Sequential Quadratic Programming Method for Nonlinear Second-Order Cone Programming Problems

Guidance

Professor Masao FUKUSHIMA

Hirokazu KATO

2004 Graduate Course

in

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2006

Abstract

Convex programming which includes linear second-order cone programming (LSOCP) and linear semidefinite programming (LSDP) has extensively been studied in the last decade, because of many important applications and desirable theoretical properties. For solving those convex programming problems, efficient interior point algorithms have been proposed and the software implementing those algorithms has been developed. On the other hand, The study of nonlinear second-order cone programming (NSOCP) and nonlinear semidefinite programming (NSDP), which are natural extensions of LSOCP and LSDP, respectively, are much more recent and still in its preliminary phase. However, NSOCP and NSDP are important research subjects, since NSOCP includes an application in the robust optimization of nonlinear programming and NSDP includes an application in the robust control design. In this paper, we propose an SQP algorithm for NSOCP. At every iteration, the algorithm solves a convex second-order cone programming subproblem in which the constraints are linear approximations of the constraints of the original problem and the objective function is a convex quadratic function. The subproblem can be transformed into an LSOCP problem which can be solved by interior point methods. To ensure global convergence, the algorithm employs line search that uses the l_1-penalty function as a merit function to determine the step sizes. Furthermore, we show that our algorithm has a fast local convergence property under some assumptions. We present numerical results to demonstrate the effectiveness of the algorithm.

Contents

1 Introduction

Linear second-order cone programming (LSOCP) [1, 10] and linear semidefinite programming (LSDP) [18, 15] have extensively been studied in the last decade, since they have desirable theoretical properties as well as many important applications. For solving those problems, efficient interior point algorithms have been proposed and the software implementing those algorithms has been developed. On the other hand, nonlinear programming (NLP) has long been studied and a number of effective methods such as sequential quadratic programming methods (SQP) [3] and interior point methods [19] have been proposed. However, the study of nonlinear second-order cone programming (NSOCP) and nonlinear semidefinite programming (NSDP), which are natural extensions of LSOCP and LSDP, respectively, are much more recent and still in its preliminary phase.

Optimality conditions for NSOCP are studied in [5, 4, 6]. Yamashita and Yabe [20] propose an interior point method for NSOCP with line search using a new merit function which combines the barrier function with the potential function. Optimality conditions for NSDP are studied in [14, 4, 6]. Globally convergent algorithms based on SQP method and sequential linearization method have been developed for solving NSDP in [7] and [9], respectively.

In this paper, we propose an SQP algorithm for NSOCP. At every iteration, the algorithm solves a subproblem in which the constraints are linear approximations of the constraints of the original problem and the objective function is a convex quadratic function. The subproblem can be transformed into an LSOCP problem, to which the interior point methods [1, 17] and the simplex method [11] can be applied. To ensure global convergence, the algorithm employs line search that uses the l_1-penalty function as a merit function to determine step sizes.

The organization of this paper is as follows: In Section 2, we formulate the nonlinear second-order cone programming problem. In Subsection 3.1, we describe our SQP algorithm for NSOCP. In Subsection 3.2, we show global convergence of the algorithm. In Subsection 3.3, we consider the local convergence behavior of the algorithm. In Section 4, we present some numerical results. In Section 5, we give the concluding remarks.

The notation used in this paper is as follows: For vector $x \in \Re^{n+1}$, x_0 denotes the first component and $\bar{x}$ is the subvector consisting of the remaining components, that is, $x = \begin{pmatrix} x_0 \\ \bar{x} \end{pmatrix}$. The second-order cone of dimension $n+1$ is defined by $K^{n+1} := \{x \in \Re^{n+1} \mid x_0 \geq \|\bar{x}\|\}$. For simplicity, $(x^T, y^T)^T$ is written as $(x, y)^T$. For vector x, the Euclidean norm is denoted $\|x\| := \sqrt{x^T x}$. Moreover, $o(t)$ is a function satisfying $\lim_{t \to 0} \frac{o(t)}{t} = 0$.

2 Nonlinear Second-Order Cone Program

In this paper, we are interested in the following nonlinear second-order cone program (NSOCP):

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \\ & h(x) \in K, \end{aligned} \tag{1}$$

where $f : \Re^n \to \Re$, $g : \Re^n \to \Re^m$ and $h : \Re^n \to \Re^l$ are twice continuously differentiable functions, K is the Cartesian product of second-order cones given by $K := K^{l_1} \times K^{l_2} \times \cdots \times K^{l_s}$, and $l := l_1 + \cdots + l_s$. Throughout this paper, we denote $h(x) = (h_1(x), \cdots, h_s(x))^T$ and $h_i(x) = (h_{i0}(x), \bar{h}_i(x))^T \in \Re^{l_i} \quad (i = 1, \cdots, s)$.

The following robust optimization problem is an important application of NSOCP [2].

Example 1 *Consider the following problem:*

$$\begin{aligned} \min \quad & p(x) \\ \text{s.t.} \quad & \inf_{\omega \in W} \omega^T q(x) \geq 0, \end{aligned} \tag{2}$$

where $p : \Re^n \to \Re$, $q : \Re^n \to \Re^k$, *and* W *is the set defined by*

$$W := \{\omega_0 + Qr \in \Re^k \mid r \in \Re^{k'}, \ \|r\| \leq 1\}$$

for a given vector $\omega_0 \in \Re^k$ and a given matrix $Q \in \Re^{k \times k'}$. It is not difficult to see that problem (2) *is reformulated as*

$$\begin{array}{ll} \min & p(x) \\ \text{s.t.} & \omega_0^T q(x) - \|Qq(x)\| \geq 0. \end{array}$$

This problem is NSOCP (1) *with $h(x) := (\omega_0^T q(x), Qq(x))^T$ and $K := K^{k'+1}$.*

The Karush-Kuhn-Tucker (KKT) conditions for NSOCP(1) are given by

$$\begin{gathered} \nabla f(x^*) - \nabla g(x^*)\zeta^* - \nabla h(x^*)\eta^* = 0 \\ g(x^*) = 0 \\ h_i(x^*) \in K^{l_i}, \quad \eta_i^* \in K^{l_i} \\ h_i(x^*)^T \eta_i^* = 0, \quad i = 1, \cdots, s, \end{gathered} \tag{3}$$

where $\zeta^* \in \Re^m$ and $\eta_i^* \in \Re^{l_i} (i = 1, \cdots, s)$ are Lagrange multiplier vectors. The KKT conditions are necessary optimality conditions under certain constraint qualifications [4]. We call a vector x^* a stationary point of problem (1) if there exist Lagrange multipliers (ζ^*, η^*) satisfying the KKT conditions (3). In this paper, we assume that there exist a triple (x^*, ζ^*, η^*) satisfying the KKT conditions (3) of problem (1).

3 Sequential Quadratic Programming Algorithm for NSOCP

3.1 Algorithm

In our sequential quadratic programming (SQP) algorithm, we solve the following subproblem at every iteration:

$$\begin{array}{ll} \min & \nabla f(x^k)^T \Delta x + \frac{1}{2} \Delta x^T M_k \Delta x \\ \text{s.t.} & g(x^k) + \nabla g(x^k)^T \Delta x = 0 \\ & h(x^k) + \nabla h(x^k)^T \Delta x \in K, \end{array} \tag{4}$$

where x^k is a current iterate and M_k is a symmetric positive definite matrix approximating the Hessian of Lagrangian function of problem (1) in some sense. The subproblem

(4) is a convex programming problem. Therefore, under certain constraint qualifications, a vector Δx is an optimal solution of (4) if and only if there exist Lagrange multiplier vectors λ and μ satisfying the following KKT conditions for (4).

$$\begin{aligned}\nabla f(x^k) + M_k \Delta x - \nabla g(x^k)\lambda - \nabla h(x^k)\mu = 0 \\ g(x^k) + \nabla g(x^k)^T \Delta x = 0 \\ h_i(x^k) + \nabla h_i(x^k)^T \Delta x \in K^{l_i}, \quad \mu_i \in K^{l_i} \\ (h_i(x^k) + \nabla h_i(x^k)^T \Delta x)^T \mu_i = 0, \quad i = 1, \cdots, s.\end{aligned} \tag{5}$$

Additionally, the subproblem (4) can be transformed into a linear second-order cone programming problem, for which an efficient interior point method is available [1, 17].

Comparing conditions (3) and (5), we readily obtain the next proposition. The proof is straightforward and hence is omitted.

Proposition 1 *Under certain constraint qualifications,* $\Delta x = 0$ *is an optimal solution of subproblem* (4) *if and only if* x^k *is a stationary point of NSOCP* (1) *.*

This proposition allows us to deduce that the SQP algorithm is globally convergent if $\{M_k\}$ is bounded and $\lim_{k\to\infty} \|\Delta x^k\| = 0$, where Δx^k is the solution of subproblem (4). A subproblem (4) may be infeasible, even if the original NSOCP (1) is feasible. In SQP methods for nonlinear programming problems, some remedies to avoid this difficulty have been proposed [3]. In this paper, we simply assume that the subproblem (4) is always feasible and hence has a unique optimal solution Δx^k.

In our algorithm, we use the exact l_1 penalty function as a merit function to determine a step size:

$$P_\alpha(x) := f(x) + \alpha\Big(\sum_{i=1}^{m} |g_i(x)| + \sum_{j=1}^{s} \max\{0, -(h_{j0}(x) - \|\bar{h}_j(x)\|)\}\Big), \tag{6}$$

where $\alpha > 0$ is a penalty parameter.

The last part of this subsection is devoted to describing our algorithm.

Algorithm 1

Step 0 *Choose* $x^0 \in \Re^n$, $\sigma \in (0,1)$, $\beta \in (0,1)$, $\alpha_0 > 0$, $\sigma \in (0,1)$, $\tau > 0$ *and set* $k := 0$.

Step 1 *Choose an $n \times n$ symmetric positive definite matrix M_k. Find the solution Δx^k and the corresponding Lagrange multipliers (λ^k, μ^k) satisfying the KKT conditions* (5) *of subproblem* (4)*. If $\|\Delta x^k\| = 0$, then STOP. Otherwise, go to step 2*

Step 2 *Set the penalty parameter as follows: If $\alpha_k \geq \max\left\{\max_{1\leq i\leq m}|\lambda_i^k|, \max_{1\leq j\leq l}\mu_{j0}^k\right\}$, then $\alpha_{k+1} := \alpha_k$; otherwise, $\alpha_{k+1} := \max\left\{\max_{1\leq i\leq m}|\lambda_i^k|, \max_{1\leq j\leq l}\mu_{j0}^k, \alpha_k\right\} + \tau$.*

Step 3 *Compute the smallest nonnegative integer r satisfying*

$$P_{\alpha_{k+1}}(x^k) - P_{\alpha_{k+1}}(x^k + (\beta)^r \Delta x^k) \geq \sigma(\beta)^r \Delta x^{kT} M_k \Delta x^k, \tag{7}$$

and set the step size $t_k := (\beta)^r$.

Step 4 *Set $x^{k+1} := x^k + t_k \Delta x^k$, $k := k+1$ and go to Step 1*

We may consider this algorithm a generalization of the sequential quadratic programming method for ordinary nonlinear programming problems [8].

3.2 Global Convergence

In this subsection, we show that Algorithm 1 has a global convergence property. For simplicity, we assume $s := 1$. The arguments in what follows apply in a similar manner to the case of $s > 1$. When $s = 1$, the KKT conditions (5) of subproblem (4) can be written

$$\begin{gathered}
\nabla f(x^k) + M_k \Delta x - \nabla g(x^k)\lambda - \nabla h(x^k)\mu = 0 \\
g(x^k) + \nabla g(x^k)^T \Delta x = 0 \\
h(x^k) + \nabla h(x^k)^T \Delta x \in K^l, \quad \mu \in K^l \\
(h(x^k) + \nabla h(x^k)^T \Delta x)^T \mu = 0
\end{gathered} \tag{8}$$

and the penalty function used as a merit function is given by

$$P_\alpha(x) = f(x) + \alpha\left(\sum_{i=1}^m |g_i(x)| + \max\{0, -(h_0(x) - \|\bar{h}(x)\|)\}\right), \tag{9}$$

where $h(x) := (h_0(x), \bar{h}(x))^T$ with $h_0 : \Re^n \to \Re$ and $\bar{h} : \Re^n \to \Re^{l-1}$.

To prove global convergence of Algorithm 1, we make the next two assumptions.

(A.1) At every iteration, subproblem (4) has the optimal solution Δx^k and corresponding Lagrange multiplier vectors (λ^k, μ^k).

(A.2) The generated sequence $\{(x^k, \lambda^k, \mu^k)\}$ is bounded.

When assumption (A.1) holds, subproblem (4) has a unique optimal solution since M_k is a positive definite matrix. Below, we will show that the optimal solution Δx^k of subproblem (4) is a descent direction of the penalty function P_{α_k} at x^k, provided the penalty parameter α_k satisfies $\alpha_k \geq \max\left\{\max_{1\leq i\leq m} |\lambda_i^k|, \mu_0^k\right\}$. Hence we can determine the step size t_k in Step 3 and Algorithm 1 is well defined. Assumption (A.2) is standard in SQP methods for nonlinear programming.

In what follows, we denote

$$\varphi(x) := \max\{0, -(h_0(x) - \|\bar{h}(x)\|)\}$$
$$\psi(x) := \sum_{i=1}^{m} |g_i(x)|.$$

The next lemma gives a formula for the directional derivative of φ.

Lemma 1 *The directional derivative $\varphi'(x; \Delta x)$ of φ at x along the direction $\Delta x = (\Delta x_0, \Delta \bar{x})^T$ is given by*

$$\varphi'(x;\Delta x) = \begin{cases} -\nabla h_0(x)^T \Delta x + \frac{(\nabla \bar{h}(x)\bar{h}(x))^T}{\|\bar{h}(x)\|}\Delta x & \left(\begin{array}{l} h_0(x) < \|\bar{h}(x)\|, \bar{h}(x) \neq 0 \textit{ or}, \\ h_0(x) = \|\bar{h}(x)\| \neq 0 \textit{ and} \\ \nabla h_0(x)^T\Delta x < \frac{(\nabla \bar{h}(x)\bar{h}(x))^T}{\|\bar{h}(x)\|}\Delta x \end{array}\right) \\ -\nabla h_0(x)^T \Delta x + \|\nabla \bar{h}(x)^T \Delta x\| & \left(\begin{array}{l} h_0(x) < \|\bar{h}(x)\|, \bar{h}(x) = 0 \textit{ or}, \\ h_0(x) = \bar{h}(x) = 0 \textit{ and} \\ \nabla h_0(x)^T\Delta x < \|\nabla \bar{h}(x)^T \Delta x\| \end{array}\right) \\ 0 & \text{(otherwise)}. \end{cases}$$

Proof We show this lemma by cases.

(i) If $h_0(x) < \|\bar{h}(x)\|$, then

$$\begin{aligned} \varphi'(x;\Delta x) &= \lim_{t\searrow 0} \frac{1}{t}(-h_0(x + t\Delta x) + \|\bar{h}(x + t\Delta x)\| + h_0(x) - \|\bar{h}(x)\|) \\ &= -\nabla h_0(x)^T\Delta x + \lim_{t\searrow 0}\frac{1}{t}(\|\bar{h}(x + t\Delta x)\| - \|\bar{h}(x)\|) \\ &= \begin{cases} -\nabla h_0(x)^T \Delta x + \frac{(\nabla \bar{h}(x)\bar{h}(x))^T}{\|\bar{h}(x)\|}\Delta x & \left(\bar{h}(x) \neq 0\right) \\ -\nabla h_0(x)^T \Delta x + \|\nabla \bar{h}(x)^T \Delta x\| & \left(\bar{h}(x) = 0\right). \end{cases} \end{aligned}$$

(ii) If $h_0(x) = \|\bar{h}(x)\| = 0$, then

$$
\begin{aligned}
\varphi'(x;\Delta x) &= \lim_{t\searrow 0}\frac{1}{t}\max\left\{0, -(h_0(x+t\Delta x) - \|\bar{h}(x+t\Delta x)\|)\right\} \\
&= \begin{cases} -\nabla h_0(x)^T\Delta x + \|\nabla\bar{h}(x)^T\Delta x\| & \left(\nabla h_0(x)^T\Delta x < \|\nabla\bar{h}(x)^T\Delta x\|\right) \\ 0 & \left(\nabla h_0(x)^T\Delta x \geq \|\nabla\bar{h}(x)^T\Delta x\|\right). \end{cases}
\end{aligned}
$$

(iii) If $h_0(x) = \|\bar{h}(x)\| \neq 0$, then

$$
\begin{aligned}
\varphi'(x;\Delta x) &= \lim_{t\searrow 0}\frac{1}{t}\max\left\{0, (-h_0(x+t\Delta x) + \|\bar{h}(x+t\Delta x)\| + h_0(x) - \|\bar{h}(x)\|)\right\} \\
&= \max\left\{0, -\nabla h_0(x)^T\Delta x + \lim_{t\searrow 0}\frac{1}{t}(\|\bar{h}(x+t\Delta x)\| - \|\bar{h}(x)\|\right\} \\
&= \begin{cases} -\nabla h_0(x)^T\Delta x + \frac{(\nabla\bar{h}(x)\bar{h}(x))^T}{\|\bar{h}(x)\|}\Delta x & \left(\nabla h_0(x)^T\Delta x < \frac{(\nabla\bar{h}(x)\bar{h}(x))^T}{\|\bar{h}(x)\|}\Delta x\right) \\ 0 & \left(\nabla h_0(x)^T\Delta x \geq \frac{(\nabla\bar{h}(x)\bar{h}(x))^T}{\|\bar{h}(x)\|}\Delta x\right). \end{cases}
\end{aligned}
$$

(iv) If $h_0(x) > \|\bar{h}(x)\|$, then $\varphi'(x;\Delta x) = 0$. ∎

In the next lemma, using the directional derivative $\varphi'(x;\Delta x)$ given in Lemma 1, we derive an inequality that is used to prove global convergence of the algorithm.

Lemma 2 *Let Δx^k be the optimal solution of subproblem (4), and (λ^k, μ^k) be corresponding Lagrange multiplier vectors. If $\alpha \geq \mu_0^k$, then the directional derivative $\varphi'(x^k;\Delta x^k)$ of φ at x^k along the direction Δx^k satisfies the inequality*

$$-\mu^{kT}h(x^k) + \alpha\varphi'(x^k;\Delta x^k) \leq 0.$$

Proof Using the formula of $\varphi'(x;\Delta x)$ given in Lemma 1, we show the desired inequality by cases.

(i) If $h_0(x^k) < \|\bar{h}(x^k)\|, \bar{h}(x^k) \neq 0$, then we have

$$
\begin{aligned}
&-\mu^{kT}h(x^k) + \alpha\varphi'(x^k;\Delta x^k) \\
&\quad = -\mu^{kT}h(x^k) + \alpha(-\nabla h_0(x^k)^T\Delta x^k + \frac{(\nabla\bar{h}(x^k)\bar{h}(x^k))^T}{\|\bar{h}(x^k)\|}\Delta x^k) \\
&\quad \leq -\mu^k h(x^k) + \alpha(h_0(x^k) - \|\bar{h}(x^k) + \nabla\bar{h}(x^k)^T\Delta x^k\| + \frac{(\nabla\bar{h}(x^k)\bar{h}(x^k))^T}{\|\bar{h}(x^k)\|}\Delta x^k) \\
&\quad = -\mu^k h(x^k) + \alpha(h_0(x^k) - \|\bar{h}(x^k) + \nabla\bar{h}(x^k)^T\Delta x^k\|
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{\bar{h}(x^k)^T(\bar{h}(x^k)+\nabla\bar{h}(x^k)^T\Delta x^k)}{\|\bar{h}(x^k)\|}-\|\bar{h}(x^k)\|) \\
&\le (\alpha-\mu_0^k)h_0(x^k)-\bar{\mu}^{kT}\bar{h}(x^k)-\alpha\|\bar{h}(x^k)\| \\
&\le (\alpha-\mu_0^k)h_0(x^k)-(\alpha-\|\bar{\mu}^k\|)\|\bar{h}(x^k)\| \\
&\le -(\mu_0^k-\|\bar{\mu}^k\|)\|\bar{h}(x^k)\| \\
&\le 0,
\end{aligned}
$$

where the first inequality holds by $h(x^k)+\nabla h(x^k)^T\Delta x^k \in K^l$ in the KKT conditions of the subproblem, the second and the third inequalities follow from Cauchy-Schwarz inequality, and the fourth and the last inequalities follow form $\alpha \ge \mu_0^k \ge \|\bar{\mu^k}\|$ and $h_0(x^k) < \|\bar{h}(x^k)\|$, $\mu^k \in K^l$, respectively.

(ii) If $h_0(x^k) < \|\bar{h}(x^k)\|, \bar{h}(x^k)=0$, then we have

$$
\begin{aligned}
-\mu^{kT}h(x^k)+\alpha\varphi'(x^k;\Delta x^k) &= -\mu_0^{kT}h_0(x^k)+\alpha(-\nabla h_0(x^k)^T\Delta x^k+\|\nabla\bar{h}(x^k)^T\Delta x^k\|) \\
&\le (\alpha-\mu_0^k)h_0(x^k) \\
&\le 0,
\end{aligned}
$$

where the first inequality follows from $h(x^k)+\nabla h(x^k)^T\Delta x^k \in K^l$ and the last inequality holds by $\alpha \ge \mu_0^k$.

(iii) If $h_0(x^k)=0, \bar{h}(x^k)=0$, then $\nabla h_0(x^k)^T\Delta x^k \ge \|\nabla\bar{h}(x^k)^T\Delta x^k\|$ implies $h_0(x^k)+\nabla h_0(x^k)^T\Delta x^k \ge \|\bar{h}(x^k)+\nabla\bar{h}(x^k)^T\Delta x^k\|$, which in turn implies $\varphi'(x^k;\Delta x^k)=0$ by the formula shown in Lemma (1). Therefore we obtain

$$
-\mu^{kT}h(x^k)+\alpha\varphi'(x^k;\Delta x^k) = -\mu^{kT}h(x^k)=0.
$$

(iv) Suppose $h_0(x^k)=\|\bar{h}(x^k)\|\ne 0$. If $\nabla h_0(x)^T\Delta x < \frac{(\nabla\bar{h}(x)\bar{h}(x))^T}{\|\bar{h}(x)\|}\Delta x$, then similarly to case (i), we have

$$
\begin{aligned}
&-\mu^{kT}h(x^k)+\alpha\varphi'(x^k;\Delta x^k) \\
&\quad= -\mu^{kT}h(x^k)+\alpha(-\nabla h_0(x^k)^T\Delta x^k+\frac{(\nabla\bar{h}(x^k)\bar{h}(x^k))^T}{\|\bar{h}(x^k)\|}\Delta x^k) \\
&\quad\le -\mu^k h(x^k)+\alpha(h_0(x^k)-\|\bar{h}(x^k)+\nabla\bar{h}(x^k)^T\Delta x^k\|+\frac{(\nabla\bar{h}(x^k)\bar{h}(x^k))^T}{\|\bar{h}(x^k)\|}\Delta x^k) \\
&\quad= -\mu^k h(x^k)+\alpha(h_0(x^k)-\|\bar{h}(x^k)+\nabla\bar{h}(x^k)^T\Delta x^k\|
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{\bar{h}(x^k)^T(\bar{h}(x^k)+\nabla\bar{h}(x^k)^T\Delta x^k)}{\|\bar{h}(x^k)\|}-\|\bar{h}(x^k)\|) \\
&\le (\alpha-\mu_0^k)h_0(x^k)-\bar{\mu}^{kT}\bar{h}(x^k)-\alpha\|\bar{h}(x^k)\| \\
&\le (\alpha-\mu_0^k)h_0(x^k)-(\alpha-\|\bar{\mu}^k\|)\|\bar{h}(x^k)\| \\
&= -(\mu_0^k-\|\bar{\mu}^k\|)\|\bar{h}(x^k)\| \\
&\le 0.
\end{aligned}
$$

Otherwise, $\varphi'(x^k, \Delta x^k)$ is equal to 0, so it follows from $\mu^k \in K^l$ and Cauchy-Schwarz inequality that

$$
\begin{aligned}
-\mu^{kT}h(x^k)+\alpha\varphi'(x^k;\Delta x^k) &= -\mu^{kT}h(x^k) \\
&= -\mu_0^k h_0(x^k)-\bar{\mu}^{kT}\bar{h}(x^k) \\
&\le -(\mu_0^k-\|\bar{\mu}^k\|)\|\bar{h}(x^k)\| \\
&\le 0.
\end{aligned}
$$

(v) If $h_0(x^k) > \|\bar{h}(x^k)\|$, then it follows from $\mu^k \in K^l$ and Cauchy-Schwarz inequality that

$$
\begin{aligned}
-\mu^{kT}h(x^k)+\alpha\varphi'(x^k;\Delta x^k) &= -\mu^{kT}h(x^k) \\
&= -\mu_0^k h_0(x^k)-\bar{\mu}^{kT}\bar{h}(x^k) \\
&\le -(h_0(x^k)-\|\bar{h}(x^k)\|)\|\bar{\mu}^k\| \\
&\le 0.
\end{aligned}
$$

■

In the next lemma, we derive an inequality regarding the directional derivative $\psi'(x;\Delta x)$ of the function ψ.

Lemma 3 *Let Δx^k be the optimal solution of subproblem* (4). *Then the directional derivative $\psi'(x^k;\Delta x^k)$ of ψ at x^k along the direction Δx^k satisfies the equality*

$$\psi'(x^k;\Delta x^k) = -\sum_{i=1}^{m}|g_i(x^k)|.$$

Proof By the definition of directional derivatives, we have

$$\psi'(x^k;\Delta x^k) \;=\; \lim_{t\searrow 0}\sum_{i=1}^{m}\frac{1}{t}\left(|g_i(x^k+t\Delta x^k)|-|g_i(x^k)|\right)$$

$$= \lim_{t \searrow 0} \sum_{i=1}^{m} \frac{1}{t} \left(|g_i(x^k) + t\nabla g_i(x^k)^T \Delta x^k + o(t)| - |g_i(x^k)| \right).$$

From the KKT conditions (5), we have $\nabla g_i(x^k)^T \Delta x^k = -g_i(x^k)$, and hence

$$\begin{aligned} \psi'(x^k; \Delta x^k) &= \lim_{t \searrow 0} \sum_{i=1}^{m} \frac{1}{t} \left(|(1-t)g_i(x^k) + o(t)| - |g_i(x^k)| \right) \\ &= -\sum_{i=1}^{m} |g_i(x^k)|. \end{aligned}$$

■

From the above lemmas, we obtain the following lemma.

Lemma 4 *Let Δx^k be the optimal solution of subproblem (4). If $\alpha \geq \max\left\{\max_{1\leq i\leq m} |\lambda_i^k|, \mu_0^k\right\}$, then the directional derivative $P'_\alpha(x^k; \Delta x^k)$ of the penalty function P_α at x^k along the direction Δx^k satisfies the inequality*

$$P'_\alpha(x^k; \Delta x^k) \leq -\Delta x^{kT} M_k \Delta x^k.$$

Proof By the KKT conditions (5) of the subproblem and Lemma 3, we have

$$\begin{aligned} &P'(x^k; \Delta x^k) \\ &\quad = \nabla f(x^k)^T \Delta x^k + \alpha(\psi'(x^k; \Delta x^k) + \varphi'(x^k, \Delta x^k)) \\ &\quad = -\Delta x^{kT} M_k \Delta x^k + \lambda^{kT} \nabla g(x^k)^T \Delta x^k + \mu^{kT} \nabla h(x^k)^T \Delta x^k + \alpha(\psi'(x^k; \Delta x^k) + \varphi'(x^k, \Delta x^k)) \\ &\quad = -\Delta x^{kT} M_k \Delta x^k - \lambda^{kT} g(x^k) - \mu^{kT} h(x^k) + \alpha(-\sum_{i=1}^{m} |g(x^k)| + \varphi'(x^k, \Delta x^k)). \end{aligned}$$

On the other hand, from the inequality $\alpha \geq \max\left\{\max_{1\leq i\leq m} |\lambda_i^k|, \mu_{i0}^k\right\}$, it follows that

$$\begin{aligned} -\lambda^{kT} g(x^k) - \alpha \sum_{i=1}^{m} |g_i(x^k)| &\leq -\sum_{i=1}^{m} (\lambda_i^k + \alpha)|g_i(x^k)| \\ &\leq 0. \end{aligned}$$

which together with Lemma 2 yields the desired inequality. ■

When $\Delta x^k \neq 0$, by Lemma 4 and the positive definiteness of the matrix M_k, we have

$$\begin{aligned} &P_\alpha(x^k) - P_\alpha(x^k + t_k \Delta x^k) - \sigma t_k \Delta x^{kT} M_k \Delta x^k \\ &\quad = -t_k P'_\alpha(x^k; \Delta x^k) + o(t_k) - \sigma t_k \Delta x^{kT} M_k \Delta x^k \\ &\quad \geq (1-\sigma) t_k \Delta x^{kT} M_k \Delta x^k + o(t_k) \\ &\quad > 0 \end{aligned}$$

for any sufficiently small $t_k > 0$. This ensures that we can always determine the step size t_k in Step 3 of Algorithm 1.

In the last part of this subsection, we establish global convergence of Algorithm 1.

Theorem 1 *Suppose that assumptions* (A.1) and (A.2) *hold. Let* $\{(x^k, \lambda^k, \mu^k)\}$ *be a sequence generated by Algorithm* 1*, and* (x^*, λ^*, μ^*) *be any accumulation point. Assume that there exist some positive scalars* γ, Γ *such that*

$$\gamma\|z\|^2 \le z^T M_k z \le \Gamma\|z\|^2, \quad \forall z \in \Re^n, \quad \forall k \in \{0, 1, 2, \cdots\}.$$

Then, (x^*, λ^*, μ^*) *satisfies the KKT conditions* (3) *of NSOCP* (1)

Proof Since $\{M_k\}$ is bounded, we only need to show $\lim_{k\to\infty} \|\Delta x^k\| = 0$ from Proposition 1. First note that, from (A.2) and the way of updating the penalty parameter, α_k stays constant $\bar{\alpha}$ eventually for all k sufficiently large. Consequently, $\{P_{\bar{\alpha}}(x^k)\}$ is monotonically nonincreasing for sufficiently large k. Meanwhile, by (7) and the positive definiteness of M_k, we have

$$P_{\bar{\alpha}}(x^k) - P_{\bar{\alpha}}(x^{k+1}) \ge \sigma t_k \Delta x^{kT} M_k \Delta x^k > 0.$$

Since $\{P_{\bar{\alpha}}(x^k)\}$ is bounded below by (A.2), we have

$$\lim_{k\to\infty} P_{\bar{\alpha}}(x^k) - P_{\bar{\alpha}}(x^{k+1}) = 0.$$

Therefore, it holds that

$$\lim_{k\to\infty} t_k \Delta x^{kT} M_k \Delta x^k = 0.$$

Moreover, it follows from the given assumption that

$$t_k \Delta x^{kT} M_k \Delta x^k \ge t_k \gamma \|\Delta x^k\|^2.$$

Hence, we have $\lim_{k\to\infty} t_k\|\Delta x^k\|^2 = 0$. It clearly holds that $\lim_{k'\to\infty} \|\Delta x^{k'}\| = 0$ for any subsequence $\{\Delta x^{k'}\}$ such that $\liminf_{k'\to\infty} t_{k'} > 0$. Let us consider an arbitrary subsequence $\{t_{k'}\}$ such that $\lim_{k'\to\infty} t_{k'} = 0$. Then, by the Armijo rule in Step 3, we have

$$P_{\bar{\alpha}}(x^{k'}) - P_{\bar{\alpha}}(x^{k'} + \bar{t}_{k'}\Delta x^{k'}) < \sigma \bar{t}_{k'} \Delta x^{k'T} M_{k'} \Delta x^{k'},$$

where $\bar{t}_{k'} := \frac{t_{k'}}{\beta}$. On the other hand, since $P'_{\bar{\alpha}}(x^{k'}; \Delta x^{k'}) \leq -\Delta x^{k'T} M_{k'} \Delta x^{k'}$ by Lemma 4, it follows that

$$P_{\bar{\alpha}}(x^{k'}) - P_{\bar{\alpha}}(x^{k'} + \bar{t}_{k'}\Delta x^{k'}) = -\bar{t}_{k'} P'(x^{k'}; \Delta x^{k'}) + o(\bar{t}_{k'}) \geq \bar{t}_{k'} \Delta x^{k'} M_{k'} \Delta x^{k'} + o(\bar{t}_{k'}).$$

Combining the above inequalities yields $\bar{t}_{k'} \Delta x^{k'} M_{k'} \Delta x^{k'} + o(\bar{t}_{k'}) < \sigma \bar{t}_{k'} \Delta x^{k'} M_{k'} \Delta x^{k'}$, and hence

$$0 > (1-\sigma)\bar{t}_{k'} \Delta x^{k'} M_{k'} \Delta x^{k'} + o(\bar{t}_{k'}) > (1-\sigma)\bar{t}_{k'} \gamma \|\Delta x^{k'}\|^2 + o(\bar{t}_{k'}).$$

Thus we obtain

$$(1-\sigma)\gamma\|\Delta x^{k'}\|^2 + \frac{o(\bar{t}_{k'})}{\bar{t}_{k'}} < 0,$$

which yields $\limsup_{k' \to \infty} \|\Delta x^{k'}\| \leq 0$. Consequently, we have $\lim_{k \to \infty} \|\Delta x^k\| = 0$. ■

3.3 Local Convergence

In this subsection, we consider local behavior of a sequence generated by Algorithm 1. For that purpose, we make use of the results for generalized equations [13].

First note that the KKT conditions of NSOCP (1) can be rewritten as the generalized equation

$$0 \in F(y) + \partial \delta_C(y), \tag{10}$$

where F is a vector valued function and $\partial \delta_C(y)$ is the normal cone of a closed convex set C at y, which is defined by

$$\partial \delta_C(y) := \begin{cases} \emptyset & \text{if} \quad y \notin C \\ \{w \mid w^T(c-y) \leq 0 \quad \forall c \in C\} & \text{if} \quad y \in C. \end{cases}$$

Indeed, by defining the Lagrangian of the NSOCP (1) by

$$L(x, \zeta, \eta) := f(x) - g(x)^T \zeta - h(x)^T \eta,$$

the KKT conditions (3) are represented as

$$0 \in \nabla_x L(x, \zeta, \eta) + \partial \delta_{\Re^n}(x)$$
$$0 \in \nabla_\zeta L(x, \zeta, \eta) + \partial \delta_{\Re^m}(\zeta)$$
$$0 \in \nabla_\eta L(x, \zeta, \eta) + \partial \delta_{K^*}(\eta),$$

where $K^* := \{\eta \in \Re^l \mid \eta^T \xi \geq 0, \forall \xi \in K\}$ is the dual cone of K. Since $\partial\delta_{\Re^n}(x) = \{0\}, \partial\delta_{\Re^m}(\zeta) = \{0\}$ and $K^* = K$, we can rewrite the KKT conditions (3) as the generalized equation (10) with $C := \Re^n \times \Re^m \times K$ and

$$F(y) := \begin{bmatrix} \nabla_x L(x, \zeta, \eta) \\ \nabla_\zeta L(x, \zeta, \eta) \\ \nabla_\eta L(x, \zeta, \eta) \end{bmatrix} \tag{11}$$

where $y := (x, \zeta, \eta)^T$.

On the other hand, if we choose $M_k := \nabla^2_{xx} L(x^k, \lambda^k, \mu^k)$, we can express the KKT conditions of subproblem (4) as

$$\begin{aligned} 0 &\in \nabla_x L(x^k, \lambda, \mu) + \nabla^2_{xx} L(x^k, \lambda^k, \mu^k)\Delta x + \partial\delta_{\Re^n}(x) \\ 0 &\in \nabla_\zeta L(x^k, \lambda, \mu) + \nabla^2_{\zeta x} L(x^k, \lambda^k, \mu^k)\Delta x + \partial\delta_{\Re^m}(\lambda) \\ 0 &\in \nabla_\eta L(x^k, \lambda, \mu) + \nabla^2_{\eta x} L(x^k, \lambda^k, \mu^k)\Delta x + \partial\delta_K(\mu), \end{aligned}$$

which is equivalent to the generalized equation

$$0 \in F(z^k) + F'(z^k)(z - z^k) + \partial\delta_C(z), \tag{12}$$

where $z^k = (x^k, \lambda^k, \mu^k), z = (x^k + \Delta x, \lambda, \mu)$ and F is defined by (11). This can be regarded as the application of Newton's method for the generalized equation (10). Thus, a sequence $\{z^k\}$ generated by (12) is expected to converge fast to a solution of (11). To be more precise, we use the notion of a regular solution [13].

Definition 1 *Let y^* be a solution of the generalized equation* (10) *and F be Fréchet differentiable at y^*. Define the set-valued mapping T by $T(y) := F(y^*) + F'(y^*)(y - y^*) + \partial\delta_C(y)$. If there exist neighborhoods U of* 0 *and V of y^* such that the mapping $T^{-1} \cap V$ is single-valued and Lipschitzian on U, then y^* is called a regular solution of the generalized equation* (10).

We suppose that F is Fréchet differentiable with Lipschitz constant L and the generalized equation (12) at $k = 0$

$$0 \in F(z^0) + F'(z^0)(z - z^0) + \partial\delta_C(z)$$

has a regular solution with Lipschitz constant Λ. Then (12) has a regular solution at every iteration k and the following inequality holds for a sequence $\{z^k\}$ generated by (12) if z^0 is sufficiently close to a regular solution y^* of the generalized equation (10) (see [13]):

$$\|y^* - z^k\| \leq (2^{(l+n+m)}\Lambda L)^{-1}(2\Lambda L\|z^0 - z^1\|)^{(2^k)},$$

which means that the sequence $\{z^k\}$ converges R-quadratically to y^*.

Next we consider the relation between the regularity of a solution and the second-order optimality conditions for NSOCP (1). We recall the notion of nondegeneracy in second-order cone programming [5].

Definition 2 *For given vectors $\hat{w}_i \in K^{l_i}(i = 1, \cdots, s)$, define the functions $\phi_i(x)(i = 1, \cdots, s)$ as follows:*
(i) *if $\hat{w}_i = 0$, then $\phi_i : \Re^{l_i} \to \Re^{l_i}$ and $\phi_i(w_i) := w_i$;*
(ii) *if $\hat{w}_{i0} > \|\bar{\hat{w}}_i\|$, then $\phi_i : \Re^{l_i} \to \Re^0$ and $\phi_i(w_i) := 0$;*
(iii) *if $\hat{w}_{i0} = \|\bar{\hat{w}}_i\| \neq 0$, then $\phi_i : \Re^{l_i} \to \Re^1$ and $\phi_i(w_i) := \|\bar{w}_i\| - w_{i0}$.*

Let x be a feasible solution of NSOCP (1). If the matrix

$$(\nabla g(x), \nabla h_1(x)\nabla\phi_1(h_1(x)), \cdots, \nabla h_s(x)\nabla\phi_s(h_s(x)))$$

has full column rank, then x is said to be nondegenerate. Here, $\nabla h_i(x)\nabla\phi_i(h_i(x)) = \nabla h_i(x)$ if $h_i(x) = 0$, $\nabla h_i(x)\nabla\phi_i(h_i(x)) = -\nabla h_{i0}(x) + \frac{\nabla\bar{h}_i(x)\bar{h}_i(x)}{\|\bar{h}_i(x)\|}$ if $h_{i0}(x) = \|\bar{h}_i(x)\| \neq 0$, and $\nabla h_i(x)\nabla\phi_i(h_i(x))$ is vacuous if $h_{i0}(x) > \|\bar{h}_i(x)\|$.

It is showed in [5] that when a local optimal solution x^* of NSOCP(1) is nondegenerate, (x^*, ζ^*, η^*) is a regular solution of the generalized equation representing the KKT conditions (3) of NSOCP (1) if and only if (x^*, ζ^*, η^*) satisfies the following second-order optimality condition:

$$d^T\nabla^2_{xx}L(x^*, \zeta^*, \eta^*)d + d^T\sum_{i=1}^{s} H_i(x^*, \zeta^*, \eta_i^*)d > 0, \ \forall d \neq 0, d \in C_0(x^*) \cap C_{K^1}(x^*) \cap \cdots \cap C_{K^s}(x^*), \tag{13}$$

where

$$C_0(x^*) \ = \ \{d \in \Re^n \mid \nabla g(x^*)^T d = 0\}$$

and for $i = 1, \cdots, s$

$$C_{K^i}(x^*) = \left\{ d \in \Re^n \left| \begin{array}{ll} \nabla h_i(x^*)^T d = 0 & \text{if } \eta_{i0}^* > \bar{\eta}_i^* \\ \nabla h_i(x^*)^T d \in \text{span}\{R_{l_i}\eta_i^*\} & \text{if } \eta_{i0}^* = \|\bar{\eta}_i^*\| \neq 0, h_i(x^*) = 0 \\ d^T \nabla h_i(x^*)\eta_i^* = 0 & \text{if } \eta_{i0}^* = \|\bar{\eta}_i^*\| \neq 0, h_{i0}(x^*) = \|\bar{h}_i(x^*)\| \neq 0 \\ no \quad condition & otherwise \end{array} \right. \right\},$$

$$H_i(x^*, \zeta^*, \eta_i^*) = \begin{cases} -\frac{\eta_{i0}^*}{h_{i0}(x^*)} \nabla h_i(x^*) R_{l_i} \nabla h_i(x^*)^T & \text{if} \quad h_{i0}(x^*) = \|\bar{h}_i(x^*)\| \neq 0 \\ 0 & otherwise \end{cases}$$

with $R_{l_i} := \begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & -I_{l_i - 1} \end{pmatrix}$. Summarizing the above arguments, we have the next theorem about the local behavior of a sequence $\{(x^k, \zeta^k, \eta^k)\}$ generated by Algorithm 1.

Theorem 2 *Suppose $M_k = \nabla_{xx}^2 L(x^k, \zeta^k, \eta^k)$ and step size t_k is equal to 1 for all $k > \bar{k}$, where $\bar{k}$ is a positive integer. If, for some $k > \bar{k}$, (x^k, ζ^k, η^k) is sufficiently close to a nondegenerate stationary point (x^*, ζ^*, η^*) of NSOCP (1) satisfying the second-order condition (13), then a sequence $\{(x^k, \zeta^k, \eta^k)\}$ generated by Algorithm 1 converges R-quadratically to (x^*, ζ^*, η^*). In particular, $\{x^k\}$ converges R-quadratically to x^*.*

4 Numerical Experiments

We implemented Algorithm 1 in MATLAB (Version 6.5) using the SDPT3-Solver (Version 3.0) [16] to solve the subproblems. The detail of transformation of a subproblem into an LSOCP problem is given in Appendix A.

We set the parameters in Algorithm 1 as follows:

$$\alpha_0 = 1, \quad \tau = 0.01, \quad \sigma = 0.2, \quad \beta = 0.95.$$

The stopping criterion of Algorithm 1 is given by $\|\Delta x^k\| < 10^{-4}$.

Experiment 1. First, we consider the following problem:

$$\begin{array}{ll} \min & x^T C x + \sum_{i=1}^{n} (d_i x_i^4 + f_i x_i) \\ \text{s.t.} & \begin{pmatrix} a_1(e^{x_1} - 1) \\ a_2(e^{x_2} - 1) \\ \vdots \\ a_n(e^{x_n} - 1) \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_s \end{pmatrix} \in K := K^{l_1} \times \cdots, \times K^{l_s}, \end{array} \tag{14}$$

where $a_i, d_i, f_i\,(i = 1, \cdots, n)$ are scalars, $b_j\,(j = 1, \cdots, s)$ are l_j-dimensional vectors, and C is an $n \times n$ symmetric positive semidefinite matrix. Note that $n = l := l_1 + \cdots + l_s$ in this problem. We generate ten problem instances for each of $n = 10, 30, 50$. We determine the constants as follows: a_i, d_i, and f_i $(i = 1, \cdots, n)$ are randomly chosen from the intervals [0,2], [0,2], and $[-1, 1]$, respectively, and C is given by $C := Z^T Z$, where Z is an $n \times n$ matrix whose elements are randomly chosen from the interval [0,1]. Vectors $b_j \in \Re^{l_j}(j = 1, \cdots, s)$ are determined as $b_{j0} = 1, \bar{b}_j = 0$. Then, problem (14) is always feasible, since $x = 0$ is satisfies the constraints. It may be worth noticing that problem (14) is not necessarily a convex programming problem despite the fact that the objective function and the constraint functions are convex, since the feasible region is not a convex set.

Each problem instance is solved by Algorithm 1 using an initial iterate whose elements are randomly generated from the interval $[-1, 1]$. The following two updating formulas for matrices M_k are tested.

Modified Newton formula. At iteration k, if the Hessian $\nabla^2_{xx} L(x^k, \mu^{k-1})$ of the Lagrangian is a positive definite matrix, then set $M_k = \nabla^2_{xx} L(x^k, \mu^{k-1})$; otherwise, set $M_k = \nabla^2_{xx} L(x^k, \mu^{k-1}) + (|\xi_k| + 0.1)I$, where ξ_k is the minimum eigenvalue of $\nabla^2_{xx} L(x^k, \mu^{k-1})$. At the first iteration, M_0 is set to be the identity matrix I.

Quasi-Newton formula. The initial matrix M_0 is set to be the identity matrix. Subsequently, M_k is updated by

$$M_{k+1} = M_k - \frac{M_k v^k v^{kT} M_k}{v^{kT} M_k v^k} + \frac{u^k u^{kT}}{v^{kT} u^k},$$

where $v^k = x^{k+1} - x^k$, $w^k = \nabla_x L(x^{k+1}, \lambda^k, \mu^k) - \nabla_x L(x^k, \lambda^k, \mu^k)$, $u^k = \theta_k w^k + (1 -$

$\theta_k)M_k v^k$, and θ_k is determined by

$$\theta_k = \begin{cases} 1 & \text{if } v^{kT}w^k \geq 0.2 v^{kT} M_k v^k \\ \frac{0.8 v^{kT} M_k v^k}{v^{kT}(M_k v^k - w^k)} & \text{otherwise.} \end{cases}$$

This is a modified BFGS update suggested in the SQP method for NLP [3].

Both update formulas ensure the positive definiteness of M_k for all k. Therefore, if the subproblem is feasible at every iteration, then the sequence generated by Algorithm 1 will converge to a stationary point of NSOCP (1). In our numerical experiments, when a subproblem becomes infeasible at some iteration, we choose a new initial point and solve the problem again.

In our experiments with the modified Newton formula, we observed that, when the sequence generated by Algorithm 1 converged to a stationary point of NSOCP (1), M_k was chosen to be $\nabla^2_{xx} L(x^k, \mu^{k-1})$ and the step size was equal to 1 in the final stage of the iteration. In the case of the quasi-Newton formula, the step size was also equal to 1 in the final stage of the iteration. Tables 1 and 2 show the average k_{ave}, the minimum k_{min}, and the maximum k_{max} numbers of iterations for ten runs, along with the problem size and the Cartesian structure of the second-order cone K of each test problem. We

Table 1: Computational results with the modified Newton formula for problem (14)

n	K	k_{ave}	k_{min}	k_{max}
10	$K^5 \times K^5$	13.05	10	19
30	$K^5 \times K^5 \times K^{20}$	17.32	11	29
50	$K^5 \times K^5 \times K^{20} \times K^{20}$	19.56	10	30

Table 2: Computational results with the quasi-Newton formula for problem (14)

n	K	k_{ave}	k_{min}	k_{max}
10	$K^5 \times K^5$	23.39	14	36
30	$K^5 \times K^5 \times K^{20}$	56.24	26	98
50	$K^5 \times K^5 \times K^{20} \times K^{20}$	67.56	37	91

find that, for problem (14), the modified Newton formula results in faster convergence than the quasi-Newton formula. This suggests that the convexity of the objective and the constraint functions can be better exploited in the modified Newton formula, since it uses the Hessian matrices of those functions in a direct manner.

Experiment 2. Next, we consider the following problem:

$$\begin{array}{ll} \min & x^T C x + \sum_{i=1}^{n} (d_i x_i^4 + e_i x_i^3 + f_i x_i) \\ \text{s.t.} & \begin{pmatrix} a_1(e^{x_1} - 1) \\ a_2(e^{x_2} - 1) \\ \vdots \\ a_n(e^{x_n} - 1) \end{pmatrix} + \begin{pmatrix} \hat{a}_1 x_1 x_2 \\ \hat{a}_2 x_2 x_3 \\ \vdots \\ \hat{a}_n x_n x_1 \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_s \end{pmatrix} \in K := K^{l_1} \times \cdots, \times K^{l_s}, \end{array} \tag{15}$$

where the constants are similar to those in the previous test problem, except that C is an $n \times n$ indefinite matrix. Note that $n = l := l_1 + \cdots + l_s$ in this problem. We generate ten problem instances for each of $n = 10, 30, 50$. We determine the constants as follows: $a_i, \hat{a}_i, e_i, f_i$ $(i = 1, \cdots, n)$ and the elements of C are randomly chosen from the interval $[-1, 1]$, and $d_i (i = 1, \cdots, n)$ are randomly chosen from the interval [0, 1]. Vectors $b_j \in \Re^{l_j} (j = 1, \cdots, s)$ are determined as $b_{j0} = 1, \bar{b}_j = 0$ similarly to the case of problem (14). Then, problem (15) is always feasible. Note that the objective function and the constraint functions are in general nonconvex unlike problem (14).

As in the previous experiment, each problem instance is solved by Algorithm 1 using an initial iterate whose elements are randomly generated from the interval $[-1, 1]$. When a subproblem becomes infeasible at some iteration, we choose a new initial point and solve the problem again. We test the two formulas for updating matrices M_k, the modified Newton formula and the quasi-Newton formula. The results are shown in Tables 3 and 4.

Because of the lack of convexity in the objective and constraint functions, the Hessian $\nabla^2_{xx} L(x^k, \mu^{k-1})$ of the Lagrangian is not likely to be positive definite even if x^k is close to a stationary point of the problem. Thus, the matrices M_k determined by the modified Newton formula may substantially differ from $\nabla^2_{xx} L(x^k, \mu^{k-1})$. We have observed that the algorithm with the modified Newton formula performs somewhat inefficiently compared with the previous experiment, although it exhibits fast local convergence, when $\nabla^2_{xx} L(x^k, \mu^{k-1})$ becomes positive definite near a solution. In fact, the comparison of Table 3 and 4 suggests that the quasi-Newton formula works more effectively especially when $\nabla^2_{xx} L(x^k, \mu^{k-1})$ is indefinite.

Table 3: Computational results with the modified Newton formula for problem (15)

n	K	k_{ave}	k_{min}	k_{max}
10	$K^5 \times K^5$	24.31	11	116
30	$K^5 \times K^5 \times K^{20}$	59.44	19	183
50	$K^5 \times K^5 \times K^{20} \times K^{20}$	68.64	20	180

Table 4: Computational results with the quasi-Newton formula for problem (15)

n	K	k_{ave}	k_{min}	k_{max}
10	$K^5 \times K^5$	24.96	12	56
30	$K^5 \times K^5 \times K^{20}$	39.75	25	91
50	$K^5 \times K^5 \times K^{20} \times K^{20}$	50.22	31	97

5 Concluding remarks

In this paper, we have proposed a sequential quadratic programming method for nonlinear second-order programming problems. We have proved global convergence of the algorithm, and examined its local convergence behavior by reformulating the KKT conditions of NSOCP into the generalized equation. Through numerical experiments, we have confirmed the effectiveness of the algorithm for nonconvex NSOCP. The algorithm presented in this paper is a prototype and it may be further improved in terms of implementation, for example by incorporating a device to deal with infeasible subproblems.

Acknowledgment

First of all, I would like to express my sincere thanks and appreciation to Professor Masao Fukushima. Although I often troubled him due to my greenness, he kindly looked after me and gave me plenty of precise advice. I would also like to tender my acknowledgments to Associate Professor Nobuo Yamashita. He gave me a lot of comment for my research. Finally, I would like to thank all the members in Fukushima's Laboratory.

A Transformation of Subproblem

In this subsection, we transform subproblem (4) to a linear second-order cone program (LSOCP).

We use SDPT3 solver (version 3.0) [16] to solve subproblems in our numerical experiments. This software can solve LSOCPs of the form

$$\begin{aligned} \min \quad & \sum_{i=1}^{n_q} c_j^{qT} x_j^q + c^{lT} x^l \\ \text{s.t.} \quad & \sum_{i=1}^{n_q} A_j^{qT} x_j^q + A^{lT} x^l = b \\ & x_i^q \in K_q^{q_j} \quad i = 1, \cdots, n_q, \quad x^l \geq 0, \end{aligned}$$

where c_i^q, x_i^q are q_i-dimensional vectors, c^l, x^l are n_l-dimensional vectors, and A_i^q, A^l are $q_i \times m, l \times m$ matrices, respectively. To transform subproblem (4) to an LSOCP of this form, we first introduce an auxiliary variable $u \geq 0$ and rewrite the problem as

$$\begin{aligned} \min \quad & \nabla f(x^k)^T \Delta x + u \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^T \Delta x = 0 \\ & u \geq \frac{1}{2} \Delta x^T M_k \Delta x \\ & h(x^k) + \nabla h(x^k)^T \Delta x \in K, \quad u \geq 0, \end{aligned}$$

which can further be rewritten as

$$\begin{aligned} \min \quad & \nabla f(x^k)^T \Delta x + u \\ \text{s.t.} \quad & g(x^k) + \nabla g(x^k)^T \Delta x = 0 \\ & (u+1)^2 \geq (u-1)^2 + 2\|M_k^{\frac{1}{2}} \Delta x\|^2 \\ & h(x^k) + \nabla h(x^k)^T \Delta x \in K, \quad u \geq 0. \end{aligned}$$

Next, by introducing auxiliary variables y, z, and putting $\Delta x = \Delta x_1 - \Delta x_2$ with $\Delta x_1 \geq 0$ and $\Delta x_2 \geq 0$, we rewrite the problem as

$$
\begin{array}{ll}
\min & \nabla f(x^k)^T(\Delta x_1 - \Delta x_2) + u \\
\text{s.t.} & g(x^k) + \nabla g(x^k)^T(\Delta x_1 - \Delta x_2) = 0 \\
& z = \begin{pmatrix} u+1 \\ u-1 \\ \sqrt{2} M_k^{\frac{1}{2}}(\Delta x_1 - \Delta x_2) \end{pmatrix} \\
& y = h(x^k) + \nabla h(x^k)^T(\Delta x_1 - \Delta x_2) \\
& y \in K, \quad z \in K^{n+2}, \quad u \geq 0, \quad \Delta x_1 \geq 0, \quad \Delta x_2 \geq 0,
\end{array}
$$

which is essentially of the standard form LSOCP for the SDPT3 solver. In the numerical experiments, we add the term $\epsilon e^T(\Delta x_1 + \Delta x_2)$ to the objective function to force the condition $\Delta x_1^T \Delta x_2 = 0$ to hold, where $e = (1, \cdots, 1)^T$ and ϵ is a sufficiently small positive number.

References

[1] F. Alizadeh and D. Goldfarb: *Second-order cone programming.* Mathematical Programming, Vol. 95, 2003, pp. 3-51.

[2] A. Ben-Tal and A. Nemirovski: *Robust convex optimization.* Mathematics of Operations Research, Vol. 23, 1998, pp. 769-805.

[3] P. T. Boggs and J. W. Tolle: *Sequential quadratic programming.* Acta Numerica. 4, 1995, pp. 1-51.

[4] J. F. Bonnans, R. Cominetti and A. Shapiro: *Second order optimality conditions based on parabolic second order tangent sets.* SIAM Journal on Optimization, Vol. 9, 1999, pp. 466-492.

[5] J. F. Bonnans and H. Ramirez: *Perturbation analysis of second-order-cone programming problems.* Technical Report, INRIA, Le Chesnay Cedex France, August 2004.

[6] J. F. Bonnans and A. Shapiro: *Perturbation Analysis of Optimization Problems.* Springer-Verlag, New York, 2000.

[7] R. Correa and H. Ramirez: *A global algorithm for nonlinear semidefinite programming.* Research Report 4672, INRIA, Le Chesnay Cedex France, 2002.

[8] S. P. Han: *A globally convergent method for nonlinear programming.* Journal of Optimization Theory and Applications, Vol. 22, 1977, pp. 297-309.

[9] C. Kanzow, C. Nagel, H. Kato and M. Fukushima: *Successive linearization methods for nonlinear semidefinite programs.* Computational Optimization and Applications Vol. 31, 2005, pp. 251-273.

[10] M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret: *Applications of second-order cone programming.* Linear Algebra and Its Applications, Vol. 284, 1998, pp. 193-228.

[11] M. Muramatsu: *A pivoting procedure for a class of second-order programming.* Manuscript, The University of Electro-Communications, Tokyo, 2004.

[12] S. M. Robinson: *Strongly regular generalized equations.* Mathematical Programming Study, Vol. 5, 1980, pp. 43-62.

[13] S. M. Robinson: *Generalized equations.* in A. Bachem et al. (eds.) *Mathematical Programming: The State of the Art,* Springer-Verlag, Berlin, 1983, pp. 346-367.

[14] A. Shapiro: *First and second order analysis of nonlinear semidefinite programs.* Mathematical Programming, Vol. 77, 1999, pp. 301-320.

[15] M. J. Todd: *Semidefinite optimization.* Acta Numerica, Vol. 10, 2001, pp. 515-560.

[16] K. C. Toh, R. H. Tütüncü and M. J. Todd: *SDPT3 version 3.02 - a MATLAB software for semidefinite-quadratic-linear programming.* updated in December 2002, http://www.math.nus.edu.sg/ mattohkc/sdpt3.html

[17] R. H. Tütüncü, K. C. Toh and M. J. Todd: *Solving semidefinite-quadratic-linear programs using SDPT3.* Mathematical Programming Vol. 95, 2003, pp. 189-217.

[18] L. Vandenberghe and S. Boyd: *Semidefinite programming.* SIAM Review Vol. 38, 1996, pp. 49-95.

[19] H. Yamashita: *A globally convergent primal-dual interior method for constrained optimization.* Optimization Methods and Software, Vol. 10, 1998, pp. 443-469.

[20] H. Yamashita and H. Yabe: *A primal-dual interior point method for nonlinear optimization over second order cones.* Manuscript, Mathematical Systems, Inc., Tokyo, 2005.