A sequential quadratically constrained quadratic programming method with an augmented Lagrangian line search function

Chun-Ming Tang\textsuperscript{a,b}, Jin-Bao Jian\textsuperscript{b,*}

\textsuperscript{a}Department of Mathematics, Shanghai University, Shanghai, PR China
\textsuperscript{b}College of Mathematics and Information Science, Guangxi University, Nanning, PR China

Received 7 March 2007; received in revised form 4 September 2007

Abstract

Based on an augmented Lagrangian line search function, a sequential quadratically constrained quadratic programming method is proposed for solving nonlinearly constrained optimization problems. Compared to quadratic programming solved in the traditional SQP methods, a convex quadratically constrained quadratic programming is solved here to obtain a search direction, and the Maratos effect does not occur without any other corrections. The “active set” strategy used in this subproblem can avoid recalculating the unnecessary gradients and (approximate) Hessian matrices of the constraints. Under certain assumptions, the proposed method is proved to be globally, superlinearly, and quadratically convergent. As an extension, general problems with inequality and equality constraints as well as nonmonotone line search are also considered.

\textcopyright 2007 Published by Elsevier B.V.

\textbf{MSC:} 90C30; 49M30

\textbf{Keywords:} SQCQP; Quadratically constrained quadratic programming; Augmented Lagrangian line search function; Nonlinear programming; Convergence

1. Introduction

For simplicity, we first consider the following inequality constrained nonlinear programming problem. (General problem with inequality and equality constraints is treated as an extension in Section 5.)

\[
\min_{x \in \mathbb{R}^n} f(x) \\
\text{s.t. } g_j(x) \leq 0, \quad j \in I \overset{\text{def}}{=} \{1, \ldots, m\},
\]

where \( f, g_j \ (j \in I) : \mathbb{R}^n \rightarrow \mathbb{R} \) are continuously differentiable (possibly highly nonlinear) functions.

During the last several decades (especially in recent five years), the sequential quadratically constrained quadratic programming (SQCQP) methods were considered for solving (highly) nonlinear programming by many authors,
An important advantage of SQCQP methods is that the Maratos effect \[18\] does not occur without the computation of a correctional direction. Roughly speaking, the SQCQP methods solve at each iteration a quadratically constrained quadratic programming (QCQP) subproblem whose objective and constraints are the quadratic approximations of the objective and constraints of the original problem, respectively. Although the QCQP subproblems are more computationally difficult than quadratic programming (QP) subproblems solved in the sequential quadratic programming (SQP) methods, we expect that fewer subproblems will be required to be solved when compared to the traditional SQP methods, since QCQPs are a higher-order (thus better) approximation of the original problem \(1.1\) than QPs. The preliminary numerical results given in \[1,13\] show that the SQCQP methods perform indeed better than SQP methods in this aspect. On the other hand, there are many efficient tools available for solving the QCQP subproblems, see e.g., \[16,17,2\].

Let \(x^k\) be the current iteration point. In the case where problem \(1.1\) is a convex programming, Fukushima et al. \[8\] proposed an SQCQP algorithm by considering the following QCQP subproblem:

\[
\begin{align*}
\min & \quad \nabla f(x^k)^T d + \frac{1}{2} d^T G^k d \\
\text{s.t.} & \quad g_j(x^k) + \nabla g_j(x^k)^T d + \frac{1}{2} x_j^k d^T \nabla^2 g_j(x^k) d \leq 0, \quad j \in I,
\end{align*}
\]

(1.2)

where \(G^k\) is a positive definite matrix. The parameters \(x_j^k \in [0, 1], \ j \in I\) are introduced to ensure the feasibility of the constraints. Subproblem \(1.2\) is a convex programming and thus can be formulated as a second-order cone program and be solved efficiently by using the interior points methods, see e.g., \[17\]. By using an \(\ell_1\) nondifferentiable exact penalty function

\[
P_r(x) = f(x) + r \sum_{j=1}^m \max\{0, g_j(x)\},
\]

Fukushima et al. \[8\] prove that, under additional Slater constraint qualification and a strong second-order sufficient condition, the algorithm is globally and quadratically convergent.

Solodov \[25\] considered the subproblems of the following structure by introducing an additional variable \(t\):

\[
\begin{align*}
\min & \quad \nabla f(x^k)^T d + \frac{1}{2} d^T G^k d + r_k t \\
\text{s.t.} & \quad g_j(x^k) + \nabla g_j(x^k)^T d + \frac{1}{2} d^T G^k_j d \leq t, \quad j \in I_k, \quad t \geq 0,
\end{align*}
\]

(1.3)

where \(r_k > 0\) is a penalty parameter, and \(G^k_j, \ j \in I_k\) are positive semidefinite (possibly different from \(\nabla^2 g_j(x^k)\)). An \(\ell_\infty\) nondifferentiable exact penalty function is employed as follows:

\[
P_r(x) = f(x) + r \max\{0, g_j(x), \ldots, g_m(x)\}.
\]

(1.4)

Compared to \[8\], the global convergence of the corresponding SQCQP algorithm does not assume the convexity of the objective and twice differentiability of the objective and constraints.

Differed from the penalty type SQCQP methods above, there is another class of SQCQP methods that uses directly the objective function as a merit function in the line search, see \[12,13\]. Particularly, Jian \[12\] proposed a feasible SQCQP method by considering a norm-relaxed type QCQP subproblem. Compared to \[8,25\], in global or local convergence analysis, the algorithm in \[12\] does not require the convexity assumption of the objective function or the constraints. In addition, the algorithm is able to generate a sequence of feasible iterates. Under weaker conditions, the global, superlinear, and quasi-quadratic convergence are obtained. Recently, Jian et al. \[13\] have relaxed the choice of the starting points for \[12\], i.e., the algorithm allows an infeasible starting point.

For local SQCQP methods, we refer the reader to Anitescu \[1\] and Fernández and Solodov \[7\]. In \[1\], a trust-region QCQP subproblem is considered, and local superlinear rate of convergence is proved under the Mangasarian–Fromovitz constraint qualification plus a quadratic growth condition. The authors in \[7\] considered the class of QCQP methods in the framework extended from optimization to more general variational problems, and proved the primal–dual quadratic convergence under the linear independent constraint qualification, strict complementarity and a second-order sufficient condition, and presented a necessary and sufficient condition for superlinear convergence of the primal sequence under a Dennis–Moré-type condition.
In this paper, we present a new sequential QCQP method that employs an augmented Lagrangian line search function. Our motivation is mainly based on the following considerations:

1. The line search functions used in [8,25] are not differentiable. This may lead to certain numerical difficulties and the line search techniques based on smooth polynomial interpolation are not applicable.
2. The augmented Lagrangian line search function is differentiable, and it does perform well in both theoretical analysis and computational implementation, see e.g., [22,23,10,6,3]. Furthermore, some excellent software is based on the augmented Lagrangian line search function, such as NPSOL [9,10] and NLPQLP [24].
3. The above-mentioned global SQCQP methods are extended hardly to general problem with equality and inequality constraints, i.e., they are only able to deal with inequality constrained problems.
4. A globally convergent method is very important in practice, since the user usually cannot find a suitable guess of the optimal solution.

In the next section, we present the QCQP subproblem and the augmented Lagrangian line search function used in this paper, respectively, and then give our SQCQP algorithm. In Section 3, we prove that the proposed algorithm is globally convergent. Superlinear and quadratic convergence are discussed in Section 4. We extend the algorithm to inequality and equality constrained problem and to nonmonotone line search function in Section 5.

2. The algorithm

We begin this section by making the following basic assumption, which is assumed to be satisfied throughout this paper without restatement.

**Assumption 1.** The functions \(f, g_j (j \in I)\) are all first-order continuously differentiable.

In order to save the unnecessary calculation of the gradients and the approximate Hessian matrices of constraints, motivated by the idea of Schittkowski [23] for SQP method, we could consider the following QCQP subproblem at the \(k\)th iteration:

\[
\begin{align*}
\min_{d \in \mathbb{R}^n} & \quad \nabla f(x_k)^T d + \frac{1}{2} d^T G^k d \\
\text{s.t.} & \quad g_j(x_k) + \nabla g_j(x_k)^T d + \frac{1}{2} d^T G_j^k d \leq 0, \quad j \in J_k^*, \\
& \quad g_j(x_k) + \nabla g_j(x_k^{(j)})^T d + \frac{1}{2} d^T G_j^{k(j)} d \leq 0, \quad j \in K_k^*,
\end{align*}
\]

(2.1)

where \(G^k\), the approximation of the Hessian of the objective function, is symmetric positive definite and \(G_j^k\) or \(G_j^{k(j)}\) \((j \in I)\), the approximations of the Hessian of the constraints, are symmetric positive semidefinite. The indices \(k(j) \leq k\) correspond to previous iterates and their definition will be clear when investigating the algorithm. \(J_k^*\) is the set of “active” constraints at iteration \(k\), and \(K_k^*\) is the set of “inactive” constraints at iteration \(k\). A suitable choice of these two sets from [23] is as follows:

\[
J_k^* = \{ j \in I : g_j(x_k) \geq -\varepsilon \text{ or } v_j^k > 0 \}, \quad K_k^* = I \setminus J_k^*,
\]

(2.2)

where \(v^k = (v_1^k, \ldots, v_m^k)^T\) is the Lagrange multiplier estimate, and \(\varepsilon\) is a positive constant.

However, the feasible region of subproblem (2.1) may be empty if \(x_k\) is not a feasible point of the original problem (1.1). Similar to Powell [20] (see also [23,6]) for QP subproblem, we introduce an additional variable \(\delta\) to (2.1), and therefore the extensive QCQP subproblem will be defined by

\[
\begin{align*}
\min_{d \in \mathbb{R}^n, \delta \in [0,1]} & \quad \nabla f(x_k)^T d + \frac{1}{2} d^T G^k d + \frac{1}{2} \delta^2 \\
\text{s.t.} & \quad (1 - \delta)g_j(x_k) + \nabla g_j(x_k)^T d + \frac{1}{2} d^T G_j^k d \leq 0, \quad j \in J_k^*, \\
& \quad g_j(x_k) + \nabla g_j(x_k^{(j)})^T d + \frac{1}{2} d^T G_j^{k(j)} d \leq 0, \quad j \in K_k^*,
\end{align*}
\]

(2.3)
where $q_k$ is a penalty parameter, which is used to reduce the perturbation of search direction by the additional variable $\delta$ as much as possible. Similar to [23], we can choose (Detailed motivation of this rule can also be found in [23].)

$$q_k = \max \left\{ q_0, \frac{q^* ((d^{k-1})^T A_{k-1} u^{k-1})^2}{(1 - \delta_{k-1})^2 (d^{k-1})^T G^{k-1} d^{k-1}} \right\},$$  

(2.4)

where $q_0, q^* > 0$. Furthermore, from the definition of $K^*_k$, it is easy to see that $(d, \delta_k)$ is a feasible solution for (2.3). We call $(d_k, \delta_k)$ a Karush–Kuhn–Tucker (KKT) point for QCQP (2.3) if there exist $u_k \in \mathbb{R}^m, v_1, v^*_2 \in \mathbb{R}$ such that

$$\nabla f(x^k) + G^k d^k + \sum_{j \in J^*_k} u^k_j (\nabla g_j(x^k) + G^k d^k) + \sum_{j \in K^*_k} u^k_j (\nabla g_j(x^{k(j)}) + G^k d^k) = 0,$$

(2.5a)

$$q_k \delta_k - \sum_{j \in J^*_k} u^k_j g_j(x^k) - v_1^k + v^*_2 = 0,$$

(2.5b)

$$w^k_j \leq 0, \quad j \in I,$$

(2.5c)

$$0 \leq \delta_k \leq 1,$$

(2.5d)

$$u^k_j \geq 0, \quad j \in I,$$

(2.5e)

$$v^*_1 \geq 0,$$

(2.5f)

$$v^*_2 \geq 0,$$

(2.5g)

$$w^k_j u^k_j = 0, \quad j \in I,$$

(2.5h)

$$v^k_1 \delta_k = 0,$$

(2.5i)

$$v^k_2 (1 - \delta_k) = 0,$$

(2.5j)

where

$$w^k_j = \left\{ \begin{array}{ll} (1 - \delta_k) g_j(x^k) + \nabla g_j(x^k)^T d^k + \frac{1}{2}(d^k)^T G^k d^k, & j \in J^*_k, \\
 g_j(x^k) + \nabla g_j(x^{k(j)})^T d^k + \frac{1}{2}(d^k)^T G^k d^k, & j \in K^*_k. \end{array} \right.$$  

(2.6)

In this paper, we use the following differentiable augmented Lagrange function whose original formulation is due to Rockafellar [21], and used as a merit function in SQP methods by [22,23,6],

$$\Phi_r(x, v) = f(x) + \sum_{j \in J} \left( v_j g_j(x) + \frac{1}{2} r_j g_j^2(x) \right) - \frac{1}{2} \sum_{j \in K} v^2_j / r_j,$$

(2.7)

where the index sets $J$ and $K$ are defined by

$$J = \{ j \in I : g_j(x) \geq v_j / r_j \}, \quad K = I \setminus J.$$  

(2.8)

The penalty parameter $r \in \mathbb{R}^m$ in (2.7) must be selected carefully such that $d$ is a sufficient descent direction of the augmented Lagrange function $\Phi_r(x, v)$. Closely related to [23], we define

$$r^k_{j+1} = \max \left\{ r^k_{j+1} \frac{2m(u^k_j - v^k_j)^2}{(1 - \delta_k)(d^k)^T Q^k d^k} \right\},$$  

(2.9)
where $Q^k$ is defined by

$$Q^k = G^k + \sum_{j \in J_k^*} u^k_j G^k_j + \sum_{j \in K_k^*} u^k_j G^{k(j)}_j,$$

(2.10)

and (as pointed out in [23,6]) the sequence $\{\sigma^k_j\}$ is introduced to allow decreasing penalty parameters at least in the beginning of the algorithm by assuming that $\sigma^k_j \leq 1$, and it should guarantee the convergence of $\{r^k_j\}$ whenever this sequence is bounded. A sufficient condition to guarantee convergence of $\{r^k_j\}$ is that there exists a positive constant $h$ with

$$\sum_{k=0}^{\infty} (1 - (\sigma^k_j)^h) < \infty,$$

(2.11)

for all $j \in I$. A possible practical choice of $\sigma^k_j$ is

$$\sigma^k_j = \min \left\{ 1, \frac{k}{\sqrt{r^k_j}} \right\}.$$

(2.12)

We note that the penalty parameter $r$ is independent from $g$—the penalty parameter used in the subproblem (2.3), and the components of $r$ are chosen individually for each constraint. These are very different from [25] and can reasonably improve the robustness of the algorithm. In [25], the QCQP subproblem (1.3) and the penalty function (1.4) use a common penalty parameter, which might lead to an ill-conditioned subproblem when the penalty parameter is too large.

Before giving the algorithm, let us define

$$\psi_k(\lambda) = \Phi_{r_{k+1}}(x^k + \lambda d^k, v^k + \lambda(u^k - v^k)),$$

(2.13)

where

$$J_k = \{ j : j \in I, g_j(x^k) \geq -v^k_j / r^k_j + 1 \}, \quad K_k = I \setminus J_k.$$

(2.14)

It follows from (2.2) that

$$J_k^* \supseteq J_k, \quad K_k^* \subseteq K_k.$$

(2.15)

The derivative of $\psi_k(\lambda)$ at $\lambda = 0$ can be written as

$$\Phi'_k(0) = \nabla \Phi_{r_{k+1}}(x^k, v^k)^T \left( \frac{d^k}{u^k - v^k} \right).$$

(2.16)

Now, keeping in mind the following facts, we will present the details of our algorithm.

(i) Numerical tests in [23,6] show that the penalty parameter $g$ and the additional variable $\delta$ could influence the numerical performance, so it is recommended to solve first subproblem (2.1) if its constraints are consistent.

(ii) The additional variable $\delta$ may lead to difficulty in proving the superlinear convergence of the algorithm.

(iii) QCQP subproblem (2.1) always has a feasible solution in the neighborhood of a solution (of the original problem) that satisfies a certain constraint qualification, see [1, Theorem 3.1].
Algorithm A.

Step 0. Initialization.

Parameters: \( \varepsilon > 0, \alpha \in (0, 0.5), \beta \in (0, 1), \tilde{\delta} \in (0, 1), \tilde{\theta} > 1, \) \( q^* > 0. \)

Data: Set \( k = 0, \) choose some starting values \( x_0, v_0 \) (with nonnegative components), \( q_0, r_0, \) and evaluate \( f(x_0), G_0, g_j(x_0), G^j_0, j \in I. \) Determine \( J^*_k, K^*_k \) and let \( k(j) = 0 \) for all \( j \in I. \)

Step 1. If subproblem (2.1) is consistent, then solve (2.1) to obtain a KKT pair \( (d^k, u^k) \), determine a new penalty parameter \( r^{k+1} \) by (2.9) with \( \tilde{\delta}_k = 0, \) and go to Step 6. Otherwise, go to Step 2.

Step 2. Solve subproblem (2.3) to get a KKT pair \( (d^k, u^k, \delta^k) \). If \( \delta^k > \tilde{\delta} \), let \( q_k := \tilde{\theta} q_k \) and solve (2.3) again.

Step 3. Determine a new penalty parameter \( r^{k+1} \) by (2.9).

Step 4. If \( \Psi_k'(0) > 0 \), then let \( q_k := \tilde{\theta} q_k \) and go to Step 2.

Step 5. Define the new penalty parameter \( \bar{q}_{k+1} \) by (2.4).

Step 6. Perform a line search. Compute the step size \( \lambda_k \), the first value of \( \lambda \) in the sequence \( \{1, \beta, \beta^2, \ldots\} \) that satisfies

\[
\Psi_k'(\lambda_k) \leq \Psi_k(0) + \alpha \lambda_k \Psi_k'(0). \quad (2.17)
\]

Step 7. Let \( x^{k+1} = x^k + \lambda_k d^k, v^{k+1} = v^k + \phi_k (u^k - v^k). \)

Step 8. Compute \( f(x^{k+1}), G^{k+1}, g_j(x^{k+1}), j \in I \) and determine sets \( J^*_k+1, K^*_k+1. \)

Evaluate \( \nabla g_j(x^{k+1}), G^j_k, j \in J^*_k, k+1(j) \) by

\[
k + 1(j) = \begin{cases} k & \text{if } j \in J^*_k, \\
k(j) & \text{if } j \in K^*_k, \end{cases}
\]

determine \( \nabla g_j(x^{k+1}(j)), G^{k+1(j)}, j \in K^*_k, k+1 \), set \( k := k + 1, \) and go to step 1.

Remark 1. (1) For completeness, a stopping criterion should be added to the algorithm. A suitable rule is that the algorithm stops if the KKT optimality conditions are satisfied subject to a tolerance. Furthermore, \( r^{k+1} \) at Step 3 is well-defined, since we can conclude that \( x^k \) is a KKT point whenever \( d^k = 0. \)

(2) We note that detecting the feasibility of subproblem (2.1) will not increase the number of function, gradient and matrix evaluations, and this procedure can be done by a QCQP solver itself, such as [16]. As mentioned above, solving first subproblem (2.1) will not only improve the numerical performance, but also bring advantage to analyze superlinear convergence.

(3) Steps 2–5 are performed only in a few early iterations, since subproblem (2.1) is always consistent near a solution. This will greatly simplify the algorithm.

(4) From (2.5b), we see that Step 2 can be finished after finite times if \( \Omega_k \) is a compact set, where

\[
\Omega_k = \{ u^k : (u^k, d^k) \text{ satisfies } (2.5a) \}.
\]

Furthermore, if \( \tilde{\delta} \) is chosen sufficiently close to 1, say \( \tilde{\delta} = 0.9, \) we expect that this loop may rarely fail. Even though it fails within a given upper bound for \( q_k, \) Schittkowski [23] suggested to use a modified search direction with the intention of minimizing the augmented Lagrange function and meanwhile maintaining global convergence (see [23, Eq. (18)]).

(5) The loop between Steps 2 and 4 is finite, and this is essential to guarantee that Algorithm A is well-defined. In fact, like [23], a lower bound for the choice of \( q_k \) can be given in the following Theorem 2.1. More precisely, the lower bound does not depend on \( d^k, u^k, \) or \( q_0, \) so if \( q_k \) increases such that it is greater than this lower bound, we would have that \( \Psi_k'(0) < 0 \) (see (2.19)), and therefore the loop between Steps 2 and 4 terminates.

For notational convenience, we define

\[
\bar{v}^k = (\bar{v}_1^k, \ldots, \bar{v}_m^k)^T, \quad \bar{v}_j^k = \begin{cases} v_j^k & \text{if } j \in J_k, \\
0 & \text{otherwise}, \end{cases}
\]

\[
\bar{w}^k = (\bar{w}_1^k, \ldots, \bar{w}_m^k)^T, \quad \bar{w}_j^k = \begin{cases} w_j^k & \text{if } j \in J_k, \\
0 & \text{otherwise}, \end{cases}
\]
\[ \bar{q}_k = (\bar{q}_1^k, \ldots, \bar{q}_m^k)^T, \quad \bar{q}_j^k = \begin{cases} (1 - \delta_k)g_j(x^k) + \nabla g_j(x^k)^T d^k & \text{if } j \in J_k, \\ 0 & \text{otherwise}, \end{cases} \]

\[ g_k = (g_1(x^k), \ldots, g_m(x^k))^T, \]

\[ \bar{g}_k = (\bar{g}_1(x^k), \ldots, \bar{g}_m(x^k))^T, \quad \bar{g}_j(x^k) = \begin{cases} g_j(x^k) & \text{if } j \in J_k, \\ 0 & \text{otherwise}, \end{cases} \]

\[ g_k' = (g_1'(x^k), \ldots, g_m'(x^k))^T, \quad g_j'(x^k) = \begin{cases} g_j(x^k) & \text{if } j \in J_k, \\ -v_j/r_{j+1}^k & \text{otherwise}, \end{cases} \]

\[ A_k = (\nabla g_1(x^k), \ldots, \nabla g_m(x^k)), \quad R_{k+1} = \text{diag}(r_1^{k+1}, \ldots, r_m^{k+1}). \] (2.18)

Now we prove that the search direction \((d^k, u^k - v^k)\) generated by the algorithm is a descent direction for the augmented Lagrange function \(\Phi_r(x, v)\) at point \((x^k, v^k)\), and thus the line search is well-defined. Our analysis is patterned after that of [23], but differs in the details. We only consider the case in which the search direction is generated by Step 2, since the analysis is a simple version for the one generated by Step 1.

**Theorem 2.1.** Let \(x^k\) be the current iterate, and the corresponding iterates be \(v^k, G^k, G_j^k, G_j^{(k)} d^k, u^k, \bar{g}_k, r_k, J^*_k \) and \(K^*_k\). Suppose that

(i) there exists a constant \(\gamma \in (0, 1]\) such that \((d^k)^T G^k d^k \geq \gamma \|d^k\|^2,

(ii) \(\delta_k \leq \delta_k\), and

(iii) \(\bar{g}_k \geq \|A_k \bar{v}_k\|^2 / \gamma (1 - \delta)^2\).

Then

\[ \nabla \Phi_{r_k+1}(x^k, v^k)^T \begin{pmatrix} d^k \\ u^k - v^k \end{pmatrix} \leq -\frac{1}{4} \gamma \|d^k\|^2. \] (2.19)

**Proof.** From the notations in (2.18), we can express the gradient of the augmented Lagrange function at \((d^k, v^k)\) in the following form:

\[ \nabla \Phi_{r_k+1}(x^k, v^k) = \begin{pmatrix} \nabla f(x^k) + A_k(\bar{g}_k + R_{k+1} \bar{g}_k) \\ \bar{g}_k' \end{pmatrix}. \] (2.20)

For simplicity, we omit the indices \(k\) and \(k + 1\) in the following discussions, and define

\[ S \overset{\text{def}}{=} -\nabla \Phi_r(x, v)^T \begin{pmatrix} d \\ u - v \end{pmatrix}. \]

Thus, by (2.20), we have

\[ S = -\nabla f(x)^T d - d^T A(\bar{v} + R \bar{g}) - \bar{g}'^T (u - v). \]
This together with the KKT condition (2.5a) and (2.15) shows that

\[ S = d^T G d + \sum_{j \in J^*} u_j \left( \nabla g_j(x) d + \frac{1}{2} d^T G_j d \right) + \sum_{j \in K^*} u_j \left( \nabla g_j(x^{(j)}) d + \frac{1}{2} d^T G_j^{(j)} d \right) \]

\[ + \frac{1}{2} \sum_{j \in J^*} u_j d^T G_j d + \frac{1}{2} \sum_{j \in K^*} u_j d^T G_j^{(j)} d - \sum_{j \in J} (v_j + r_j g_j(x)) \nabla g_j(x) d \]

\[ - \sum_{j \in J^*} g_j(x)(u_j - v_j) + \sum_{j \in K} \frac{1}{r_j} v_j (u_j - v_j) \]

\[ = d^T G d + \sum_{j \in J^*} (u_j - v_j) \left( \nabla g_j(x) d + \frac{1}{2} d^T G_j d \right) \]

\[ + \sum_{j \in K^*} u_j \left( \nabla g_j(x^{(j)}) d + \frac{1}{2} d^T G_j^{(j)} d \right) + \sum_{j \in J^* \setminus J} v_j \left( \nabla g_j(x) d + \frac{1}{2} d^T G_j d \right) \]

\[ + \frac{1}{2} \sum_{j \in J} v_j d^T G_j d - \sum_{j \in J} r_j g_j(x) \nabla g_j(x) d \]

\[ - \sum_{j \in J} g_j(x)(u_j - v_j) + \sum_{j \in K} \frac{1}{r_j} v_j (u_j - v_j) + A_1. \] (2.21)

By further using (2.6) and (2.5h), we have

\[ S = d^T G d + \sum_{j \in J^*} (u_j - v_j) w_j - (1 - \delta) \sum_{j \in J^*} (u_j - v_j) g_j(x) + \sum_{j \in K^*} u_j w_j \]

\[ - \sum_{j \in K^*} u_j g_j(x) + \sum_{j \in J^* \setminus J} v_j w_j - (1 - \delta) \sum_{j \in J^* \setminus J} v_j g_j(x) \]

\[ + \frac{1}{2} \sum_{j \in J} (v_j + r_j g_j(x)) d^T G_j d - \sum_{j \in J} r_j g_j(x) w_j + (1 - \delta) \sum_{j \in J} r_j g_j(x)^2 \]

\[ - \sum_{j \in J} (u_j - v_j) g_j(x) + \sum_{j \in K} \frac{1}{r_j} v_j (u_j - v_j) + A_1 \]

\[ = d^T G d - \sum_{j \in J} w_j (v_j + r_j g_j(x)) - \sum_{j \in J^* \setminus J} v_j w_j - 2 \sum_{j \in J} (u_j - v_j) g_j(x) \]

\[ - \sum_{j \in J^* \setminus J} (u_j - v_j) g_j(x) + \delta \sum_{j \in J^* \setminus J} (u_j - v_j) g_j(x) - \sum_{j \in K^*} u_j g_j(x) \]

\[ + \sum_{j \in J^* \setminus J} v_j w_j - (1 - \delta) \sum_{j \in J^* \setminus J} v_j g_j(x) + (1 - \delta) \sum_{j \in J} r_j g_j(x)^2 \]

\[ + \sum_{j \in K} \frac{1}{r_j} v_j (u_j - v_j) + A_1 + A_2. \] (2.22)
From (2.8), we know $v_j + r_j g_j(x) \geq 0$ for $j \in J$. This shows that $\Delta_2 \geq 0$, and thus from (2.22) and (2.5c) we have

$$S \geq d^T G d - 2 \sum_{j \in J} (u_j - v_j) g_j(x) + 2 \sum_{j \in J} \frac{1}{r_j} v_j (u_j - v_j) - \sum_{j \in J} \frac{1}{r_j} v_j (u_j - v_j)$$

$$- \sum_{j \in J \setminus J^*} u_j g_j(x) + \delta \sum_{j \in J^*} (u_j - v_j) g_j(x) - \sum_{j \in K} u_j g_j(x)$$

$$+ (1 - \delta) \sum_{j \in J} r_j g_j(x)^2 + \delta \sum_{j \in J \setminus J^*} v_j g_j(x) + \Delta_1$$

$$= d^T G d - 2 \bar{g}^T (u - v) - \sum_{j \in J} \frac{1}{r_j} v_j (u_j - v_j) - \sum_{j \in K \setminus K^*} \frac{1}{r_j} v_j (u_j - v_j)$$

$$- \sum_{j \in K \setminus K^*} u_j g_j(x) + \delta \sum_{j \in J \setminus J^*} u_j g_j(x) + \delta \sum_{j \in J} (u_j - v_j) g_j(x)$$

$$- \sum_{j \in J^*} u_j g_j(x) + (1 - \delta) \sum_{j \in J} r_j g_j(x)^2 + \Delta_1$$

$$= d^T G d - 2 \bar{g}^T (u - v) - \sum_{j \in J} \frac{1}{r_j} v_j (u_j - v_j) + 2 \sum_{j \in K^*} \frac{1}{r_j} v_j^2$$

$$- \sum_{j \in K \setminus K^*} u_j \left( g_j(x) + \frac{v_j}{r_j} \right) + \sum_{j \in K^*} \frac{1}{r_j} v_j^2 + \delta \sum_{j \in J \setminus J^*} u_j g_j(x)$$

$$+ \delta \sum_{j \in J} (u_j - v_j) g_j(x) + (1 - \delta) \sum_{j \in J} r_j g_j(x)^2 + \Delta_1$$

$$= d^T G d - 2 \bar{g}^T (u - v) + (1 - \delta) \bar{g}^T \bar{g} - \sum_{j \in J} u_j \left( g_j(x) + \frac{v_j}{r_j} \right)$$

$$+ \sum_{j \in K \setminus K^*} \frac{1}{r_j} v_j^2 + \delta \sum_{j \in J^* \setminus J} u_j g_j(x) + \delta \sum_{j \in J} (u_j - v_j) g_j(x) + \Delta_1.$$
This together with (2.5b), $\bar{v} \geq 0$, and the definition of $\bar{q}(\leq 0)$ in (2.18) shows that

$$S \geq d^T Gd - 2\bar{g}^T (u - v) + (1 - \delta)\bar{g}^T R\bar{g} + \bar{q}^2$$

$$- v_1\delta + v_2\delta - \delta \bar{v}^T \bar{g} + \frac{\delta}{1 - \delta} \bar{q}^T \bar{v} + A_1$$

$$= d^T Gd + \left\| \sqrt{1 - \delta} R^{1/2} \bar{g} - \frac{1}{\sqrt{1 - \delta}} R^{-1/2} (u - v) \right\|^2$$

$$- \frac{1}{1 - \delta} (u - v)^T R^{-1} (u - v) + \bar{q}^2 + \delta \left( \bar{v}^T \bar{g} + \frac{1}{1 - \delta} \bar{q}^T \bar{v} \right) + A_1$$

$$\geq d^T Gd - \frac{1}{1 - \delta} (u - v)^T R^{-1} (u - v) + \left( \sqrt{\bar{q}^2} + \frac{1}{2\sqrt{\bar{q}}} \left( \bar{v}^T \bar{g} + \frac{1}{1 - \delta} \bar{q}^T \bar{v} \right) \right)^2$$

$$- \frac{1}{4\bar{q}} \left( \bar{v}^T \bar{g} + \frac{1}{1 - \delta} \bar{q}^T \bar{v} \right)^2 + A_1$$

$$\geq \frac{1}{2} d^T Gd + \frac{1}{2} d^T Gd - \frac{1}{1 - \delta} (u - v)^T R^{-1} (u - v) - \frac{1}{4\bar{q}} \left( \bar{v}^T \bar{g} + \frac{1}{1 - \delta} \bar{q}^T \bar{v} \right)^2 + A_1$$

$$\geq \frac{1}{2} \gamma \|d\|^2 + \frac{1}{2} d^T Gd + A_1 - \frac{1}{1 - \delta} \sum_{j=1}^{m} \frac{1}{r_j} (u_j - v_j)^2 - \frac{1}{4\bar{q}} \left( \bar{v}^T \bar{g} + \frac{1}{1 - \delta} \bar{q}^T \bar{v} \right)^2,$$

where $R^\gamma = \text{diag}(r_1^\gamma, \ldots, r_m^\gamma)$.

From the definitions of $r_j$ (2.9) and $A_1$ in (2.21), we have from the last inequality above that

$$S \geq \frac{1}{2} \gamma \|d\|^2 + \frac{1}{2} d^T \left( G + \sum_{j \in J^*} u_j G_j + \sum_{j \in K^*} u_j G_j^{(j)} \right) d$$

$$- \frac{1}{1 - \delta} \sum_{j=1}^{m} \frac{1}{2m(u_j - v_j)^2} (u_j - v_j)^2 - \frac{1}{4\bar{q}} \left( \bar{v}^T \bar{g} + \frac{1}{1 - \delta} \bar{q}^T \bar{v} \right)^2$$

$$= \frac{1}{2} \gamma \|d\|^2 - \frac{1}{4\bar{q}(1 - \delta)^2} (\bar{v}^T (- (1 - \delta) \bar{g} + \bar{q}))^2$$

$$= \frac{1}{2} \gamma \|d\|^2 - \frac{1}{4\bar{q}(1 - \delta)^2} (\bar{v}^T A^T d)^2$$

$$\geq \frac{1}{2} \gamma \|d\|^2 - \frac{\gamma(1 - \delta)^2}{4(1 - \delta)^2 \|A\|_2^2 \|A\|_2^2} \|A\|_2^2 \|d\|^2$$

$$\geq \frac{1}{4} \gamma \|d\|^2,$$

which completes the proof. □

From Theorem 2.1 and taking into account the structure of the algorithm (especially noting that the algorithm can be stopped if $d^k = 0$ at Step 3), we have the following corollary.

**Corollary 2.1.** The loop between Steps 2 and 4 is finite, and thus the line search of Algorithm A is well-defined.

**Remark 2.** Due to the analysis given by Schittkowski [23] or Dai and Schittkowski [6], the assumptions (i)–(iii) of Theorem 2.1 are not restrictive at all. Particularly, assumption (i) is a standard assumption required in the theory of quasi-Newton algorithms, and it can be forced by choosing a (suitably small) $\gamma$ and performing a restart with $G^k$ being an identity matrix whenever (i) is violated. Assumption (ii) can be guaranteed by Step 2 of Algorithm A. Assumption (iii) allows at least a local estimate of the parameter $q_k$, and since the lower bounded in assumption (iii) does not depend on $d^k$, $u^k$, or $q_k$, the loop between Steps 2 and 4 is finite.
3. Global convergence

In this section, we establish the global convergence of the proposed algorithm. First, let us recall Lemma 4.2 of [6], which shows the convergence of the penalty parameters $r^k$.

**Lemma 3.1.** Suppose that $\{r^k\}_{k \in \mathbb{N}}$ is bounded and $\sigma_j \leq 1$ for all $k$. If (2.11) holds for a constant $h > 0$, then there is a $r^*_j \geq 0$, $j \in I$, such that $\lim_{k \to \infty} r^k_j = r^*_j$.

The following theorem will be very important for the global convergence analysis.

**Theorem 3.1.** Suppose that the statements (i)–(iii) of Theorem 2.1 hold, and that $\{x^k\}$, $\{d^k\}$, and $\{u^k\}$ are bounded. Then there is an infinite subset $\mathcal{K}$, such that $d^k \to 0$, $k \in \mathcal{K}$.

**Proof.** Without loss of generality, suppose by contradiction that there exists a constant $c > 0$ such that

$$d^k \geq c \quad \text{for all } k.$$  \hfill (3.1)

Next we will prove that, by considering the line search condition (2.17), there is a constant $\bar{\lambda} > 0$ such that the step size $\lambda_k \geq \bar{\lambda}$ for all $k$.

It follows from the boundedness of $\{u^k\}$ that $\{v^k\}$ is also bounded since $\bar{\lambda}_k \leq 1$ and is generated by the Armijo-type line search (2.17). Thus, for each iteration index $k$, from Taylor expansion, (2.16), (2.19) and (3.1), we have

$$\Psi_k(\lambda) - \Psi_k(0) + \lambda \Psi_k'(0) = (1 - \lambda)\lambda \nabla \Phi_{x_{k+1}}(x^k, v^k)^T (u^k - v^k) + o(\lambda)$$

$$\leq - \frac{1}{4} (1 - \lambda)\bar{\lambda} \gamma \|d^k\|^2 + o(\lambda)$$

$$\leq - \frac{1}{4} (1 - \lambda)\bar{\lambda} c^2 + o(\lambda).$$

This together with $\lambda \in (0, 0.5)$ shows that (2.17) holds for all $k$ and $\lambda > 0$ sufficiently small, i.e., there is a constant $\bar{\lambda} > 0$ such that the step size $\lambda_k \geq \bar{\lambda}$ for all $k$.

Again from (2.13) and (2.17), we have

$$\Phi_{x_k}(x^{k+1}, v^{k+1}) \leq \Phi_{x_k}(x^k, v^k) - \lambda \bar{\lambda} \gamma \|d^k\|^2$$

$$\leq \Phi_{x_k}(x^k, v^k) - \frac{1}{4} \lambda c^2$$

(3.2)

On the other hand, from the definition of $r^{k+1}$ (2.9), we know that either $r_j^{k+1} \leq \sigma_j r_j^0$ or there is a $k_0 \leq k$ such that

$$r_j^{k+1} \leq \frac{2m(u_j^{k_0} - v_j^{k_0})^2}{(1 - \delta_k)(d^{k_0})^2 Q_{k_0}^2}, \quad j \in I.$$  \hfill (3.3)

This together with assumption (i) of Theorem 2.1, (2.10) and (3.1) shows that

$$r_j^{k+1} \leq \frac{2m(u_j^{k_0} - v_j^{k_0})^2}{(1 - \delta_k) c^2}, \quad j \in I.$$  \hfill (3.3)

Then, from the boundedness of $\{u^k\}$ and $\{v^k\}$ (mentioned above), it follows that $\{r^k\}$ is bounded, and thus from Lemma 3.1 and (2.12) we have

$$\lim_{k \to \infty} r^k = r \quad \text{with } r_j > 0, \quad j \in I.$$
Similar to the analysis in [23, Theorem 4.6], we consider the difference
\[
\Phi_{R_k+2}(x^{k+1}, v^{k+1}) - \Phi_{R_k+1}(x^{k+1}, v^{k+1})
= \sum_{j \in J_{k+1}} \left( v_j^{k+1} g_j(x^{k+1}) + \frac{1}{2} r_j^{k+2} g_j(x^{k+1})^2 \right) - \frac{1}{2} \sum_{j \in K_{k+1}} (v_j^{k+1})^2 / r_j^{k+2}
- \sum_{j \in J_k} \left( v_j^{k+1} g_j(x^{k+1}) + \frac{1}{2} r_j^{k+1} g_j(x^{k+1})^2 \right) + \frac{1}{2} \sum_{j \in \bar{K}_k} (v_j^{k+1})^2 / r_j^{k+1},
\]
(3.4)
where
\[
J_{k+1} = \{ j : j \in I, g_j(x^{k+1}) \geq - v_j^{k+1} / r_j^{k+2} \},
J_k = \{ j : j \in I, g_j(x^{k+1}) \geq - v_j^{k+1} / r_j^{k+1} \},
\]
and \( K_{k+1}, \bar{K}_k \) are the corresponding complements. From (3.4), (3.3) and the boundedness of \( g_j(x^{k+1}) \) and \( v^{k+1} \), we have, for all \( k \) sufficiently large,
\[
\Phi_{R_k+2}(x^{k+1}, v^{k+1}) - \Phi_{R_k+1}(x^{k+1}, v^{k+1}) \leq \frac{1}{8} \gamma c^2.
\]
This together with (3.2) shows that, for all \( k \) sufficiently large,
\[
\Phi_{R_k+2}(x^{k+1}, v^{k+1}) \leq \Phi_{R_k+1}(x^{k+1}, v^{k+1}) + \frac{1}{8} \gamma c^2 \leq \Phi_{R_k+1}(x^k, v^k) - \frac{1}{8} \gamma c^2,
\]
which leads to a contradiction, since \( \{ \Phi_{R_k+1}(x^k, v^k) \} \) is bounded below. This completes the proof. \( \square \)

**Remark 3.** In practice, the boundedness of \( \{ x^k \} \) can be ensured by adding upper and lower bounds to \( x \) in problem (1.1), i.e., \( x_l \leq x \leq x_u \). And then the boundedness of \( \{ d^k \} \) is guaranteed by adding to subproblem a corresponding constraint: \( x_l - x^k \leq d < x_u - x^k \). Furthermore, the boundedness of \( \{ u^k \} \) holds at least when \( x^k \) approaches to a feasible point satisfying a certain constraint qualification (see a similar result in [25, Proposition 4]).

Based on the theorem above, we are able to prove the global convergence of Algorithm A.

**Theorem 3.2.** Suppose that all assumptions of Theorem 3.1 holds, and that \( \{ G^k \}, \{ G_j^k \}, \) and \( \{ G_j^k(\tilde{x}) \} \) generated by Algorithm A are bounded. Then there exists an accumulation point \((x^*, u^*)\) of \((x^k, u^k)\) such that \( x^* \) is a KKT point for problem (1.1) with multiplier \( u^* \).

**Proof.** Let \( \mathcal{H} \) be the infinite index set defined by Theorem 3.1. From Theorem 3.1 and the boundedness of \( \{ x^k \} \) and \( \{ u^k \} \), we can conclude that there exist an infinite subset \( \mathcal{H}' \subseteq \mathcal{H}, x^* \in \mathbb{R}^n \) and \( u^* \in \mathbb{R}^m \) such that
\[
\{ x^k, u^k, d^k \} \to \{ x^*, u^*, 0 \}, \quad k \in \mathcal{H}'.
\]
(3.5)

In view of the boundedness of \( \{ G^k \} \) and \( \{ G_j^k(\tilde{x}) \} \) as well as the fact that \( 0 \leq \delta_k \leq \delta \in (0, 1) \), passing to the limit \( k \to \infty \), \( k \in \mathcal{H}' \) in KKT conditions (2.5c), (2.5e), (2.5h), we have
\[
g_j(x^*) \leq 0, \quad u_j^* \geq 0, \quad u_j^* g_j(x^*) = 0, \quad j \in I.
\]
(3.6)

Since \( J_k^* \) and \( K_k^* \) are subsets of the finite set \( I = \{ 1, \ldots, m \} \), we may assume that, by passing to a subsequence if necessary,
\[
J_k^* = J^*, \quad K_k^* = K^*, \quad \forall k \in \mathcal{H}'.
\]
So by the definition of \( K_k^* \), we know that \( g_j(x^*) < 0, j \in K^* \) and thus \( u_j^* = 0, j \in K^* \). Passing to the limit \( k \to \infty \), \( k \in \mathcal{H}' \) in (2.5a), we have from (3.5) and the boundedness of \( \{ G^k \}, \{ G_j^k \}, \) and \( \{ G_j^k(\tilde{x}) \} \) that
\[
\nabla f(x^*) + \sum_{j \in J^*} u_j^* \nabla g_j(x^*) = 0.
\]
This together with (3.6) shows that \( x^* \) is a KKT point for problem (1.1) with multiplier \( u^* \). \( \square \)
Remark 4. We note that the boundedness of \( \{ G_k \} \) and \( \{ G_j^k \} \) is a basic assumption in the SQCQP methods.

4. Superlinear and quadratic convergence

We now examine the local convergence behavior of Algorithm A. Let us first recall the definition of the active set at a feasible point \( x^* \):

\[
I(x^*) \overset{\text{def}}{=} \{ j \in I : g_j(x^*) = 0 \}.
\]

In order to obtain fast local rate of convergence, besides the assumptions made in Section 3, the following assumptions are necessary.

Assumption 2. Assume that

(a) \( \lim_{k \to \infty} (x^k, u^k) = (x^*, u^*) \), where \( (x^*, u^*) \) is a KKT pair of problem (1.1),
(b) \( \lim_{k \to \infty} d^k = 0 \),
(c) \( \{ r^k \} \) is bounded.
(d) the Mangasarian–Fromovitz constraint qualification holds at \( x^* \), i.e., there exists a vector \( \bar{d} \in \mathbb{R}^n \) such that

\[
\nabla g_j(x^*)^T \bar{d} < 0, \quad \forall j \in I(x^*),
\]

(e) the strict complementarity assumption holds at \( x^* \), i.e., \( u^* > 0 \) for \( j \in I(x^*) \),
(f) \( f, g_j, j \in I \) are twice continuously differentiable, and

\[
(\nabla^2 f(x^k) - G_k^k)d^k = o(\|d^k\|), \quad (\nabla^2 g_j(x^k) - G_j^k)d^k = o(\|d^k\|), \quad j \in J_k,
\]

(g) \( \{ G_k \} \), \( \{ G_j^k \} \), and \( \{ G_j^{k(j)} \} \) generated by Algorithm A are bounded.

Remark 5. Assumption 2(a) and (b) are typically made in the local convergence analysis of SQP methods. Assumption 2(d) is weaker than the linear independent constraint qualification used in [7]. Assumption 2(e) is also made in many (old or new) SQP methods (see e.g., [5,4,15,26]) and in the SQCQP method of [7]. Assumption 2(f) is weaker than using the exact Hessian matrices of the objective function and constraints that appears in [8,1,7,12,13], more precisely, our superlinear convergence analysis does not use the exact Hessian matrices but their approximations. So, in practice, our SQCQP algorithm is well-defined whether the exact Hessian matrices exist or not.

The following lemma will show an important result that subproblem (2.1) always has a feasible solution in the neighborhood of \( x^* \), i.e., the constraints of subproblem (2.1) are consistent for all \( k \) sufficiently large. This property will greatly simplify our algorithm.

Lemma 4.1. Suppose that Assumption 2 holds. Then

(i) \( \lim_{k \to \infty} \| u^k - v^k \| = 0 \),
(ii) the constraints of subproblem (2.1) are consistent for all \( k \) sufficiently large,
(iii) for \( k \) sufficiently large, it follows that

\[
g_j(x^k) + \nabla g_j(x^k)d^k + \frac{1}{2}(d^k)^T G_j^k d^k = 0, \quad j \in J_k.
\]

Proof. (i) From the definition of \( i^{k+1} \), we have

\[
\frac{(u_j^k - u_j^*)^2}{r_j^{k+1}} \leq \frac{(1 - \delta_k)(d^k)^T Q d^k}{2m} \leq \frac{1}{2}(d^k)^T Q d^k.
\]

This together with Assumption 2(b), (c) and (g) shows that part (i) holds.
(ii) For any \( j \notin I(x^*) \), there exists a constant \( \zeta_1 > 0 \) such that \( g_j(x^*) < - \zeta_1 \). Then it follows from Assumption 2(a) that there exists a \( k_1 \) such that

\[
g_j(x^k) < - \frac{\zeta_1}{2}, \quad \forall k \geq k_1, \quad j \notin I(x^*).
\]

This implies that there is a constant \( \tau_1 > 0 \) such that

\[
g_j(x^k) + \nabla g_j(x^k)^T(\tau \tilde{d}) + \frac{1}{2}(\tau \tilde{d})^T G_j^k (\tau \tilde{d}) < 0, \quad j \notin I(x^*) \tag{4.2}
\]

holds for all \( k \geq k_1 \) and all \( \tau \leq \tau_1 \), where \( \tilde{d} \) is given by Assumption 2(d).

For any \( j \in I(x^*) \), from Assumption 2(d), we have that there exist a constant \( \zeta_2 > 0 \) and a vector \( \tilde{d} \in \mathbb{R}^n \) such that

\[
\nabla g_j(x^*)^T \tilde{d} < - \zeta_2, \quad \forall j \in I(x^*).
\]

Then it follows from Assumption 2(a) that there exists a \( k_2 \geq k_1 \) such that

\[
\nabla g_j(x^k)^T \tilde{d} < - \frac{\zeta_2}{2}, \quad \forall j \in I(x^*), \quad \forall k \geq k_2.
\]

Thus for any \( j \in I(x^*) \subseteq J_k^* \), we have that there exists a \( \tau_2 \leq \tau_1 \) such that, for any \( \tau \leq \tau_2 \) and \( k \geq k_2 \),

\[
g_j(x^k) + \nabla g_j(x^k)^T(\tau \tilde{d}) + \frac{1}{2}(\tau \tilde{d})^T G_j^k (\tau \tilde{d}) = g_j(x^k) + \tau (\nabla g_j(x^k)^T \tilde{d} + \frac{1}{2} \tau \tilde{d}^T G_j^k \tilde{d})
\]

\[
< g_j(x^k) - \frac{\tau \zeta_2}{4}.
\]

This together with \( g_j(x^k) \to g_j(x^*) = 0, \quad j \in I(x^*) \) shows that there exists a \( k_3 \geq k_2 \) such that

\[
g_j(x^k) + \nabla g_j(x^k)^T(\tau \tilde{d}) + \frac{1}{2}(\tau \tilde{d})^T G_j^k (\tau \tilde{d}) < 0, \quad j \in I(x^*) \tag{4.3}
\]

hold for all \( \tau \leq \tau_2 \) and \( k \geq k_3 \). From (4.2) and (4.3), we can conclude that \( \tau \tilde{d} \) with \( \tau \leq \tau_2 \) satisfies the first group of constraints of subproblem (2.1) for \( k \geq k_3 \). Further noting that \( g_j(x^k) \leq - \epsilon \) for \( j \in K_k^* \), we have that, decrease \( \tau \) if necessary, \( \tau \tilde{d} \) also satisfies the second group of constraints of subproblem (2.1). Hence, \( \tau \tilde{d} \) is a (strictly) feasible solution of subproblem (2.1) for \( k \) sufficiently large, and this completes the proof of part (ii).

(iii) For any \( j \in I(x^*) \subseteq J_k^* \), from Assumption 2(e), we have that \( u_j^k > 0 \) for \( k \) sufficiently large, and thus

\[
g_j(x^k) + \nabla g_j(x^k)d^k + \frac{1}{2}(d^k)^T G_j^k d^k = 0, \quad j \in I(x^*).
\]

On the other hand, for any \( j \notin I(x^*) \), we have \( g_j(x^*) < 0 \) and \( u_j^* = 0 \). Then it follows from (3.3) that

\[
g_j(x^*) < - u_j^*/r_j.
\]

Since \( u_j^k \to u_j^* \) from Lemma 4.1(i), we have that

\[
g_j(x^k) < - u_j^k/r_j^{k+1}
\]

holds for all \( k \) sufficiently large. This implies \( j \notin J_k \) and thus \( J_k \subseteq I(x^*) \). Hence part (iii) holds. □

**Remark 6.** From the proof of part (ii) of Lemma 4.1, and in view of the fact that subproblem (2.1) is a strictly convex programming, we know that (2.1) has a Slater point, and thus its solution is equivalent to its KKT point.

Due to the property obtained from Lemma 4.1(ii), Algorithm A is significantly simplified after early iterations, i.e., Steps 2–5 are not performed. So in the following local convergence analysis, we only need to consider the simplified version of Algorithm A without Steps 2–5. As suggested by a referee, a simple example is given below to illustrate the above properties.
**Example 1.** Consider the single inequality constrained optimization

\[
\begin{align*}
\text{min} & \quad f(x) = x_1^2 + x_2^4 - 4x_1 - x_2 \\
\text{s.t.} & \quad g_1(x) = x_1^4 + x_2^2 - \frac{1}{4} \leq 0.
\end{align*}
\]

The approximately optimal solution (computed by Matlab 6.5 using the function “fmincon”) of this example is \(x^* = (0.7162, 0.2649)\)^T. Now, we suppose that an iteration sequence \(\{x^k\}\) generated by Algorithm A converges to \(x^*\) (this is guaranteed by the global convergence property of the algorithm), and then we will analyze that how far the iteration point is close to \(x^*\) will lead to a feasible QCQP subproblem. For Example 1, the only constraint included in the QCQP subproblem (2.1) is

\[
q_1(d) \overset{\text{def}}{=} g_1(x) + \nabla g_1(x)^T d + \frac{1}{2} d^T G_1 d \leq 0. \tag{4.4}
\]

For simplicity, we choose \(G_1 = \nabla^2 g_1(x)\), and thus from \(\nabla g_1(x) = (4x_1^3, 2x_2)^T\) and \(\nabla^2 g_1(x) = \begin{pmatrix} 12x_1^2 & 0 \\ 0 & 2 \end{pmatrix}\) we obtain

\[
q_1(d) = x_1^4 + x_2^2 - \frac{1}{3} + (4x_1^3, 2x_2) \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} + \frac{1}{2} (d_1, d_2) \begin{pmatrix} 12x_1^2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \\
= (6x_1^2d_1^2 + 4x_1^3d_1) + (d_2^2 + 2x_2d_2) + x_1^4 + x_2^2 = \frac{1}{3}.
\]

We first consider the case of \(x_1 = 0\), i.e.,

\[
q_1(d) = d_2^2 + 2x_2d_2 + x_2^2 - \frac{1}{3},
\]

which is a quadratic function of variable \(d_2\), so it is easy to see that the minimal value of \(q_1(d)\) is \(-\frac{1}{3}\) which achieves at \(d_2 = -x_2\). Next, we consider the case of \(x_1 \neq 0\), and in this case \(q_1(d)\) is a sum of two quadratic functions of separate variables \(d_1\) and \(d_2\), so it is not difficult to get that the minimal value of \(q_1(d)\) is \(\frac{1}{3}x_1^4 - \frac{1}{3}\) which achieves at \(d = (-\frac{1}{3}x_1, -x_2)^T\).

From the two cases analyzed above, we can conclude that the constraint (4.4) is a feasible constraint whenever \(\frac{1}{3}x_1^4 - \frac{1}{3} \leq 0\) (\(x_2\) is irrelevant), namely, \(-1 \leq x_1 \leq 1\). So, the interval \((-1, 1)\) could be seen as a neighborhood of \(x_1^* = 0.7162\), and along with the increase of iteration number \(k\), the first component \(x_1^k\) will asymptotically lie within \((-1, 1)\), and finally all the rest of iterations lie within \((-1, 1)\) for \(k\) sufficiently large. In summary, the constraint (4.4) is feasible (i.e., the QCQP subproblem (2.1) is consistent) for \(k\) large enough such that \(x_1^k \in (-1, 1)\), and therefore the Steps 2–5 are not performed.

The subsequent theorem will be fundamental for the local convergence analysis, which shows that the step size is one in a neighborhood of the solution.

**Theorem 4.1.** Under all abovementioned assumptions, the step size in Algorithm A always equals one, i.e., \(\lambda_k \equiv 1\), for all \(k\) sufficiently large.

**Proof.** Define

\[
\bar{A}_k = (a_1(x^k), \ldots, a_m(x^k)), \quad a_j(x^k) = \begin{cases} \nabla g_j(x^k) & \text{if } j \in J_k, \\ \mathbf{0}_{n \times 1} & \text{otherwise,} \end{cases}
\]

\[
Z_k = \text{diag}(z_1^k, \ldots, z_m^k), \quad z_j^k = \begin{cases} 0 & \text{if } j \in J_k, \\ -1/r_j^{k+1} & \text{otherwise.} \end{cases}
\]

For notational convenience, we omit the index \(k\) in the subsequent proof. The Hessian matrix of \(\Phi_r(x, v)\) can be expressed as

\[
\nabla^2 \Phi_r(x, v) = \begin{pmatrix} \nabla^2 f(x) + \sum_{j \in J} (v_j \nabla^2 g_j(x) + r_j \nabla g_j(x) \nabla g_j(x)^T + r_j g_j(x) \nabla^2 g_j(x)) & \bar{A}^T \\ \bar{A} & Z \end{pmatrix}.
\]
Let us denote
\[ M_r(x, v) = \nabla^2 f(x) + \sum_{j \in J} (v_j \nabla^2 g_j(x) + r_j \nabla g_j(x) \nabla g_j(x)^T + r_j g_j(x) \nabla^2 g_j(x)), \]
\[ p^k = \left( \begin{array}{c} d^k \\ u^k \end{array} \right) \]
and thus
\[ p^T \nabla^2 \Phi_r(x, v)p = \left( \begin{array}{c} d \\ u - v \end{array} \right)^T \left( \begin{array}{cc} M_r(x, v) & \bar{A} \\ \bar{A}^T & Z \end{array} \right) \left( \begin{array}{c} d \\ u - v \end{array} \right) \]
\[ = (d^T M_r(x, v) + (u - v)^T \bar{A}^T \bar{A} + (u - v)^T Z) \left( \begin{array}{c} d \\ u - v \end{array} \right) \]
\[ = d^T M_r(x, v)d + (u - v)^T \bar{A}^T \bar{A} + Z(u - v) \]
\[ = d^T \nabla^2 f(x)d + \sum_{j \in J} (v_j d^T \nabla^2 g_j(x)d + r_j d^T \nabla g_j(x) \nabla g_j(x)^T d + r_j g_j(x)d^T \nabla^2 g_j(x)d) \]
\[ + (u - v)^T \bar{A}^T \bar{A} + Z(u - v) \]
\[ = d^T \nabla^2 f(x)d + \sum_{j \in J} (v_j d^T \nabla^2 g_j(x)d + r_j (\nabla g_j(x)d)^2 + r_j g_j(x)d^T \nabla^2 g_j(x)d) \]
\[ + 2d^T \bar{A}(u - v) + (u - v)^T Z(u - v). \]

From Taylor expansion, we have
\[ \Phi_r(x + d, v + (u - v)) - \Phi_r(x, v) = \alpha \nabla \Phi_r(x, v)^T p \]
\[ = (1 - \alpha) \nabla \Phi_r(x, v)^T p + \frac{1}{2} p^T \nabla^2 \Phi_r(x, v)p + o(\|p\|^2) \]
\[ = (1 - \alpha) \nabla \Phi_r(x, v)^T p + \frac{1}{2} (\nabla \Phi_r(x, v)^T p + p^T \nabla^2 \Phi_r(x, v)p) + o(\|p\|^2). \]

Denote
\[ Y = \nabla \Phi_r(x, v)^T p + p^T \nabla^2 \Phi_r(x, v)p, \]
and then we will prove that \( Y = o(\|d\|^2) \).

Substituting (2.20) and (4.5) to (4.7), we have
\[ Y = \nabla f(x)^T d + \sum_{j \in J^*} (v_j \nabla g_j(x)^T d + r_j g_j(x) \nabla g_j(x)^T d) + \bar{g}^T (u - v) \]
\[ + d^T \nabla^2 f(x)d + \sum_{j \in J} (v_j d^T \nabla^2 g_j(x)d + r_j (\nabla g_j(x)d)^2 + r_j g_j(x)d^T \nabla^2 g_j(x)d) \]
\[ + 2d^T \bar{A}(u - v) + (u - v)^T Z(u - v). \]

This together with (2.5a) shows that
\[ Y = -d^T Gd - \sum_{j \in J^*} u_j (\nabla g_j(x)^T d + d^T G_j d) - \sum_{j \in K^*} u_j (\nabla g_j(x_k(j))^T d + d^T G_j^{k(j)} d) \]
\[ + \sum_{j \in J} (v_j + r_j g_j(x)) \nabla g_j(x)^T d + \bar{g}^T (u - v) \]
\[ + d^T \nabla^2 f(x)d + \sum_{j \in J} (v_j + r_j g_j(x))d^T \nabla^2 g_j(x)d + \sum_{j \in J} r_j (\nabla g_j(x)d)^2 \]
\[ + 2d^T \bar{A}(u - v) + (u - v)^T Z(u - v). \]
Thus from the definition of \( A_1 \) in (2.21), we have
\[
Y = d^T(\nabla^2 f(x) - G)d - \sum_{j \in J^*} (u_j - v_j) \left( \nabla g_j(x)^T d + \frac{1}{2} d^T G_j d \right) - \sum_{j \in J^*} v_j \left( \nabla g_j(x)^T d + \frac{1}{2} d^T G_j d \right) \\
- \sum_{j \in K^*} u_j \left( \nabla g_j(x^{k(j)})^T d + \frac{1}{2} d^T G_j^{k(j)} d \right) - A_1 + \sum_{j \in J} (v_j + r_j g_j(x)) \nabla g_j(x)^T d + \tilde{g}^T (u - v) \\
+ \sum_{j \in J} (v_j + r_j g_j(x)) d^T \nabla^2 g_j(x) d + \sum_{j \in J} r_j (\nabla g_j(x)^T d + 2d^T \tilde{A}(u - v) + (u - v)^T Z(u - v) \\
= d^T(\nabla^2 f(x) - G)d - \sum_{j \in J^*} (u_j - v_j) \left( \nabla g_j(x)^T d + \frac{1}{2} d^T G_j d \right) - \sum_{j \in J^*} u_j \left( \nabla g_j(x^{k(j)})^T d + \frac{1}{2} d^T G_j^{k(j)} d \right) \\
- A_1 - \sum_{j \in J^* \setminus J} v_j \left( \nabla g_j(x)^T d + \frac{1}{2} d^T G_j d \right) + \sum_{j \in J} r_j g_j(x) \nabla g_j(x)^T d + \tilde{g}^T (u - v) \\
+ \frac{1}{2} \sum_{j \in J} v_j d^T (\nabla^2 g_j(x) - G_j) d + \frac{1}{2} \sum_{j \in J} v_j d^T \nabla^2 g_j(x) d + \sum_{j \in J} r_j (\nabla g_j(x)^T d + 2d^T \tilde{A}(u - v) + (u - v)^T Z(u - v) \\
+ (u - v)^T Z(u - v) + o(\|d\|^2)
\]

Hence from Assumption 2(f) and the definition of \( w_j \) (2.6) with \( \delta = 0 \), we have
\[
Y = - \sum_{j \in J^*} (u_j - v_j) \left( \nabla g_j(x)^T d + \frac{1}{2} d^T G_j d \right) - \sum_{j \in J^*} u_j \left( \nabla g_j(x^{k(j)})^T d + \frac{1}{2} d^T G_j^{k(j)} d \right) - A_1 \\
- \sum_{j \in J^* \setminus J} v_j \left( \nabla g_j(x)^T d + \frac{1}{2} d^T G_j d \right) + \tilde{g}^T (u - v) + \sum_{j \in J} r_j g_j(x) \nabla g_j(x)^T d + \tilde{g}^T (u - v) \\
+ \frac{1}{2} \sum_{j \in J} (v_j + r_j g_j(x)) d^T \nabla^2 g_j(x) d + \sum_{j \in J} r_j (\nabla g_j(x)^T d + 2d^T \tilde{A}(u - v) + (u - v)^T Z(u - v) \\
+ (u - v)^T Z(u - v) + o(\|d\|^2)
\]
\[
= \sum_{j \in J} w_j (v_j + r_j g_j(x)) + \sum_{j \in J^* \setminus J} v_j w_j + 2 \sum_{j \in J} (u_j - v_j) g_j(x) + \sum_{j \in J^* \setminus J} (u_j - v_j) g_j(x) \\
+ \sum_{j \in J^* \setminus J} u_j g_j(x) - A_1 - \sum_{j \in J^* \setminus J} v_j w_j + \sum_{j \in J^* \setminus J} v_j g_j(x) - \sum_{j \in J} \frac{1}{r_j} v_j (u_j - v_j) \\
- \sum_{j \in J} r_j g_j(x)^2 + \frac{1}{2} \sum_{j \in J} (v_j + r_j g_j(x)) d^T \nabla^2 g_j(x) d + \sum_{j \in J} r_j (\nabla g_j(x)^T d)^2 \\
+ 2d^T \tilde{A}(u - v) + (u - v)^T Z(u - v) + o(\|d\|^2).
\]
This together with $w_j \leq 0$ and $v_j + r_j g_j(x) > 0$, $j \in J$ shows that

$$
\begin{align*}
Y &\leq 2 \sum_{j \in J} (u_j - v_j) g_j(x) + \frac{1}{2} \sum_{j \in J} u_j g_j(x) + \sum_{j \in J} u_j g_j(x) - \Delta_1 - \sum_{j \in K} \frac{1}{r_j} v_j (u_j - v_j) \\
&\quad - \sum_{j \in J} r_j g_j(x)^2 + \frac{1}{2} \sum_{j \in J} (v_j + r_j g_j(x)) d^T \nabla^2 g_j(x) d + \sum_{j \in J} r_j (\nabla g_j(x)^T d)^2 \\
&\quad + 2 d^T \bar{A}(u - v) + (u - v)^T Z(u - v) + o(\|d\|^2) \\
&= 2 \sum_{j \in J} (u_j - v_j) g_j(x) - 2 \sum_{j \in J} \frac{1}{r_j} v_j (u_j - v_j) + \sum_{j \in J} \frac{1}{r_j} v_j (u_j - v_j) + \sum_{j \in K} u_j g_j(x) \\
&\quad - \left( \frac{1}{2} \sum_{j \in J} u_j d^T G_j d + \frac{1}{2} \sum_{j \in K} u_j d^T G_j^{(k)} d \right) \\
&\quad - \sum_{j \in J} r_j g_j(x)^2 + \frac{1}{2} \sum_{j \in J} (v_j + r_j g_j(x)) d^T \nabla^2 g_j(x) d + \sum_{j \in J} r_j (\nabla g_j(x)^T d)^2 \\
&\quad + 2 \sum_{j \in J} (u_j - v_j) \nabla g_j(x)^T d - \sum_{j \in K} \frac{1}{r_j} (u_j - v_j)^2 + o(\|d\|^2).
\end{align*}
$$

Further from $u_j^k > 0$ and the positive semidefiniteness of $G_j$ and $G_j^{(k)}$, we have

$$
\begin{align*}
Y &\leq 2 \sum_{j \in J} (u_j - v_j) g_j(x) - 2 \sum_{j \in J} \frac{1}{r_j} v_j (u_j - v_j) + \sum_{j \in K} u_j \left( g_j(x) + \frac{v_j}{r_j} \right) - \sum_{j \in K} \frac{v_j^2}{r_j} \\
&\quad - \sum_{j \in J} r_j g_j(x)^2 + \frac{1}{2} \sum_{j \in J} (u_j - v_j) d^T G_j d + \frac{1}{2} \sum_{j \in J} r_j g_j(x)^2 \nabla^2 g_j(x) d + \sum_{j \in J} r_j (\nabla g_j(x)^T d)^2 \\
&\quad + 2 \sum_{j \in J} (u_j - v_j) \nabla g_j(x)^T d - \sum_{j \in K} \frac{1}{r_j} (u_j - v_j)^2 + o(\|d\|^2) \\
&\leq 2 \sum_{j \in J} (u_j - v_j) (g_j(x) + g_j(x)^T d) - \frac{1}{2} \sum_{j \in J} (u_j - v_j) d^T G_j d \\
&\quad - \left( 2 \sum_{j \in K} \frac{1}{r_j} v_j (u_j - v_j) + \sum_{j \in K} \frac{v_j^2}{r_j} + \sum_{j \in K} \frac{1}{r_j} (u_j - v_j)^2 \right) \\
&\quad - \sum_{j \in J} r_j g_j(x)^2 + \frac{1}{2} \sum_{j \in J} r_j g_j(x)^2 \nabla^2 g_j(x) d + \sum_{j \in J} r_j (\nabla g_j(x)^T d)^2 + o(\|d\|^2).
\end{align*}
$$

Then it follows from Lemma 4.1(iii) that

$$
\begin{align*}
Y &\leq - \frac{3}{2} \sum_{j \in J} (u_j - v_j) d^T G_j d - \sum_{j \in K} r_j (v_j + (u_j - v_j))^2 \\
&\quad - \sum_{j \in J} r_j g_j(x)^2 + \frac{1}{2} \sum_{j \in J} r_j g_j(x)^2 \nabla^2 g_j(x) d + \sum_{j \in J} r_j (\nabla g_j(x)^T d)^2 + o(\|d\|^2) \\
&\leq - \frac{3}{2} \sum_{j \in J} (u_j - v_j) d^T G_j d + \frac{1}{2} \sum_{j \in J} r_j g_j(x)^2 \nabla^2 g_j(x) d \\
&\quad + \sum_{j \in J} r_j ((\nabla g_j(x)^T d)^2 - g_j(x)^2) + o(\|d\|^2).
\end{align*}
$$

(4.8)
Using Assumption 2 and Lemma 4.1, we can conclude that, for \( j \in J \),

\[
(u_j - v_j) d^T G_j d = o(||d||^2),
\]

\[
g_j(x)d^T \nabla^2 g_j(x)d = o(||d||^2),
\]

\[
(\nabla g_j(x)^T d)^2 - g_j(x)^2 = o(||d||^2).
\]

From the above relations and (4.8), we have that \( Y = o(||d||^2) \).

Combining (4.6) and (4.7), we have

\[
\Phi_r(x + d, v + (u - v)) - \Phi_r(x, v) - \gamma \nabla \Phi_r(x, v)^T p
\]

\[
= \left( \frac{1}{2} - \gamma \right) \nabla \Phi_r(x, v)^T p + O(||d||^2) + o(||p||^2).
\]

Then it follows from (2.19) and Assumption 2(b), (c) that

\[
\Phi_r(x + d, v + (u - v)) - \Phi_r(x, v) - \gamma \nabla \Phi_r(x, v)^T p
\]

\[
\leq - \frac{\gamma}{4} \left( \frac{1}{2} - \gamma \right) ||d||^2 + O(||d||^2)
\]

\[
\leq 0
\]

holds for all \( k \) sufficiently large, i.e., the step size in Algorithm A always equals one for \( k \) sufficiently large. \( \square \)

Now we are able to establish the local rate of convergence for Algorithm A.

**Theorem 4.2.** Suppose that all abovementioned assumptions hold, \( f, g_j, j \in I \) have Lipschitz-continuous second derivatives, and that the matrix \( \nabla^2 f(x^*) + \sum_{j \in I} u_j^* \nabla^2 g_j(x^*) \) is positive definite, where \( x^* \) and \( u^* \) are stated in Assumption 2(a). Then Algorithm A is superlinearly convergent, i.e.,

\[
\|x^{k+1} - x^*\| = o(||x^k - x^*||).
\]

Furthermore, if the exact Hessian matrices are used, i.e., \( G_k = \nabla^2 f(x^k) \) and \( G_j^k = \nabla^2 g_j(x^k) \), \( j \in J_k^* \), for \( k \) sufficiently large, then Algorithm A is quadratically convergent, i.e.,

\[
\|x^{k+1} - x^*\| = O(||x^k - x^*||^2).
\]

**Proof.** We first define \( A_k \) to be the active set of subproblem (2.1), i.e.,

\[
A_k \overset{\text{def}}{=} \{ j \in J_k^* : g_j(x^k) + \nabla g_j(x^k)^T d^k + \frac{1}{2} (d^k)^T G_j^k d^k = 0 \}
\]

\[
\cup \{ j \in K_k^* : g_j(x^k) + \nabla g_j(x^k)^T d^k + \frac{1}{2} (d^k)^T G_j^k d^k = 0 \}.
\]

It is easy to see that \( A_k \subseteq J_k^* \) for all \( k \) sufficiently large. Since \( A_k \) only takes a finite number of distinct values, it is sufficient to prove (4.11) and (4.12) hold for \( A_k \equiv A \) for some subset \( A \subseteq I \). From the KKT conditions of subproblem (2.1) (similar to (2.5)), we have

\[
\nabla f(x^k) + \sum_{j \in A} u_j^* \nabla g_j(x^k) = 0, \quad g_j(x^k) = 0, \quad u_j^* \geq 0, \quad j \in A,
\]

\[
\nabla f(x^k) + G^k d^k + \sum_{j \in A} u_j^* (\nabla g_j(x^k) + G_j^k d^k) = 0,
\]

\[
g_j(x^k) + \nabla g_j(x^k)^T d^k + \frac{1}{2} d^T G_j^k d^k = 0, \quad j \in A.
\]
It follows from Assumption 2(f), (4.13) and (4.14) that
\[
0 = (\nabla f(x^k) - \nabla f(x^*)) + G^k d^k + \sum_{j \in A} (u^k_j \nabla g_j(x^k) - u^*_j \nabla g_j(x^*) + u^*_j G_j d^k)
\]
\[
= (\nabla f(x^k) - \nabla f(x^*)) + \nabla^2 f(x^k) d^k + \sum_{j \in A} (u^k_j \nabla g_j(x^k) - u^*_j \nabla g_j(x^*) + u^*_j \nabla^2 g_j(x^k) d^k) + o(\|d^k\|)
\]
\[
= \left(\nabla^2 f(x^k) + \sum_{j \in A} u^k_j \nabla^2 g_j(x^k)\right) (x^k + d^k - x^*) + \sum_{j \in A} (u^k_j - u^*_j) \nabla g_j(x^*)
\]
\[
+ (\nabla f(x^k) - \nabla f(x^*) - \nabla^2 f(x^k)(x^k - x^*)) \sum_{j \in A} u^k_j (\nabla g_j(x^k) - \nabla g_j(x^*) - \nabla^2 g_j(x^k - x^*))
\]
\[
+ o(\|d^k\|).
\]

(4.16)

On the other hand, we have from mid-value expression that
\[
\nabla f(x^k) - \nabla f(x^*) - \nabla^2 f(x^k)(x^k - x^*) = O(\|x^k - x^*\|^2),
\]
\[
\nabla g_j(x^k) - \nabla g_j(x^*) - \nabla^2 g_j(x^k)(x^k - x^*) = O(\|x^k - x^*\|^2), \quad j \in A.
\]

Let us denote
\[
\nabla^2 \mathcal{L}(x^k, u^k) = \nabla^2 f(x^k) + \sum_{j \in A} u^k_j \nabla^2 g_j(x^k),
\]
\[
\nabla g_A(x^*) = (\nabla g_j(x^*), \quad j \in A), \quad u^*_A = (u^*_j, \quad j \in A), \quad u^*_A = (u^*_j, \quad j \in A).
\]

Then we have
\[
\nabla^2 \mathcal{L}(x^k, u^k)(x^k + d^k - x^*) + \nabla g_A(x^*)(u^k_A - u^*_A) = O(\|x^k - x^*\|^2) + o(\|d^k\|).
\]

(4.17)

In addition, from (4.15) we obtain, for \(j \in A\)
\[
0 = g_j(x^k) + \nabla g_j(x^k)^T d^k + \frac{1}{2} d^T G_j^T d^k - g_j(x^*)
\]
\[
= \nabla g_j(x^*)^T (x^k + d^k - x^*) + (g_j(x^k) - g_j(x^*) - \nabla g_j(x^*)^T (x^k - x^*))
\]
\[
+ (\nabla g_j(x^*) - \nabla g_j(x^*))^T d^k + \frac{1}{2} d^T G_j^T d^k
\]
\[
= \nabla g_j(x^*)^T (x^k + d^k - x^*) + O(\|x^k - x^*\|^2) + O(\|x^k - x^*\| \cdot \|d^k\|) + O(\|d^k\|^2),
\]

(4.18)

which implies
\[
\nabla g_j(x^*)^T (x^k + d^k - x^*) = O(\|x^k - x^*\|^2) + o(\|d^k\|).
\]

This together with (4.17) and the fact \(x^{k+1} = x^k + d^k\) shows that
\[
\begin{pmatrix}
\nabla^2 \mathcal{L}(x^k, u^k) & \nabla g_A(x^*) \\
\nabla g_A(x^*)^T & 0
\end{pmatrix}
\begin{pmatrix}
x^{k+1} - x^* \\
u^k_A - u^*_A
\end{pmatrix}
= O(\|x^k - x^*\|^2) + o(\|d^k\|).
\]

(4.19)

Let \(\{\nabla g_j(x^*), \quad j \in A' \subseteq A\}\) be a maximum linearly independent subset of \(\{\nabla g_j(x^*), \quad j \in A\}\). Then there exists a vector \(\tilde{u}^k_{A'} \in R^{A'}\) such that \(\nabla g'_{A'}(x^*) \tilde{u}^k_{A'} = \nabla g_A(x^*)(u^k_A - u^*_A)\), so it follows from (4.19) that
\[
\begin{pmatrix}
\nabla^2 \mathcal{L}(x^k, u^k) & \nabla g_A(x^*) \\
\nabla g_A(x^*)^T & 0
\end{pmatrix}
\begin{pmatrix}
x^{k+1} - x^* \\
\tilde{u}^k_{A'}
\end{pmatrix}
= O(\|x^k - x^*\|^2) + o(\|d^k\|).
\]

(4.20)

It can be proved that (see the similar proof in Theorem 4.2 of [12]), for \(k\) sufficiently large, the matrix
\[
N_k \overset{\text{def}}{=} \begin{pmatrix}
\nabla^2 \mathcal{L}(x^k, u^k) & \nabla g_A(x^*) \\
\nabla g_A(x^*)^T & 0
\end{pmatrix}
\]
is nonsingular, and there exists a constant $\tilde{c} > 0$ such that $\|N_k^{-1}\| \leq \tilde{c}$. So, from (4.20) we have
\[
\|x^{k+1} - x^\ast\| = O(\|x^k - x^\ast\|^2) + o(\|d^k\|)
= o(\|x^k - x^\ast\|) + o(\|d^k\|)
\leq o(\|x^k - x^\ast\|) + o(\|d^k\|)\|(x^{k+1} - x^\ast) - (x^k - x^\ast)\|
\leq o(\|x^k - x^\ast\|) + o(\|d^k\|)\|(x^{k+1} - x^\ast) + \|x^k - x^\ast\|)
= o(\|x^k - x^\ast\|) + o(\|x^{k+1} - x^\ast\|),
\]
which further implies
\[
\frac{\|x^{k+1} - x^\ast\|}{\|x^k - x^\ast\|} \left(1 - \frac{o(\|x^{k+1} - x^\ast\|)}{\|x^{k+1} - x^\ast\|}\right) \leq \frac{o(\|x^k - x^\ast\|)}{\|x^k - x^\ast\|},
\]
and therefore the relation (4.11) holds.

Now we prove the second part of the theorem. If the exact Hessian matrices are used, then the term $o(\|d^k\|)$ in (4.16) will vanish, so in view of (4.18) we can conclude that the right-hand side of (4.20) will be
\[
O(\|x^k - x^\ast\|^2) + O(\|x^k - x^\ast\| \cdot \|d^k\|) + O(\|d^k\|^2).
\]
Thus, using the properties of $N_k$ again, we have
\[
\|x^{k+1} - x^\ast\| = O(\|x^k - x^\ast\|^2) + O(\|x^k - x^\ast\| \cdot \|d^k\|) + O(\|d^k\|^2).
\]
(4.21)
On the other hand, from the proved result (4.11), we can obtain a well-known conclusion
\[
\|x^k - x^\ast\| \sim \|x^{k+1} - x^\ast\| \sim \|d^k\|.
\]
This together with (4.21) shows that (4.12) holds. □

Remark 7. The statement “the matrix $\nabla^2 f(x^\ast) + \sum_{j \in I} u_j^* \nabla^2 g_j(x^\ast)$ is positive definite” has been used in Theorem 10 (to prove quadratic convergence) of [25], and it holds automatically from our foregoing assumptions if the exact Hessian matrices are used for $k$ sufficiently large, i.e., $G^k = \nabla^2 f(x^k)$ and $G^j = \nabla^2 g_j(x^k), j \in J_k^e$.

5. Extension to general constraints and nonmonotone line search

5.1. Equality and inequality constrained problem

In this subsection, we extend Algorithm A to solve the following equality and inequality constrained optimization problem:
\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad f(x) \\
\text{s.t.} & \quad g_j(x) = 0, \quad j \in E \overset{\text{def}}{=} \{1, \ldots, m_e\}, \\
& \quad g_j(x) \leq 0, \quad j \in I \overset{\text{def}}{=} \{m_e + 1, \ldots, m\}.
\end{align*}
\]
(5.1)
In order that Algorithm A can solve problem (5.1), we only need to modify QCQP subproblems (2.1) and (2.3). Particularly, subproblem (2.1) is generalized as
\[
\begin{align*}
\min_{d \in \mathbb{R}^n} & \quad \nabla f(x^k)^T d + \frac{1}{2} d^T G^k d \\
\text{s.t.} & \quad g_j(x^k) + \nabla g_j(x^k)^T d + \frac{1}{2} d^T G^k d = 0, \quad j \in E, \\
& \quad g_j(x^k) + \nabla g_j(x^k)^T d + \frac{1}{2} d^T G^k d \leq 0, \quad j \in I_k^e, \\
& \quad g_j(x^k) + \nabla g_j(x^{k(j)})^T d + \frac{1}{2} d^T G^j d \leq 0, \quad j \in K_k^e,
\end{align*}
\]
(5.2)
where
\[ I^*_k = \{ j \in I : g_j(x^k) \geq -\varepsilon \text{ or } v^k_j > 0 \} \quad \text{and} \quad K^*_k = I \setminus I^*_k. \]

Accordingly, subproblem (2.3) is generalized as
\[
\begin{align*}
\min_{d \in \mathbb{R}^n, \lambda \in [0, 1]} & \quad \nabla f(x^k)^T d + \frac{1}{2} d^T G^k d + \frac{1}{2} g_k \delta^2 \\
\text{s.t.} & \quad (1 - \delta)g_j(x^k) + \nabla g_j(x^k)^T d + \frac{1}{2} d^T G^k d = 0, \quad j \in E, \\
& \quad (1 - \delta)g_j(x^k) + \nabla g_j(x^k)^T d + \frac{1}{2} d^T G^k d \leq 0, \quad j \in I^*_k, \\
& \quad g_j(x^k) + \nabla g_j(x^{k(j)})^T d + \frac{1}{2} d^T G^{k(j)} d \leq 0, \quad j \in K^*_k.
\end{align*}
\] (5.3)

Define
\[ J^*_k = E \cup I^*_k, \quad J_k = E \cup \{ j \in I : g_j(x^k) \geq -v^k_j/r^k_{j+1} \}. \] (5.4)

And the augmented Lagrangian function (2.7) only needs to modify the index set \( J \) according to \( J_k \) defined in (5.4). After these modifications, by slightly modifying the related proof in Sections 2 and 3, we can conclude that Algorithm A is still globally convergent if there is a constant \( \gamma > 0 \) such that \( (d^k)^T Q^k d^k \geq \gamma \|d^k\|^2 \) (Recall (2.10) for the definition of \( Q^k \)). However, it seems that there are some difficulties in proving superlinear convergence, since we are not able to prove Lemma 4.1(ii) for subproblem (5.2).

5.2. Nonmonotone line search

The nonmonotone line search is often used to design optimization algorithms, since it can improve the efficiency of the original monotone algorithm. The seminal work of nonmonotone line search goes back to Grippo et al. [11] for the unconstrained case, and was extended to constrained optimization by many authors (see [6] for more detailed references), for example, Panier and Tits [19] and Bonnans et al. [4] used nonmonotone line search in SQP methods to avoid the Maratos effect. Here, we use the following nonmonotone line search which has been used in [6] to replace line search (2.17)
\[
\Psi_k(\lambda) \leq \max_{k-l(k) \leq j \leq k} \Psi_j(0) + x^l \Psi'_k(0),
\] (5.5)
where \( l(k) \) is a predetermined parameter with \( l(k) \in \{ 0, \ldots, \min\{k, L\} \} \), \( L \) a given tolerance.

Following the analysis in [6], it is not difficult to prove the global convergence of Algorithm A with nonmonotone line search (5.5).

6. Concluding remarks

In this paper, we have presented a SQCQP method that uses an augmented Lagrangian function in the line search. After early iterations, the algorithm becomes very simple, that is, only a convex QCQP needs to be solved to obtain a search direction. Interesting features of the proposed algorithm are: (i) both the objective function and the constrained functions are not required to be convex; (ii) arbitrary starting points and general constraints (mixed equality and inequality) can be handled; (iii) the global, superlinear, and quadratic convergence are obtained under suitable assumptions.

Acknowledgments

The authors wish to thank the Principal Editor and the anonymous referee for their extensive and helpful comments on the earlier version of this paper.

References


