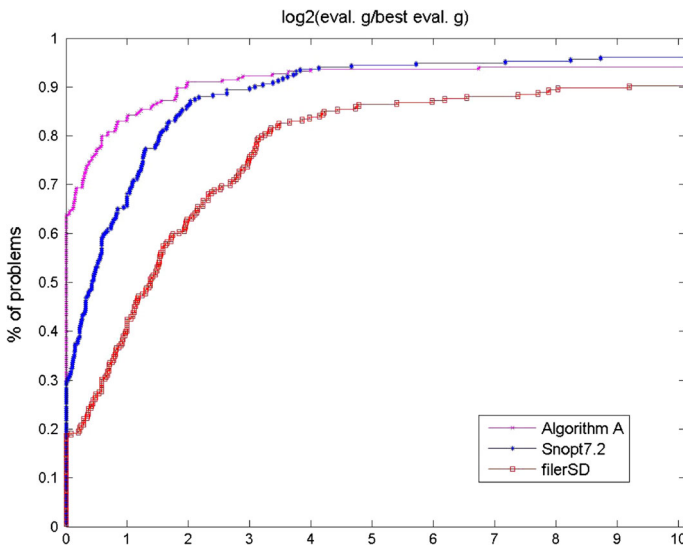


Fig. 3 Performance profile on \mathcal{P} for NG

$$y = \frac{1}{235} \text{size} \left\{ \mathbf{p} \in \mathcal{P} : \log_2 \left(\frac{\# \text{iter}(s, \mathbf{p})}{\text{best iter}(\mathbf{p})} \right) \leq x \right\}.$$

According to this definition, what we are primarily interested in is the behavior for x close to zero, and we observed that

- In terms of NIT, Algorithm A, filterSD and SNOPT perform the best among roughly 40, 45 and 26 % of \mathcal{P} , respectively (We remark that the total of the three percentiles is greater than one because more than one solver receive the same NIT as the best on some problems. Similar phenomenons occur for NF and NG in the following),
- In terms of NF, Algorithm A, filterSD and SNOPT perform the best among roughly 61, 21 and 31 % of \mathcal{P} , respectively, and
- In terms of NG, Algorithm A, filterSD and SNOPT perform the best among roughly 63, 19 and 30 % of \mathcal{P} , respectively.

It can be seen then that Algorithm A are of better performance than SNOPT in the set of test problems. We observed also that, in terms of NIT, filterSD performs a little bit better than Algorithm A, which is partially attributed to the fact that the inner subproblem inside the filterSD is a minimization of a nonlinear objective function subject to linear constraint, which possibly could approximate the original problem better than the quadratic approximation (e.g., Algorithm A). However, on the other hand, although filterSD generally requires less outer iterations, the computational costs in the inner subproblem used in filterSD might be more expensive, as more evaluations of the objective function, its gradients, constraints and their Jacobian matrices are needed. This partially is revealed by Figs. 2 and 3. Overall, our numerical experiment shows the robustness and efficiency of our approach.

6.2 Test for a degenerate problem

In order to verify the performance of our algorithm in the case that the RCPLD condition holds while the MFCQ condition does not, we construct a new test example by adding a constraint into the constraints of problem *hs074*.

hs074 (selected from [27])

$$\begin{aligned}
 \min \quad & f(x_1, x_2, x_3, x_4) = 3x_1 + 10^{-6}x_1^3 + 2x_2 + \frac{2}{3000000}x_2^3 \\
 \text{s.t.} \quad & c_1(x_1, x_2, x_3, x_4) = 1000 \sin(-x_3 - 0.25) + 1000 \sin(-x_4 - 0.25) \\
 & \quad \quad \quad + 894.8 - x_1 = 0; \\
 & c_2(x_1, x_2, x_3, x_4) = 1000 \sin(x_3 - 0.25) + 1000 \sin(x_3 - x_4 - 0.25) \\
 & \quad \quad \quad + 894.8 - x_2 = 0; \\
 & c_3(x_1, x_2, x_3, x_4) = 1000 \sin(x_4 - 0.25) + 1000 \sin(x_4 - x_3 - 0.25) \\
 & \quad \quad \quad + 1294.8 = 0; \\
 & c_4(x_1, x_2, x_3, x_4) = -x_4 + x_3 - 0.55 \leq 0; \\
 & c_5(x_1, x_2, x_3, x_4) = -x_3 + x_4 - 0.55 \leq 0; \\
 & c_6(x_1, x_2, x_3, x_4) = -x_1 \leq 0; \\
 & c_7(x_1, x_2, x_3, x_4) = x_1 - 1200 \leq 0; \\
 & c_8(x_1, x_2, x_3, x_4) = -x_2 \leq 0; \\
 & c_9(x_1, x_2, x_3, x_4) = x_2 - 1200 \leq 0; \\
 & c_{10}(x_1, x_2, x_3, x_4) = -0.55 - x_3 \leq 0;
 \end{aligned}$$

$$\begin{aligned}
 c_{11}(x_1, x_2, x_3, x_4) &= -0.55 + x_3 \leq 0; \\
 c_{12}(x_1, x_2, x_3, x_4) &= -0.55 - x_4 \leq 0; \\
 c_{13}(x_1, x_2, x_3, x_4) &= -0.55 + x_4 \leq 0.
 \end{aligned}$$

Modified *hs074*

$$\begin{aligned}
 \min \quad & f(x_1, x_2, x_3, x_4) \\
 \text{s.t.} \quad & c_0(x_1, x_2, x_3, x_4) = ac_2(x_1, x_2, x_3, x_4) + bc_3(x_1, x_2, x_3, x_4) = 0; \\
 & c_i(x_1, x_2, x_3, x_4) = 0, \quad i = 1, \dots, 3; \\
 & c_4(x_1, x_2, x_3, x_4) \leq 0, \quad i = 4, \dots, 13.
 \end{aligned} \tag{6.1}$$

We observed that the added constraint in modified *hs074* is a combination of the 2nd and 3rd constraints of *hs074*. So they have the same solutions. Let $a = 300, b = 1000$. By computation, we obtain that the active index set of *hs074* is $\{1, 2, 3\}$ and the active index set of modified *hs074* is $\{0, 1, 2, 3\}$. By the construction of modified *hs074*, we easily know that it has unbounded multipliers because the first active constraint is a combination of the 2nd and 3rd active constraints of *hs074*. By direct calculation, we have

$$\begin{pmatrix} \nabla c_0(x^*)^T \\ \nabla c_1(x^*)^T \\ \nabla c_2(x^*)^T \\ \nabla c_3(x^*)^T \end{pmatrix} = \begin{pmatrix} -1.0000e + 003 & -3.0000e + 002 & -3.4579e + 005 & -1.2788e + 006 \\ 0 & 0 & -7.2131e + 002 & 1.5197e + 003 \\ 0 & -1.0000e + 000 & 1.9565e + 003 & -9.6506e + 002 \\ -1.0000e + 000 & 0 & -9.3273e + 002 & -9.8933e + 002 \end{pmatrix}$$

and its row rank is 3, where

$$x^* = (6.7995e + 002, 1.0261e + 003, 1.1888e - 001, -3.9623e - 001)^T$$

is the optimal point computed by our algorithm. Obviously, $\nabla c_0(x^*) = a\nabla c_2(x^*) + b\nabla c_3(x^*)$. Because of this combination, the row rank deficiency of above matrix is 1. Therefore, the RCPLD condition holds at x^* while the MFCQ condition is violated. Calling Algorithm A, we find the optimal point $(6.7995e + 002, 1.0261e + 003, 1.1888e - 001, -3.9623e - 001)^T$ of *hs074* which is the same as x^* . Hence, for *hs074*, the gradients of active constraints at x^* are linearly independent, i.e., the row rank of

$$\begin{pmatrix} \nabla c_1(x^*)^T \\ \nabla c_2(x^*)^T \\ \nabla c_3(x^*)^T \end{pmatrix}$$

is 3. So, for *hs074*, the multiplier is unique. From Table 1, one can see that there is no difference between *hs074* and modified *hs074* with respect to NIT, NF, NG and Fopt. For *hs074*, the Lagrange multipliers corresponding to the active set at the

Table 1 Numerical results

Problem	NIT	NF	NG	Fopt
<i>hs074</i>	12	13	13	5.1265e+003
Modified <i>hs074</i>	12	13	13	5.1265e+003

Table 2 Information on *hs074*

<i>k</i>	KKT error	Infeasibility error	Complementarity error
1	1.3642e−12	8.2566e+00	8.2566e+00
2	1.3642e−12	5.9930e+00	5.9930e+00
3	2.2737e−12	4.7086e+00	4.7086e+00
4	5.2427e+02	2.2387e+00	2.2387e+00
5	1.8520e+01	9.9339e−02	9.9339e−02
6	7.6770e+01	2.6273e−04	2.6273e−04
7	6.7290e+01	1.6795e−05	1.6795e−05
8	1.9627e+02	4.1125e−04	4.1125e−04
9	1.3907e+02	9.1835e−03	9.1835e−03
10	1.6849e+00	4.8833e−03	4.8833e−03
11	1.7736e−02	3.7848e−07	3.7848e−07
12	3.2973e−04	1.2730e−10	1.2730e−10
13	1.0687e−10	3.2854e−14	3.2854e−14

Table 3 Information on modified *hs074*

<i>k</i>	KKT error	Infeasibility error	Complementarity error
1	1.3642e−12	1.1171e+01	1.1171e+01
2	2.2737e−12	7.4710e+00	7.4710e+00
3	2.7285e−12	5.8364e+00	5.8364e+00
4	4.3240e+02	2.7722e+00	2.7722e+00
5	6.4729e+00	6.2214e−02	6.2214e−02
6	3.1479e+01	8.5421e−05	8.5421e−05
7	8.4390e+01	4.1526e−05	4.1526e−05
8	2.9027e+02	9.5573e−04	9.5573e−04
9	3.5025e+01	1.8405e−02	1.8405e−02
10	7.5391e−02	3.1893e−04	3.1893e−04
11	8.8307e−04	8.8294e−10	8.8294e−10
12	1.4623e−05	2.7222e−13	2.7222e−13
13	1.8190e−12	2.8271e−15	2.8271e−15

optimal point are

$$(\lambda_1, \lambda_2, \lambda_3) = (-2.1253e + 02 - 3.9780e + 025.2934e + 02),$$

and for the modified **hs074**, the Lagrange multipliers are

$$(\lambda_0, \lambda_1, \lambda_2, \lambda_3) = (4.4994e + 02 - 2.1253e + 02 - 2.3900e + 020.0000e + 00).$$

We noticed that the norm of the multipliers for modified **hs074** is clearly larger than that for **hs074**. However, our algorithm still works well on the modified **hs074**, which is mainly attributed to the property that the implied multipliers (see $\tilde{\lambda}_k$ in (4.15)) for modified **hs074** is bounded. To reveal the iteration process in solving **hs074** and modified **hs074** much clearer, we present the following two tables (Tables 2, 3), from which one can see that iterations generated by Algorithm A for both cases converge quickly to some local minimizer. All these show that our algorithm is efficient in solving some degenerate problems for which the MFCQ condition fails to hold at the local solutions.

7 Conclusion

In this paper, we proposed a robust SQP algorithm for general constrained optimization problems. Our method mainly employs a modified QP subproblem to generate search directions and adopts a specific nonmonotone line search technique to detect appropriate step-size. It should be emphasized that our algorithm does not introduce any penalty function or filter. Compared with the filter-based SQP methods, our algorithm does not involve a feasibility restoration phase which might be computationally expensive. Another appealing feature of our algorithm is that it can deal with some degenerate optimization problem, for which the Mangasarian–Fromovitz constraint qualification (MFCQ) or strict complementarity (SC) fails to hold at a solution. This is demonstrated preliminarily in our numerical testing. Under the RCPLD, which is weaker than the CPLD and the MFCQ, the global convergence is proved, and the local fast convergence is also established under some weaker conditions (the SMFCQ and the SOSQ). Preliminary numerical experiments show the effectivity of our algorithm.

Acknowledgments We are indebted to the editor and two anonymous referees for their many valuable comments and suggestions that have improved the quality of this paper significantly. This research is supported by National Natural Science Foundation of China (Nos. 11101281, 11101257 and 11271259) and Innovation Program of Shanghai Municipal Education Commission (No. 12YZ172).

References

1. Anitescu, M.: Degenerate nonlinear programming with a quadratic growth condition. *SIAM J. Optim.* **10**(4), 1116–1135 (2000)
2. Andreani, R., Haeser, G., Schuverdt, M., Silva, P.: A relaxed constant positive linear dependence constraint qualification and applications. *Math. Program.* (2011). doi:[10.1007/s10107-011-0456-0](https://doi.org/10.1007/s10107-011-0456-0)
3. Andreani, R., Martinez, J.M., Schuverdt, M.L.: On the relation between constant positive linear dependence condition and quasinormality constraint qualification. *J. Optim. Theory Appl.* **125**, 473–485 (2005)
4. Bielschowsky, R.H., Gomes, F.A.M.: Dynamical control of infeasibility in nonlinearly constrained optimization. *SIAM J. Optim.* **19**(3), 1299–1325 (2008)
5. Bongartz, I., Conn, A.R., Gould, N.I.M., Toint, PhL: CUTE: constrained and unconstrained testing environment. *ACM Trans. Math. Softw.* **21**, 123–160 (1995)

6. Bonnans, J.F., Gilbert, JCh., Lemaréchal, C., Sagastizábal, C.: *Numerical Optimization: Theoretical and Practical Aspects*, 2nd edn. Springer, Berlin (2006)
7. Boggs, P.T., Tolle, J.W., Wang, P.: On the local convergence of quasi-newton methods for constrained optimization. *SIAM J. Control Optim.* **20**, 161–171 (1982)
8. Bonnans, J.F., Ioffe, A.: Second-order sufficiency and quadratic growth for nonisolated minima. *Math. Oper. Res.* **20**(4), 801–817 (1995)
9. Burke, J.V., Han, S.P.: A robust sequential quadratic programming method. *Math. Program.* **43**, 277–303 (1989)
10. Byrd, R., Lopez-Calva, G., Nocedal, J.: A line search exact penalty method using steering rules. *Math. Program.* **133**(1–2), 39–73 (2012)
11. Chamberlain, R.M., Powell, M.J.D., Lemarechal, C., Pedersen, H.C.: The watchdog technique for forcing convergence in algorithms for constrained optimization. *Math. Program. Study* **16**, 1–17 (1982)
12. Dolan, E.D., Moré, J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**, 201–213 (2002)
13. Fernández, D., Izmailov, A.F., Solodov, M.V.: Sharp primal superlinear convergence results for some Newtonian methods for constrained optimization. *SIAM J. Optim.* **20**(6), 3312–3334 (2010)
14. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. *Math. Program.* **91**, 239–269 (2002)
15. Fletcher, R., Leyffer, S., Toint, PhL: On the global convergence of a filter-SQP algorithm. *SIAM J. Optim.* **13**(1), 44–59 (2002)
16. Fletcher, R.: A sequential linear constraints programming algorithm for NLP. *SIAM J. Optim.* **22**(3), 772–794 (2012)
17. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM J. Optim.* **12**(4), 979–1006 (2002)
18. Gould, N., Toint, PhL: Nonlinear programming without a penalty function or a filter. *Math. Program.* **122**(1), 155–196 (2010)
19. Gould, N., Robinson, D.P., Toint, Ph.L.: Corrigendum: Nonlinear Programming Without a Penalty Function or a Filter. NAXYS Technical Report naxys-07-2011, Namur Center for Complex Systems (NAXYS) (2011)
20. Gould, N., Robinson, D.P.: A second derivative SQP method: global convergence. *SIAM J. Optim.* **20**(4), 2023–2048 (2010)
21. Gould, N., Robinson, D.P.: A second derivative SQP method: local convergence. *SIAM J. Optim.* **20**(4), 2049–2079 (2010)
22. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton’s method. *SIAM J. Numer. Anal.* **23**, 707–716 (1986)
23. Hager, W.W.: Stabilized sequential quadratic programming. *Comput. Optim. Appl.* **12**, 253–273 (1999)
24. Han, S.P.: Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Math. Program.* **11**, 263–282 (1976)
25. Han, S.P.: A globally convergent method for nonlinear programming. *J. Optim. Theory Appl.* **22**, 297–309 (1977)
26. Hestenes, M.R.: *Optimization Theory: The Finite-Dimensional Case*. Wiley, New York, NY (1975)
27. Hock, W., Schittkowski, K.: *Test Examples for Nonlinear Programming Codes*. Springer, Berlin (1981)
28. Kyparisis, J.: On uniqueness of Kuhn–Tucker multipliers in nonlinear programming. *Math. Program.* **32**(2), 242–246 (1984)
29. Liu, X.W., Yuan, Y.X.: A sequential quadratic programming method without a penalty function or a filter for nonlinear equality constrained optimization. *SIAM J. Optim.* **21**(2), 545–571 (2011)
30. Morales, J.L., Nocedal, J., Wu, Y.: A sequential quadratic programming algorithm with an additional equality constrained phase. *IMA J. Numer. Anal.* (2011). doi:[10.1093/imanum/drq037](https://doi.org/10.1093/imanum/drq037)
31. Nocedal, J., Wright, S.: *Numerical Optimization*, 2nd edn. Springer, New York (2006)
32. Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: *Waston, G.A. (ed.) Numerical Analysis. Proceedings, Biennial Conference, Dundee 1977. Lecture Notes in Mathematics 630*, pp. 144–157. Springer, Berlin (1978)
33. Powell, M.J.D.: Variable metric methods for constrained optimization. In: *Bachem, A., Grotscchel, M., Korte, B. (eds.) Mathematical Programming: The State of Art, Bonn, 1982*. Springer, Berlin (1983)
34. Qi, L., Wei, Z.: On the constant positive linear dependence condition and its application to SQP methods. *SIAM J. Optim.* **10**, 963–981 (2000)

35. Shen, C.G., Leyffer, S., Fletcher, R.: A nonmonotone filter method for nonlinear optimization. *Comput. Optim. Appl.* **52**, 583–607 (2012)
36. Ulbrich, M., Ulbrich, S.: Nonmonotone trust-region methods for nonlinear equality constrained optimization without a penalty function. *Math. Program.* **95**, 103–135 (2003)
37. Wright, S.J.: Superlinear convergence of a stabilized SQP method to a degenerate solution. *Comput. Optim. Appl.* **11**(3), 253–275 (1998)
38. Wright, S.J.: Modifying SQP for degenerate problems. *SIAM J. Optim.* **13**(2), 470–497 (2002)
39. Wright, S.J.: Constraint identification and algorithm stabilization for degenerate nonlinear programs. *Math. Program.* **95**(1), 137–160 (2003)
40. Wright, S.J.: An algorithm for degenerate nonlinear programming with rapid local convergence. *SIAM J. Optim.* **15**(3), 673–696 (2005)
41. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: motivation and global convergence. *SIAM J. Optim.* **16**, 1–31 (2005)
42. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: local convergence. *SIAM J. Optim.* **16**, 32–48 (2005)
43. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **106**, 25–57 (2006)
44. Xue, W.J., Shen, C.G., Pu, D.G.: A penalty-function-free line search SQP method for nonlinear programming. *J. Comput. Appl. Math.* **228**(1), 313–325 (2009)
45. Zhang, J.L., Zhang, X.S.: A modified SQP method with nonmonotone line search technique. *J. Glob. Optim.* **21**, 201–218 (2001)
46. Zhou, G.L.: A modified SQP method and its global convergence. *J. Glob. Optim.* **11**, 193–205 (1997)